

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1.ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ.....	5
2.ПРАКТИЧЕСКИЙ РАЗДЕЛ	11
3.ОХРАНА ТРУДА	21
ЗАКЛЮЧЕНИЕ	23
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	24
ПРИЛОЖЕНИЕ	

ВВЕДЕНИЕ

В настоящее время необходимо хранить большое количество информации. Для этого необходимо использовать современные и многофункциональные базы данных. Они позволяют автоматизировать работу с большим количеством информации, обеспечить сортировку по различным критериям, добавление, изменение, удаление новых данных.

Данный программный продукт значим и актуален в настоящее время, так как он позволяет оптимизировать процесс работы сотрудника риэлторского агентства за счет более удобного пользовательского интерфейса, оптимизации использования аппаратных ресурсов.

В процессе выбора темы и разработки данного программного продукта была поставлена цель – разработка и реализация информационной системы «Риэлторское агентство», которая позволит автоматизировать деятельность работника данной компании.

Для достижения поставленной цели, необходимо выполнить следующие задачи:

- 1) Составить полный список эксплуатационных и функциональных требований.
- 2) Составить ER – модель программного продукта, словарь данных для таблиц базы данных.
- 3) Реализовать информационную систему в выбранной среде программирования.
- 4) Реализовать этапы тестирования программного продукта.
- 5) Организовать ввод в эксплуатацию данной информационной системы.

1.ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

1.1 Постановка задачи и определение предметной области.

Целью данной курсовой работы является создание информационной системы «Риэлторское агентство», которая будет включать в себя базу данных и пользовательский интерфейс, позволяющий работать с базой данных.

Правильно организованная работа информационной системы данной компании позволит облегчить сотруднику работу с большим количеством информации.

Данная информационная система будет позволять обрабатывать объявления, добавлять их, а также управлять менеджерами. Проверка логина и пароля происходит из базы данных. Данные обо всех объектах базы данных хранятся в таблицах, которые создаются в формах Visual Studio. Информация о менеджерах и объявлениях будет храниться в соответствующих таблицах. Получение информации из базы данных в Visual Studio будет организовываться путем создания и выполнения запросов на языке SQL.

Ввод данных в базу осуществляется несколькими способами. Первый – ручной способ, когда данные вносятся в базу данных напрямую в Access. Второй – ввод данных через пользовательское приложение, созданное в Visual Studio. Для большего удобства предпочтительнее использовать второй способ.

Работа предприятия схематично показана на рисунке 1.

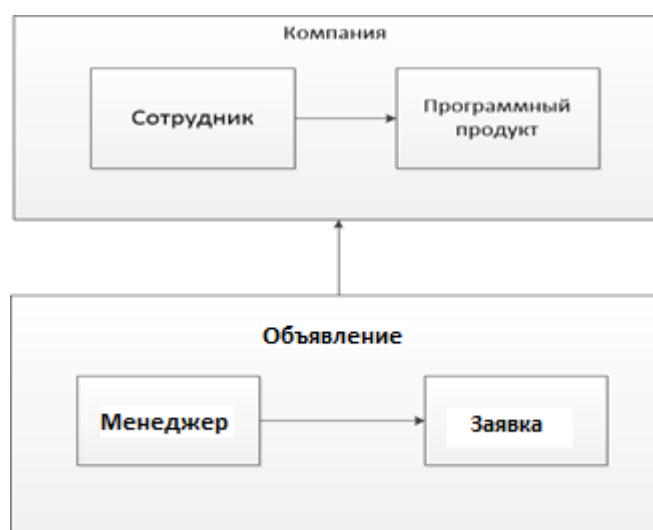


Рисунок 1 - Схема работы предприятия

1.2 Построение базы данных программного продукта

Правильно и логически построенная база данных позволит ускорить процесс создания программного продукта для информационной системы. Для быстрого построения базы данных, необходимо построить ER – модель данных, на которой будет показано взаимодействие основных сущностей. Этими сущностями являются менеджеры и объявления. ER-модель данных представлена на рисунке 2.

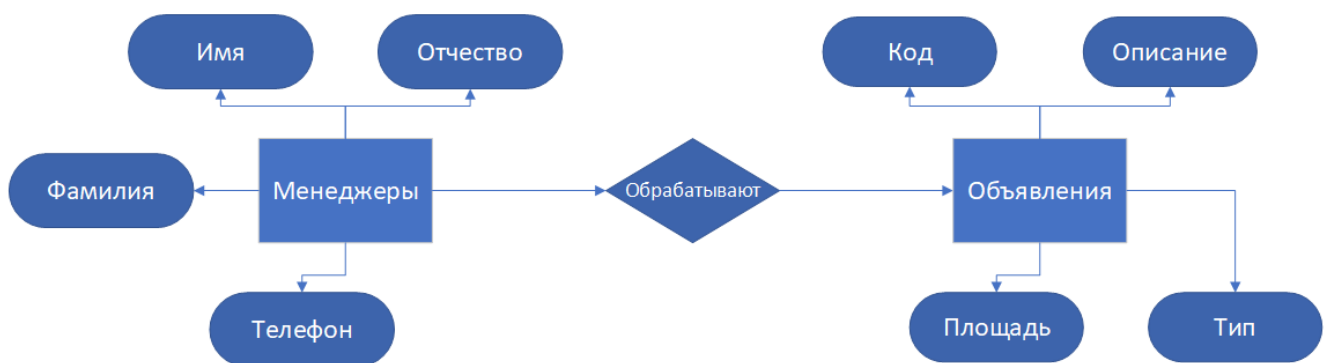


Рисунок 2 - ER – модель данных

Построение таблиц базы данных начинается с основы предприятия – объявления. Таблица «Объявления» будет содержать в себе информацию об объявлениях, их описании и т.д. Схема таблицы «Объявления» представлена на рисунке 3.

