**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc104496616)

[1 ТЕХНИКО-ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ПРЕДМЕТНОЙ ОБЛАСТИ 3](#_Toc104496617)

[1.1 Общее представление об автоматизированных рабочих местах 3](#_Toc104496618)

[2 СТРУКТУРА ОБЪЕКТА АВТОМАТИЗАЦИИ 5](#_Toc104496620)

[2.1 Формализация функций педагога дополнительного образования 5](#_Toc104496621)

[2.2 Разработка технической документации на этапе проектирования объекта автоматизации 5](#_Toc104496622)

[2.3 Функциональная модель автоматизированного рабочего места 6](#_Toc104496623)

[2.4 Объектно-ориентированная модель объекта автоматизации 10](#_Toc104496624)

[2.5 Общая характеристика организации решения задачи на ПК 18](#_Toc104496625)

[3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА 20](#_Toc104496626)

[3.1. Выбор средства разработки 20](#_Toc104496627)

[3.2. Разработка базы данных 20](#_Toc104496628)

[4.1 Обзор аналогов разрабатываемого программного обеспечения 33](#_Toc104496629)

[4.2 Оценка эффективности внедряемого программного продукта 34](#_Toc104496630)

[ЗАКЛЮЧЕНИЕ 36](#_Toc104496631)

[Список информационных источников 37](#_Toc104496636)

[Приложение 1 39](#_Toc104496637)

[Приложение 2 39](#_Toc104496638)

[Приложение 3 41](#_Toc104496639)

# ВВЕДЕНИЕ

Научно-технический прогресс привел к тому, что практически каждый использует электронные вычислительные машины в своей деятельности, будь то обычный стационарный компьютер, ноутбук или планшет.

Кроме того, в настоящее время в России большое внимание на государственном уровне уделяется информированию общества в целом.

Постоянное увеличение объемов и интенсивности информационных потоков приводит к необходимости использования информационные средства и технологии для повышения эффективности обработки, надежности и удобства представления. Но для полного перехода на традиционный бумажный документооборот преподавательского состава необходимо предусмотреть инструментарий для отображения в электронном виде как минимум тех же функций, однако часть функций при этом необходимо автоматизировать.

Целью квалификационной работы является разработка приложения для учета товаров для ИП Шитиков В.В на языке программирования C#.

Для достижения целей и задач необходимо решить ряд задач:

* Анализ требований к разработке программного продукта. На этом этапе будет проведён сбор требований к программному обеспечению, их систематизация, выявление взаимосвязей, а также точное определение функции продукта и способа его интеграции в существующие процессы,
* Выбор среды разработки – исследование существующих сред разработки и выбор наиболее подходящей для выбранного языка программирования и простой в использовании,
* Проектирование информационной системы – проектирование экранных форм, объектов данных и их структуры, определение способов взаимодействия пользователя с программой,
* Разработка схемы базы данных – описание таблиц для хранения данных о каждой сущности в базе данных, определение связей между таблицами и генерация сценария для подготовки схемы данных к использованию на сервере,
* Реализация информационной системы – преобразование результатов проектирования в готовый программный продукт с использованием выбранного стека технологий,
* Тестирование и отладка информационной системы – установление программных ошибок, их локализация и последующее устранение.

# 

# 1 ТЕХНИКО-ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ПРЕДМЕТНОЙ ОБЛАСТИ

# Общее представление об автоматизированных рабочих местах

Автоматизированное рабочее место (АРМ) комплекс среди вычислительной техники и программного обеспечения, размещающийся, напрямую на рабочем месте работника и предназначенный для компьютеризации его работы в рамках квалификации. Автоматизированные рабочие места должны формироваться строго в соответствии с их рассчитываемым функциональным назначением. В прочем общие законы формирования автоматизированного рабочего места остаются незыблемыми, к ним относят:

* Системность
* Гибкость
* Устойчивость
* Эффективность

Автоматизация — это широкое понятие, которое применимо к современным рабочим местам. Практически в любом бизнесе и во всех отраслях есть возможности для большей автоматизации.

Автоматизация не заменит людей, но освободит их от выполнения рутинных и повторяющихся задач.

Существует распространенное заблуждение, что автоматизация предполагает высокую робототехнику, но это может быть так же просто, как набор инструментов, размещенных в обычных бизнес-программах. По своей сути автоматизация заключается во внедрении системы для выполнения повторяющихся и легко воспроизводимых задач без использования человеческого труда.

Автоматизируя повторяющиеся бизнес-процессы, сотрудники освобождаются для выполнения задач, которые являются более ценными, чем те, которые могут быть выполнены машинами.

# 2 СТРУКТУРА ОБЪЕКТА АВТОМАТИЗАЦИИ

# 2.1 Формализация функций педагога дополнительного образования

Торговля является огромной отраслью экономики России. В эту сферу вовлечено практически все население страны либо в качестве продавцов, либо в качестве покупателей. В условиях рыночных отношений торговля занимает все более значительное место в структуре общественного производства.

Основными хозяйственными процессами в торговле являются приобретение товаров, доставка, хранение товаров и их дальнейшая реализация.  
 С развитием компьютерных технологий и программного обеспечения, появилась необходимость создания программного продукта, способного повысить производительность труда, при этом сократив усилия и затраты человека. Сегодня любое предприятие, в том числе и самое малое, заинтересованно во внедрении подобного программного обеспечения как неотъемлемой части своей деятельности.

Компьютерный учет имеет свои особенности и радикально отличается от обычного. Компьютер не только облегчает учет, сокращая время, требующееся на оформление документов и обобщение накопленных данных для анализа хода торговой деятельности, необходимого для управления ею.

Отчеты о положении в торговле, получаемые с помощью компьютера, можно получить и без него – никакой особой математики в компьютере не содержится – но на расчеты уйдет столько времени, что они уже ни на что не будут нужны; или ими придется занять такое количество расчетчиков, что на их зарплату уйдет значительно больше, чем будет получено прибыли в результате их расчетов.

Таким образом при применении компьютера “количество переходит в качество”: увеличение скорости расчетов делает возможным качественное улучшение самой схемы построения торговли. Программное обеспечение для работы с базами данных используется на персональных компьютерах уже довольно давно. К сожалению, эти программы либо были элементарными диспетчерами хранения данных и не имели средств разработки приложений, либо были настолько сложны и трудны, что даже хорошо разбирающиеся в компьютерах люди избегали работать с ними до тех пор, пока не получали полных, ориентированных на пользователя приложений.

# 2.2 Разработка технической документации на этапе проектирования объекта автоматизации

Техническое задание — документ или несколько документов, определяющих цель, структуру, свойства и методы какого-либо проекта, и исключающие двусмысленное толкование различными исполнителями.

Техническое задание используется для того, чтобы поставщики, покупатели и пользователи материалов, продуктов или услуг понимали и согласовывали все требования .

Техническое задание - это письменное соглашение между вами и подрядчиком, взаимное видение того, как должен выглядеть ваш веб-сайт или приложение. Это документ, в котором прописаны обязанности подрядчика, имеется возможность вносить изменения в техническое задание . Техническое задание представлено в Приложении 1.

# 2.3 Функциональная модель автоматизированного рабочего места

Контекстная диаграмма, также называемая диаграммой потока данных уровня 0, является самым высоким уровнем в Диаграмме потока данных. Это инструмент, популярный среди бизнес-аналитиков, которые используют его для понимания деталей и границ системы, которая будет разработана в проекте. Он указывает на поток информации между системой и внешними компонентами.

Контекстная диаграмма является частью документа требований в проекте. В отличие от других проектных диаграмм, Контекстная диаграмма предназначена не для использования инженерами/техниками, а для заинтересованных сторон проекта. Поэтому он должен быть изложен простым и понятным языком, чтобы заинтересованные стороны могли легко понять пункты при его анализе.

