



**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Колледж программирования и кибербезопасности**

**Отчет о выполнении практического задания  
по УП. 01.01 Разработка программных модулей  
на тему «Создание окна регистрации пользователя. Обработка  
исключительных ситуаций. Регистрация пользователя в базе  
данных»**

**Специальность – 09.02.07 информационные системы и программирование**

**Выполнил студент:**

\_\_\_\_\_Синюков Д.В.

**Группа: ПКС-35**

**Руководитель:**

\_\_\_\_\_Стоколос М.Д

**Работа защищена с оценкой** \_\_\_\_\_

**Дата защиты** \_\_\_\_\_

**Москва**

**2024**

## 1. Создание окна регистрации пользователя

Окна приложения выполнены в вертикальном формате. Данный формат был выбран для масштабирования системы под телефоны. Преобладающими цветами в приложении выступают оттенки зеленого, синий и черный. Пример окна регистрации представлен на рисунке 1.

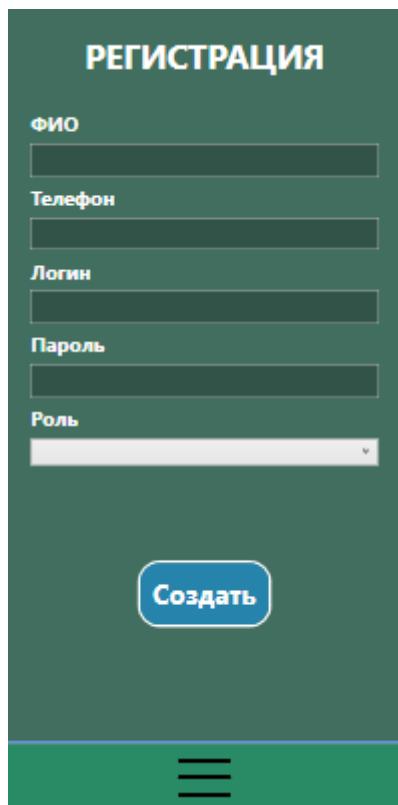
The image shows a vertical registration form on a dark green background. At the top, the word 'РЕГИСТРАЦИЯ' is written in white. Below it are five input fields: 'ФИО', 'Телефон', 'Логин', 'Пароль', and 'Роль'. The 'Роль' field is a dropdown menu. A blue button with the text 'Создать' is centered below the fields. At the very bottom, there is a green bar containing a white hamburger menu icon.

Рис. 1 Окно регистрации пользователя

Окно регистрации называется «MainWindow.xaml» и состоит из трех областей:

- основная область действий;
- нижняя область навигации;
- всплывающая область навигации.

Пример всплывающей области представлен на рисунке 2.

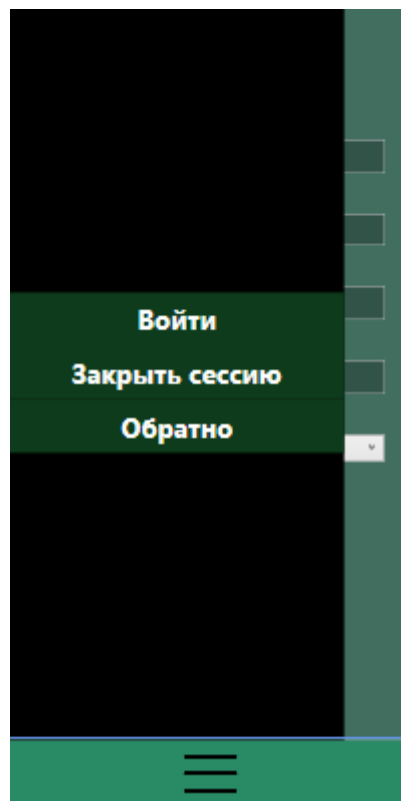


Рис. 2 Пример всплывающей области навигации

Основная область хранит в себе поля для ввода «ФИО», «Логин», «Телефон» и «Пароль» (выполнены с помощью инструмента «TextBox»), а также поле для выбора роли (выполнено с помощью инструмента «ComboBox»). Кнопка «Создать» проверяет введенные данные и отправляет их на регистрацию в систему. Заголовок окна и вспомогательный текст выполнены с помощью инструмента «TextBlock». Всплывающая область навигации хранит в себе кнопки, которые выполнены с помощью инструмента «Button». Кнопка «Обратно» закрывает всплывающую область, однако также есть способ закрыть ее нажатием на основную область. Кнопка «Заккрыть сессию» закрывает программу, после подтверждения пользователя. Кнопка «Войти» открывает окно авторизации пользователя. Нижняя панель навигации позволяет открыть всплывающую панель нажатием на кнопку с тремя полосками (выполнена с помощью инструмента «Button»).

