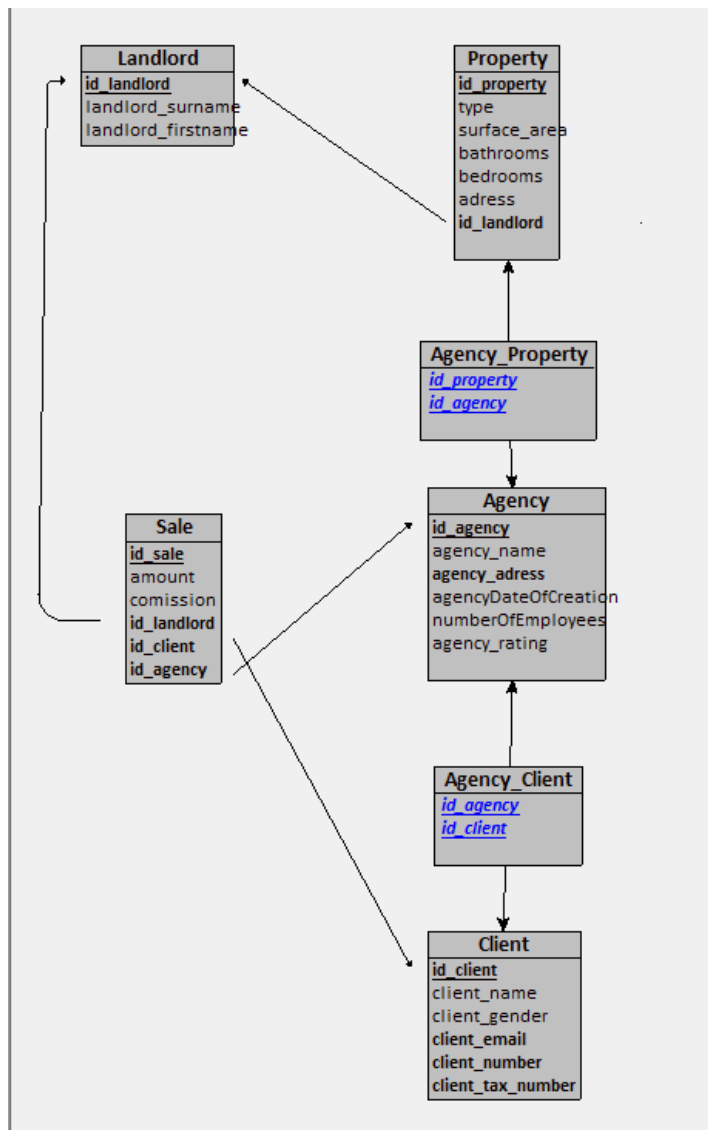**Members:** Sulpice Alban, Abouleila Selim

Title: UML DIAGRAMS

The diagram appears to be a flowchart or process loop that emphasizes repetitive steps within a cycle. It shows how a certain process or set of actions feeds into itself in a continuous loop, where each action is connected to the next, creating a cycle of actions or feedback. This could relate to various systems or processes, such as quality control, iterative processes, or systems with ongoing feedback mechanisms. Without more context, it's a bit hard to specify the exact nature of the process, but the looping structure indicates a recurring, cyclical activity.

Agency Project Use Case - ALBAN

This is a **UML Use Case Diagram** illustrating the interactions between three main actors—**Client**, **Landlord**, and **Agency**—within an agency project.

Relations:

- Use cases often include dependencies such as:

    - <<include>> relationships signify mandatory steps or sub-actions (e.g., "View Properties" includes "Add to Favorite").

