Eproma
-
Easy Property Management

document development

version	description	date
0.1	Initial Implementation	08-19-2022
0.2	Second Implementation	09-02-2022

Table of Contents

1 vision

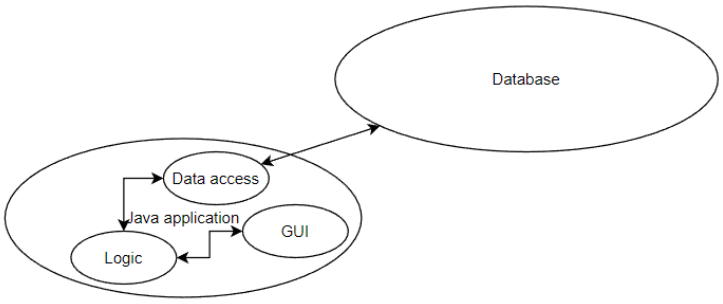
1.1 Vision and brief description of the project

We intend to develop a software system, Eproma, for users of a real estate agency. With this Project it has the purpose of streamlining and facilitating the work operations within the agency, for different user roles, such as agency administrators and commercial agents. With the System we intend that users can manage the real estate, managing the client portfolio, managing the properties available for sale or rent and managing the purchases or rentals of said properties.

2 approximate specification

2.1 Link to existing systems

This system is composed of two subsystems. On the one hand we have the Java application, which will be made up of the user interface, the business logic layer and the data access layer. This last layer communicates in turn with another subsystem: database. The base will act as a store of structured information to feed the application. The information flow is bidirectional and sequential: The user can work in the interface layer to perform actions, which will travel to the database, and vice versa: the user can consult information to the database and it will send the information in the other direction until it reaches the interface and can be viewed.



Services included in the system

- ***i)*** **Create, edit, delete and list clients.**
- ***ii)*** **Create, edit, delete and list properties.**
- ***iii)*** **Sell, buy, rent property and see all the operations carried out.**

Services not included in the system

- **i)** **community administration**
- **ii)** **Management of payments and non-payments**

Interfaces
Graphic User Interface

- **i)** **createCustomer, updateCustomer**
- **ii)** **createProperty, updateProperty, deleteProperty, listProperty**
- **iii)** **sellProperty, buyProperty, rentProperty, listContracts**

business logic

- **iv) createCustomer, updateCustomer**
- **v) createProperty, updateProperty, deleteProperty, listProperty**
- **vi) sellProperty, buyProperty, rentProperty, listContracts**

data access

- **i) createCustomer, updateCustomer, deleteCustomer, listCustomers.**
- **ii) createProperty, updateProperty, deleteProperty, listProperty**
- **iii) createContract, listContracts**

2.2 Summary of required functionality

Through the system, the actors will be able to manage clients, properties and operations.

Regarding the client portfolio, the Sales Agent actor will be able to list, create, edit a client file; the Adminsitrator actor, apart from the previous operations on clients, will also be able to delete a client file.

Regarding real estate, the actor Sales Agent, will be able to list, create, edit the record of a property; the Adminsitrator actor, apart from the previous operations on real estate, will also be able to delete the record of a real estate.

Finally, the actors will be able to register sales and rental operations. In both cases, there will need to be a buyer client and a seller client, likewise we will need the file of the property on which the operation is going to be applied to be registered and updated in the system. These operations can be carried out indistinctly by a Sales Agent or an Administrator.

2.3 Essential quality requirements and framework conditions

Application type: desktop application
Database: MySQL
Framework for the graphical user interface (GUI): JavaFX
Hardware: PC or laptop.64bits. Windows(>=8)/Linux(>=18.04)/Mac OS(>=10.14)
Minimal screen resolution: 1280 x 1024
Min Ram: 8GB
Data retention policy with respect to GDPR: we store the information in a database server managed by azure. Azure SaaS takes care of meeting these GDPR requirements.
Deadline: 7-10-2022
The application will have an average reaction time of 2 seconds for queries and actions.
The database will be stored in the cloud and the system has authentication and authorization, through the login that requests Username and password. Having a high reliability of service. Likewise, it will be portable and easy to use and learn for the user.

3 detailed specification

3.1 System actors (people)

Administrators: *once who manages the properties through access to the functions that allow you to create, modify and delete properties as well as access the list of properties.*

Sales agent: *who has access to the functions of creating, editing, registering the sale and rental of properties as well as accessing the list of the same available in the database.*

3.2 Detailed functional requirements (scenarios and screens)

3.2.1 <Manage customer>

The purpose of this function is to be able to manage users, from the point of view of the sales agent.

3.2.1.1 Case of use. Create customer

Title	Create customer
Short description	*The actor enters the data requested from the client, confirms them and proceeds to create the entity in the system.*
actuators	*sales agent*
Prerequisites (entry)	*The new client identifier cannot be previously registered in the system.*
Description of the process (course)	*In a graphical interface where a form appears with various fields referring to the client's information, the actor completes these fields, validates them and finally activates a button to enter the client in the system.*
Impact (Output)	*The client is added to the system*
Observations	

3.2.1.2 Case of use. Edit customer

Title	Edit customer
Short description	*The actor updates the data of a client.*
actuators	*sales agent*
Prerequisites (entry)	*The client must exist in the system*
Description of the process (course)	*After selecting the client we want to update from a list, we go to a screen where we see a form filled out with the current data. The fields are editable and can be changed by the actor. After finishing the changes, you have to click a button to save them.*
Impact (Output)	*Customer data is updated*
Observations	

3.2.1.3 Case of use. listcustomer

Title	listcustomer
Short description	*The actor lists all the clients that are registered in the system*
actuators	*sales agent*
Prerequisites (entry)	
Description of the process (course)	*The actor asks through the action of a button to show all the registered clients that are in the system*
Impact (Output)	*A list of all clients*
Observations	

3.2.1.4 Case of use. Delete customer

Title	Delete customer
Short description	*The actor removes a client.*
actuators	*Administrator*
Prerequisites (entry)	*The client must exist in the system*
Description of the process (course)	*After selecting the client that we want to eliminate from a list, we go to a dialog box that asks us if we are sure (Yes/no). In case of selecting yes, the client is deleted, in case of selecting no, the client is not deleted.*
Impact (Output)	*The client is deleted if you selected yes*
Observations	

3.2.1.5 Special dependencies/non-functional requirements

3.2.2 Manage properties

3.2.2.1 Case of use. Createproperty

title	Createproperty
Short description	*The actor inserts a new property to the system*
actuators	*sales agent*
Prerequisites (entry)	*The property cannot already be entered* *The seller of the property must be registered as a customer*
Description of the process (course)	*In a graphical interface where a form appears with various fields referring to the property information, the actor completes these fields, validates them and finally activates a button to enter the property in the system.*
Impact (Output)	*The property has been entered into the system*
Observations	

3.2.2.2 Case of use. Edit property

title	edit-property
Short description	*The actor updates the data of a property.*
actuators	*sales agent*
Prerequisites (entry)	*The property must exist in the system*
Description of the process (course)	*After selecting the property that we want to update in a list, we go to a screen where we see a form completed with the current data. The fields are editable and can be changed by the actor. After finishing the changes, you have to click a button to save them.*
Impact (Output)	*The property data is updated*
Observations	

3.2.2.3 Case of use. List property

title	listproperties
Short description	*The actor lists all the properties that are registered in the system*
actuators	*sales agent*
Prerequisites (entry)	
Description of the process (course)	*The actor requests by means of a button to show all the properties that are registered in the system*
Impact (Output)	*A list of all properties*
Observations	

3.2.2.4 Case of use. Deleteproperty

title	Deleteproperty
Short description	*The actor removes a property.*
actuators	*sales agent*
Prerequisites (entry)	*The property must exist in the system*
Description of the process (course)	*After selecting the property that we want to update from a list, we go to a dialog box that asks us if we are sure (Yes/no). If you select yes, the property is deleted, if you select no, the property is not deleted.*
Impact (Output)	*The property is deleted if you selected "yes"*
Observations	

