СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1. ОБЩАЯ ЧАСТЬ 4

1.1 Описание предметной области 4

1.2 Выбор языка программирования 4

2. СПЕЦИАЛЬНАЯ ЧАСТЬ 11

2.1 Постановка задачи 11

2.2 Структура данных 12

2.3 Эскизный проект 16

2.4 Технический проект 18

2.5 Руководство пользователя 30

3. ЗАКЛЮЧЕНИЕ 38

Список использованных источников 37

ПРИЛОЖЕНИЕ А Листинг программы

ПРИЛОЖЕНИЕ Б Результаты

# ВВЕДЕНИЕ

«Азбука» является обществом с ограниченной ответственностью занимающееся издательством книг.

Актуальной задачей для компании является разработка информационной системы, автоматизации процесса приёма заказов и обновление клиентской базы, что позволит снизить нагрузку на менеджеров и позволит им вести учёт заказов и клиентов в более удобной форме.

Объектами исследования являются существующие решения в области оформления заказов, клиентской базы организации.

Целью курсовой работы является разработка информационной системы для автоматизации оформления заказов, работы с клиентской базой и отслеживания изменения статуса заказов.

Основные задачи:

* Разработка понятного и удобного пользовательского интерфейса;
* Разработка базы данных для хранения информации о клиентах, сделанных ими заказах и их статусах;
* Реализация функций оформления, удаления, редактирования заказов;
* Реализация функций добавления, удаления, редактирования авторов и книг;

# 1. ОБЩАЯ ЧАСТЬ

## 1.1 Описание предметной области

Книжное издательство «Азбука» предоставляет следующий функционал для пользователя: добавление, удаление и редактирование данных в БД через интерфейс системы, введение клиентов.

Разрабатываемая система должна содержать в себе информацию о списке клиентов, списке предоставляемых услуг, заказах и отслеживать изменение статуса заказов.

Основным пользователем системы будет являться сотрудник- ведущий редактор. Его учётная запись должна иметь функционал, такой как: добавление ФИО авторов, их книг( название, издание, дата, стоимость), детали заказчиков, такие как, наименование компании, телефон и адрес, сами заказы с датой, стоимостью, статусом заказа, названием книги и данными заказчика.

## 1.2 Выбор языка программирования

**C#** — это язык программирования, предназначенный для разработки самых разнообразных приложений, предназначенных для выполнения в среде .NET Framework. Язык C# прост, безопасен и объектно-ориентирован. Благодаря множеству нововведений C# обеспечивает возможность быстрой разработки приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C.

**Visual C#** — это реализация языка C# корпорацией Майкрософт. Поддержка Visual C# в Visual Studio обеспечивается с помощью полнофункционального редактора кода, компилятора, шаблонов проектов, конструкторов, мастеров кода, мощного и удобного отладчика и многих других средств. Библиотека классов .NET Framework предоставляет доступ ко многим службам операционной системы и другим полезным, правильным классам, что существенно ускоряет цикл разработки.

У языка программирования C# на сегодняшний день множество поклонников, среди них большое количество компаний и стартапов создающих инди-игры и 3D-игры.

К многочисленным преимуществам этого языка относят:

* поддержку подавляющего большинства продуктов Microsoft;  
  для небольших компаний и некоторых индивидуальных разработчиков бесплатными являются такие инструменты, Visual Studio, облако Azure, Windows Server, Parallels Desktop для Mac Pro и многие другие;
* большое количество синтаксического сахара, представляющего собой специальные конструкции, разработанные для понимания и написания кода. Они не имеют значения при компиляции;
* порог вхождения у языка C# низкий. Его синтаксис имеет много схожего с другими языками программирования, благодаря чему облегчается переход для программистов. Язык C# считается наиболее понятным и подходящим для новичков;
* после покупки Xamarin на C# можно писать программы и приложения для таких операционных систем, как iOS, Android, MacOS и Linux;
* имеется целое сообщество из опытных программистов;
* сегодня в любом регионе России имеется много вакантных мест на должность C#-программиста.

Кроме многочисленных плюсов эта программа имеет некоторые недостатки. Среди них следует выделить:

* приоритетная ориентированность на Windows платформу;
* язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся. Крупной компании покупка лицензионной версии этого языка обойдется в круглую сумму;
* в языке осталась возможность использования оператора безусловного перехода.

Необходимо отметить, что язык C# не представляет собой сложности для новичков, так как его легко изучить и понять.

Технология WPF (Windows Presentation Foundation) является часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.

Если при создании традиционных приложений на основе WinForms за отрисовку элементов управления и графики отвечали такие части ОС Windows, как User32 и GDI+, то приложения WPF основаны на DirectX. В этом состоит ключевая особенность рендеринга графики в WPF: используя WPF, значительная часть работы по отрисовке графики, как простейших кнопочек, так и сложных 3D-моделей, ложиться на графический процессор на видеокарте, что также позволяет воспользоваться аппаратным ускорением графики.

Одной из важных особенностей является использование языка декларативной разметки интерфейса XAML, основанного на XML: вы можете создавать насыщенный графический интерфейс, используя или декларативное объявление интерфейса, или код на управляемых языках C# и VB.NET, либо совмещать и то, и другое.

