**.NET**

**Find Property System**

***Question***

***MODULE LIST***

***MODULE LIST and MODULE DETAILS:***

***1. Owners***

***2. Customers***

***3. Property***

***4. Registration & Login***

***• Authorization***

***• Authentication***

***REQUIREMENT***

***There is no functional change on this sprint. In this sprint, you have to use:***

***- Angular instead of Console***

***- Use Entity Framework instead of ADO.NET***

***- Create Web API and host separately.***

***- Functionality will be similar to services logic.***

***- Unit Testing using Microsoft Test Framework***

***Create Client Application in Angular and Services using ASP.NET Core Web API***

**Project Structure: (Description of the Modules)**

* Angular Client Application
* This will be the front-end application developed using Angular.
* It will provide the user interface for both owners and customers to interact with the property find system.

**ASP.NET Core Web API**

* This will serve as the backend of the application, providing RESTful APIs for communication between the Angular client application and the database.
* It will handle requests from the client application, process data, and interact with the database using Entity Framework Core.

**Entity Framework Core**

* This will be used for database access and ORM (Object-Relational Mapping) in the ASP.NET Core Web API.
* It will handle the mapping between database entities and C# objects.

**Module Details:**

***Owners Module***

* Allow owners to register and log in.
* Provide functionality for owners to manage their properties, including adding new properties, updating property details, and viewing property listings.

***Customers Module***

* Allow customers to register and log in.
* Provide functionality for customers to search for properties based on various criteria, view property details, and contact property owners for inquiries.

***Property Module***

* Provide functionality for registering properties on the platform.
* Allow property owners to update property details such as description, location, price, and availability status.
* Implement search functionality for customers to find properties based on different criteria.

**Registration & Login Module**

* Handle user registration and authentication processes.
* Ensure secure user authentication and authorization to access the system.

**Authorization Module**

* Manage user roles and access permissions within the system.
* Define access control policies to restrict unauthorized actions.

**Authentication Module**

* Validate user credentials during the login process.
* Generate authentication tokens for secure access to protected resources.

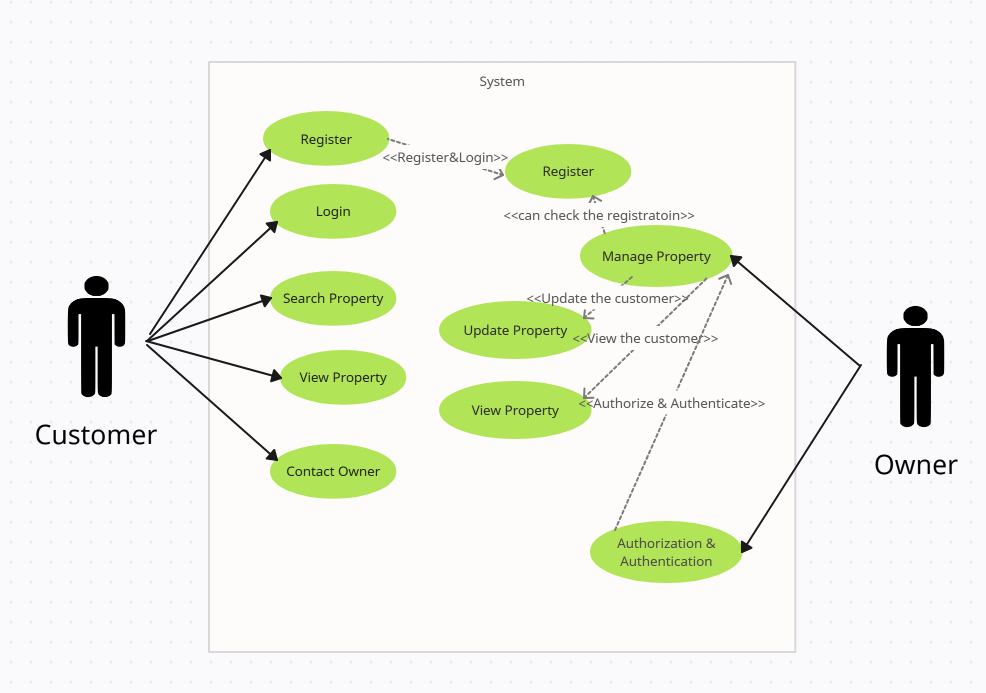
**Sprint Requirement:**

* Develop the Angular client application to replace the console application.
* Utilize Entity Framework Core instead of ADO.NET for database access.
* Create a separate ASP.NET Core Web API project to host the services.
* Implement the same functionality as in the previous sprint but with Angular for the client application and ASP.NET Core Web API for the services.
* Write unit tests using the Microsoft Test Framework to ensure the reliability of the code.

**Summary:**

This sprint focuses on transitioning the property find system from a console application to an Angular-based client application communicating with an ASP.NET Core Web API backend. The functionality remains the same, but the technologies used for development are updated to Angular, Entity Framework Core, and ASP.NET Core Web API. Additionally, unit tests will be written to verify the correctness of the implemented functionalities.

Top of Form

****

In this diagram:

Customer can register, login, and search for properties.

**Registration & Login** module allows users to register and login to the system.

**Owner** can manage, update, and view properties they own.

**Authorization** controls access to functionalities based on user roles.

**Customer** can search for properties, view property details, and contact property owners.

**Property** module encompasses actions related to properties such as managing, updating, viewing, and searching properties.

This diagram outlines the interactions between actors and modules in the property find system.Top of Form

**Property** module encompasses actions related to properties such as managing, updating, viewing, and searching properties.