

## **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ .....	
1.ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ.....	
1.1 Постановка задачи и определение предметной области .....	
1.2 Построение базы данных программного продукта .....	
2.ПРАКТИЧЕСКИЙ РАЗДЕЛ .....	
2.1 Создание базы данных .....	
2.2 Создание программного продукта.....	
3.ОХРАНА ТРУДА .....	
ЗАКЛЮЧЕНИЕ .....	
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	
ПРИЛОЖЕНИЕ .....	

## **ВВЕДЕНИЕ**

В настоящее время необходимо хранить большое количество информации. Для этого нужно использовать современные и функциональные базы данных. Они позволяют автоматизировать поиск по различным критериям, сортировку, форматирование и добавление новых данных.

Данный программный продукт значим и актуален в настоящее время, так как он позволяет оптимизировать процесс работы сотрудников приемной комиссии учебного заведения за счет удобного графического интерфейса, полного использования базы данных и оптимизации использования ресурсов аппаратного обеспечения.

В процессе выбора темы и разработки программного продукта была поставлена цель - разработка и реализация информационной системы «Приемная комиссия», которая позволит автоматизировать деятельность работника приемной комиссии.

Для достижения поставленной цели, необходимо выполнить следующие задачи:

- 1) Составить полный список эксплуатационных и функциональных требований к данному программному продукту.
- 2) Составить ER – модель программного продукта, словарь данных для таблиц базы данных.
- 3) Реализовать информационную систему в выбранной среде программирования.
- 4) Реализовать этапы тестирования.
- 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 «Приемная комиссия», которая будет включать в себя базу данных с информацией об абитуриентах и пользователях программы, и пользовательский интерфейс для работы с данной базой данных.

Приемная комиссия – отдел образовательного учреждения, организующий сбор, хранение и обработку информации об абитуриентах, их личных данных и фото. Правильно организованная работа приемной комиссии облегчает работнику приемной комиссии – более эффективное выполнение его работы.

Данная информационная система будет позволять обрабатывать информацию об абитуриентах, добавлять, редактировать и удалять записи из базы данных. Данные обо всех объектах базы данных хранятся в таблицах, которые создаются в формах Visual Studio. Информация об абитуриентах и фото их документов будут храниться в соответствующих таблицах и папках. Получение информации об абитуриентах и их документах будет организовываться путем создания и выполнения запросов на языке SQL.

Ввод данных в базу данных должен осуществляться несколькими способами. Первый – ввод данных в базу данных вручную. Второй способ – ввод данных в формах Visual Studio. Простые данные, например данные о документах, можно вносить в базу данных вручную, но для большего удобства лучше использовать внос через пользовательское приложение.

Работа приемной комиссии схематично показана на рисунке 1.



Рисунок 1 - Схема работы приемной комиссии

## 1.2 Построение базы данных программного продукта

Правильно и логически построенная база данных позволит ускорить создание самого программного продукта. Основными сущностями базы данных являются: абитуриенты, их паспорт и аттестат.

Для начала необходимо построить ER – модель базы данных, на которой будет показано взаимодействие основных сущностей базы данных. Схема взаимодействия представлена на рисунке 2.

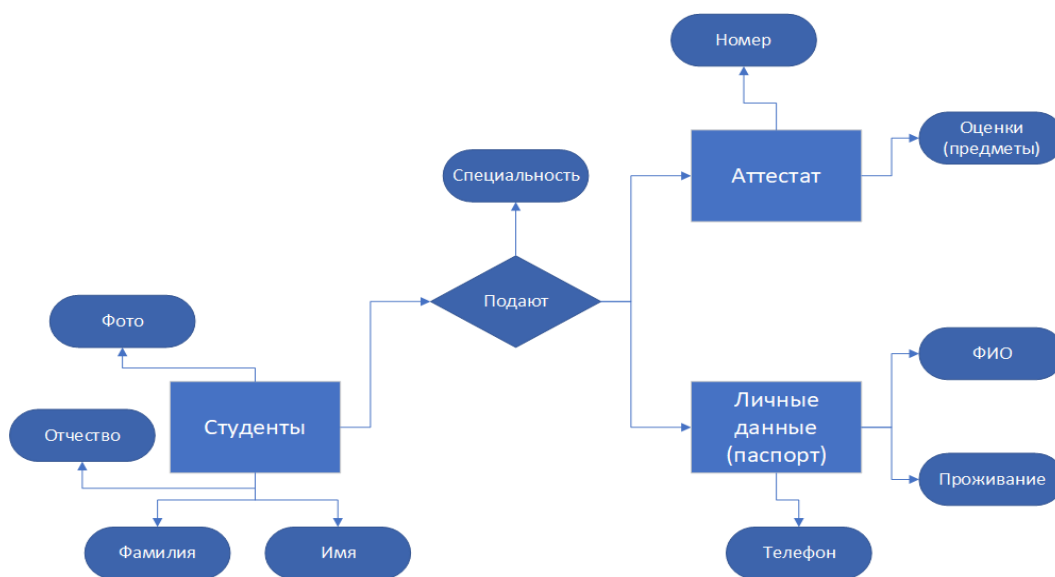


Рисунок 2 - ER – модель базы данных

Построение таблиц базы данных начинается с основной темы программного продукта – абитуриенты. Таблица Абитуриенты будет содержать в себе информацию об абитуриенте, его основные данные и ссылку на фото. Схема таблицы Абитуриенты представлена на рисунке 3.

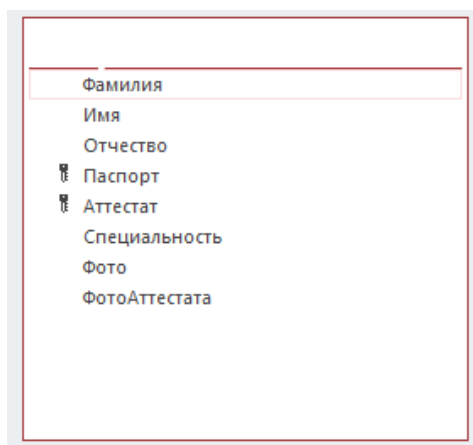
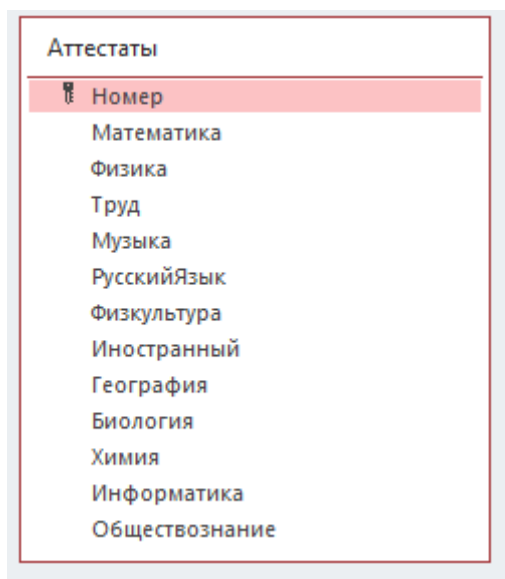


