|  |
| --- |
| **Министерство науки и ВЫСШЕГО ОБРАЗОВАНИЯ**  **Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Уральский государственный экономический университет»**  **(УрГЭУ)** |

**КУРСОВАЯ РАБОТА**

по дисциплине

**«Проектирование автоматизированных систем управления»**

**Тема: ПРОГРАММНЫЙ КОМПЛЕКС «Терминал выдачи Товарно-материальных ценностей»**

|  |  |  |
| --- | --- | --- |
| Институт непрерывного  и дистанционного образования  Направление подготовки  *09.03.01 Информатика и вычислительная техника*  Направленность (профиль)  *Автоматизированные системы управления производством*  Кафедра  *Информационных технологий и статистики*  Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Студент *М.С. Рублев*  Группа *ИНО ЗБ ИВТ-19-2*  Руководитель  *Преподаватель С.А. Власов* |

Екатеринбург

2021 г.

# СОДЕРЖАНИЕ

Введение 3

1. Организационная культура 5

1.1. Анализ предметной области 5

1.2 Анализ организации прикладных и информационных процессов учета данных приема-выдачи ключей 6

1.2.1 Описание существующей организации прикладных процессов 6

1.3 Постановка задачи автоматизации (информатизации) 8

1.3.1 Цели и задачи проекта автоматизации 8

1.3.2 Построение и обоснование модели новой организации 9

1.4 Обоснование выбора инструментальной среды 10

2 Проектная часть 15

2.1 Информационное обеспечение системы 15

2.1.1 Инфологическая модель и схема данных 15

2.1.2 Входная информация 19

2.1.3 Классификаторы и нормативно-справочная информация 20

2.1.4 Выходная информация 21

2.2 Программное обеспечение информационной системы 21

2.2.1 Структура программного обеспечения 21

2.2.2 Спецификации программных модулей 24

2.3 Графический интерфейс приложения 27

2.4 Тестирование системы и оценка качества 34

Заключение 40

Список используемых источников 43

Приложение 48

# ВВЕДЕНИЕ

Информационные технологии в жизни человека неотъемлемая часть современного мира, так и для любой компании использовать современные технологии — это важный процесс взаимодействия с миром, а главное это автоматизация рутинных действий и сокращение расходов на обслуживание своих систем.

Важной составляющей информационных технологий организаций, является наличие системы учета данных, благодаря которым можно провести анализ деятельности, и вести контроль по организации определённых бизнес-процессов организации или отдельных отделов организации. Данное направление можно назвать главной задачей оптимизации этого бизнес-процесса.

Оптимизация учета данных, а точнее выдача материальных ценностей хранящихся на складе организации, это важный и необходимый процесс в организации, без контроля и оперативного управления складом материальных ценностей невозможно следить за состоянием и наличием ТМЦ на складе. Конечно использование специальных программ подобных 1С Предприятие дает большой спектр возможностей и контроля ТМЦ, но для разгрузки руководителя и оптимизации бизнес процесса можно использовать разработанный программный комплекс или как его можно представить «Терминал выдачи ТМЦ», без предоставления прав доступа сотрудникам к 1С Предприятию, и с дружественным интерфейсом.

Программный комплекс «Терминал выдачи ТМЦ» позволит сотруднику самостоятельно заполнять наряд и получать необходимое ТМЦ на складе, без участия руководителя отдела и доступа к 1С предприятие, тем самым не отвлекая руководителя отдела от его прямых обязанностей по контролю и планированию работ отдела в нагруженные часы. Программа позволяет подготовить наряд на выдачу материала, получить ТМЦ на складе и оформить запись о получении ТМЦ в базе данных. Тем самым сотрудник самостоятельно определяет необходимые ТМЦ для работы и принимает решение о необходимости получения материалов. Правильно разработанная система, намного повышает эффективность управления отделом и складом, разгружая руководителя от необходимости каждый раз выписывать ТМЦ и повышая ответственность сотрудников.

Цель курсовой работы состоит в создании программного комплекса «Терминал выдачи ТМЦ», обеспечивающего использование структурированной базы данных для автоматизации деятельности и обеспечения хранения, накопления, предоставления, редактирования информации для выдачи материала сотруднику отдела. Данная программа должна иметь возможность добавления, удаления информации о материальных ценностях, о сотрудниках, категориях ТМЦ, вести информационный журнал статистики, логирование действий выдачи и производить поиск в системе по параметрам.

Задачами, которые необходимо решить в данной работе являются:

* Рассмотрение общих понятий информационной системы
* Рассмотрение и обоснование используемых технологий для разработки системы
* Определить основные цели внедрения информационной системы
* Сократить издержки времени на взаимодействие смежных отделов
* Сократить время реакции на принятие решения на выполнение поставленной задачи
* Оптимизация бизнес-процессов организации

# 1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

# 1.1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Анализ предметной области – важнейший этап разработки программного обеспечения.

Анализ работы организации позволяет определить ключевые понятия, выяснить задачи, требующие решения в рамках проекта, проанализировать функциональные требования будущего продукта (набор функций, предоставляемых программой) и нефункциональные (надёжность, эффективность, изучаемость, модифицируемость и т.п.).

Компонентами данной предметной области являются сотрудники, записи, материалы, категории ТМЦ. Данное приложение должно позволять администратору создавать записи, удалять, редактировать, сохранять и находить записи, создаваемые в программе. Пользователю выписывать наряд для получения ТМЦ на складе без участия руководителя отдела.

Нефункциональные требования к программному средству:

* надежность: программа должна быть автономной;
* эффективность: программа должна иметь минимальные требования к аппаратному обеспечению. Необходим принтер для печати нарядов, так же программа может работать без использования принтера
* изучаемость: программа должна быть интуитивно понятна, иметь удобный пользовательский интерфейс
* модифицируемость: программа должна быть легко модифицируемой вследствие небольшого исходного размера и объектно-ориентированного подхода.

# 1.2 АНАЛИЗ ОРГАНИЗАЦИИ ПРИКЛАДНЫХ И ИНФОРМАЦИОННЫХ ПРОЦЕССОВ УЧЕТА ДАННЫХ ПРИЕМА-ВЫДАЧИ КЛЮЧЕЙ

# 1.2.1 ОПИСАНИЕ СУЩЕСТВУЮЩЕЙ ОРГАНИЗАЦИИ ПРИКЛАДНЫХ ПРОЦЕССОВ

Организация процессов выдачи ТМЦ, в организациях везде разный, рассмотрим наиболее часто встречающийся процесс выдачи ТМЦ в организациях, а именно фиксирование информации в программном комплексе 1С Предприятие в электронном виде непосредственно через руководителя отдела.

Под организацией прикладных процессов можно понять, что это совокупность определенных действий, направленных на получение конечного, естественно положительно результата, в нашем случае в рассматриваемой организации положительным результатом считается созданный наряд на сотрудника отдела самим сотрудником, без привлечения руководителя отдела.

Далее рассмотрен стандартный сценарий деятельности организации. В начале рабочего дня сотрудник отдела запрашивает у руководителя отдела наряд ТМЦ для работы по заявкам на день/неделю/месяц. Руководитель отвлекается от основной деятельности управления и планирования работ отделом. Запускает программу 1С Предприятие, создает наряд на получение ТМЦ, запрашивает у сотрудника наименование ТМЦ необходимое сотруднику, заполняет наряд и отправляет в печать, ставит подпись и выдает сотруднику наряд. Простая схема выдачи ТМЦ, но при этом отнимающая 3-5 минут времени от планирования и контроля в самые важные часы работы руководителя. Тем самым руководитель может совершить ошибку в планировании работ. В течении дня приблизительное время потери от неэффективности системы выдачи наряда руководитель теряет от 30 минут до 60 минут в рабочий день, что может привести к ошибкам в планировании и несвоевременном решении допущенных ошибок в работе руководителя или сотрудников отдела.

Далее отображен бизнес-процесс на диаграмме с применением методологии функционального моделирования IDEF0.

Диаграмма описанных выше организации прикладных процессов с использованием методологии функционального моделирования IDEF0 приведена на рисунке 1 и 2

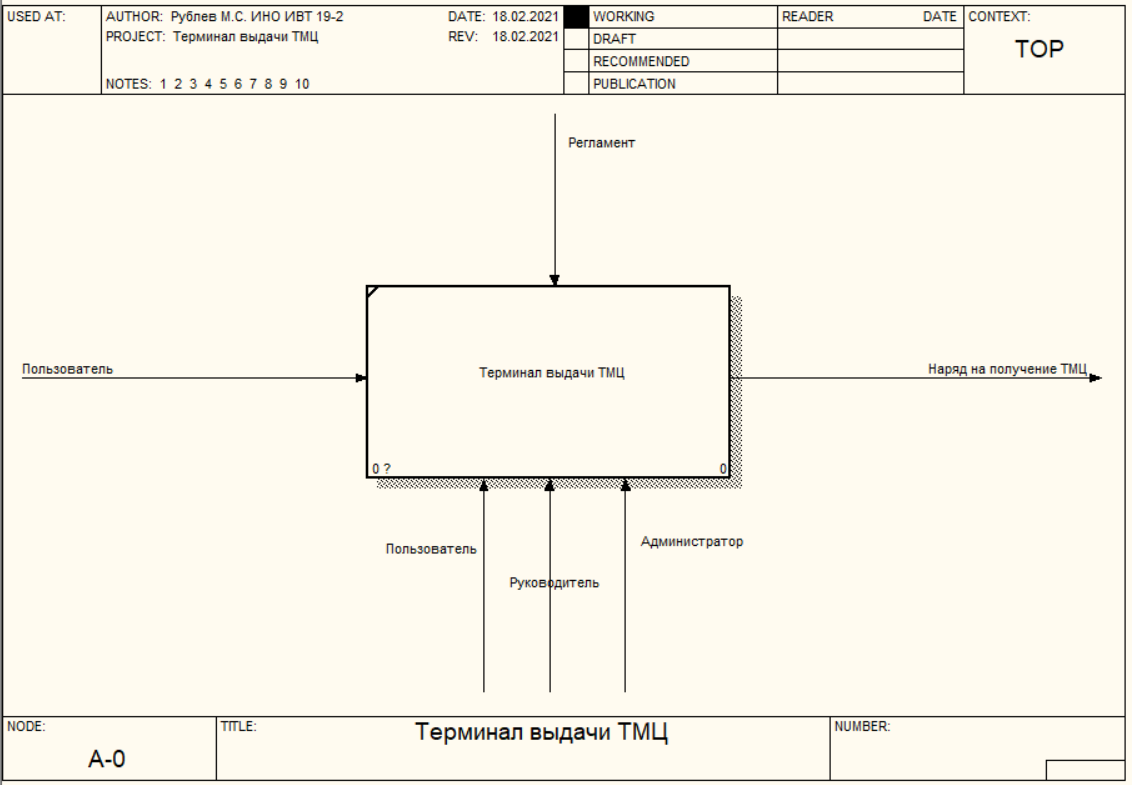


Рисунок 1 – Диаграмма прикладных процессов[[1]](#footnote-1)

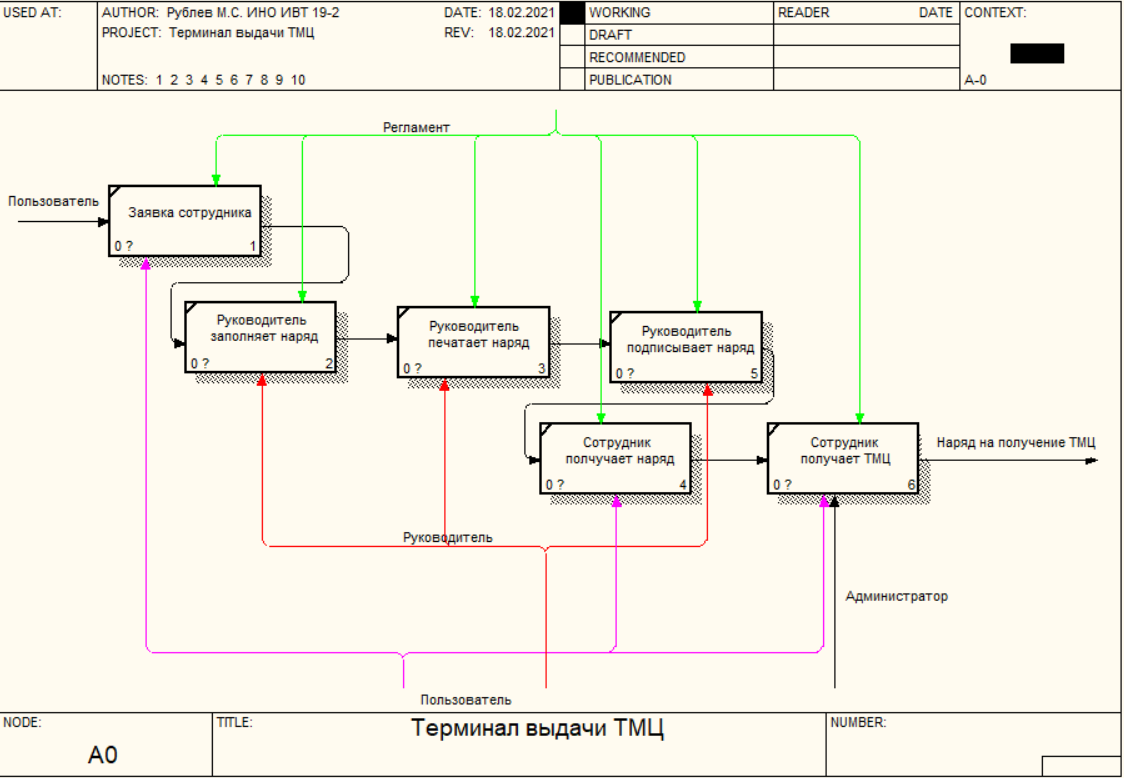


Рисунок 2 – Диаграмма прикладных процессов[[2]](#footnote-2)

Таким образом, заявка от сотрудника, созданная после обращения, проходит руководителя и некоторое количество процессов. Контроль выполнения заявки осуществляется руководителем отдела.

# 1.3 ПОСТАНОВКА ЗАДАЧИ АВТОМАТИЗАЦИИ

# 1.3.1 ЦЕЛИ И ЗАДАЧИ ПРОЕКТА АВТОМАТИЗАЦИИ

Проектируемая информационная система создаётся для автоматизации процесса выдачи товарно-материальных ценностей без участия руководителя отдела.

Целью автоматизации данного процесса является контроль, организация поиска необходимой информации, создание информационного пространства, самостоятельное заполнение нарядов сотрудником для получения ТМЦ на складе.

Создание информационной системы «Терминал выдачи ТМЦ» преследует следующие цели:

* Уменьшение времени заполнения заявки на выдачу ТМЦ
* Учет выданных ТМЦ
* Снятие нагрузки с руководителя
* Повышения ответственности и самостоятельности сотрудников отдела

Задачами, которые должна решать информационная система являются:

* Оптимизация поиска, обработки и хранения информации
* Оптимизация предоставляемой информации информационной системой
* Создание единой информационной системы для всех участников процессов решений и взаимодействия
* Исключение дублирования данных в системе, избыточности
* Защита данных от несанкционированного доступа, потери данных
* Исключение человеческого фактора на работоспособность системы
* Предоставление статистики
* Предоставление данных о сотрудниках

# 1.3.2 ПОСТРОЕНИЕ И ОБОСНОВАНИЕ МОДЕЛИ НОВОЙ ОРГАНИЗАЦИИ

В проектируемой информационной системе предлагается использование новой модели бизнес-процессов. Руководитель отдела заполняет систему данными о сотруднике и товарно-материальных ценностях, идеальным вариантом является получение информации из 1С Предприятия о ТМЦ и сотрудниках, в рассматриваемом примере упор сделан на базы данных MSSQL. В случае, если сотрудник не внесен в информационную систему, руководитель (администратор) добавляет нового сотрудника в систему. Сотрудник авторизовывается в системе, система открывает модуль заполнения наряда для получения ТМЦ. Далее сотрудник заполнив наряд на ТМЦ отправляет наряд в печать при необходимости, далее идет на склад для получения ТМЦ. Информация фиксируется в журнале статистики. Если необходимое ТМЦ отсутствует на складе, сотруднику предоставляется информация о том, что выбранное ТМЦ отсутствует и выдача его не возможна, в этом случае привлекается руководитель для создания наряда при аварийных работах (только в ручном режиме). Руководитель в 1С Предприятии создает заявку на пополнение ТМЦ на складе. Для удобства поиска ТМЦ используется система поиска и фильтрации данных.

Данная система позволяет сотруднику самостоятельно получать информацию о номенклатуре и наличии ТМЦ, а также выписывать ТМЦ самостоятельно на себя. Что значительно ускоряет процесс выдачи товарно-материальных ценностей и повышает контроль за соблюдение регламента, предотвращает человеческий фактор при заполнении наряда ТМЦ, утерю ТМЦ сотрудником и планировании работ.

Далее представлена диаграмма описанного выше бизнес-процесса с использованием методологии функционального моделирования IDEF0, рисунке 3.