Диаграмма контекста системы, имеющая большое значение для заинтересованных сторон проекта, привлекает все внимание к внешним факторам и событиям, которые необходимо учитывать при разработке всего набора ограничений и требований системы.

Диаграмма контекста системы представляет все внешние компоненты, которые могут взаимодействовать с системой, следовательно, отображает всю программную систему как единое целое. Этот тип диаграммы помещает систему в середину и окружает ее внешними частями, связанными объектами и окружением. Детали внутренней структуры системы явно опущены.

Кроме того, контекстная диаграмма может быть использована для значительного снижения рисков в проекте.

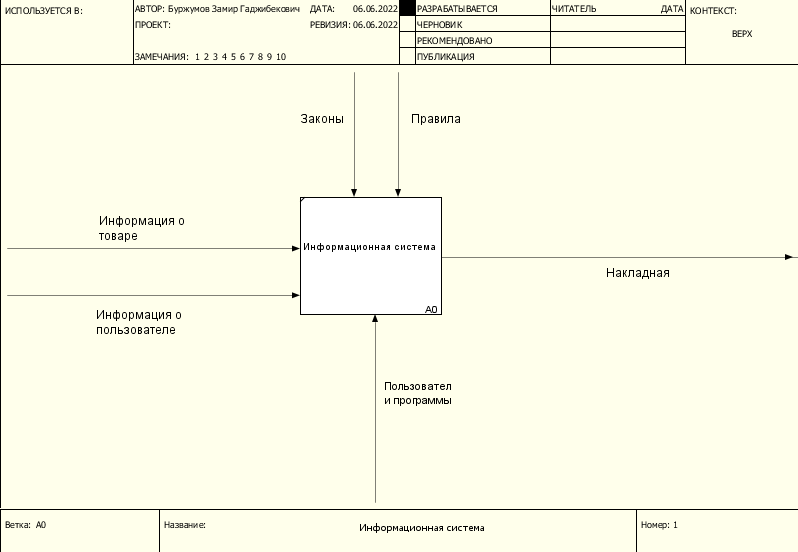


Рисунок 1-Контекстная диаграмма информационной системы

Диаграмма декомпозиции процессов (часто называемая декомпозицией) объясняет разбивку процессов в рамках проекта, бизнес-области или функциональной области. Цель состоит в том, чтобы показать все процессы и определить взаимосвязи и зависимости между ними. Обратите внимание, что декомпозиция не углубляется в "как"; она просто описывает "что".

Процессы, под которыми находятся другие процессы, называются родительскими процессами. Процессы, которые подчиняются другому процессу, называются дочерними процессами.

Разбивается родительский процесс, по крайней мере на два дочерних процесса; в противном случае это не настоящий родительский процесс.

Все дочерние процессы вместе должны полностью описывать все действия в родительском процессе.

Диаграмма не следует какой-либо определенной последовательности.

В отличие от моделей, отображающих структуру организации, работа на диаграмме верхнего уровня в IDEF0 − это не элемент управления нижестоящими работами. Работы нижнего уровня − это то же самое, что и работы верхнего уровня, но в более детальном изложении. Как следствие этого границы работы верхнего уровня - это то же самое, что и границы диаграммы декомпозиции.

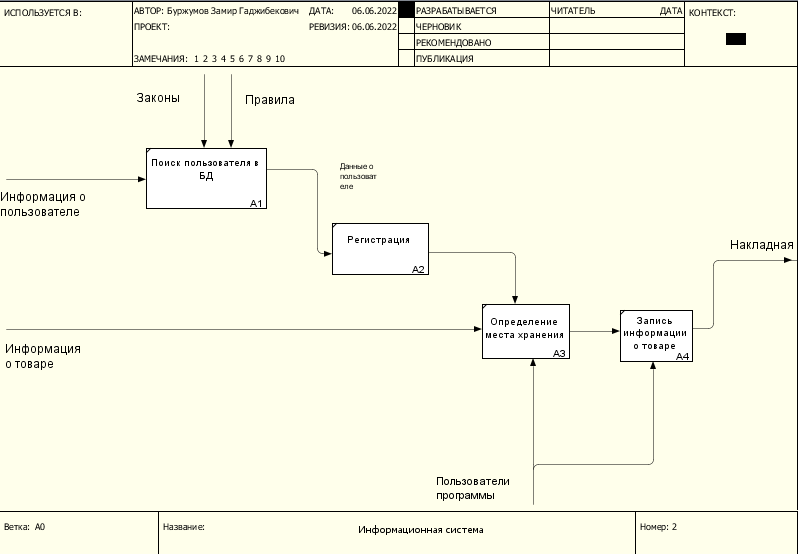


Рисунок 2 - Диаграмма декомпозиции первого уровня

Диаграмма декомпозиции предназначена для детализации работы. В отличие от моделей, отображающих структуру организации, работа на диаграмме верхнего уровня в IDEF0− это не элемент управления нижестоящими работами. Работы нижнего уровня − это то же самое, что и работы верхнего уровня, но в более детальном изложении. Как следствие этого границы работы верхнего уровня - это то же самое, что и границы диаграммы декомпозиции.

При декомпозиции функции входящие в нее и исходящие из нее дуги автоматически появляются на диаграмме декомпозиции (миграция дуг), но при этом не касаются блоков.

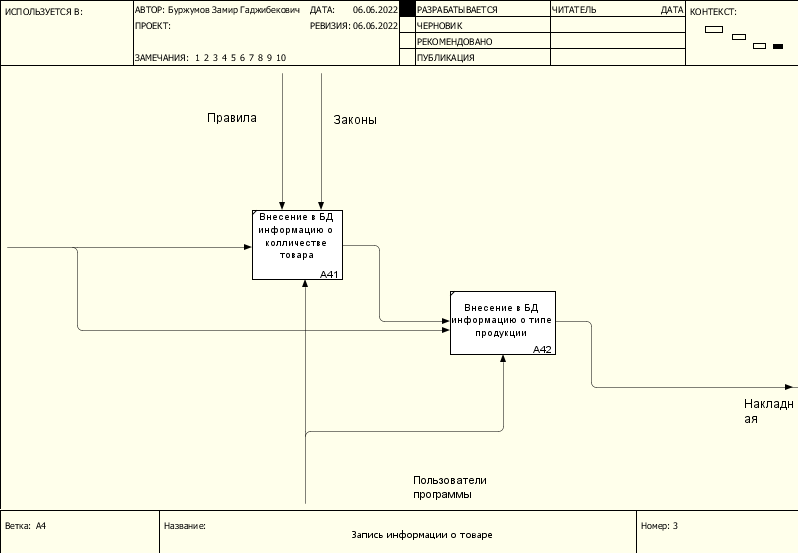


Рисунок 3 - Диаграмма декомпозиции второго уровня

# 2.4 Объектно-ориентированная модель объекта автоматизации

Диаграмма прецедентов

Диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент - возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

Диаграмма вариантов использования UML - это основная форма требований к системе или программному обеспечению для новой недостаточно разработанной программы. Варианты использования определяют ожидаемое поведение , а не точный метод его реализации .

Варианты использования, однажды заданные, могут быть обозначены как текстовым, так и визуальным представлением (т.е. диаграммой вариантов использования). Ключевой концепцией моделирования вариантов использования является то, что оно помогает нам проектировать систему с точки зрения конечного пользователя. Это эффективный метод передачи информации о поведении системы в терминах пользователя путем указания всего видимого извне поведения системы.

В схеме вариантов использования не отображаются подробные сведения о вариантах использования:

* В нем суммируются лишь некоторые взаимосвязи между вариантами использования, субъектами и системами.
* Он не показывает порядок, в котором выполняются шаги для достижения целей каждого варианта использования.

Диаграмма вариантов использования должна быть простой и содержать всего несколько фигур.

