

SimpleNote API

PURPOSE:

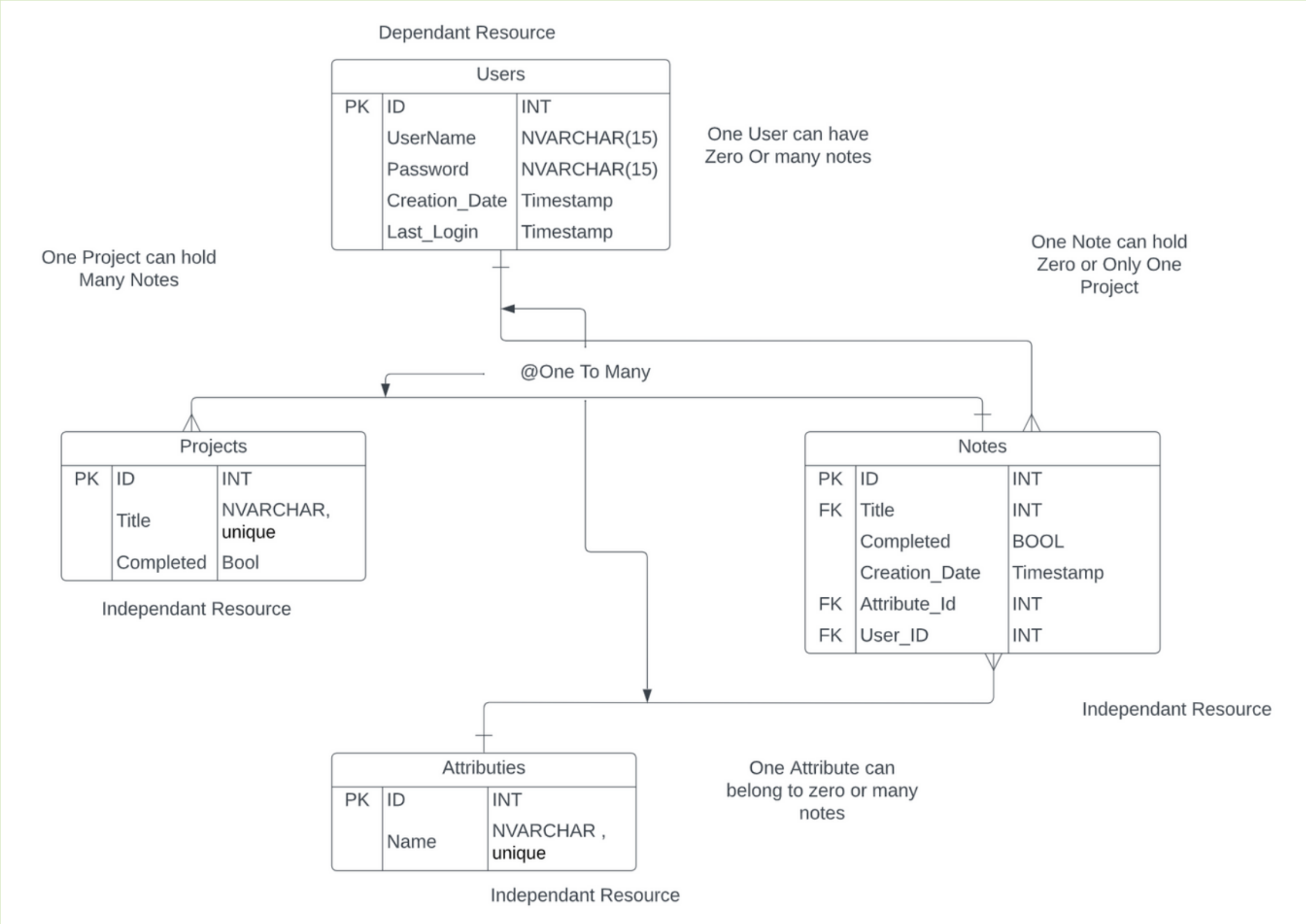
Registration system for users.

Users can create projects and add notes for them

e.g: creating a Selling app project, so I create project and add notes with attributes

By Tahmina Munni

The Database:



API Endpoints: (Page - 1)

HTTP Methods	Description	Request Body	Response Body
GET/Login	Requesting to login in to the system	UserName,Password	After getting authenticated, The respond status code is either 401 or 200
GET / Logout	User will request logging out and that attached with his token	Token	status code: Success: 200 failed: 400
POST/Note	This endpoint is to create a new note with passing all the required attributies.	(token, noteid, NoteText, Project, Attribute[])	status code: Success: 201 failed: 400
Update/Note	This endpoint is to update an existing note with passing all the required attributies.	(token, noteid, NoteText, Project, Attribute[])	status code: Success: 200,204 failed: 400
Delete/Note	This endpoint is to delete an existing note with an id	Id	status code: Success: 200,204 failed: 400

API Endpoints: (Page - 2)

HTTP Methods	Description	Request Body	Response Body
POST/Project	This endpoint is to create a new note with passing all the required attributies.	(title)	status code: Success: 201 failed: 400
Update/Project	This endpoint is to update an existing project with passing all the required attributies.	(token, noteld,completed)	status code: Success: 200,204 failed: 400
Delete/Project	This endpoint is to delete an existing Project with an id	id	status code: Success: 200,204 failed: 400
GET/ProjectNotesCounts	Returns a count of notes for each project, an a count of notes not belonging to any project	(token)	status code: Success: 200,204 failed: 400
GET/AttributeNoteCount	Returns a count of notes for each attribute/without attribute.	(token)	status code: Success: 200,204 failed: 400

SQL Commands for Endpoints:

Select * FROM notes WHERE attribute_Id = NULL

```
namespace SimpleNote_WebAPI.Infrastructure.IRepositoriesImp
{
    public class NotesRepoImp : GenericRepoImp<NotesEntity>, INotesRepo
    {
        public NotesRepoImp(ApplicationDbContext db) : base(db)
        {
        }

        public async Task<IEnumerable<NotesEntity>> GetNotes(int? projectId, int? AttributeId)
        {
            var Allnotes = await GetAllAsync();

            if (projectId.HasValue && AttributeId == null)// returning all the notes that has this project id only
            {
                var notes = from note in Allnotes
                            where note.Project_ID_FK == projectId
                            select note;

                return notes;
            }

            else if(AttributeId.HasValue && projectId == null)// returning all the notes that has this project id onl
            {
                var notes = from note in Allnotes
                            where note.Attribute_Id_FK == AttributeId
                            select note;

                return notes;
            }
            else
            {
                return Allnotes;
            }
        }
    }
}
```

Concepts

1. ORM (Object Relational Mapping) >> (Entity Framework)EF
2. ERD >> Entity Relational Diagram
3. Database connection: MSSQL Database
4. Security : JWT
5. Interfaces and Generics
6. Collection Library
7. LINQ Library
8. DDD architecture (Domain Driven Design)
 - Infrastructure Layer
 - Service Layer
 - Controller Layer
 - DTOS Layer (Data Transfer Objects)

*Thank
you!*