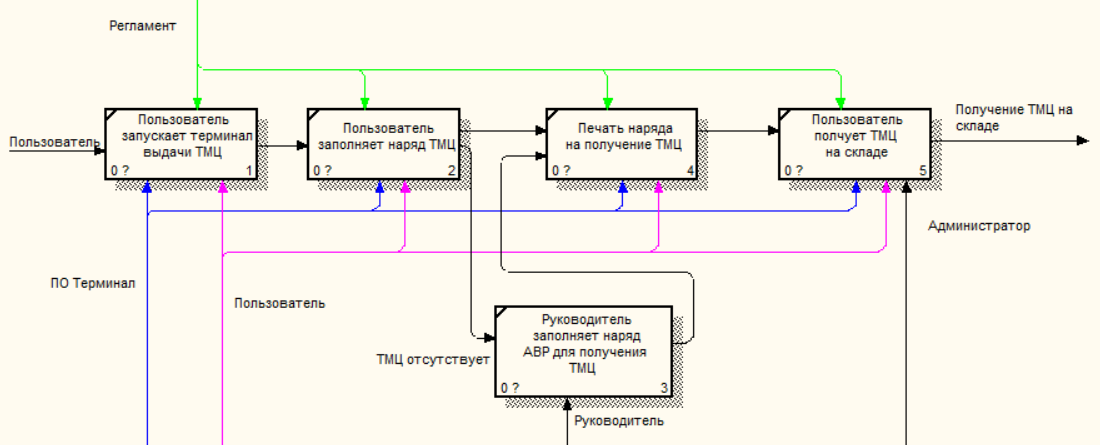


Рисунок 3 – Диаграмма новых прикладных процессов[[3]](#footnote-3)

# 1.4 ОБОСНОВАНИЕ ВЫБОРА ИНСТРУМЕНТАЛЬНОЙ СРЕДЫ

При выполнении курсовой работы, для разработки информационной системы были применены технологии компании Microsoft, в частности, использована бесплатная платформа разработки .NET Framework, которая имеет открытый код. Для разработки информационной системы была выбрана архитектура Windows Forms, как простая и удобная система моделирования интерфейса.

Программа выполнена с помощью программы Microsoft Visual Studio 2019 с подключенной к ней базой данных, выполненной в Microsoft SQL Server 2017 , так как эти программные продукты обладают широкими возможностями для создания различных программ и возможностью соединения проектов между собой.

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Visual Studio включает один или несколько компонентов из следующих:

* Visual Basic .NET;
* Visual C++;
* Visual C#;
* Visual F# (включён начиная с Visual Studio 2010);
* Microsoft SQL Server либо Microsoft SQL Server Express.
* Microsoft SQL Server 2017 — это надежная, производительная и интеллектуальная платформа данных, способная отвечать нуждам наиболее ресурсоемких бизнес-приложений. Она позволяет сократить время и издержки на разработку и сопровождение приложений, а также предоставлять практически применимую информацию на каждое рабочее место предприятия.
* Система SQL Server отталкивается от концепции платформы данных Майкрософт: она упрощает управление любыми данными в любом месте и в любой момент времени. Система позволяет хранить в базах данных информацию, полученную из структурированных, полуструктурированных и неструктурированных источников, таких как изображения и музыка. В SQL Server имеется большой набор интегрированных служб, расширяющих возможности использования данных: можно составлять запросы, выполнять поиск, проводить синхронизацию, делать отчеты, анализировать данные. Все данные хранятся на основных серверах, входящих в состав центра обработки данных. К ним осуществляется доступ с настольных компьютеров и мобильных устройств. Таким образом, пользователь полностью контролируете данные независимо от того, где их сохранили.
* Система SQL Server позволяет обращаться к данным из любого приложения, разработанного с применением технологий Microsoft .NET и Visual Studio.
* SQL Server обеспечивает высокий уровень безопасности, надежности и масштабируемости для критически важных приложений.
* Данные программные продукты обладают огромными функциональными возможностями для проектирования, создания, редактирования приложений различного типа и сложности. Они обладают понятным интерфейсом управления, широкой визуализацией проектирования и возможностью соединения с другими программами.»
* В Visual Studio имеется множество встроенных мастеров и дизайнеров, которые помогут быстро и эффективно создать архитектуру доступа к данным во время разработки и оснастить приложение надежным механизмом доступа к данным, затратив минимум усилий на написание кода. Наряду с этим все возможности объектной модели ADO.NET доступны программно, что позволяет реализовать нестандартные функции или создавать приложения, ориентированные на нужды пользователя.

За доступ к данным в разрабатываем программном комплексе отвечает Entity Framework 6, метод Code First. Суть данного подхода состоит в том, что сначала делается модель, а потом по ней создается база данных.

Entity Framework 6 (EF6) — это проверенное средство объектно-реляционного сопоставления (O/RM) для .NET, которое разрабатывалось и совершенствовалось в течение нескольких лет.

В качестве O/RM EF6 уменьшает несогласованность между реляционным и объектно-ориентированным мирами, позволяя разработчикам создавать приложения, которые взаимодействуют с данными, хранящимися в реляционных базах данных, с помощью строго типизированных объектов .NET, представляющих прикладную область, и устраняя необходимость писать большой объем инфраструктурного кода для доступа к данным.

В EF6 реализованы многие популярные возможности O/RM:

* Сопоставление классов сущностей [POCO](https://docs.microsoft.com/ru-ru/ef/ef6/resources/glossary#poco) не зависит ни от каких типов EF
* Автоматическое отслеживание изменений.
* Разрешение идентификаторов и единицы работы.
* Безотложная, отложенная и явная загрузка.
* Преобразование строго типизированных запросов с помощью [LINQ (Language INtegrated Query)](https://aka.ms/AA6hsvu)
* Широкие возможности сопоставления, включая поддержку следующих элементов:
  + Отношения «один к одному», «один ко многим» и «многие ко многим»
  + Наследование (одна таблица на иерархию, одна таблица на тип и одна таблица на конкретный класс).
  + Сложные типы
  + Хранимые процедуры
* Визуальный конструктор для создания моделей сущностей.
* Интерфейс класс создается при создании базы через Entity Framework методом Model First для создания моделей сущностей путем написания кода.
* Модели можно создавать на основе существующих баз данных и затем редактировать вручную или же создавать с нуля и затем использовать для создания новых баз данных.
* Интеграция с моделями приложений .NET Framework, включая ASP.NET, и через привязку данных с помощью WPF и WinForms.
* Возможность подключения к базам данных SQL Server, Oracle, MySQL, SQLite, PostgreSQL, DB2 и т. д. на основе ADO.NET и различных [поставщиков](https://docs.microsoft.com/ru-ru/ef/ef6/fundamentals/providers/).

Модель EF всегда состоит из нескольких компонентов:

* Объекты предметной области приложения или сами типы сущностей. Этот уровень часто называют уровнем объектов.
* Концептуальная модель, состоящая из соответствующих предметной области типов сущностей и связей, описываемых с помощью [модели EDM](https://docs.microsoft.com/ru-ru/ef/ef6/resources/glossary#entity-data-model). Этот уровень часто указывается с буквой C, означающей conceptual (концептуальный) .
* Модель хранения, представляющая таблицы, столбцы и связи, как определено в базе данных. Этот уровень часто указывается с буквой S, означающей S (хранение) .
* Сопоставление концептуальной модели со схемой базы данных. Это сопоставление, часто называют сопоставлением C-S.

# 2 ПРОЕКТНАЯ ЧАСТЬ

# 2.1 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ

# 2.1.1 ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ И СХЕМА ДАННЫХ

Целью проектирования инфологической модели является структурирование информационной модели предметной области для информационной системы.

Для информационной системы «Терминал выдачи ТМЦ» на основе проведенного анализа данных были выделены следующие сущности:

* Статистика: сущность содержит в себе информацию для сбора статистики.
* Справочник материалы: сущность содержит информацию о товарно-материальных ценностях, т.е. наименование ТМЦ, производителя, категорию оборудования, количество на складе, цена за одну позицию
* Справочник категория ТМЦ: сущность содержит информацию о категории ТМЦ
* Справочник пользователи: сущность содержит в себе информацию о сотрудниках

В процессе инфологического проектирование базы данных для ИС были созданы таблицы сущностей предметной области, которые представлены ниже.

Таблица 1 – Сущность «Журнал»[[4]](#footnote-4)

|  |  |  |
| --- | --- | --- |
| Название | Ключ сущности | Атрибуты сущности |
| Статистика  (BillTbl) | Код журнала | Имя пользователя  Имя клиента  Количество тмц |

Таблица 2 – Сущность «Справочник материалы»[[5]](#footnote-5)

|  |  |  |
| --- | --- | --- |
| Название | Ключ сущности | Атрибуты сущности |
| Справочник материалы  (BookTbl) | Код материала | Наименование  Производитель  Категория  Количество  Цена |

Таблица 3 – Сущность «Серия»[[6]](#footnote-6)

|  |  |  |
| --- | --- | --- |
| Название | Ключ сущности | Атрибуты сущности |
| Справочник серия  (Seria) | Код серии | Наименование |

Таблица 4 – Сущность «Справочник пользователи»[[7]](#footnote-7)

|  |  |  |
| --- | --- | --- |
| Название | Ключ сущности | Атрибуты сущности |
| Справочник пользователи  (UserTbl) | Код пользователя | Логин  Телефон  Адрес  Пароль |

Следующим шагом является создание логической структуры реляционной базы, где каждый объект модели отображается в соответствующей реляционной таблице. Структура таких таблиц состоит из соответствующего информационного объекта. Ключевой реквизит объекта образует уникальный ключ. Для каждого столба проектируемого объекта указывается формат и размер данных.

Далее представлены структуры проектируемых объектов в таблицах:

Таблица 5 – Структура таблицы «Статистика» (BillTbl)[[8]](#footnote-8)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Наименование поля | Тип поля |
| Ключ | BillId | Числовой |
|  | Uname | Текстовый |
|  | ClientName | Текстовый |
|  | Amount | Числовой |

Таблица 6 – Структура таблицы «Справочник материалы» (BookTbl)[[9]](#footnote-9)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Наименование поля | Тип поля |
| Ключ | BId | Числовой |
|  | BTitle | Текстовый |
|  | BAuthor | Текстовый |
|  | BCat | Текстовый |
|  | BQty | Числовой |
|  | BPrice | Числовой |

Таблица 7 – Структура таблицы «Справочник Серия» (Seria)[[10]](#footnote-10)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Наименование поля | Тип поля |
| Ключ | Id | Числовой |
|  | Name | Текстовый |

Таблица 8 – Структура таблицы «Справочник сотрудники» (UserTbl)[[11]](#footnote-11)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Наименование поля | Тип поля |
| Ключ | UId | Числовой |
|  | UName | Текстовый |
|  | UPhone | Текстовый |
|  | UAdd | Текстовый |
|  | UPass | Текстовый |

Следующий шаг в проектировании информационной системы в логической схеме данных, которая представлена на рисунке 4.

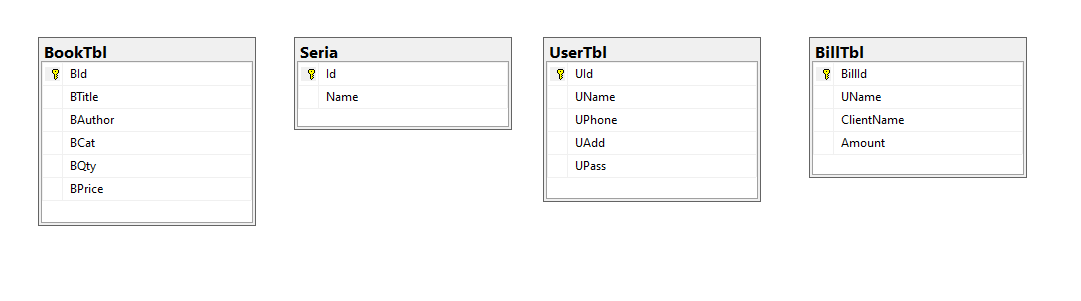


Рисунок 4 - Логическая схема данных[[12]](#footnote-12)

Логическая схема данных при проектировании информационной системы является одним из главных факторов которая определяет в итоге эффективность программы. Данная схема построена на основе анализа вхождения в программу информационных значений и оптимальной схеме взаимосвязи единиц с учётом минимизации объема выделяемой памяти и времени обработки данных.

# 2.1.2 ВХОДНАЯ ИНФОРМАЦИЯ

В процессе исследования работы отдела организации, была выявлена входная информация, необходимая для полноценного функционирования ИС. Источниками каждого входного потока являются различные процессы и объекты деятельности организации, далее в таблице № 9 представлены входные информационные потоки:

Таблица 9 – Входные информационные потоки[[13]](#footnote-13)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование информационного потока | Обозначение документа/потока | Периодичность поступления | Источник информации | Пользователи источника |
| Обращение/Действия сотрудника | Заявка | При обращении сотрудника | Сотрудник организации | Сотрудник, Руководители |

Информационные потоки, по отношению к информационной системе являются внешними, могут быть выполнены на разных носителях таких как бумажные документы, электронные заявки, а также устными, при общении по непосредственно к руководителю, либо самостоятельно сотрудником.

# 2.1.3 КЛАССИФИКАТОРЫ И НОРМАТИВНО-СПРАВОЧНАЯ ИНФОРМАЦИЯ

Логическая схема данных и представленные выше таблицы имеют справочники, которые представлены в приведенных ниже таблицах:

Таблица 10 – Структура таблицы «Справочник материалы» (BookTbl)[[14]](#footnote-14)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Наименование поля | Тип поля |
| Ключ | BId | Числовой |
|  | BTitle | Текстовый |
|  | BAuthor | Текстовый |
|  | BCat | Текстовый |
|  | BQty | Числовой |
|  | BPrice | Числовой |

Таблица 11 – Структура таблицы «Справочник Серия» (Seria)[[15]](#footnote-15)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Наименование поля | Тип поля |
| Ключ | Id | Числовой |
|  | Name | Текстовый |

Таблица 12 – Структура таблицы «Справочник сотрудники» (UserTbl)[[16]](#footnote-16)

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Наименование поля | Тип поля |
| Ключ | UId | Числовой |
|  | UName | Текстовый |
|  | UPhone | Текстовый |
|  | UAdd | Текстовый |
|  | UPass | Текстовый |

# 2.1.4 ВЫХОДНАЯ ИНФОРМАЦИЯ

Выходные документы информационной системы формируются в процессе её работы. Далее в таблице 13 приведен перечень выходных документов.

Таблица 13 – Перечень выходных информационных потоков[[17]](#footnote-17)

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование документа | Обозначение документа/потока | Периодичность формирования | Получатель |
| Список необходимого тмц | Таблица выдачи ТМЦ | По запросу Пользователя | Пользователь |
| Наряд для склада | Формирование наряда и печать | По запросу Пользователя | Пользователь организации |

Выходные информационные потоки представляют собой результирующие документы определенного вида, могут быть электронными таблицами, электронным письмом, отчётами, предоставляемыми в информационном окне программы.

# 2.2 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

# 2.2.1 СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Информационная система, имеет ряд следующих основных программных классов, которые описывают структурную составляющую программы и отображают основные и важные функции и методы. Далее рассматриваются следующие основные классы:

* Класс «AdminLoginForm» – «Форма авторизации» – данный класс предназначен для авторизации в системе администратора приложения, после авторизации администратору открывается окно приложения с формой управления – создание, редактирование, удаление записей в системе.
* Класс «Login» – «Форма авторизации» - класс формы предназначен для авторизации в рабочем режиме программы, т.е. для предназначен для конечных пользователей системы, после авторизации пользователь доступна форма наряда для ТМЦ.
* Класс «Billing» – «Форма наряда» – класс формы наряда для выписывания тмц, в этом классе реализован функционал заполнения наряда, поиска материалов и вывода наряда на печать.
* Класс «Materials» – «Справочник материалы» – класс формы материалы, выводит из базы данных информацию по ТМЦ. В своей структуре класс имеет функции поиска, добавления, редактирования, изменения, удаления данных о материальных ценностях (ТМЦ).
* Класс «Users» – «Справочник пользователи» – класс выводит в таблицу информацию о пользователях системы, В своей структуре класс имеет функции поиска, добавления, редактирования, изменения, удаления данных о пользователе.
* Класс «DashBoard» – «Класс статистики», класс формы собирает в едином месте информацию о операциях, количестве сотрудников в базе данных и стоимость выданного материала.
* Класс «ModelKeyBTcontext» - Является базовым классом Entity Framework и предоставляет широкие возможности по работе с базой данных: создание запросов, отслеживание изменений и сохранение данных в базе.
* Класс «BillTbl» – описывает модель таблицы базы данных, со своими полями.
* Класс «Materials» – класс создается при создании базы через Entity Framework методом Code First, описывает модель таблицы материалы.
* Класс «Seria» – класс создается при создании базы через Entity Framework методом Code First, описывает модель таблицы категории.
* Класс «UserTbl» – класс создается при создании базы через Entity Framework методом Code First, описывает модель таблицы пользователи

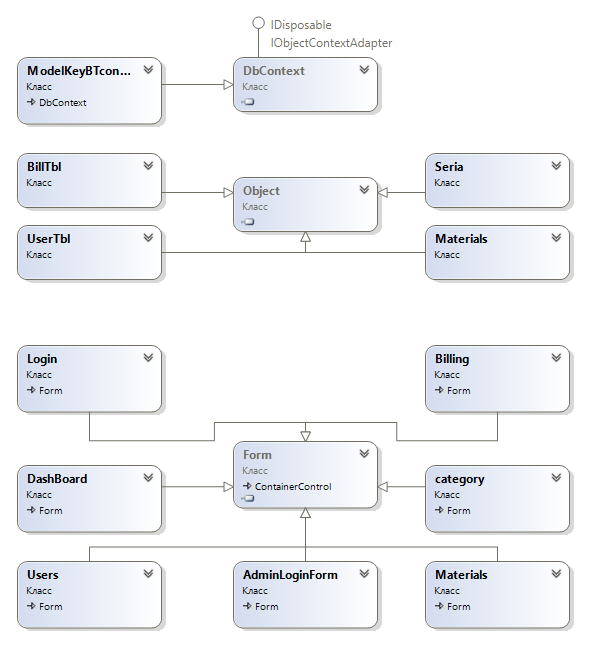


Рисунок 5 – Диаграмма классов[[18]](#footnote-18)

# 2.2.2 СПЕЦИФИКАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

Разрабатываемая информационная система имеет в своей основе модульный принцип разработки программы. Этот принцип был выбран в связи с тем, что, его применение позволяет разработчику абстрагироваться от излишней детализации, так же это является средством упрощения проектирования ИС и помогает распределить процесс разработки на несколько этапов. Помимо этого, данный принцип может помочь в дальнейшей поддержке программного кода и программы в целом, к примеру, при добавлении какой-либо новой функциональности или модуля разработчикам не потребуется изменять полностью или большую часть кода программы для работоспособности всей интегрированной части нового модуля. Очень хорошо демонстрирует работу данного принципа – принцип объектно-ориентированного программирования.