Диаграммы вариантов использования обычно разрабатываются на ранней стадии разработки, и люди часто применяют моделирование вариантов использования для следующих целей:

* Указать контекст системы
* Фиксировать требования системы
* Проверка архитектуры системы
* Управление внедрением
* Создание тестовых примеров

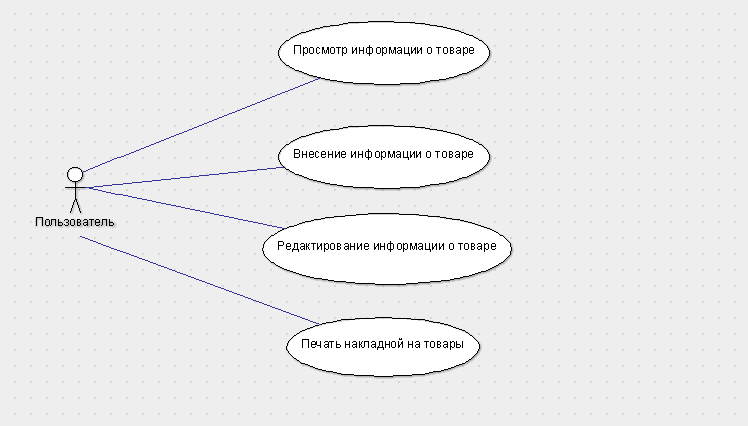


Рисунок 4 - Диаграмма прецедентов

Диаграмма кооперации

Диаграммы кооперации отображают поток событий через конкретный сценарий варианта использования, упорядочены по времени, а кооперативные диаграммы больше внимания заостряют на связях между объектами.

Оно определяет набор участников и отношений, которые имеют значение для заданного набора целей:

* Сотрудничество между объектами, работающими вместе, обеспечивает новые желаемые функциональные возможности в объектно-ориентированных системах.
* Каждый объект частично поддерживает возникающие функциональные возможности.
* Объекты способны создавать (пригодные для использования) высокоуровневые функциональные возможности, работая вместе.
* Объекты сотрудничают, обмениваясь данными (передавая сообщения) друг с другом, чтобы работать вместе

В отличие от диаграммы последовательности, диаграмма кооперации показывает взаимосвязи между объектами. Диаграммы последовательности и диаграммы кооперации отображают схожую информацию, но показывают ее по-разному.

Из-за формата диаграммы кооперации они, как правило, лучше подходят для аналитических действий .В частности, они, как правило, лучше подходят для изображения более простых взаимодействий меньшего числа объектов. Однако, если количество объектов и сообщений растет, диаграмму становится все труднее читать. Кроме того, трудно отобразить дополнительную описательную информацию, такую как время, точки принятия решений или другую неструктурированную информацию, которую можно легко добавить к примечаниям на диаграмме последовательности.

Объекты в диаграммах кооперации можно использовать следующими способами:

* Каждый объект в совместной работе имеет имя и указан его класс
* Не все классы должны отображаться
* В классе может быть более одного объекта
* Класс объекта может быть не указан. Обычно вы сначала создаете диаграмму совместной работы с объектами, а затем указываете их классы.
* Объекты могут быть безымянными, но вы должны дать им имена, если хотите различать разные объекты одного и того же класса.

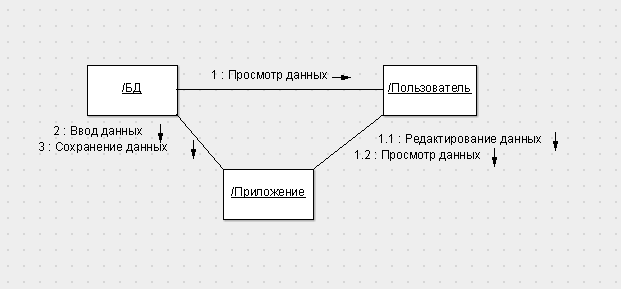


Рисунок 5 - Диаграмма кооперации

Диаграмма деятельности

Диаграммы деятельности представляют собой графическое представление рабочих процессов поэтапных действий и действий с поддержкой выбора, итерации и параллелизма. Они описывают поток управления целевой системой, такой как исследование сложных бизнес - правил и операций, а также описание прецедентов и бизнес-процессов.

Объекты в диаграммах деятельности можно использовать следующими способами:

* Каждый объект в совместной работе имеет имя и указан его класс
* Не все классы должны отображаться
* В классе может быть более одного объекта
* Класс объекта может быть не указан. Обычно вы сначала создаете диаграмму совместной работы с объектами, а затем указываете их классы.
* Объекты могут быть безымянными, но вы должны дать им имена, если хотите различать разные объекты одного и того же класса.

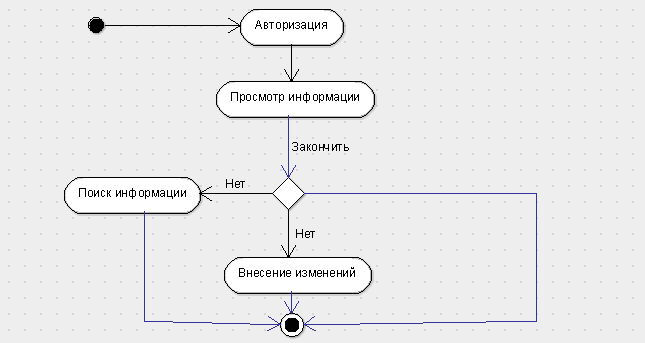


Рисунок 6 - Диаграмма деятельности

Диаграмма состояния

Диаграмма состояния – тип диаграммы, используемый для описания поведения системы.

Поведение объекта является не только прямым следствием его входных данных, но и зависит от его предыдущего состояния. Прошлую историю объекта лучше всего можно смоделировать с помощью диаграммы состояния.

Диаграммы состояния UML (или иногда называемые конечным автоматом) показывают различные состояния объекта. Диаграммы состояния также могут показывать, как объект реагирует на различные события, переходя из одного состояния в другое. Диаграмма состояния - это UML-диаграмма, используемая для моделирования динамической природы системы.

Диаграмма состояния обычно используется для описания зависящего от состояния поведения объекта. Объект по-разному реагирует на одно и то же событие в зависимости от того, в каком состоянии он находится. Диаграммы состояния обычно применяются к объектам, но могут быть применены к любому элементу, который имеет поведение по отношению к другим объектам, таким как: субъекты, варианты использования, методы, подсистемы систем и т.д., И они обычно используются в сочетании с диаграммами взаимодействия (обычно диаграммами последовательности).

Диаграмма конечного автомата описывает все события (а также состояния и переходы для одного объекта)

Диаграмма последовательности описывает события для одного взаимодействия между всеми задействованными объектами

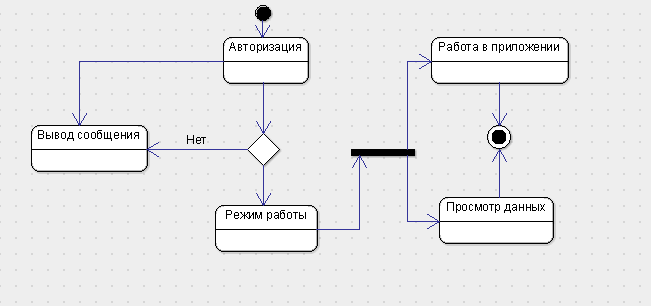


Рисунок 7 - Диаграмма состояний

Диаграмма классов

Структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними.

В программной инженерии диаграмма классов на Унифицированном языке моделирования (UML) представляет собой тип статической структурной диаграммы, которая описывает структуру системы, показывая классы системы, их атрибуты, операции (или методы) и отношения между объектами. Назначение диаграмм классов показывает статическую структуру классификаторов в системе.

Диаграмма предоставляет базовую нотацию для других структурных диаграмм, предписанных UML

Диаграмма классов UML состоит из набора классов и набора отношений между классами.

