

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ / Г.И. Радченко  
« \_\_\_ » \_\_\_\_\_ 2020 г.

Десктопное приложение «Поиск материалов  
и покупных комплектующих изделий по цене реализации в среде Интернет»

### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,  
к.т.н., доцент каф. ЭВМ  
\_\_\_\_\_ / В.А. Парасич  
« \_\_\_ » \_\_\_\_\_ 2020 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ / Н.М. Милёшин  
« \_\_\_ » \_\_\_\_\_ 2020 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ / С.В. Сяськов  
« \_\_\_ » \_\_\_\_\_ 2020 г.

Челябинск 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ / Г.И. Радченко

«\_\_\_» \_\_\_\_\_ 2020 г.

## **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Милёшину Никите Михайловичу  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

**1. Тема работы:** Десктопное приложение «Поиск материалов и покупных комплектующих изделий по цене реализации в среде Интернет». Утверждена приказом по университету от 24.04.2020. № 627.

**2. Срок сдачи студентом законченной работы:** 1 июня 2020 г.

**3. Исходные данные к работе:**

Задачей является разработка десктопного приложения для сбора и сортировки информации по закупочным материалам с целью автоматизировать рутинные задачи по приобретению товаров сотрудниками предприятия.

**4. Перечень подлежащих разработке вопросов:**

- сбор и обработка информации, в том числе сбор данных по ключевым словам, обработка по заданным пользователем критериям, таким как цена, поставщик, срок поставки;

- простой и удобный интерфейс, понятный пользователю любого уровня технической подготовки;

- возможность импорта и экспорта полученных данных в Базу Данных и дальнейшая работа с Базой. Версия MySQL Server 8.0.20.

**5. Дата выдачи задания:** 1 декабря 2019 г.

Руководитель работы \_\_\_\_\_ / В.А. Парасич  
Студент \_\_\_\_\_ / Н.М.Милёшин

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2020	
Разработка единой концепции системы	01.04.2020	
Реализация системы	01.05.2020	
Тестирование, отладка, эксперименты	15.05.2020	
Компоновка текста работы и сдача на нормоконтроль	24.05.2020	
Подготовка презентации и доклада	30.05.2020	

Руководитель работы \_\_\_\_\_ / В.А. Парасич  
Студент \_\_\_\_\_ / Н.М. Милёшин

## АННОТАЦИЯ

Милёшин Н.М. Десктопное приложение «Поиск материалов и покупных комплектующих изделий по цене реализации в среде Интернет». – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2020, 63 с., 11 ил., библиогр. список – 14 наим.

В рамках выпускной квалификационной работы производится разработка надёжного и быстрого программного обеспечения, с помощью которого можно в любой момент времени по запросу пользователя предоставлять необходимый перечень материалов и покупных комплектующих изделий с возможностью фильтрации по указанным пользователем критериям или искать аналоги при отсутствии определенного вида товаров. При разработке данного приложения были использованы такие технологии как, язык программирования C#, для интерфейса использовались библиотеки Windows Forms, Selenium, HTMLAgilityPack, также применялись регулярные выражения в качестве фильтра для парсинга необходимой информации, а для работы с выгруженной информацией используется MySQL Server. По результатам поисковой выдачи производится выборка и анализ результатов, удаляется ненужная информация.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	8
1.1 Постановка задачи.....	8
1.2 Обзор аналогов.....	8
1.3 Недостаток существующих решений.....	9
1.4 Анализ и выбор технологических решений.....	10
1.5 Вывод.....	13
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ.....	13
2.1 Функциональные требования.....	14
3 ПРОЕКТИРОВАНИЕ.....	15
3.1 Архитектура предлагаемого проекта.....	15
3.2 Интерфейс.....	16
3.3 Модуль запроса и отбора.....	17
4 РЕАЛИЗАЦИЯ.....	17
4.1 Реализация взаимодействия с базой данных.....	17
4.1.1 Реализация подключения.....	18
4.1.2 Реализация экспорта и импорта.....	18
4.1.3 Реализация вставки и удаления.....	19
4.2 Реализация получение информации с интернет страницы.....	20
4.3 Реализация сортировки, класс Compare.cs.....	21
5 ТЕСТИРОВАНИЕ.....	22
5.1 Системные требования.....	22
5.2 Работа с приложением.....	22
ЗАКЛЮЧЕНИЕ.....	28
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	29
ПРИЛОЖЕНИЕ А.....	31
Листинг А.1- Реализация Form.cs(Вход в базу данных).....	31
Листинг А.2 - Файл с реализацией Form2.cs (Поиск материалов).....	37
Листинг А.3 - Реализация файла Comparer.cs.....	49
Листинг А.4 - Реализация файла ParsItem.cs.....	51
Листинг А.5 - Реализация файла MySQL.cs.....	56

## **ВВЕДЕНИЕ**

В наши дни практически все аспекты маркетинговой деятельности любого предприятия связаны с использованием Интернета. Традиционные инструменты продвижения и рекламы, которые используются в сети, с каждым годом обновляют свой функционал, становятся более эффективными, а Интернет предоставляет предприятиям новые возможности для взаимодействия со своими партнерами, поиска необходимой информации, электронной торговли. В настоящее время работники предприятия для поиска необходимых комплектующих пользуются устаревшими методами ручного поиска. Разрабатываемое в рамках данной выпускной квалификационной работы решение предназначено для автоматизации этих рутинных задач и более оптимального использования рабочего времени сотрудников.

## **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

На данный момент во всех предприятиях используют Интернет с целью решения возникших проблем, а также для улучшения производства или использования в качестве маркетингового инструмента. Важную роль играет гигантский объём информации, который надо как-то упорядочить и использовать в своих целях. В нынешнее время для пользователя нужно тратить колоссальное количество времени для сортировки нужной информации.

### **1.1 Постановка задачи**

Задача выпускной квалификационной работы — разработать десктопное приложение для сбора и сортировки информации по закупочным материалам с целью автоматизировать рутинные задачи по приобретению товаров службами предприятия.

