Министерство науки и высшего образования РФ  
ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСиС»

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

**КУРСОВАЯ РАБОТА**

по дисциплине «Технологии программирования»

на тему «Генератор справок»

Выполнил:

Студент БИВТ-20-6

Татура А.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:   
ст. преп. каф. ИКТ

Карпишук А.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

Оглавление

[**1.** **Формулирование требований** 4](#_Toc108548253)

[1.1.Описание программы 4](#_Toc108548254)

[1.2.Функции выполняемые программой. 4](#_Toc108548255)

[1.2.1.Требования к организации входных данных. 5](#_Toc108548256)

[1.2.2.Требования к организации выходных данных. 6](#_Toc108548257)

[1.3. Схема интерфейса 7](#_Toc108548258)

[1.3.1 Окно авторизации 7](#_Toc108548259)

[1.3.2 Окно авторизации 8](#_Toc108548260)

[1.3.3 Окно заказа справки 9](#_Toc108548261)

[1.3.4 Окно создания справки 10](#_Toc108548262)

[**2.Технический проект** 11](#_Toc108548263)

[2.1. Входные данные 11](#_Toc108548264)

[2.2. Схема данных 14](#_Toc108548265)

[2.2.1 Логическая схема данных 14](#_Toc108548266)

[2.2.2 Физическая схема данных 14](#_Toc108548267)

[2.2.3 Словарь данных 14](#_Toc108548268)

[2.3.Структура проекта 15](#_Toc108548269)

[2.3.1.Структурная схема 15](#_Toc108548270)

[2.3.2.Описание схемы 16](#_Toc108548271)

[2.4.Функциональная схема 18](#_Toc108548272)

[2.5.Схема интерфейса 18](#_Toc108548273)

[**3.Кодирование модулей** 19](#_Toc108548274)

[3.1.Код базы данных 19](#_Toc108548275)

[3.2.Код программы 19](#_Toc108548276)

[3.3.Выполнение программы 32](#_Toc108548277)

[3.3.1 Выполнение программы с описанием функций 32](#_Toc108548278)

[**4.Тестирование** 36](#_Toc108548279)

[4.1.Возможность авторизации как пользователь 36](#_Toc108548280)

[4.2. Возможность авторизации как администратор 37](#_Toc108548281)

[4.3. Возможность заказа справки 37](#_Toc108548282)

[4.4. Возможность создания справки 38](#_Toc108548283)

[4.5 Возможность отправка справки по почте 38](#_Toc108548284)

[4.6.Возможность перехода по разделам, обработке кнопок, соответствие эскизному проекту 39](#_Toc108548285)

[4.7.Возможность регистрации 40](#_Toc108548286)

[4.8.Результаты тестирования 42](#_Toc108548287)

[4.8.1 Возможность авторизации как пользователь 42](#_Toc108548288)

[4.8.2.Возможность авторизации как администратор 45](#_Toc108548289)

[4.8.3.Возможность заказа справки 47](#_Toc108548290)

[4.8.4.Возможность создания справки 47](#_Toc108548291)

[4.8.5.Возможность отправки справки по почте 50](#_Toc108548292)

[4.8.6.Возможность перехода по разделам, обработке кнопок, соответствие эскизному проекту в окнах: Авторизация, Регистрация, Заказ справка, Создание справки. 51](#_Toc108548293)

[4.8.7.Возможность регистрации 53](#_Toc108548294)

[**5. Руководство пользователя** 56](#_Toc108548295)

[**6. Список литературы** 58](#_Toc108548296)

1. **Формулирование требований**

## 1.1.Описание программы

Наименование - «Программа генератор справок».

Принцип действия: Пользователь заказывает нужную ему справку, а администратор получает заказ пользователя и заполняет необходимые поля для генерации справки, после чего отправляет её на почту пользователя.

## 1.2.Функции выполняемые программой.

Программа должна обеспечивать возможность выполнения перечисленных

ниже функций:

1. Генерация справки по заранее заготовленному шаблону (Файл с

определенным путем, названием, расширение \*.doсx, определенные строчки имеют

слова – {Name} – ФИО студента, {Date} – дата выдачи справки, {Group} – номер

группы, {Kyrs}- номер курса, данная функция доступна администратору;

2. Возможность сохранения документа, в удобное место на жестком диске

компьютера по средствам Microsoft Word;

3. Возможность регистрации, данные хранятся в базе данных;

4. Возможность авторизации, при авторизации происходит сравнение с

данными из базы данных;

5. Возможность отправки по почте, при отправке документ необходимо

вписать Email-адрес человека, которому предназначена справка, после нажатия на

кнопку «Отправить» откроется диалоговое, где нужно выбрать файл;

6. Возможность делать заказ справки. Данные о заказе отображаются в

табличке у Администратора;

7. Наличие двух ролей:

Администратор – выполняет функции создания справки из шаблона

документа, отправки заказчику на почту.

Пользователь – выполняет функцию заказа справки

### 1.2.1.Требования к организации входных данных.

|  |  |  |
| --- | --- | --- |
| № | Наименование | Описание |
| 1 | 2 | 3 |
| 1 | Почта | Строковый, Максимальное количество символов – 20,  Минимальное количество символов – 3, только латинские символы, цифры |
| 2 | Пароль | Строковый, Максимальное количество символов – 12,  Минимальное количество символов – 6, латинские символы, цифры |
| 3 | Фамилия | Строковый, Максимальное количество символов – 15,  Минимальное оличество символов – 3, русские буквы |
| 4 | Имя | Строковый, Максимальное количество символов – 15,  Минимальное количество символов – 3, русские буквы |
| 5 | Отчество | Строковый, Максимальное количество символов – 15, Минимальное количество символов – 3, русские буквы |
| 6 | Дата рождения | Строковый, Максимальное количество символов – 8,  Минимальное количество символов – 8, цифры |
| 7 | Роль пользователя | Строковый |
| 8 | Выбор справки | Строковый |
| 9 | Дата выдачи | Строковый, Максимальное количество символов – 8,  Минимальное количество символов – 8, цифры |
| 10 | Группа | Строковый, Максимальное количество символов – 10, Минимальное количество символов – 3, русские буквы, |
| 11 | Курс | Строковый |
| 12 | ФИО | Строковый, Максимальное количество символов – 60, Минимальное количество символов – 9, русские буквы, символы |

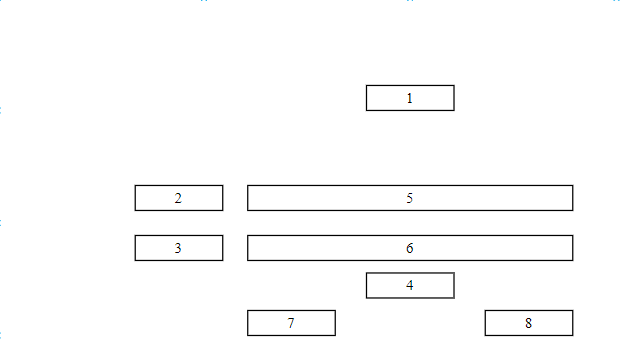
### 1.2.2.Требования к организации выходных данных.

|  |  |  |
| --- | --- | --- |
| № | Наименование | Описание |
| 1 | 2 | 3 |
| 1 | Почта | Строковый, Максимальное количество символов – 20,  Минимальное количество символов – 3, только латинские символы, цифры |
| 2 | Фамилия | Строковый, Максимальное количество символов – 15,  Минимальное количество символов – 3, русские буквы |
| 3 | Имя | Строковый, Максимальное количество символов – 15, Минимальное количество символов – 3, русские буквы |
| 4 | Отчество | Строковый, Максимальное количество символов – 15,  Минимальное количество символов – 3, русские буквы |
| 5 | Дата рождения | Строковый, Максимальное количество символов – 8, Минимальное количество символов – 8, цифры |
| 6 | Роль пользователя | Строковый |
| 7 | Выбор справки | Строковый |
| 8 | Дата выдачи | Строковый, Максимальное количество символов – 8, Минимальное количество символов – 8, цифры |
| 9 | Группа | Строковый, Максимальное количество символов – 10, Минимальное количество символов – 3, русские буквы,  символы |
| 1  0 | Курс | Строковый |
| 1  1 | ФИО | Строковый, Максимальное количество символов – 60, Минимальное количество символов – 9, русские буквы, символы |

## 1.3. Схема интерфейса

### 1.3.1 Окно авторизации

На Рисунке 1 отображена схема окна авторизации.

 Рисунок 1 – Схема окна авторизации.