Преимущества WPF

* Использование традиционных языков .NET-платформы - C# и VB.NET для создания логики приложения
* Возможность декларативного определения графического интерфейса с помощью специального языка разметки XAML, основанном на xml и представляющем альтернативу программному созданию графики и элементов управления, а также возможность комбинировать XAML и C#/VB.NET
* Независимость от разрешения экрана: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением.
* Новые возможности, которых сложно было достичь в WinForms, например, создание трехмерных моделей, привязка данных, использование таких элементов, как стили, шаблоны, темы и др.
* Хорошее взаимодействие с WinForms, благодаря чему, например, в приложениях WPF можно использовать традиционные элементы управления из WinForms.
* Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимаций, привязка данных, стили, шаблоны, темы и многое другое
* Аппаратное ускорение графики - вне зависимости от того, работаете ли вы с 2D или 3D, графикой или текстом, все компоненты приложения транслируются в объекты, понятные Direct3D, и затем визуализируются с помощью процессора на видеокарте, что повышает производительность, делает графику более плавной.
* Создание приложений под множество ОС семейства Windows - от Windows XP до Windows 10

В тоже время WPF имеет определенные ограничения. Несмотря на поддержку трехмерной визуализации, для создания приложений с большим количеством трехмерных изображений, прежде всего игр, лучше использовать другие средства - DirectX или специальные фреймворки, такие как Monogame или Unity.

Также стоит учитывать, что по сравнению с приложениями на Windows Forms объем программ на WPF и потребление ими памяти в процессе работы в среднем несколько выше. Но это с лихвой компенсируется более широкими графическими возможностями и провышенной производительностью при отрисовке графики.

Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работает с объектами.

Первая версия Entity Framework - 1.0 вышла еще в 2008 году и представляла очень ограниченную функциональность, базовую поддержку ORM (object-relational mapping - отображения данных на реальные объекты) и один единственный подход к взаимодействию с бд - Database First. С выходом версии 4.0 в 2010 году многое изменилось - с этого времени Entity Framework стал рекомендуемой технологией для доступа к данным, а в сам фреймворк были введены новые возможности взаимодействия с бд - подходы Model First и Code First.

Дополнительные улучшения функционала последовали с выходом версии 5.0 в 2012 году. И наконец, в 2013 году был выпущен Entity Framework 6.0, обладающий возможностью асинхронного доступа к данным.

Центральной концепцией Entity Framework является понятие сущности или entity. Сущность представляет набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

Любая сущность, как и любой объект из реального мира, обладает рядом свойств. Например, если сущность описывает человека, то мы можем выделить такие свойства, как имя, фамилия, рост, возраст, вес. Свойства необязательно представляют простые данные типа int, но и могут представлять более комплексные структуры данных. И у каждой сущности может быть одно или несколько свойств, которые будут отличать эту сущность от других и будут уникально определять эту сущность. Подобные свойства называют ключами. При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из бд, но и получать объекты, связанные различными ассоциативными связями. Другим ключевым понятием является Entity Data Model. Эта модель сопоставляет классы сущностей с реальными таблицами в БД.

Entity Data Model состоит из трех уровней: концептуального, уровень хранилища и уровень сопоставления (маппинга).

На концептуальном уровне происходит определение классов сущностей, используемых в приложении.

Уровень хранилища определяет таблицы, столбцы, отношения между таблицами и типы данных, с которыми сопоставляется используемая база данных.

Уровень сопоставления (маппинга) служит посредником между предыдущими двумя, определяя сопоставление между свойствами класса сущности и столбцами таблиц. Таким образом, мы можем через классы, определенные в приложении, взаимодействовать с таблицами из базы данных.

Среда SQL Server Management Studio — это интегрированная среда для управления инфраструктурой SQL Server и базы данных SQL Azure. Среда Management Studio предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server. Она также предоставляет средства для развертывания, отслеживания и обновления данных-уровня компонентов, таких как базы данных и хранилищ данных, используется приложениями, а также для создания запросов и скриптов.

# 2. СПЕЦИАЛЬНАЯ ЧАСТЬ

## 2.1 Постановка задачи

Книжное издательство выпускает и продаёт книги заказчикам: организациям, магазинам, библиотекам и прочим компаниям, имеющим отношение и разрешение в этой сфере. Пользователем системы будет являться сотрудник. Его учётная запись должна иметь функционал, такой как: добавление заказов, заказчиков, новых книг и их авторов, а также редактирование и удаление всего перечисленного. Основной функцией является оформление заказов. Программа хранит в себе историю заказов, клиентскую базу и предоставляемые услуги.

Данные будут хранится на сервере баз данных компании — это позволит оградить их от несанкционированного доступа и обеспечить доступ данных с любого авторизованного устройства с установленной копией программы, что обеспечивает мобильность, так как она в любое время может быть инсталлирована, но данные останутся в полной сохранности. Также хотелось бы отметить, что для доступа к данным используется Entity Framework — это обеспечивает полную защиту базы данных от SQL-инъекций.

Для корректной работы системы необходим компьютер, соответствующий минимальным системным требованиям:

* Процессор с частотой 1ГГц:
* Оперативная память 2 ГБ;
* Операционная система Windows 7 и выше.

## 2.2 Структура данных

Для информационной системы была разработана база данных, которая имеет следующую схему:

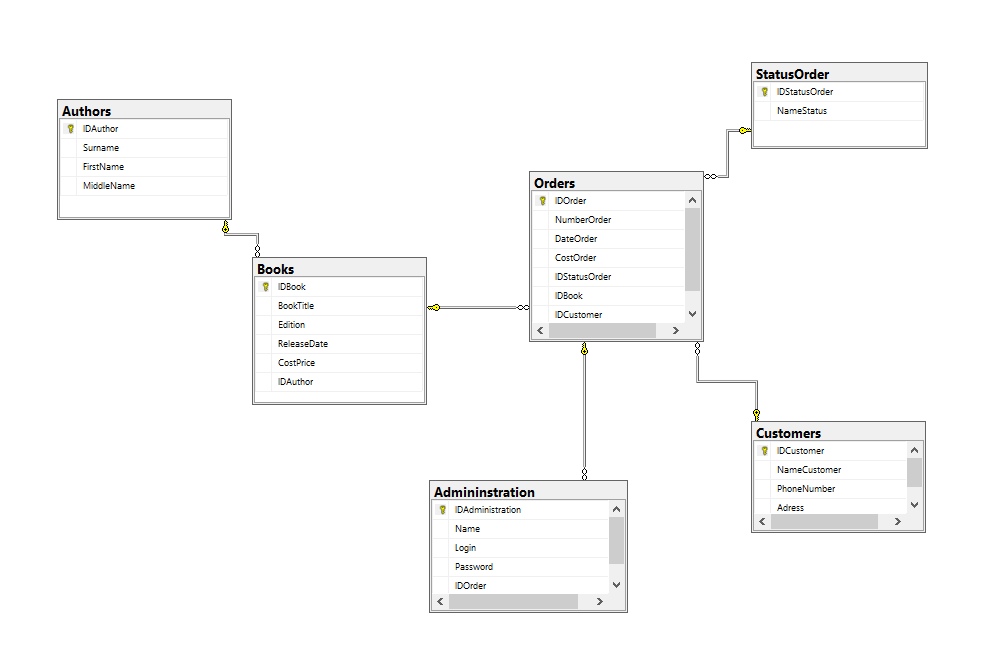


Рисунок 1— Схема базы данных

Таблица Authors хранит информацию о авторах.

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| idAuthor | INT | Первичный ключ |
| Surname | NVARCHAR | Фамилия автора |
| Firstname | NVARCHAR | Имя автора |
| MidlleName | NVARCHAR | Фамилия автора |

Таблица 1— Authors

Таблица Books предназначена для хранения информации о менеджерах.

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| idBook | INT | Первичный ключ |
| BookTitle | NVARCHAR | Название |
| Edition | NVARCHAR | Издание |
| ReleaseDate | NVARCHAR | Дата релиза |
| CostPrice | NVARCHAR | Цена |
| IdAuthor | NVARCHAR | Внешний ключ |

Таблица 2— Books

Таблица Orders хранит информацию о заказах.

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| IdOrder | INT | Первичный ключ |
| NumberOrder | INT | Номер заказа |
| DateOrder | NVARCHAR | Дата Заказа |
| CostOrder | INT | Стоимость заказа |
| IDStatusOrder | INT | Внешний ключ |
| IDBook | INT | Внешний ключ |
| IDCustomer | INT | Внешний ключ |

Таблица 3— Order

Таблица Customers хранит информацию о заказчиках.

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| IDCustomer | INT | Первичный ключ |
| NameCustomer | NVARCHAR | Наименование компании |
| PhoneNumber | INT | телефон |
| Adress | NVARCHAR | Описание услуги |

Таблица 4— Customers

Таблица StatusOrder хранит информацию о статусе заказа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| IDStatusOrder | INT | Id типа услуги |
| NameStatus | NVARCHAR | Наименование статуса |

Таблица 5— StatusOrder

Таблица Administration хранит информацию о администраторах

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| idAdministration | INT | Первичный ключ |
| Name | NVARCHAR | Имя |
| Login | NVARCHAR | Логин |
| Password | NVARCHAR | Пароль |
| IdOrder | INT |  |
| DatePayment | INT | Дата оплаты |

Таблица 6 — Administration

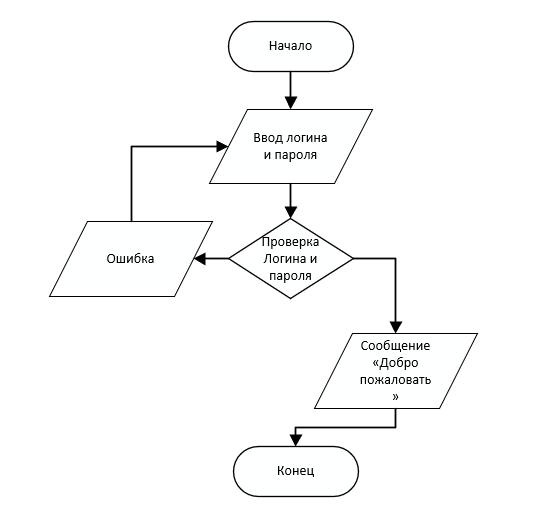
## 2.3 Эскизный проект

Для реализации всего необходимого функционала требуется разработка следующих подпрограмм:

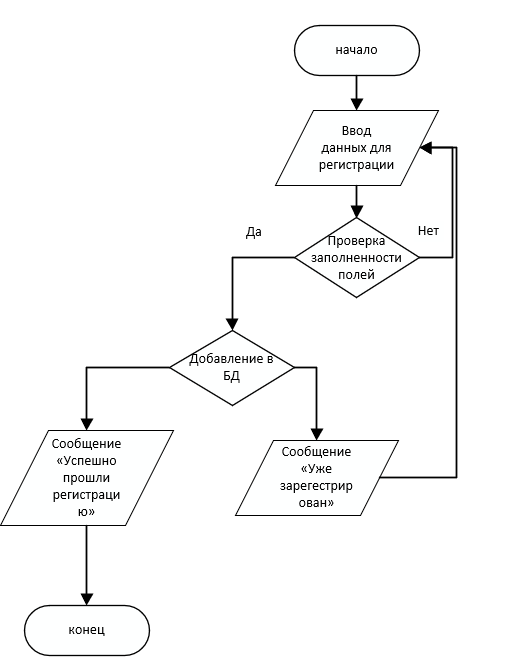
* Регистрация новых клиентов – сотрудник компании, ведущий редактор книжного издательства, производит регистрацию новых клиентов. Добавление новых клиентов осуществляется через графический интерфейс пользователя, в котором вводится следующая информация о клиенте: имя, логин и пароль. Все введённые данные автоматически добавляются в базу данных.
* Добавление новых данных об авторах - сотрудник компании, ведущий редактор, производит добавление новых данных, которое осуществляется через графический интерфейс пользователя, в котором вводится следующая информация об услуге: фамилия, имя и отчество автора, который хочет осуществить заказ. Все введенные данные автоматически добавляются в базу данных.
* Добавление новых данных о книгах - сотрудник компании, ведущий редактор, производит добавление новых данных, которое осуществляется через графический интерфейс пользователя, в котором вводится следующая информация об услуге: название книги, ее издание, дата, цена и автор. Все введенные данные автоматически добавляются в базу данных.
* Добавление новых данных о заказчиках- сотрудник компании, ведущий редактор, производит добавление новых данных, которое осуществляется через графический интерфейс пользователя, в котором вводится следующая информация об услуге: наименование компании, телефон заказчика и адрес. Все введенные данные автоматически добавляются в базу данных.
* Добавление новых данных о заказах - сотрудник компании, ведущий редактор, производит добавление новых данных, которое осуществляется через графический интерфейс пользователя, в котором вводится следующая информация об услуге: номер, дата, стоимость, статус заказа, название книги, и имя заказчика. Все введенные данные автоматически добавляются в базу данных.
* Удаление данных - сотрудник компании, менеджер по продажам, производит удаление пиломатериалов. Удаление осуществляется через графический интерфейс пользователя, в котором менеджеру необходимо выбрать при помощи клика нужную услугу, а затем нажать кнопку удалить. Все данные о пиломатериале автоматически удаляются из базы данных.
* Регистрация новых заказов - сотрудник компании, менеджер по продажам, производит регистрацию новых заказов компании. Регистрация новых заказов осуществляется через графический интерфейс пользователя, в котором вводится следующая информация о заказе: номер заказа, вид заказа, дата регистрации заказа, уникальный идентификатор клиента, уникальный идентификатор категории, уникальный идентификатор менеджера и оплаты. Все введённые данные автоматически добавляются в базу данных.
* Возможность редактировать все данные - сотрудник компании, менеджер по продажам, производит редактирование имеющихся данных. Редактирование осуществляется через графический интерфейс пользователя, в котором менеджеру необходимо редактировать нужные ему данные прямо в таблице. Все внесенные изменения автоматически сохраняются в базе данных.
* Удаление заказов - сотрудник компании, менеджер по продажам, производит удаление заказов. Удаление осуществляется через графический интерфейс пользователя, в котором менеджеру необходимо выбрать при помощи клика нужный заказ, а затем нажать кнопку удалить. Все данные о заказе автоматически удаляются из базы данных.
* Удаление клиентов из клиентской базы - сотрудник компании, менеджер по продажам, производит удаление клиентов из базы. Удаление осуществляется через графический интерфейс пользователя, в котором менеджеру необходимо выбрать при помощи клика нужного клиента, а затем нажать кнопку удалить. Все данные о клиенте автоматически удаляются из базы данных.

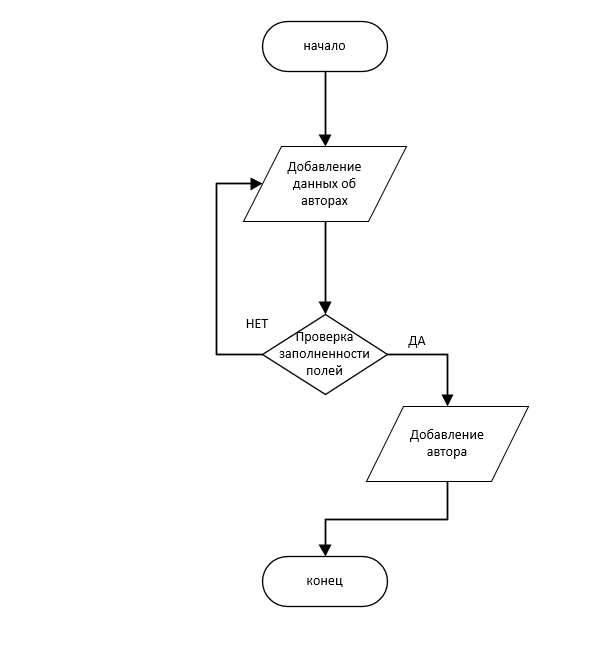
## 2.4 Технический проект

Авторизация.

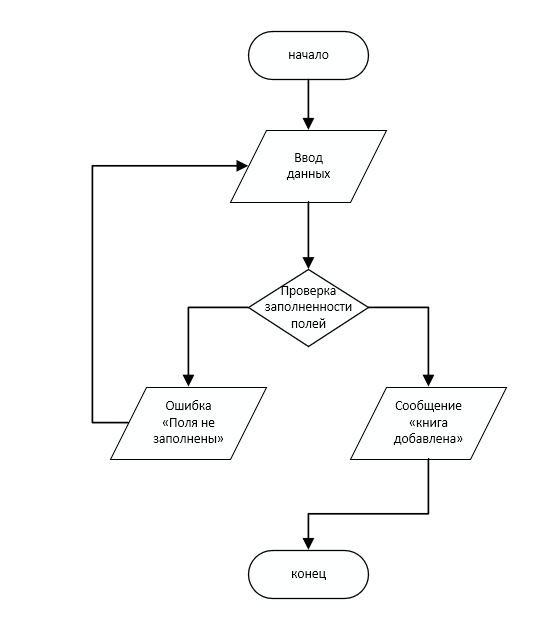


Регистрация.

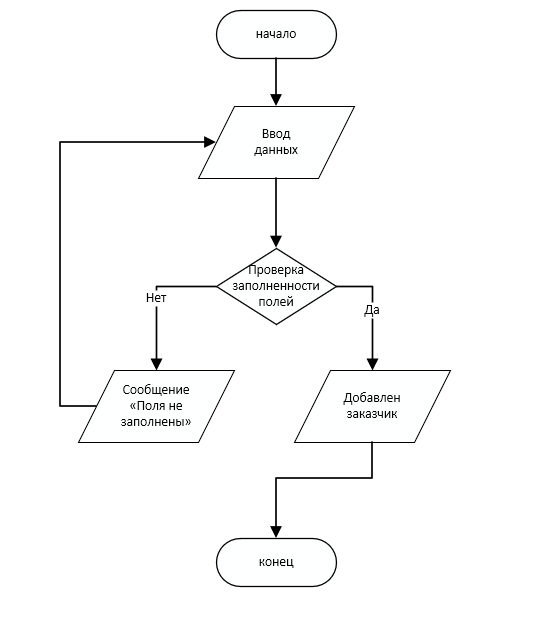


Добавление автор. 