### **1.2 Обзор аналогов**

В настоящее время в открытом доступе в сети Интернет находится большое количество парсеров — программ, предназначенных для поиска и сортировки нужной информации, однако, каждый из них выполняет задачу для конкретного предприятия и не может быть приспособлен под критерии поиска, которые были обозначены в данной выпускной квалификационной работе. Несмотря на невозможность использования для заданных задач уже готовых решений, перед началом проектирования был проведен анализ и выявлены следующие аналогичные программные продукты.

*ParseHub.*

ParseHub - зарубежная программа для парсинга сайтов и интернет-магазинов, которая предназначена для сканирования одного или нескольких

веб-сайтов. Эта программа предоставляет возможность собирать и хранить данные с любой страницы JavaScript и AJAX. У данного продукта присутствует платная и бесплатная версия. Приложение использует технологию самообучения и способно распознать самые сложные документы в сети, затем генерирует выходной файл в том формате, который нужен пользователю.[1]

### *Import.io*

Import.io - предлагает разработчику легко формировать собственные пакеты данных: нужно только импортировать информацию с определенной веб-страницы и экспортировать. Также предлагает извлекать тысячи веб-страниц за считанные минуты, не написав ни строчки кода, и создавать тысячи API согласно заданным требованиям.[2]

### *Octoparse.*

Octoparse – это платформа для парсинга. Она используется для сбора данных с нужных сайтов и превращать полученную неструктурированную или частично структурированную информацию в упорядоченный набор данных без использования программирования. Платформа предоставляет готовые шаблоны для сбора, включая eBay, Twitter, BestBuy и многие другие.[3]

## **1.3 Недостатки существующих решений**

Каждый из аналогов имеет ряд преимуществ, однако, среди всех имеющихся решений существуют следующие недостатки:

- 1) Главный недостаток всех представленных парсеров - их цена, так как большинство готовых решений были разработаны с коммерческими целями, по этой причине они либо предоставляет усеченный функционал в бесплатной версии, либо не предоставляют его без оплаты вовсе;
- 2) Отсутствие локализации. Работа с интерфейсом на английском языке вызывает дополнительные трудности для тех сотрудников предприятия,

которые не обладают навыками владения иностранным языком. Это приводит к дополнительным задержкам и трате рабочего времени на изучение доступного функционала и выполнение поставленных задач;

3) Невозможность модифицирования готовых решений. Для каждого конкретного предприятия существуют свои уникальные задачи, которые требуют поиска материалов, которые используются исключительно в данной организации. Т.к. готовые решения не предоставляют возможности вносить изменения в свой исходный код или добавлять новые модули, использовать данные инструменты для решения своих задач либо проблематично, либо невозможно.

#### **1.4 Анализ и выбор технологий**

При разработке проекта использовался ряд технологий, среди которых язык программирования C#, Windows Forms в качестве элементов интерфейса, а также библиотеки Selenium, HTMLAgilityPack, для сортировки и фильтрации информации применялись регулярные выражения.

##### *Язык программирования C#.*

C# является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все чаще основана на создании программных компонентов в форме автономных и самостоятельных пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов — это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, предоставляющие декларативные сведения о компоненте, а также встроенные элементы документации. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию

работы. Благодаря этому C# отлично подходит для создания и разработки десктопных приложений.[4][5]

#### *Windows Forms.*

Windows Forms позволяют создать простые графические приложения с окнами, формами и подобными вещами, также позволяет разрабатывать интеллектуальные клиенты. Интеллектуальный клиент — это приложение с полнофункциональным графическим интерфейсом, простое в развертывании и обновлении, способное работать при наличии или отсутствии подключения к Интернету и использующее более безопасный доступ к ресурсам на локальном компьютере по сравнению с традиционными приложениями Windows.[6]

#### *MySQL Server.*

MySQL — свободная система управления базами данных(СУБД). Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, также в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Версия программы 8.0.20 не ниже.[7][8]

#### *MySql.Data.MySqlClient.cs*

MySql.Data.MySqlClient.cs - данная библиотека предоставляет работу с базой данных, а точнее позволяет подключиться непосредственно к самой базе и позволяет работать с таблицей, например записывать, удалять или сортировать существующие атрибуты базы данных. Также позволит осуществить выполнение запросов. [9][10]

#### *System.NET.*

*System.NET* представляет собой .NET библиотеку которая предоставляет простой программный интерфейс, для многих протоколов, используемых в современных сетях, также позволят разрабатывать приложения использующие

ресурсы Интернета не заботясь о конкретных особенностях отдельных протоколов. Она позволит выгрузить всю HTML страницу на компьютер и это позволяет ускорить процесс поиска нужной информации. Также позволяет выполнить типичные задачи, как например, вытащить ссылки со страницы.[11]

### *System.Text.RegularExpressions*

*System.Text.RegularExpressions* – данная библиотека предоставляет эффективный и гибкий метод по обработке больших текстов, она позволяет использовать регулярные выражения, что существенно уменьшит объем кода по сравнению с использованием стандартных операций со строками. [12]

### *Регулярные выражения.*

Регулярные выражения — формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. Для поиска используется строка-образец, её часто называют «шаблоном», «маской», состоящая из символов и метасимволов и задающая правило поиска. Таким образом, с помощью регулярных выражений мы можем задавать область поиска нужной информации в рамках заданных условий фильтрации.[13]

Присутствуют также технологии как CsQuery, AngleSharp, Regex, Benchmark, но у них есть ряд своих недостатков, а некоторые предназначены для решения других задач. Например CsQuery проигрывает другим по скорости обработки, AngleSharp относительно недавно появился и у него есть недоработки и малая аудитория, поэтому литературу или хорошие примеры как с ним взаимодействовать найти сложно.

По этой причине, выбор регулярных выражений в качестве основной технологии является самым оптимальным вариантом из всех доступных.