В таблице 1 отображены кнопки в окне авторизации.

Таблица 1

|  |  |  |
| --- | --- | --- |
| № | Тип объект | Описание элемента схемы |
| 1 | 2 | 3 |
| 1 | Label | Текст «Генератор справок». |
| 2 | Label | Текст «Почта». |
| 3 | Label | Текст «Пароль». |
| 4 | Label | Текст «Логин или пароль не верны!». |
| 5 | textbox | Поле ввода почты. |
| 6 | maskedTextBox | Поле ввода пароля. |
| 7 | button | Кнопка «Регистрация». Осуществляет переход к окну регистрация |
| 8 | button | Кнопка «Войти». Осуществляет вход в аккаунт, к окну  заказа справки или создание справки. |

### 1.3.2 Окно авторизации

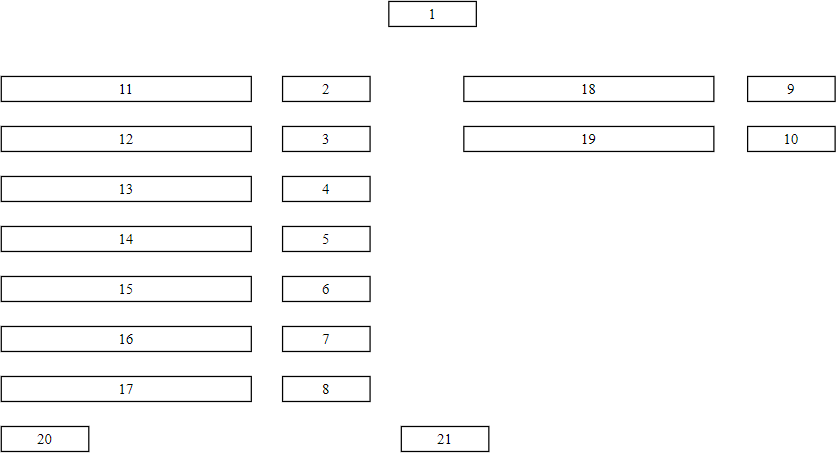
На Рисунке 2 отображена схема окна регистрация.

Рисунок 2 – Схема окна регистрации.

Таблица2- Описание окна регистрации

|  |  |  |
| --- | --- | --- |
| № | Тип объект | Описание элемента схемы |
| 1 | 2 | 3 |
| 1 | Label | Текст «Регистрация». |
| 2 | Label | Текст «Фамилия». |
| 3 | Label | Текст «Имя». |
| 4 | Label | Текст «Отчество». |
| 5 | Label | Текст «Почта». |
| 6 | Label | Текст «Пароль». |
| 7 | Label | Текст «Дата рождения». |
| 8 | Label | Текст «Роль пользователя». |
| 9 | Label | Текст «Группа». |
| 10 | Label | Текст «Курс». |
| 11 | textbox | Поле ввода фамилии. |
| 12 | textbox | Поле ввода имени. |
| 13 | textbox | Поле ввода отчества. |
| 14 | textbox | Поле ввода почты. |
| 15 | textbox | Поле ввода пароля. |
| 16 | dateTimePicke r | Поле выбора даты рождения. |
| 17 | comboBox | Поле выбора роли пользователя. |
| 18 | textbox | Поле ввода группы. |
| 19 | comboBox | Поле выбора курса. |
| 20 | button | Кнопка «Назад». Осуществляет переход к окну авторизации. |
| 21 | button | Кнопка «Зарегистрироваться». Осуществляет запись данных в базу данных. |

### 1.3.3 Окно заказа справки

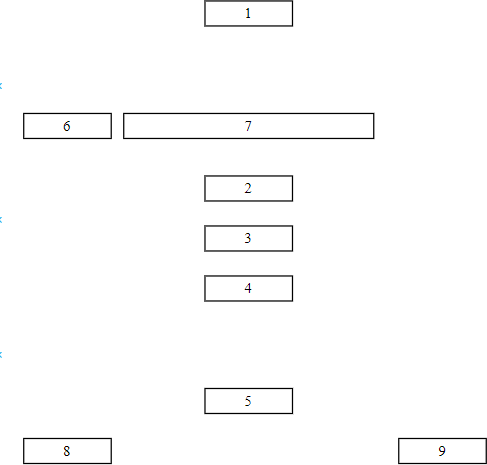
На рисунке 3 отображена схема окна заказа справок.

Рисунок 3 – схема окна заказа справки.

В таблице 3 отображены кнопки в окне заказа справки.

|  |  |  |
| --- | --- | --- |
| № | Тип объект | Описание элемента схемы |
| 1 | 2 | 3 |
| 1 | Label | Текст «Заказ справки». |
| 2 | Label | Текст «Справка 1 – Для военкомата». |
| 3 | Label | Текст «Справка 2 – Для работы». |
| 4 | Label | Текст «Справка 3 – Для получения льгот». |
| 5 | Label | Текст «Для заказа справки необходимо выбрать справку». |
| 6 | Label | Текст «Выбрать справку». |
| 7 | comboBox | Поле выбора справки. |
| 8 | button | Кнопка «Назад». Осуществляет переход к окну авторизации. |
| 9 | button | Кнопка «Заказ». Осуществляет запись данных в базу данных. |

### 1.3.4 Окно создания справки

На рисунке 4 отображена схема окна создания справок.

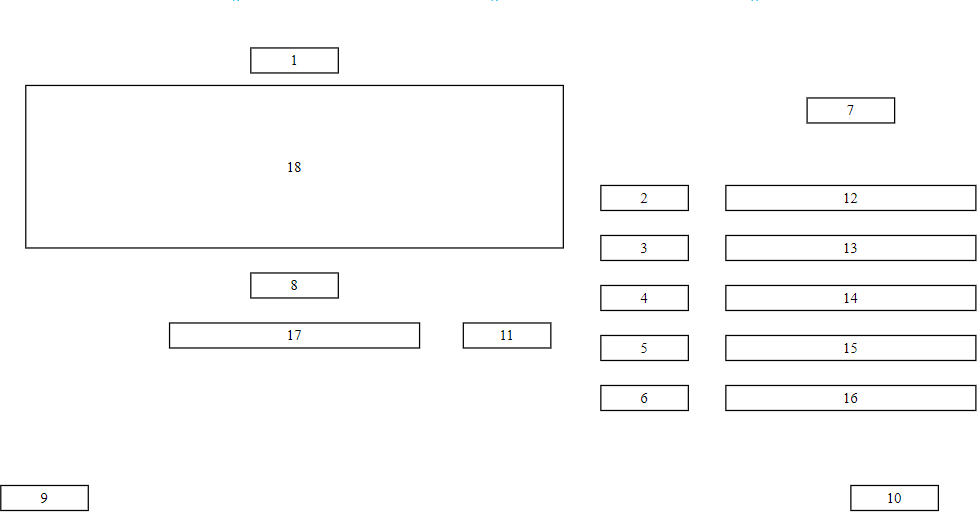
Рисунок 4 – Схема окна создания справок.

Таблица 4 – Описание окна создания справок.

|  |  |  |
| --- | --- | --- |
| № | Тип объект | Описание элемента схемы |
| 1 | 2 | 3 |
| 1 | Label | Текст «Информация о студентах». |
| 2 | Label | Текст «Выбор справки». |
| 3 | Label | Текст «ФИО». |
| 4 | Label | Текст «Дата выдачи». |
| 5 | Label | Текст «Группа». |
| 6 | Label | Текст «Курс». |
| 7 | Label | Текст «Для создания справки нужно заполнить поля: «Выбор справки», «ФИО», «Дата выдачи», «Группа», «Курс» и нажать заполнить. |
| 8 | Label | Текст «Адрес электронной почты». |
| 9 | button | Кнопка «Назад». Осуществляет переход к окну авторизации. |
| 10 | button | Кнопка «Заполнить». Осуществляет запись данных в документ  .docx. |
| 11 | button | Кнопка «Отправить». Осуществляет открытие диалогового окна с выбором файла .docx, его отправку. |
| 12 | comboBox | Поле выбора справки. |
| 13 | textbox | Поле ввода ФИО. |
| 14 | dateTimePicke r | Поле выбора даты выдачи. |
| 15 | textbox | Поле ввода группы. |
| 16 | textbox | Поле ввода курса. |
| 17 | textbox | Поле ввода адреса электронной почты. |
| 18 | dataGridView | Поле вывода данных из базы данных. |

# **2.Технический проект**

## 2.1. Входные данные

Входные данные приложения Генератор справок представлены в Таблице 5.