Далее представлены схемы вызовов программных модулей разрабатываемой информационной системы:

Авторизация администратор

Справочник материалы

Справочник пользователь

Статистика

Авторизация пользователь

Форма заполнения наряда

Справочник категории

Рисунок 6 – Схема вызовов программных модулей[[19]](#footnote-19)

В представленной выше схеме вызова программных модулей реализуется стандартная точка входа в программу и предоставляется стандартный интерфейс пользователя с определенными модулями. В разработанной информационной системе, есть разграничении прав доступа , в организации данный процесс необходимо реализовать так, чтобы сотрудник был ответственным за свой наряд и не мог выписать на другого пользователя ТМЦ. Программа призвана показать возможности оптимизации бизнес-процессов для контроля за выдачей ТМЦ. Идеальным вариантом было бы внедрение программного комплекса завязанного с 1С Предприятием, для оперативного выписывания наряда и списания материала без участия руководителя. Далее представлены подмодули основных программных модулей стандартного рабочего пространства информационной системы.

Справочник материалы

Список ТМЦ

Вывод информации о ТМЦ

Рисунок 7 – Подмодуль справочник материалы[[20]](#footnote-20)

Модуль «Справочник материалы» реализует функционал добавления, редактирования и удаления ТМЦ, а также поиск в базе данных.

Справочник категория

Список категорий ТМЦ

Вывод информации о категории ТМЦ

Рисунок 8 – Подмодуль справочник категории ТМЦ[[21]](#footnote-21)

Модуль «Справочник категории ТМЦ» реализует функционал добавления, редактирования и удаления районов, а также поиск в базе данных.

Статистика программы

Статистика по операциям

Вывод информации по статистике

Рисунок 9 – Подмодуль статистика программы[[22]](#footnote-22)

Модуль «статистика программы» реализует функционал вывода информации из таблиц базы данных, и статистических данных анализа в системе.

Справочник пользователи

Список пользователей

Вывод информации о пользователях

Рисунок 10 – Подмодуль справочник пользователи[[23]](#footnote-23)

Модуль «Справочник пользователи» реализует функционал добавления, редактирования и удаления сотрудников, а также поиск в базе данных.

Далее представлена блок-схема программных модулей диспетчера

Авторизация администратор

Справочник материалы

Справочник пользователь

Статистика программы

Авторизация пользователь

Форма заполнения наряда

Справочник категории

Рисунок 11 – Блок-схема программных модулей диспетчера[[24]](#footnote-24)

Как видно из блок-схемы, система разделена на два модуля. Администратор занимается административными фикциями по добавлению, редактированию, удалению записей в базе данных. Модуль пользователя предназначен только для заполнения наряда и вывода наряда на печать. Пользователь не может повлиять на справочники в системе т.к. не имеет правд доступа, что обезопасит в свою очередь всю систему от несанкционированного доступа и нанесению ущерба организации.

# 2.3 ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПРИЛОЖЕНИЯ

Пользовательский интерфейс информационной системы разрабатывается с применением Windows Forms. Windows Forms  — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде.

При запуске разработанной информационной системы, открывается окно авторизации в системе, в зависимости от того авторизовался пользователь или администратор, открывается соответствуете правам доступа рабочее окно информационной системы или главная форма со своим внутренним функционалом.

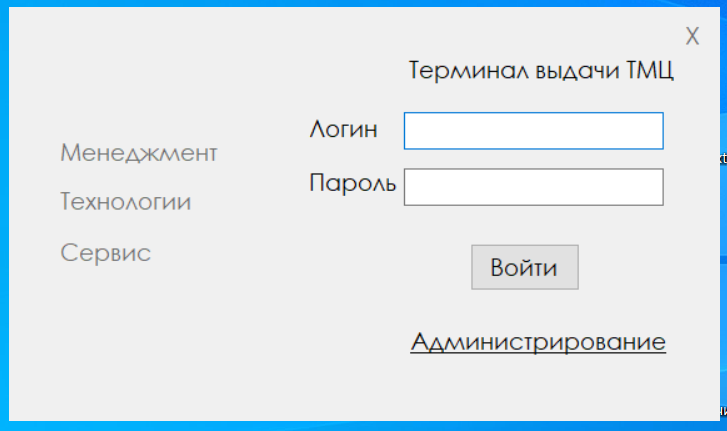


Рисунок 12 – Форма авторизации пользователя[[25]](#footnote-25)

Рассмотрим рабочее пространство пользователя. Главная форма представляет собой пространство журнала материалов ТМЦ на которой имеются все необходимые для работы пользователя элементы управления. В верхней левой части формы находятся логотип который отображает пользователю имя учетной записи под которой авторизован пользователь.

Далее информационное пространство приложения разделено на три части. Первая часть располагается в левой части экрана, представляет собой левую таблицу с заполненными данными из базы данных таблицы Материалы.

Вторая часть располагается в правой части экрана, представляет собой правую таблицу «Наряд на получение ТМЦ», с данными которые выписывает пользователь для получения на складе.

Третья часть располагается в нижней части экрана, представляет собой группу элементов управления, в ее состав входят:

* «Поиск» – строка поиска по таблице Материалы. Пользователь может производить поиск по ID, Наименованию, Производителю, Категории. При вводе текста в поле «Поиск», данные фильтруются и выводятся в таблице Материалы. Тем самым упрощая поиск в системе необходимых ТМЦ.
* Текстовое поле «Пользователь» – поле в котором указывается пользователь получатель ТМЦ. Для пользователя не администратора или не руководителя данное поле недоступно для редактирования, во избежание подмены получателя ТМЦ. Тем самым безопасность и корректность работы в системе.
* Текстовое поле «Наименование» – поле в котором указывается наименование ТМЦ выбранное пользователем в таблице материалы.
* Текстовое поле «Количество» – поле в котором указывается количество выбранного ТМЦ.
* Текстовое поле «Цена» – поле в котором указывается цена за единицу выбранного ТМЦ.
* Кнопка «Выписать» – элемент управления отвечает за перенос выбранного ТМЦ в таблицу Выписать. Сотрудник наполняет таблицу «Наряд на получение ТМЦ» необходимыми ТМЦ, после чего проводит наряд, и отправляет в печать документ для получения ТМЦ на складе.
* Кнопка «Очистить» – данная кнопка очищает поля Поиск, Наименование, Количество, Цена.
* Кнопка «Удалить» – данная кнопка удаляет из таблицы «Наряд на получение ТМЦ» позиции в которых не нуждается пользователь, либо по ошибке добавил в наряд.
* Кнопка «Печать» – данная кнопка формирует документ для печати наряда, который пользователь распечатывает и получает ТМЦ.
* Элемент «Итоговая стоимость» – информация по общей стоимости выписанных материалов в наряде.
* Элемент «Вернуться» – данная кнопка возвращает пользователя к окну авторизации.

Далее на рисунке представлено информационное пространство пользователя «Терминал выдачи ТМЦ».

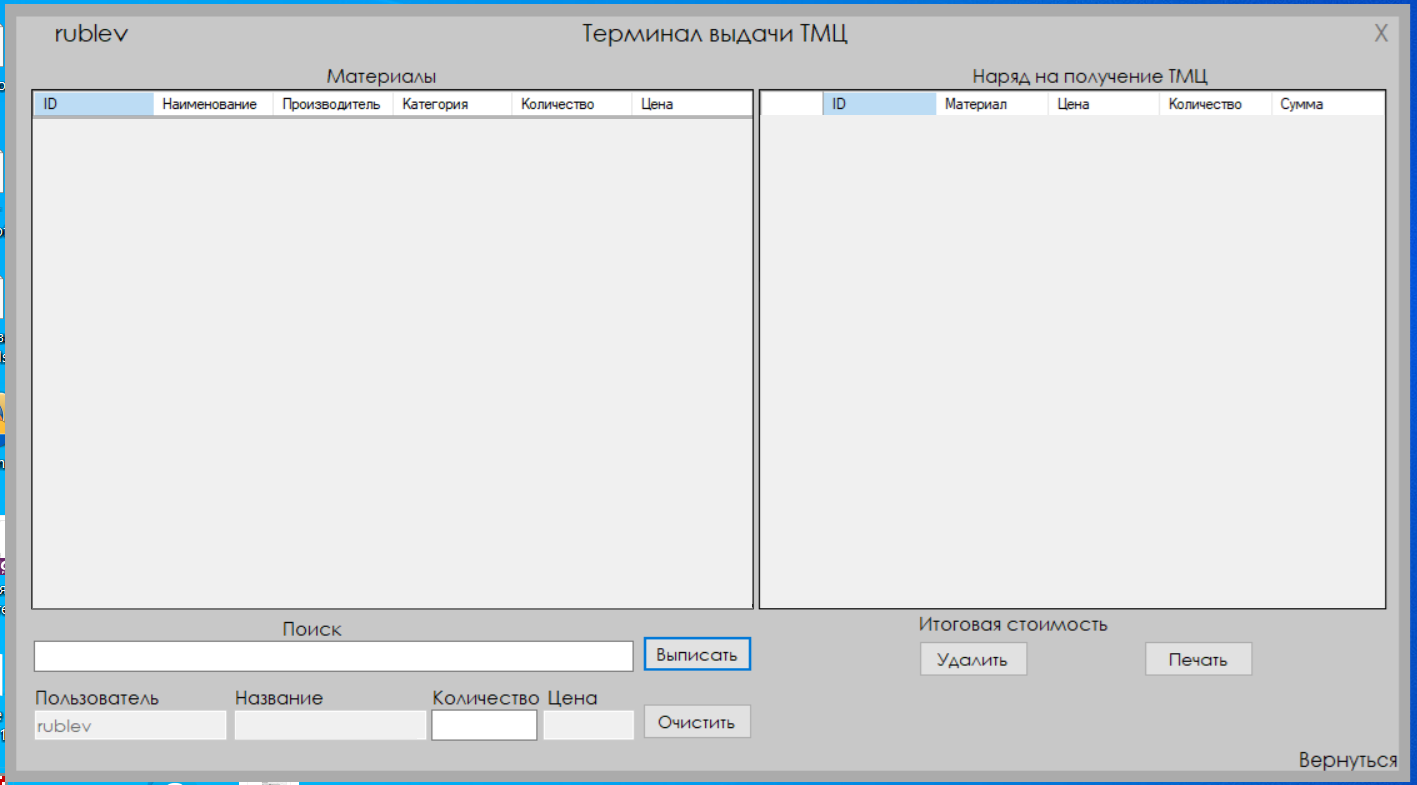


Рисунок 13 - Информационное пространство пользователя[[26]](#footnote-26)

Также рассмотрим пользовательский интерфейс администратора приложения (руководителя отдела). При запуске информационной системы, либо после возвращения из пользовательского режима, открывается окно авторизации в системе, авторизовавшись под учетной записью администратора, открывается соответствующее правам доступа рабочее окно информационной системы или главная форма со своим внутренним функционалом администратора.

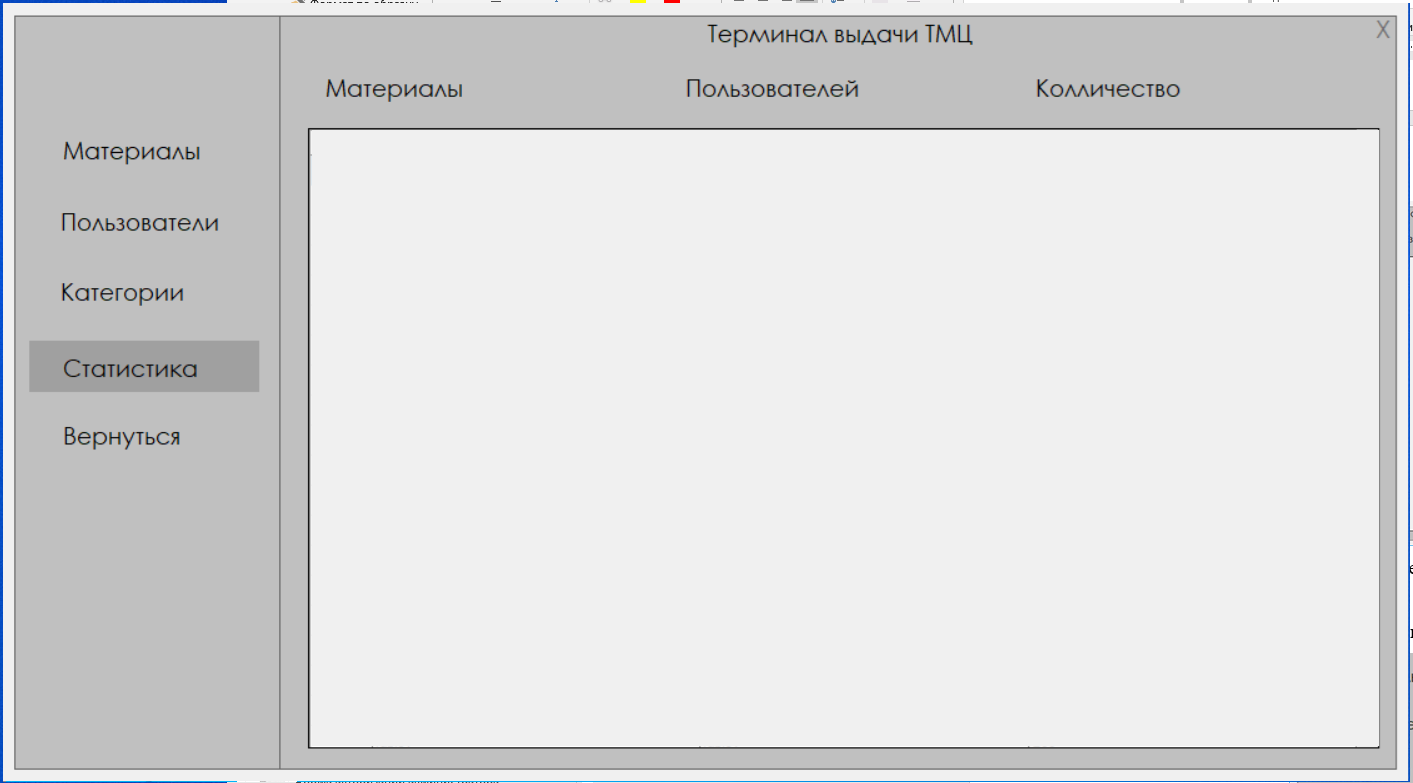


Рисунок 14 – Информационное пространство выдача ключей[[27]](#footnote-27)

Рассмотрим подробнее справочники информационной системы.

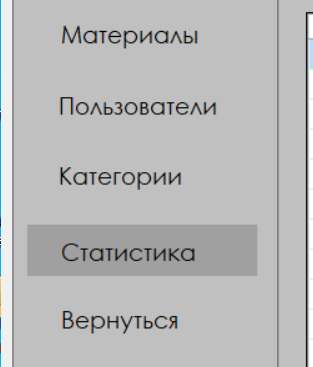


Рисунок 15 – Меню справочников главной формы[[28]](#footnote-28)

Справочники «Материалы», «Пользователи», «Категории» представляет собой формы с данными из базы данных, и выводится информацию о ТМЦ, пользователях, категории ТМЦ. Администратор может как добавлять записи, так и редактировать выбранные записи по средствам элементов управления «Добавить», «Изменить», «Удалить». Также на формах присутствует строка поиска, что значительно упрощает поиск записей для редактирования и получения информации о записи из базы данных. Текстовые поля в формах автоматически заполняются при выборе одной записи. Редактирование записи производится путем редактирования информации в текстовых полях форм. Далее на рисунках 15, 16, 17 представлены формы справочников:

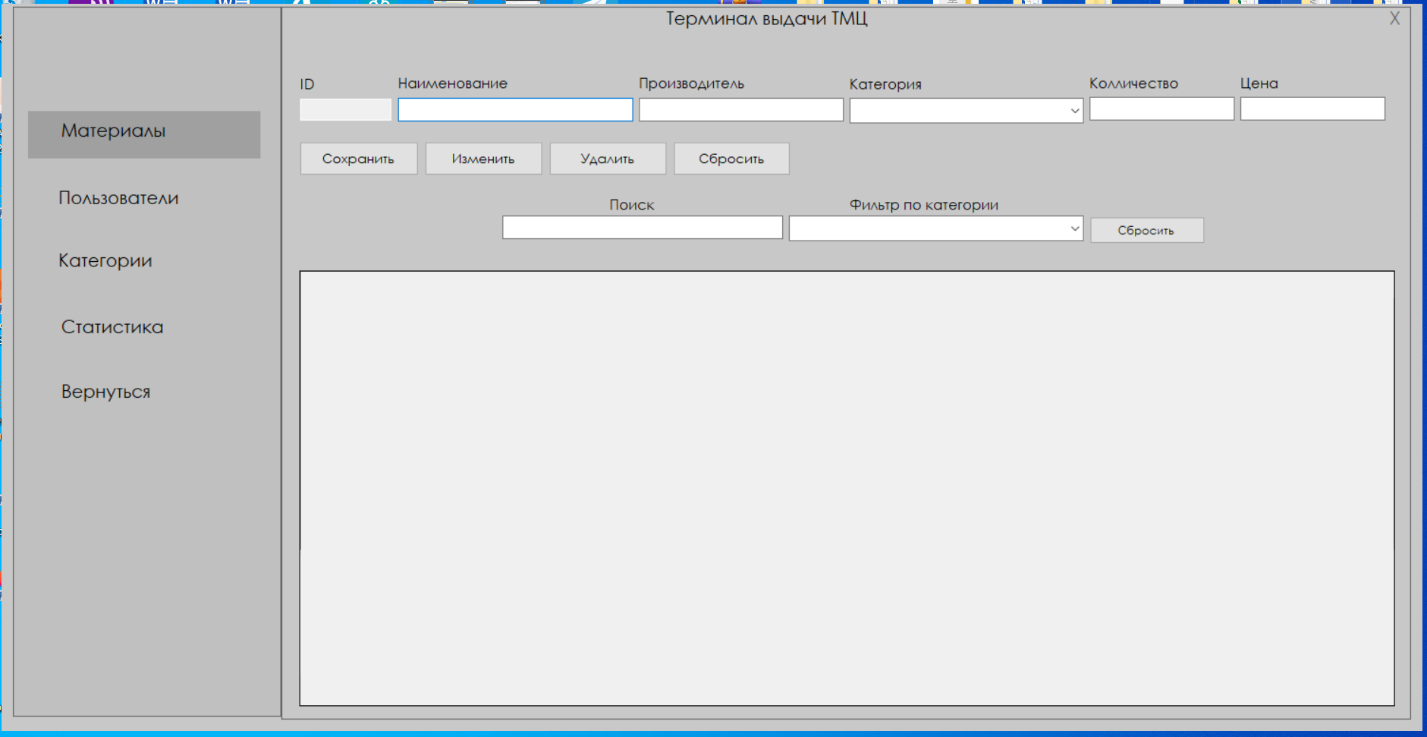


Рисунок 15 – Форма справочника материалы[[29]](#footnote-29)

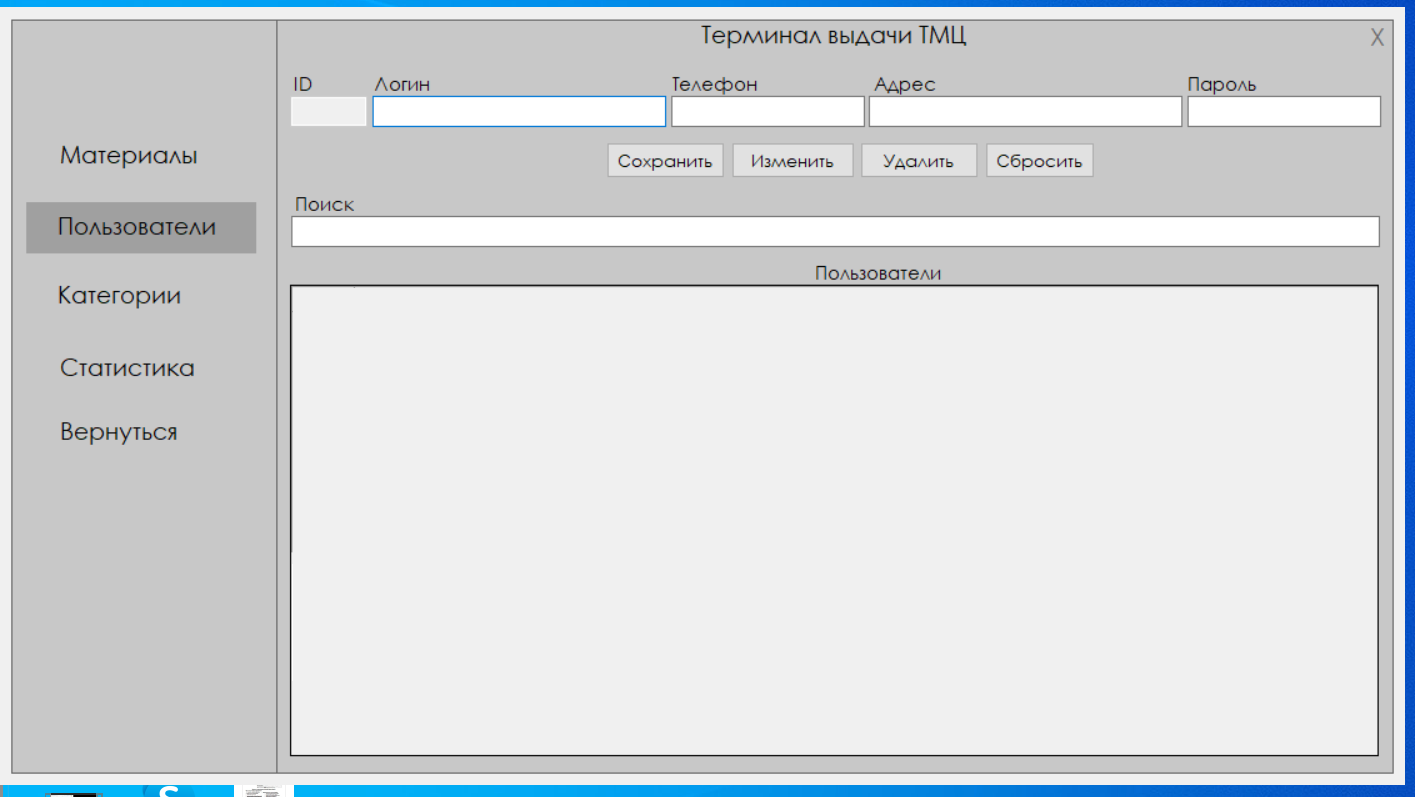


Рисунок 16 – Форма справочника пользователи[[30]](#footnote-30)

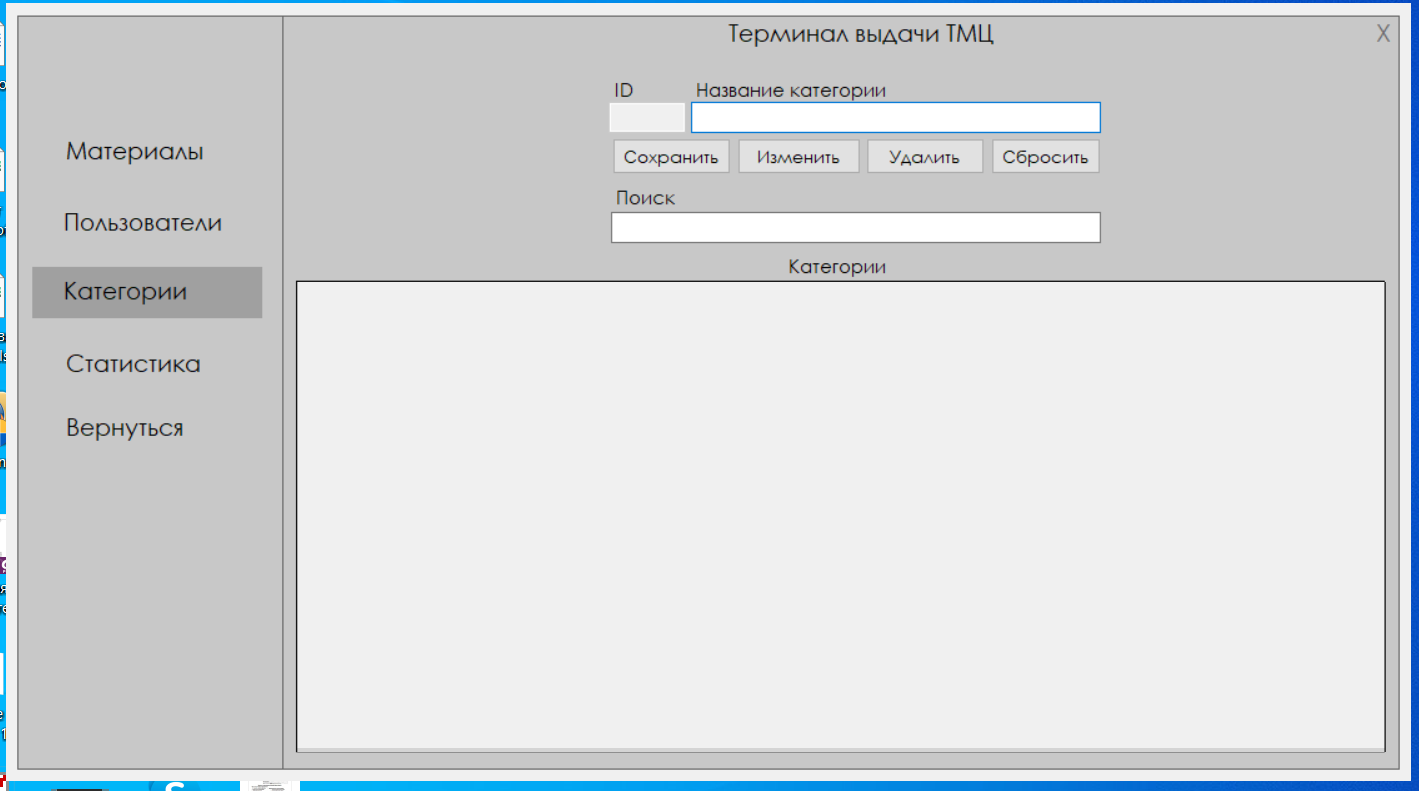


Рисунок 17 – Форма справочника категории[[31]](#footnote-31)

# 2.4 ТЕСТИРОВАНИЕ СИСТЕМЫ И ОЦЕНКА КАЧЕСТВА

Следующим этапом в разработке является тестирование и оценка качества, которая включает в себя проверку правильности и бесперебойности работы функций и методов информационной системы.

Запустив программное обеспечение Пользователя встречает форма авторизации в системе, с запросом ввода логина и пароля, рисунок 18.

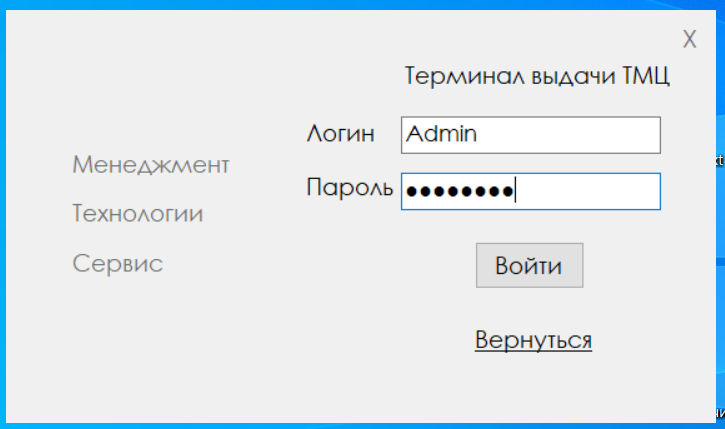


Рисунок 18 – Форма авторизации администратора[[32]](#footnote-32)

После успешной авторизации в системе, администратор попадает на главную форму «Статистика». Из данной формы нам доступны справочники «Материалы», «Пользователи» и «Категории». Наполним все справочники системы для корректной работы пользователей с «Терминал выдачи ТМЦ». Первым заполним справочник пользователи, предоставив права доступа для заполнения наряда на ТМЦ.

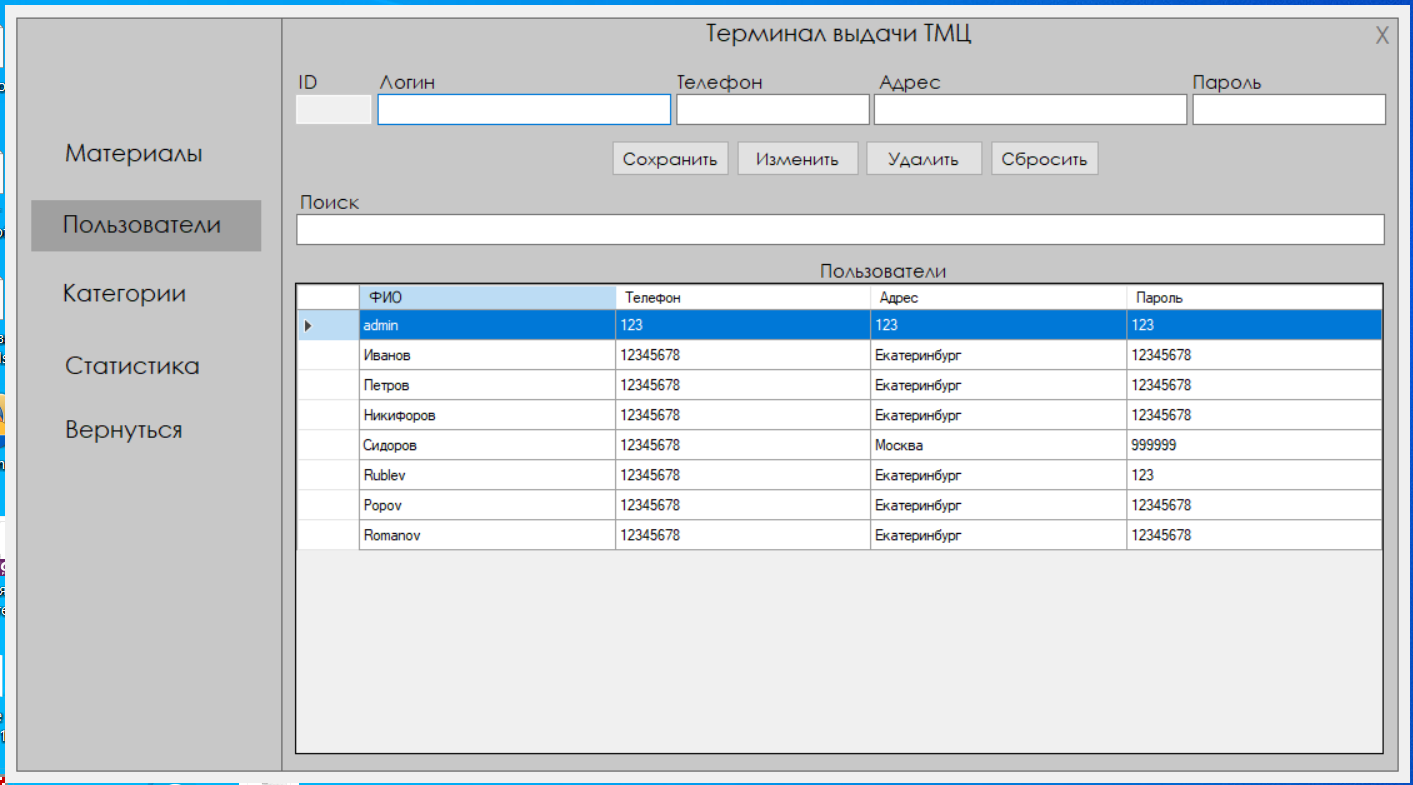


Рисунок 19 – Форма справочника пользователи [[33]](#footnote-33)

Далее заполним справочник категории, представлен на рисунке

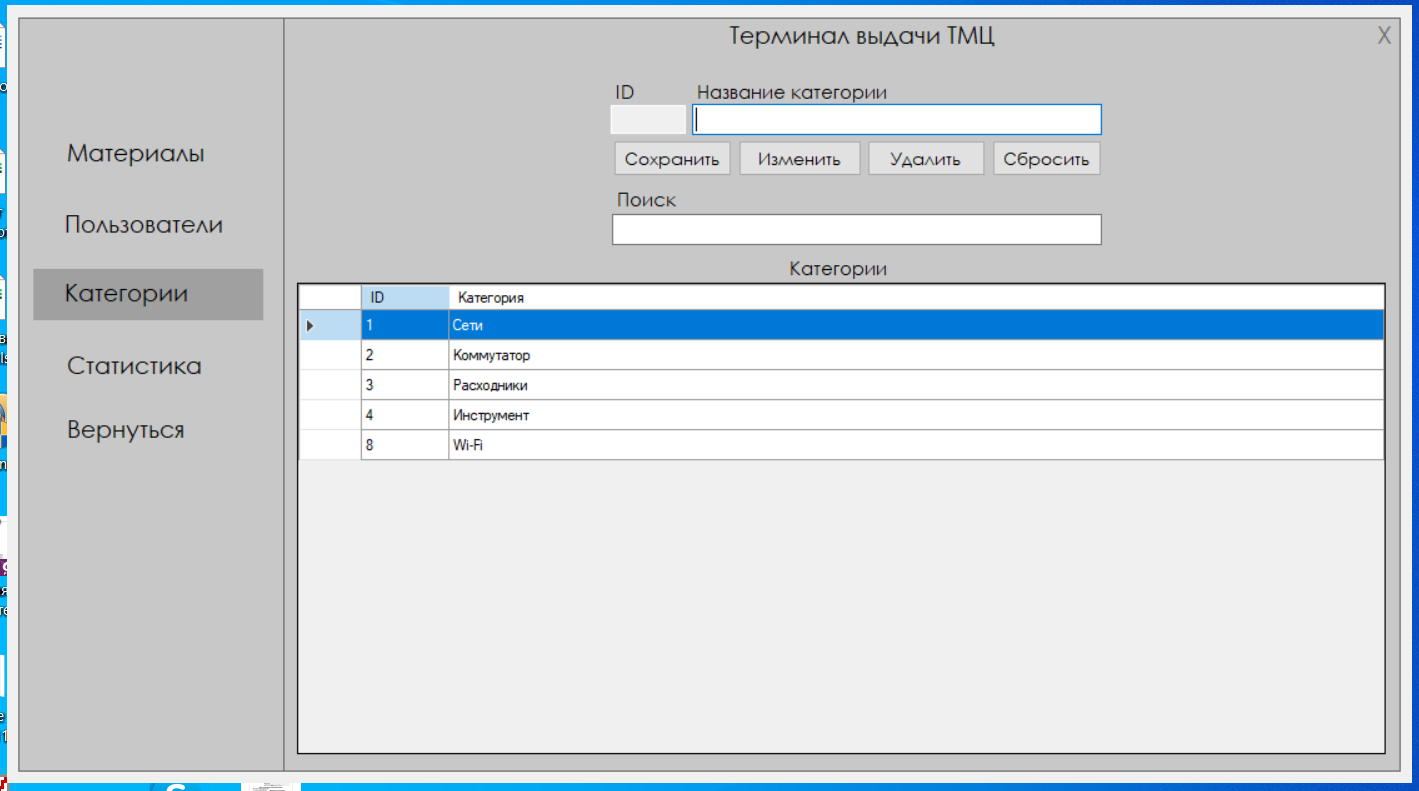


Рисунок 20 – Форма справочника категории [[34]](#footnote-34)