## **1.5 Вывод**

В рамках данного раздела был проведен анализ предметной области, анализ существующих технологических решений, а также разобраны их недостатки. Выявлены подходящие технологии для разработки программного продукта и сформулированы критерии, по которым выбиралась программная среда разработки ПО.

## **2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ**

При постановке задачи, проектировании технического задания и разборе существующих аналогов был выявлен и определен ряд требований, которым должен соответствовать разрабатываемый программный продукт. Они разделяются на функциональные и требования к аппаратной части системы.

### **2.1 Функциональные требования**

Одним из самых важных моментов при постановке задачи является определение функциональных требований, т. е. функций, необходимых пользователю, которые нужно реализовать для достижения цели, а также программных решений, которые стоит учесть при проектировании для оптимизации затрат времени на разработку.

Итак, основными функциональными требованиями являются:

- сбор и обработка информации, в том числе сбор данных по ключевым словам, обработка по заданным пользователем критериям, таким как цена, поставщик, срок поставки;
- простой и удобный интерфейс, понятный пользователю любого уровня технической подготовки;
- возможность импорта и экспорта данных в Базу Данных и дальнейшая работа с ними;

- возможность импорта полученных данных в ПО Excel, для сохранения определенной выборки материалов и дальнейшего решения по закупке найденных комплектующих изделий.

Для наглядного обозначения можно использовать диаграммы вариантов использования бизнес процессов, которые изображены на рисунках 1 и 2.



Рисунок 1 – Существующие бизнес процессы

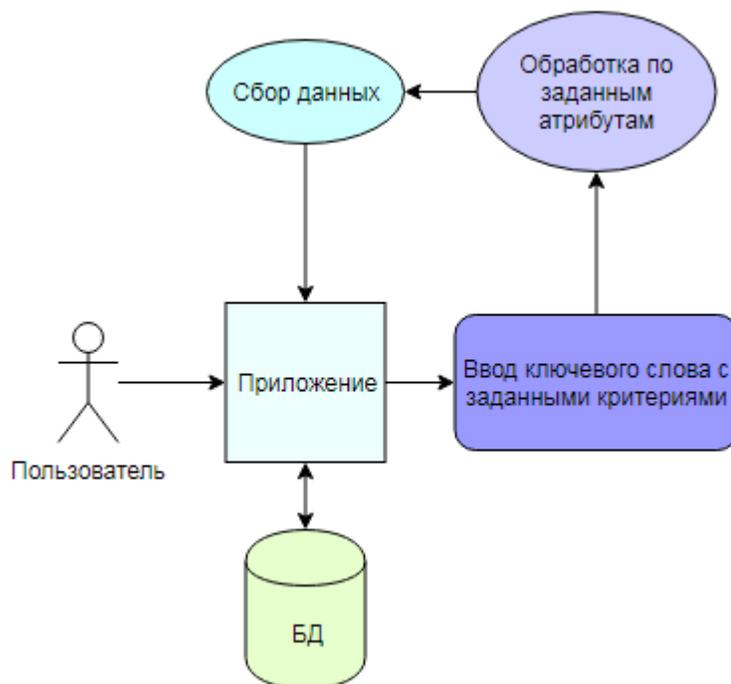


Рисунок 2 – Бизнес процессы разрабатываемого проекта

### 3. ПРОЕКТИРОВАНИЕ

Разрабатываемый проект логически можно разделить на следующие модули:

- модуль запроса;
- модуль интерфейса;
- модуль обработки.

#### 3.1 Архитектура предлагаемого решения

Структура программного обеспечения, разрабатываемого в данной выпускной квалификационной работе схематично изображена на рисунке 3.

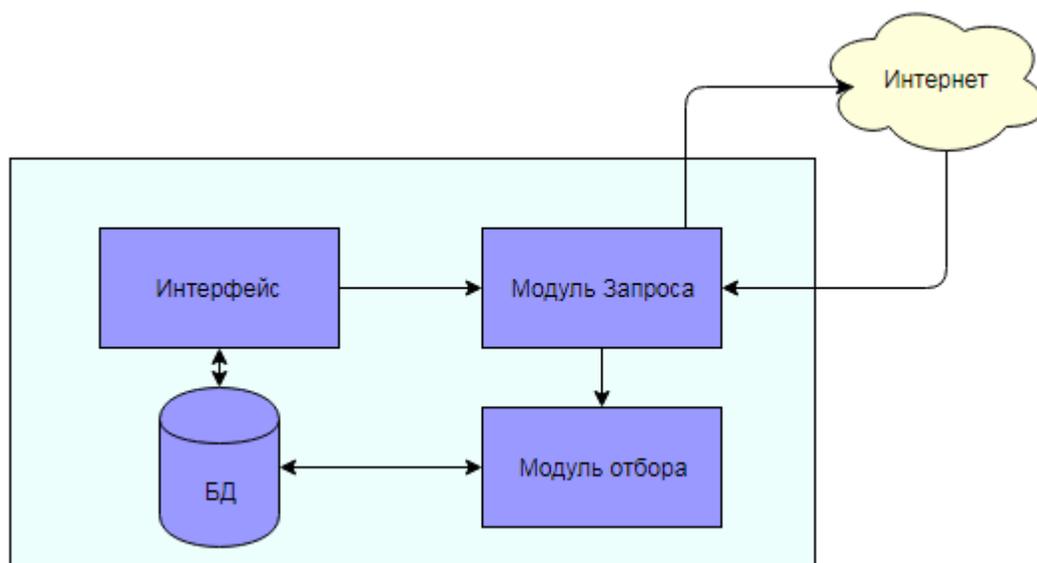


Рисунок 3 – Архитектура предлагаемого решения

Основной блок программы содержит 2 модуля: модуль поискового запроса и модуль отбора информации.

### 3.2 Интерфейс

Данный модуль предоставляет пользователю существующий инструментал, он необходим для того чтобы человек мог управлять программным обеспечением. Например, ввод названия нужного комплектующего изделия, ввод допустимой цены или наличие комплектующего изделия в продаже.