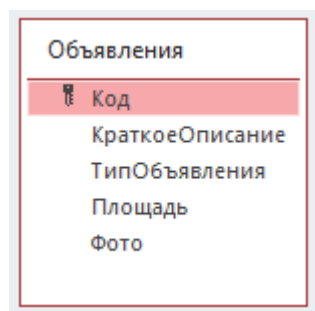


Рисунок 3 - Схема таблицы Объявления

Следующая часть базы данных – таблица «Сотрудники». В этой таблице будет содержаться информация о менеджерах, их информации и т.д. Схема этой таблицы представлена на рисунке 4.

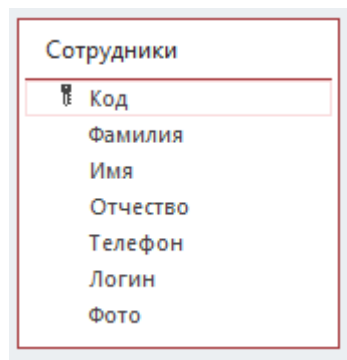


Рисунок 4 - Схема таблицы Сотрудники

Для входа в приложение, сотруднику необходимо авторизоваться в системе. Для этого создана таблица, которая хранит в себе логины и пароли работников приложения. Для дополнительной защиты пароли захешированы при помощи алгоритма MD5. Схема таблицы «Пароли» представлена на рисунке 5.

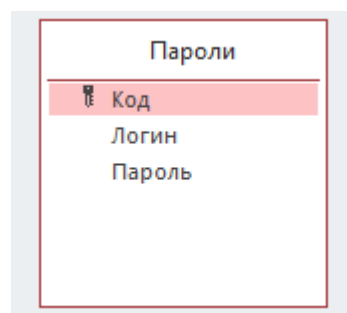


Рисунок 5 - Схема таблицы «Пароли»

Для логичного построения базы данных так же необходим словарь данных. Словарь данных представлен в таблице 1.

База данных строится в СУБД MS Access. В этой СУБД несложно построить достойную базу данных, которую можно без особых трудностей подключить к программному продукту.

Таблица 1 - Словарь данных

Объявления			
Ключ	Поле	Обязательное	Примечание
Первичный	Код	Да	Счетчик
	КраткоеОписание	Да	Текстовое
	ТипОбъявления	Да	Текстовое
	Площадь	Да	Текстовое

	Фото	Да	Текстовое
Сотрудники			
Ключ	Поле	Обязательное	Примечание
Первичный	Код	Да	Счетчик
	Фамилия	Да	Текстовое
	Имя	Да	Текстовое
	Отчество	Да	Текстовое
	Телефон	Да	Текстовое
	Логин	Да	Текстовое
	Фото	Да	Текстовое
Пароли			
Ключ	Поле	Обязательное	Примечание
Первичный	Код	Да	Счетчик
	Объявление	Да	Текстовое
	Сотрудник	Да	Текстовое

Access - это система управления базами данных (СУБД), то есть комплекс программ, предназначенный для хранения больших массивов данных в определенном формате (формате таблицы) и их автоматизированной обработки. С помощью Access можно разрабатывать удобные формы ввода и просмотра данных, составлять сложные отчеты.

Access входит в состав пакета приложений MS Office, который работает под управлением операционной системы Windows, поэтому в Access все операции выполняются по одним правилам.

Продукт корпорации Microsoft - Access объединяет сведения из разных источников в одной реляционной базе данных. Создаваемые в нем формы, запросы и отчеты позволяют быстро и эффективно обновлять данные, получать ответы на вопросы, осуществлять поиск нужных данных, анализировать данные, печатать отчеты, диаграммы. В системе адресации - в таблице БД адресуется и обрабатывается текущая запись со всеми полями, после всего ее полям нужно

задать тип вводимых данных, далее сначала в памяти создается своя папка, а затем в нем автоматически сохраняются все ее таблицы, формы, запросы.

Access - это реляционная СУБД, то есть может содержать несколько связанных между собой отношениями таблиц БД, что помогает упростить структуру данных, исключить их дублирование и облегчить выполнение работы.

Реляционная база данных – это набор данных с predetermined связями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке – значение атрибута. Каждая строка таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей. К этим данным можно получить доступ многими способами, и при этом реорганизовывать таблицы БД не требуется. База данных для программного продукта так же является реляционной. Большая часть всех баз данных являются реляционными, так как этот вид организации информации удобен и прост в использовании. База данных для ИС «Риэлторское агентство» так же является реляционной.

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

Интегрированная среда разработки — система программных средств, используемая программистами для разработки программного обеспечения (ПО). Обычно среда разработки включает в себя:

- текстовый редактор
- компилятор и/или интерпретатор
- средства автоматизации сборки
- отладчик.

История создания Visual Studio берет свое начало с 1997 года. Visual Studio 97 — первая выпущенная версия Visual Studio, в ней впервые были собраны вместе различные средства разработки ПО. Она была выпущена в двух версиях Professional и Enterprise. Visual Studio 97 — была первой попыткой Microsoft создать единую среду для разработки на разных языках программирования: Visual C++, Visual J++, Visual InterDev, и MSDN использовали одну среду, называемую Developer Studio.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

2. ПРАКТИЧЕСКИЙ РАЗДЕЛ

2.1 Создание базы данных

База данных создана по ER- модели, представленной на рисунке 2. Для каждой сущности создана своя таблица.