## 2. Обработка исключительных ситуаций

Для обработки исключительных ситуаций были созданы классы такие как:

- «Help.cs»;
- «CheckInputData.cs».

Класс «Help.cs» хранит в себе шаблоны создания сообщений пользователю. Пример скрипта для создания сообщений представлен ниже.

```
// Вывод информационного сообщения
public object createInformationMessageBox(string message) =>
MessageBox.Show(message, "Тех-поддержка", MessageBoxButton.OK,
MessageBoxImage.Information);
```

```
// Вывод уточняющих сообщений
public bool createQuestionMessageBox(string message) =>
(MessageBox.Show(message, "Тех-поддержка", MessageBoxButton.YesNo,
MessageBoxImage.Question) == MessageBoxResult.Yes);
```

класс «CheckInputData.cs» проверяет вводимую информацию на наличие SQL-инъекций и корректность. Пример скрипта для проверки SQL-инъекций и корректность номера телефона представлен ниже.

```
// Проверка нмоера телефона
public bool checkInputPhoneNumber(string inputPhoneNumberExample)
{
// Допустимый формат номера: 8NNNNNNNNNNN
Regex outPlusPhoneNumberController = new Regex(
"[8][0-9]{10}");
return
outPlusPhoneNumberController.IsMatch(inputPhoneNumberExample);
}
// Защита от SQLI
public bool checkSQLI(string example)
{
if (example.Length == 0)
return false;
string[] incorrectSymbols = new string[3] { "'", "-", ";" };
for (int symbolIndex = 0; symbolIndex < 3; symbolIndex++)
{
if (example.Contains(incorrectSymbols[symbolIndex]))
return false;
}
return true;
}
```

### 3. Регистрация пользователя в базы данных

Регистрация пользователя происходит в классе «DB.cs» в котором хранится весь код, отвечающий за работу с базой данных. Для регистрации в метод «addNewAccount()» передается массив строковых данных, в коором хранятся все необходимые данные. Пример метода представлен ниже.

```
// Регистрация пользователя в бд
public bool addNewAccount(string[] regInformation)
{
    if(!checkDuplicate(regInformation))
        return false;
    ulong id = getId(regInformation[4]);
    try
    {
        using (SqlConnection connection = new SqlConnection(url))
        {
            connection.Open();
            SqlCommand commandToAddInformationFromTable = new SqlCommand($"@
INSERT INTO {regInformation[4]}
VALUES({id},      N'{regInformation[0]}',      N'{regInformation[1]}',
N'{regInformation[2]}',      N'{regInformation[3]}',
N'{regInformation[4]}')");
            commandToAddInformationFromTable.Connection = connection;
            commandToAddInformationFromTable.ExecuteNonQuery();
        }
        return true;
    } catch(Exception)
    {
        return false;
    }
}
```

Перед регистрацией данных они проверяются на дубликаты и совпадения с уже существующими данными. Для этого используется метод «CheckDuplicate()», который запрашивает из интересуемой таблицы все кортежи и сверяет их с полученным. Пример метода представлен ниже.

```

// Проверка данных на дубликаты в таблице
private bool checkDuplicate(string[] information)
{
    string phoneNumber = information[0],
    fio = information[1],
    login = information[2],
    password = information[3],
    table = information[4];
    using (SqlConnection conn = new SqlConnection(url))
    {
        conn.Open();
        using (SqlCommand cmd = new SqlCommand())
        {
            cmd.Connection = conn;
            cmd.CommandText = $"
            SELECT *
            FROM {table};
            ";
            SqlDataReader dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                // Если были совпадения
                if (login.Trim() == ("" + dr["Логин"]).Trim() ||
                    password.Trim() == ("" + dr["Пароль"]).Trim() ||
                    fio.Trim() == ("" + dr["ФИО"]).Trim() ||
                    phoneNumber.Trim() == ("" + dr["Телефон"]).Trim())
                {
                    new Helps().createInformationMessageBox("Данные заняты другим
                    пользователем!");
                    return false;
                }
            }
            dr.Close();
        }
        conn.Close();
    }
    return true;
}

```