3.2.3 Manage sales

3.2.3.1 Case of use. Sell a property

title	Sell a property
Short description	*The actor sells a property from a seller (customer) to a buyer (customer)*
actuators	*sales agent*
Prerequisites (entry)	*The property must exist in the system.* *The seller has to be registered* *The buyer must be registered*
Description of the process (course)	*The actor selects a property for sale, when selecting the property, he already visualizes the seller's data. When he has the property, he can modify its price at that moment. The actor selects a customer for sale from the customer list. Finally, after validating the data of the sale, he activates a button to create the contract of sale.*
Impact (Output)	*Register a sales contract*
Observations	

3.2.3.2 Case of use. Rent a property

title	Rent a property

Short description	*The actor rents a property from an owner (customer) to a tenant (customer)*
actuators	*sales agent*
Prerequisites (entry)	*The property must exist in the system.* *The owner has to be registered* *Tenant must be registered*
Description of the process (course)	*The actor selects a property for rent, when selecting the property, he already visualizes the data of the owner. When he has the property, he can modify his fee at that time. The actor selects a tenant for the rental from the list of clients. Finally, after validating the data of the operation, he presses a button to create the rental contract.*
Impact (Output)	*Register a rental agreement*
Observations	

3.3 interfaces

3.3.1 User Interfaces (GUIs)

3.3.1.1 LOGIN PROTOTYPE

Starting from screen 3.3.1.1, after pressing the "SIGN IN" button, if the inputs are correct (the user exists, and his credentials are correct), we go to screen 3.3.1.2

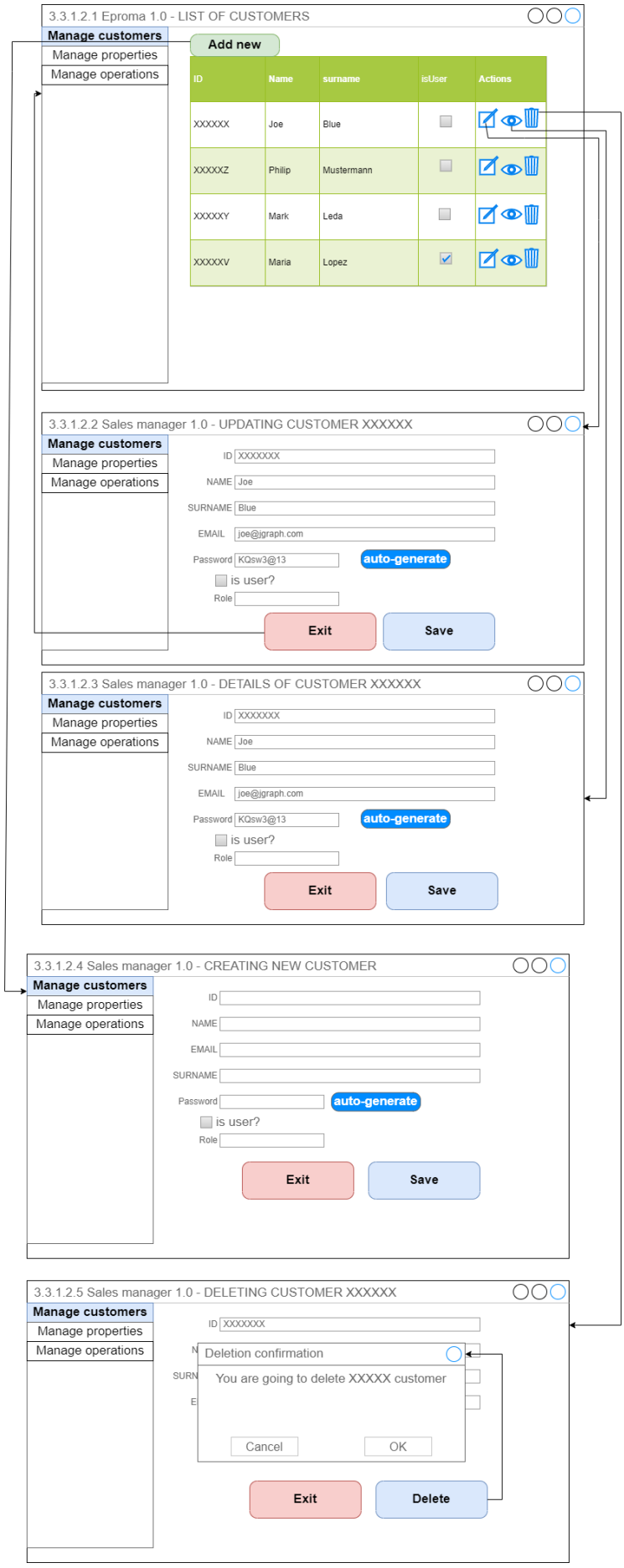
3.3.1.1.1 SYSTEM LOGIN SCREEN

On this screen the user can view a form to fill in two fields: username and password and a button to activate when these two fields have been filled in.

3.3.1.1.2 WELCOME SCREEN

On this screen, the user sees a welcome message and some relevant system indicators: such as sales, rentals per week, etc. On the left will appear the navigation menu of the remaining functionalities.

3.3.1.2 MANAGE CUSTOMERS PROTOTYPE


3.3.1.2.1 Eproma 1.0 - LIST OF CUSTOMERS
Manage customers
Manage properties
Manage operations
Add new
ID
Name
surname
isUser
Actions
XXXXXX
Joe
Blue
XXXXXZ
Philip
Mustermann
XXXXXY
Mark
Leda
XXXXXV
Maria
Lopez
3.3.1.2.2 Sales manager 1.0 - UPDATING CUSTOMER XXXXXX
ID XXXXXXX
NAME Joe
SURNAME Blue
EMAIL joe@jgraph.com
Password KQsw3@13
auto-generate
is user?
Role
Exit
Save
3.3.1.2.3 Sales manager 1.0 - DETAILS OF CUSTOMER XXXXXX
3.3.1.2.4 Sales manager 1.0 - CREATING NEW CUSTOMER
3.3.1.2.5 Sales manager 1.0 - DELETING CUSTOMER XXXXXX
Deletion confirmation
You are going to delete XXXXX customer
Cancel
OK
Delete

When the user enter the manage customers menu, will arrive at screen 3.3.1.2.1, from this screen the user can interact with a list of customers. For each customer in the list, the user can do an edit or delete operation. The user can also click on the "CREATE NEW" button to create a new customer.

- If from 3.3.1.2.1 the user activates the edit button  will navigate to screen 3.3.1.2.2 where he can change the customer data and save it by pressing the "SAVE" button, or exit the screen without saving by pressing the "EXIT" button.
- If from 3.3.1.2.1 the user activates the view button  will navigate to screen 3.3.1.2.3. where he can view the customer data without the option to edit any attribute (read mode)
- If from 3.3.1.2.1 the user clicks on the "CREATE NEW" button, will navigate to screen 3.3.1.2.4 where he can fill in the data of a new customer, and register it by clicking on the "SAVE" button or exit without saving changes by clicking on EXIT button.
- If from 3.3.1.2.1 the user activates the delete button  , he would navigate to screen 3.3.1.2.5, where he can view the information of the customer that he is going to delete and then he can confirm the deletion with the "Delete" button that will take him to a confirmation dialog, or he can exit without making changes by doing click on "Exit"

3.3.1.2.1 List of customers screen

The user visualizes a table with different columns that refer to data of each customer. In the final column on the right there are a series of buttons to perform actions on the customer represented in that row of the table.

3.3.1.2.2 Update customer screen

The user views a form with fields referring to properties of a customer. The fields are editable, but to save the changes on the underlying entity, you would have to click on the "SAVE" button.

3.3.1.2.3 Customer details screen

The user will see a form with fields referring to properties of a customer in reading mode. You may not alter any property or record changes in the system.