This structured diagram maps out the system functionality for all involved stakeholders.
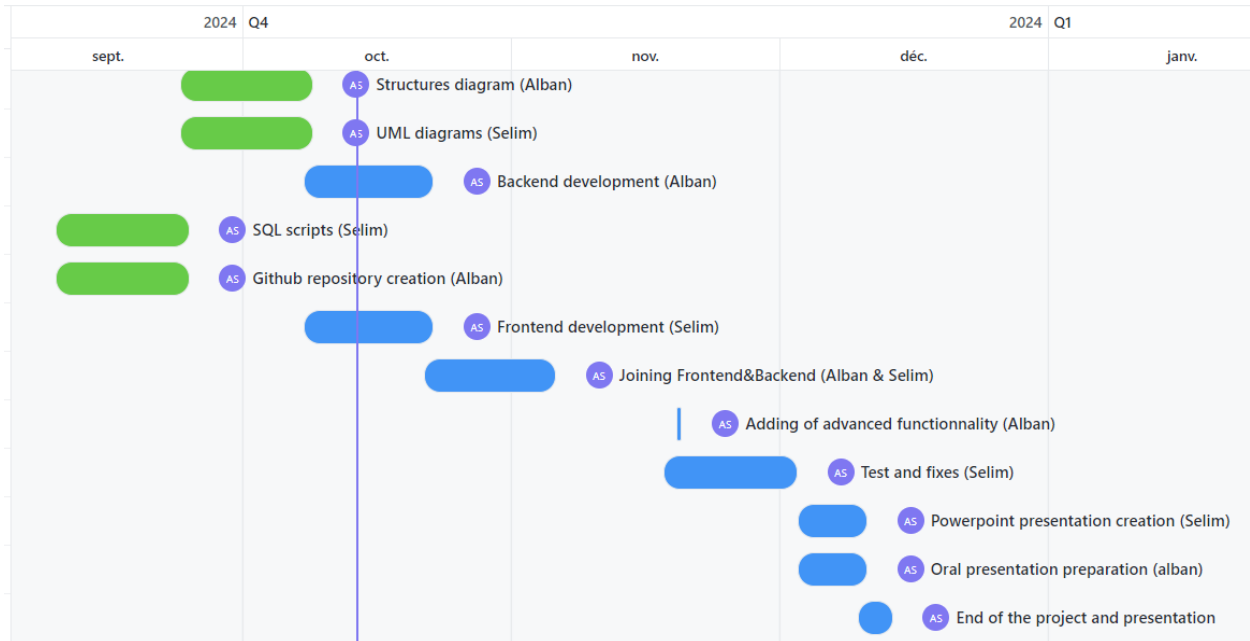
This is a **Gantt chart** that outlines a project timeline for the **Agency Project** from September 2024 to January 2025. It organizes tasks assigned to two team members, **Alban** and **Selim**, and indicates their collaboration phases.
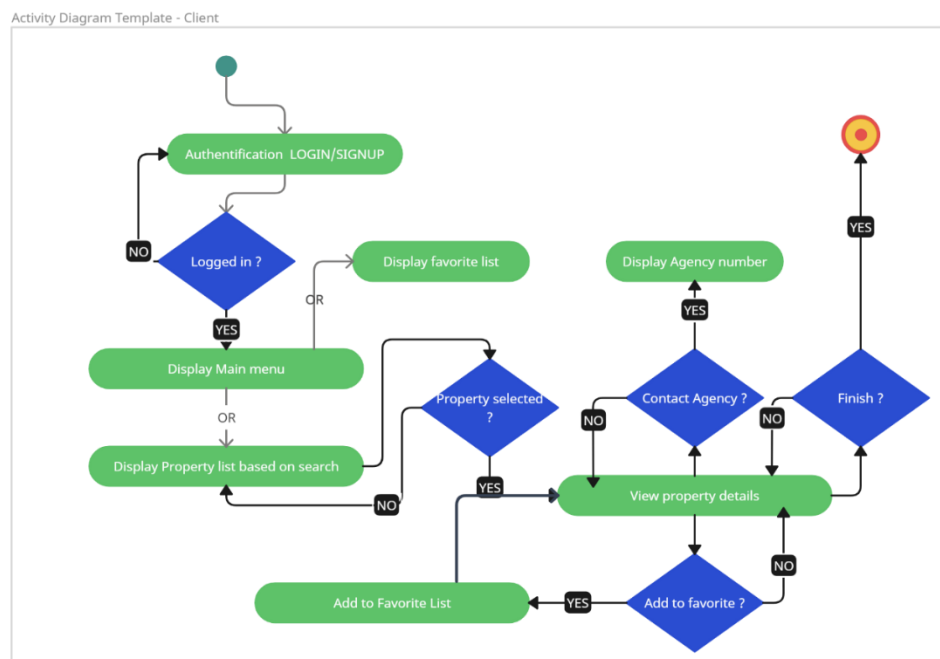
This is an **Activity Diagram** for a **Client's process flow** in an agency system.

Key Flow:

1. **Login/Signup Authentication**:

   o   If **not logged in**, return to authentication.

   o   If **logged in**, proceed to the main menu.