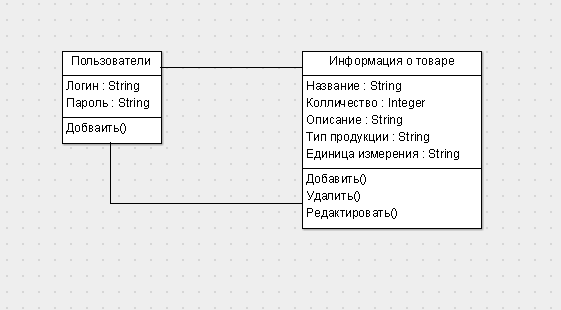


Рисунок 8 - Диаграмма классов

# 2.5 Общая характеристика организации решения задачи на ПК

Объектом автоматизации выбрано рабочее место для учета товаров для АЗС.

Для того, что создать автоматизированное рабочее место необходимо:

* Разработать формы
* Выбрать информационную технологию
* Создать базу данных

К входной информации относятся сведения о товарах на АЗС.

К выходной информации относятся накладная по товарам АЗС.

Главной целью является внедрение разработанного программного продукта для сотрудников АЗС.

Приложение позволяет:

* упростить работу сотрудника.
* сократить количество бумажных документов.
* снизить финансовые затраты на печать документов.
* снизить вероятность человеческой ошибки.
* формализация деятельности.

# 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА

# 3.1. Выбор средства разработки

Хранение данных в работе: база данных MySQL Workbench.

Распечатать можно посредством документа Word.

В программном средстве, благодаря среде разработки Microsoft Visual Studio 2019, был разработан удобный, понятный и простой в использовании интерфейс.

Программа реализована полностью в соответствии с поставленной задачей.

Проект реализован с помощью среды разработки Microsoft Visual Studio, с использованием языка C# и системы управления базами данных MySQL Workbench.

Данная среда разработки предназначена для написания, проведения тестов и запуска кода. Широкий набор различных утилит, включенных в Visual Studio, позволяет выполнять проектирование и разработку полноценных приложений с использованием графического пользовательского интерфейса без привлечения стороннего ПО. Помимо разработки программ для настольных и десктопных операционных систем, поддерживается разработка веб-приложений.

Без того немалый функционал можно расширить за счет использования сторонних плагинов. В последних версиях приложения большой упор сделан на расширение средств для коллективной разработки. Внесено много улучшений и дополнений для обеспечения эффективной командной работы над проектами любого масштаба.

В Visual Studio представлены как стандартные в подобном софте инструменты — отладчик, редактор кода, — так и собственные наработки. Кроме этого, доступны решения сторонних компаний, реализованные в виде подключаемых плагинов. Для повышения продуктивности используются большое количество автоматизированных возможностей: машинные подсказки, автоматическая очистка кода, рефакторинг. Функция IntelliSense позволяет работать с документацией к программе в интерактивном режиме, получая необходимую информацию непосредственно во время написания кода. Как и во всех продуктах от Microsoft, в приложении есть строка умного поиска, благодаря которой найти нужный пункт меню или описание функционала становится значительно легче.

Список основных возможностей и преимуществ выглядит следующим образом:

* собственный компилятор;
* возможность для развертывания приложения собственными средствами;
* редактор кода, включающий в себя инструменты авто завершения и форматирования;
* отладка кода.

Автоматизированные возможности рефакторинга кода, включающего поддержку различных языков, помогают упростить процесс написания кода. В целом, Microsoft Visual Studio является хорошим выбором как для начинающих, так и для разработчиков профессионального уровня.

MySQL Workbench — это унифицированный визуальный инструмент для архитекторов баз данных и разработчиков БД.

Основной единицей хранения данных является база данных, представляющая собой совокупность таблиц с типизированными столбцами.

MySQL Workbench предоставляет возможность моделирование данных, разработку SQL и комплексные инструменты администрирования для конфигурации сервера, администрирования пользователей, резервного копирования и многое другое. MySQL Workbench доступен на Windows, Linux и Mac OS X.

# 3.2. Разработка базы данных

Для того, чтобы разработать программный продукт я создал базы данных, функции храниения и заполнил таблицы (Рисунок 9, рисунок 10, рисунок 11 рисунок 12 , рисунок 13)

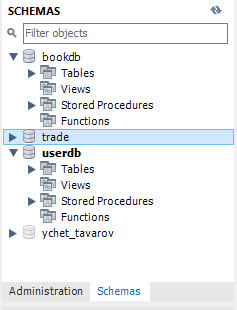


Рисунок 9 - Базы данных.

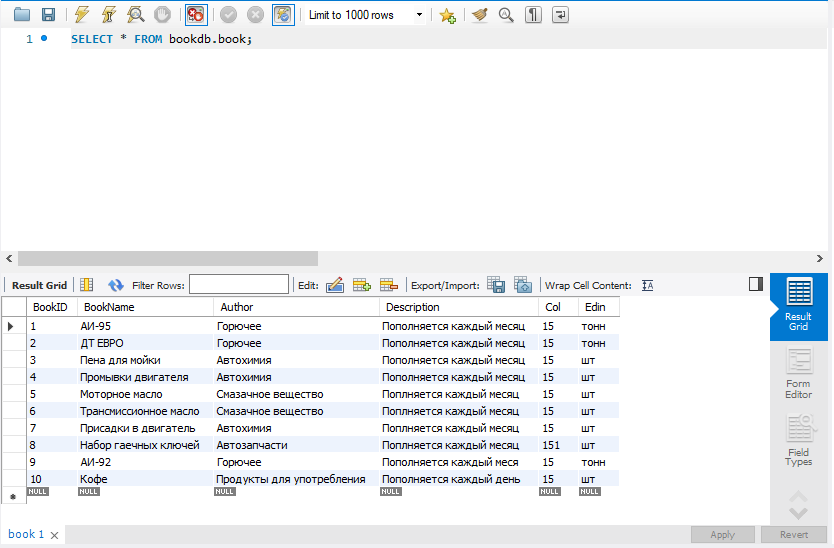


Рисунок 10 – Таблица «book»

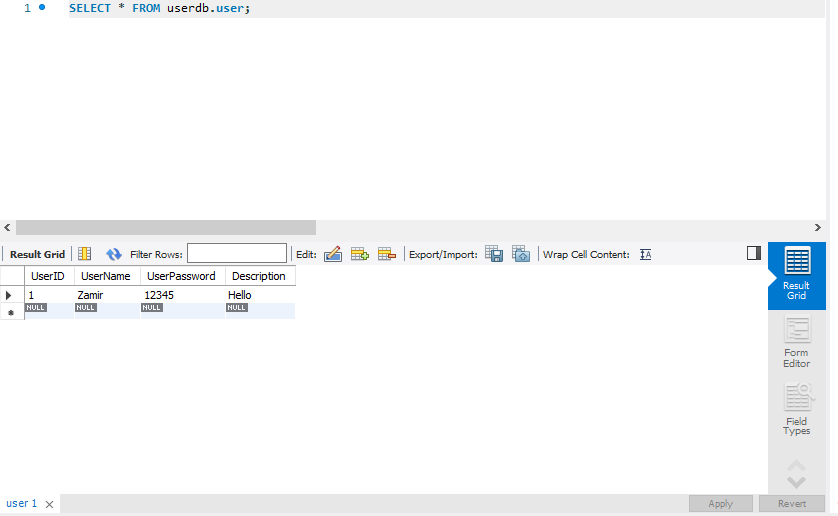


Рисунок 11- Таблица «user»