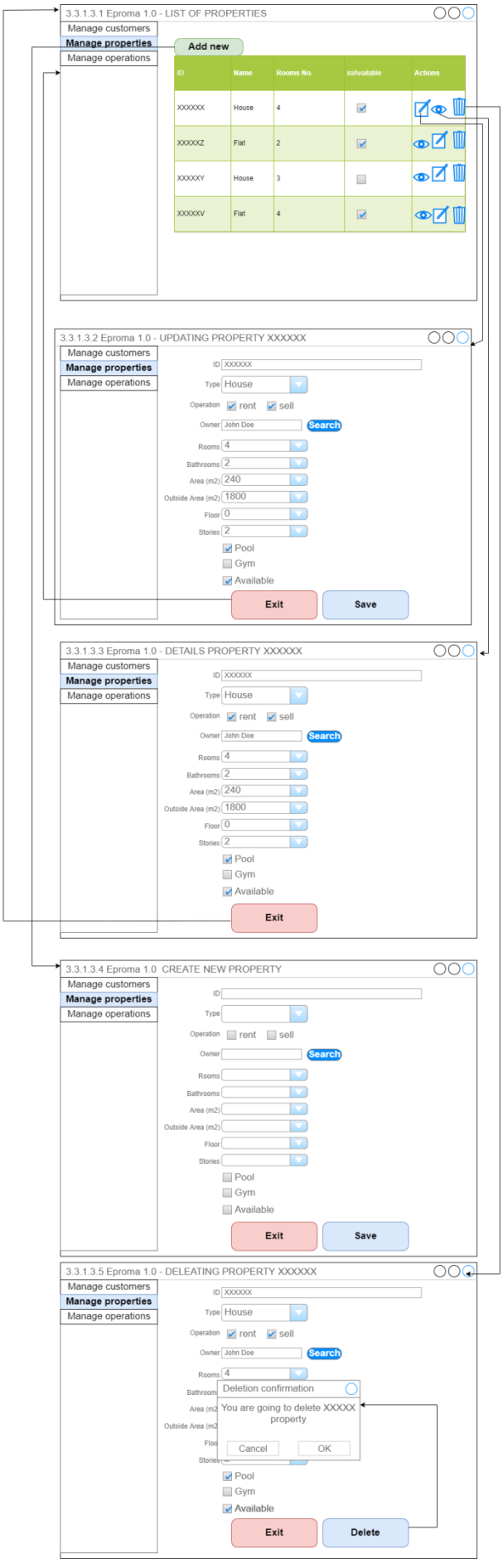
3.3.1.2.4 New customer screen

The user displays a form with empty fields efferent to properties of a new customer. The fields are editable, but to save the changes on the new underlying entity, you would have to click on the "SAVE" button.

3.3.1.2.5 Delete customer screen

The user views a form with fields referring to properties of a customer. The fields are not editable. If he want to confirm the deletion of the customer, he should click on the "DELETE" button and then reconfirm the deletion after the security dialog.

3.3.1.3 MANAGE PROPERTIES PROTOTYPE


3.3.1.3.1 Eproma 1.0 - LIST OF PROPERTIES
Manage customers
Manage properties
Manage operations
Add new
ID
Name
Rooms No.
isAvailable
Actions
XXXXXX
House
4
XXXXXZ
Flat
2
XXXXXY
House
3
XXXXXV
Flat
4
3.3.1.3.2 Eproma 1.0 - UPDATING PROPERTY XXXXXX
Manage customers
Manage properties
Manage operations
ID XXXXXX
Type House
Operation rent sell
Owner John Doe
Search
Rooms 4
Bathrooms 2
Area (m2) 240
Outside Area (m2) 1800
Floor 0
Stories 2
Pool
Gym
Available
Exit
Save
3.3.1.3.3 Eproma 1.0 - DETAILS PROPERTY XXXXXX
Manage customers
Manage properties
Manage operations
ID XXXXXX
Type House
Operation rent sell
Owner John Doe
Search
Rooms 4
Bathrooms 2
Area (m2) 240
Outside Area (m2) 1800
Floor 0
Stories 2
Pool
Gym
Available
Exit
3.3.1.3.4 Eproma 1.0 CREATE NEW PROPERTY
Manage customers
Manage properties
Manage operations
ID
Type
Operation rent sell
Owner
Search
Rooms
Bathrooms
Area (m2)
Outside Area (m2)
Floor
Stories
Pool
Gym
Available
Exit
Save
3.3.1.3.5 Eproma 1.0 - DELEATING PROPERTY XXXXXX
Manage customers
Manage properties
Manage operations
ID XXXXXX
Type House
Operation rent sell
Owner John Doe
Search
Rooms 4
Deletion confirmation
You are going to delete XXXXX property
Cancel
OK
Pool
Gym
Available
Exit
Delete

When we enter the manage properties menu, we arrive at screen 3.3.1.3.1, from this screen we can interact with a list of properties. For each property in the list, I can do an edit or delete operation. I can also click the "CREATE NEW" button to create a new property.

- If from 3.3.1.3.1 we activate the edit button  We navigate to screen 3.3.1.3.2 where we can alter the data of the property and save it by pressing the "SAVE" button, or exit the screen without saving by pressing the "EXIT" button.
- If from 3.3.1.3.1 we activate the view button  We navigate to screen 3.3.1.3.3. where we can view the data of the property without the option to edit any attribute (read mode)
- If from 3.3.1.3.1 we click on the "CREATE NEW" button, we navigate to screen 3.3.1.3.4 where we can fill in the data of a new property, and register it by clicking on the "SAVE" button or exit without saving changes by clicking on EXIT button.
- If from 3.3.1.3.1 we activate the delete button  , we would navigate to screen 3.3.1.3.5, where we can view the information of the property that we are going to delete and then we can confirm the deletion with the "Delete" button that will take us to a confirmation dialog, or we can exit without making changes clicking "Exit"

3.3.1.3.1 List of properties screen

The user visualizes a table with different columns that refer to data of each property. In the final column on the right there are a series of buttons to perform actions on the property represented in that row of the table.

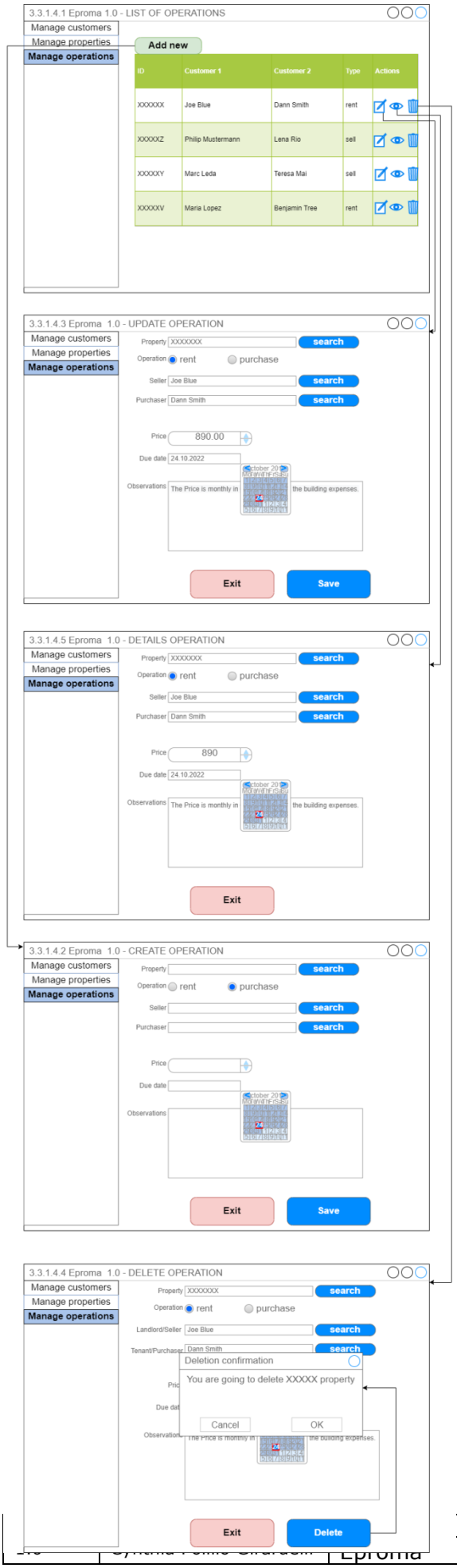
3.3.1.3.2 Update property screen
The user views a form with fields referring to properties of a particular property. The fields are editable, but to save the changes on the underlying entity, you would have to click on the "SAVE" button.