Заполним справочник Материалы, представлен на рисунке ниже

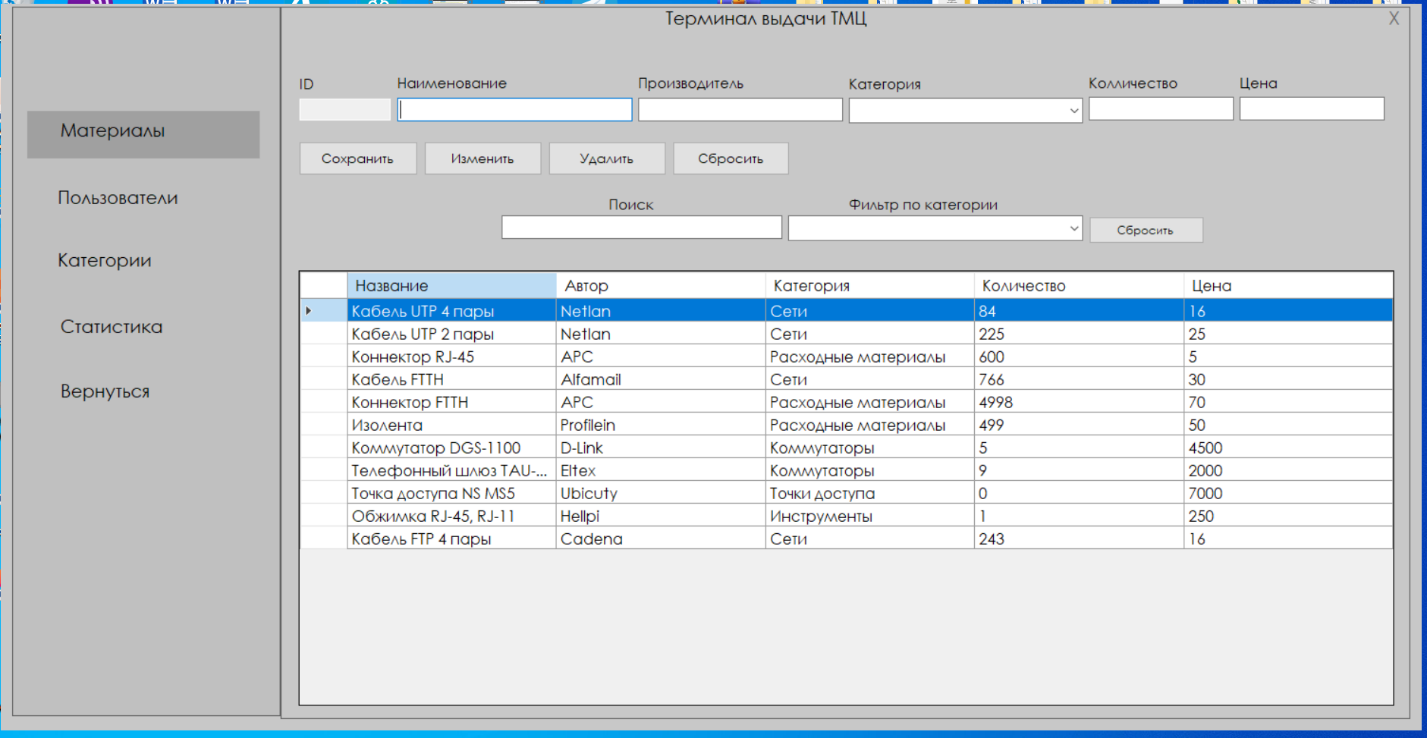


Рисунок 21 – Форма справочника материалы [[35]](#footnote-35)

Как видно на рисунках 19, 20, 21 все формы справочников имеют элементы управления «Добавить», «Изменить», «Удалить», «Сбросить» и строку поиск для удобного поиска и отображения информации. Основная информация для заполнения нарядов на получение ТМЦ внесена в базу данных программного обеспечения «Терминал выдачи ТМЦ».

В режиме администратора проведено тестирование системы, все функции выполняются, данные в систему учета внесены. Далее продолжим тестирование от учетной записи пользователя. Вернемся на форму авторизации и авторизуемся используя учетные данные пользователя Логин – Rublev, пароль – 123.

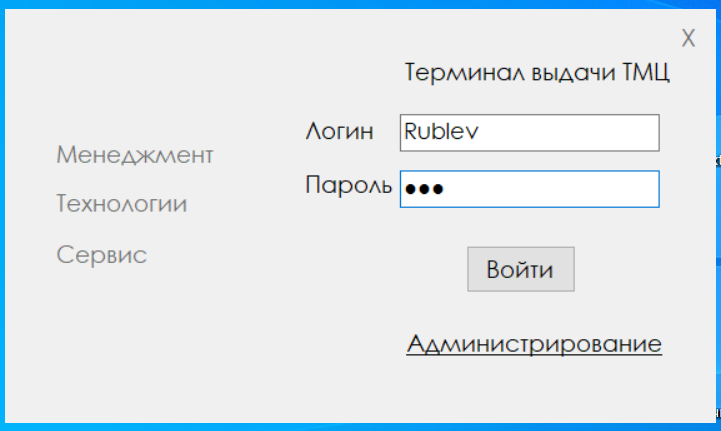


Рисунок 22 – форма авторизации пользователя [[36]](#footnote-36)

После успешной авторизации в системе откроется главное рабочее окно пользователя «Терминал выдачи ТМЦ», с заполненной таблицей «Материалы» из базы данных.

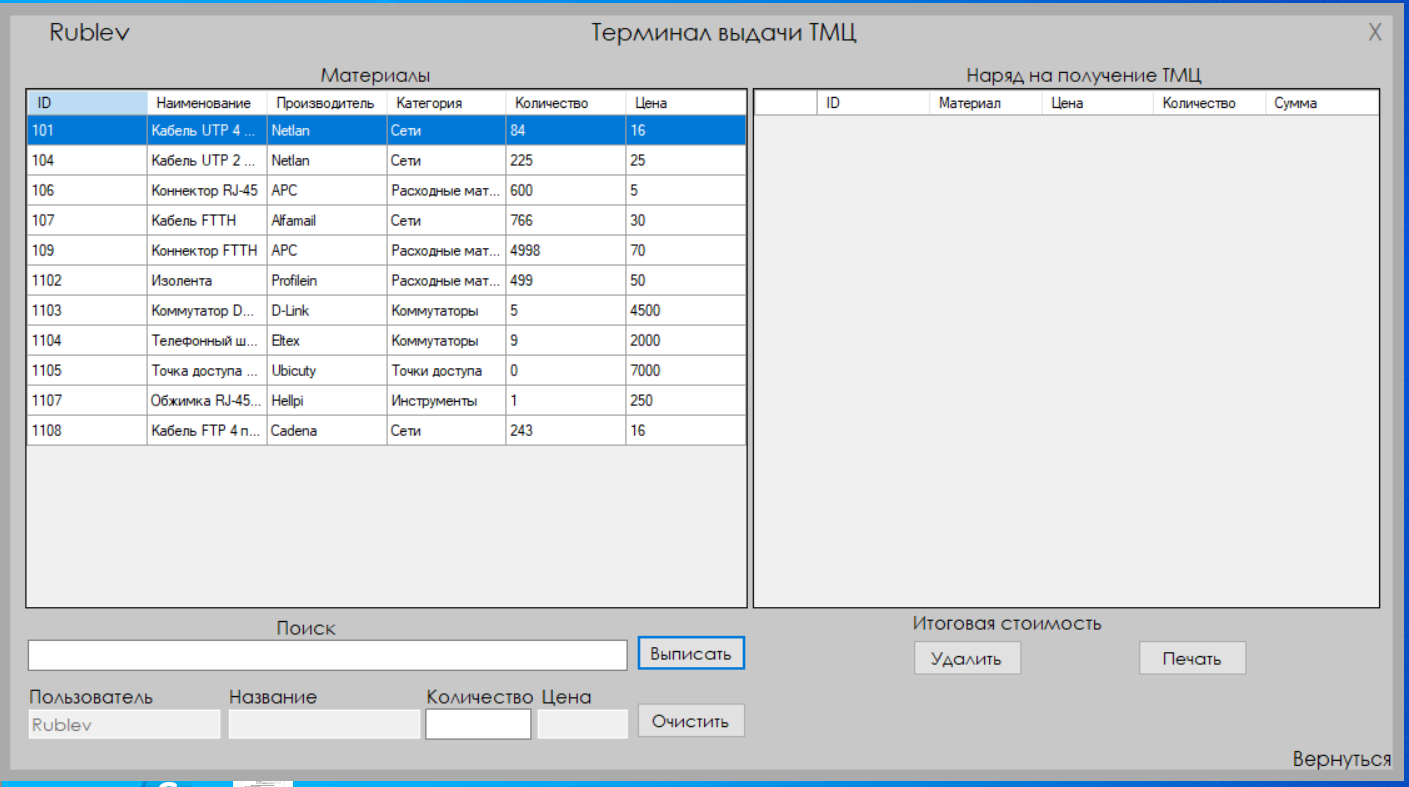


Рисунок 23 – Главная форма пользовательского интерфейса [[37]](#footnote-37)

В верхней левой части рабочего окна выведен логин пользователя который авторизован в системе, также и в нижней части окна в поле Пользователь указана учетная запись пользователя на которого будет выписан материал ТМЦ. В целях предотвращения ошибок и преднамеренного изменения получателя ТМЦ, поле пользователь недоступно для редактирования.

Заполним наряд для получения ТМЦ из таблицы «Материалы», выбрав позиции или воспользовавшись поисковой строкой, Указав количество требуемого ТМЦ, нажмем кнопку «Выписать», данные в системе изменятся и будут переданы в таблицу справа «Наряд на получение ТМЦ».

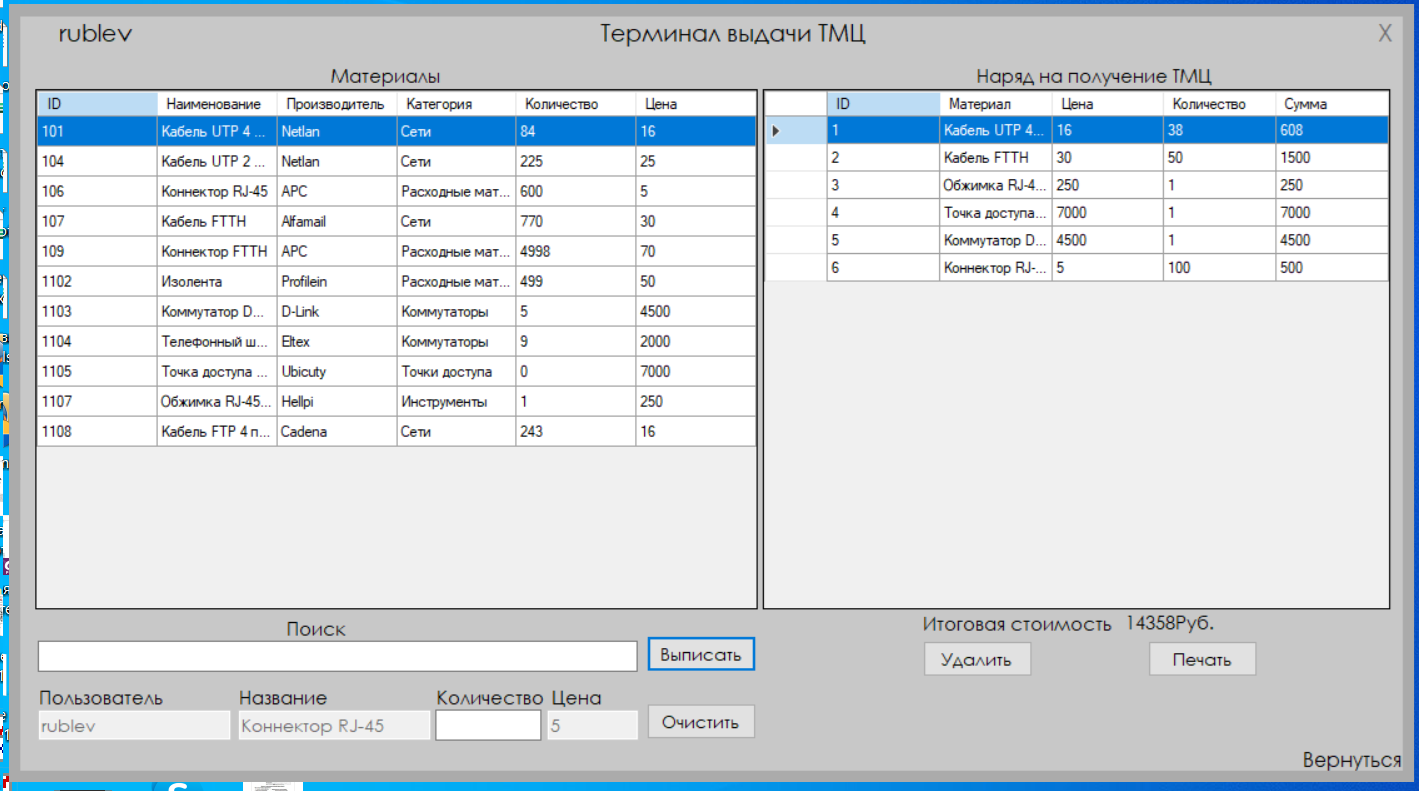


Рисунок 24 –Заполнение наряда пользователем [[38]](#footnote-38)

Заполнив наряд пользователь может удалить ошибочно выбранную или лишнюю позицию в таблице наряд на получение ТМЦ. Итоговая стоимость материалов ТМЦ подведена в нижней части таблицы. Далее пользователь отправляет наряд на печать, нажав на кнопку «Печать». Откроется диалоговое окно предварительного просмотра наряда, представлен на рисунке 25.

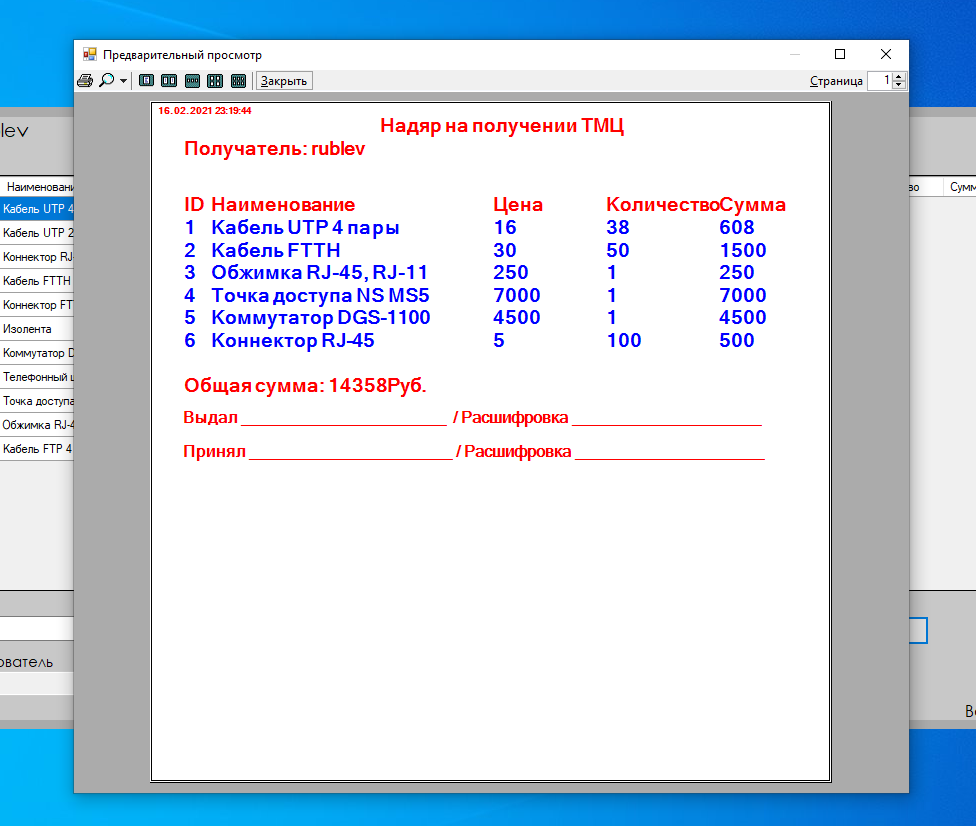


Рисунок 25 – Окно предварительного просмотра заполненного наряда[[39]](#footnote-39)

Далее распечатываем наряд. В наряде указана текущая дата документа, получатель, перечень выписанного ТМЦ с количеством и ценой, ответственный за выдачу ТМЦ и получивший ТМЦ пользователь.

После проведения отчета, информация поступает в форму и таблицу «Статистика» доступной только для администратора. В статистике администратор видит общее число пользователей, общее количество материалов на складе, и количество выданного ТМЦ. Таким образом участие руководителя отдела не является обязательной частью для выписывания наряда сотрудникам в отделе, что освобождает время на планирование работ, исключает человеческий фактор неправильно выписанного ТМЦ или неверно указанного получателя ТМЦ, повышена ответственность сотрудников. Пользовательский интерфейс максимально прост по сравнению с 1С Предприятием, имеет весь необходимый функционал для простой и удобной работы только в той зоне ответственности, которая предписана пользователю.