В данный интерфейс также будет входить следующий функционал:

- отображение собранной информации в таблице;
- удаление или поиск нужной строки;
- сортировка по цене или другим доступным критериям;
- вывод сообщения в случае неправильного ввода пароля или в случае отсутствия подключения к БД.

### 3.3 Модуль запроса и модуль отбора

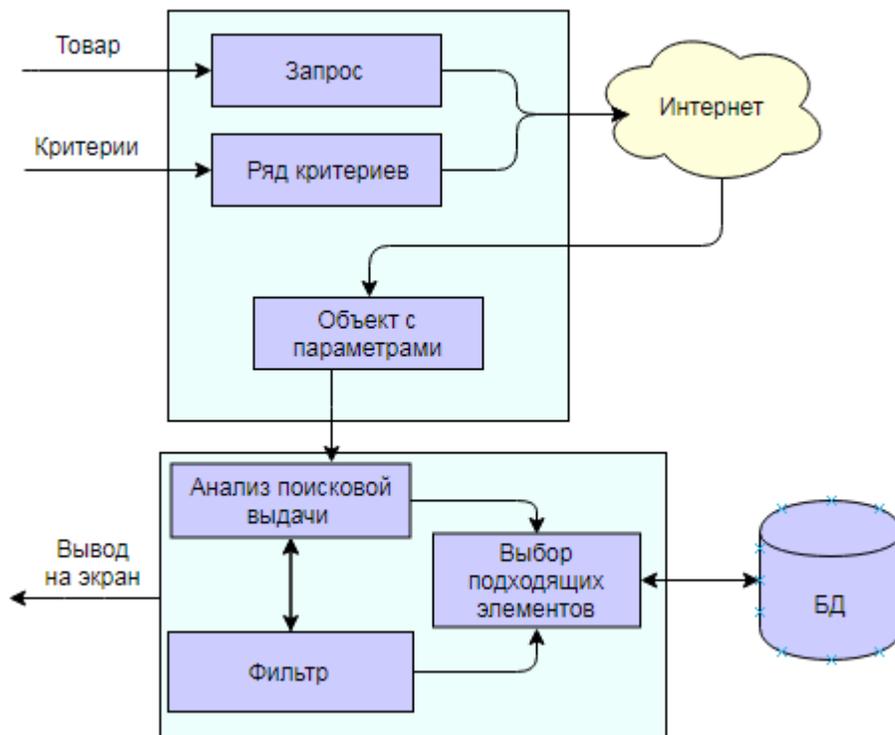


Рисунок 4 – Архитектура модуля разработки

Модуль запроса представляет собой запрос в интернет с нужным названием товара и критериями, далее происходит поиск по соответствующим атрибутам и все найденные товары записывается в таблицу.

Модуль отбора сопоставляет все критерии объекта по соответствующим столбцам. Далее фильтр поможет отобразить товар только по одному или нескольким необходимым критериям, например, отобразить только по критерию “присутствует доставка”.

## 4 РЕАЛИЗАЦИЯ

### 4.1 Реализация взаимодействия с базой данных (MySQL.cs)

Одним из основных компонентов приложения является база данных. Она предназначена для хранения найденных результатов поиска, а также содержит

инструменты по работе с ними, включающие в себя подключение базы данных, экспорт и импорт, вставку и удаление информации (элементов). Пример приведён в приложение А - листинг А.5

#### **4.1.1 Реализация подключения**

Для подключения к базе данных, пользователю в обязательном порядке нужно ввести данные для авторизации и дальнейшего использования приложения. Само подключение реализовано в классе «MySQL.cs», следующими методами :

- `GetDBConnection`, он использует параметры, адрес сети, в котором находится база данных, порт, имя схемы - название базы данных, а также данные пользователя;

- `Close` закрывает соединение, также закрывает все активные объекты, связанные с соединением, метод `IsOpen` возвращает состояние подключения БД;

- `CheckExist` нужен для проверки существует ли таблица, в противном случае создания таблицы с префиксом, который задаст пользователь. Префикс нужен для именованного определенного товара, например именуем 'Wire', и в БД будет храниться таблица `Product_Wire` к которой обратиться;

- `ReadStart` начало потока чтения из таблицы которую выберет пользователь;

- `ReadRow` считывает все существующие строки с таблицы;

- `ReadEnd` остановка чтения.

#### **4.1.2 Реализация экспорта и импорта**

При работе с базой данных иногда возникает потребность в экспорте и импорте всех данных. Для более удобного представления информации и взаимодействия с ней в приложении используется Excel таблица.

Импорт и экспорт БД реализован таким образом, что при нажатии на соответствующую кнопку будет появляться диалоговое окно, где достаточно выбрать место и написать наименование файла. Реализация была осуществлена с помощью существующего функционала в библиотеке «MySql.Data.MySqlClient». Создается объект типа MySqlCommand данный объект управляет запросами SQL, далее создается объект MySqlBackup на основе MySqlCommand, указывая открытую БД, далее вводится название файла.

Экспорт в Excel осуществляется так, что запускается процесс Excel создаются нужные по ширине колонки, задается название колонок в соответствии с типом объекта, затем считываем и заполняем Excel-таблицу.

#### **4.1.3 Реализация вставки и удаления**

После запроса необходимой информации, пользователю требуется сохранить перечень товаров для последующей работы с ними. Особенную сложность в реализации данного действия составляет добавление новой записи с первичным ключом, не совпадающим с тем, что уже существует. Именно с этой целью при запуске базы данных мы получаем максимальный ID и увеличиваем его при добавлении новой записи.

Реализация вставки начинается с увеличением максимального ID затем конвертирования цены в decimal, так как в БД тип данных decimal. Далее используя библиотеку «MySql.Data.MySqlClient.cs», создаем SQL запрос на вставку, и указываем, в какую БД и таблицу происходит вставка.