Таблица 5 – Входные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Поле | Ограничение | Тип | Описание |
| 1 | 2 | 3 | 4 | 5 |
| Окно авторизации | | | | |
| 1 | Почта | [A-Z a-z 0-9] {3,20} | varchar | Ввод почты для авторизации |
| 2 | Пароль | [A-Z a-z 0-9] {6,12} | varchar | Ввод пароля для  регистрации |
| Окно регистрации | | | | |
| 1 | Почта | [A-Z a-z 0-9] {3,20} | varchar | Ввод почты для регистрации |
| 2 | Пароль | [A-Z a-z 0-9] {6,12} | varchar | Ввод пароля для  регистрации |
| 3 | Фамилия | [А-Я а-я] {3, 15} | varchar | Ввод фамилии для регистрации |
| 4 | Имя | [А-Я а-я] {3, 15} | varchar | Ввод имени для  регистрации |
| 5 | Отчество | [А-Я а-я] {3, 15} | varchar | Ввод отчества для регистрации |
| 6 | Дата  рождения | yyyy.mm.dd | varchar | Выбор даты рождения  для регистрации |
| 7 | Роль пользовател я | Администратор,  Обычный пользователь | varchar | Выбор роли пользователя для регистрации |
| 8 | Группа | [А-Я а-я] {3, 10} | varchar | Ввод группы для  регистрации |
| 9 | Курс | 1, 2, 3, 4 | varchar | Выбор курса для регистрации |
| Окна заказа справки | | | | |
| 1 | Справка | Справка 1, Справка 2,  Справка 3 | varchar | Выбор справки для заказа справки |
| Окно создания справки | | | | |
| 1 | Справка | Справка 1, Справка 2,  Справка 3 | varchar | Выбор справки для заказа справки |
| 2 | ФИО | [А-Я а-я] {9, 60} | varchar |  |
| 3 | Дата выдачи | yyyy.mm.dd | varchar | Выбор даты рождения для регистрации |
| 4 | Группа | [А-Я а-я] {3, 10} | varchar | Ввод группы для  регистрации |
| 5 | Курс | 1, 2, 3, 4 | varchar | Выбор курса для регистрации |
| 6 | Почта | [A-Z a-z 0-9] {3,20} | varchar | Ввод почты для  регистрации |

## 2.2. Схема данных

### 2.2.1 Логическая схема данных

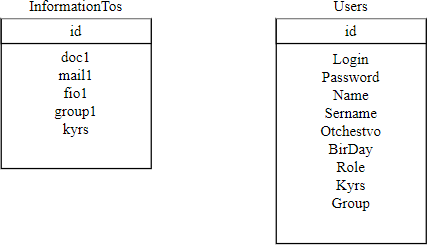
На Рисунке 5 представлена логическая схема данных в приложение Генератор справок.

Рисунок 5 – логическая схема.

### 2.2.2 Физическая схема данных

Изображение выглядит как текст

Автоматически созданное описаниеНа Рисунке 6 представлена схема данных, состоящая из связей между таблицами в приложении Генератор справок.

Рисунок 6 – Физическая схема.

### 2.2.3 Словарь данных

Таблица 6 – Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ключ | Наименование | Тип данных | Обязательность заполнения | Назначение |
| 1 | 2 | 3 | 4 | 5 |
| Users | | | | |
| PK | id | int |  | Уникальный идентификатор, номер пользователя |
|  | Login | varchar | null | Почта |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Password | varchar | null | Пароль |
|  | Name | varchar | null | Имя |
|  | Sername | varchar | null | Фамилия |
|  | Otchestvo | varchar | null | Отчество |
|  | BirDay | varchar | null | Дата рождения |
|  | Role | varchar | null | Роль пользователя |
|  | Kyrs | varchar | null | Номер курса |
|  | Group | varchar | null | Номер группы |
| InformationTos | | | | |
| PK | id | int | not null | Уникальный идентификатор, номер заказа |
|  | doc1 | varchar | null | Наименование заказанной  справки |
|  | mail1 | varchar | null | Почта |
|  | fio1 | varchar | null | ФИО студента |
|  | group1 | varchar | null | Номер группы |
|  | kyrs | varchar | null | Номер курса |

## 2.3.Структура проекта

### 2.3.1.Структурная схема

На Рисунке 7 представлена структурная схема, состоящая из форм:

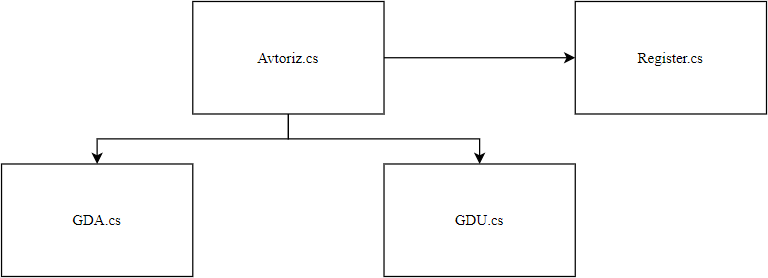
«Avtoriz.cs», «Register.cs», «GDA.cs», «GDU.cs» в проекте «Генератор справок». 

Рисунок 7 – Структурная схема.

### 2.3.2.Описание схемы

В Таблице 7 представлено описание структурной схемы, состоящей из форм «Avtoriz.cs», «Register.cs», «GDA.cs», «GDU.cs», в проекте «Генератор справок».

Таблица 7 – Описание структурной схемы

|  |  |  |
| --- | --- | --- |
| Номе р | Название модуля | Описание модуля |
| 1 | 2 | 3 |
| 1 | Avtoriz.cs | Окно с полями ввода почты, пароля, кнопки, по которой происходит процесс проверки почты и пароля, и происходит авторизация. Наличие кнопки регистрации, при нажатии происходит переход в окно регистрации. |
| 2 | Register.cs | Окно с полями ввода фамилии, имени, отчества, почты, пароля, группы. Поля выбора: дата рождения, роль пользователя, курс. Наличие кнопок: «Назад» - при нажатии на кнопку происходит переход в окно авторизации;  «Зарегистрироваться» - при нажатии на кнопку данные записываются в базу данных. |
| 3 | GDA.cs | Окно с выбором справки. Наличие кнопок:  «Назад» - при нажатии на кнопку происходит переход в окно авторизации; «Заказать» - при нажатии на кнопку данные записываются в базу данных. |
| 4 | GDU.cs | Окно с таблицей вывода данных из базы данных, полями ввода ФИО, группа, курс, адрес электронной почты. Поля выбора: выбор справки, дата выдачи. Наличие кнопок:  «Назад» - при нажатии на кнопку происходит переход в окно авторизации; «Заполнить» - при нажатии на кнопку происходит заполнение документа с форматом .dotx; «Отправить» - при нажатии на кнопку происходит открытие диалогового окна, в котором необходимо выбрать документ с форматом .docx. |

## 2.4.Функциональная схема

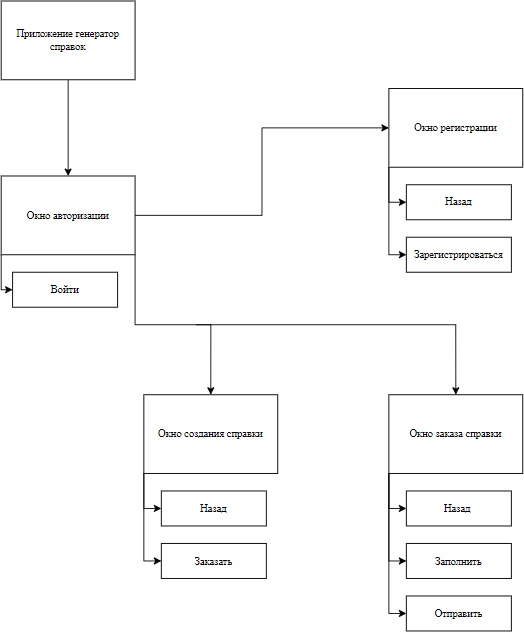
На рисунке 8 представлена схема интерфейса в приложении «Генератор справок».

Рисунок 8 – Функциональная схема

## 2.5.Схема интерфейса

На Рисунке 9 представлена схема интерфейса, состоящая из окон «Авторизация», «Регистрация», «Заказ справки», «Создание справки», в приложении Генератор справок.

Рисунок 9 – Схема интерфейса.