3.3.1.3.3 Details property screen
The user will see a form with fields referring to the properties of a property in reading mode. You may not alter any properties or log changes to the system.

3.3.1.3.4 Create new property screen
The user displays a form with empty fields efferent to properties of a new property. The fields are editable, but to save the changes on the new underlying entity, you would have to click on the "SAVE" button.

3.3.1.3.5 Delete property screen
The user displays a form with fields referring to properties of a property. The fields are not editable. If we want to confirm the deletion of the property, we should click on the "DELETE" button and then reconfirm the deletion after the security dialog.

3.3.1.3 MANAGE OPERATION PROTOTYPE


3.3.1.4.1 Eproma 1.0 - LIST OF OPERATIONS
Manage customers
Manage properties
Manage operations
Add new
ID
Customer 1
Customer 2
Type
Actions
XXXXXX
Joe Blue
Dann Smith
rent
XXXXXZ
Philip Mustermann
Lena Rio
sell
XXXXXY
Marc Leda
Teresa Mai
sell
XXXXXV
Maria Lopez
Benjamin Tree
rent
3.3.1.4.3 Eproma 1.0 - UPDATE OPERATION
Property
XXXXXXX
search
Operation
rent
purchase
Seller
Joe Blue
Purchaser
Dann Smith
Price
890.00
Due date
24.10.2022
Observations
The Price is monthly in
the building expenses.
Exit
Save
3.3.1.4.5 Eproma 1.0 - DETAILS OPERATION
890
3.3.1.4.2 Eproma 1.0 - CREATE OPERATION
3.3.1.4.4 Eproma 1.0 - DELETE OPERATION
Landlord/Seller
Tenant/Purchaser
Deletion confirmation
You are going to delete XXXXX property
Cancel
OK
Delete

When we enter the manage operation menu, we arrive at screen 3.3.1.4.1, from this screen we can interact with a list of operations identified by client. For each operation in the list, I can do an edit or delete operation. I can also click the "ADD NEW" button to create a new operation.

- *Yes since 3.3.1.4.1 press the edit button and navigate to screen 3.3.1.4.2 where we can change the operation data and save it by pressing the "SAVE" button, or exit the screen without saving by pressing the "EXIT" button.*
- *Yes since 3.3.1.4.1 press the view button to navigate to screen 3.3.1.4.3. where we can view the data of the operation without the option to edit any attribute (reading mode)*
- *If from 3.3.1.4.1 we click on the "ADDNEW", we navigate to screen 3.3.1.4.4 where we can fill in the data of a new property, and register it by pressing the "SAVE" button or exit without saving changes by pressing the "EXIT" button.*
- *Yes since 3.3.1.4.1 we activate the delete button, we would navigate to screen 3.3.1.3.5, where we can view the information of the operation that we are going to delete and then we can confirm the deletion with the "Delete" button that will take us to a confirmation dialog , or we can exit without making changes by clicking on "Exit"*

3.3.1.4.1 List of operation screen
The user visualizes a table with different columns that refer to data of each property. In the final column on the right there are a series of buttons to perform actions on the property represented in that row of the table.

3.3.1.4.2 Update operation screen
The user views a form with fields referring to properties of a particular property. The fields are editable, but to save the changes on the underlying entity, you would have to click on the "SAVE" button.

3.3.1.4.3 Details operation screen
The user will see a form with fields referring to the properties of an operation in reading mode. You cannot alter any operations or record changes to the system.

3.3.1.4.4 Create new operation screen
The user displays a form with empty fields efferent to properties of a new property. The fields are editable, but to save the changes on the new underlying entity, you would have to click on the "SAVE" button.

3.3.1.4.5 Delete operation screen
The user views a form with fields referring to properties of an operation. The fields are not editable. If we want to confirm the deletion of the operation, we should click on the "DELETE" button and then reconfirm the deletion after the security dialog.

3.3.2 system interface

3.3.2.1 <interface a>

layer 1 . Gui: interface developed with Java and JavaFX. Through which the user interacts with the controls and forms.

3.3.2.2 <Interface b>

layer 2. Business logic: interface developed with Java. This layer is invoked by layer 1 (gui) to call methods that contain the necessary operations to work with the entities of the system and perform the necessary tasks with the data supplied by the user.

3.3.2.3 <Interface c>

layer 3. data layer: interface developed with java. to access the underlying data store. in this case we use a jdbc driver to access the mysql database engine through authentication. This layer is invoked by layer 2 (business logic) and implements the necessary methods to perform the basic operations of reading, modifying and deleting on the database.

3.4 non-functional requirements

3.4.1 Specifications for hardware and software.

Application type: desktop application
Database: MySQL 8.0.28 (release date from January 2022)
Framework for the graphical user interface (GUI): JavaFX 18
Hardware: PC or laptop.64bits. Windows(>=8)/Linux(>=18.04)/Mac OS(>=10.14)
Minimal screen resolution: 1280 x 1024
Data retention policy with respect to GDPR: we store the information in a database server managed by azure. Azure SaaS takes care of meeting these GDPR requirements.
Deadline: 7-10-2022

3.4.2 Performance

The response time for each query/action will take an average of 2 seconds. We will work to optimize the sql queries, and try to carry out a correct database design (through normalization and indexes) so that the operations on the database are as light as possible. We guarantee a fast and light application for and adequate performance for the usage scenario for which it is supposed to work.

3.4.3 Resources and hardware requirements

It will be necessary to have the following:
PC/laptop with a minimum of 8GB ram, 256 GB hard drive, keyboard and mouse. Monitor with a minimum resolution of 1280 x 1024. Stable internet connection.

3.4.4 Security

The system has authentication and authorization, through the login that requests Username and password. Regarding the database hosted in the cloud by the Azure provider in a SaaS environment, which complies with the corresponding data protection and security specifications (ISO 27000), and is subject to continuous security and data processing audits. .

3.4.5 reliability

We promise to give a reliability of 90% of no service outages. For them we have, on the part of the database provider, with replication options in case of failure.

3.4.6 Maintenance

We promise to provide free maintenance service for the first 3 months from the delivery of the product. And the maintenance cost after the first 3 months will be €100/hour.

3.4.7 Portability / Scalability / Reuse

The software can be installed on several PCs with no limitation on the number of licenses.

3.4.8 usability

Learning:*The learning time of the system by a user will not exceed 2 minutes in any functionality. The system has a user manual.*
Design:*the system will present a nice, clear and intuitive graphical interface.*

3.4.9 Accessibility

It is not adapted for the blind or hearing impared.

3.5 System delimitation, system architecture and data management.

3.5.1 Database structure

3.5.1.1 Person Table

Column	Type	Description
ID	*INT (AUTOINCREMENT)*	*PRIMARY KEY*
NAME	*VARCHAR(25)*	
SURNAME	*VARCHAR(50)*	
PHONE	*VARCHAR(25)*	
EMAIL	*VARCHAR(50)*	
PASS	*VARCHAR(10)*	
COUNTRY	*VARCHAR(25)*	
REGION	*VARCHAR(25)*	
CITY	*VARCHAR(25)*	
ZIP	*VARCHAR(25)*	
STREET	*VARCHAR(50)*	
ROLE	*NVARCHAR(15)*	
ISDELETED	*BIT*	