В каждой из таблиц, при помощи конструктора заданы имена полей, которые помогают детально раскрыть ее смысл. Все поля названы логически и избавлены от многозначности. С помощью схемы данных установлена логически построенная схема базы данных.

В итоге получили целостную базу данных, которая удовлетворяют всем требованиям программного продукта.

2.2 Создание программного продукта

Первым этапом является определение цели программного продукта. Данный программный продукт ориентирован на работника предприятия, поэтому в ходе разработки были реализованы форма авторизации и рабочая форма.

1) Форма авторизации. Данная форма содержит текстовые поля для ввода логина и пароля, а также кнопку регистрации, при нажатии на которую произойдет появление панели с регистрацией пользователя. Внешний вид формы представлен на рисунке 6. На рисунке 7 представлен вид панели регистрации.

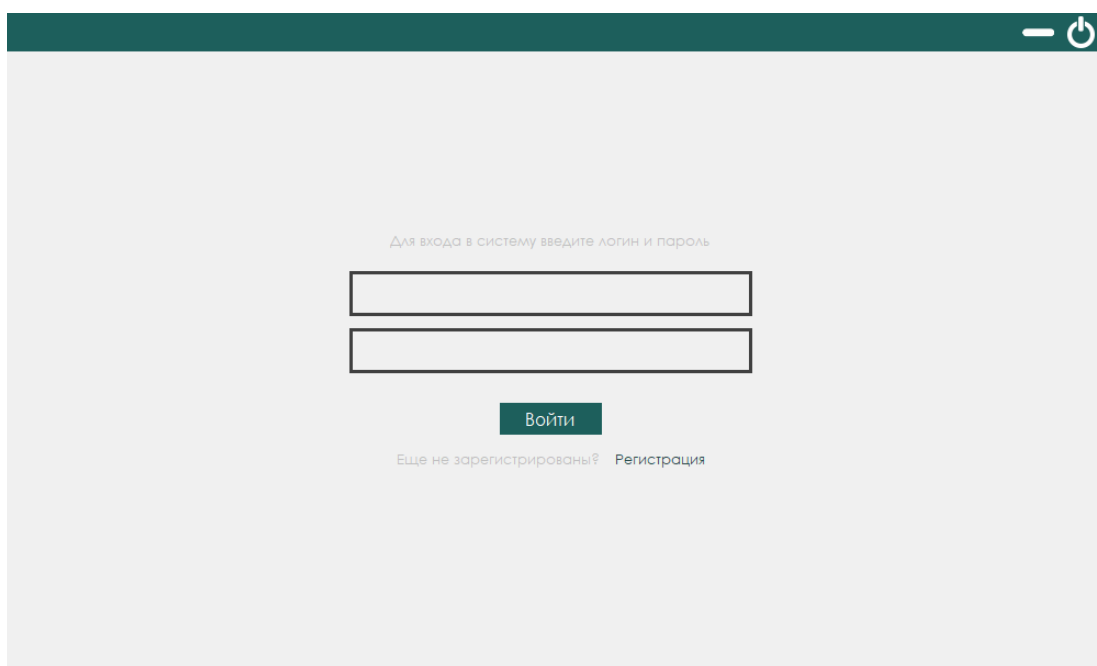


Рисунок 6 – Форма авторизации

Рисунок 7 – Внешний вид панели регистрации

При нажатии на регистрацию, произойдет отправка 5-ти значного кода подтверждения на электронный адрес, указанный в текстовом поле. После ввода верного кода произойдет отправка новой информации в базу данных и закрытие панели регистрации. После этих действий будет доступна авторизация с новыми данными.

2) Главная форма. На данной форме будут происходить все остальные действия работника. Внешний вид главной формы представлен на рисунке 8.

Код	Описание	Тип
1	Квартира на съем от 1 недели. Все удобства: санузел, газ, электричество. Адрес: Соловьиная роща, 11...	Снятие
2	Квартира на съем от 1 недели. Все удобства: санузел, газ, электричество. Адрес: Соловьиная роща, 11...	Снятие
3	Квартира на съем от 1 недели. Все удобства: санузел, газ, электричество. Адрес: Соловьиная роща, 11...	Снятие
4	sadfasdfadfafddsfafasdfas	Продажа

Рисунок 8 – Главная рабочая форма

Главная рабочая форма, как и форма авторизации, имеет вместо стандартной панели Windows пользовательскую, со своими кнопками завершения программы, сворачивания окна и кнопки «Гамбургер», которая разворачивает и сворачивает боковое меню. Внешний вид бокового меню представлен на рисунке 9. Код кнопки «Гамбургер» представлен на рисунке 10.