# **3.Кодирование модулей**

## 3.1.Код базы данных

CREATE TABLE "Users" (

"Id" INTEGER NOT NULL,

"Login"TEXT, "Password" TEXT,

"Name"TEXT, "Sername" TEXT,

"Otchestvo" TEXT, "BirDay" TEXT,

"Role" TEXT,

CONSTRAINT "PK\_Users" PRIMARY KEY("Id" AUTOINCREMENT)

);

CREATE TABLE "InformationTos" (

"id" INTEGER,

"doc1" TEXT,

"mail1"TEXT,

"fio1" TEXT,

"group1" TEXT,

"kyrs" TEXT,

PRIMARY KEY("id" AUTOINCREMENT)

);

## 3.2.Код программы

User.cs

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Text;

namespace GD

{

public class User

{

public int Id { get; set; }

public string Login { get; set; }

public string Password { get; set; }

public string Name { get; set; }

public string Sername { get; set; }

public string Otchestvo { get; set; }

public string BirDay { get; set; }

public string Role { get; set; }

public string Kyrs {get; set; }

public string Group { get; set; }

}

}

InformationTo.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace GD

{

public class InformationTo

{

public string doc1 { get; set; }

public string mail1 { get; set; }

public string fio1 { get; set; }

public string group1 { get; set; }

public string kyrs { get; set; }

public int id { get; set; }

}

}

ApplicationContext.cs

using Microsoft.EntityFrameworkCore;

namespace GD

{

public class ApplicationContext : DbContext

{

public DbSet<User> Users { get; set; }

public DbSet<InformationTo> InformationTos { get; set; }

public object User { get; internal set; }

public ApplicationContext()

{

Database.EnsureCreated();

}

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlite("Filename=User.sqlite");

}

}

}

Avtoriz.cs

using Microsoft.Data.Sqlite;

using Microsoft.Extensions.Options;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static Microsoft.EntityFrameworkCore.DbLoggerCategory.Database;

namespace GD

{

public partial class Avtoriz : Form

{

public Avtoriz()

{

InitializeComponent();

}

public void avtoris()

{

Avtoriz avtoriz = new Avtoriz();

ApplicationContext db = new ApplicationContext();

{

User user1 = new User { Login = textBox2.Text, Password = maskedTextBox2.Text };

if (db.Users.Any(u => (u.Login == textBox2.Text) && (u.Password == maskedTextBox2.Text)))

{

User user = db.Users.Single(u => (u.Login == textBox2.Text) && (u.Password == maskedTextBox2.Text));

if (user.Role == "Администратор")

{

GDU gdu = new GDU();

gdu.Show();

label7.Visible = false;

}

else

{

GDA gda = new GDA();

gda.Show();

label7.Visible = false;

}

Program.CurrentUser = user;

}

else

{

label7.Visible = true;

}

}

}

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

{

label7.Visible = false;

}

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

{

if (textBox2.Text != "" & maskedTextBox2.Text != "")

{

avtoris();

this.Hide();

}

else

{

MessageBox.Show("Логин или пароль не заполнены!");

}

}

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

{

Register register = new Register();

register.Show();

this.Hide();

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar == (int)Keys.Space) & (e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= '0' && e.KeyChar <= '9') || (e.KeyChar >= '@' && e.KeyChar <= '@') || (e.KeyChar >= '.' && e.KeyChar <= '.') || e.KeyChar == (char)Keys.Back)

{

}

else

{

e.KeyChar = '\0';

e.Handled = true;

}

}

private void maskedTextBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar == (int)Keys.Space) & (e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar == (char)Keys.Back)

{

}

else

{

e.KeyChar = '\0';

e.Handled = true;

}

}

}

}

Register.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace GD

{

public partial class Register : Form

{

public Register()

{

InitializeComponent();

}

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

{

if (textBox1.Text != "" & textBox2.Text != "" & textBox4.Text != "" & textBox5.Text != "" & comboBox1.Text != "" & textBox6.Text != "" & comboBox2.Text != "")

{

Record();

}

else

{

MessageBox.Show("Заполните все поля!");

}

}

public void Record()

{

using (ApplicationContext db = new ApplicationContext())

{

if (!db.Users.Any(u => (u.Login == textBox4.Text)))

{

User user1 = new User { Login = textBox4.Text, Password = textBox5.Text, Sername = textBox1.Text, Name = textBox2.Text, Otchestvo = textBox3.Text, BirDay = dateTimePicker1.Text, Role = comboBox1.Text, Group = textBox6.Text, Kyrs = comboBox2.Text };

db.Users.Add(user1);

db.SaveChanges();

MessageBox.Show("Данные успешно добавлены");

}

else

{

MessageBox.Show("Данный логин уже зарегистрирован");

}

}

}

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

{

Avtoriz avtoriz = (Avtoriz)Application.OpenForms[0];

avtoriz.Show();

Close();

}

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

{

comboBox1.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

comboBox2.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

}

private void textBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar == (int)Keys.Space) & (e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= '0' && e.KeyChar <= '9') || (e.KeyChar >= '@' && e.KeyChar <= '@') || (e.KeyChar >= '.' && e.KeyChar <= '.') || (e.KeyChar >= '\_' && e.KeyChar <= '\_') || (e.KeyChar >= '-' && e.KeyChar <= '-') || e.KeyChar == (char)Keys.Back)

{

}

else

{

e.KeyChar = '\0';

e.Handled = true;

}

}

private void textBox5\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar == (int)Keys.Space) & (e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a' && e.KeyChar <= 'z') || (e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar == (char)Keys.Back)

{

}

else

{

e.KeyChar = '\0';

e.Handled = true;

}

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

char c = e.KeyChar;

e.Handled = c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z' || c >= '0' && c <= '9' || e.KeyChar == (char)Keys.Space;

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

char c = e.KeyChar;

e.Handled = c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z' || c >= '0' && c <= '9' || e.KeyChar == (char)Keys.Space;

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

char c = e.KeyChar;

e.Handled = c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z' || c >= '0' && c <= '9' || e.KeyChar == (char)Keys.Space;

}

private void textBox6\_KeyPress(object sender, KeyPressEventArgs e)

{

char c = e.KeyChar;

e.Handled = c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z' || c >= 'а' && c <= 'я' || e.KeyChar == (char)Keys.Space;

}

}

}

GDA.cs

using System;

using System.Linq;

using System.Windows.Forms;

namespace GD

{

public partial class GDA : Form

{

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

{

Avtoriz avtoriz = (Avtoriz)Application.OpenForms[0];

avtoriz.Show();

Close();

}

public GDA()

{

InitializeComponent();

}

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

{

if (comboBox1.Text == "")

{

MessageBox.Show("Выберите справку!");

}

else

{

perenos();

}

}

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

{

comboBox1.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

}

public void perenos()

{

using ApplicationContext db = new ApplicationContext();

User user = db.Users.SingleOrDefault(u => u.Login == Program.CurrentUser.Login);

InformationTo informationTo = new InformationTo()

{

doc1 = comboBox1.Text,

mail1 = user.Login,

fio1 = user.Sername + ' ' + user.Name + ' ' + user.Otchestvo,

group1 = user.Group,

kyrs = user.Kyrs

};

db.InformationTos.Add(informationTo);

db.SaveChanges();

}

}

}

GDU.cs

using System;

using System.Diagnostics;

using System.Linq;

using System.Net;

using System.Net.Mail;

using System.Net.Mime;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace GD

{

public partial class GDU : Form

{

public static string file;

private readonly string TemplateFileName1 = @"D:\GD\GD\documents\Справка1.dotx";

private readonly string TemplateFileName2 = @"D:\GD\GD\documents\Справка2.dotx";

private readonly string TemplateFileName3 = @"D:\GD\GD\documents\Справка3.dotx";

public string mail, doc, fio, group, document1, document1New;

public GDU()

{

InitializeComponent();

}

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

{

if (comboBox1.Text == "Справка №1")

{

var Name = textBox1.Text; //ФИО

var Date = dateTimePicker1.Value.ToShortDateString(); //Дата выдачи

var Group = textBox2.Text; //Группа

var Kyrs = textBox3.Text; //Курс

var wordApp = new Word.Application();

try

{

var wordDocument = wordApp.Documents.Open(TemplateFileName1);

ReplaceWord("{Name}", Name, wordDocument);

ReplaceWord("{Date}", Date, wordDocument);

ReplaceWord("{Group}", Group, wordDocument);

ReplaceWord("{Kyrs}", Kyrs, wordDocument);

wordApp.Visible = true;

}

catch

{

MessageBox.Show("Ошибка в заполнении");