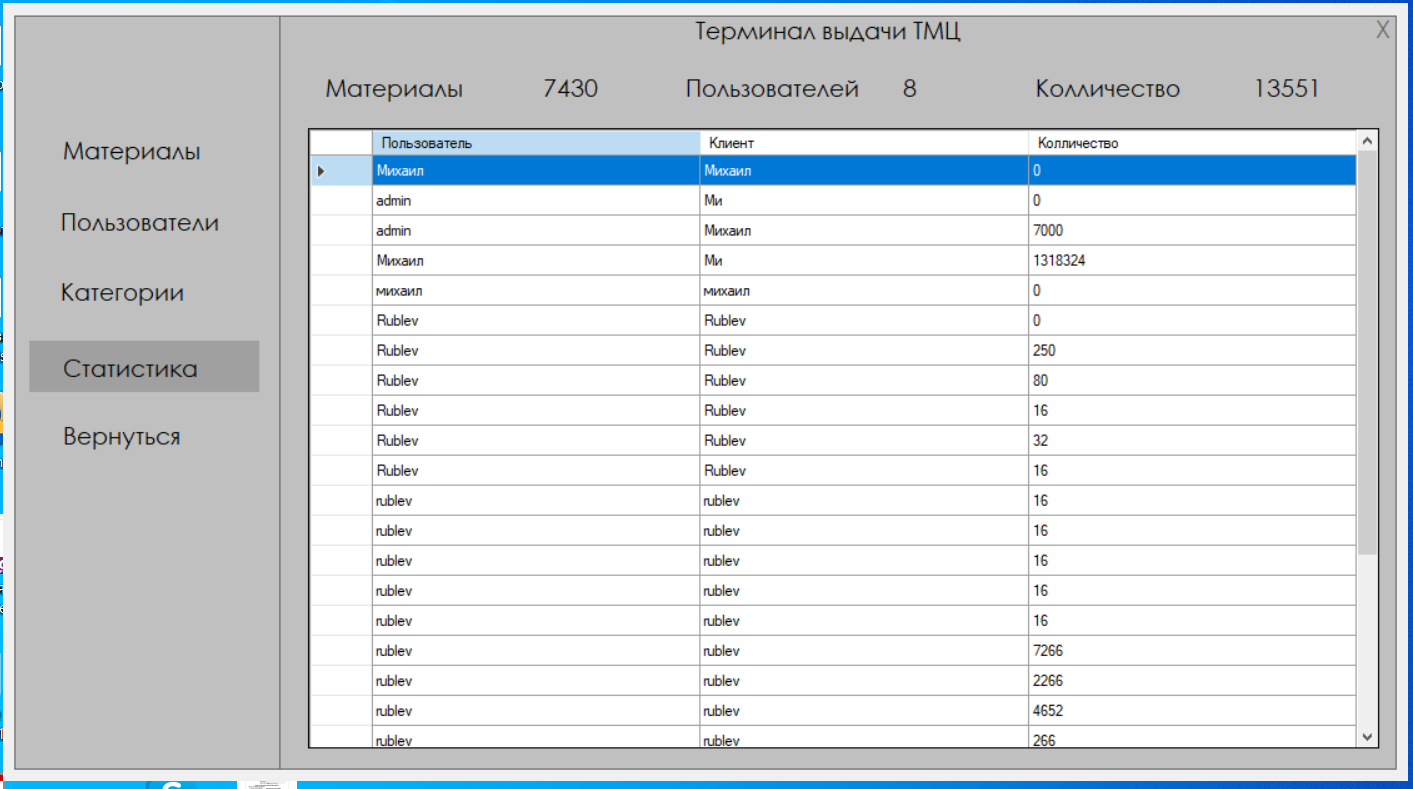


Рисунок 26 – Форма Статистика[[40]](#footnote-40)

Таким образом в данном параграфе протестирована и продемонстрирована работа информационного пространства администратора и пользователя, на примере работы всех модулей программного обеспечения.

В ходе тестирования разработанной программы были выявлены незначительные недостатки в некоторых модулях, которые были полностью исправлены. После чего было проведено дополнительное тестирование программы, которое более не выявило каких-либо ошибок.

# ЗАКЛЮЧЕНИЕ

В курсовой работе была спроектирована и разработана информационная система для оптимизации работы отдела «Терминал выдачи ТМЦ», которая позволяет повысить эффективность работы руководителя и сотрудников, уменьшить издержки времени, повысить контроль выданного ТМЦ, исключить ошибки допускаемые по вине человеческого фактора. Программа имеет интуитивно понятный интерфейс и максимально проста в использовании.

В рамках выполнения курсовой работы была проделана работа по анализу бизнес-процесса отдела организации, выявлены сильные и слабые стороны бизнес-процесса. Так же была проведена оценка автоматизации и организации прикладных процессов, произведен анализ недостатков и сформировано предложение по автоматизации этих бизнес-процессов, и устранению выявленных недостатков.

В ходе выполнения работы была полностью спроектирована база данных для информационной системы, на основе полученной информации в ходе исследования. Были проанализированы, обоснованы и выбраны технологии, которые использованы для разработки информационной системы «Терминал выдачи ТМЦ», спроектирован и разработан современный, пользовательский интерфейс, который получил положительную оценку от сотрудников организации в ходе тестового использования ИС, так как система максимально имеет дружественный интерфейс.

Так же помимо всего вышеописанного было произведено тестирование ИС, которое показало высокое качество выполнения программы, произведены тесты оценки качества системы на отказоустойчивость.

Разработанная информационная система, имеет в своей основе модульный подход, разрабатывалась с учётом применения ООП, таким образом является доступной дальнейшая поддержка и разработка дополнительных модулей программы. При предоставлении доступа к 1С предприятие, процесс выдачи и списания материала с сотрудников будет максимально упрощён и автоматизирован.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Казанский, А. А. Объектно-ориентированный анализ и программирование на Visual Basic 2013 [Электронный ресурс] : Учебник для вузов / Казанский А. А. - Москва : Юрайт, 2020. - 290 с. <https://urait.ru/bcode/451412>
2. Entity Framework Core — это современный модуль сопоставления "объект — база данных" для .NET. [Электронный ресурс] :

https://docs.microsoft.com/ru-ru/ef/ef6/modeling/

1. Многоформатная библиотека обработки изображений 1D / 2D штрих-кода с открытым исходным кодом, реализованная на Java, с портами на другие языки. [Электронный ресурс] :

https://github.com/zxing/zxing/wiki/Getting-Started-Developing

1. Лаврищева, Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства [Электронный ресурс] : Учебник для вузов / Лаврищева Е. М. - 2-е изд. - Москва : Юрайт, 2020. - 280 с. <https://urait.ru/bcode/452156>
2. Тузовский, А. Ф. Объектно-ориентированное программирование [Электронный ресурс] : Учебное пособие для вузов / Тузовский А. Ф. - Москва : Юрайт, 2020. - 206 с. <https://urait.ru/bcode/451429>
3. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# [Электронный ресурс] : Учебное пособие : ВО - Бакалавриат / Московский энергетический институт. - 1. - Москва : Издательство «ФОРУМ», 2020. - 200 с. <http://new.znanium.com/go.php?id=1069921>
4. Гагарина, Л. Г. Технология разработки программного обеспечения [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям подготовки 09.04.01 и 09.03.03 «Информатика и вычислительная техника» / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. - Москва : ФОРУМ: ИНФРА-М, 2019. - 400 с. <https://new.znanium.com/catalog/product/1011120>
5. Гуриков, С. Р. Основы алгоритмизации и программирования в среде LAZARUS [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальности 09.03.01 «Информатика и вычислительная техника» / С. Р. Гуриков. - Москва : ФОРУМ: ИНФРА-М, 2019. - 336 с. <https://new.znanium.com/catalog/product/1010784>
6. Зыков, С. В. Программирование. Объектно-ориентированный подход [Электронный ресурс] : учебник и практикум для академического бакалавриата: для студентов вузов, обучающихся по инженерно-техническим направлениям и специальностям / С. В. Зыков. - Москва : Юрайт, 2019. - 155 с. <https://www.biblio-online.ru/bcode/434106>
7. Кузин, А. В. Основы программирования на языке Objective-C для iOS [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям подготовки 09.03.04 «Прикладная инженерия», 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии», 09.03.03 «Прикладная информатика» (квалификация (степень) «бакалавр») / А. В. Кузин, Е. В. Чумакова. - Москва : ИНФРА-М, 2019. - 118 с. <https://new.znanium.com/catalog/product/1003260>
8. Подбельский, В. В. Программирование. Базовый курс С# [Электронный ресурс] : учебник для бакалавриата и специалитета: для студентов вузов, обучающихся по инженерно-техническим направлениям / В. В. Подбельский. - Москва : Юрайт, 2019. - 369 с. <https://www.biblio-online.ru/bcode/439068>
9. Федоров, Д. Ю. Программирование на языке высокого уровня Python [Электронный ресурс] : Учебное пособие Для прикладного бакалавриата / Федоров Д. Ю. - 2-е изд. - Москва : Юрайт, 2019. - 161 с. <https://urait.ru/bcode/437489>
10. Гуриков, С. Р. Основы алгоритмизации и программирования на Python [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям подготовки 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии» (квалификация (степень «бакалавр») / С. Р. Гуриков. - Москва : ФОРУМ: ИНФРА-М, 2018. - 343 с. <https://new.znanium.com/catalog/product/924699>
11. Плещев, В. В. Основы программирования на языках C++ и C# с примерами и упражнениями [Текст] : учебное пособие / В. В. Плещев, Е. И. Шишков ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2018. - 286 с. <http://lib.usue.ru/resource/limit/ump/18/p490708.pdf> (40 экз.)
12. Кислицын, Е. В. Разработка приложений на языке Java [Текст] : учебное пособие / Е. В. Кислицын, Е. И. Шишков ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2017. - 86 с. <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf> (30 экз.)
13. Седжвик, Р. Алгоритмы на Java [Текст] : научное издание / Роберт Седжвик, Кевин Уэйн ; [пер. с англ. А. А. Моргунова; под ред. Ю. Н. Артеменко]. - 4-е изд. - Москва : Вильямс, 2017. - 843 с. (20 экз.)
14. Шилдт, Г. Java 8 [Текст] : руководство для начинающих / Герберт Шилдт ; [пер. с англ. и ред. А. Г. Гузикевича]. - 6-е изд. - Москва : Вильямс, 2017. - 712 с. (30 экз.)
15. Эйдлина, Г. М. Delphi: программирование в примерах и задачах. Практикум [Электронный ресурс] : учебное пособие / Г. М. Эйдлина, К. А. Милорадов. - Москва : РИОР: ИНФРА-М, 2017. - 119 с. <https://new.znanium.com/catalog/product/858775>
16. Моделирование и виртуальное прототипирование [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлению подготовки дипломированного спциалиста 27.05.01 «Спец. орг.-техн. системы», 230101 «Моделирование и исслед. операций в орг.-техн. системах» / И. И. Косенко [и др.]. - Москва : Альфа-М: ИНФРА-М, 2016. - 176 с. <https://new.znanium.com/catalog/product/555214>
17. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] : производственно-практическое издание / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес; [пер. с англ. А. Слинкина]. - Санкт-Петербург [и др.] : Питер, 2016. - 366 с. (3 экз.)
18. Абрамов, Е. С. Машинно-ориентированное программирование [Электронный ресурс] : учебное пособие / Е. С. Абрамов, И. Д. Сидоров ; М-во образования и науки Рос. Федерации, Юж. федер. ун-т, Инжен.-технол. акад. - Таганрог : Издательство Южного федерального университета, 2016. - 87 с. <https://new.znanium.com/catalog/product/991870>
19. Емельянов, Д. А. Основы технологии программирования в среде Delphi [Текст] : учебное пособие / Д. А. Емельянов ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2016. - 185 с. <http://lib.usue.ru/resource/limit/ump/16/p487066.pdf> (30 экз.)
20. Йенер, М. Java EE. Паттерны проектирования для профессионалов [Текст] : научно-популярная литература / Мурат Йенер, Алекс Фидом; [пер. с англ. И. Пальти]. - Санкт-Петербург [и др.] : Питер, 2016. - 239 с. (2 экз.)
21. Пушкарева, Т. П. Основы компьютерной обработки информации [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям подготовки: 22.03.01 «Материаловедение и технологии материалов», 29.03.04 «Технология художественной обработки материалов», 22.04.01 «Материаловедение и технология» / Т. П. Пушкарёва ; М-во образования и науки Рос. Федерации, Сибир. федер. ун-т. - Красноярск : Сибирский федеральный университет, 2016. - 180 с. <https://new.znanium.com/catalog/product/967586>
22. Хорстманн, Кей С. Java™ [Текст] : [в 2 томах]. Т. 1 : Основы. - Москва; Санкт-Петербург; Киев : Вильямс, 2016. - 862 с. (1 экз.)
23. Шилдт, Г. Java 8 [Текст] : руководство для начинающих / Герберт Шилдт ; [пер. с англ. и ред. А. Г. Гузикевича]. - 6-е изд. - Москва : Вильямс, 2016. - 712 с. (2 экз.)
24. Гонсалвес, Э. Изучаем Java EE 7 [Текст] : производственно-практическое издание / Энтони Гонсалвес ; [пер. с англ.: Е. Зазноба [и др.]. - Санкт-Петербург [и др.] : Питер, 2015. - 640 с. (2 экз.)
25. Клеменс, Б. Язык С в XXI веке [Электронный ресурс] : производственно-практическое издание / Бен Клеменс ; [пер. с англ. А. А. Слинкина]. - Москва : ДМК Пресс, 2015. - 376 с. <https://new.znanium.com/catalog/product/1027782>
26. Комлев, Н. Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей [Электронный ресурс] : Практическое пособие / Н. Ю. Комлев. - Москва : СОЛОН-ПРЕСС, 2015. - 298 с. <https://new.znanium.com/catalog/product/884394>
27. Павловская, Т. А. С/С++. Процедурное и объектно-ориентированное программирование [Текст] : учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов «Информатика и вычислительная техника» / Татьяна Павловская. - Санкт-Петербург [и др.] : Питер, 2015. - 495 с. (1 экз.)
28. Тюгашев, А. А. Языки программирования [Текст] : учебное пособие для студентов вузов, обучающихся по специальности 10.05.03 (090303) «Информационная безопасность автоматизированных систем» / А. А. Тюгашев. - Санкт-Петербург [и др.] : Питер, 2015. - 333 с. (2 экз.)
29. Хорстманн, Кей С. Java™ [Текст] : [в 2 томах]. Т. 1 : Основы. - Москва; Санкт-Петербург; Киев : Вильямс, 2015. - 862 с. (1 экз.)
30. Хорстманн, Кей С. Java™ [Текст] : [в 2 томах]. Т. 2 : Расширенные средства программирования. - Москва; Санкт-Петербург; Киев : Вильямс, 2015. - 1000 с. (2 экз.)
31. Шакин, В. Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio .Net [Электронный ресурс] : учебное пособие / В. Н. Шакин, Г. К. Сосновиков, А. В. Загвоздкина. - Москва : ФОРУМ: ИНФРА-М, 2015. - 400 с. <https://new.znanium.com/catalog/product/501448>
32. Информатика и программирование [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям: 080801.65 «Прикладная информатика», 080801.65.01 «Прикладная информатика в экономике», 080801.65.02 «Прикладная информатика в менеджменте», 080801.65.29 «Прикладная информатика в рекламе», 080801.65.28 «Прикладная информатика в международном бизнесе» / Р. Ю. Царев [и др.] ; М-во образования и науки Рос. Федерации, Сибир. федер. ун-т. - Красноярск : Сибирский федеральный университет, 2014. - 132 с. <https://new.znanium.com/catalog/product/506203>
33. Парфенов, Д. В. Язык Си: кратко и ясно [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям 010400 «Прикладная математика и информатика» и 010300 «Фундаментальная информатика и информационные технологии» / Д. В. Парфенов. - Москва : Альфа-М: ИНФРА-М, 2014. - 320 с. <https://new.znanium.com/catalog/product/459254>
34. Рихтер, Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке С# [Текст] : производственно-практическое издание / Джеффри Рихтер ; [пер. с англ. Е. Матвеева]. - 4-е изд. - Санкт-Петербург [и др.] : Питер, 2014. - 895 с. (1 экз.)
35. Эванс, Б. Java. Новое поколение разработки [Текст] : техники Java 7 и многоязычное программирование / Бенджамин Эванс, Мартин Вербург; [пер. с англ. О. Сивченко]. - Санкт-Петербург [и др.] : Питер, 2014. - 556 с. (3 экз.)
36. Аллан, А. Программирование для мобильных устройств на iOS [Текст] : профессиональная разработка приложений для iPhone, iPad и iPod Touch / А. Аллан ; [пер. с англ. Е. Матвеева]. - Санкт-Петербург [и др.] : Питер, 2013. - 410 с. (1 экз.)
37. Емельянов, Д. А. Основы технологии программирования в среде C++Builder [Текст] : учебное пособие / Д. А. Емельянов, Л. И. Миронова ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2013. - 126 с. <http://lib.usue.ru/resource/limit/ump/14/p480356.pdf> (30 экз.)
38. Орлов, С. А. Теория и практика языков программирования [Текст] : учебник по направлению «Информатика и вычислительная техника» / С. А. Орлов. - Санкт-Петербург [и др.] : Питер, 2013. - 688 с. (1 экз.)
39. Рихтер, Дж. CLR via C# . Программирование на платформе Microsoft.NET Framework 4.5 на языке C# [Текст] : производственно-практическое издание / Джеффри Рихтер; [пер. с англ. Е. Матвеева ] . - 4-е изд. - Санкт-Петербург [и др.] : Питер, 2013. - 895 с. (1 экз.)
40. Язык программирования C# [Текст] : научное издание / А. Хейлсберг [и др.] ; [пер. с англ. Р. Тетерин]. - 4-е изд. - Санкт-Петербург [и др.] : Питер, 2012. - 773 с. (2 экз.)
41. Емельянов, Д. А. Объектно-ориентированное программирование в среде Delphi [Текст] : учебное пособие / Д. А. Емельянов ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2012. - 123 с. <http://lib.usue.ru/resource/limit/ump/12/p475921.pdf> (25 экз.)
42. Хорев, П. Б. Объектно-ориентированное программирование [Текст] : учебное пособие для студентов, обучающихся по направлению «Информатика и вычислительная техника» / П. Б. Хорев. - 4-е изд., стер. - Москва : Академия, 2012. - 447 с. (3 экз.)
43. Дейтел, Х. М. Как программировать на C++ [Текст] : [учебник] / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. под ред. В. В. Тимофеева. - 5-е изд. - Санкт-Петербург : Бином, 2011. - 1454 с. (3 экз.)
44. Троелсен, Э. Язык программирования C# 2010 и платформа .NET 4 [Текст] : переводное издание / Эндрю Троелсен ; [пер. с англ.: Я. П. Волкова, А. А. Моргунов, Н. А. Мухин ; под ред. Ю. Н. Артеменко]. - 5-е изд. - Москва : Вильямс, 2011. - 1392 с. (2 экз.)
45. Истомин, Е. П. Высокоуровневые методы информатики и программирования [Текст] : учебник для студентов вузов, обучающихся по специальности 351400 «Прикладная информатика (по областям)» и др. междисциплинарным специальностям / Е. П. Истомин, В. В. Новиков, М. В. Новикова ; Рос. гос. гидрометеорол. ун-т. - Санкт-Петербург : Андреевский издательский дом, 2010. - 228 с. (5 экз.)
46. Павловская, Т. А. С/С++. Программирование на языке высокого уровня [Текст] : учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов «Информатика и вычислительная техника» / Т. А. Павловская. - Санкт-Петербург [и др.] : Питер, 2010. - 460 с. (10 экз.)