3.5.1.2 Property Table

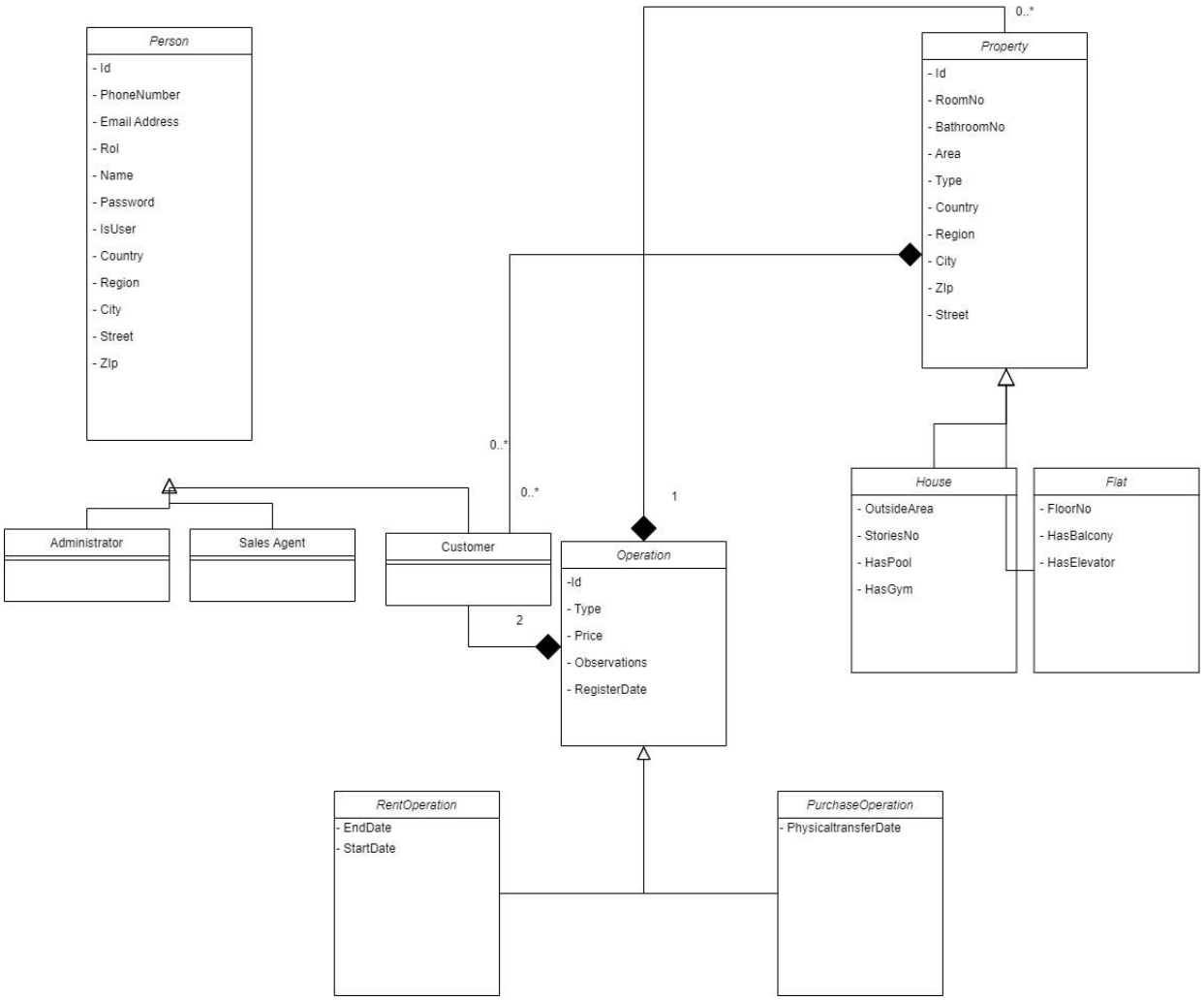
Column	Type	Descrtiption
ID	*INT (AUTOINC)*	*PRIMARY KEY*
PERSON_ID	*INT*	*FOREIGN KEY (PERSON)*
ROOMNO	*INT*	
BATHROOMNO	*INT*	
AREA	*FLOAT*	
TYPE	*NVARCHAR(10)*	
OUTSIDEAREA	*FLOAT*	
STORIESNO	*INT*	

HASPOOL	*BIT*	
HASGYM	*BIT*	
FLOORNO	*INT*	
HASBALCONY	*BIT*	
HASELEVATOR	*BIT*	
COUNTRY	*VARCHAR(25)*	
REGION	*VARCHAR(25)*	
CITY	*VARCHAR(25)*	
ZIP	*VARCHAR(25)*	
STREET	*VARCHAR(50)*	
ISDELETED	*BIT*	

3.5.1.3 Operation Table

Column	*Type*	*Description*
ID	*INT(AUTOINC)*	*PRIMARY KEY*
SELLER_ID	*INT*	*FOREIGN KEY (PERSON)*
PURCHASER_ID	*INT*	*FOREIGN KEY(PERSON*
PROPERTY_ID	*INT*	*FOREIGN KEY(PROPERTY)*
PRICE	*FLOAT*	
REGISTERDATE	*DATETIME*	
OBSERVATIONS	*NVARCHAR(200)*	
TYPE	*NVARCHAR(10)*	
ENDDATE	*DATETIME*	
STARTDATE	*DATETIME*	
PHYSICALTRANSFERDATE	*DATETIME*	
ISDELETED	*BIT*	

3.5.2 Class Diagram



3.5.2.1 Person class

The Person abstract class has the attributes that store the: Id (numeric), PhoneNumber (numeric), EmailAdress (String), Role(String), Name, Password and IsUser. The Id will work as a Primary key to identify the people entered in the System and in turn through the Role attributes (Administrator or SalesAgent) and IsUser (which identifies if it is a client or employee) each actor and their permissions will be identified: Thus such as Country, region, city, postal code and street.

The person class has as child classes the Administrator, Sales Agent and Customer classes, which will inherit the attributes of the person class.

3.5.2.2 Administrator class

The Administrator class is a child class of the Person class, therefore it inherits all the behavior and attributes of Person.

3.5.2.3 Customer class

The Customer class is a child class of the Person class, therefore it inherits all the behavior and attributes of Person.

3.5.2.4 Property class

It is an abstract class that models the characteristics of a property, such as: number of rooms, number of bathrooms, area, type, country, region, city, zip, street.

3.5.2.5 House class

The House class is a child class of the Property class; therefore it inherits all the behavior and attributes of Property. It has its own attributes because its purpose is to represent a house-type property, some of its attributes are: outsideArea, HasPool, HasGym.

3.5.2.6 Flat class

The Flat class is a child class of the Property class, therefore it inherits all the behavior and attributes of Property. It has its own attributes because its purpose is to represent a floor type property, some of its attributes are: FloorNo, HasElevator, HasBalcony.

3.5.2.7 Operation

It is an abstract class that models the characteristics of an operation on a property, such as: price of the operation, registration date, type, Observations.

3.5.2.8 RentOperation

The RentOperation class is a child class of the Operation class, therefore it inherits all the behavior and attributes of the OPeration. It has its own attributes because its purpose is to represent a rental type operation, some of its attributes are: EndDate, StartDate.

3.5.2.9 PurchaseOperation

The PurchaseOperation class is a child class of the OPeration class, therefore it inherits all the behavior and attributes of OPeration. It has its own attributes because its purpose is to represent a sale type operation, some of its attributes are: PhysicaltransferDate

3.5.2.10 Property-Customer relationship

This relationship models that a client can own several properties

3.5.2.11 Address-Property relationship

This relationship models that a property has an address.