2. **Display Options**:

   o   View **favorite list** or search and display property lists.

3. **Property Interaction**:
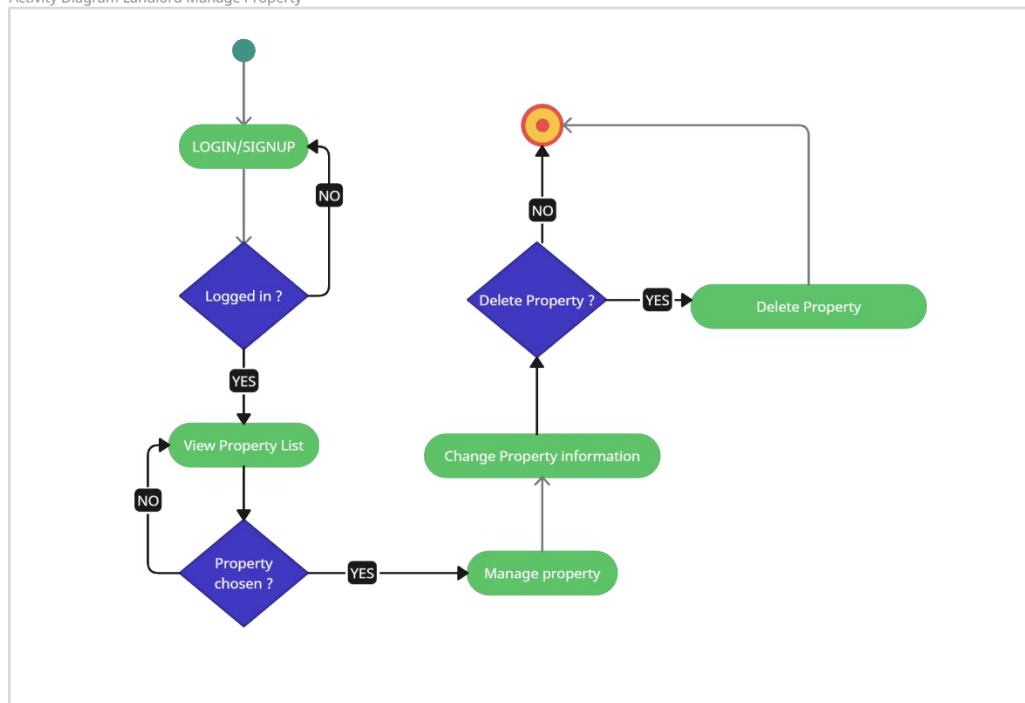
   o   If a **property is selected**:

      ▪   View details.

      ▪   Option to **add to favorites**.

   o   Option to **contact agency** for more information.

4. **Completion**:

   o   The process ends when the user chooses to finish.

This diagram illustrates user decision points and actions while navigating the system.

Activity Diagram Landlord Manage Property

This is anther **Activity Diagram** focused on the **Landlord's property management process** within the system.

Key Flow:

1. **Login/Signup**:

   - If **not logged in**, return to the login process.

   - If **logged in**, proceed to view the property list.

2. **Property Selection**:

   - If no property is chosen, the process halts.

   - If a property is chosen, the landlord can manage it.

3. **Property Management Options**:

   - Decide whether to:

     - **Delete the property** (if yes, the property is removed).

     - **Change property information** (e.g., update details).

4. **Completion**:

    ○   The process ends after managing or deleting the property.

This diagram outlines the key decision points and actions landlords can perform in the property management workflow.

ASPIRE PROPERTIES

Client

Agency

Property

Landlord

Sale

ABOUT US

Sustainable Building Practices

Community and Green Spaces

Waste Reduction and Recycling

Energy Conservation

ASPIRE PROPERTIES

Back to the list   Add a new client

ID	NAME	SHOW DATASHEET	EDIT client	DELETE client

1. **First Wireframe**: A homepage layout with a navigation bar, a vertical menu, an "About Us" section, and expandable sections highlighting environmental practices.

2. **Second Wireframe**: A client management page featuring a top navigation bar, links for actions, a table with client information, and options to view, edit, or delete records.