# ПРИЛОЖЕНИЕ

ПРОГРАММНЫЙ КОД

### Форма пользователя Терминал выдачи ТМЦ

using BoolTest.DataBase;

using System;

using System.ComponentModel;

using System.Data;

using System.Data.Entity;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Windows.Forms;

namespace BoolTest

{

public partial class Billing : Form

{

public Billing()

{

InitializeComponent();

populate();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=BookShopDb;Integrated Security=True");

// Инициализируем объект нашего контекста

ModelKeyBTcontext db;

private void populate()

{

using (db)

{

// Создаём объект нашего контекста

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.BookTbls.Load();

// Привязываем данные к dataGridView

BookDGV.DataSource = db.BookTbls.Local.ToBindingList();

// Заполнение шапки таблицы

BookDGV.RowHeadersVisible = false;

BookDGV.Columns[0].HeaderText = "ID";

BookDGV.Columns[1].HeaderText = "Наименование";

BookDGV.Columns[2].HeaderText = "Производитель";

BookDGV.Columns[3].HeaderText = "Категория";

BookDGV.Columns[4].HeaderText = "Количество";

BookDGV.Columns[5].HeaderText = "Цена";

}

}

private void UpdateBook()

{

int newQty = stock - Convert.ToInt32(QtyTb.Text);

try

{

if (BookId > 0)

{

string id = BookDGV.SelectedRows[0].Cells[0].Value.ToString();

Con.Open();

SqlCommand query = new SqlCommand("UPDATE BookTbl SET BQty=@BQty WHERE Bid = @id", Con);

query.Parameters.AddWithValue("@BQty", newQty);

query.Parameters.AddWithValue("id", id);

query.ExecuteNonQuery();

Con.Close();

populate();

//Reset();

}

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

int n = 0;

int Grdtotal = 0;

int total = 0;

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

{

if (QtyTb.Text == "" || Convert.ToInt32(QtyTb.Text) > stock)

{

MessageBox.Show("Превышено колличество по наличию");

}

else

{

total = Convert.ToInt32(QtyTb.Text) \* Convert.ToInt32(PriceTb.Text);

DataGridViewRow newRow = new DataGridViewRow();

newRow.CreateCells(BillDgv);

newRow.Cells[0].Value = n + 1;

newRow.Cells[1].Value = BTitle.Text;

newRow.Cells[2].Value = PriceTb.Text;

newRow.Cells[3].Value = QtyTb.Text;

newRow.Cells[4].Value = total;

BillDgv.Rows.Add(newRow);

n++;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UpdateBook();

Reset();

Grdtotal = Grdtotal + total;

TotalLbl.Text = Grdtotal + "Руб.";

}

}

private void label5\_Click(object sender, EventArgs e)

{

Application.Exit();

}

public int BookId;

public int stock = 0;

// Действие при выборе ячейки

private void BookDGV\_CellClick(object sender, DataGridViewCellEventArgs e)

{

BookId = Convert.ToInt32(BookDGV.SelectedRows[0].Cells[0].Value);

BTitle.Text = BookDGV.SelectedRows[0].Cells[1].Value.ToString();

PriceTb.Text = BookDGV.SelectedRows[0].Cells[5].Value.ToString();

stock = Convert.ToInt32(BookDGV.SelectedRows[0].Cells[4].Value);

}

// очистка полей

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

{

Reset();

}

// метод очистки полей

private void Reset()

{

txtSearch.Text = "";

BTitle.Text = "";

QtyTb.Text = "";

PriceTb.Text = "";

}

// Печать наряда

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

{

if (ClientNameTb.Text == "" || BTitle.Text == "")

{

MessageBox.Show("Не заполнено поле Пользователь");

}

else

{

try

{

Con.Open();

string query = "INSERT INTO BillTbl values (N'" + UserNameLbl.Text + "', N'" + ClientNameTb.Text + "', " + Grdtotal + ")";

SqlCommand cmd = new SqlCommand(query, Con);

cmd.ExecuteNonQuery();

Con.Close();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

// Запуск диалогового окна печати документа

printDocument1.DefaultPageSettings.PaperSize = new System.Drawing.Printing.PaperSize("pprnm", 600, 600);

if (printPreviewDialog1.ShowDialog() == DialogResult.OK)

{

printDocument1.Print();

}

}

}

int prodid, prodqty, prodprice, tottal, pos = 60, posTabl = 80;

// Удалить позицию из наряда

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

{

foreach (DataGridViewRow row in BillDgv.SelectedRows)

{

BillDgv.Rows.RemoveAt(row.Index);

}

total = Convert.ToInt32(QtyTb.Text) \* Convert.ToInt32(PriceTb.Text);

Grdtotal = Grdtotal - total;

TotalLbl.Text = Grdtotal + "Руб.";

QtyTb.Text = "";

}

// Действия при выборе ячейки

private void BillDgv\_CellClick(object sender, DataGridViewCellEventArgs e)

{

BookId = Convert.ToInt32(BillDgv.SelectedRows[0].Cells[0].Value);

BTitle.Text = BillDgv.SelectedRows[0].Cells[1].Value.ToString();

QtyTb.Text = BillDgv.SelectedRows[0].Cells[3].Value.ToString();

PriceTb.Text = BillDgv.SelectedRows[0].Cells[4].Value.ToString();

//stock = Convert.ToInt32(BillDgv.SelectedRows[0].Cells[4].Value);

}

// Поисковая строка

private void textBox1\_TextChanged(object sender, EventArgs e)

{

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.BookTbls.Load();

// Привязываем данные к dataGridView

BindingList<DataBase.Materials> list = db.BookTbls.Local.ToBindingList();

BookDGV.DataSource = list;

BookDGV.RowHeadersVisible = false;

// Запрос поиска

BookDGV.DataSource = db.BookTbls.Where(x => x.BId.ToString().Contains(txtSearch.Text)

|| x.BTitle.ToString().Contains(txtSearch.Text)

|| x.BCat.ToString().Contains(txtSearch.Text)

|| x.BAuthor.Contains(txtSearch.Text)).ToList();

}

// Открыть форму авторизации

private void label10\_Click(object sender, EventArgs e)

{

Login flogin = new Login();

flogin.Show();

this.Hide();

}

// Передача параметров в форму

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

{

UserNameLbl.Text = Login.UserName;

ClientNameTb.Text = Login.UserName;

}

string prodname;

// Метод формирования и печати наряда

private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawString(DateTime.Now.ToString(), new Font("Centery Gothic", 6, FontStyle.Bold), Brushes.Red, new Point(5, 2));

e.Graphics.DrawString("Надяр на получении ТМЦ", new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(200,10));

e.Graphics.DrawString("Получатель: " + UserNameLbl.Text, new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(26, 30));

e.Graphics.DrawString("ID", new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(26, pos + 20));

e.Graphics.DrawString("Наименование", new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(50, pos + 20));

e.Graphics.DrawString("Цена", new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(300, pos + 20));

e.Graphics.DrawString("Количество", new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(400, pos + 20));

e.Graphics.DrawString("Сумма", new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(500, pos + 20));

foreach (DataGridViewRow row in BillDgv.Rows)

{

prodid = Convert.ToInt32(row.Cells["Column1"].Value);

prodname = "" + row.Cells["Column2"].Value;

prodprice = Convert.ToInt32(row.Cells["Column3"].Value);

prodqty = Convert.ToInt32(row.Cells["Column4"].Value);

tottal = Convert.ToInt32(row.Cells["Column5"].Value);

e.Graphics.DrawString("" + prodid, new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Blue, new Point(26, posTabl + 20));

e.Graphics.DrawString("" + prodname, new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Blue, new Point(50, posTabl + 20));

e.Graphics.DrawString("" + prodprice, new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Blue, new Point(300, posTabl + 20));

e.Graphics.DrawString("" + prodqty, new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Blue, new Point(400, posTabl + 20));

e.Graphics.DrawString("" + tottal, new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Blue, new Point(500, posTabl + 20));

posTabl = posTabl + 20;

}

e.Graphics.DrawString("Общая сумма: " + Grdtotal + " Руб.", new Font("Centery Gothic", 12, FontStyle.Bold), Brushes.Red, new Point(26, posTabl + 40));

e.Graphics.DrawString("Выдал \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Расшифровка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", new Font("Centery Gothic", 10, FontStyle.Bold), Brushes.Red, new Point(26, posTabl + 70));

e.Graphics.DrawString("Принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Расшифровка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", new Font("Centery Gothic", 10, FontStyle.Bold), Brushes.Red, new Point(26, posTabl + 100));

BillDgv.Rows.Clear();

BillDgv.Refresh();

pos = 100;

Grdtotal = 0;

}

}

}

### Форма справочник материалы

using BoolTest.DataBase;

using System;

using System.ComponentModel;

using System.Data;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

using System.Windows.Forms;

namespace BoolTest

{

public partial class Users : Form

{

ModelKeyBTcontext db;

public Users()

{

InitializeComponent();

populate();

}

// Закрыть программу

private void label5\_Click(object sender, EventArgs e)

{

Application.Exit();

}

// добавить данные в таблицу

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

{

if (UnameTb.Text == "" || PhoneTb.Text == "" || AddTb.Text == "" || PassTb.Text == "")

{

MessageBox.Show("Missing information");

}

else

{

try

{

// Создаём экземпляр класса Справочник\_сотрудники,

// т.е получаем данные о нашем сотруднике из текстовых полей

UserTbl user = new UserTbl();

{

user.UName = UnameTb.Text;

user.UPhone = PhoneTb.Text;

user.UAdd = AddTb.Text;

user.UPass = PassTb.Text;

}

// Заносим данные в нашу таблицу

db.UserTbls.Add(user);

// Обязательно сохраняем изменения

db.SaveChanges();

// Обновляем наш dataGridView, чтобы в нём отобразилась новая страна

UserDGV.Refresh();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

// изменить данные в таблице

private void btnClear(object sender, EventArgs e)

{

try

{

// Проверяем, что выбрана запись

if (txtID.Text == String.Empty)

{

MessageBox.Show("Не выбрана запись!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

// Получаем id из текстового поля

int id = Convert.ToInt32(txtID.Text);

// Находим запись по этому id с помощью метода Find()

UserTbl user = db.UserTbls.Find(id);

if (user == null) return;

user.UName = UnameTb.Text;

user.UPhone = PhoneTb.Text;

user.UAdd = AddTb.Text;

user.UPass = PassTb.Text;

// Добавляем или обновляем запись

db.UserTbls.AddOrUpdate(user);

// Сохраняем в бузу данных

db.SaveChanges();

// Обновляем таблицу с данными

UserDGV.Refresh();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

// Удалить данные из таблицы

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

{

// Проверяем, что выбрана запись

if (txtID.Text == String.Empty)

{

MessageBox.Show("Не выбрана запись!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

//

if (MessageBox.Show("Удалить запись?", "Сообщение", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Получаем id из текстового поля

int id = Convert.ToInt32(txtID.Text);

// Находим запись по этому id с помощью метода Find()

UserTbl user = db.UserTbls.Find(id);

if (user == null) return;

//Если запись выбрана, вносим значения в поля

user.UName = UnameTb.Text;

user.UPhone = PhoneTb.Text;

user.UAdd = AddTb.Text;

user.UPass = PassTb.Text;

// Удаляем запись из базы данных

db.UserTbls.Remove(user);

// Сохраняем в бузу данных

db.SaveChanges();

// Обновляем таблицу с данными

UserDGV.Refresh();

Reset();

}

}

// Кнопка очистить

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

{

Reset();

}

// Метод очистки текстовых полей

private void Reset()

{

UnameTb.Text = "";

PhoneTb.Text = "";

AddTb.Text = "";

PassTb.Text = "";

}

//Метод загрузки данных в таблицу

private void populate()

{

// Открываем и закрываем соединение с базой данных с использованием очисти памяти

using (db)

{

// Создаём объект нашего контекста

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.UserTbls.Load();

// Привязываем данные к dataGridView

UserDGV.DataSource = db.UserTbls.Local.ToBindingList();

UserDGV.Columns[0].Visible = false;

UserDGV.Columns[1].HeaderText = "ФИО";

UserDGV.Columns[2].HeaderText = "Телефон";

UserDGV.Columns[3].HeaderText = "Адрес";

UserDGV.Columns[4].HeaderText = "Пароль";

}

}

// Действия при выборе ячейки в таблице

private void UserDGV\_CellClick(object sender, DataGridViewCellEventArgs e)

{ // Проверка выборки строк

// Если строка не выбрана, то дальше ничего не происходит

if (UserDGV.CurrentRow == null) return;

// Получаем выделенную строку и приводим её у типу Countries

UserTbl location = UserDGV.CurrentRow.DataBoundItem as UserTbl;

// Если мы щёлкаем по пустой строке, то ничего дальше не делаем

if (location == null) return;

txtID.Text = location.UId.ToString();

UnameTb.Text = UserDGV.SelectedRows[0].Cells[1].Value.ToString();

PhoneTb.Text = UserDGV.SelectedRows[0].Cells[2].Value.ToString();

AddTb.Text = UserDGV.SelectedRows[0].Cells[3].Value.ToString();

PassTb.Text = UserDGV.SelectedRows[0].Cells[4].Value.ToString();

}

// Открыть форму авторизации

private void label11\_Click(object sender, EventArgs e)

{

Login flogin = new Login();

flogin.Show();

this.Hide();

}

// Открыть форму справочника материалы

private void label7\_Click(object sender, EventArgs e)

{

Materials flogin = new Materials();

flogin.Show();

this.Hide();

}

// Открыть форму справочника статистика

private void label10\_Click(object sender, EventArgs e)

{

DashBoard flogin = new DashBoard();

flogin.Show();

this.Hide();

}

// Открыть форму справочника категории

private void label14\_Click(object sender, EventArgs e)

{

category fCat = new category();

fCat.Show();

this.Hide();

}

// Строка поиска

private void textBox1\_TextChanged(object sender, EventArgs e)

{

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.UserTbls.Load();

// Привязываем данные к dataGridView

BindingList<DataBase.Materials> list = db.BookTbls.Local.ToBindingList();

UserDGV.DataSource = list;

UserDGV.RowHeadersVisible = false;

// Запрос поиска

UserDGV.DataSource = db.UserTbls.Where(x => x.UId.ToString().Contains(txtSearch.Text)

|| x.UName.ToString().Contains(txtSearch.Text)

|| x.UAdd.ToString().Contains(txtSearch.Text)

|| x.UPhone.Contains(txtSearch.Text)).ToList();

}

}

}

### Форма справочник пользователи

using BoolTest.DataBase;

using System;

using System.ComponentModel;

using System.Data;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

using System.Windows.Forms;

namespace BoolTest

{

public partial class Users : Form

{

ModelKeyBTcontext db;

public Users()

{

InitializeComponent();

populate();

}

// Закрыть программу

private void label5\_Click(object sender, EventArgs e)

{

Application.Exit();

}

// добавить данные в таблицу

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

{

if (UnameTb.Text == "" || PhoneTb.Text == "" || AddTb.Text == "" || PassTb.Text == "")

{

MessageBox.Show("Missing information");

}

else

{

try

{

// Создаём экземпляр класса Справочник\_сотрудники,

// т.е получаем данные о нашем сотруднике из текстовых полей

UserTbl user = new UserTbl();

{

user.UName = UnameTb.Text;

user.UPhone = PhoneTb.Text;

user.UAdd = AddTb.Text;

user.UPass = PassTb.Text;

}

// Заносим данные в нашу таблицу

db.UserTbls.Add(user);

// Обязательно сохраняем изменения

db.SaveChanges();

