Code.Hub

Technikon
The technicians on the Web

Requirements specification document

September 2022

Code.Hub

Contents

Code.Hub

1. Project general description

A Renovation Contractor Agency, Technico, within the framework of its operation, needs an 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. All the data are stored in an appropriate database.

2. User roles

The roles of the application users are
Admin (Employee of the Agency)
Property Owner, User of the platform, customer

3. Functional requirements

3.1 Property-Owners' Functionalities

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

Create Owner of the Property with the following attributes:

- VAT number (ΑΦΜ), which is a unique identifier that characterizes users),
- Name,
- Surname,
- Address,
- Phone number,
- Email,
- Username
- password

Search: It will contain the following fields to search:

- VAT number
- email

Update: the address, email, password can be changed.
Delete: the owner can be safely deleted.

3.2 Property Functionalities

It will contain the following functions: Search, Create, Delete, Update.
Create a Property class with the following attributes:

- A Property Identification Number, which coincides with the corresponding number of E9 and will uniquely characterise 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: Can change any wrongly inserted data.

Delete: the property can be safely or permanently deleted.

Business rules
Validations when entering data: For example, you cannot enter another user with existing email or VAT number. Exceptions will be thrown.
Note that an Owner may have more than one property.

3.3 Property-Repair functionalities

It will contain the following functions: Search, Create, Delete, Update.
Create a Property-Repair class with the following attributes:

- Date (datetime) of the scheduled repair
- Short Description of repair,
- Type of repair (Painting, Insulation, Frames, plumbing, electrical work),
- Status of the repair (Pending, In progress and Complete - default standby mode)
- Cost of repair,
- Owner id for whom the repair is made,
- Property id for which 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: Can change any wrongly inserted data

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: the property repair can be safely deleted.
Note that there may be more than one repairs to one property.

.

4. Use cases

4.1 Data population

Create a few data to insert into the database. The data will be owners, properties, and repairs.

4.2 Usage simulation (testing)

Create a property-owner object with the all the necessary details. Test all interesting cases with appropriate calls.

5. Non-functional Requirements

- Eclipse Programming environment
- Java SE 17
- Database: MySQL Server
- Dependency Injection is strongly recommended
- Coding and architectural standards and conventions
- VCS: Project's source code must be delivered on GitHub for each team member's contribution depicted by commits with clear and descriptive messages. Use branches for each feature development.

6. 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.

- Break It is suggested breakdown of 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. Also, creation of the GitHub repository and the accounts of the team members.
- Implementation of the domain classes and corresponding tables in database
- Implement the CRUD functionalities in the repository classes, i.e., create / search / modification / deletion
- Implementation of the required functions for repairs and nice-to-have functionalities.
- Scenarios in use cases