This sequence diagram illustrates the process of a user logging in or signing up, submitting property details, and the data flow through the website interface, backend system, and database. It highlights the interactions and confirmations at each step to ensure successful data submission.
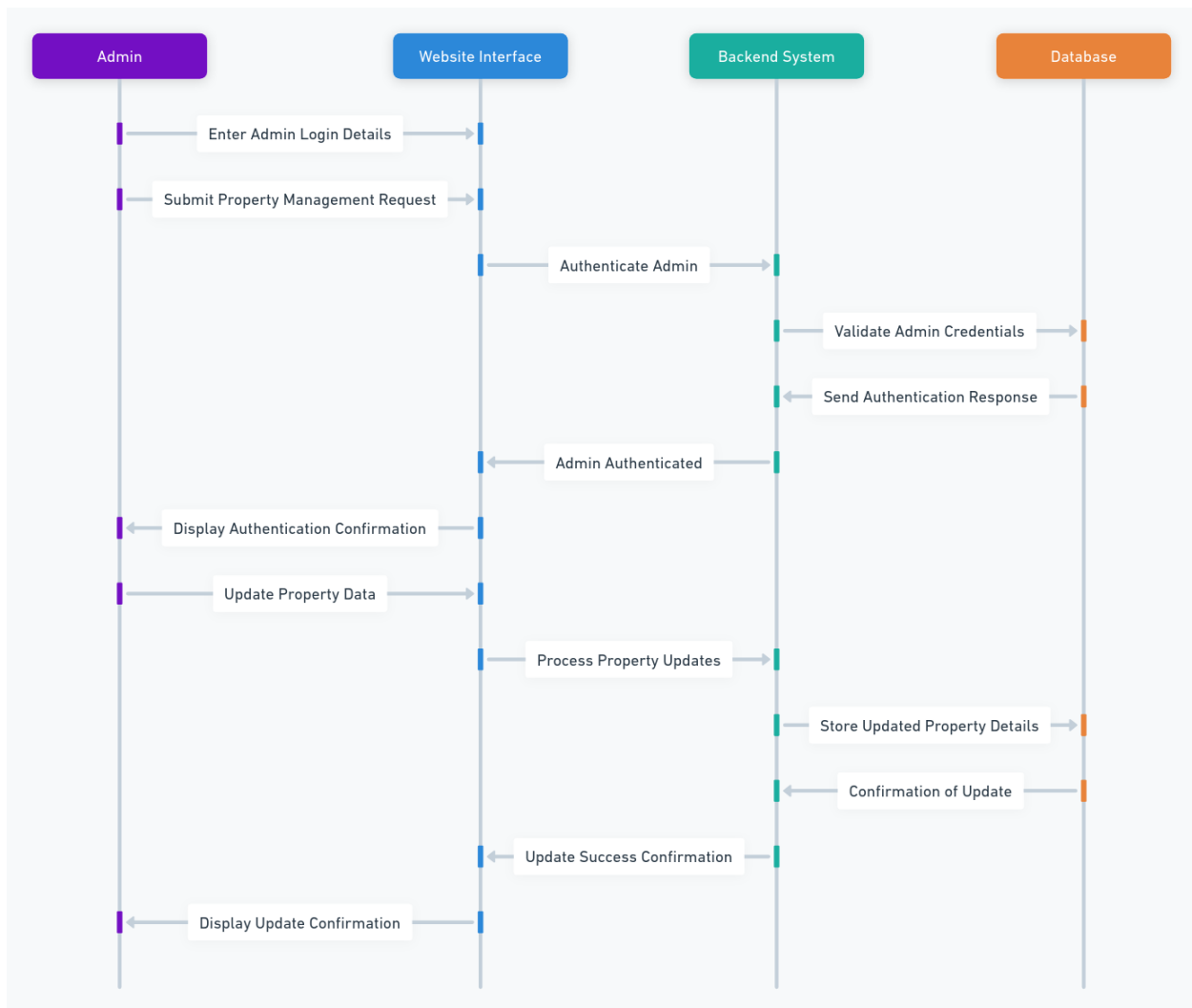
This sequence diagram outlines the process for an admin to manage property data, starting with authentication