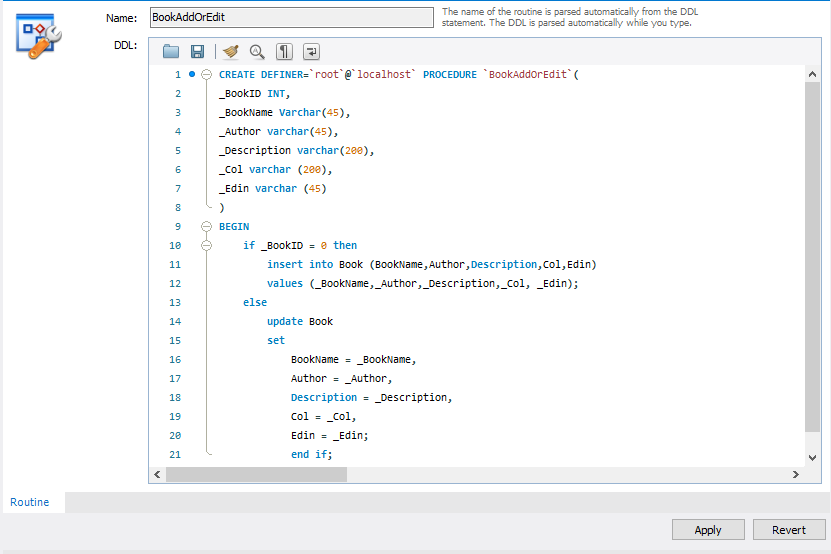


Рисунок 12 – Функция хранения «BookAddOrEdit»

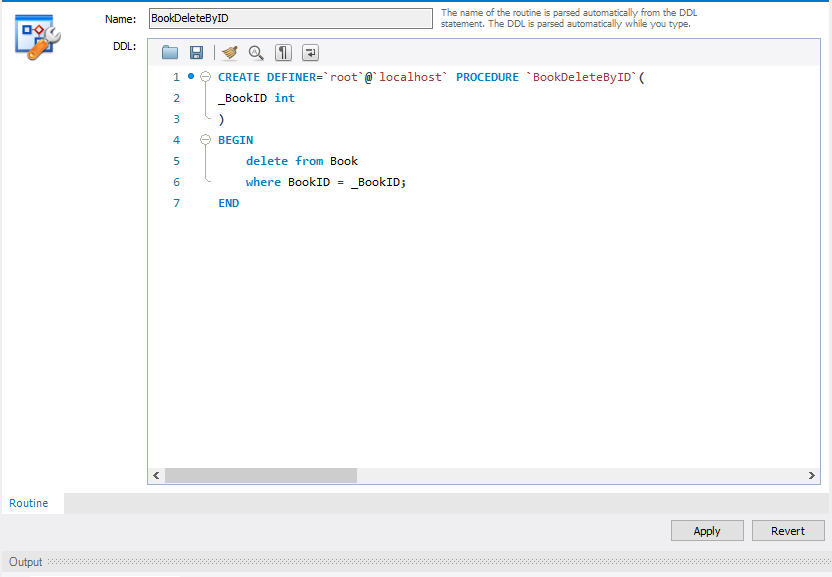


Рисунок 13 – Функция хранения «BookDeleteOrID»x`

**3.3 Разработка интерфейса**

В ходе разработки было принято решение использовать в системе графический интерфейс.

Графический интерфейс позволяет осуществлять взаимодействие человека с компьютером в форме диалога и с использованием окон, меню, и прочих элементов управления. Их относительно просто изучить и использовать. Пользователи, не имеющие опыта работы с компьютером, могут легко и быстро научиться работать с графическим интерфейсом.

С точки зрения удобства в программе важно создать такой пользовательский интерфейс, который сделает работу эффективной и производительной, а также обеспечит удовлетворенность пользователя от работы с программой.

В разрабатываемой информационной системе будет реализован интерфейс со свободной навигацией.

Интерфейс со свободной навигацией обеспечивает возможность осуществления любых допустимых в конкретном состоянии операций, доступ к которым возможен через различные интерфейсные компоненты. Предполагается, что взаимодействие с пользователем осуществляется посредством выбора и перемещения пиктограмм соответствующей объектно-ориентированной области.

Реализованный в информационной системе графический интерфейс включает в себя следующие формы:

* форма авторизации,
* форма регистрации,
* форма просмотра наличия товаров,
* форма редактирования товаров.

На Рис 14 изображена форма загрузки, которая запускает программу:



Рисунок 14 – Форма загрузки.

После формы загрузки появляется форма авторизация (Рис 15). После успешной аутентификации пользователь увидит форму пользователь.

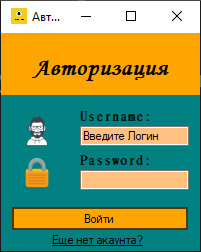


Рисунок 15 – Сообщение об ошибке

Если пользователя первый раз в приложении, то он должен зайти в форму регистрации (Рис 16) нажав на «Регистрация». После успешной регистрации пользователь увидит форму «Авторизации».

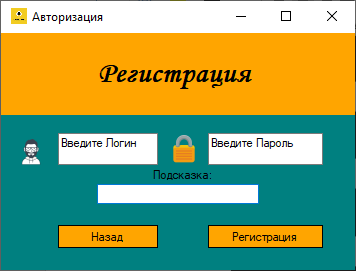


Рис 16 - Форма регистрация.

В форме Авторизация (Рис 15) нажав на ссылку «Еще нет аккаунта появиться форма регистрации для пользователя (Рис 16), где пользователь может зарегистрироваться для дальнейшей работы с приложением. После регистрации пользователю придется заново авторизоваться. Нажав на кнопку «Назад», пользователь увидит форму Авторизации (Рис 15).

После успешной авторизации пользователь попадет на форму учета товаров (Рис 17) Она предоставляет пользователю просмотр всех товаров, имеющихся в наличии. Нажав на любую кнопку, откроется соответствующая форма. Нажав на кнопку «Распечатать товары», пользователь увидит всплывающее окно, в котором будут выделенные материалы для печати. Нажав на кнопку «Добавить продукт», пользователь увидит форму добавления продукта (Рис 18), где он видит текущие данные и может как добавлять продукты, так и удалять.

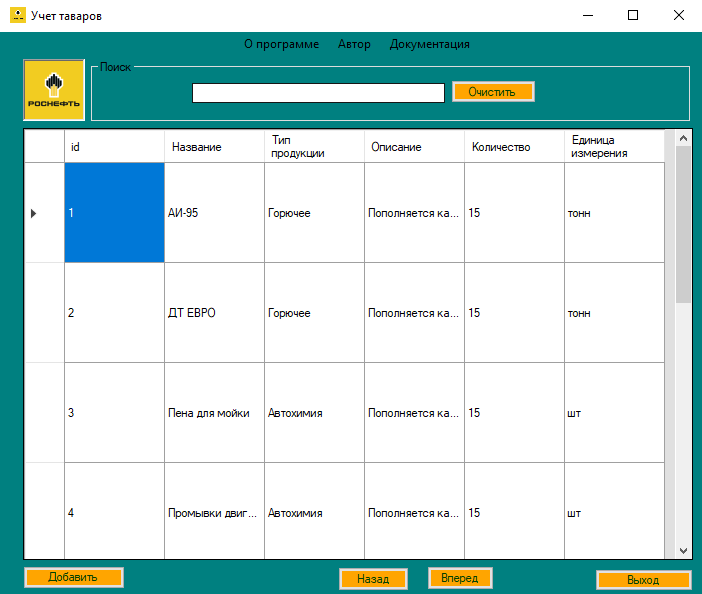


Рисунок 17 – Форма учета товаров.

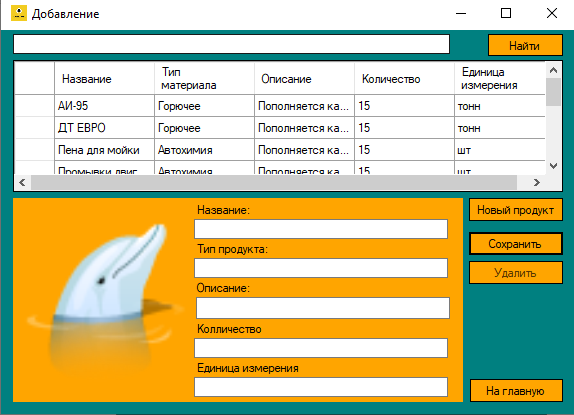


Рисунок 18 - Форма добавления.