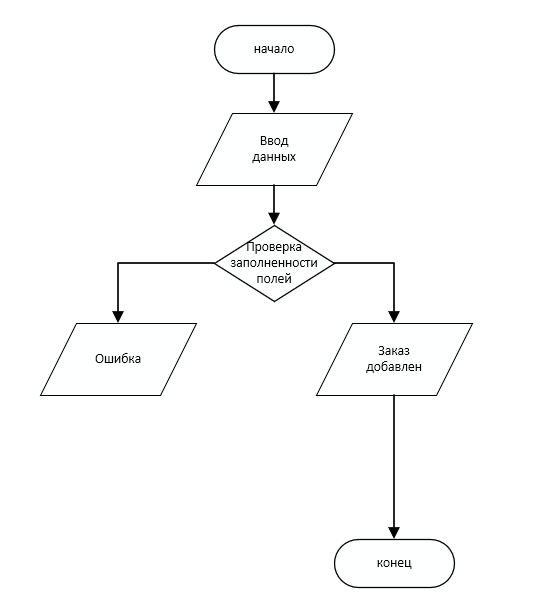
Добавление книги.



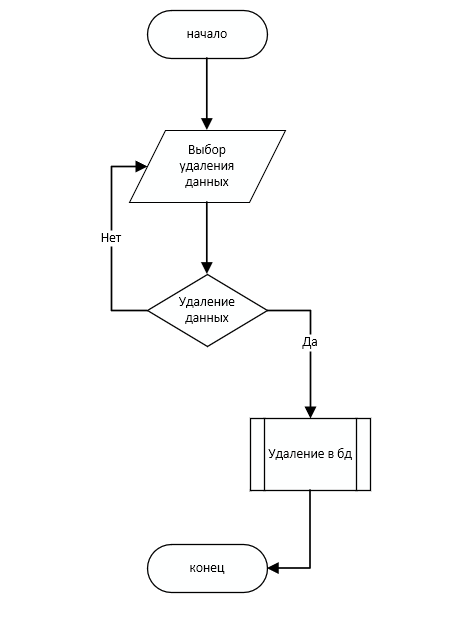
Добавление Заказчики.



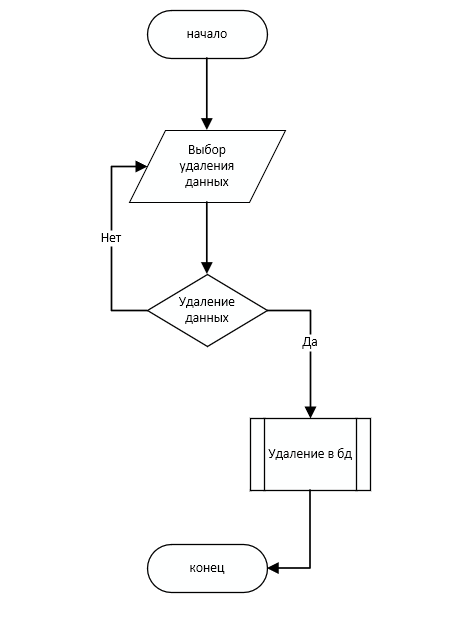
Добавление заказы.



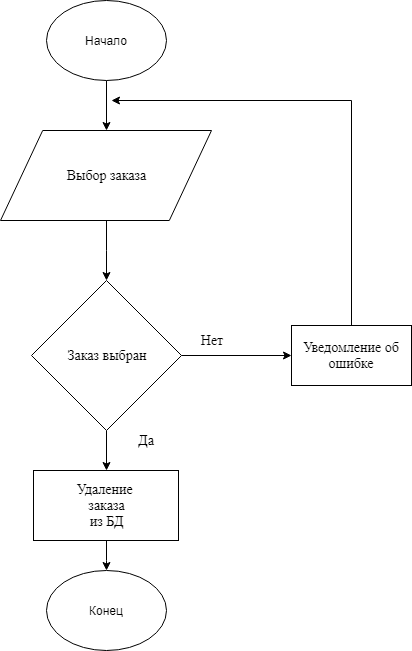
Удаление книги.



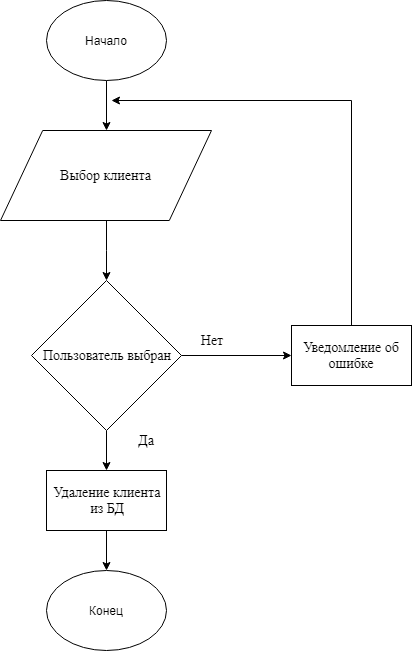
Удаление заказчики.



Удаление заказа.



Удаление клиента.



## 2.5 Руководство пользователя

При запуске программы пользователю необходимо пройти процесс авторизации (Рисунок 2 – Окно авторизации).



Рисунок 2 – Окно авторизации.

В данном окне пользователю необходимо ввести свои логин и пароль в соответствующие поля. Если пользователь допустит ошибку, при написании логина или пароля, программа укажет ему на это (Рисунок 3 – Сообщение об ошибке в логине или пароле).