and ending with updates to the database. It highlights the interactions between the admin, website interface, backend system, and database for secure and efficient property management.
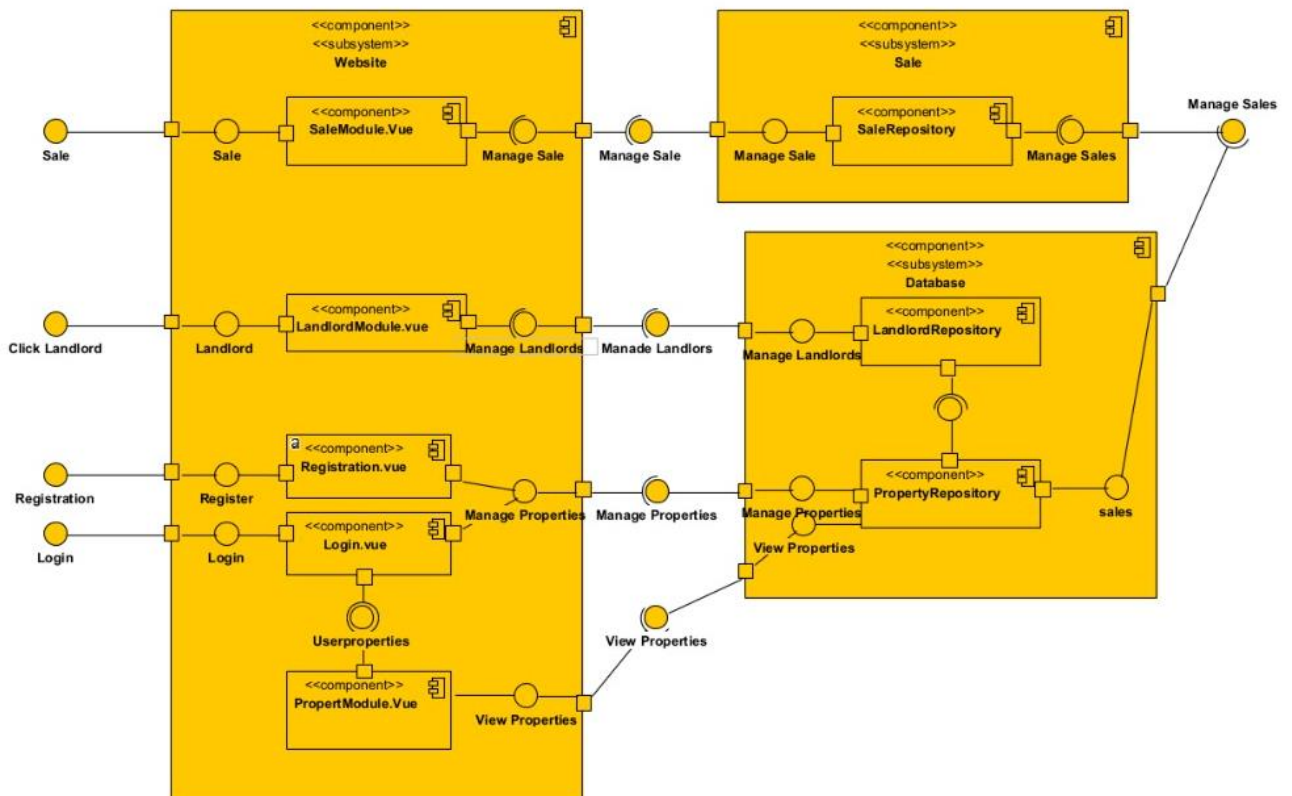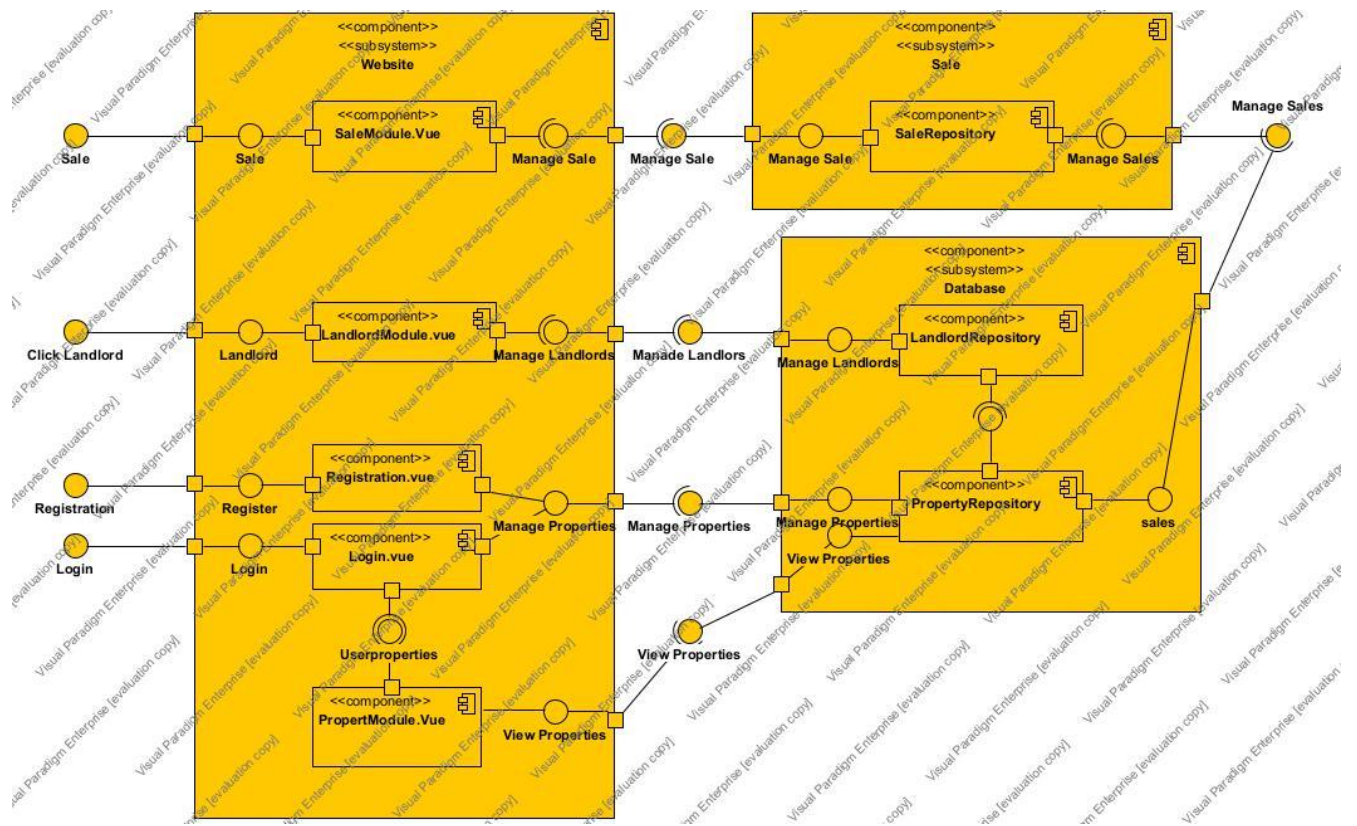
The Diagram above is the component diagram showcasing a few of the components and how they interact among each other through the views, the database and the business logic