После добавления, редактирования или удаления товаров, есть возможность импортировать ее в Excel. (Риc 19)

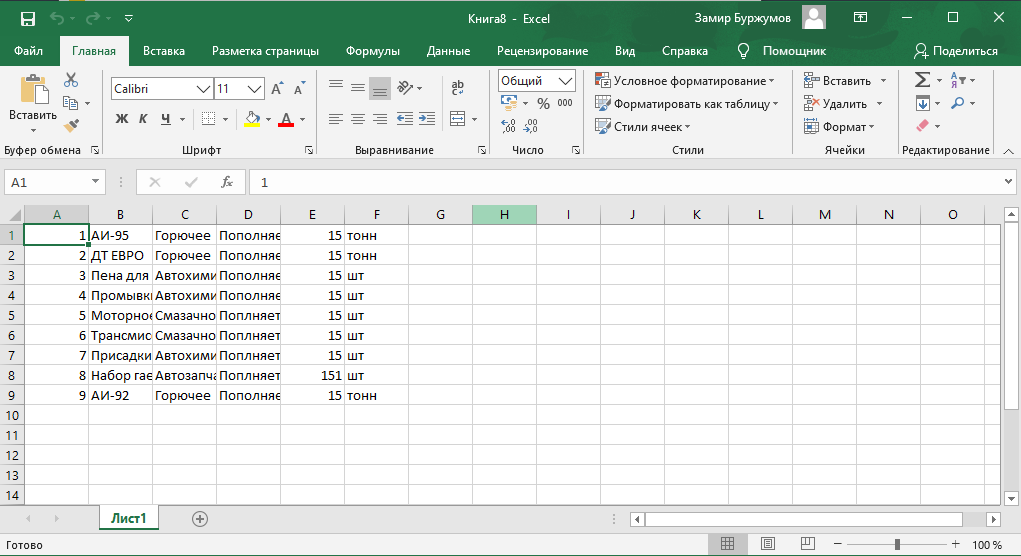


Рис 19 – Таблица “book” в Excel

Код всех форм представлен в приложении 3.

**4 ОБОСНОВАНИЕ ПРЕКТНЫХ РЕШЕНИЙ**

# 4.1 Обзор аналогов разрабатываемого программного обеспечения

Автоматизированное рабочее место - это элемент системы управления, оснащенный средствами, с помощью которых обеспечивается участие человека в реализации функций автоматизации.

Иными словами, это проблемно-ориентированный комплекс программных, технических и лингвистических средств. Он устанавливается непосредственно на рабочем месте пользователя и предназначен для автоматизации обработки операций при проектировании и решении необходимых задач.

Было время, когда термин “автоматизация” был синонимом передовых производственных предприятий, полных робототехники. Хотя замена человеческого труда машинным трудом является ярким примером автоматизации рабочего места, это далеко не единственный пример. Автоматизация присутствует в современных компаниях всех размеров – включая тонкие функции в общих программных приложениях и более очевидные реализации, такие как самоуправляемые транспортные средства или автономные роботы.

В настоящее время рынок систем автоматизации достаточно насыщен и на нем представлены специализированные решения для учета товаров, которые изначально разрабатывались с учетом российских законов. Ниже представлены системы автоматизации деятельности учета товаров:

* Битрикс24.
* AmoCRM.
* ПланФикс.
* Мегаплан.
* Простой бизнес.
* CloudShop.
* EKAM.
* 1С.
* Контур.Маркет.

# 4.2 Оценка эффективности внедряемого программного продукта

Внедренное АРМ является надежным, своевременно удовлетворяет информационные потребности пользователей , прост в обслуживании и имеет дружественный интерфейс.

Выполняемая сотрудником работа является эффективной. Все проводимые автоматизированные процессы не вызывают дискомфорта, который оказал бы воздействие на изначальные функциональные обязанности работника.

Так же соответствует таким требованиям как:

* быстрая обработка запросов;
* соответствие уровню знаний работника;
* понятный управляемый интерфейс;
* возможность получения новых знаний, улучшения их.

Соответствует таким фундаментальным принципам формирования автоматизированного рабочего места как:

* Устойчивость
* Системность
* Гибкость
* Эффективность

Автоматизированное рабочее место является индивидуальным, оно применимо к конкретному сотруднику и представляет собой место пользователя специалиста. Решает информационные и вычислительные задачи.

# ЗАКЛЮЧЕНИЕ

# В данной дипломной работе было описано приложение для автоматизации рабочего места сотрудника АЗС. В ходе работы были выполнены такие задачи как:

* Информационные- к таким задачам относятся кодирование, классификация, сбор, структурная организация, корректировка, хранение, поиск и выдача информации. Так же включают несложные вычислительные и логические процедуры арифметического и текстового характера и отношения .
* Вычислительные - эти задачи являются как формализуемыми, так и не полностью формализуемыми. Формализуемые задачи решаются на базе формальных алгоритмов и делятся на две группы: задачи прямого счета и задачи на основе математических моделей. Задачи прямого счета решаются с помощью простейших алгоритмов.

# Путем внедрения приложения была облегчена работа сотрудника. Приложение ориентировано на конечного пользователя, оно эргономично, то есть создает для пользователя комфортные условия труда и имеет дружественный и приятный интерфейс.

# В результате внедрения приложения была облегчена работа сотрудника.

# Система предусматривает дальнейшее совершенствование. Пути совершенствования и развития системы создание новых регистрирующих и отчетных модулей, модулей импорта и экспорта.

# Список информационных источников

1. Белов, В.В. Проектирование информационных систем: Учебник / В.В. Белов. - М.: Академия, 2018. - 144 c
2. Евдокимов П.В, C# на примерах. 4-е издание: Учебник / Евдокимов П.В. – М.:  Наука и Техника, 2019. - 320 с
3. Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова. - М.: Академия, 2018. - 272 c.
4. Петцольд Ч. Программирование для Microsoft Windows 8. 6-е изд. Питер: 2015. – 1008с.
5. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. – М: 2016. - 656с.
6. Троелсен Эндрю, Джепикс Филипп. Язык программирования C# 7 и платформы .NET и .NET Core. – Вильямс, 2018. – 1328с.
7. Турский В. Методология программирования. – М.: Мир, 2015. – 435с.
8. Фокс Дж. Программное обеспечение и его разработка. – М.: Мир, 2015.
9. Тестирование. Фундаментальная теория // Хабр [Электронный ресурс]. URL: https://habr.com/ru/post/279535/ (дата обращения: 05.05.2020)
10. Способы тестирования программного обеспечения// Хабр [Электронный ресурс]. URL: https://habr.com/ru/company/otus/blog/443418/ (дата обращения: 05.05.2020)
11. Полное руководство по языку программирования С# 8.0 и платформе .NET Core 3 // Metanit [Электронный ресурс]. URL: https://metanit.com/sharp/tutorial/ (дата обращения: 10.05.2020)
12. Техническое задание // Википедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Техническое\_задание (дата обращения: 10.05.2020)
13. Принципы составления технического задания [Электронный ресурс] Режим доступа: https://www.studsell.com/view/128118/
14. Виды обеспечения АИС [Электронный ресурс]. URL: http://studopedia.ru/14\_93295\_vidi-obespecheniya-ais.html (дата обращения 20.05.2020).
15. Проклашин, Д.Н. Автоматизация учета на предприятии Д.Н. Проклашин Автоматизация учета на предприятии . – Режим доступа: http://cyberleninka.ru/article/n/avtomatizatsiya-ucheta-na-predpriyatii.
16. Типы связей между таблицами (визуальные инструменты для баз данных) // – Режим доступа: https://technet.microsoft.com/ru-ru/library/ms190651(v=sql.105).aspx.
17. Сайт MySQL Режим доступа: https://www.mysql.com/, свободный. – Загл. с экрана. – Яз. рус.