Реализация удаления осуществлена по ID, создается SQL запрос «"DELETE FROM product" + prefix + " WHERE ID = " + id» далее обращение к БД и выполнение запроса. Пример приведён в приложение А - листинг А.5

## **4.2 Реализация получение информации с интернет страницы, класс ParsItem.cs.**

Для реализации парсинга (сканирования интернет страницы) обычно первым шагом загружают исходный код данного сайта, который включает в себя HTML разметку. С этой разметкой удобно работать для получения и выделения в ней необходимой информации.

В качестве тестовой страницы в приложении используется сайт “Яндекс маркет”[14]. На данном сайте вся информация о товарах содержится в индивидуальных блоках, каждый из которых содержит данные этих товаров: цену, рейтинг, название и др.

В данном классе также был создан класс “LinkInfo”, он был создан для получения ссылок по ID, в нем присутствуют два метода на возвращение ID и ссылки.

Сам класс “ParsItem.cs” имеет динамический само расширяющийся массив объектов “LinkInfo”, такой подход дает возможность добавлять объекты столько, сколько необходимо. Были реализованы такие простые методы, как “Savelink”, “getLink”, один метод заполняет список, другой возвращает ссылку в соответствии с “ID”.

Загрузка “HTML” разметки реализована путем указания протокола безопасности, обращение к нужной URL, методом “LoadPage”.

Для получения данного блока информации необходимо выделить раздел с товарами, для этого воспользуемся стандартной функцией IndexOf, которая получает номер символа первого вхождения заданной строки. Блок товара имеет подзаголовок n-snippet-card2.

Реализация поиска всех критериев реализована поиском первой карточки товара, которая в HTML разметке начинается с «n-snippet-card2 », данная задумка позволяет найти начало карточки товара, далее таким же методом я

нахожу вторую карточку товара, и считываю, сколько символов между ними, то есть разницу второго и первого товара, и это разницу записываю.

Следующим шагом идет обработка записанного фрагмента товара, ищется наименование следующим регулярным выражением «<a class=\"link+[^<>]+title=\"+[^<>]+», оно позволяет найти кусок, где находится наименование товара. Далее с помощью IndexOf определяется, с какого символа находится начало наименования товара, путем поиска «title=\", далее требуется найти конец наименования товара тем же методом, а конец названия товара закрывается символом \"

Получение цены происходит точно таким же образом, получение кусочка, где находится цена с помощью данного регулярного выражения «<div class=\"price\">+[^<>]+[0-9]+», далее нахождение индекса начала цены и конца. Также присутствует проверка по цене, если ценовой диапазон меньше или больше заданного, товар будет отбрасываться.

Получение поля «Есть доставка», реализована тем же алгоритмом поиска нужного кусочка:

- «Есть+\\s+доставка» данное регулярное выражение помогает найти, что у товара есть доставка;

- «,+[\s0-9-]+д+[а-я]+</span>» данное регулярное выражение помогает найти, что у товара есть доставка и примерное время по дням. Пример приведён в приложение А - листинг А.4

### **4.3 Реализация сортировки, класс Compare.cs.**

Сортировка реализовано с помощью интерфейса «IComparer», где сравниваются два объекта, также используется «ListView.ListViewItemSorter» данное свойство возвращает или задает блок сравнения сортировки для данного

элемента управления, то есть при нажатии на определенную колонку будет вызываться сортировка.

Для управления данной сортировкой были созданы три переменные, это “order”, “columnIndex”, “number”. Все они нужны для:

- “order” порядок сортировки, от меньшего к большему или наоборот;
- “columnIndex” индекс колонки, то есть какую колонку мы сортируем;
- “number” чтобы сравнивать строку, как число. Приложение А - листинг

А.3.

## **5 ТЕСТИРОВАНИЕ**

### **5.1 Системные требования**

Для установки и запуска разрабатываемого программного обеспечения необходимо обеспечить наличие следующих программных модулей и драйверов:

- сервер базы данных MySQL версии не ниже 8.0.2;
- Net.Framework 4.6;
- операционная система Windows 7 или Windows 10.

### **5.2 Работа с приложением**

Для запуска приложения необходимо запустить файл *Searcher.exe*, который находится в корневом каталоге.

При открытии программы показывается приветственное окно с формой входа в базу данных, включающая в себя основные поля заполнения данных пользователя и кнопка входа. Реализация формы «Вход в Базу Данных» в приложение А – листинг А.1

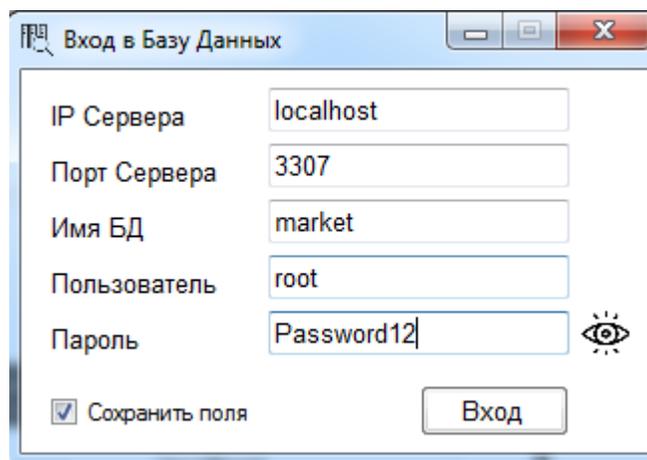


Рисунок 5 – Вход в приложение

После введения всех данных открывается окно “Поиск материалов”, также отображает пользователя, и в какой таблице находится сам пользователь. Оно содержит несколько вкладок:

- общее;
- список;
- поиск.

По умолчанию активной при запуске является вкладка “Общее”, она содержит такой функционал как:

- импорт и экспорт БД;
- экспорт в Excel
- создание новой таблицы для соответствующего товара;
- выбор таблицы, удаление таблицы.