Landlord
- int landlordID
- String firstname
- String surname
- LandlordRepository landlordRepository

- getlandlordID()
- getfirstname()
- getsurname()

Client
- int ClientID
- String name
- String email
- String gender
- ClientRepository

- getClientID()
- getgender()
- getName()
- getEmail()
- getProperties()

Agency
- String name
- String address
- Date dateofcreation
- int agency_rating
- int nb_employees

- getname()
- getaddress()
- getdateofcreation()
- get agency_rating()
- get nb_employees()
- getLandlords()
- getClients()

HAS

Inherits

HAS

HAS

Property
- int PropertyID
- int surface_area
- int bathrooms
- int bedrooms
- String type
- String address
- PropertyRepository

- getPropertyID()
- getaddress()
- getbathrooms()
- getbedrooms()
- getsurface_area()
- gettype()

Sale
- int PropertyID
- int landlordID
- int clientID
- int saleID
- String date
- double amount
- int commission
- SaleRepository

- getDate()
- getAmount()
- getSaleRepository()
- getPropertyID()
- getlandlordID()
- getclientID()
- getsaleID()
- getcomission()

- **Landlord**: Represents a landlord with fields for name, email, a reference to a LandlordRepository, and a list of properties. It includes methods to get the name, email, and properties.
- **Client**: Represents a client with fields for name, email, a reference to a ClientRepository, and a list of properties. It includes methods to get the name, email, and properties.
- **Agency**: Represents an agency with fields for name and location. It includes methods to get lists of landlords and clients.
- **Property**: Represents a property with fields for address, price, type, and a reference to a PropertyRepository. It includes methods to get the address, price, and type.
- **Sale**: Represents a sale with fields for date, amount, and a reference to a Sale Repository. It includes methods to get the date, amount, and the sale repository