Dept. of Computer Science & Engineering, University of Dhaka.

Project Report

Object Oriented Programming (CSE-2112)

Application Name

Real Estate Management System

Submitted By

Diptajoy Mistry(34)
Md. Rasel Hossen(39)
Md. Mushiur Rahman(58)

Code:https://github.com/RaselHossen0/Real-Estate-Management-System

A. Requirement Analysis

The name of our software is Real Estate Management System. At present, the need of having one's own property is one of the basic things. Many people don't know where to find the property. And also who are the landowners, who don't know how and where they will sell their property at an acceptable rate. We know in our country there are many land-selling or land-buying agencies or real estate companies. And these real estate agencies don't have any system to handle their daily workload. So, we made a software through which real estate agents can easily list the property with images, cost, and necessary information from the original seller who wants to sell or rent their property. When the buyer comes to the real estate agent, the agent can find properties in their locations that are available for rent or sale. The agent will have the option to calculate the monthly payment for purchasing the property of the buyer. The agent will take pictures from the seller and can show them to the buyer. The agent can edit, remove or add property. And all features will be available in the dashboard. And in the dashboard, the agent can see many statistics of the property like how many buyers have in his system, how many properties are available for sale or how many are available for rent, and how many are registered property sellers.

B. System Design

Property: This class contains the details of a property. It has all types of details of property like the price of the property, the description of the property, location of the property. The property can be of two types, one **Land** and the other **Rent**. Based on this Property has two subclasses.

Land: This class inherits the property class and has special attributes in it. It contains the size of the land, the shape of the land, zoning(Like Dag No., Dolil No. , Land Registration num, etc.), a natural feature like any natural pond, canal, etc., and soil type.
+It has getter and setter methods.
Flat: This class also inherits the property class. It contains the number of rooms, baths, floors, condition(new, renovated, old), amenities (swimming pool, cafe), and accessibility (school, mosque, hospital).
-It is composed through the property manager and person class to have its buyer and seller info.
-It's all getters() and setters() functions.

Property Manager: This class contains a list of properties. The property can have two types of statuses sold or available, Based on this agent can remove or edit the property status. And it have

> **addProperty:** Adds a new property to the collection of properties managed by the PropertyManager. This function takes the necessary parameters to create a new property object and stores it accordingly.
>
> **removeProperty:** Removes a property from the collection of properties managed by the PropertyManager. This function takes an identifier or unique property ID and deletes the corresponding property from the collection.
>
> **getProperties:** Retrieves all properties stored in the PropertyManager's collection. It returns a list of property objects.
>
> **getTotalProperties:** Returns the total number of properties in the collection.
>
> **setTotalProperties:** Sets the total number of properties in the collection.
>
> **getByID:** Retrieves a specific property from the collection based on its unique identifier or property ID.

getPriceOf: Retrieves the price of a specific property based on its identifier.

deleteFromFile: Deletes a property record from the file where the property data is stored.

getType: Retrieves the type of a property (e.g., land, flat) based on its identifier.

getAvailability: Retrieves the availability status of a property (e.g., available, sold, rented) based on its identifier.

getArea: Retrieves the area of a property (e.g., land area, flat area) based on its identifier.

retrieve: Loads property data from a file and initializes the PropertyManager's collection with the retrieved data.

getImagePaths: Retrieves the file paths of images associated with a property based on its identifier.

customer_set: Sets the customer for a specific property based on the property's identifier.

get_num_of_seller: Retrieves the total number of property sellers.

get_num_of_buyer: Retrieves the total number of property buyers.

get_total_sell: Retrieves the total sales made by the PropertyManager.

Customer: This class contains some common information about landowners and buyers of the properties like name, contact information, email, and location.
-This class contains all getters and setters methods.
owned.

-this class is a composition of the property manager class so that agents can identify total properties against an agent.

The software includes the following features:

1. Login: Agents can log in to their accounts using their credentials.
   -> First and foremost, our login page offers a straightforward way for users to access their accounts.