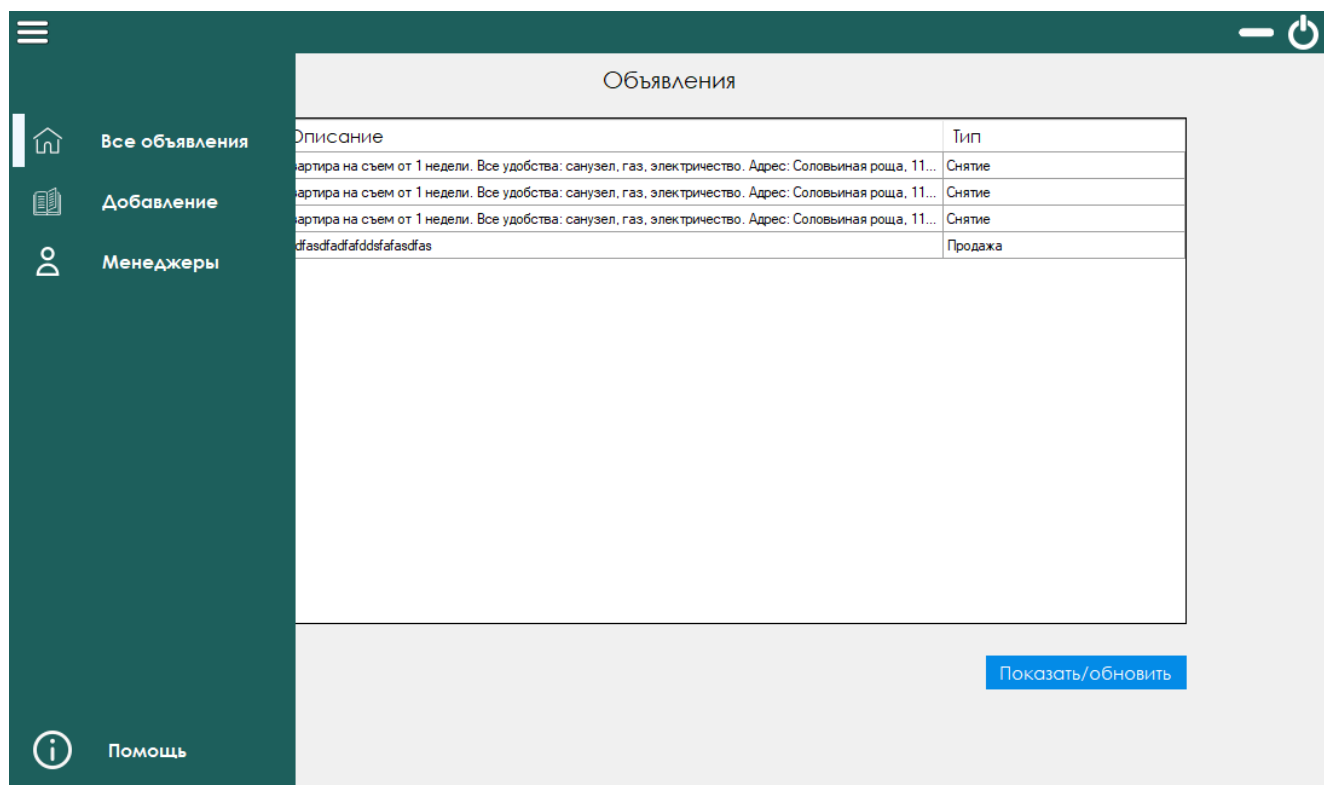


Рисунок 9 – Внешний вид бокового меню

```

ссылка:1
private void menuButton_Click(object sender, EventArgs e)
{
    if (menuPanel.Visible == false)
       unifuTransition1.ShowSync(menuPanel);
    else if (menuPanel.Visible == true)
       unifuTransition1.HideSync(menuPanel);
}

```

Рисунок 10 – Код кнопки «Гамбургер»

Боковое меню имеет кнопки «Все объявления», «Добавление», «Менеджеры» и «Помощь». При нажатии на них происходит открытие соответствующих пользовательских элементов управления. Также на боковом меню для удобства присутствует панель белого цвета, которая располагается рядом с последней нажатой кнопкой. При выборе любого из пунктов, боковое меню автоматически сворачивается. При нажатии на кнопку «Все объявления» происходит открытие пользовательского элемента управления «Объявления». Он

открывается по умолчанию при переходе на главную рабочую форму. На данном элементе расположен DataGridView, который выводит на экран информацию обо всех имеющихся объявлениях. При двойном клике на любом из объявлений произойдет открытие панели с подробным содержанием объявления. Внешний вид панели представлен на рисунке 11.