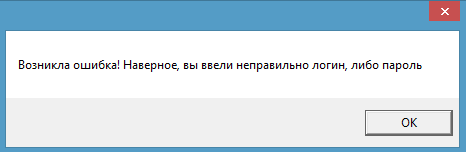


Рисунок 3 – Сообщение об ошибке в логине или пароле.

После нажатия на кнопку «Войти» менеджер авторизуется в системе и переходит в окно, котором ему нужно выбрать, что он хочет сделать (Рисунок 4 – Выбор дальнейших действий).

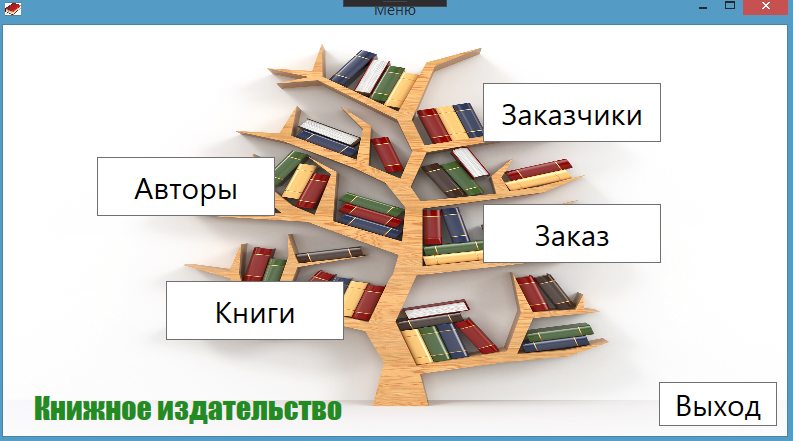


Рисунок 4 – Выбор дальнейших действий.

При нажатии на кнопку Авторы пользователя переносит в другое окно, где отображена информация о авторах. Окно с авторами показана на рисунке 5.

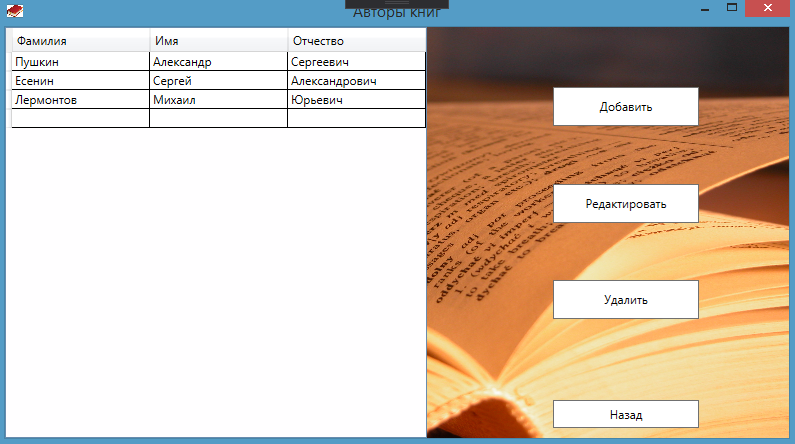


Рисунок 5 – Ассортимент пиломатериалов.

В данном окне сотрудник может добавить, удалить или редактировать данные об авторе при нажатии на соответствующую кнопку (Рисунок 6 – Добавление автора).

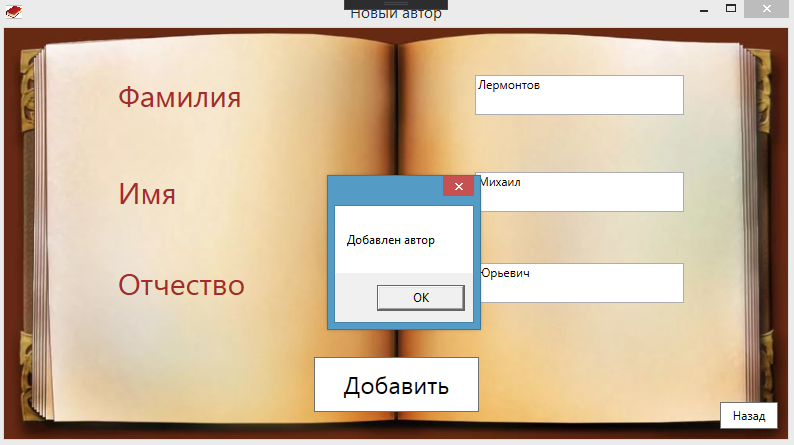


Рисунок 6 – Добавление автора.

Сотрудник должен заполнить все существующие поля. Если какое-либо поле не будет заполнено, программа предупредит об этом пользователя соответствующим уведомлением (Рисунок 7 – Уведомление о не заполненных полях).

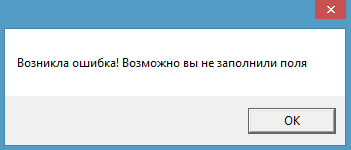


Рисунок 7 – Уведомление о не заполненных полях.

Редактирование вносится в самой таблице, потом по нажатию кнопки редактировать, информация записывается в БД. Редактирование данных об авторе отображено на рисунке 8

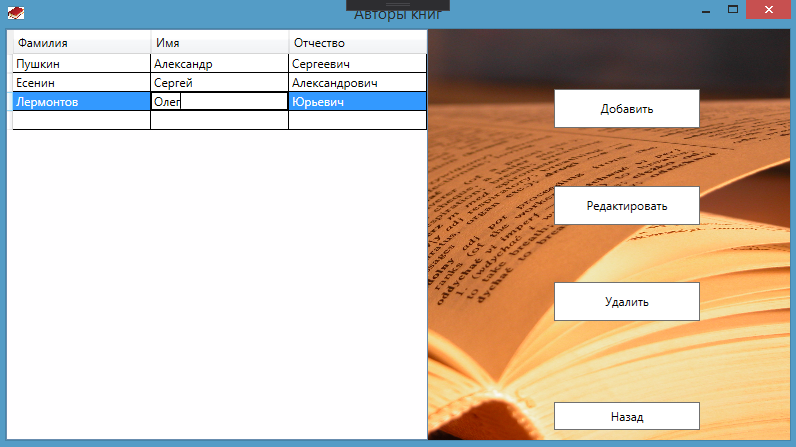


Рисунок 8 – Редактирование данных об авторах.

