UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Commercial Space Management and Leasing

Name: Pantea Tania
Group: 30234

Table of Contents

Deliverable 1

Project Specification

This project is a Spring Boot application designed to manage commercial spaces and related entities such as owners, tenants, and rental contracts. The system enables users to create, retrieve, update, and delete (CRUD) commercial spaces while maintaining relationships between different entities.

*The application follows the **MVC (Model-View-Controller) architecture**, ensuring a clear separation of concerns between data management, business logic, and user interaction. An in-memory repository is used for storing data, making it easy to expand with a database in the future.*

Additionally, mock data generation is implemented to populate the system with sample owners, tenants, and rental contracts at startup. The project is structured to allow future enhancements, such as database integration and additional business rules.

Functional Requirements

- Add, update, and delete commercial spaces.
- Assign owners and tenants to commercial spaces.
- View a list of all commercial spaces.
- Automatically generate mock data on startup.
- Handle errors gracefully for missing or invalid data.

Use Case Model

Use Cases Identification:

1. Create Commercial Space

- Level: User goal
- Primary Actor: Owner
- Main Success Scenario: The owner fills in the form to create a new commercial space and submits it. The system saves the commercial space and displays it in the list of commercial spaces.
- Extensions: If the admin doesn't provide valid data, an error message appears.

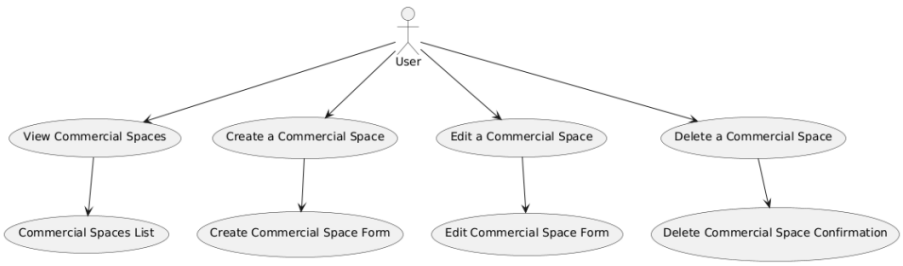
2. Edit Commercial Space

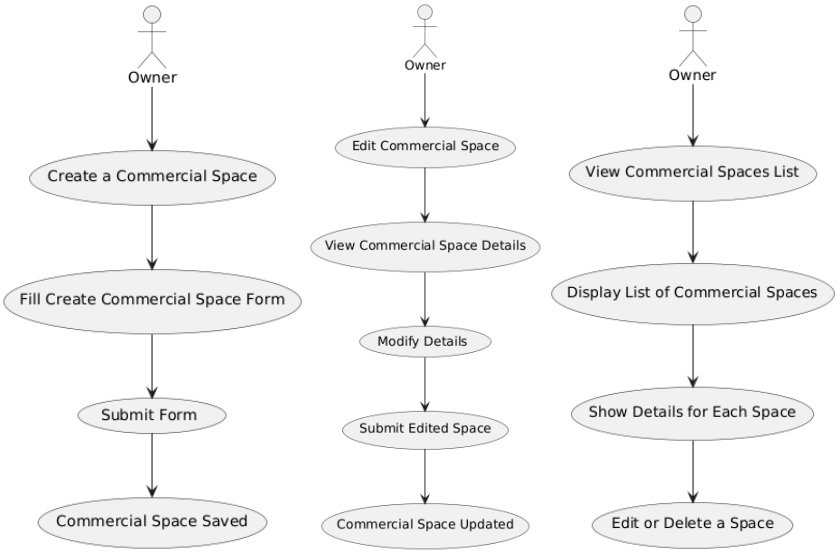
- Level: User goal
- Primary ActorOwner
- Main Success Scenario: The owner selects a commercial space to edit, makes changes, and submits the form. The system updates the commercial space information.
- Extensions: If the commercial space does not exist, an error message is shown.

3. Delete Commercial Space

- Level: User goal
- Primary Actor: Owner
- Main Success Scenario: The owner selects a commercial space to delete, confirms the action, and the system removes the commercial space from the list.
- Extensions: If the commercial space does not exist, an error message is displayed.

UML Use Case Diagrams





Supplementary Specification

Non-functional Requirements

1. ***Leasing Functionality***
- *The system currently does not allow users to lease commercial spaces. Leasing logic needs to be implemented, including linking tenants to spaces, visible to both owners and tenants.*
2. ***Visibility for Owners and Tenants***
- *Users should see the spaces they own or lease, but this feature is not yet implemented. All users can currently view all commercial spaces.*
3. ***Limited Controllers***
- *Only a controller for managing commercial spaces exists. Controllers for managing owners, tenants, and rental contracts are not yet implemented.*
4. ***Repository Usage***
- *Repositories are only used for commercial spaces. No logic has been implemented for managing or interacting with Owner, Tenant, or RentalContract entities using their respective repositories.*

Design Constraints

The system is being developed using Java with the Spring Framework, following the MVC (Model-View-Controller) design pattern for clear separation of concerns. The use of Spring Boot and Thymeleaf templates has been mandated for the web application. The repository pattern is employed for data management, and the application relies on in-memory storage for commercial spaces and related entities. Additionally, the system must follow the current structure of models and controllers for each entity.
The application is designed to be extendable for future features, such as leasing functionality and role-based access control, but must remain modular.

Glossary

1. **Commercial Space**: A property used for business purposes like a shop, office, warehouse, or restaurant.
2. **Owner**: A person or entity that owns a commercial space.
3. **Tenant**: A person or entity that rents or leases a commercial space from an owner.
4. **Leasing**: The process of renting a commercial space from the owner.

5. **Rental Contract**: An agreement between the owner and tenant outlining the terms of leasing a space.
6. **MVC**: A software pattern separating data (Model), display (View), and input handling (Controller).
7. **Repository**: A pattern for managing data access for entities like spaces, owners, and tenants.

Deliverable 2

Domain Model

[Define the domain model and create the conceptual class diagrams]

Architectural Design

Conceptual Architecture

[Define the system's conceptual architecture; use an architectural style and pattern - highlight its use and motivate your choice.]

Package Design

[Create a package diagram]

Component and Deployment Diagram

[Create the component and deployment diagrams.]

Deliverable 3

Design Model

Dynamic Behavior

[Create the interaction diagrams (2 sequence) for 2 relevant scenarios]

Class Diagram

[Create the UML class diagram; apply GoF patterns and motivate your choice]

Data Model

[Create the data model for the system.]

System Testing

[Describe the testing methides and some test cases.]

Future Improvements

[Present some features that apply to the application scope.]