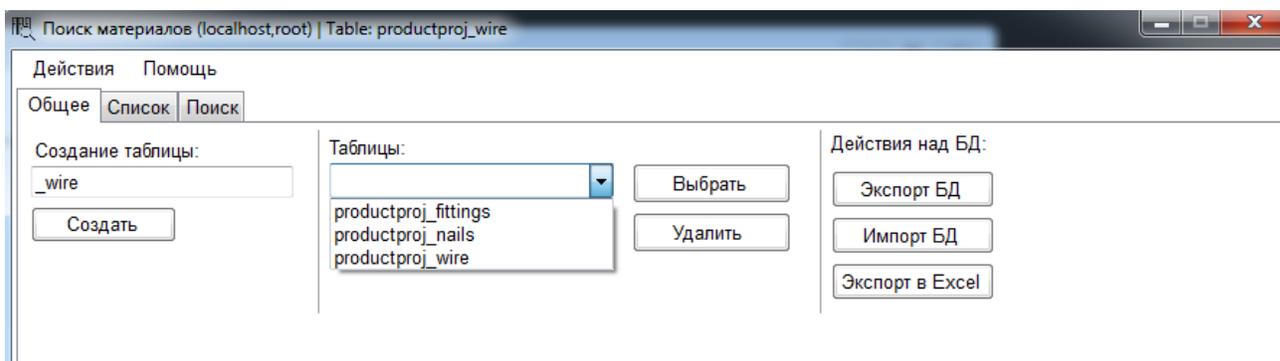


Рисунок 6 – Окно “Поиск материалов”

Следующим шагом, для выполнения поиска, необходимо перейти во вкладку “Поиск” и ввести свой поисковый запрос в соответствующее поле, а также выбрать необходимые опции (атрибуты) и затем нажать кнопку “Начать”. На рисунке 7 представлен результат поиска по запросу “Арматура”.

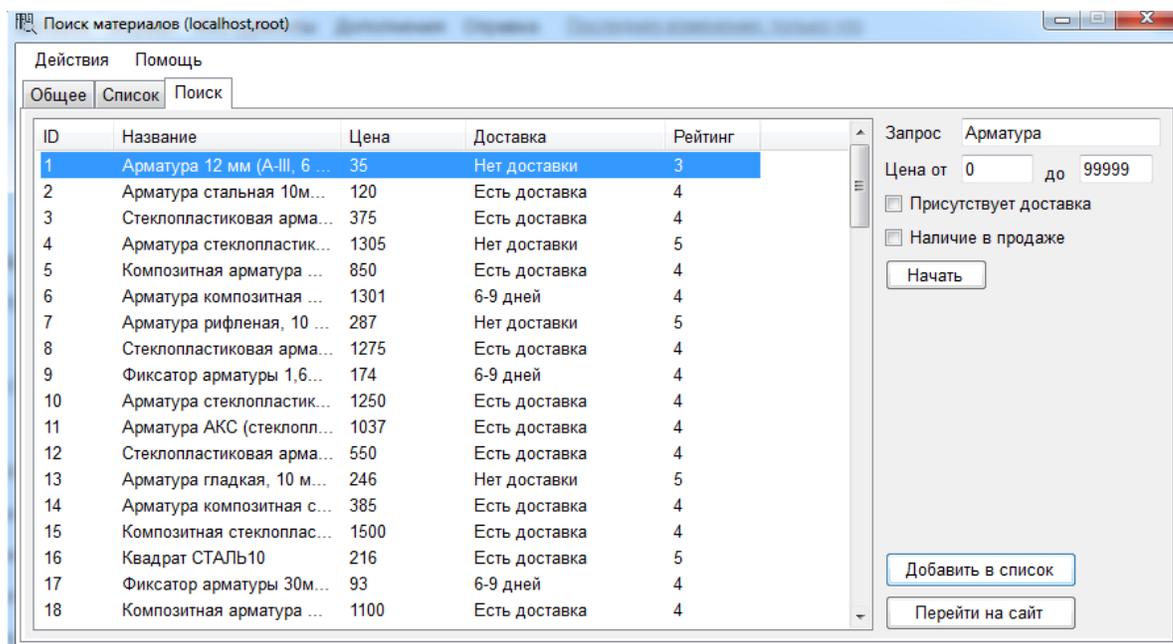


Рисунок 7 - Вкладка “Поиск материалов”

Также в данной вкладке присутствуют инструменты работы с поисковой выдачей, такие как добавление в список покупок (сохранить) и переход на сайт

производителя. В таблице поиска обозначены поля “название”, “цена” “доставка”, “рейтинг”, реализована возможность их сортировки по возрастанию или убыванию. Пример приведён в приложение А - листинг А.2

На рисунке 8 можно видеть, что при указании ценового диапазона, находятся товары только с соответствующей ценой.

ID	Название	Цена	Доставка	Рейтинг
1	Проволока медная 0,2...	45	Есть доста...	5
2	Проволока медная 0,2...	55	Есть доста...	5
3	Проволока медная дл...	10	Нет доставки	4
4	Медная проволока дл...	50	Есть доста...	4
5	Проволока медная Fo...	Нет в наличии	Нет доставки	Нет рей...
6	Проволока медная Fo...	Нет в наличии	Нет доставки	Нет рей...
7	Проволока медная Fo...	Нет в наличии	Нет доставки	Нет рей...
8	Проволока медная 0,2...	39	6-9 дней	5
9	Медная проволока дл...	85	Есть доста...	4
10	Медная проволока дл...	70	Есть доста...	4
11	Проволока медная, d ...	61	6-9 дней	5
12	Медная проволока дл...	85	Есть доста...	4
13	Медная проволока дл...	90	Есть доста...	4
14	Медная проволока дл...	70	Есть доста...	5
15	Проволока 0008314 ме...	79	Есть доста...	5
16	Медная проволока дл...	65	Есть доста...	4
17	Проволока Gamma DG...	43	5-6 дней	5

Рисунок 8 - Вкладка “Поиск материалов”

На рисунке 9 явно видно, что при выборе критериев «Присутствует доставка» и «Наличие в продаже» находятся только те товары, которые соответствуют выбранным критериям.