Рисунок 3 - Схема таблицы Абитуриенты

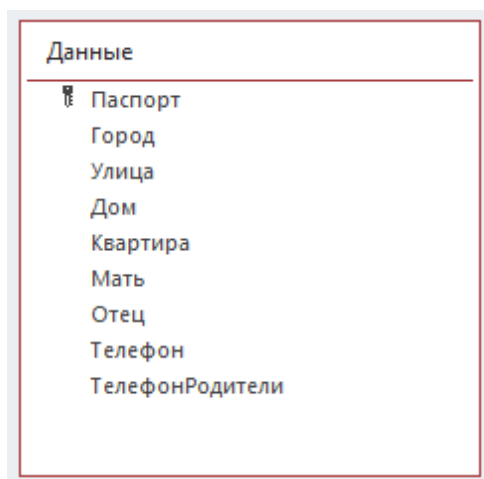
Следующая реляция базы данных – таблица Аттестаты. В этой таблице будет содержаться информация о аттестате, оценки по предметам. Схема таблицы Аттестаты представлена на рисунке 4.



№	Номер
	Математика
	Физика
	Труд
	Музыка
	РусскийЯзык
	Физкультура
	Иностранный
	География
	Биология
	Химия
	Информатика
	Обществознание

Рисунок 4 - Схема таблицы Аттестаты

Личная информация об абитуриенте содержится в таблице Данные. Она имеет такие поля как адрес проживания, телефон, ФИО родителей и т.д. Схема таблицы Данные представлена на рисунке 5.



№	Паспорт
	Город
	Улица
	Дом
	Квартира
	Мать
	Отец
	Телефон
	ТелефонРодители

Рисунок 5 - Схема таблицы Данные

Окончательная схема базы данных представлена на рисунке 6. 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.

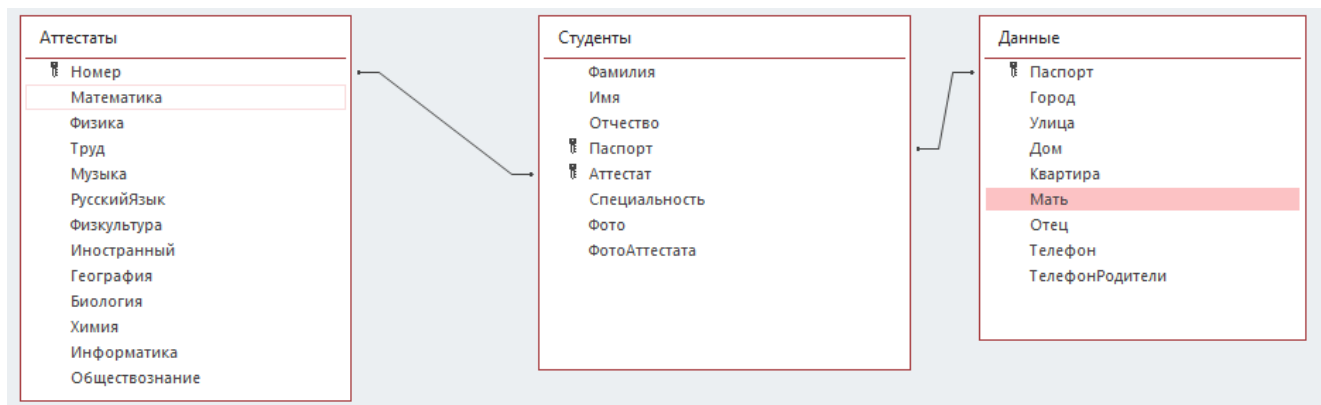


Рисунок 6 - Схема базы данных

Таблица 1 - Словарь данных

Студенты			
Ключ	Поле	Обязательное	Примечание
	Фамилия	Да	Текстовое
	Имя	Да	Текстовое
	Отчество	Да	Текстовое
Первичный	Паспорт	Да	Текстовое
Первичный	Аттестат	Да	Текстовое
	Специальность	Да	Текстовое
	Фото	Да	Текстовое
	ФотоАттестата	Да	Текстовое
Данные			
Ключ	Поле	Обязательное	Примечание
Первичный	Паспорт	Да	Текстовое
	Город	Да	Текстовое
	Улица	Да	Текстовое
	Дом	Да	Текстовое
	Квартира	Да	Текстовое
	Мать	Нет	Текстовое
	Отец	Нет	Текстовое

	Телефон	Да	Текстовое
	ТелефонРодители	Нет	Текстовое
<b>Аттестаты</b>			
<b>Ключ</b>	<b>Поле</b>	<b>Обязательное</b>	<b>Примечание</b>
Первичный	Номер	Да	Текстовое
	Математика	Да	Текстовое
	Физика	Да	Текстовое
	Труд	Да	Текстовое
	Музыка	Да	Текстовое
	РусскийЯзык	Да	Текстовое
	Физкультура	Да	Текстовое
	Иностранный	Да	Текстовое
	География	Да	Текстовое
	Биология	Да	Текстовое
	Химия	Да	Текстовое
	Информатика	Да	Текстовое
	Обществознание	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.

База данных для данной информационной системы «Приемная комиссия» построена по принципу реляционной модели. Вся информация хранится в таблицах (Данные, Абитуриенты, Аттестаты), которые состоят из необходимых полей и имеют свои атрибуты.

## 2. ПРАКТИЧЕСКИЙ РАЗДЕЛ

### 2.1 Создание базы данных

База данных создана по схеме, представленной на рисунке 6. 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.

На выходе получилась целостная база данных, которая удовлетворяет всем требованиям программного продукта.

### 2.2 Создание программного продукта

Изначально необходимо определиться с целью программного продукта. Программный продукт ориентирован на работу сотрудника приемной комиссии с информационной системой, поэтому программный продукт содержит в себе:

1) Форма инициализации. На данной форме имеются два текстовых поля для ввода логина и пароля соответственно. Пароль пользователя шифруется и в базе данных хранится только md5-сумма пароля. Имеется кнопка входа, в которой проверяется правильность введенных данных. Так же имеется кнопка выхода из программы и кнопка сворачивания окна. Стандартная панель Windows убрана и вместо нее сделана пользовательская панель. Общий вид формы входа представлен на рисунке 7.

