EUROPEAN DYNAMICS

Microsoft® .NET

Coding Bootcamp for Junior .NET Developers

120 Hours

Code.Hub

Individual Project Assignment

C# and .NET Bootcamp

Contents

Code.Hub

Technico
- Repair Management Project

Business Case Description

This project aims to develop a Home Repair and Renovation Service Platform that connects homeowners (apartment or house owners) with qualified professionals who offer repair, maintenance, and renovation services. The platform will act as a marketplace, allowing customers to request services such as plumbing, electrical work, carpentry, painting, and full-scale renovations. It will streamline the process of finding, hiring, and managing service providers, ensuring high-quality work, transparency in pricing, and convenient scheduling for homeowners.

The home repair and renovation market is vast, with homeowners regularly seeking professional help for minor repairs and larger projects. The global home improvement services market was valued at $316 billion in 2021 and is projected to grow steadily due to increased homeownership, urbanization, and renovation trends. Many homeowners, especially in urban areas, prefer using digital platforms to find services, making this an ideal time to launch such a solution.

The web app will be called Technico. Your client wishes to create a proof of concept of the application, with separate screens for the Project Creator and the backer (authorization not required for the proof of concept), enabling him to showcase its usage to Angel Investors to receive funding.

Project Requirements

The project is an <u>individual assignment</u>, the Technico application. In a second team project , the implementation will take place in two parts, namely the back and front end. In this individual project assignment only the back end will be implemented as a console application.
The functional and non-functional requirements for the project are given.

Functional requirements

There are three basic business entities, the Property-Owner, the Property-Item and the Repair.

Property-Owner services

1.   The self-registration service requires the following fields:
    • VAT number (ΑΦΜ), which is a unique identifier that characterizes users)
    • Name
    • Surname
    • Address
    • Phone number
    • E-mail (used as Username)
    • password
    • type of user
2.   The display service allows displaying the following that concern the current user:
    • Property Owner details
    • Property items details
    • Repairs details
It is expected to align with the GDPR requirements.
3.   The service also includes the following options: Update, Delete.

Property-Item services

It will contain the following services: View, Create, Delete, and Update.
View: Displays all the details of the property.
Create: Form with the following fields:
    •   A Property Identification Number, which coincides with the corresponding number of E9 and will uniquely characterize the property,
    • Property address,
    • Year of construction,
    • Type of property (Detached house, Maisonet, Apartment building), (Note: this field is now removed by the owner
    • VAT number of its owner.
Update (nice to have): All the details of the property can be edited.
Delete: The property can be permanently deleted or deactivated.

Property-Repair services

It will contain the following services: Search, Create, Delete, and Update.

Create:

- Date (datetime) of the scheduled repair
- Type of repair (Painting, Insulation, Frames, plumbing, electrical work)
- Repair description as a free-text field for the work to be done (e.g., installation of a solar water heater)
- Repair address
- Status of the repair (Pending, In progress, Complete - default is the pending status)
- Cost of repair
- Owner for whom the repair is made

Update: it will be possible to update the repair details (e.g., status)

Delete: The repair can be permanently deleted or deactivated.

Nice to have

- Validations, for example, a user with a blank VAT number cannot be entered.
- Property-items may belong to more than one property-owner.

Non-functional requirements

1. Use of .NET core programming stack and MS-SQL server
2. Use of EF Core
3. Use of DI and Clean Architecture
4. Use of C# Coding Conventions and .NET Architectural Standards
5. Source code delivered on **GitHub**

Criteria of assessment

Backend development

1. Design of the Model, Code First Approach
2. Definition of project's Services in Interfaces.
3. Completeness of project's functionality.
4. Testing scenarios

Git usage

Development progress should be depicted with commits which have clear and descriptive commit messages

References

https://www.alliedmarketresearch.com/press-release/home-improvement-services-market.html

Code.Hub