2. Signup: New agents can create an account by providing the necessary details.
   -> For new users, we have a convenient sign-up option, allowing them to create an account hassle-free.We ensure that user information is securely stored in our system for future logins.

3. Forget Password: Agents can recover their account password through a password reset mechanism.
   -> Through security questions and answers provided during registration, users can regain access to their accounts. We prioritize the security of their information throughout this process.

4. Profile: Agents can view their profile information. The profile section automatically gets updated for different emails.
   -> Here,profile information get updated for new signup and new login.
   ->To enhance navigation and provide a more intuitive user interface, we have implemented a sliding side menu bar. This menu bar can be accessed by clicking on the arrow sign and smoothly slides into view, revealing additional options such as a home button and a cancel button. This sliding

feature adds a touch of elegance to the overall user experience. If we click this home button it will allow the user to go to the home page.
->There is a back button also to get back to the previous page

5. **Dashboard:** Provides an overview of key statistics, such as availability, by type, by area, add property, edit property ,remove property, loan calculator, add customer.

    - Availability:Provides an overview of total sell, total buyers, total sellers, and total listed properties.
    - By Area/Type:Agents can view details of listed properties, search properties by area or type (flat/land)
    - Add/Remove/Edit Property:Agents can add new properties, remove existing properties, and edit property information.
    - Loan Calculator: Includes a loan calculator tool to calculate loan details based on user inputs.
    - Add Customer: Agents can manage customer details, including adding new customers.

Use Cases:

- Agent Login: The agent enters their credentials to log in to the system.
- Property Search: The agent searches for properties by area or type.
- Property Addition: The agent adds a new property to the listing.
- Property Editing: The agent modifies the details of an existing property.
- Loan Calculation: The agent uses the loan calculator to calculate loan details.
- Customer Addition: The agent adds a new customer to the system.

Discussion:

The real estate management project for brokers utilizes several major Java features throughout its codebase. Here is an overview of the major features used and their purpose in accomplishing various tasks within the project:

1. Object-Oriented Programming (OOP):

- Classes: The project defines multiple classes, such as PropertyManager, Property, Land, Flat, and Customer, to model different entities and their behaviors.
- Objects: Objects of these classes are created and manipulated to represent properties, customers, and other entities within the system.
- Inheritance: Inheritance is used to establish an inheritance hierarchy between classes. For example, Land and Flat inherit from Property.
- Encapsulation: Private member variables and public getter/setter methods are used to encapsulate and control access to the class properties.

2. Collections:

- ArrayList: Lists such as properties, lands, flats, rents, and sales are defined as ArrayLists to store and manage property-related objects.

3. File Handling:

- FileReader and BufferedReader: These classes are used to read property data from files in the retrieve function of PropertyManager.
- FileWriter and BufferedWriter: These classes are used to write property data to files in the deleteFromFile function of PropertyManager.

4. Exception Handling:

- IOException: IOException is used to handle input/output errors that may occur during file handling operations.
- SQLException: SQLException is used to handle exceptions related to database connections in the retrieve function of PropertyManager.

5. Database Connectivity:

This project used mysql database to store customer data when added .It will help to see the customers of the agent.

6. Method Overriding:

- The getPriceOf and getType methods in PropertyManager override the corresponding methods in the Property class to provide specific implementations.

7. String Manipulation:
- Splitting and Joining Strings: The project utilizes various string manipulation techniques to split and join strings while reading and writing property data from/to files.

8. Conditional Statements and Loops:
- If-else statements and for-each loops are used throughout the project to implement different conditional and iterative behaviors.

9. Integer and String Manipulation:
- Integer.valueOf and Integer.toString methods are used to convert integers to strings and vice versa.

10. Static Members and Static Methods:
- Static lists (properties, lands, flats, rents, and sales) and static methods are utilized in the PropertyManager class to maintain and manipulate property-related data.