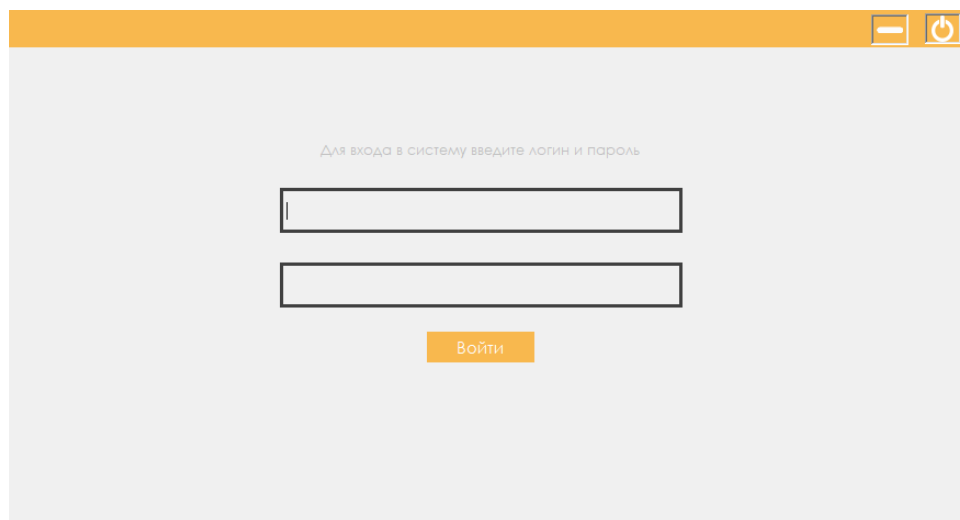


Рисунок 7 - Общий вид формы входа



При вводе правильного логина и пароля происходит переход на рабочую форму приложения. Код перехода на следующую форму представлен на рисунке 8:

```
private void enterButton_Click(object sender, EventArgs e)
{
    myConnection = new OleDbConnection(connectstring);
    myConnection.Open();
    string login = studentLogin_input.Text;
    string password = GetHash(studentPassword_input.Text);

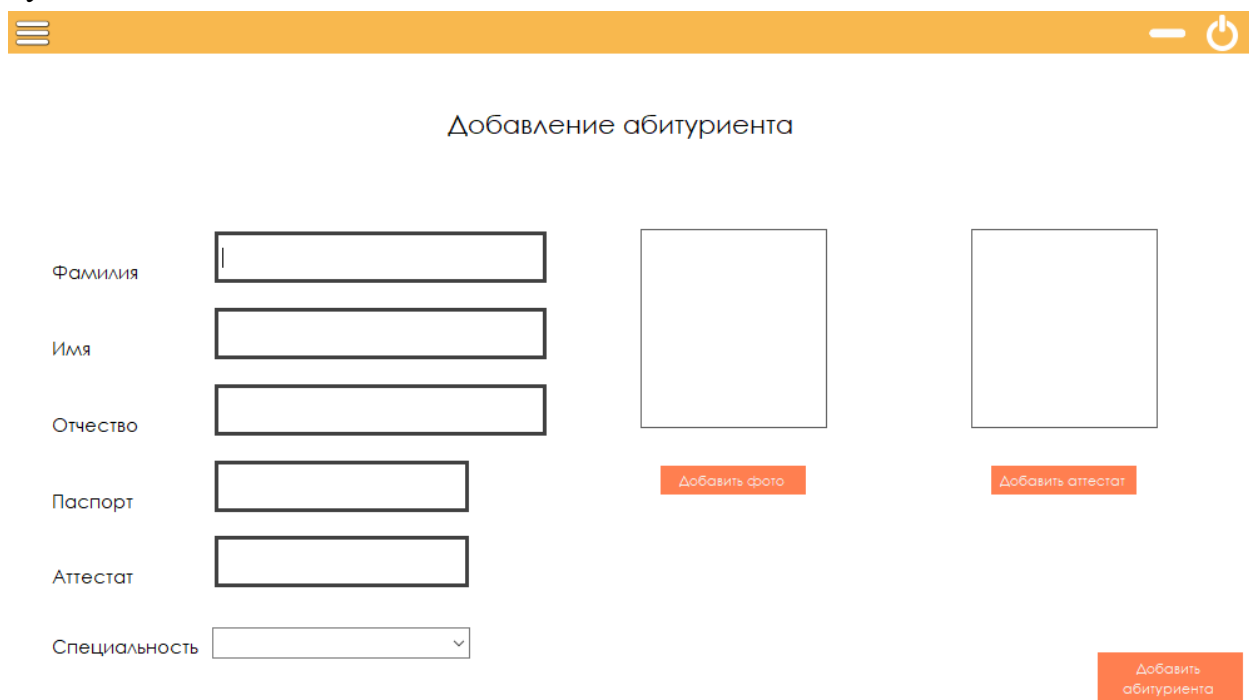
    string query = $"SELECT Логин, Пароль FROM Пароли WHERE Логин = '{login}'";
    OleDbCommand command = new OleDbCommand(query, myConnection);
    OleDbDataReader reader = command.ExecuteReader();
    string loginFromDatabase = "";
    string passwordFromDatabase = "";
    while (reader.Read())
    {
        loginFromDatabase = reader[0].ToString();
        passwordFromDatabase = reader[1].ToString();
    }
    reader.Close();

    if (login == "" || password == "")
    {
        MessageBox.Show("Вы не ввели логин или пароль");
        myConnection.Close();
    }
    else if (login == loginFromDatabase && password == passwordFromDatabase)
    {
        MainForm newf = new MainForm();
        newf.Show();
        this.Hide();
        myConnection.Close();
    }
    else
    {
        MessageBox.Show("Неверный логин или пароль");
        myConnection.Close();
    }
    myConnection.Close();
}
```