// Обновляем наш dataGridView, чтобы в нём отобразилась новая страна

UserDGV.Refresh();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

// изменить данные в таблице

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

{

try

{

// Проверяем, что выбрана запись

if (txtID.Text == String.Empty)

{

MessageBox.Show("Не выбрана запись!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

// Получаем id из текстового поля

int id = Convert.ToInt32(txtID.Text);

// Находим запись по этому id с помощью метода Find()

UserTbl user = db.UserTbls.Find(id);

if (user == null) return;

user.UName = UnameTb.Text;

user.UPhone = PhoneTb.Text;

user.UAdd = AddTb.Text;

user.UPass = PassTb.Text;

// Добавляем или обновляем запись

db.UserTbls.AddOrUpdate(user);

// Сохраняем в бузу данных

db.SaveChanges();

// Обновляем таблицу с данными

UserDGV.Refresh();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

// Удалить данные из таблицы

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

{

// Проверяем, что выбрана запись

if (txtID.Text == String.Empty)

{

MessageBox.Show("Не выбрана запись!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

//

if (MessageBox.Show("Удалить запись?", "Сообщение", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Получаем id из текстового поля

int id = Convert.ToInt32(txtID.Text);

// Находим запись по этому id с помощью метода Find()

UserTbl user = db.UserTbls.Find(id);

if (user == null) return;

//Если запись выбрана, вносим значения в поля

user.UName = UnameTb.Text;

user.UPhone = PhoneTb.Text;

user.UAdd = AddTb.Text;

user.UPass = PassTb.Text;

// Удаляем запись из базы данных

db.UserTbls.Remove(user);

// Сохраняем в бузу данных

db.SaveChanges();

// Обновляем таблицу с данными

UserDGV.Refresh();

Reset();

}

}

// Кнопка очистить

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

{

Reset();

}

// Метод очистки текстовых полей

private void Reset()

{

UnameTb.Text = "";

PhoneTb.Text = "";

AddTb.Text = "";

PassTb.Text = "";

}

//Метод загрузки данных в таблицу

private void populate()

{

// Открываем и закрываем соединение с базой данных с использованием очисти памяти

using (db)

{

// Создаём объект нашего контекста

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.UserTbls.Load();

// Привязываем данные к dataGridView

UserDGV.DataSource = db.UserTbls.Local.ToBindingList();

UserDGV.Columns[0].Visible = false;

UserDGV.Columns[1].HeaderText = "ФИО";

UserDGV.Columns[2].HeaderText = "Телефон";

UserDGV.Columns[3].HeaderText = "Адрес";

UserDGV.Columns[4].HeaderText = "Пароль";

}

}

// Действия при выборе ячейки в таблице

private void UserDGV\_CellClick(object sender, DataGridViewCellEventArgs e)

{ // Проверка выборки строк

// Если строка не выбрана, то дальше ничего не происходит

if (UserDGV.CurrentRow == null) return;

// Получаем выделенную строку и приводим её у типу Countries

UserTbl location = UserDGV.CurrentRow.DataBoundItem as UserTbl;

// Если мы щёлкаем по пустой строке, то ничего дальше не делаем

if (location == null) return;

txtID.Text = location.UId.ToString();

UnameTb.Text = UserDGV.SelectedRows[0].Cells[1].Value.ToString();

PhoneTb.Text = UserDGV.SelectedRows[0].Cells[2].Value.ToString();

AddTb.Text = UserDGV.SelectedRows[0].Cells[3].Value.ToString();

PassTb.Text = UserDGV.SelectedRows[0].Cells[4].Value.ToString();

}

// Открыть форму авторизации

private void label11\_Click(object sender, EventArgs e)

{

Login flogin = new Login();

flogin.Show();

this.Hide();

}

// Открыть форму справочника материалы

private void label7\_Click(object sender, EventArgs e)

{

Materials flogin = new Materials();

flogin.Show();

this.Hide();

}

// Открыть форму справочника статистика

private void label10\_Click(object sender, EventArgs e)

{

DashBoard flogin = new DashBoard();

flogin.Show();

this.Hide();

}

// Открыть форму справочника категории

private void label14\_Click(object sender, EventArgs e)

{

category fCat = new category();

fCat.Show();

this.Hide();

}

// Строка поиска

private void textBox1\_TextChanged(object sender, EventArgs e)

{

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.UserTbls.Load();

// Привязываем данные к dataGridView

BindingList<DataBase.Materials> list = db.BookTbls.Local.ToBindingList();

UserDGV.DataSource = list;

UserDGV.RowHeadersVisible = false;

// Запрос поиска

UserDGV.DataSource = db.UserTbls.Where(x => x.UId.ToString().Contains(txtSearch.Text)

|| x.UName.ToString().Contains(txtSearch.Text)

|| x.UAdd.ToString().Contains(txtSearch.Text)

|| x.UPhone.Contains(txtSearch.Text)).ToList();

}

}

}

### Форма справочника категории

using BoolTest.DataBase;

using System;

using System.ComponentModel;

using System.Data;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

using System.Windows.Forms;

namespace BoolTest

{

public partial class category : Form

{

public category()

{

InitializeComponent();

}

ModelKeyBTcontext db;

private void label5\_Click(object sender, EventArgs e)

{

Application.Exit();

}

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

{

if (txtNameCategoty.Text == "")

{

MessageBox.Show("Missing information");

}

else

{

try

{

// Создаём экземпляр класса Справочник\_сотрудники,

// т.е получаем данные о нашем сотруднике из текстовых полей

Seria cat = new Seria();

{

cat.Name = txtNameCategoty.Text;

}

// Заносим данные в нашу таблицу

db.Serias.Add(cat);

// Обязательно сохраняем изменения

db.SaveChanges();

// Обновляем наш dataGridView, чтобы в нём отобразилась новая страна

dvgCategory.Refresh();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

}

void Reset()

{

txtNameCategoty.Text = "";

txtID.Text = "";

txtSearch.Text = "";

}

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

{

try

{

// Проверяем, что выбрана запись

if (txtID.Text == String.Empty)

{

MessageBox.Show("Не выбрана запись!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

// Получаем id из текстового поля

int id = Convert.ToInt32(txtID.Text);

// Находим запись по этому id с помощью метода Find()

Seria cat = db.Serias.Find(id);

if (cat == null) return;

cat.Name = txtNameCategoty.Text;

// Добавляем или обновляем запись

db.Serias.AddOrUpdate(cat);

// Сохраняем в бузу данных

db.SaveChanges();

// Обновляем таблицу с данными

dvgCategory.Refresh();

Reset();

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message);

}

}

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

{

// Проверяем, что выбрана запись

if (txtID.Text == String.Empty)

{

MessageBox.Show("Не выбрана запись!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

//

if (MessageBox.Show("Удалить запись?", "Сообщение", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Получаем id из текстового поля

int id = Convert.ToInt32(txtID.Text);

// Находим запись по этому id с помощью метода Find()

Seria cat = db.Serias.Find(id);

if (cat == null) return;

//Если запись выбрана, вносим значения в поля

cat.Name = txtNameCategoty.Text;

// Удаляем запись из базы данных

db.Serias.Remove(cat);

// Сохраняем в бузу данных

db.SaveChanges();

// Обновляем таблицу с данными

dvgCategory.Refresh();

Reset();

}

}

private void txtSearch\_TextChanged(object sender, EventArgs e)

{

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.Serias.Load();

// Привязываем данные к dataGridView

BindingList<DataBase.Materials> list = db.BookTbls.Local.ToBindingList();

dvgCategory.DataSource = list;

dvgCategory.RowHeadersVisible = false;

// Запрос поиска

dvgCategory.DataSource = db.Serias.Where(x => x.Id.ToString().Contains(txtSearch.Text)

|| x.Name.Contains(txtSearch.Text)).ToList();

}

private void dvgCategory\_CellClick(object sender, DataGridViewCellEventArgs e)

{

// Проверка выборки строк

// Если строка не выбрана, то дальше ничего не происходит

if (dvgCategory.CurrentRow == null) return;

// Получаем выделенную строку и приводим её у типу Countries

Seria location = dvgCategory.CurrentRow.DataBoundItem as Seria;

// Если мы щёлкаем по пустой строке, то ничего дальше не делаем

if (location == null) return;

txtID.Text = location.Id.ToString();

txtNameCategoty.Text = dvgCategory.SelectedRows[0].Cells[1].Value.ToString();

}

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

{

using (db)

{

// Создаём объект нашего контекста

db = new ModelKeyBTcontext();

// Загружаем данные из таблицы в кэш

db.Serias.Load();

// Привязываем данные к dataGridView

dvgCategory.DataSource = db.Serias.Local.ToBindingList();

dvgCategory.Columns[0].Width = 70;

dvgCategory.Columns[0].HeaderText = "ID";

dvgCategory.Columns[1].HeaderText = "Категория";

}

}

private void label7\_Click(object sender, EventArgs e)

{

Materials flogin = new Materials();

flogin.Show();

this.Hide();

}

private void label6\_Click(object sender, EventArgs e)

{

Users userForm = new Users();

userForm.Show();

this.Hide();

}

private void label10\_Click(object sender, EventArgs e)

{

DashBoard flogin = new DashBoard();

flogin.Show();

this.Hide();

}

private void label11\_Click(object sender, EventArgs e)

{

Login flogin = new Login();

flogin.Show();

this.Hide();

}

private void label5\_Click\_1(object sender, EventArgs e)

{

Application.Exit();

}

}

}

### Форма статистика

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace BoolTest

{

public partial class DashBoard : Form

{

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=BookShopDb;Integrated Security=True");

public DashBoard()

{

InitializeComponent();

}

private void label11\_Click(object sender, EventArgs e)

{

Login flogin = new Login();

flogin.Show();

this.Hide();

}

private void label5\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void label6\_Click(object sender, EventArgs e)

{

Users flogin = new Users();

flogin.Show();

this.Hide();

}

private void label7\_Click(object sender, EventArgs e)

{

Materials flogin = new Materials();

flogin.Show();

this.Hide();

}

private void label3\_Click(object sender, EventArgs e)

{

category fCat = new category();

fCat.Show();

this.Hide();

}

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

{

Con.Open();

string querty = "SELECT SUM(BQty) FROM BookTbl";

SqlDataAdapter sda = new SqlDataAdapter(querty, Con);

DataTable dt = new DataTable();

sda.Fill(dt);

lbMStock.Text = dt.Rows[0][0].ToString();

string qAmount = "SELECT SUM(Amount) FROM BillTbl";

SqlDataAdapter sdAm = new SqlDataAdapter(qAmount, Con);

DataTable dt1 = new DataTable();

sdAm.Fill(dt1);

lbqAmount.Text = dt1.Rows[0][0].ToString();

string qUsers = "SELECT Count(\*) FROM UserTbl";

SqlDataAdapter sUser = new SqlDataAdapter(qUsers, Con);

DataTable dt2 = new DataTable();

sUser.Fill(dt2);

lbUserStok.Text = dt2.Rows[0][0].ToString();

string q = "SELECT \* FROM BillTbl";

// Заполнение таблицы

SqlDataAdapter sdvg = new SqlDataAdapter(q, Con);

DataSet ds = new System.Data.DataSet();

sdvg.Fill(ds);

datagridDash.DataSource = ds.Tables["Table"].DefaultView;

// Заполнение шапки таблицы

datagridDash.Columns[0].Visible = false;

datagridDash.Columns[1].HeaderText = "Пользователь";

datagridDash.Columns[2].HeaderText = "Клиент";

datagridDash.Columns[3].HeaderText = "Колличество";

Con.Close();

}

}

}

### Форма авторизации администратора

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace BoolTest

{

public partial class AdminLoginForm : Form

{

public AdminLoginForm()

{

InitializeComponent();

}

// Завершение работы программы

private void label5\_Click(object sender, EventArgs e)

{

Application.Exit();

}

// Открыть форму авторизации Пользователя

private void label4\_Click(object sender, EventArgs e)

{

Login login = new Login();

login.Show();

this.Hide();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=BookShopDb;Integrated Security=True");

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

{

if (UnameTb.Text == "admin" && UPassTb.Text == "123")

{

string querty = "SELECT COUNT (\*) FROM UserTbl where Uname=N'" + UnameTb.Text + "' and UPass=N'" + UPassTb.Text + "'";

Con.Open();

SqlDataAdapter sda = new SqlDataAdapter(querty, Con);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows[0][0].ToString() == "1")

{

Materials obj = new Materials();

obj.Show();

this.Hide();

Con.Close();

}

}

else

{

MessageBox.Show("Wrong your password");

}

Con.Close();

}

}

}

### Форма авторизации администратора

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace BoolTest

{

public partial class Login : Form

{

public Login()

{

InitializeComponent();

}

SqlConnection Con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=BookShopDb;Integrated Security=True");

private void label5\_Click(object sender, EventArgs e)

{

Application.Exit();

}

public static string UserName = "";

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

{

string querty = "SELECT COUNT (\*) FROM UserTbl where Uname=N'" + UnameTb.Text + "' and UPass=N'" + UPassTb.Text + "'";

Con.Open();

SqlDataAdapter sda = new SqlDataAdapter(querty, Con);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows[0][0].ToString() == "1")

{

UserName = UnameTb.Text;

Billing obj = new Billing();

obj.Show();

this.Hide();

Con.Close();

}

else

{

MessageBox.Show("Wrong your password");

}

Con.Close();

}

private void label4\_Click(object sender, EventArgs e)

{

AdminLoginForm admFrom = new AdminLoginForm();

admFrom.Show();

this.Hide();

}

}

}

# Классы базы данных

### Определение производного класса DbContext

using System.Data.Entity;

namespace BoolTest.DataBase

{

public partial class ModelKeyBTcontext : DbContext

{

public ModelKeyBTcontext()

: base("name=Model1")

{

}

public virtual DbSet<BillTbl> BillTbls { get; set; }

public virtual DbSet<Materials> BookTbls { get; set; }

public virtual DbSet<Jornal> Jornals { get; set; }

public virtual DbSet<Seria> Serias { get; set; }

public virtual DbSet<UserTbl> UserTbls { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

modelBuilder.Entity<Seria>()

.Property(e => e.Name)

.IsFixedLength();

}

}

}

### Класс формы статистики

namespace BoolTest.DataBase

{

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

[Table("BillTbl")]

public partial class BillTbl

{

[Key]

public int BillId { get; set; }

[Required]

[StringLength(50)]

public string UName { get; set; }

[Required]

[StringLength(50)]

public string ClientName { get; set; }

public int Amount { get; set; }

}

}

### Класс справочник материалы

namespace BoolTest.DataBase

{

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

[Table("BookTbl")]

public partial class Materials

{

[Key]

public int BId { get; set; }

[Required]

[StringLength(100)]

public string BTitle { get; set; }

[Required]

[StringLength(50)]

public string BAuthor { get; set; }

[Required]

[StringLength(50)]

public string BCat { get; set; }

public int BQty { get; set; }

public int BPrice { get; set; }

}

}

### Класс справочник категории

namespace BoolTest.DataBase

{

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

[Table("Seria")]

public partial class Seria

{

public int Id { get; set; }

[Required]

[StringLength(50)]

public string Name { get; set; }

}

}

### Класс справочника пользователи

namespace BoolTest.DataBase

{

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

[Table("UserTbl")]

public partial class UserTbl

{

[Key]

public int UId { get; set; }

[Required]

[StringLength(50)]

public string UName { get; set; }

[Required]

[StringLength(50)]

public string UPhone { get; set; }

[Required]

[StringLength(50)]

public string UAdd { get; set; }

[Required]

[StringLength(50)]

public string UPass { get; set; }

}

}

1. Составлено автором в среде программы “BPwin” [↑](#footnote-ref-1)
2. Составлено автором в среде программы “ BPwin ” [↑](#footnote-ref-2)
3. Составлено автором в среде программы “Bpwin” [↑](#footnote-ref-3)
4. Составлено автором в среде MS Word [↑](#footnote-ref-4)
5. Составлено автором в среде MS Word [↑](#footnote-ref-5)
6. Составлено автором в среде MS Word [↑](#footnote-ref-6)
7. Составлено автором в среде MS Word [↑](#footnote-ref-7)
8. Составлено автором в MS Word [↑](#footnote-ref-8)
9. Составлено автором в MS Word [↑](#footnote-ref-9)
10. Составлено автором в MS Word [↑](#footnote-ref-10)
11. Составлено автором в MS Word [↑](#footnote-ref-11)
12. Составлено автором в среде разработки Microsoft SQL Server Management Studio 18.4 [↑](#footnote-ref-12)
13. Составлено автором в среде MS Word [↑](#footnote-ref-13)
14. Составлено автором в MS Word [↑](#footnote-ref-14)
15. Составлено автором в MS Word [↑](#footnote-ref-15)
16. Составлено автором в MS Word [↑](#footnote-ref-16)
17. Составлено автором в среде MS Word [↑](#footnote-ref-17)
18. Составлено автором в среде MS Visual Studio 2019 [↑](#footnote-ref-18)
19. Составлено автором в среде MS Word [↑](#footnote-ref-19)
20. Разработано автором в среде MS Word [↑](#footnote-ref-20)
21. Разработано автором в среде MS Word [↑](#footnote-ref-21)
22. Разработано автором в среде MS Word [↑](#footnote-ref-22)
23. Разработано автором в среде MS Word [↑](#footnote-ref-23)
24. Составлено автором в среде MS Word [↑](#footnote-ref-24)
25. Составлено автором в среде MS Word [↑](#footnote-ref-25)
26. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-26)
27. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-27)
28. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-28)
29. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-29)
30. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-30)
31. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-31)
32. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-32)
33. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-33)
34. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-34)
35. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-35)
36. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-36)
37. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-37)
38. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-38)
39. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-39)
40. Разработано автором в среде MS Visual Studio 2019 [↑](#footnote-ref-40)