

# **Лабораторная работа №20**

## **Реализация разграничения прав доступа пользователей**

### **1 Цель работы**

1.1 Научиться разграничивать права доступа пользователей на уровне интерфейса приложения;

1.2 Научиться изменять настройки подключения к БД средствами Entity Framework Core.

### **2 Литература**

2.1 Федорова, Г. Н. Разработка, администрирование и защита баз данных [Текст]: учебник / Г. Н. Федорова. - 4-е изд., стер. – Москва: Издательский центр «Академия», 2020. – гл.9.

### **3 Подготовка к работе**

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание лабораторной работы.

### **4 Основное оборудование**

4.1 Персональный компьютер.

### **5 Задание**

5.1 Создание таблиц с пользователями и ролями пользователей в БД (CinemaUsers, CinemaUserRoles).

5.1.1 Добавить в БД таблицу ролей пользователей:

- идентификатор (целое число, автоинкрементное значение, PK)
- название роли (строка длиной до 20 символов)

Созданную таблицу заполнить значениями Администратор, Билетер, Посетитель.

5.1.2 Добавить в БД таблицу пользователей:

- идентификатор (целое число, автоинкрементное значение, PK)
- логин (строка длиной до 50 символов)
- пароль (строка длиной до 20 символов)
- идентификатор роли (целое число, FK к таблице ролей пользователей)

В созданную таблицу добавить данные о пяти пользователях.

5.1.3 Создать модели данных (роли, пользователи, билеты, сеансы, фильмы) и контекст БД, при создании указать имя контекста, создать директории для модели и контекста. Сервисы добавлять по мере необходимости с требуемыми методами.

5.1.4 Создать клиентское приложение WPF, на главном окне которого отобразить список названий фильмов.

### **5.2 Авторизация пользователя**

5.2.1 Добавить в приложение форму авторизации с полями ввода логина и пароля, кнопками «ОК» и «Отмена». Реализовать ее открытие при нажатии на кнопку «Войти» на главной форме).

5.2.2 При нажатии на «Отмена» закрывать форму авторизации.

5.2.3 При нажатии на «ОК» требуется проверять, существует ли пользователь с указанными логином и паролем в таблице пользователей, используя SingleOrDefault() с условием и AsNoTracking().

Если пользователь существует, то приветствовать его и перенаправлять на главную форму, иначе — сообщать, что введены некорректные данные и запрашивать повторный ввод.

### 5.3 Хранение данных сессии пользователя

#### 5.3.1 Добавить в приложение класс UserSession с:

- открытым свойством CurrentUser типа Пользователь (Пользователь — тип данных из модели), сделать set приватным,
- открытым методом void SetCurrentUser(пользователь) для присваивания значения свойству CurrentUser,
- открытым методом void Clear() для присваивания значения null свойству CurrentUser.

5.3.2 Для того, чтобы сделать класс UserSession реализующим паттерн синглтон, добавить в класс следующий код:

```
private static readonly UserSession _instance = new();

private UserSession () {}

public static UserSession Instance => _instance;
```

Для использования методов класса UserSession использовать следующий код:  
UserSession.Instance.Метод(...)

5.3.3 При авторизации полученный пользователь должен записываться в CurrentUser, а на главной форме должен отображаться его логин.

5.3.4 После авторизации отображать кнопку «Выйти», при нажатии на которую очищать данные пользователя.

### 5.4 Разграничение прав доступа

5.4.1 На главную форму добавить кнопки «Личный кабинет», «Проверка билета», «Пользователи» и реализовать переход с главной формы к формам с соответствующими заголовками (данные из БД можно не отображать).

В зависимости от роли пользователя CurrentUser на главной форме должны отображаться различные кнопки для перехода к соответствующим формам:

- Личный кабинет — для посетителя
- Проверка билета — для билетера
- Пользователи — для администратора

5.4.2 Реализовать возможность отображения списка логинов и ролей пользователей для администратора.

5.4.3 Реализовать возможность отображения логина и пароля пользователя с возможностью смены пароля.

5.4.4 Реализовать возможность проверки, что билет есть в БД по его номеру для билетера и вывода номера ряда и места указанного билета.

## 5.5 Создание формы регистрации

5.5.1 Добавить в приложение форму регистрации с полями ввода логина, пароля и подтверждения пароля, кнопками «ОК» и «Отмена». Переход к форме регистрации должен быть реализован с формы авторизации при нажатии на кнопку «Зарегистрироваться»

5.5.2 При нажатии на «Отмена» закрывать форму регистрации.

5.5.3 При нажатии на «ОК» требуется проверять, существует ли пользователь с указанными логином в таблице Users.

Если существует, сообщать, что указанное имя занято.

Если не существует, записывать данные о новом пользователе в БД (роль для пользователей по умолчанию — Посетитель) и перенаправлять на форму Авторизации.

Пустой логин недопустим.

## 6 Порядок выполнения работы

6.1 Выполнить все задания из п.5.в одном проекте LabWork20.

6.2 Ответить на контрольные вопросы.

## 7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

## 8 Контрольные вопросы

8.1 Как изменить настройки подключения к БД в клиентском приложении?

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

8.3 Что такое «авторизация»?

8.4 Что такое «регистрация»?