**1. Группы пользователей (роли):**

* **Администратор** — управляет системой, пользователями, доступом, настройками.
* **Мастер** — работает с заявками, фиксирует ремонт, ставит статусы.
* **Клиент** — создает заявки, отслеживает статус своих заказов.
* **Оператор/Менеджер** — контролирует процесс ремонта и отчёты.

**2. Принципы регистрации:**

* **Клиенты** регистрируются через форму с вводом: ФИО, телефон, email, пароль.
* **Мастера, администраторы, менеджеры** создаются администратором вручную в базе (с помощью интерфейса админ-панели или SQL).
* Логин — уникальный идентификатор пользователя.
* Пароль — хранится в базе в виде **хэша** (например, SHA-256 с солью).

**3. Система паролей:**

* Пароли **никогда не хранятся в открытом виде**.
* Использовать криптографические хэши с солью.
* При регистрации пароль хэшируется и сохраняется.
* При входе введённый пароль хэшируется и сравнивается с сохранённым хэшем.
* Можно внедрить правила сложности пароля (минимум 8 символов, обязательные цифры и буквы).
* Возможность восстановления пароля через email или смену администратором.

**4. Таблица пользователей (повторно, с деталями):**

CREATE TABLE Пользователи (

ID\_Пользователя INT IDENTITY(1,1) PRIMARY KEY,

Логин NVARCHAR(50) NOT NULL UNIQUE,

ПарольHash NVARCHAR(256) NOT NULL,

Соль NVARCHAR(50) NOT NULL,

Роль NVARCHAR(20) NOT NULL CHECK (Роль IN ('Администратор', 'Мастер', 'Клиент', 'Менеджер')),

Дата\_регистрации DATETIME DEFAULT GETDATE(),

Активен BIT DEFAULT 1 -- статус аккаунта

);

**5. Пример логики регистрации (на уровне приложения):**

* Пользователь вводит данные.
* Генерируется соль (например, случайная строка).
* Пароль + соль хэшируются.
* В базу записываются логин, хэш пароля, соль, роль.
* При авторизации из БД берется соль и хэш, входящий пароль хэшируется с этой же солью и сравнивается.