Рисунок 8 - Вход в приложение

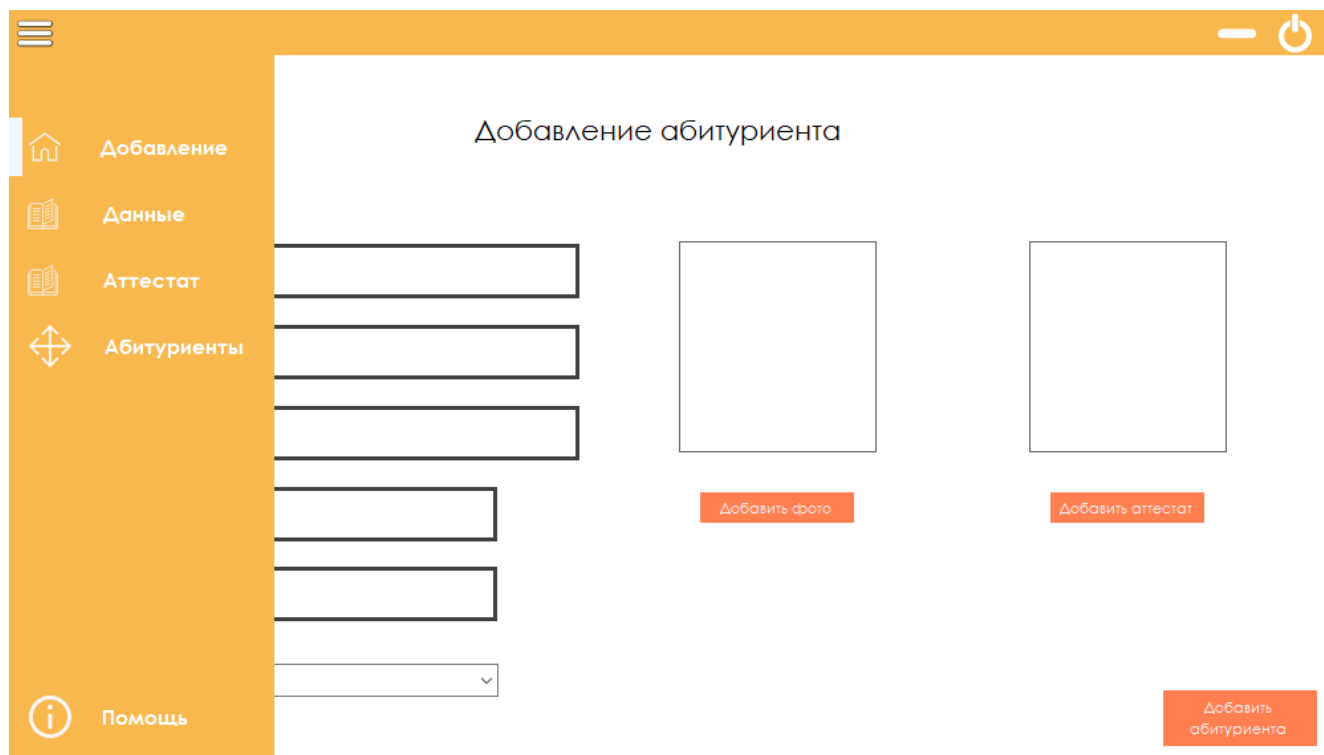
2) Рабочая форма. При открытии рабочей формы, представленной на рисунке 9, открывается первый пользовательский элемент управления «Добавление абитуриента». Этот элемент имеет несколько полей для ввода, таких как Фамилия, Имя, Отчество, Паспорт и т.д. Так же на этом элементе управления присутствует возможность прикрепить фотографии паспорта и аттестата к личному делу абитуриента. Рабочая форма имеет пользовательскую панель управления вместо стандартной панели Windows, на которой расположено 3 кнопки. В правом верхнем углу расположена кнопка выхода, при нажатии на которую выпадает панель с подтверждением выхода. Там же находится кнопка свернуть, при нажатии на которую происходит сворачивание окна приложения. В левом верхнем углу

находится кнопка меню «гамбургер», которая разворачивает и сворачивает меню программы. Меню приложения показано на рисунке 10. Кнопка меню «Гамбургер» разворачивает и сворачивает боковое меню при нажатии на нее. При выборе любого элемента, само боковое меню автоматически сворачивается. Код сворачивания представлен на рисунке 11. Код кнопки меню представлен на рисунке 12.



The image shows a web form titled "Добавление абитуриента" (Add Applicant). At the top, there is an orange header bar containing a hamburger menu icon on the left and a minus sign and a power icon on the right. The form itself has several input fields: "Фамилия" (Surname), "Имя" (Name), "Отчество" (Patronymic), "Паспорт" (Passport), "Аттестат" (Certificate), and "Специальность" (Specialty) which is a dropdown menu. To the right of these fields are two large empty square boxes for "Добавить фото" (Add photo) and "Добавить аттестат" (Add certificate). At the bottom right, there is a red button labeled "Добавить абитуриента" (Add applicant).

Рисунок 9 - Рабочая форма и первый элемент управления



This image shows the same "Добавление абитуриента" form as in Figure 9, but with the side menu expanded. The side menu is an orange vertical bar on the left side of the page. It contains a hamburger menu icon at the top, followed by four items: "Добавление" (Add) with a house icon, "Данные" (Data) with a document icon, "Аттестат" (Certificate) with a document icon, and "Абитуриенты" (Applicants) with a four-way arrow icon. At the bottom of the menu is "Помощь" (Help) with an information icon. The main content area of the form is visible behind the menu, showing the input fields and buttons as described in Figure 9.

Рисунок 10 - Выпадающее меню приложения

```

addNewUser1.Visible = false;
studentInfo1.Visible = false;
attestatInfo1.Visible = false;
allStudents1.Visible = true;
helpForm1.Visible = false;
sidePanel.Height = allStudents.Height;
sidePanel.Top = allStudents.Top;
bunifuTransition1.HideSync(menuPanel);

```

Рисунок 11 - Код автоматического сворачивания меню

```

if (menuPanel.Visible == false)
    bunifuTransition1.ShowSync(menuPanel);
else if (menuPanel.Visible == true)
    bunifuTransition1.HideSync(menuPanel);

```

Рисунок 12 - Кнопка меню «Гамбургер»

При нажатии на кнопку добавить абитуриента происходят несколько запросов. Первый запрос проверяет существует ли в базе данных абитуриент с таким паспортом. Если такой существует, программа выдаст уведомление об этом. Если такого нет, программа выполнит SQL-запрос на сохранение информации о новом абитуриенте.

Выпадающее меню содержит в себе 5 кнопок: Добавление, Данные, Аттестат, Абитуриенты, Помощь. По нажатию на эти кнопки открываются пользовательские элементы управления с соответствующими названиями.