Для того что бы удалить данные сотрудник выбирает в таблице нужную строку и нажимает соответствующую кнопку удалить, далее подтверждает своё действие. Подтверждение удаления показано на рисунке 9. После подтверждения данный об авторе сразу удаляются из таблицы базы данных

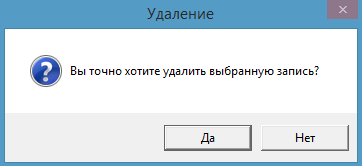


Рисунок 9 –Подтверждение удаления данных.

При нажатии на кнопку Книги в главном меню, открывается окно с данными о книгах. В данном окне сотрудник может добавить, удалить или редактировать данные о книгах при нажатии на соответствующую кнопку (Рисунок 10 – Добавление книги).

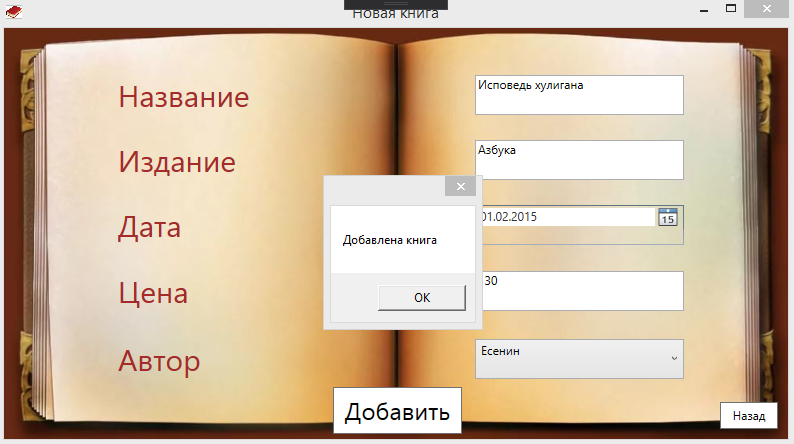


Рисунок 10 –Уведомление об успешном добавлении книги.

Так же, как и в предыдущем окне сотрудник должен заполнить все поля для заполнения. При не заполнении полей, пользователю сообщат об ошибке.

Нажимаем кнопку Ок, и новая книга добавляется в список книг.

Для того что бы удалить данные сотрудник выбирает в таблице нужную строку и нажимает соответствующую кнопку удалить, далее подтверждает своё действие. Подтверждение удаления показано на рисунке 11. После подтверждения данная книга сразу удаляются из таблицы базы данных.

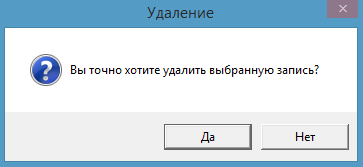
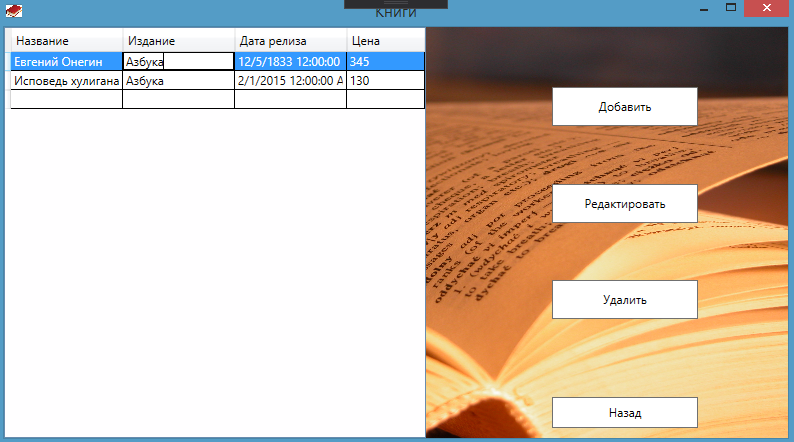


Рисунок 11 – Подтверждение удаление книги.

Редактирование книг вносится в самой таблице, потом по нажатию кнопки редактировать, информация записывается в БД. Редактирование данных об авторе отображено на рисунке 12.

  
Рисунок 12 – Редактирование книг.

При нажатии на кнопку Заказчики в главном меню, открывается окно с данными о Заказчиках. В данном окне сотрудник может добавить, удалить или редактировать данные о заказчиков при нажатии на соответствующую кнопку (Рисунок 13 – Добавление заказчика).

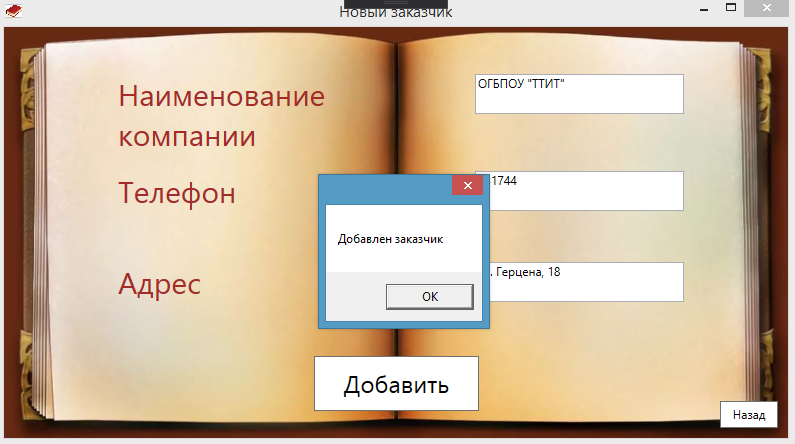


Рисунок 13 – Уведомление об успешном добавлении заказчика.

