**Epic**

**1. User Endpoint Management**

**TODO Description**

Within this endpoint we will have the following functionalities:

* Pull User information
* Store User information
* Create User table
* Update User information

**2. Database Management**

**TODO Description**

This epic is going to be responsible on handling all database structure management and changes

* ApplicationDbContext (Separate ClassLib)
* UnitOfWork for the DbContext
* Repositories
* Wiring the AppDbContext to the startup class

**Issues**

**1. Create a DbContext ClassLib**

Create a DbContext ClassLib which will be responsible on all db related tasks.

* Name: SohatNotebook.DataService
* Packages: All EfCore and Sqlite packages

Add DataService to the Api as reference

**2. Add User Entity**

- First Name

- Last Name

- Email

- Phone

- Date of Birth

- Country

**3.Create a Users Controller**

Create 3 Endpoints

**- Get : Get user information based on id of type guid**

\* Return a user dto after it has successfully got the user info

\* Return status is 200

**- Post : Add new user: it will take a dto of user**

\* Return a user dto after it has successfully got the user info

\* Return status is 201

**- Get All users: it does not take any params**

\* Return a list of user dto

\* Return status is 200

**4. Implement Unit Of Work Pattern and Repository Pattern in the Application**

1. We need to create the IRepository Interface

2. Create a Generic Repository

3. Create the IUnitOfWork Class

4. Create the UnitOfWork Class

5. Add dependency injection in the startup class for the UoW

6. Connect the Controller to the UoW

7. Create a Repository for Users

8. Update the UsersController to use the UoW instead of direct context initialization