По нажатию на кнопку Данные открывается пользовательский элемент управления «Личные данные». Его первоначальный вид представлен на рисунке 13. Изначально на нем присутствует всего одно поле. В нем происходит ввод номера паспорта. При нажатии на кнопку Внести происходят несколько запросов к базе данных. Первый запрос проверяет существование абитуриента с данным паспортом. Если такого не существует, его следует создать на вкладке «Добавление». Если такой существует, произойдет открытие элемента управления со всеми его полями, внешний вид которого показан на рисунке 14.

На этом элементе управления присутствуют поля для ввода таких значений, как место жительства (город, улица, дом, квартира), данные о родителях и контактные данные. При нажатии на кнопку Добавить данные, выполнится SQL-

запрос на добавление данных в таблицу. Код кнопки представлен на рисунке 15. Так же на этом элементе управления присутствует кнопка Отмена. При нажатии на нее все поля для ввода очистятся, а пользовательский элемент управления вернется к первоначальному виду, ожидая ввод номера паспорта.

The screenshot shows a web interface with an orange header bar containing a menu icon on the left and a minus and power icon on the right. The main content area is white and contains the text "Номер паспорта" above a single-line text input field. Below the input field is an orange button with the text "Внести".

Рисунок 13 - Первичное отображение пользовательского элемента «Данные»

The screenshot shows a web interface with an orange header bar containing a menu icon on the left and a minus and power icon on the right. The main content area is white and contains the text "Личные данные" at the top. Below this text, there are two columns of input fields. The left column is titled "Проживание" and contains four input fields labeled "Город", "Улица", "Дом", and "Квартира". The right column is titled "Данные" and contains four input fields labeled "Мать (ФИО)", "Отец (ФИО)", "Телефон", and "Телефон (родители)". At the bottom of the form, there are two orange buttons: "Отменить" on the left and "Добавить данные" on the right.

Рисунок 14 - Пользовательский элемент управления «Данные»

```

myConnection.Open();
try
{
    string updateInfo = $"UPDATE Данные SET " +
        $"Город = '{cityInput.Text}'," +
        $"Улица = '{streetInput.Text}'," +
        $"Дом = '{homeInput.Text}'," +
        $"Квартира = '{kvarInput.Text}'," +
        $"Мать = '{motherInput.Text}'," +
        $"Отец = '{fatherInput.Text}'," +
        $"Телефон = '{numberInput.Text}'," +
        $"ТелефонРодители = '{numberParentInput.Text}' WHERE Паспорт = '{passportInput.Text}'";
    OleDbCommand commandForPassport = new OleDbCommand(updateInfo, myConnection);
    commandForPassport.ExecuteNonQuery();
    MessageBox.Show("Данные успешно добавлены", "Успешно");
    cityInput.Text = "";
    streetInput.Text = "";
    homeInput.Text = "";
    kvarInput.Text = "";
    motherInput.Text = "";
    fatherInput.Text = "";
    numberInput.Text = "";
    numberParentInput.Text = "";
    passportInput.Text = "";
    panel1.Visible = true;
}
catch (System.Data.OleDb.OleDbException q) when (q.Message == "Отсутствует значение для одного из")
{
    MessageBox.Show("Проверьте наличие введенных данных", "Ошибка");
}
myConnection.Close();

```

Рисунок 15 - Код кнопки «Добавить данные»

```

myConnection.Open();
if (attestInput.Text == "")
{
    MessageBox.Show("Проверьте правильность и наличие введенных данных", "Ошибка");
}
else
{
    string selectPassport = $"SELECT Номер FROM Аттестаты WHERE Номер='{attestInput.Text}'";
    OleDbCommand commandForPassport = new OleDbCommand(selectPassport, myConnection);
    commandForPassport.ExecuteNonQuery();

    OleDbDataReader reader = commandForPassport.ExecuteReader();
    string attesstFromDatabase = string.Empty;
    while (reader.Read())
    {
        attesstFromDatabase = reader[0].ToString();
    }
    reader.Close();
    if (attestInput.Text == attesstFromDatabase)
    {
        panel1.Visible = false;
    }
    else
    {
        MessageBox.Show("Такого аттестата не существует, проверьте данные", "Ошибка");
    }
}
myConnection.Close();

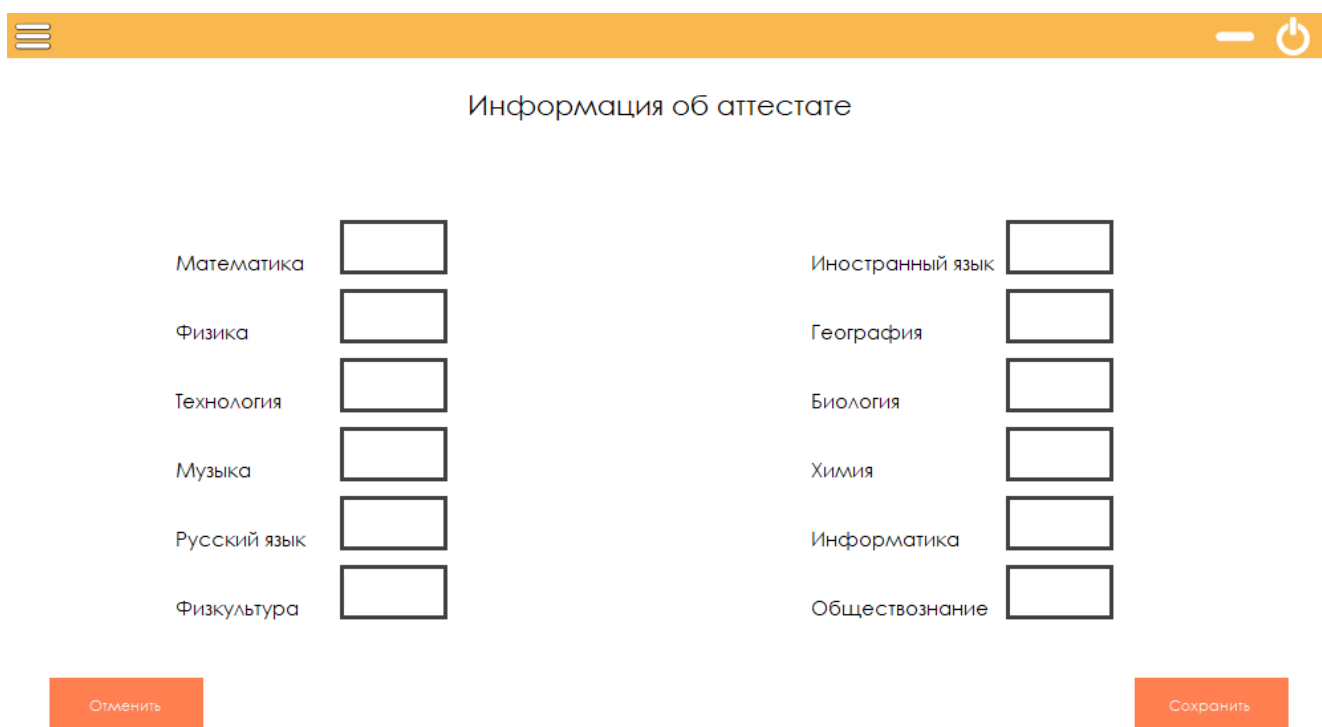
```

Рисунок 16 - Проверка существования аттестата



The screenshot shows a web application interface with an orange header bar containing a menu icon on the left and a minus and power icon on the right. The main content area is white and contains a single text input field labeled "Номер аттестата" (Certificate number). Below the input field is an orange button with the text "Внести" (Enter).

