**Проектная документация**

1. Функциональное назначение

Это приложение помогает управлять пользователями. В нём можно:

* Войти в систему (ввести логин и пароль)
* Поменять пароль
* Управлять пользователями через панель администратора
* Работать с данными пользователей, которые хранятся в базе данных

2. Описание используемых методов

2.1 Модель пользователя (Models/User.cs)

Класс: User  
Это описание, как выглядит один пользователь. У него есть:

* Id — уникальный номер
* Username — имя пользователя
* Password — пароль (обычно хранится в зашифрованном виде)
* Role — роль (например, админ или обычный пользователь)

2.2 Работа с базой данных (Data/DataAccess.cs)

Класс: DataAccess  
Отвечает за связь с базой данных. Вот что он умеет:

* GetUserByUsername(username) — найти пользователя по имени
* UpdateUserPassword(userId, newPassword) — поменять пароль
* GetAllUsers() — получить список всех пользователей

2.3 Окно входа (Forms/LoginForm.cs)

Класс: LoginForm  
Проверяет логин и пароль пользователя при входе.  
Метод:

* btnLogin\_Click(...) — срабатывает, когда нажимают кнопку "Войти"

2.4 Окно смены пароля (Forms/ChangePasswordForm.cs)

Класс: ChangePasswordForm  
Позволяет пользователю сменить пароль.  
Метод:

* btnChangePassword\_Click(...) — срабатывает при нажатии кнопки "Сменить пароль"

2.5 Панель администратора (Forms/AdminDashboard.cs)

Класс: AdminDashboard  
Даёт администратору доступ к управлению пользователями.  
Методы:

* LoadUsers() — загружает список пользователей
* btnAddUser\_Click(...) — добавляет нового пользователя

2.6 Конфигурация (App.config)

Файл с настройками — хранит параметры подключения к базе данных и другие настройки приложения.

3. Взаимодействие компонентов

1. Пользователь запускает программу и входит через окно входа.
2. Если нужно — может поменять пароль.
3. Администратор может открывать панель для управления другими пользователями.
4. Все действия с пользователями идут через DataAccess, который работает с базой данных.