



## **Technico**

### **The technicians on the Web**

Requirements specification document

**November 2024**

## Project general description

A Renovation Contractor Agency, **Technico**, within the framework of its operation, needs a web application that will enable the employees - managers of its platform to have access to information concerning customers and repairs. It will also enable its customers to oversee the progress of repair / renovation work on their property.

## User roles

- Admin (**Employee of the Agency**)
- Property Owner, User of the platform, Customer

## Functional requirements

### Property Owners' Page

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

**Create Owner of the Property:** A form will appear with the following fields:

- VAT number (AΦM), which is a unique identifier and characterizes users,
- Name,
- Surname,
- Address,
- Phone number,
- Email,
- Username
- Password,
- Property type (Detached house, Maisonette, Apartment)

**Search:** It will contain the following fields to search:

- VAT number
- Email

**Update:** On this page, the complete details of a user will be displayed, and the administrator will be able to modify them. The navigation on the page can be done with a corresponding link on the search page.

**Delete:** Through the update page, a special button will be provided that will allow this function. It is recommended to press the JavaScript window to confirm the action at the touch of a button.

**Note:** The Update, Delete options can be found next to the search results for each user / owner.

### Property Page

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

**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, Maisonette, Apartment building), (Note: this field is now removed by the owner)
- VAT number of its owner.

**Search:** The administrator will be able to search for properties based on various criteria, but mainly

- Property ID Number
- VAT number

**Update:** Similar to Owner's page.

**Delete:** Similar to Owner's page.

Nice to have

- Validations via JavaScript for data entry forms. For example, a user with a blank VAT number cannot be entered.
- Validations in Backend when entering data: For example, you cannot enter a user with existing email or VAT number
- Owner has more than one properties.
- Authentication & Authorization: After their identification, depending on their type, ordinary users should not have access to the administrators' pages (i.e., changing the URL).

## Property Repair Page

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

**Create:**

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

**Update:** As in the Owner page, where it will be possible to update the repair details (e.g., status)

**Search:** Based on

- Date or Range of dates
- User ID in case we want to display all the repairs made for a property owner.

**Delete:** Through the update page, a special button will be provided that will allow this function. It is recommended to press the JavaScript window to confirm the action at the touch of a button.

### Login Page

The user will enter his email and password and depending on his type / role, will be redirected to one of the following 2 pages

1. Property Owner (User) Home Page
2. Administrator Home Page

### Home Pages

- The **homepage** of the ordinary user should show only the repairs that concern him.
- The **administrator homepage** should display the next 10 repairs to be made, based on the day and time that will be set.

### The administrator Homepage

It will present the active repairs of the day (ongoing repairs) and will allow administrators to navigate to the following pages:

- Property Owners
- Repairs

### Property Owners (admin view)

For user management - property owner the following pages should be created:

1. Create a property owner with the corresponding form (create-owner-page)
2. Owner data processing page. Stylistically it will look like creating an owner just the fields will be pre-filled. In addition, there will be a button which when pressed will delete the corresponding Owner. (edit-owner- page)
3. User search page: A form with the search criteria mentioned and below a table with the results (initially empty on the 1st visit to the page). (search-owner- page)

### Repairs (admin view)

The corresponding pages for the repair are defined in the same way. Specifically:

1. Create-repair- page
2. Edit-repair- page
3. Search-repair- page
4. Create-property- page (nice to have)
5. Edit-property- page (nice to have)
6. Search-property- page (nice to have)

## Non-functional Requirements

1. Programming environment
  1. Java 21
  2. Database: MySQL Server
  3. Spring Boot
  4. Angular
2. Code first approach
3. Dependency Injection is required
4. Coding and architectural standards and conventions
5. VCS: Project's source code must be delivered on GitHub. Descriptive and clear commit messages depict each team member's contribution.

## Milestones

The milestones are given mainly as a logical separation of the various tasks and will help us to allocate the different stages of implementation in time. It is not obligatory to observe them, but it is suggested that they be followed, so that there is an indication of the progress of the deliverable.

- It is suggested to breakdown the tasks to the various members of the team in a way that will be discussed (functionality, component etc) and the use of Trello board for better management and supervision.
- Preparation and implementation of domain entities and consequently the shape of the database and the structure of the webapp and import into an IDE (e.g., IntelliJ). Also, creation of the GitHub repository and the accounts of the team members.
- Implementation of the home pages of Admin and Property Owner / User
- Implement import / search / modification / deletion
- Property Owners and Repairs by Admin.
- Implementation of login functionality, security features.
- Implementation of the required functions for repairs and nice-to-have functionalities.