Рисунок 17 - Первичный вид элемента управления «Информация об аттестате»



The screenshot shows a web application interface with an orange header bar containing a menu icon on the left and a minus and power icon on the right. The main content area is white and contains the title "Информация об аттестате" (Certificate information). Below the title is a table with two columns of subjects and their corresponding grade input fields. At the bottom of the form are two orange buttons: "Отменить" (Cancel) on the left and "Сохранить" (Save) on the right.

Информация об аттестате			
Математика	<input type="text"/>	Иностранный язык	<input type="text"/>
Физика	<input type="text"/>	География	<input type="text"/>
Технология	<input type="text"/>	Биология	<input type="text"/>
Музыка	<input type="text"/>	Химия	<input type="text"/>
Русский язык	<input type="text"/>	Информатика	<input type="text"/>
Физкультура	<input type="text"/>	Обществознание	<input type="text"/>

Рисунок 18 - Основной вид элемента управления «Информация об аттестате»

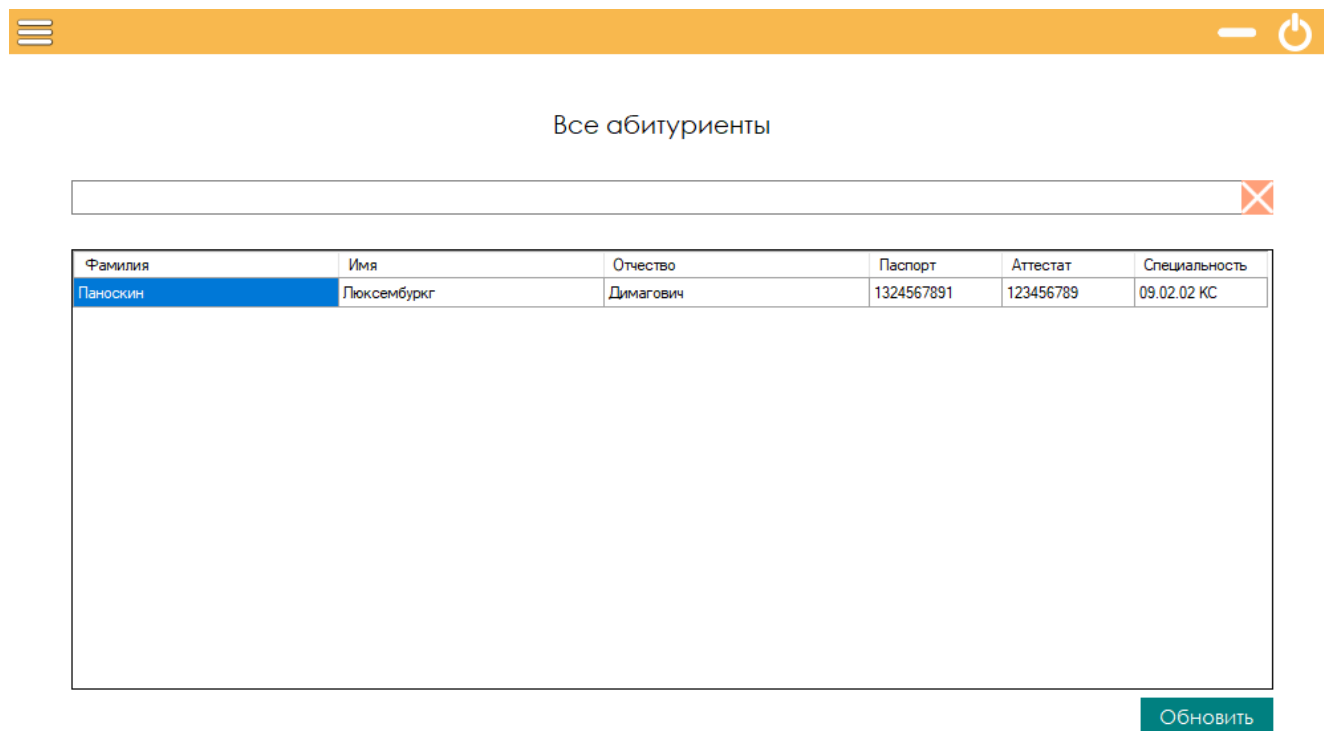
При нажатии на боковом меню кнопки «Аттестат» произойдет открытие пользовательского элемента управления «Информация об аттестате». Внешний вид и функции этого элемента аналогичны пользовательскому элементу управления «Данные». Изначально открывается первичный вид элемента, на котором доступен ввод номера аттестата для проверки его существования. Код проверки

