**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**МДК 11.01**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИСПП-35 | |  |  | *Бестужев С.В.* |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  |  | *Маломан Ю. С.* |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

# **Лабораторная работа №20**

**Реализация разграничения прав доступа пользователей**

1. **Цель работы** 
   1. Научиться разграничивать права доступа пользователей на уровне интерфейса приложения;
   2. Научиться изменять настройки подключения к БД средствами Entity Framework Core.
2. **Контрольные вопросы**

Вопрос: 8.1 Как изменить настройки подключения к БД в клиентском приложении?

Настройки подключения к базе данных в клиентском приложении обычно определяются через конфигурационный файл или переменные окружения. Вот некоторые способы их изменения:

1. Через конфигурационный файл (например, appsettings.json):

{

"ConnectionStrings": {

"DefaultConnection": "Server=myserver;Database=MyDB;User Id=myuser;Password=mypassword;"

}

}

2. Через переменные окружения:

ConnectionStrings:DefaultConnection="Server=myserver;Database=MyDB;User Id=myuser;Password=mypassword;"

3. Через код:

Configuration.GetConnectionString("DefaultConnection") = "Server=newserver;Database=NewDB;User Id=newuser;Password=newpassword;";

```

4. Через пользовательский класс конфигурации:

public class ConnectionStringConfig

{

public string DefaultConnection { get; set; }

}

// В программе:

var connectionString = new ConnectionStringConfig();

connectionString.DefaultConnection = "Server=newserver;Database=NewDB;User Id=newuser;Password=newpassword;";

Вопрос: Какими способами можно обеспечить хранение пользователей и ролей пользователей в БД (отобразить в виде ERD)?

Вот пример ERD для хранения пользователей и их ролей:

1. Таблица Users:

- Id (PK)

- Username

- PasswordHash

- Email

- CreatedDate

2. Таблица Roles:

- Id (PK)

- Name

- Description

3. Таблица UserRoles:

- UserId (FK)

- RoleId (FK)

4. Таблица Permissions:

- Id (PK)

- Name

- Description

5. Таблица RolePermissions:

- RoleId (FK)

- PermissionId (FK)

ERD:

Users (1) ---\* UserRoles (2) ---\* Roles (3)

|---\* Permissions (4) ---\* RolePermissions (5)

|---\* Permissions (4) ---\* RolePermissions (5)

|

---\* UserRoles (2) ---\* Roles (3)

Этот подход позволяет:

- Хранить информацию о пользователях

- Связывать пользователей с ролями

- Связывать роли с разрешениями

- Динамически назначать разрешения пользователям через роли

Вопрос: Что такое «авторизация»?

Авторизация - это процесс проверки прав доступа пользователя к ресурсу или операции [1]. Она определяет, какие действия пользователь может выполнять в системе на основе его ролей или привилегий.

Основные аспекты авторизации:

- Проверка прав доступа пользователя к ресурсам

- Управление доступом к функциям и операциям

- Реализация механизмов безопасности для защиты данных

Вопрос: Что такое «регистрация»?

Регистрация - это процесс создания учетной записи пользователя в системе [1]. Она включает в себя:

- Создание нового пользователя

- Установка уникального идентификатора (например, username или email)

- Создание пароля или установка метода аутентификации

- Заполнение дополнительных полей профиля (если необходимо)

После регистрации пользователь получает возможность входа в систему и использования предоставленных ей возможностей.

Регистрация обычно включает следующие шаги:

1. Ввод персональной информации

2. Выбор пароля

3. Подтверждение регистрации (например, по электронной почте)

4. Активация аккаунта

1. **Вывод**
   1. В ходе лабораторной работы мы научились разграничивать права доступа пользователей на уровне интерфейса приложения;
   2. Научились изменять настройки подключения к БД средствами Entity Framework Core.