}

}

if (comboBox1.Text == "Справка №2")

{

var Name = textBox1.Text; //ФИО

var Date = dateTimePicker1.Value.ToShortDateString(); //Дата выдачи

var Group = textBox2.Text; //Группа

var Kyrs = textBox3.Text; //Курс

var wordApp = new Word.Application();

try

{

var wordDocument = wordApp.Documents.Open(TemplateFileName2);

ReplaceWord("{Name}", Name, wordDocument);

ReplaceWord("{Date}", Date, wordDocument);

ReplaceWord("{Group}", Group, wordDocument);

ReplaceWord("{Kyrs}", Kyrs, wordDocument);

wordApp.Visible = true;

}

catch

{

MessageBox.Show("Ошибка в заполнении");

}

}

if (comboBox1.Text == "Справка №3")

{

var Name = textBox1.Text; //ФИО

var Date = dateTimePicker1.Value.ToShortDateString(); //Дата выдачи

var Group = textBox2.Text; //Группа

var Kyrs = textBox3.Text; //Курс

var wordApp = new Word.Application();

try

{

var wordDocument = wordApp.Documents.Open(TemplateFileName3);

ReplaceWord("{Name}", Name, wordDocument);

ReplaceWord("{Date}", Date, wordDocument);

ReplaceWord("{Group}", Group, wordDocument);

ReplaceWord("{Kyrs}", Kyrs, wordDocument);

wordApp.Visible = true;

}

catch

{

MessageBox.Show("Ошибка в заполнении");

}

}

}

private void ReplaceWord(string stubToPeplace, string text, Word.Document wordDocument)

{

var range = wordDocument.Content;

range.Find.ClearFormatting();

range.Find.Execute(FindText: stubToPeplace, ReplaceWith: text);

}

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

{

GTable();

ProcessStartInfo si = new ProcessStartInfo();

si.FileName = @"D:\GD\GD\documents\Справка1.dotx";

si.UseShellExecute = true;

Process.Start(si);

comboBox1.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

}

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

var selectedRow = (sender as DataGridView).SelectedRows[0];

comboBox1.SelectedItem = selectedRow.Cells[0].Value.ToString();

textBox1.Text = selectedRow.Cells[2].Value.ToString();

textBox2.Text = selectedRow.Cells[3].Value.ToString();

textBox3.Text = selectedRow.Cells[4].Value.ToString();

textBox4.Text = selectedRow.Cells[1].Value.ToString();

}

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

{

if (textBox4.Text != "")

{

using (OpenFileDialog openFileDialog = new OpenFileDialog())

{

openFileDialog.Filter = "docx files (\*.docx)|\*.docx|All files (\*.docx\*)|\*.docx\*";

openFileDialog.ShowDialog();

file = openFileDialog.FileName;

string polych;

polych = textBox4.Text;

MailAddress from = new MailAddress("nngajcu@gmail.com", "Андрей");

MailAddress to = new MailAddress(polych);

System.Net.Mail.MailMessage m = new System.Net.Mail.MailMessage(from, to);

m.Subject = "Справка";

m.Body = "";

Attachment attach = new Attachment(file, MediaTypeNames.Application.Octet);

m.Attachments.Add(attach);

m.IsBodyHtml = true;

SmtpClient smtp = new SmtpClient("smtp.gmail.com", 587);

smtp.Credentials = new NetworkCredential("nngajcu@gmail.com", "O380550rus");

smtp.EnableSsl = true;

smtp.Send(m);

}

}

else

{

MessageBox.Show("Заполните почту!");

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button7\_Click(object sender, EventArgs e)

{

Avtoriz avtoriz = (Avtoriz)Application.OpenForms[0];

avtoriz.Show();

this.Close();

}

public void GTable()

{

using ApplicationContext db = new ApplicationContext();

dataGridView1.DataSource = db.InformationTos.ToList();

}

}

}

## 3.3.Выполнение программы

## 3.3.1 Выполнение программы с описанием функций

Авторизация - при запуске приложения появляется окно с авторизацией. Для того, чтобы авторизоваться необходимо ввести логин и пароль и нажать на кнопку войти, если почта и пароль неверны, то появляется поле с ошибкой (Рисунок 12). При успешной авторизации как пользователь произойдет переход в окно «Заказа справки» (Рисунок 10), при успешной авторизации как администратор происходит переход к окну «Создание справки» (Рисунок 11).

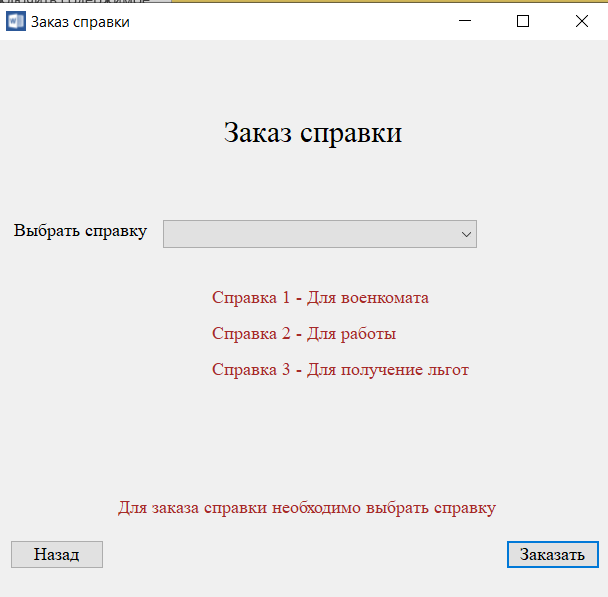


Рисунок 10 – Авторизация как пользователь

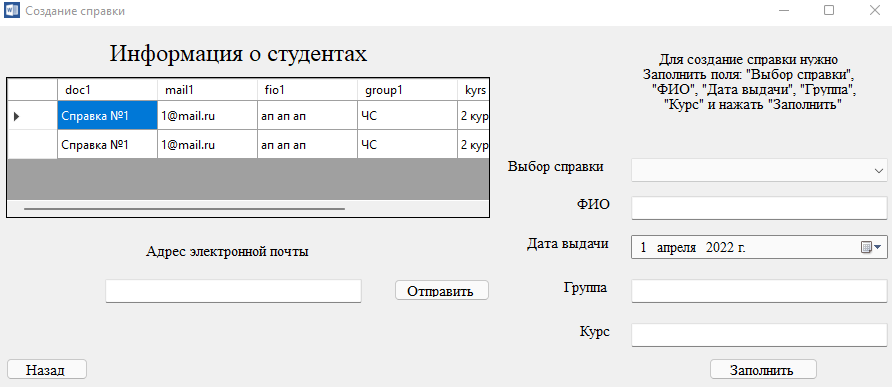


Рисунок 11 – Авторизация как администратор

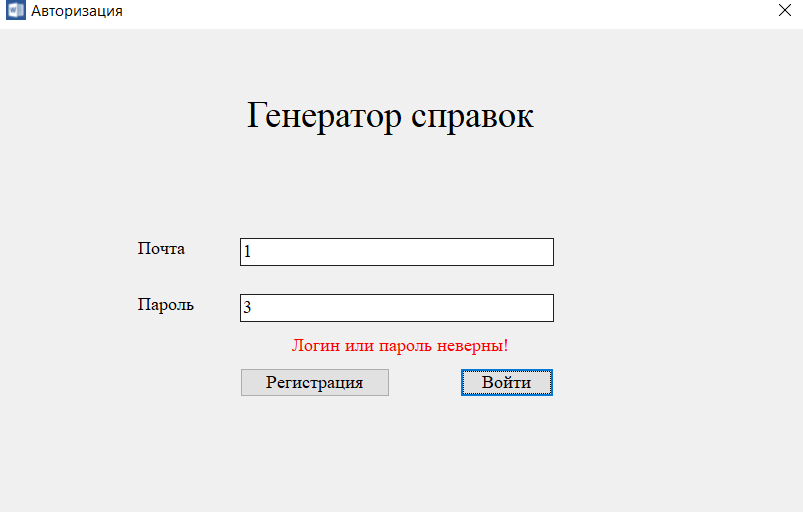


Рисунок 12 – Появление ошибки о неправильно введенных данных

Заказ справки – это окно отвечает за заказ справки, то есть запись данных пользователя в базу данных: «Номер справки», «Почта», «ФИО»,

«Группа» «Курс», для заказа справки необходимо авторизироваться как

пользователь и выбрать справку и нажать заказать (Рисунок 13). Если справка не выбрана, то появляется окно с ошибкой (Рисунок 14).