существования аттестата аналогичен проверке паспорта и представлен на рисунке 16. Первичный вид представлен на рисунке 17. При успешной проверке существования данного аттестата происходит переход на основной вид элемента управления, представленного на рисунке 18. На этом элементе аналогично элементу с паспортом, имеются поля для ввода оценок по предметам, кнопки Сохранить и Отменить. При нажатии на кнопку Сохранить к базе данных отправляется запрос на обновление данных у конкретного аттестата. При нажатии на кнопку Отменить происходит возврат элемента управления к первоначальному виду и очистка всех полей для ввода. Код кнопки Сохранить представлен на рисунке 19.

```
myConnection.Open();
try
{
    string insertData = $"UPDATE Аттестаты SET " +
        $"Математика = '{mathInput.Text}'," +
        $"Физика = '{physicsInput.Text}'," +
        $"Труд = '{technologyInput.Text}'," +
        $"Музыка = '{musicInput.Text}'," +
        $"РусскийЯзык = '{russianInput.Text}'," +
        $"Физкультура = '{peInput.Text}'," +
        $"Иностранный = '{foreignInput.Text}'," +
        $"География = '{geografyInput.Text}'," +
        $"Биология = '{biologyInput.Text}'," +
        $"Химия = '{chemistryInput.Text}'," +
        $"Информатика = '{csInput.Text}'," +
        $"Обществознание = '{socialInput.Text}' WHERE Номер = '{attestInput.Text}';
    OleDbCommand commandForPassport = new OleDbCommand(insertData, myConnection);
    commandForPassport.ExecuteNonQuery();
    MessageBox.Show("Данные успешно добавлены", "Успешно");
    mathInput.Text = "";
    physicsInput.Text = "";
    technologyInput.Text = "";
    musicInput.Text = "";
    russianInput.Text = "";
    peInput.Text = "";
    foreignInput.Text = "";
    geografyInput.Text = "";
    biologyInput.Text = "";
    chemistryInput.Text = "";
    csInput.Text = "";
    socialInput.Text = "";
    attestInput.Text = "";
    panel1.Visible = true;
}
catch (System.Data.OleDb.OleDbException q) when (q.Message == "Отсутствует значение дл
{
    MessageBox.Show("Проверьте наличие введенных данных", "Ошибка");
}
myConnection.Close();
```

Рисунок 19 - Код кнопки «Сохранить»

При нажатии на кнопку «Абитуриенты» на выпадающем меню, произойдет открытие пользовательского элемента управления «Все абитуриенты». Внешний вид этого элемента представлен на рисунке 20.



Фамилия	Имя	Отчество	Паспорт	Аттестат	Специальность
Паноскин	Люксембург	Димагович	1324567891	123456789	09.02.02 КС

Рисунок 20 - Внешний вид пользовательского элемента управления «Все абитуриенты»

На данном элементе управления присутствуют такие элементы, как DataGridView, поле для поиска и кнопка Обновить. Все добавленные абитуриенты будут отображаться в таблице элемента DataGridView. Для поиска по фамилии, необходимо ввести в поле для ввода фамилию абитуриента и нажать клавишу Enter. Для сброса результата поиска, необходимо нажать на кнопку с крестиком. Так же присутствует предпросмотр карточки абитуриента, который открывается двойным кликом по номеру паспорта. Внешний вид карточки представлен на рисунке 21. На ней доступно редактирование основных данных. Для сохранения данных необходимо нажать на кнопку Сохранить. Стоит отметить, что между элементами управления можно свободно перемещаться. В любой момент можно открыть любой элемент управления и работать в нем, работа предыдущего элемента управления не будет утеряна.



Фамилия: Паноскин

Имя: Люксембург

Отчество: Димагович

Паспорт: 1324567891

Аттестат: 123456789

	Аттестат	Специальность
1	123456789	09.02.02 КС

Сохранить

Обновить

Рисунок 21 - Внешний вид карточки абитуриента

Отдельное внимание стоит уделить пользовательскому элементу управления «Помощь», который открывается при нажатии на выпадающем меню на кнопку «Помощь». При нажатии на нее происходит открытие пользовательского элемента управления «Помощь», внешний вид которого представлен на рисунке 22. Параллельно с этим открывается документация к программе, представленная в виде html-страницы. Внешний вид документации представлен на рисунке 23.

Помощь

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

Почта для обратной связи

Отправить

Рисунок 22 - Элемент управления «Помощь»

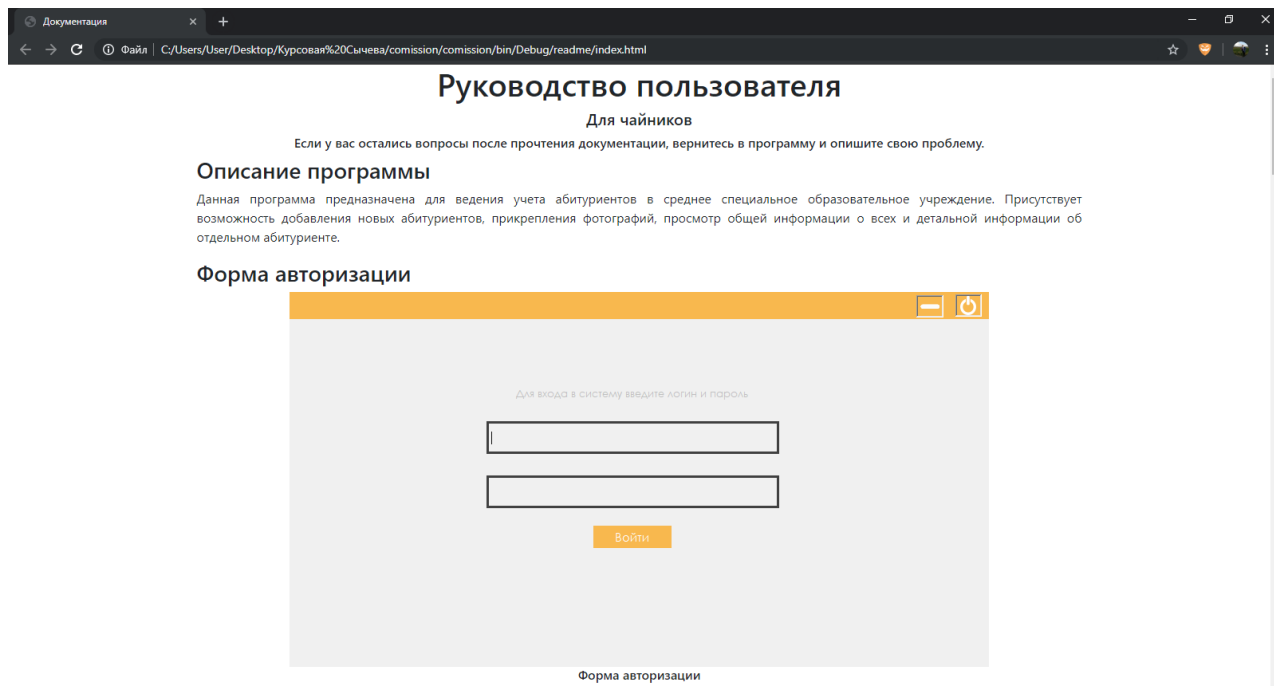


Рисунок 23 - Внешний вид документации к программе

На пользовательском элементе управления «Помощь» присутствует 2 поля ввода. Первое предназначено для ввода своей почты, чтобы у разработчиков программы была возможность связаться с тем, кто обратился с проблемой. Второе поле ввода предназначено для ввода своих пожеланий или отчетов об ошибках для разработчиков. При нажатии на кнопку «Отправить» происходит отправка писем разработчику и пользователю с уведомлением об отправке заявки.