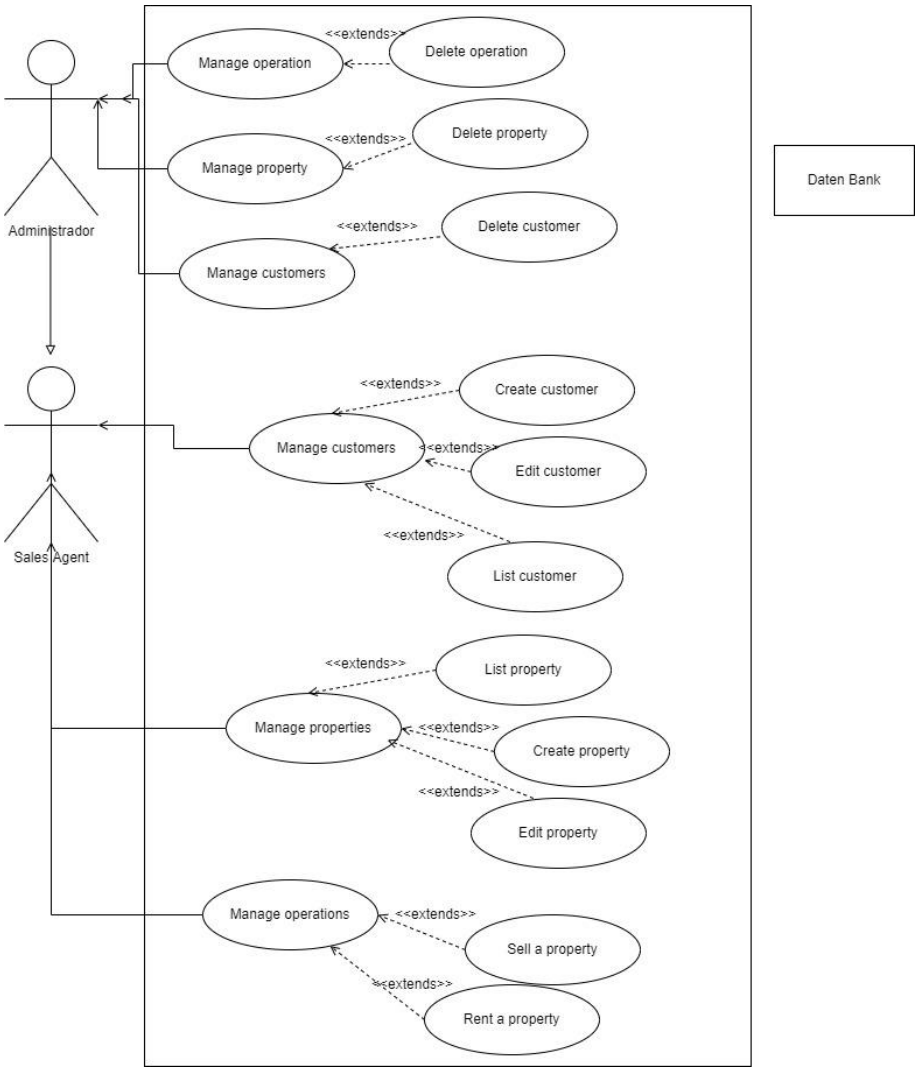
3.5.2.12 Property-operation relationship

This relationship models that an operation is performed on a property. Likewise, a property may have participated in several operations.

3.5.2.13 Property-customer relationship

This property models that it is necessary for 2 clients to participate in an operation, either for rent (lessor-lessee) or for sale (seller-buyer).

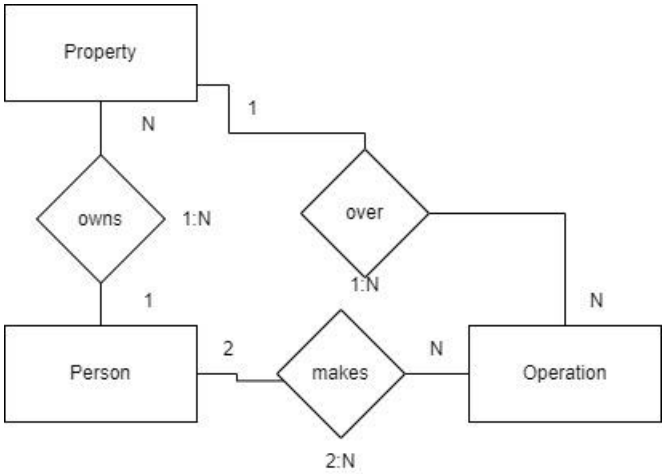
3.5.3 Use Case Diagramm



In the case of case diagrams we can see on one side two actors, Administrator and Sales Agent, who are the ones that interact with the Eproma system and on the other side the Database where all the information will be stored on the server.

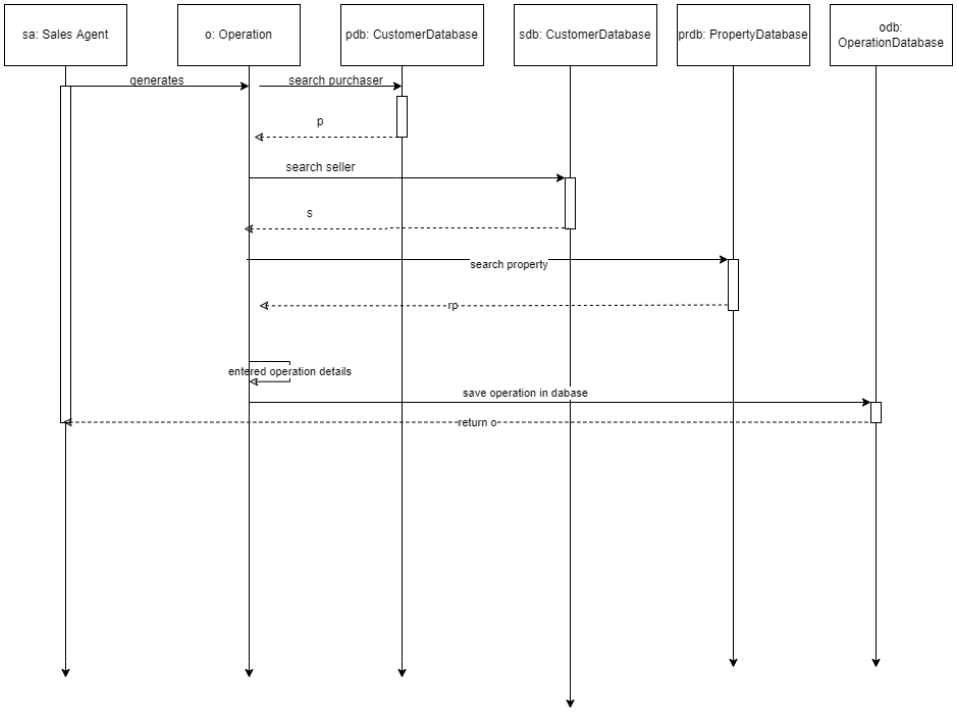
The Sales Agent is an actor, who can carry out the following actions in the system: in terms of Manage sales, he can carry out rental and sale operations. As for Manage properties, you can List, create and edit Properties. And finally regarding Manage customer you can create, edit and list customers. By create the Sales Agent will be able to enter data in a template, by edit you will be able to edit them and with list you will be able to call the data to see a list of the category.
But you will not be able to Delet customer or Delet properti, actions reserved to the Administrator, who will inherit the functionalities of the Sale Agent, that is, you will be able to carry out all the actions that it performs and you will also be able to Delete customer, Delete Operation and delete Property, which are exclusive functionalities of the same .

3.5.4 Entity Relation Diagramm



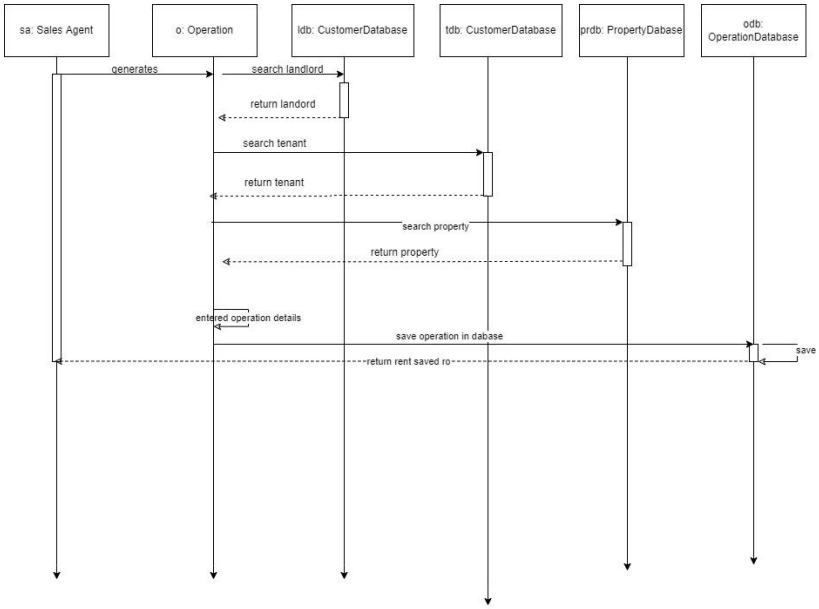
In this diagram we see the relationships between the different entities. We could say that a Person (customer) owns one or many properties. In turn, two People, who would be Customers (Seller and Purchaser, Landlord and Tenant) makes one or many Operations over one Property.