Изображение выглядит как текст

Автоматически созданное описание

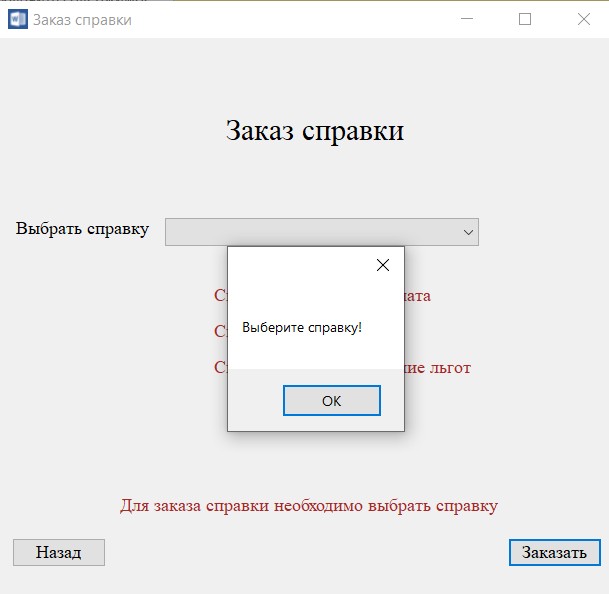
Рисунок 13 – Выполнение заказа на справку

Рисунок 14 – Появление ошибки

Создание справки – это окно отвечает за создание справки, то есть запись данных в документ Word, а также отправки справки по почте. Для записи данных в документ необходимо авторизироваться как администратор,

нажать на строку в таблице и нажать на кнопку «Заполнить» (Рисунки 14 - 17).

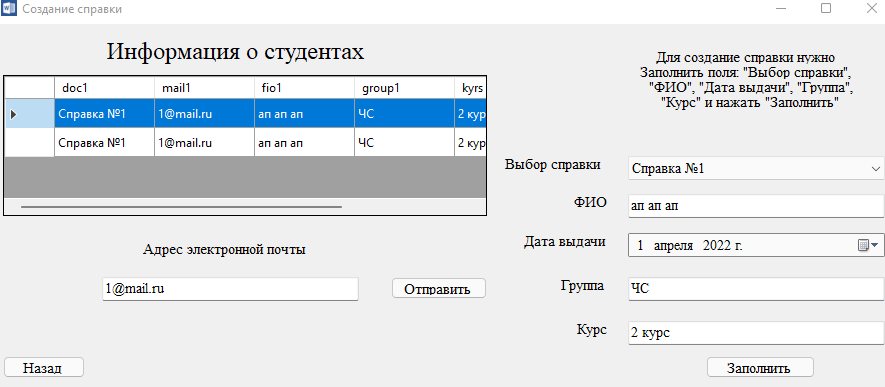


Рисунок 14 – Создание справки

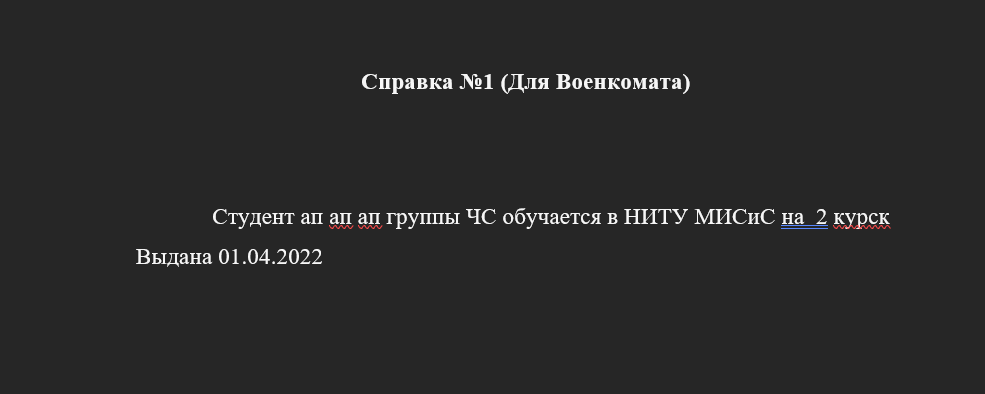


Рисунок 15 - Справка

# **4.Тестирование**

## 4.1.Возможность авторизации как пользователь

Для успешной авторизации обычного пользователя необходимо в окне

«Avtoris» ввести Почту, Пароль и нажать на кнопку «Войти».

Использовать следующие данные:

Почта – 1@mail.ru;

Пароль – 123.

Таблица 8 – Проверка Авторизации как пользователь

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Почта – « [liasnikov99@mail.ru»](mailto:liasnikov99@mail.ru) Пароль – «qwerty» | Переход к окну заказа справки |
| 2 | Почта – « »  Пароль – « » | Сообщение «Логин или пароль не заполнены» |
| 3 | Почта – «123» Пароль – «admin» | Сообщение «Логин или пароль не верны!» |
| 4 | Почта – «admin»  Пароль – « » | Сообщение «Логин или пароль не заполнены» |
| 5 | Почта – « »  Пароль – «admin» | Сообщение «Логин или пароль не заполнены» |
| 7 | Почта – «y» Пароль – «pilot» | Сообщение «Логин или пароль не верны!» |
| 8 | Почта – «админ»  Пароль – «pf» | Сообщение «Логин или пароль не верны!» |
| 9 | Почта – «letchik» Пароль – «admin1» | Сообщение «Логин или пароль не верны!» |
| 10 | Почта – «letchik»  Пароль – «letchik» | Сообщение «Логин или пароль не верны!» |

4.2. Возможность авторизации как администратор

Для успешной авторизации администратора необходимо в окне

«Avtoris» ввести Почту, Пароль и нажать на кнопку «Войти».

Почта – «2@mail.ru»

Пароль – «123»;

Таблица 9 – Проверка Авторизации как администрирования

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Почта –  [«2@mail.ru»](mailto:assassins2380@gmail.com)  »  Пароль – «123» | Переход к окну создания справки |
| 2 | Почта – « »  Пароль – « » | Сообщение «Логин или пароль не заполнены» |
| 3 | Почта – «123» Пароль – «admin» | Сообщение «Логин или пароль не верны!» |
| 4 | Почта – «admin»  Пароль – « » | Сообщение «Логин или пароль не заполнены» |
| 5 | Почта – « »  Пароль – «admin» | Сообщение «Логин или пароль не заполнены» |
| 7 | Почта – «y» Пароль – «pilot» | Сообщение «Логин или пароль не верны!» |
| 8 | Почта – «админ»  Пароль – «pf» | Сообщение «Логин или пароль не верны!» |
| 9 | Почта – «letchik» Пароль – «admin1» | Сообщение «Логин или пароль не верны!» |
| 10 | Почта – «letchik»  Пароль – «letchik» | Сообщение «Логин или пароль не верны!» |

4.3. Возможность заказа справки

Для успешного заказа справки необходимо в окне «GDA» выбрать справку и нажать на кнопку «Заказать».

Выбрать справку - «Справка 1»;

Таблица 10 – Проверка заказа справки

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Выбрать справку –  «Справка 1» | Выполнение заказа справки |
| 2 | Выбрать справку – « » | Сообщение «Выберите справку!» |

4.4. Возможность создания справки

Для успешного создания справки необходимо в окне «GDU» нажать на левый край таблицы нужной строчки, данные перенесутся в поля ввода и нажать на кнопку «Заполнить».

Строка «1»;

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Нажатие на строку «1» | Перенос значений в поле ввода |

4.5 Возможность отправка справки по почте

Для успешной отправки справки необходимо в окне «GDU» выполнить перенос данных в поля ввода (нажать на левый край таблицы нужной строчки), данные почты перенесутся в поле ввода почты и нажать на кнопку

«Отправить», появиться диалоговое окно, в котором необходимо выбрать файл c расширением .docx.

Выбором файла с расширением .docx;

Таблица 11 – Проверка отпраки по почте

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Выбор файл с расширением .docx | Отправка файл на почту |

## 4.6.Возможность перехода по разделам, обработке кнопок, соответствие эскизному проекту

Окна: Авторизация, Регистрация, Заказ справка, Создание справки.

Соответствие окна «Avtoriz» эскизному проекту.