Редактирование заказчиков вносится в самой таблице, потом по нажатию кнопки редактировать, информация записывается в БД. Редактирование данных об авторе отображено на рисунке 14.

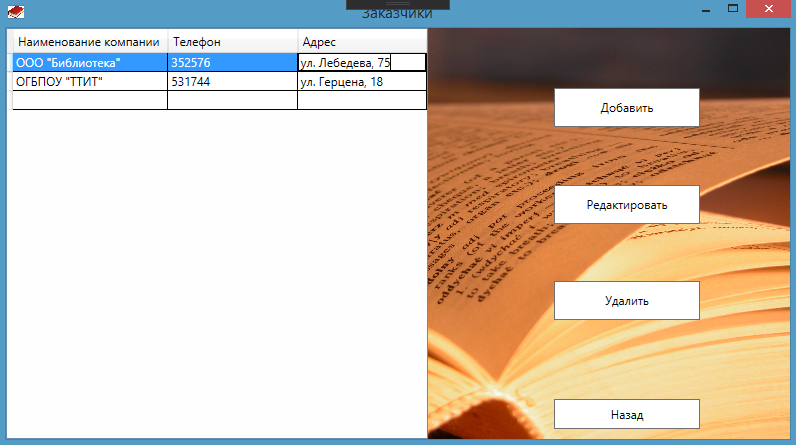


Рисунок 14 – Редактирование заказчиков

Для того что бы удалить данные сотрудник выбирает в таблице нужную строку и нажимает соответствующую кнопку удалить, далее подтверждает своё действие. Подтверждение удаления показано на рисунке 15. После подтверждения данный заказчик сразу удаляется из таблицы базы данных.

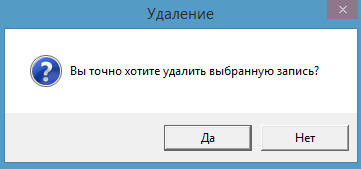


Рисунок 15 – Подтверждение удаления заказчиков

При нажатии на кнопку Книги в главном меню, открывается окно с данными о книгах. В открывшемся окне, выводится подробная информация о всех заказах. сотрудник имеет возможность добавить, удалить или редактировать заказ. Добавление нового заказа показано на рисунке 16.

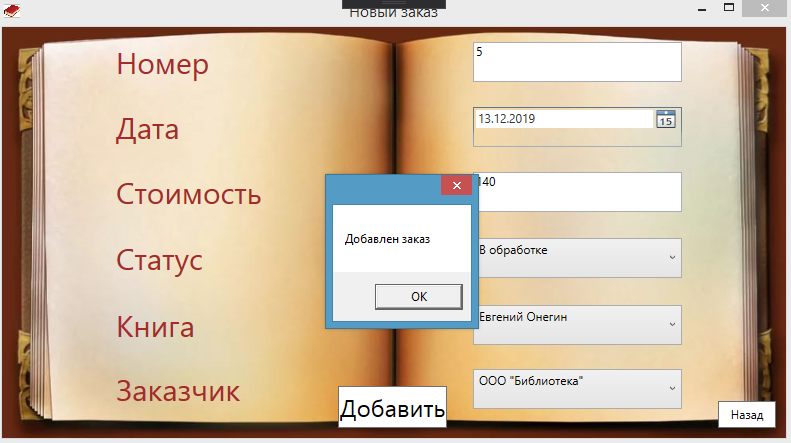


Рисунок 16 – Добавление заказа.

Для того что бы удалить данные сотрудник выбирает в таблице нужную строку и нажимает соответствующую кнопку удалить, далее подтверждает своё действие. Подтверждение удаления показано на рисунке 17. После подтверждения заказ сразу же пропадает из списка всех заказов.

.

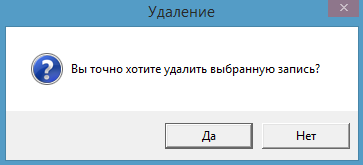


Рисунок 17 – Уведомление об успешном удалении заказа.

# 3. ЗАКЛЮЧЕНИЕ

В процессе разработки информационной системы была проделана крупная работа. В соответствии с техническим заданием выполнены все поставленные требования:

* разработан удобный и понятный интерфейс;
* разработан тематический дизайн
* в программе есть регистрация и авторизация;
* для сотрудника предусмотрены функции добавления, редактирования и удаления данных;
* проект выполнен точно в срок.

Приложение простое и доступное для начинающих пользователей, так как имеет простой и удобный интерфейс. Пользователь легко может добавить данные, или внести изменения. Есть возможность доработки приложения: добавление нового функционала, его расширения, для работы не только сотрудника, но и клиентов. В ходе выполнения проекта были получены и закреплены знания и умения по созданию информационных систем. Полученный опыт будет полезен в будущем, при работе с другими более масштабными и сложными проектами.

# Список использованных источников

1) Виктор Гольцман. MySQL 5.0. Библиотека программиста. 1-е издание. - СПб.: Питер, 2016. – 546 с.

2) Джеффри Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е издание. - СПб.: Питер, 2013. – 1245 с.

3) Эндрю Троелсен. Язык программирования C# 6.0 и платформа .NET 4.6. 7-e издание. – Москва.: Вильямс, 2016. – 1440 с.

2) Джеффри Рихтер. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е издание. - СПб.: Питер, 2013. – 1245 с.

3) Эндрю Троелсен. Язык программирования C# 6.0 и платформа .NET 4.6. 7-e издание. – Москва.: Вильямс, 2016. – 1440 с.