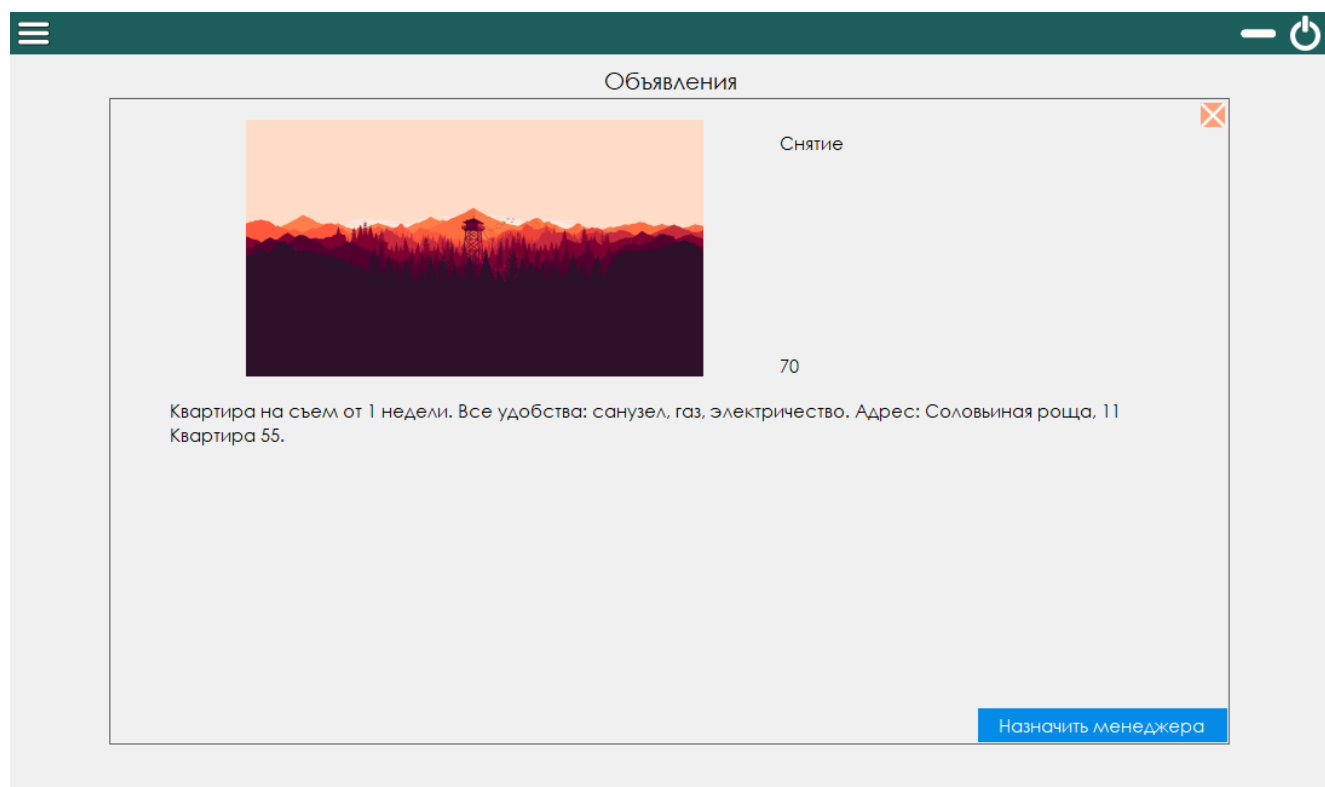


Рисунок 11 – Панель подробного просмотра объявления

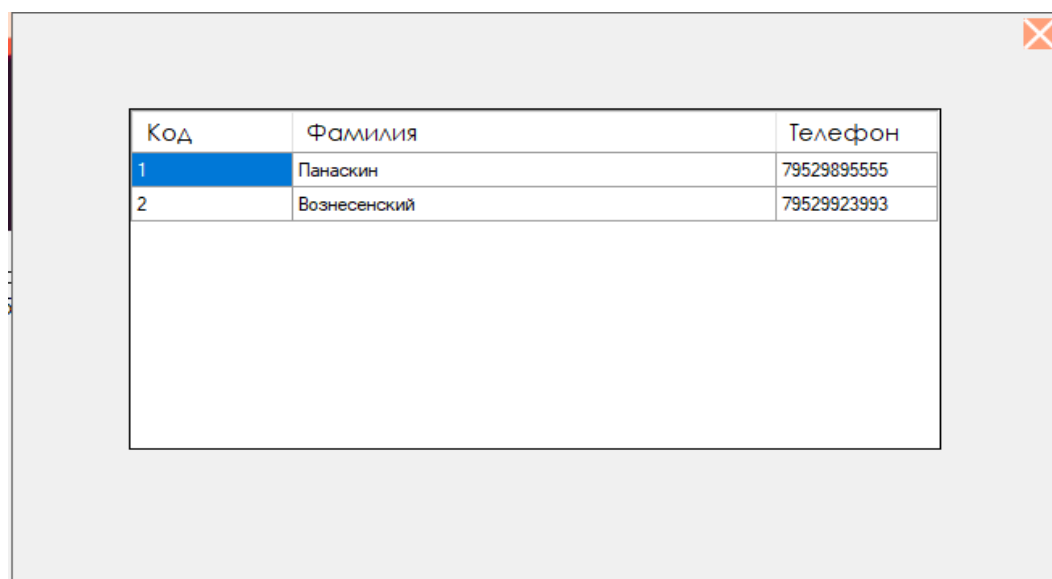


Рисунок 12 – Панель выбора менеджера

На этой панели присутствует кнопка «Назначить менеджера». При нажатии на нее откроется окно с выбором менеджера, который будет обрабатывать данное

объявление с клиентом. Внешний вид окна представлен на рисунке 12. При двойном клике на менеджера произойдет сохранение выбранной информации в базе данных.

При выборе пункта меню «Добавление» произойдет открытие пользовательского элемента управления «Добавление объявления», внешний вид которого представлен на рисунке 13.

Рисунок 13 – Элемент управления «Добавление объявления»

```
private void addPhoto_Click(object sender, EventArgs e)
{
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.InitialDirectory = "c:\\\\";
        openFileDialog.Filter = "png files (*.png)|*.png|jpg files (*.jpg)|*.jpg";
        openFileDialog.FilterIndex = 2;
        openFileDialog.RestoreDirectory = true;

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            //Get the path of specified file
            filePath = openFileDialog.FileName;
        }
        photoBox.ImageLocation = filePath;
        photoBox.BorderStyle = BorderStyle.None;
    }
}
```

Рисунок 14 – Код кнопки прикрепления фотографии

На этом элементе управления присутствуют поля для ввода информации об объявлении: описание, площадь, тип объявления, прикрепление фотографии. Код кнопки прикрепления фотографии представлен на рисунке 14.

При выборе пункта меню «Менеджеры» произойдет открытие пользовательского элемента управления «Менеджеры», внешний вид которого представлен на рисунке 15.