Таблица 12 – Обработчик кнопок, соответствие эскизному проекту в окне «Авторизация»

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Окно «Авторизация» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам |

Таблица 13 - Обработчик кнопок, соответствие эскизному проекту в окне «Регистрация»

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Окно «Регистрация» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам |

Таблица 14 - Обработчик кнопок, соответствие эскизному проекту в окне «Заказ справки»

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Окно «Заказ справки» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам |

Таблица 15 - Обработчик кнопок, соответствие эскизному проекту в окне «Создание справки»

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Окно «Создание справки» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам |

## 4.7.Возможность регистрации

Для успешной регистрации необходимо в окне «Register» необходимо заполнить все и нажать на кнопку «Зарегистрироваться».

Фамилия – Антипов, Татура;

Имя – Илья, Андрей;

Отчество – Владимирович, Евгеньевич;

Почта – [antipov2000@gmail.com](mailto:antipov2000@gmail.com), [m2003573@edu.misis.ru](mailto:m2003573@edu.misis.ru))

Пароль – 1234567, 7654321;

Дата рождения – 12.08.2000, 31.12.2002;

Роль пользователя – Пользователь, Администратор;

Группа – (БПМ20-4-, БИВТ-20-6)

Курс – 2, 2.

Таблица 16 – Проверка регистрации

|  |  |  |
| --- | --- | --- |
| № п/п | Действие (Входное значение) | Ожидаемый результат |
| 1 | 2 | 3 |
| 1 | Фамилия – Антипов; Имя – Илья;  Отчество – Владиславович; Почта –  Antipov2000@gmail.com; Пароль – 1234567;  Дата рождения – 12.08.2000; Роль пользователя –  Обычный пользователь; Группа – БПМ-20-4  Курс - 2 | Сообщение «Данные успешно добавлены» |
| 2 | Фамилия – Татура; Имя – Андрей;  Отчество – Евгеньевич; Почта –  [m2003573@edu.misis.ru](mailto:kanatchikov2002@gmail.com)  ;  Пароль – 7654321;  Дата рождения – 18.05.2002; Роль пользователя  –Администратор; Группа – БИВТ-20-6;  Курс – 2. | Сообщение «Данные успешно добавлены» |
| 3 | Фамилия – Татура; Имя – Андрей;  Отчество – ;  Почта –;  Пароль – 7654321;  Дата рождения – 18.05.2002; Роль пользователя – ; Группа – ;  Курс – 2. | Сообщение «Заполните все поля!» |

## 4.8.Результаты тестирования

Все тестовые данные для проведения тестирования приложения

«Генератор справок» представлены в таблицах 17 – 27.

* + 1. Возможность авторизации как пользователь

При авторизации пользователь должен ввести свою почту и пароль:

* + 1. Почта – 1@mail.ru;
    2. Пароль – 123.

Таблица 17 – Проверка Авторизации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Логин – «  [1@mail.ru»](mailto:1@mail.ru)  Пароль – «123» | Переход к окну заказа справки | Переход к окну заказа справки | Пройден |
| 2 | Логин – « »  Пароль – « » | Сообщение  «Логин или пароль не заполнены» | Сообщение  «Логин или пароль не заполнены» | Пройден |
| 3 | Логин – «123» Пароль – «admin» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 4 | Логин – «admin» Пароль – « » | Сообщение  «Логин или пароль не заполнены» | Сообщение  «Логин или пароль не заполнены» | Пройден |
| 5 | Логин – « »  Пароль – «admin» | Сообщение  «Логин или пароль не заполнены» | Сообщение  «Логин или пароль не заполнены» | Пройден |
| 6 | Логин – «y» Пароль – «pilot» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 7 | Логин – «админ» Пароль – «pf» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 8 | Логин – «letchik» | Сообщение | Сообщение | Пройден |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Пароль – «admin1» | «Логин или пароль не  верны!»» | «Логин или пароль не  верны!»» |  |
| 9 | Логин – «letchik» Пароль – «letchik» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 10 | Почта – «letchik» Пароль – «letchik» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |

4.8.2.Возможность авторизации как администратор

При авторизации пользователь должен ввести свою почту и пароль

* + 1. Почта – «[2@mail.ru»;](mailto:2@mail.ru)
    2. Пароль – «123»;

Таблица 18 - Проверка авторизации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Логин – « [2@mail.ru»](mailto:2@mail.ru)  Пароль – «123» | Переход к окну заказа справки | Переход к окну заказа справки | Пройден |
| 2 | Логин – « »  Пароль – « » | Сообщение  «Логин или пароль не заполнены» | Сообщение  «Логин или пароль не заполнены» | Пройден |
| 3 | Логин – «123» Пароль – «admin» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 4 | Логин – «admin» Пароль – « » | Сообщение  «Логин или пароль не заполнены» | Сообщение  «Логин или пароль не заполнены» | Пройден |
| 5 | Логин – « »  Пароль – «admin» | Сообщение  «Логин или пароль не заполнены» | Сообщение  «Логин или пароль не заполнены» | Пройден |
| 6 | Логин – «y» Пароль – «pilot» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 7 | Логин – «админ» Пароль – «pf» | Сообщение  «Логин или | Сообщение  «Логин или | Пройден |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | пароль не  верны!» | пароль не  верны!» |  |
| 8 | Логин – «letchik» Пароль – «admin1» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 9 | Логин – «letchik» Пароль – «letchik» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |
| 10 | Почта – «letchik» Пароль – «letchik» | Сообщение  «Логин или пароль не верны!» | Сообщение  «Логин или пароль не верны!» | Пройден |

4.8.3.Возможность заказа справки

Для заказа справки обычный пользователь должен выбрать справку

* + 1. Выбрать справку – «Справка 1»;

Таблица 19 - Проверка заказа справки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Выбрать справку – «Справка 1» | Выполнение заказа справки | Выполнение заказа справки | Пройден |
| 2 | Выбрать справку – « » | Сообщение  «Выберите справку!» | Сообщение  «Выберите справку!» | Пройден |

### 4.8.4.Возможность создания справки

Для создания справки администратор должен нажать на левый край строчки таблицы, данные перенесутся в поля ввода и нажать кнопку

«Заполнить».

* + 1. Строка «1»;

Таблица 20 - Проверка создания справки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Нажатие на строку «1» | Перенос | Перенос | Пройден |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | значений в  поле ввода | значений в  поле ввода |  |

4.8.5.Возможность отправки справки по почте

Для отправки справки по почте администратор должен нажать на левый край строчки таблицы, данные перенесутся в поля ввода почты и нажать кнопку «Отправить», откроется диалоговое окно, в котором необходимо выбрать файл с расширением .docx.

* + 1. Выбором файла с расширением .docx;

Таблица 21 – Возможность отправки справки по почте

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Выбор файл с расширением .docx | Отправка файл на почту | Отправка файл на почту | Пройден |

4.8.6.Возможность перехода по разделам, обработке кнопок, соответствие эскизному проекту в окнах: Авторизация, Регистрация, Заказ справка, Создание справки.

Таблица 22 - Обработчик кнопок, соответствие эскизному проекту в окне «Авторизация»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Окно «Авторизация» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам | Соответствует эскизному проекту, работа кнопок | Пройден |

Таблица 23 - Обработчик кнопок, соответствие эскизному проекту в окне «Регистрация»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Окно «Регистрация» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам | Соответствует эскизному проекту, работа кнопок | Пройден |

Таблица 24 - Обработчик кнопок, соответствие эскизному проекту в окне «Заказ справки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Окно «Заказ справки» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам | Соответствует эскизному проекту, работа кнопок | Пройден |

Таблица 25 - Обработчик кнопок, соответствие эскизному проекту в окне «Создание справки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Окно «Создания справки» | Соответствует эскизному проекту, работа кнопок, возможность перехода по разделам | Соответствует эскизному проекту, работа кнопок | Пройден |

### 4.8.7.Возможность регистрации

При регистрации пользователь должен заполнить следующие поля: Фамилия, Имя, Отчество, Почта, Пароль, Дата рождения, Роль пользователя, Группа, Курс и нажать на кнопку «Зарегистрироваться»

Фамилия – Антипов, Татура;

Имя – Илья, Андрей;

Отчество – Владимирович, Евгеньевич;

Почта – [antipov2000@gmail.com](mailto:antipov2000@gmail.com), [m2003573@edu.misis.ru](mailto:m2003573@edu.misis.ru))

Пароль – 1234567, 7654321;

Дата рождения – 12.08.2000, 31.12.2002;

Роль пользователя – Пользователь, Администратор;

Группа – (БПМ20-4-, БИВТ-20-6)

Курс – 2, 2.

