Практическая работа №21
Реализация разграничения прав доступа пользователей

1Цель работы

1.1 Научиться разграничивать права доступа пользователей на уровне интерфейса приложения;

1.2 Научиться изменять настройки подключения к БД.

2 Литература

2.1 Фленов, М. Е. Библия C#. – 3 изд. – Санкт-Петербург: БХВ-Петербург, 2016. – URL: https://ibooks.ru/bookshelf/353561/reading. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.16.

3 Задание

Работа должна выполняться в оконном приложении, в котором создана модель EDM для подключения к БД.

3.1 Создание таблиц с пользователями и ролями пользователей в БД

3.1.1 Добавить в БД таблицу UserRoles:

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

Созданную таблицу заполнить значениями Администратор, Продавец, Покупатель

3.1.2 Добавить в БД таблицу Users:

- 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 UserRoles)

В созданную таблицу добавить данные о пяти пользователях.

3.1.3 Обновить модель EDM в существующем клиентском приложении (или создать новое приложение, в модель которого включить UserRoles и Users).

3.2 Создание формы авторизации

3.2.1 Добавить в приложение форму авторизации с полями ввода логина и пароля, кнопками «ОК» и «Отмена».

Способ запуска формы авторизации на свое усмотрение (например, перед запуском главной формы или при нажатии на кнопку «Авторизоваться» на главной форме).

3.2.2 При нажатии на «Отмена» закрывать форму авторизации.

3.2.3 При нажатии на «ОК» требуется проверять, существует ли пользователь с указанными логином и паролем в таблице Users, используя LINQ (метод FirstOrDefault() с условием). Требуется учитывать регистр пароля и не учитывать регистр логина.

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

3.3 Разграничение прав доступа

3.3.1 Добавить в приложение статический класс Connection с открытым полем (или свойством) CurrentUser типа Пользователь (Пользователь — тип данных из модели).

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

3.3.3 На главную форму добавить кнопки «Товары», «Заказы», «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:

- Товары — для покупателя
- Товары и Заказы — для менеджера
- Товары, Заказы и Пользователи — для администратора

3.3.4 На форму «Товары» добавить панель инструментов или меню или панель с кнопками «Удалить», «Добавить», «Сохранить».

Кнопки для модификации БД должны быть доступны только пользователю с ролью менеджер.

3.4 Создание формы регистрации

3.4.1 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»

3.4.2 При нажатии на «Отмена» закрывать форму регистрации.

3.4.3 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 пользователе в БД (роль для пользователей по умолчанию — Покупатель) и перенаправлять на форму Авторизации.

3.5 Создание интерфейса для настройки подключения к БД через Entity Framework

3.5.1 Добавить в класс Connection открытые строковые поля для хранения имени сервера, БД, логина и пароля пользователя, присвоить им значения по умолчанию.

3.5.2 Добавить в класс Connection открытый метод / свойство только на чтение для возврата строки подключения к БД MSSQL. Для реализации использовать значения полей класса и SqlConnectionStringBuilder.

3.5.3 Создать в классе Core метод смены строки подключения контекста, в котором указать следующий код:

context.Database.Connection.ConnectionString = *строка подключения*;

Вместо *строка подключения* вызвать метод/свойство, формирующий строку подключения.

3.5.4 Добавить в приложение форму, отображающую в полях ввода названия сервера, БД, логин и пароль пользователя (значения заполняются на основе значений соответствующих свойств класс Connection). При нажатии на кнопку «Сохранить» значения свойств класса Connection должны изменяться на указанные в полях ввода.

Реализовать переход к форме настроек с главной формы приложения.

4 Порядок выполнения работы

4.1 Выполнить все задания из п.3 в одном проекте PractWork21.

4.2 Ответить на контрольные вопросы.

5 Содержание отчета

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

6 Контрольные вопросы

6.1 Как изменить настройки подключения к БД в клиентском приложении?

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

6.3 Что такое «авторизация»?

6.4 Что такое «регистрация»?