3.5.5 Squence Diagramm Sell Operation



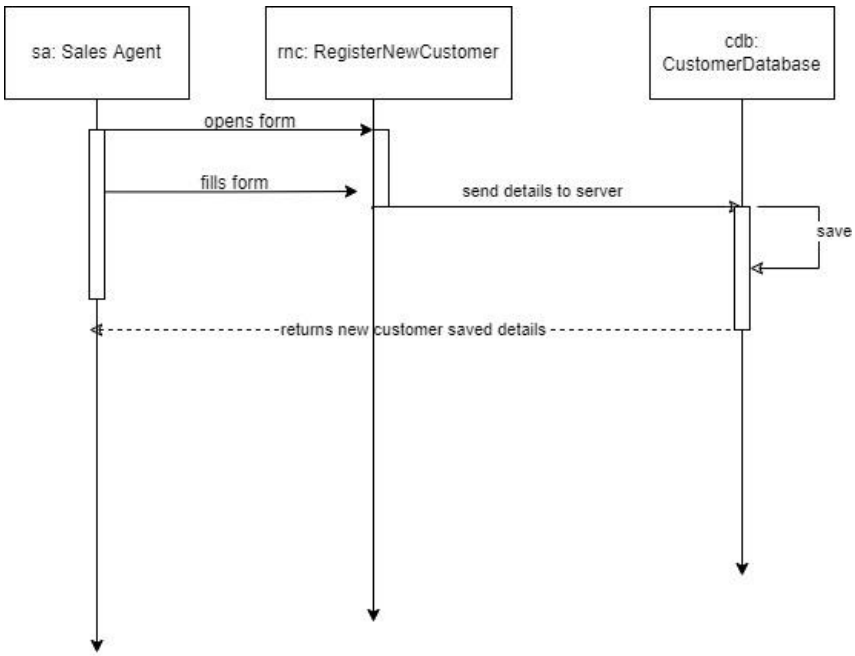
In this diagram we see the sequence of the sale operation. The Sales Agent, by clicking add new operation, generates the operation template, and selects sell, then it searches for the Purchaser's data and the database returns the purchaser's data, then it searches for the seller and returns the Purchaser's data. seller , then it looks up the property and the database returns the property data. Then once the details of the operation have been entered, it is saved in the database and the database returns the saved operation.

3.5.6 Sequenz diagram Rent Operation



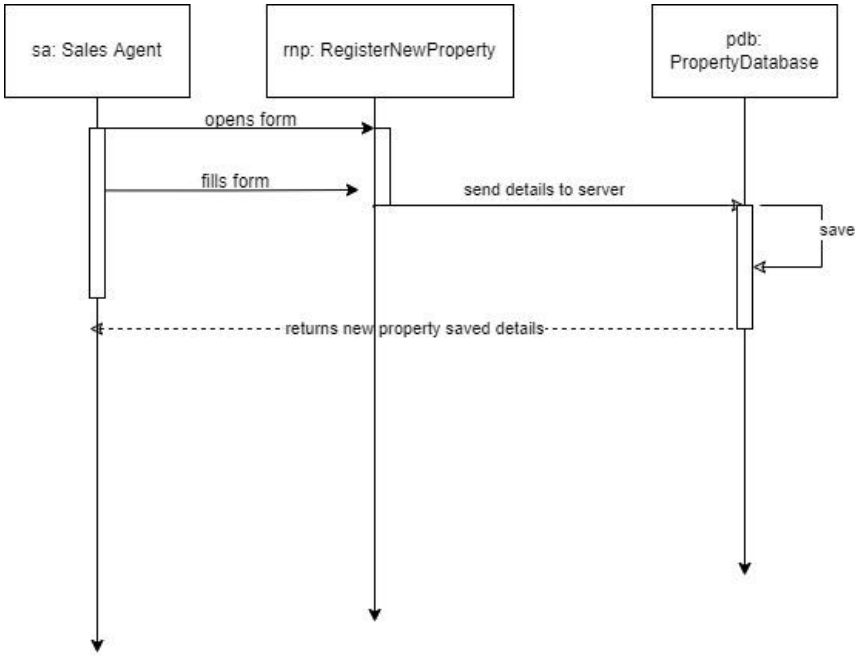
In this diagram we see the sequence of the rental operation. The Sales Agent, by clicking on add new operation, generates the operation template, and selects rent, then in it he searches for the Landlord's data and the database returns the Landlord's data, then he searches for the Tenant and returns the Landlord's data. Tenant, then searches for the property and the database returns the property data. Then once the details of the operation have been entered, it is saved in the database and the database returns the saved operation.

3.5.7 Sequence diagramma register customer



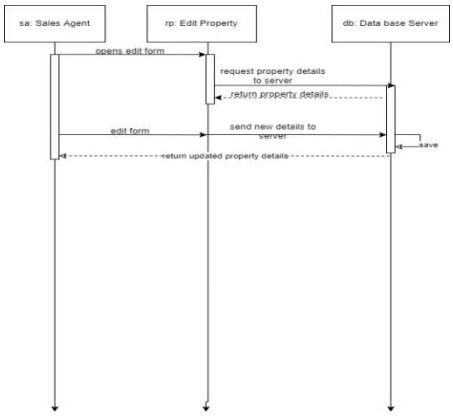
The sales agent opens the form add new customer then fills in the form the system sends the data entered to the customer database on the server there the data of the new customer is saved and the database returns the new customer saved details to the Software Eproma.

3.5.8 Sequence diagramm register property



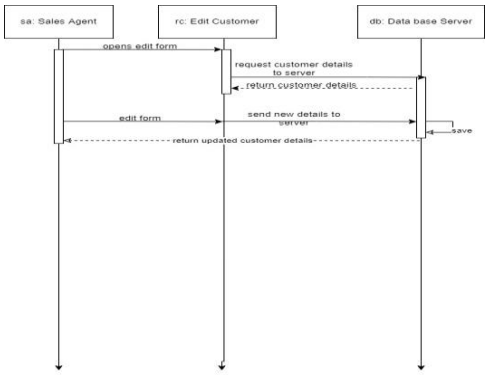
The sales agent opens the add new property form, then fills in the form, the system sends the data entered to the property database on the server, there the data of the new property is saved and the database returns the new property saved details to the Software Eproma.

3.5.9 Sequence diagramm edit property



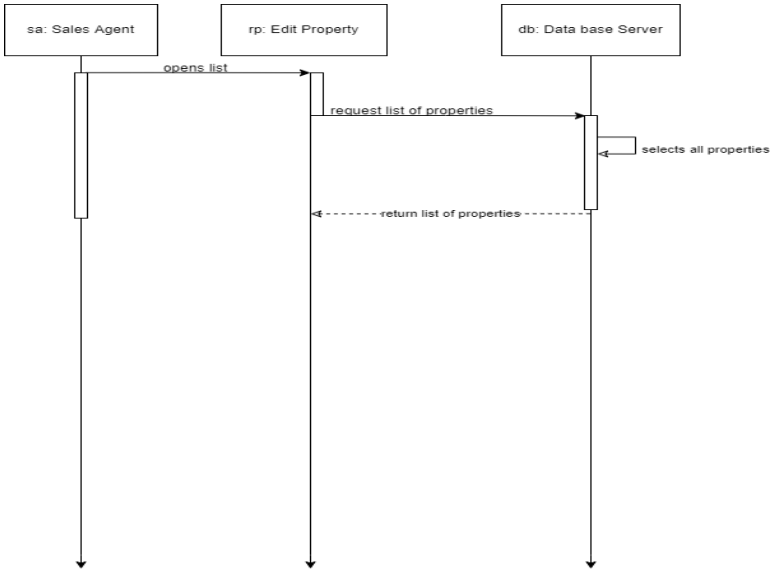
The sales agent opens the form edit property, the system requests data from property and return this data to be edited, then fills in the form with the new data, the system sends the data entered to the property database on the server, there the modified data of the property is saved and the database return the property with the updated saved details to the Software Eproma.