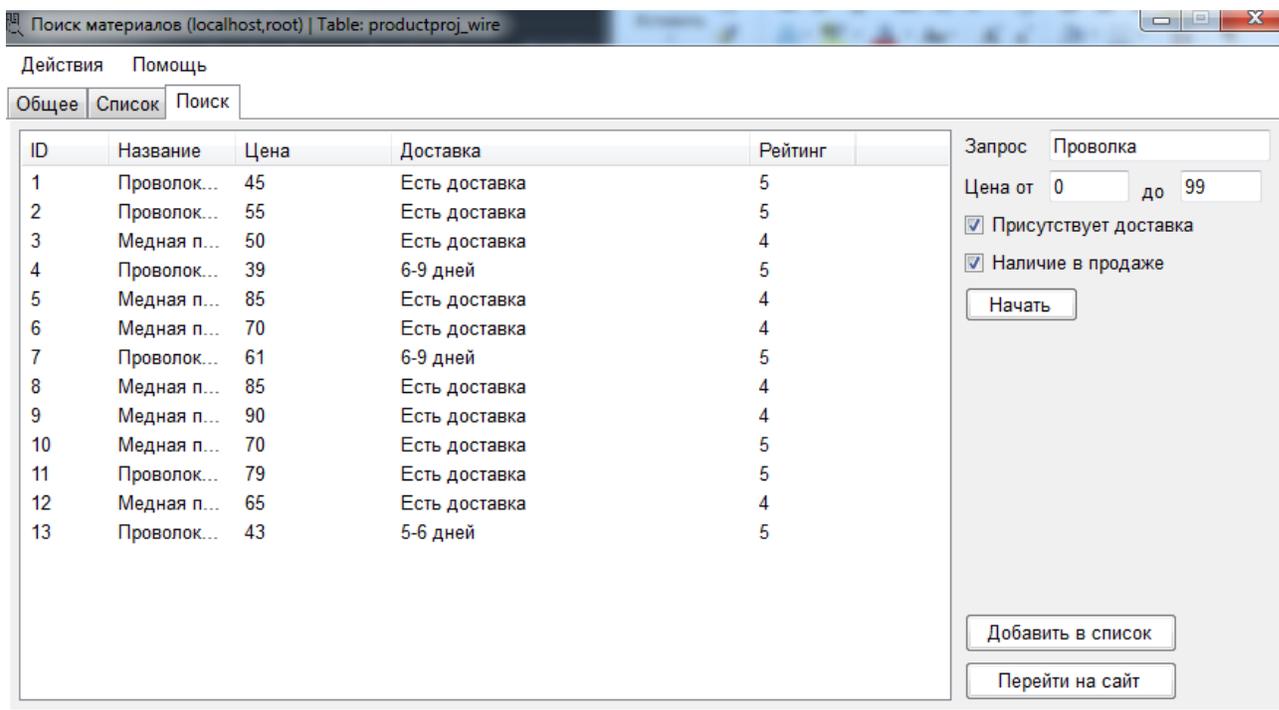


Рисунок 9 - Вкладка “Поиск материалов”

Для просмотра сохраненных товаров нужно зайти во вкладку “список” - там отображается перечень товаров, которые мы сохранили. Также в данной вкладке мы можем удалить выделенный элемент, перейти на сайт.

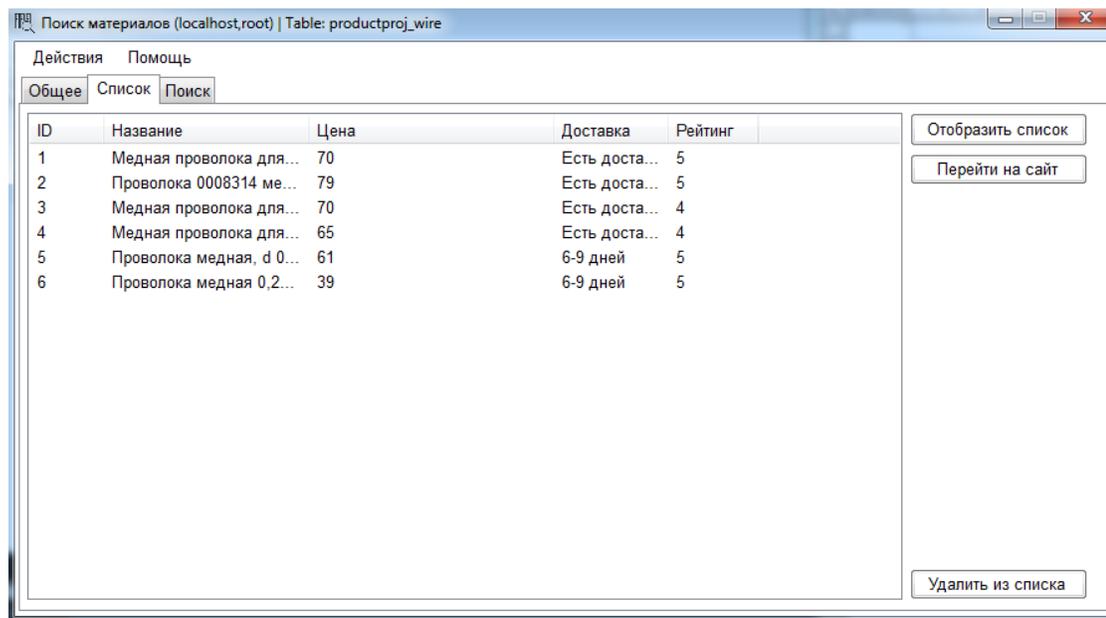


Рисунок 10 - Вкладка “Список”

На рисунке 11 видно, если пользователь уже выбрал таблицу, удалить или выбрать он не сможет. Удалить и выбрать можно только не выбранные таблицы.