# Приложение 1

**Техническое задание на разработку** **информационной системы   
учета товаров на языке программирования С# для   
 ИП Шитиков**

**Содержание**

1. Введение

1.1. Наименование программы

1.2. Назначение и область применения

2. Требования к программе

2.1. Требования к функциональным характеристикам

2.2. Требования к надежности

2.2.1. Требования к обеспечению надежного функционирования программы

2.2.2. Время восстановления после отказа

2.2.3. Отказы из-за некорректных действий пользователей системы

3. Условия эксплуатации

3.1. Климатические условия эксплуатации

3.2. Требования к квалификации и численности персонала

3.3. Требования к составу и параметрам технических средств

3.4. Требования к информационной и программной совместимости

3.4.1. Требования к исходным кодам и языкам программирования

3.4.2. Требования к программным средствам, используемым программой

3.4.3. Требования к защите информации и программ

3.5. Специальные требования

4. Требования к программной документации

4.1. Предварительный состав программной документации

5. Технико-экономические показатели

5.1. Экономические преимущества разработки

6. Стадии и этапы разработки

6.1. Стадии разработки

6.2. Этапы разработки

6.3. Содержание работ по этапам

7. Порядок контроля и приемки

7.1. Виды испытаний

7.2. Общие требования к приемке работы

# Приложение 2

Руководство пользователя по программному продукту «Учета товаров»

**1.Назначение программы.**

Программа предназначена для создания, управления содержимым базы данных, содержащим следующие данные:

* Пользователи
* Товары

Программа предоставляет интерфейс для управления содержимым базы данным в соответствии с предъявляемыми требованиями.

**2.Условия выполнения программы.**

Объем оперативного запоминающего устройства, свободного места на жестком диске 1 ГБ.

Программа должна работать на платформах Windows 7/8/10.

Программная документация поставляется в электронном и печатном виде.

Программа поставляется на лазерном носителе информации.

**3.Выполнение программы.**

Программа запускается через ярлык на рабочем столе.

# Приложение 3

Код формы авторизации:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

namespace Ychet\_Tavarov

{

public partial class Avtoriz : Form

{

public Avtoriz()

{

InitializeComponent();

txtLogin.Text = "Введите Логин";

txtLogin.ForeColor = Color.Black;

}

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

{

string login = txtLogin.Text;

string pass = txtPassword.Text;

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM userdb.user WHERE UserName = @ul AND UserPassword = @up", db.GetConnection());

command.Parameters.Add("@ul", MySqlDbType.VarChar).Value = login;

command.Parameters.Add("@up", MySqlDbType.VarChar).Value = pass;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

this.Hide();

ProductForm form = new ProductForm();

form.ShowDialog();

}

else

MessageBox.Show("Ошибка авторизации!!!");

}

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

{

this.Close();

}

Point lastpoint;

private void Avtoriz\_MouseMove(object sender, MouseEventArgs e)

{

if(e.Button == MouseButtons.Left)

{

this.Left += e.X - lastpoint.X;

this.Top += e.Y - lastpoint.Y;

}

}

private void Avtoriz\_MouseDown(object sender, MouseEventArgs e)

{

lastpoint = new Point(e.X, e.Y);

}

private void txtLogin\_Enter(object sender, EventArgs e)

{

if (txtLogin.Text == "Введите Логин")

{

txtLogin.Text = "";

txtLogin.ForeColor = Color.Black;

}

}

private void txtLogin\_Leave(object sender, EventArgs e)

{

if (txtLogin.Text == "")

{

txtLogin.Text = "Введите Логин";

txtLogin.ForeColor = Color.Black;

}

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

this.Hide();

Reg form = new Reg();

form.ShowDialog();

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

}

Код формы Регистрация:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Ychet\_Tavarov

{

public partial class Reg : Form

{

string connectionString = @"server=localhost;userid=root;password=710052473459;database=userdb;charset=utf8";

int UserID;

public Reg()

{

InitializeComponent();

textBox3.Text = "Введите Логин";

textBox3.ForeColor = Color.Black;

textBox4.Text = "Введите Пароль";

textBox4.ForeColor = Color.Black;

}

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

{

}

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

{

using (MySqlConnection mysqlCon = new MySqlConnection(connectionString))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("UserAddOrEdit", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_UserID", UserID);

mySqlCmd.Parameters.AddWithValue("\_UserName", textBox3.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_UserPassword", textBox4.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_Description", txtDescription.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Успешно");

}

}

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

{

this.Hide();

Avtoriz form = new Avtoriz();

form.ShowDialog();

}

private void textBox3\_Enter(object sender, EventArgs e)

{

if (textBox3.Text == "Введите Логин")

{

textBox3.Text = "";

textBox3.ForeColor = Color.Black;

}

}

private void textBox3\_Leave(object sender, EventArgs e)

{

if (textBox3.Text == "")

{

textBox3.Text = "Введите Логин";

textBox3.ForeColor = Color.Black;

}

}

private void textBox4\_Enter(object sender, EventArgs e)

{

if (textBox4.Text == "Введите Пароль")

{

textBox4.Text = "";

textBox4.ForeColor = Color.Black;

}

}

private void textBox4\_Leave(object sender, EventArgs e)

{

if (textBox4.Text == "")

{

textBox4.Text = "Введите Пароль";

textBox4.ForeColor = Color.Black;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

}

}

Код формы Добавления:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

namespace Ychet\_Tavarov

{

public partial class Add : Form

{

string connectionString = @"server=localhost;userid=root;password=710052473459;database=bookdb;charset=utf8";

int bookID;

private object dgvBook;

public Add()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

Clear();

GridFill();

dataGridView1.Columns[1].HeaderText = "Название";

dataGridView1.Columns[2].HeaderText = "Тип материала";

dataGridView1.Columns[3].HeaderText = "Описание";

dataGridView1.Columns[4].HeaderText = "Количество";

dataGridView1.Columns[5].HeaderText = "Единица измерения";

}

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

{

using(MySqlConnection mysqlCon = new MySqlConnection(connectionString))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("BookAddOrEdit", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_BookID", bookID);

mySqlCmd.Parameters.AddWithValue("\_BookName", txtBookName.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_Author", txtAuthor.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_Description", txtDescription.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_Col", txtCol.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_Edin", txtEdin.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Успешно");

Clear();

GridFill();

}

}

void GridFill()

{

using (MySqlConnection mysqlCon = new MySqlConnection(connectionString))

{

mysqlCon.Open();

MySqlDataAdapter sqlDa = new MySqlDataAdapter("BookViewAll", mysqlCon);

sqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

DataTable dtblBooK = new DataTable();

sqlDa.Fill(dtblBooK);

dataGridView1.DataSource = dtblBooK;

dataGridView1.Columns[0].Visible = false;

}

}

void Clear()

{

txtBookName.Text = txtAuthor.Text = txtDescription.Text = txtSearch.Text = "";

bookID = 0;

btnSave.Text = "Сохранить";

btnDelete.Enabled = false;

}

private void dataGridView1\_DoubleClick(object sender, EventArgs e)

{

if(dataGridView1.CurrentRow.Index != -1)

{

bookID = Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value.ToString());

btnDelete.Enabled = Enabled;

}

}

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

{

using (MySqlConnection mysqlCon = new MySqlConnection(connectionString))

{

mysqlCon.Open();

MySqlDataAdapter sqlDa = new MySqlDataAdapter("BookSearchByValue", mysqlCon);

sqlDa.SelectCommand.CommandType = CommandType.StoredProcedure;

sqlDa.SelectCommand.Parameters.AddWithValue("\_SearchValue", txtSearch.Text);

DataTable dtblBooK = new DataTable();

sqlDa.Fill(dtblBooK);

dataGridView1.DataSource = dtblBooK;

dataGridView1.Columns[0].Visible = false;

}

}

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

{

Clear();