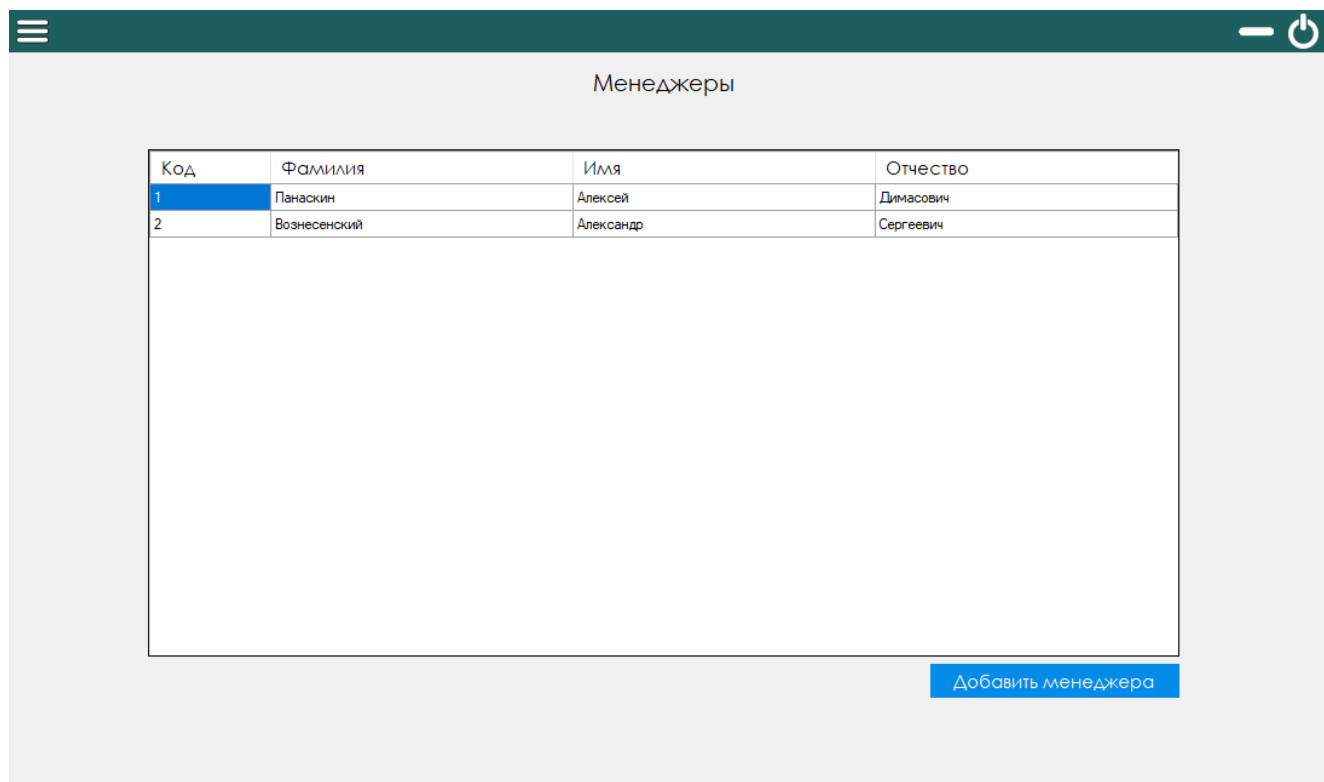


Рисунок 15 – Пользовательский элемент управления «Менеджеры»

На этом пользовательском элементе управления также присутствует DataGridView, при помощи которого на экран выводится информация обо всех менеджерах. При двойном клике на любого менеджера произойдет открытие панели с подробной информацией о выбранном менеджере. Внешний вид панели представлен на рисунке 16. При нажатии на кнопку «Добавить менеджера» произойдет открытие этой же панели, но с пустыми текстовыми полями, в которые можно внести информацию о новом менеджере. Внешний вид пустой панели представлен на рисунке 17. На панели подробной информации есть возможность внести обновленную информацию о менеджере. При нажатии на кнопку «Сохранить» произойдет обновление информации в базе данных.

Менеджеры

Фамилия: Панаскин

Имя: Алексей

Отчество: Димасович

Добавить фото

Телефон: 79529895555

Логин: alexpoz@gmail.com

Сохранить

Рисунок 16 – Панель подробной информации о менеджере

Менеджеры

Фамилия:

Имя:

Отчество:

Добавить фото

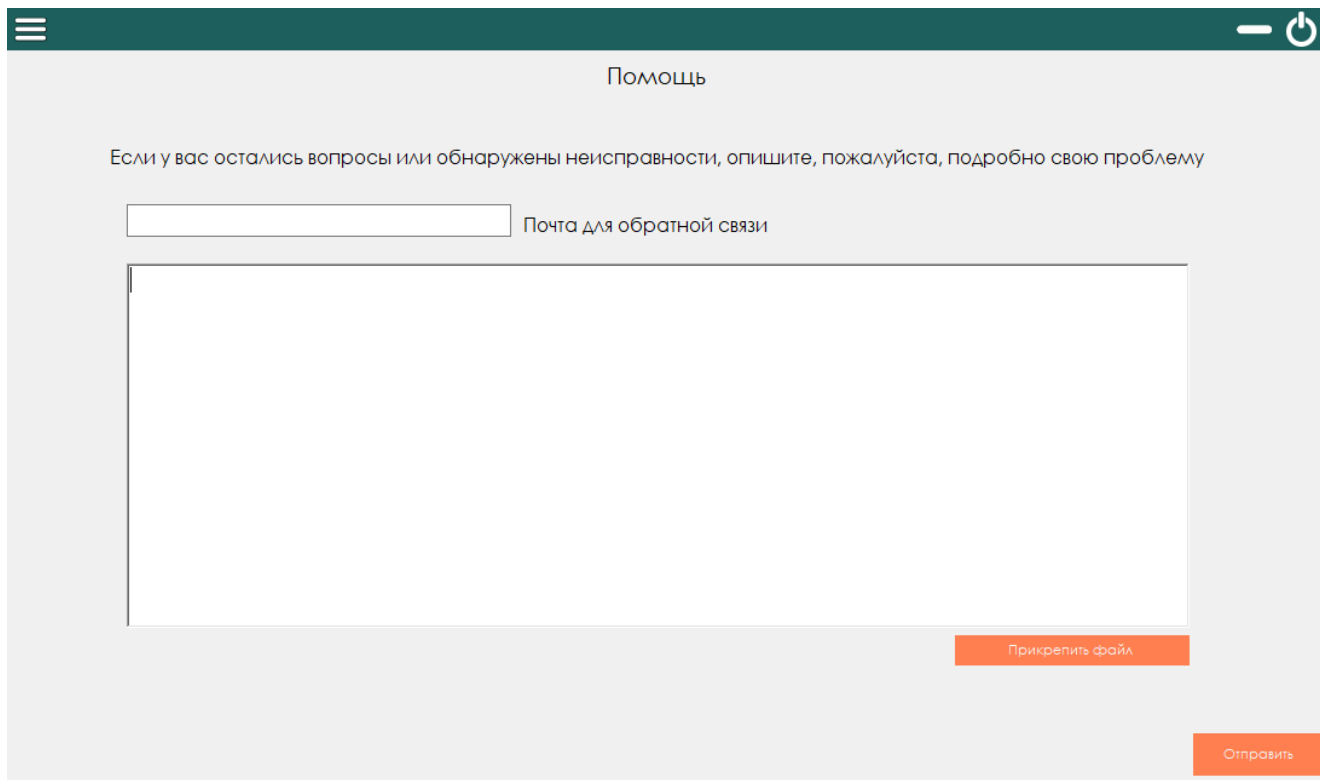
Телефон:

Логин:

Сохранить

Рисунок 17 – Пустая панель подробной информации

При выборе последнего пункта меню под названием «Помощь», произойдет открытие пользовательского элемента помощь, внешний вид которого представлен на рисунке 18. Он имеет текстовые поля для ввода сообщения и почты для обратной связи. Так же существует возможность прикрепить файл к отправляемому сообщению. Код отправки сообщения представлен на рисунке 19.



Помощь

Если у вас остались вопросы или обнаружены неисправности, опишите, пожалуйста, подробно свою проблему

Почта для обратной связи

Прикрепить файл

Отправить

Рисунок 18 – Пользовательский элемент управления «помощь»

```
MailMessage message1 = new MailMessage("newhouse.bonch@bk.ru", "voznenskiy.shura@gmail.com",  
SmtpClient client = new SmtpClient("smtp.mail.ru");  
client.Port = 587;  
client.Credentials = new System.Net.NetworkCredential("newhouse.bonch@bk.ru", "lZb43rQv27");  
client.EnableSsl = true;  
if (filePath != string.Empty)  
{  
    Attachment data = new Attachment(filePath, MediaTypeNames.Application.Octet);  
    ContentDisposition disposition = data.ContentDisposition;  
    disposition.CreationDate = File.GetCreationTime(filePath);  
    disposition.ModificationDate = File.GetLastWriteTime(filePath);  
    disposition.ReadDate = File.GetLastAccessTime(filePath);  
    message1.Attachments.Add(data);  
}  
client.Send(message1);
```

Рисунок 19 – Код отправки сообщения на электронную почту

2.3 Тестирование программного продукта

Тестирование данного программного продукта происходило посредством модульного тестирования. Тестировались такие модули, как: «Менеджеры», «Объявления» и главная рабочая форма.

ID 1	Тестирование «Менеджеры»	
Автор: Вознесенский Александр Сергеевич		Приоритет - высокий
1. Ввод данных в базу данных на элементе «Менеджеры»	Ввод произведен	Ввод произведен
2. Изменение данных в базе данных на элементе «Менеджеры»	Изменение произведено	Изменение произведено
3. Удаление данных в базе данных на элементе «Менеджеры»	Удаление произведено	Удаление не произведено – ошибка в построении SQLзапроса
Статус: не пройден		
ID 2	Тестирование «Объявления»	
Автор: Вознесенский Александр Сергеевич		Приоритет - высокий
1. Ввод данных в базу данных на элементе «Объявления»	Ввод произведен	Ввод произведен
2. Изменение данных в базе данных на элементе «Объявления»	Изменение произведено	Изменение произведено
3. Удаление данных в базе данных на элементе «Объявления»	Удаление произведено	Удаление произведено
Статус: пройден		
ID 3	Тестирование «Главная рабочая форма»	
Автор: Вознесенский Александр Сергеевич		Приоритет - высокий

1. Ввод данных в базу данных на элементе «Главная рабочая форма»	Ввод произведен	Ввод произведен
2. Изменение данных в базе данных на элементе «Главная рабочая форма»	Изменение произведено	Изменение не произведено. Возникла исключительная ситуация «неверный тип данных»
3. Удаление данных в базе данных на элементе «Главная рабочая форма»	Удаление произведено	Удаление произведено
Статус: не пройден		

В ходе тестирования информационной системы были выявлены и исправлены ошибки на главной рабочей форме и на пользовательском элементе «Менеджеры».

3. ОХРАНА ТРУДА

1. Работать на персональном компьютере может лицо, которое прошло первичный и вводный инструктажи по охране труда на рабочем месте, прошло контроль знаний по требованиям охраны труда и не имеющее противопоказаний по состоянию здоровья к данной деятельности;

2. В помещениях, где проводятся работы на ПК, необходимо создать оптимальные условия зрительной работы. Основной поток естественного света должен быть слева, солнечные лучи и блики не должны попадать в поле зрения работающего и на экраны видеомониторов.

3. Общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении ПК. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору.

4. Для обеспечения нормируемых значений освещенности в помещениях использования ПК следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

5. Монитор ПК должен находиться на расстоянии 50-70 см от глаз оператора и иметь антибликовое покрытие. Покрытие должно также обеспечивать снятие электростатического заряда с поверхности экрана, исключать искрение и накопление пыли.

6. Нельзя загораживать заднюю стенку системного блока или ставить ПК вплотную к стене, это приводит к нарушению охлаждения системного блока.

7. Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

8. Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

9. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

10. Режим работы и отдыха должен зависеть от характера выполняемой работы. При вводе данных, редактировании программ, считывании информации с экрана непрерывная продолжительность работы с ПК не должна превышать 4 часа за рабочий день при 8-ми часовом рабочем дне. Через каждый час работы необходимо делать перерывы на отдых по 5-10 минут или по 15-20 минут каждые два часа работы.

11. Рабочее место должно быть оборудовано так, чтобы исключать неудобные позы и длительные статические напряжения тела.

12. Запрещается оставлять без присмотра включенное оборудование; вскрывать устройства ПК.

13. Не допускается прием еды и воды за персональным компьютером;

14. По завершению работы за компьютером нужно убедиться, что персональный компьютер отключен от электрической сети и привести свое рабочее место в надлежащий вид.

ЗАКЛЮЧЕНИЕ

При создании информационной системы в рамках курсовой работы была достигнута поставленная цель, а именно: создать программный продукт, в котором реализована оптимизация работы сотрудника компании.

Для достижения цели были решены следующие задачи:

1. Составлен полный список эксплуатационных и функциональных требований к данному продукту
2. Составлена ER – модель программного продукта, словарь данных для таблиц базы данных
3. Реализована информационная система в выбранной среде программирования
4. Реализованы этапы тестирования
5. Организована документация к программному продукту

В данном программном продукте планируется доработать сжатие и восстановление базы данных напрямую в коде для правильной работы счетчиков в таблицах и исключении ошибок. Также планируется добавить поиск по менеджерам.

Данный программный продукт является практически значимым на рынке программного обеспечения, так как он имеет более удобный пользовательский интерфейс и удобную обработку информации.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 266 с.
2. Дубнов, П.Ю. Access 2000. Проектирование баз данных / П.Ю. Дубнов. - М.: ДМК, 2015. - 272 с.
3. Дунаев, В. В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2017. - 288 с.
4. Послед, Б.С. Access 2000. Базы данных и приложения. Лекции и упражнения / Б.С. Послед. - К.: DiaSoft, 2017. - 512 с.
5. Прайс, Джейсон Oracle Database 11g: SQL. Операторы SQL и программы PL/SQL / Джейсон Прайс. - М.: ЛОРИ, 2018. - 660 с.
6. Хаббард, Дж. Автоматизированное проектирование баз данных / Дж. Хаббард. - М.: Мир, 2014. - 296 с.
7. Энсор Oracle проектирование баз данных / Энсор, Стивенсон Дейв; , Йен. - М.: BHV Киев, 2014. - 560 с.

ПРИЛОЖЕНИЕ

Листинг для переключения элементов управления на главной форме

```
private void allButton_Click(object sender, EventArgs e)
{
    allManagers1.Visible = false;
    allTasks1.Visible = true;
    helpForm1.Visible = false;
    addTask1.Visible = false;
    sidePanel.Height = allButton.Height;
    sidePanel.Top = allButton.Top;
    bunifuTransition1.HideSync(menuPanel);
}

ссылка: 1
private void addButton_Click(object sender, EventArgs e)
{
    allTasks1.Visible = false;
    allManagers1.Visible = false;
    helpForm1.Visible = false;
    addTask1.Visible = true;
    sidePanel.Height = addButton.Height;
    sidePanel.Top = addButton.Top;
    bunifuTransition1.HideSync(menuPanel);
}

ссылка: 1
private void managersButton_Click(object sender, EventArgs e)
{
    allTasks1.Visible = false;
    helpForm1.Visible = false;
    addTask1.Visible = false;
    allManagers1.Visible = true;
    sidePanel.Height = managersButton.Height;
    sidePanel.Top = managersButton.Top;
    bunifuTransition1.HideSync(menuPanel);
}
```

Функция заполнения из базы данных

```
public void zapol()
{
    //заполнение бд
    myConnection.Open();
    dataGridView1.Rows.Clear();
    string req1 = "SELECT * FROM Клиенты ORDER BY Код_клиента";
    OleDbCommand command1 = new OleDbCommand(req1, myConnection);
    OleDbDataReader reader1 = command1.ExecuteReader();
    List<string[]> data_1 = new List<string[]>();
    while (reader1.Read())
    {
        data_1.Add(new string[7]);
        data_1[data_1.Count - 1][0] = reader1[0].ToString();
        data_1[data_1.Count - 1][1] = reader1[1].ToString();
        data_1[data_1.Count - 1][2] = reader1[2].ToString();
        data_1[data_1.Count - 1][3] = reader1[3].ToString();
        data_1[data_1.Count - 1][4] = reader1[4].ToString();
        data_1[data_1.Count - 1][5] = reader1[5].ToString();
        data_1[data_1.Count - 1][6] = reader1[6].ToString();
    }
    reader1.Close();

    foreach (string[] s in data_1)
    {
        dataGridView1.Rows.Add(s);
    }
    myConnection.Close();
}
```