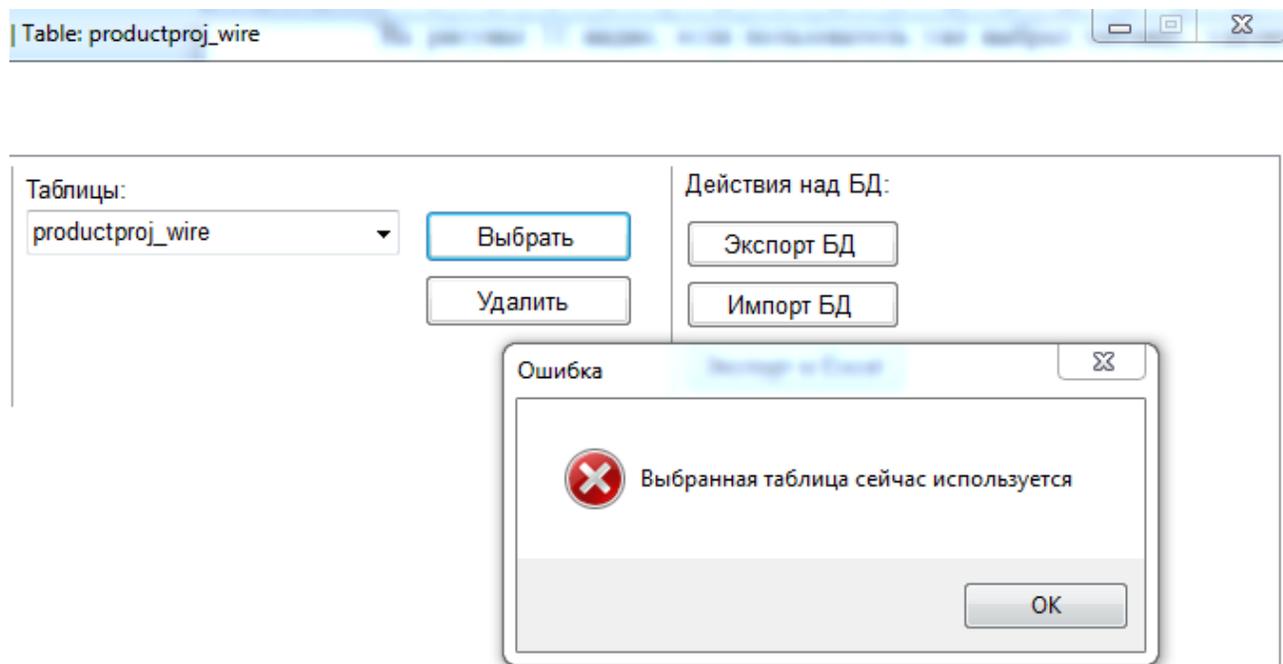


Рисунок 11 – Удаление таблицы

## **ЗАКЛЮЧЕНИЕ**

В выпускной квалификационной работе были представлены этапы создания надёжного и быстрого программного обеспечения, с помощью которого можно в любой момент времени по запросу пользователя предоставить требуемый перечень материалов и комплектующих изделий с возможностью фильтрации по указанным пользователем критериям или искать аналоги при отсутствии определенного вида товаров. При разработке данного приложения были использованы такие технологии как язык программирования С#, для интерфейса использовались библиотеки Windows Forms, Selenium, HTMLAgilityPack, также применялись регулярные выражения в качестве фильтра для парсинга необходимой информации, а для работы с выгруженной информацией используется MySQL Server. По результатам поисковой выдачи производится выборка и анализ результатов, удаляется ненужная информация.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Parsehub / Страница скачивания URL //: - <https://www.parsehub.com/> (Дата обращения 21.01.2019)
2. Import.io / Страница скачивания URL //: - <https://www.import.io/> (Дата обращения 21.01.2019)
3. Octoparse / Страница скачивания URL //: - <https://www.octoparse.com/> (Дата обращения 21.01.2019).
4. Руководство по Visual Studio C# / Страница скачивания URL //: - <https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/?view=vs-2019> (Дата обращения 31.05.2020).
5. Язык программирования C# и .NET / Страница скачивания URL //: - <https://metanit.com/sharp/general.php> (Дата обращения 31.05.2020).
6. Обзор конструктора Windows Forms / Страница скачивания URL //: - <https://docs.microsoft.com/ru-ru/visualstudio/designers/windows-forms-designer-overview?view=vs-2019> (Дата обращения 31.05.2020).
7. MySQL Installer 8.0.20 / Страница скачивания URL //: - <https://dev.mysql.com/downloads/installer/> (Дата обращения 4.04.2020).
8. MySQL Installer 8.0.20/Connector/Python 8.0.20 / Страница скачивания URL //: - <https://dev.mysql.com/downloads/connector/python/?os=3> (Дата обращения 4.04.2020).
9. Подключение к базе данных SQL Server с использованием C# / Страница скачивания URL //: - <https://o7planning.org/ru/10511/connecting-to-sql-server-database-using-csharp> (Дата обращения 2.05.2020).
10. Microsoft.Data.SqlClient / Страница скачивания URL //: <https://www.nuget.org/packages/Microsoft.Data.SqlClient/> (Дата обращения 2.05.2020).

11. System.NET / Страница скачивания URL //: - <https://docs.microsoft.com/ru-ru/dotnet/api/system.net?view=dotnet-plat-ext-3.1> (Дата обращения 20.05.2020)
12. System.Text.RegularExpressions / Страница скачивания URL //: - <https://docs.microsoft.com/ru-ru/dotnet/api/system.text.regularexpressions?view=netcore-3.1> (Дата обращения 22.05.2020)
13. Регулярные выражения C# / Страница скачивания URL //: - <http://www.nookery.ru/regular-expressions/> (Дата обращения 16.05.2020).
14. Яндекс маркет / Страница скачивания URL //: - <https://market.yandex.ru/> (Дата обращения 31.05.2020)