}

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

{

using (MySqlConnection mysqlCon = new MySqlConnection(connectionString))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("BookDeleteByID", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_BookID", bookID);

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Успешно");

Clear();

GridFill();

}

}

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

{

this.Hide();

ProductForm form = new ProductForm ();

form.ShowDialog();

}

}

}

Код главной формы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data;

using MySql.Data.MySqlClient;

using Excel = Microsoft.Office.Interop.Excel;

namespace Ychet\_Tavarov

{

public partial class ProductForm : Form

{

Bitmap bitmap;

int scrollVal;

MySqlCommand materialCommand;

MySqlDataAdapter materialAdapter;

DataSet materialDataset;

DataTable materialTable;

public ProductForm()

{

InitializeComponent();

this.dataGridViewProduct.AllowUserToAddRows = false;

this.dataGridViewProduct.RowTemplate.Height = 100;

}

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

{

try

{

txtSearch.Text = "";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

MessageBox.Show("Параметры очищены");

}

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

{

DialogResult iExit;

try

{

iExit = MessageBox.Show("Вы уверены о прекращении работы?", "MySQl Connector", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (iExit == DialogResult.Yes)

{

this.Hide();

Avtoriz form = new Avtoriz();

form.ShowDialog();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

try

{

e.Graphics.DrawImage(bitmap, 0, 0);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Form1\_Load(object sender, EventArgs e)

{

string connectionString = @"server=localhost;userid=root;password=710052473459;database=bookdb;charset=utf8";

string sql = "SELECT \* FROM book";

MySqlConnection connection = new MySqlConnection(connectionString);

try

{

connection.Open();

materialCommand = new MySqlCommand(sql, connection);

materialAdapter = new MySqlDataAdapter(materialCommand);

materialDataset = new DataSet();

materialAdapter.Fill(materialDataset, "book");

materialTable = materialDataset.Tables["book"];

connection.Close();

//MessageBox.Show("Успешное соединение с базой данных!");

}

catch

{

MessageBox.Show("Нет соединения с базой данных!");

}

dataGridViewProduct.DataSource = materialTable;

dataGridViewProduct.Columns[0].HeaderText = "id";

dataGridViewProduct.Columns[1].HeaderText = "Название";

dataGridViewProduct.Columns[2].HeaderText = "Тип продукции";

dataGridViewProduct.Columns[3].HeaderText = "Описание";

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

dataGridViewProduct.Columns[5].HeaderText = "Единица измерения";

dataGridViewProduct.ReadOnly = true;

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

dataGridViewProduct.DataSource = materialTable;

}

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

{

DataView productDataView = new DataView(materialTable, "", "", DataViewRowState.CurrentRows);

productDataView.RowFilter = "(BookName like '%' + '" + txtSearch.Text + "' + '%') or (Description like '%' + '" + txtSearch.Text + "' + '%') ";

//productDataView.Sort = $"{comboBoxSort.Text}";

if (productDataView.Count == 0)

{

MessageBox.Show("Отсутствуют реультаты поиска");

}

dataGridViewProduct.DataSource = productDataView;

}

private Task ProcessData(List<string> list, IProgress<ProgressReport> progress)

{

int index = 1;

int totalProcess = list.Count;

var progressReport = new ProgressReport();

return Task.Run(() =>

{

for (int i = 0; i < totalProcess; i++)

{

progressReport.PercentComplete = index++ \* 100 / totalProcess;

progress.Report(progressReport);

//Thread.Sleep(10);

}

});

}

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

{

DataView productDataView = new DataView(materialTable, "", "", DataViewRowState.CurrentRows);

productDataView.RowFilter = "(Title like '%' + '" + txtSearch.Text + "' + '%') or (Description like '%' + '" + txtSearch.Text + "' + '%') ";

if (productDataView.Count == 0)

{

MessageBox.Show("Отсутствуют реультаты поиска");

}

dataGridViewProduct.DataSource = productDataView;

this.dataGridViewProduct.AllowUserToAddRows = false;

}

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

{

MessageBox.Show("Буржумов Замир Гаджибекович");

}

private void button1\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Add form = new Add();

form.ShowDialog();

}

public void ShowPage(int pageStart, int pageCount, string stream = " ")

{

if (stream == "+")

{

scrollVal = scrollVal + pageCount;

if (scrollVal > 20)

{

scrollVal = 20;

}

}

if (stream == "-")

{

scrollVal = scrollVal - pageCount;

if (scrollVal <= 0)

{

scrollVal = 0;

}

}

DataSet productSet = new DataSet();

materialAdapter.Fill(productSet, pageStart, pageCount, "book");

materialTable = productSet.Tables["book"];

materialTable.Columns[0].ColumnName = "BookID";

materialTable.Columns[1].ColumnName = "BookName";

materialTable.Columns[2].ColumnName = "Author";

materialTable.Columns[3].ColumnName = "Description";

materialTable.Columns[4].ColumnName = "Col";

// materialTable.Columns[5].ColumnName = "MaterialTypeId";

dataGridViewProduct.DataSource = materialTable;

}

private void button3\_Click\_1(object sender, EventArgs e)

{

ShowPage(scrollVal, 4, "-");

}

private void button4\_Click\_1(object sender, EventArgs e)

{

ShowPage(scrollVal, 4, "+");

}

private void dataGridViewProduct\_CellMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)

{

this.Hide();

Add form = new Add();

form.ShowDialog();

}

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

{

MessageBox.Show("Компания реализует автомобильный бензин и дизельные топлива крупным и мелким оптом. Крупнооптовые поставки осуществляются с возможностью отгрузки железнодорожным транспортом. Заявки на приобретение любых видов автомобильных бензинов и дизельного топлива принимаются через Cлужбу единого окна. Срок рассмотрения заявки составляет 10 календарных дней. В течение этого срока с Вами свяжутся сотрудники Компании.");

}

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

{

MessageBox.Show("Организации Группы «ЛУКОЙЛ» осуществляют реализацию моторных топлив мелким оптом с возможностью отгрузки автомобильным транспортом с нефтебаз в регионах Европейской части России и на Урале. Полную информацию и контакты для закупки топлива можно получить на сайте организации Группы, осуществляющей реализацию продукции в конкретном регионе. Для перехода на сайт воспользуйтесь картой регионов сбыта.");

}

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

{

MessageBox.Show("ЛУКОЙЛ при оптовой реализации автомобильных бензинов и дизельного топлива руководствуется следующими основными принципами: √приоритетность поставок на внутренний рынок РФ; √обеспечение недискриминационных условий покупателям; √информационная открытость организаций Группы «ЛУКОЙЛ». " +

"Эти принципы закреплены в торговых политиках Группы «ЛУКОЙЛ» в области оптовой реализации автомобильных бензинов и дизельного топлива в Российской Федерации, которые согласованы с Федеральной антимонопольной службой.");

}

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

{

try

{

int height = dataGridViewProduct.Height;

dataGridViewProduct.Height = dataGridViewProduct.RowCount \* dataGridViewProduct.RowTemplate.Height \* 2;

bitmap = new Bitmap(dataGridViewProduct.Width, dataGridViewProduct.Height);

dataGridViewProduct.DrawToBitmap(bitmap, new Rectangle(0, 0, dataGridViewProduct.Width, dataGridViewProduct.Height));

printPreviewDialog1.PrintPreviewControl.Zoom = 1;

printPreviewDialog1.ShowDialog();

dataGridViewProduct.Height = height;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

Excel.Application exApp = new Excel.Application();

exApp.Workbooks.Add();

Excel.Worksheet wsh = (Excel.Worksheet)exApp.ActiveSheet;

int i, j;

for (i = 0; i<= dataGridViewProduct.RowCount-2; i++)

{

for (j=0; j<= dataGridViewProduct.ColumnCount-1; j++)

{

wsh.Cells[i + 1, j + 1] = dataGridViewProduct[j, i].Value.ToString();

}

}

exApp.Visible = true;

}

}

}