## 2.3 Тестирование программного продукта

Тестирование данного программного продукта происходило посредством модульного тестирования. Тестировались такие модули, как: «Добавление», «Данные» и «Абитуриенты».

<b>ID 1</b>	<b>Тестирование «Добавление»</b>	
Автор: Сычева Валерия Сергеевна		Приоритет - высокий
1. Ввод данных в базу данных на элементе «Добавление»	Ввод произведен	Ввод произведен

2. Изменение данных в базе данных на элементе «Добавление»	Изменение произведено	Изменение произведено
3. Удаление данных в базе данных на элементе «Добавление»	Удаление произведено	Удаление не произведено – ошибка в построении SQLзапроса

**Статус: не пройден**

<b>ID 2</b>	<b>Тестирование «Данные»</b>	
Автор: Сычева Валерия Сергеевна		Приоритет - высокий
1. Ввод данных в базу данных на элементе «Данные»	Ввод произведен	Ввод произведен
2. Изменение данных в базе данных на элементе «Данные»	Изменение произведено	Изменение произведено
3. Удаление данных в базе данных на элементе «Данные»	Удаление произведено	Удаление произведено

**Статус: пройден**

<b>ID 3</b>	<b>Тестирование «Абитуриенты»</b>	
Автор: Сычева Валерия Сергеевна		Приоритет - высокий
1. Ввод данных в базу данных на элементе «Абитуриенты»	Ввод произведен	Ввод произведен
2. Изменение данных в базе данных на элементе «Абитуриенты»	Изменение произведено	Изменение не произведено. Возникла исключительная

		ситуация «неверный тип данных»
3. Удаление данных в базе данных на элементе «Абитуриенты»	Удаление произведено	Удаление произведено
<b>Статус: не пройден</b>		

### **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 приемной комиссии с информационной системой.

Для достижения поставленной цели были выполнены поставленные задачи, а именно:

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

Для данной информационной системы планируется добавить обработку исключительных ситуаций, которые позволят избежать ввода ошибочных значений. Также планируется добавить сжатие и восстановление базы данных в коде программного продукта для более корректной работы счетчика в таблицах.

Данный программный продукт будет являться практически значимым на рынке программного обеспечения, так как он имеет ряд преимуществ перед другими продуктами, например боковое слайд-меню, удобный интерфейс и приятное оформление.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Культин Н. Б. Visual C# в задачах и примерах. — 2-е изд., исправл. — СПб.: БХВ-Петербург, 2014. — 320 с.: ил. ISBN 978-5-9775-3323-2
2. Понамарев В. А. Программирование на C++/C# в Visual Studio .NET 2003. — СПб.: БХВ-Петербург, 2015.- 340 с.: ил. ISBN 978-5-9775-1224-4
3. Смоленцев Н.К. MATLAB. Программирование на Visual C#, Borland C#, JBuilder, VBA [Электронный ресурс]: учебный курс/ Смоленцев Н.К.— Электрон. текстовые данные.— Саратов: Профобразование, 2017.— 456 с.— Режим доступа: <http://www.iprbookshop.ru/63595.html>.— ЭБС «IPRbooks»
4. Стилмен Э., Грин Дж. Изучаем C#. 3-е изд. — СПб.: Питер, 2014. — 816 с.: ил. — (Серия «Head First O'Reilly»). ISBN 978-5-496-00867-9
5. Туральчук К.А. Параллельное программирование с помощью языка C# [Электронный ресурс]/ Туральчук К.А.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019.— 189 с.— Режим доступа: <http://www.iprbookshop.ru/79714.html>.— ЭБС «IPRbooks»
6. Тюльпинова Н.В. Алгоритмизация и программирование [Электронный ресурс]: учебное пособие/ Тюльпинова Н.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2019.— 200 с.— Режим доступа: <http://www.iprbookshop.ru/80539.html>.— ЭБС «IPRbooks»
7. Фленов М. Е. Библия C#. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2016. — 544 с.: ил. ISBN 978-5-9775-3533-5
8. Шарп Д. Microsoft Visual C#. Подробное руководство. 8-е изд./ Шарп Д. - СПб.: Питер, 2017. — 848 с. - ISBN 978-5-496-02372-6



## ПРИЛОЖЕНИЕ

### Листинг для переключения элементов управления на главной форме

```
ссылка:1
private void addButton_Click(object sender, EventArgs e)
{
    addNewUser1.Visible = true;
    attestatInfo1.Visible = false;
    studentInfo1.Visible = false;
    allStudents1.Visible = false;
    helpForm1.Visible = false;
    sidePanel.Height = addButton.Height;
    sidePanel.Top = addButton.Top;
    bunifuTransition1.HideSync(menuPanel);
}

ссылка:1
private void allStudents_Click(object sender, EventArgs e)
{
    addNewUser1.Visible = false;
    studentInfo1.Visible = false;
    attestatInfo1.Visible = false;
    allStudents1.Visible = true;
    helpForm1.Visible = false;
    sidePanel.Height = allStudents.Height;
    sidePanel.Top = allStudents.Top;
    bunifuTransition1.HideSync(menuPanel);
}

ссылка:1
private void helpButton_Click(object sender, EventArgs e)
{
    Process.Start(@"readme\index.html");
    studentInfo1.Visible = false;
    addNewUser1.Visible = false;
    allStudents1.Visible = false;
    attestatInfo1.Visible = false;
    helpForm1.Visible = true;
    sidePanel.Height = helpButton.Height;
    sidePanel.Top = helpButton.Top;
    bunifuTransition1.HideSync(menuPanel);
}

ссылка:1
private void dataButton_Click(object sender, EventArgs e)
{
    attestatInfo1.Visible = false;
    addNewUser1.Visible = false;
    allStudents1.Visible = false;
    helpForm1.Visible = false;
    studentInfo1.Visible = true;
    sidePanel.Height = dataButton.Height;
    sidePanel.Top = dataButton.Top;
    bunifuTransition1.HideSync(menuPanel);
}

ссылка:1
private void attbutton_Click(object sender, EventArgs e)
{
    addNewUser1.Visible = false;
    allStudents1.Visible = false;
    helpForm1.Visible = false;
    studentInfo1.Visible = false;
    attestatInfo1.Visible = true;
    sidePanel.Height = attbutton.Height;
    sidePanel.Top = attbutton.Top;
    bunifuTransition1.HideSync(menuPanel);
}
```