3.5.10 Sequence diagram edit customer



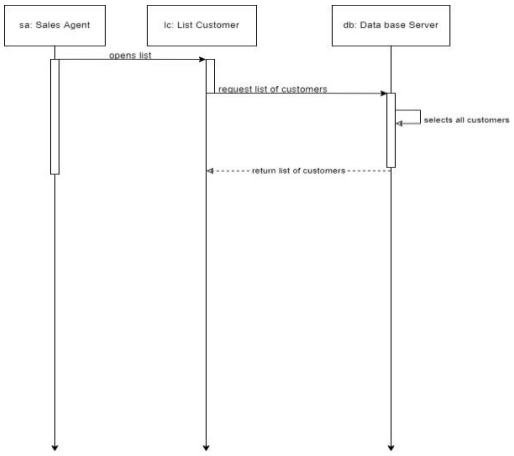
The sales agent opens the form edit customer, the system requests data from customer and return this data to be edited, then fills in the form with the new data, the system sends the data entered to the customer database on the server, there the modified data of the customer is saved and the database return the customer with the updated saved details to the Software Eproma.

3.5.11 Sequence diagramm List property



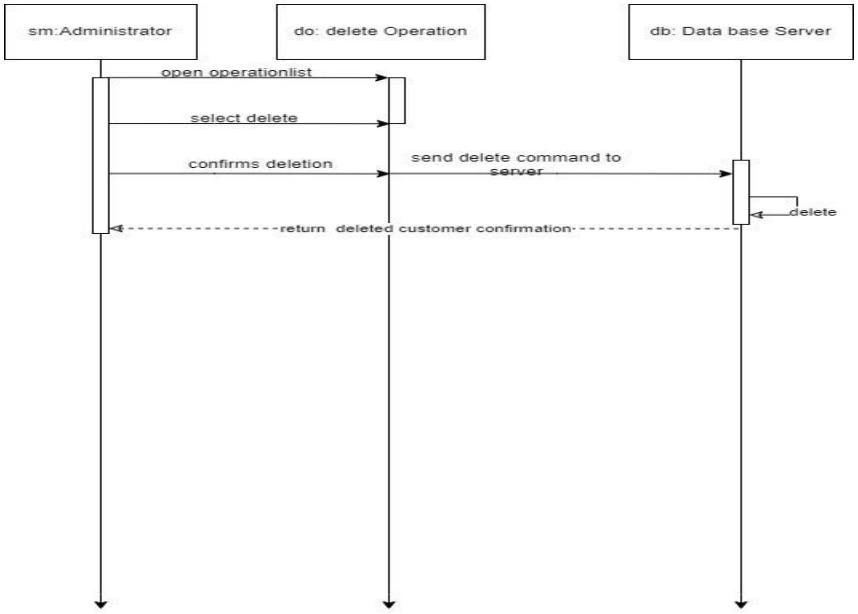
The sales agent click in manage property then the system sends the request for the data of the list of properties existing in the database on the server, there the data is searched and selected, and it returns the property list to the Software Eproma.

3.5.12 Sequence diagramm list customer



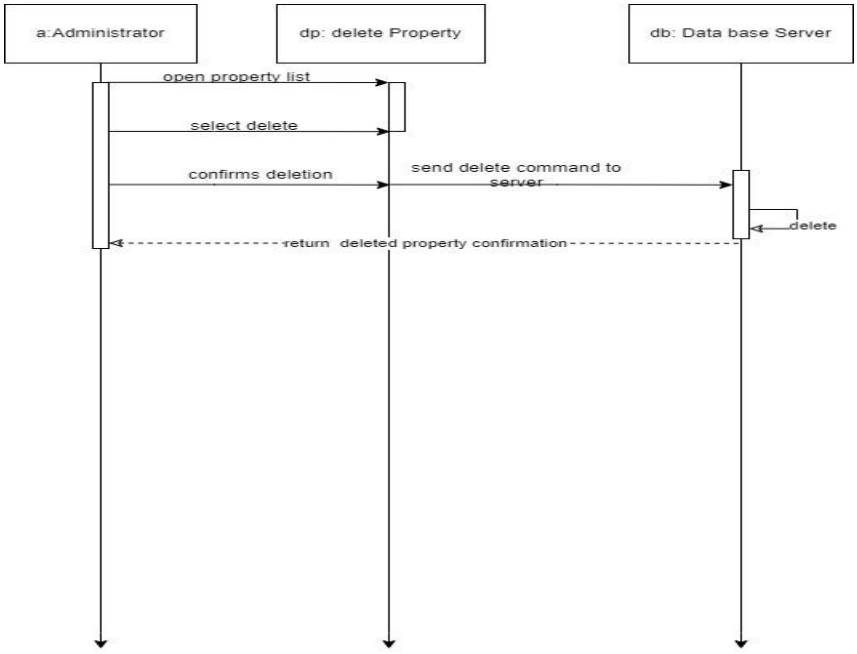
The sales agent click in manage customers then the system sends the data request of the list of existing customers in the database on the server, there the data is searched and selected, and it returns the property list to the Software Eproma.

3.5.13 Sequence diagram delete operation



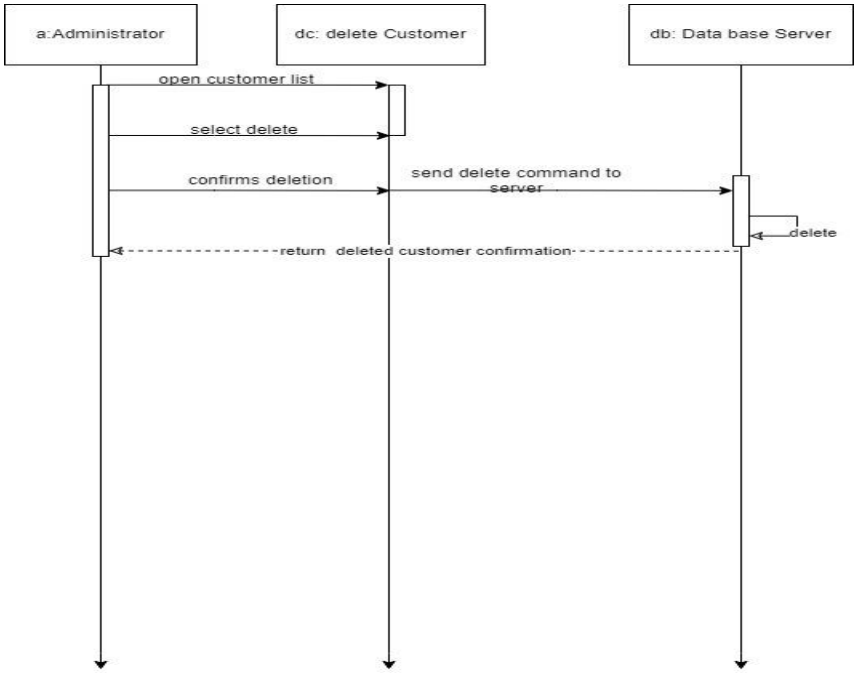
The administrator opens the operation list, then selects delete, confirms delete and the delete command will operate and then the database will "delete" the operation, which will be in database appear as "deleted" and the database will return the operation as "deleted".

3.5.14 Sequence diagram delete property



The administrator opens the property list, then selects delete, confirms delete and the delete command will operate and then the database will "delete" the property, which will be in database appear as "deleted" and the database will return the property as "deleted".

3.5.15 Sequence diagram delete customer



The administrator opens the customer list, then selects delete, confirms delete and the delete command will operate and then the database will "delete" the customer, which will be in database appear as "deleted" and the database will return the customer as "deleted".

3.6 general conditions

4 Definitions and abbreviations

term alphabetical]	description

5 User's guide