Таблица 26 - Проверка регистрации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Действие (Входное значение) | Ожидаемый результат | Фактический результат | Статус теста (пройде н / не пройден  ) |
| 1 | Фамилия – Антипов; Имя – Илья  Отчество – Владимирович;  Почта – antipov2000@gmail.com; Пароль – 1234567;  Дата рождения – 12.08.2000; | Сообщение  «Данные успешно добавлены» | Сообщение  «Данные успешно добавлены» | Пройден |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Роль пользователя – Обычный пользователь;  Группа – БПМ-20-4;  Курс - 2 |  |  |  |
| 2 | Фамилия – Татура; Имя – Андрей;  Отчество – Евгеньевич; Почта –  [m2003573@edu.misis.ru](mailto:kanatchikov2002@gmail.com)  ;  Пароль – 7654321;  Дата рождения – 18.05.2002; Роль пользователя  –Администратор; Группа – БИВТ-20-6;  Курс – 2. | Сообщение  «Данные успешно добавлены» | Сообщение  «Данные успешно добавлены» | Пройден |
| 3 | Фамилия – Татура; Имя – Андрей;  Отчество – ;  Почта –;  Пароль – 7654321;  Дата рождения – 18.05.2002; Роль пользователя – ; Группа – ;  Курс – 2. | Сообщение  «Заполните все поля!» | Сообщение  «Заполните все поля!» | Пройден |
| 4 | Фамилия – ;  Имя – Илья; Отчество – Владимирович;  Почта – antipov2000@gmail.com; Пароль – 1234567;  Дата рождения – 12.08.2000; Роль пользователя – Обычный пользователь;  Группа – БПМ-20-4;  Курс - 2 | Сообщение  «Заполните все поля!» | Сообщение  «Заполните все поля!» | Пройден |

# **5. Руководство пользователя**

Авторизация - при запуске приложения появляется окно с авторизацией. Для того, чтобы авторизоваться необходимо ввести логин и пароль и нажать на кнопку войти, если почта и пароль неверны, то появляется поле с ошибкой (Рисунок 12). При успешной авторизации как пользователь произойдет переход в окно «Заказа справки» (Рисунок 10), при успешной авторизации как администратор происходит переход к окну «Создание справки» (Рисунок 11).

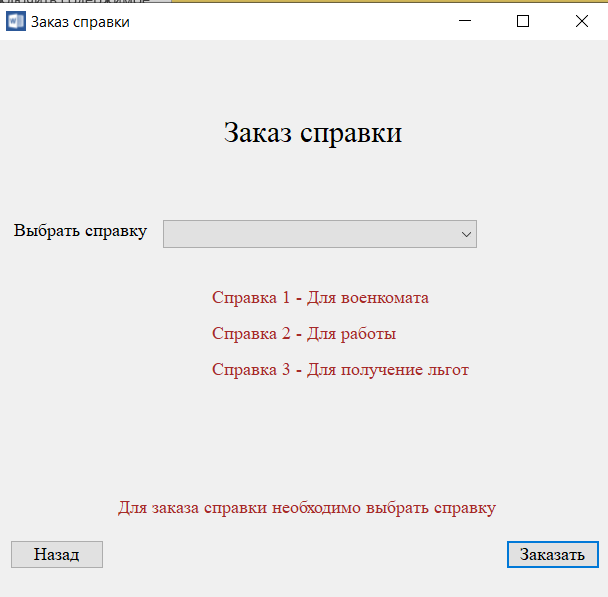


Рисунок 10 – Авторизация как пользователь

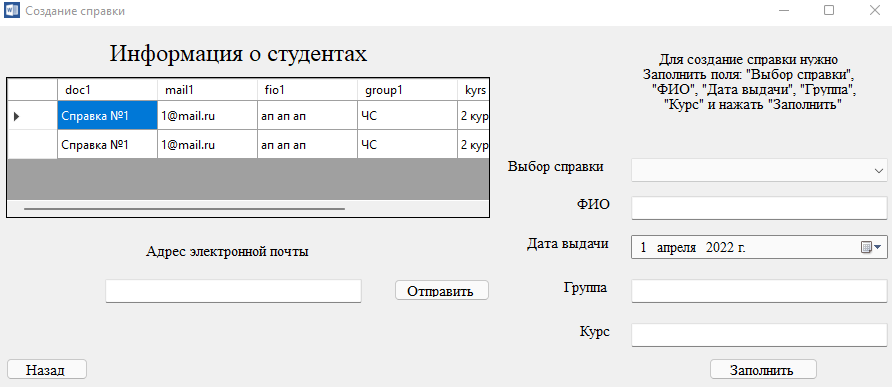


Рисунок 11 – Авторизация как администратор

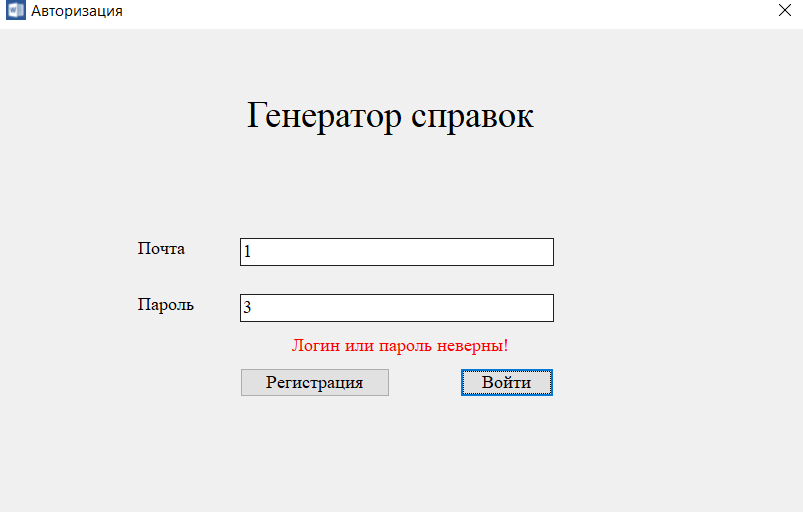


Рисунок 12 – Появление ошибки о неправильно введенных данных

Заказ справки – это окно отвечает за заказ справки, то есть запись данных пользователя в базу данных: «Номер справки», «Почта», «ФИО»,

«Группа» «Курс», для заказа справки необходимо авторизироваться как

пользователь и выбрать справку и нажать заказать (Рисунок 13). Если справка не выбрана, то появляется окно с ошибкой (Рисунок 14).

Изображение выглядит как текст

Автоматически созданное описание

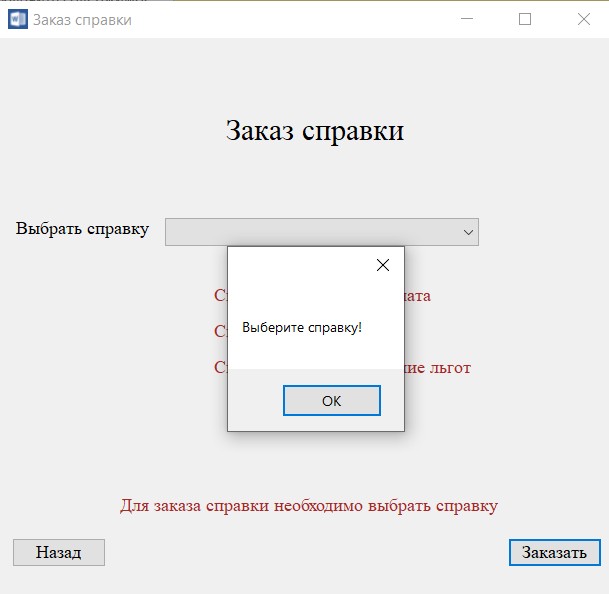
Рисунок 13 – Выполнение заказа на справку

Рисунок 14 – Появление ошибки

# **6. Список литературы**

1. Агуров, Павел C#. Сборник рецептов / Павел Агуров. - М.: "БХВ-Петербург", **2012**. - 432 c.  
2. Албахари, Джозеф C# 3.0. Справочник / Джозеф Албахари , Бен Албахари. - М.: БХВ-Петербург, 2012. - 944 c.

3. Windows Forms. Программирование на C# [Электронный ресурс]. — Режим доступа: http://csharpcoding.org/category/windows-forms/ (дата обращения: 15.03.2017).

4. Васильев, А. C#. Объектно-ориентированное программирование: Учебный курс / А. Васильев. - СПб.: Питер, 2012. - 320 c.

5. Грабер, Мартин SQL для простых смертных / Мартин Грабер. - М.: ЛОРИ, 2014. - 378 c.