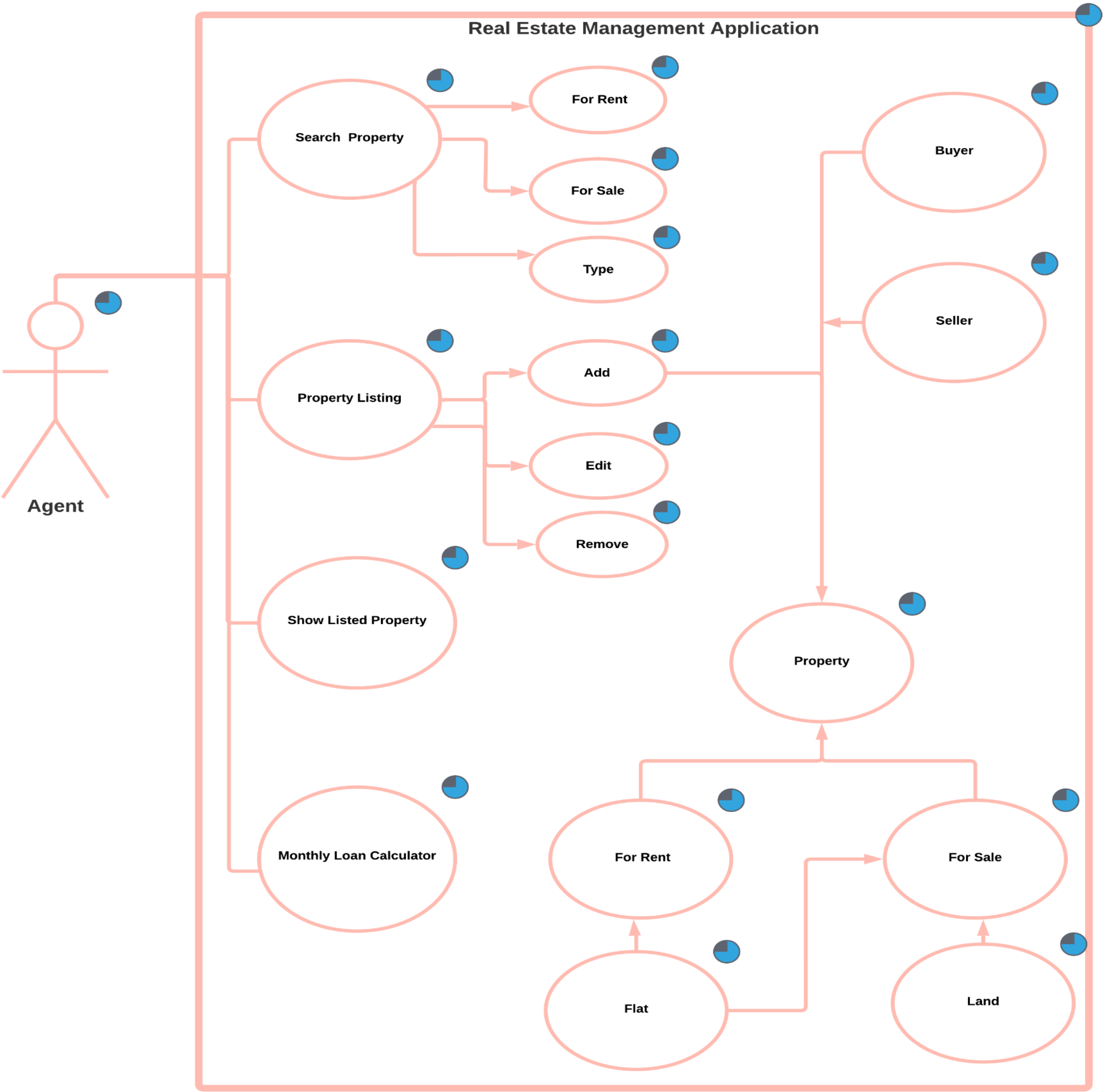
These are the major Java features employed in the real estate management project for brokers. Each feature serves a specific purpose, enabling the project to perform tasks such as adding and removing properties, retrieving data from files, managing property types and availability, interacting with customers, and performing calculations on property data.

Conclusion and Future Work:

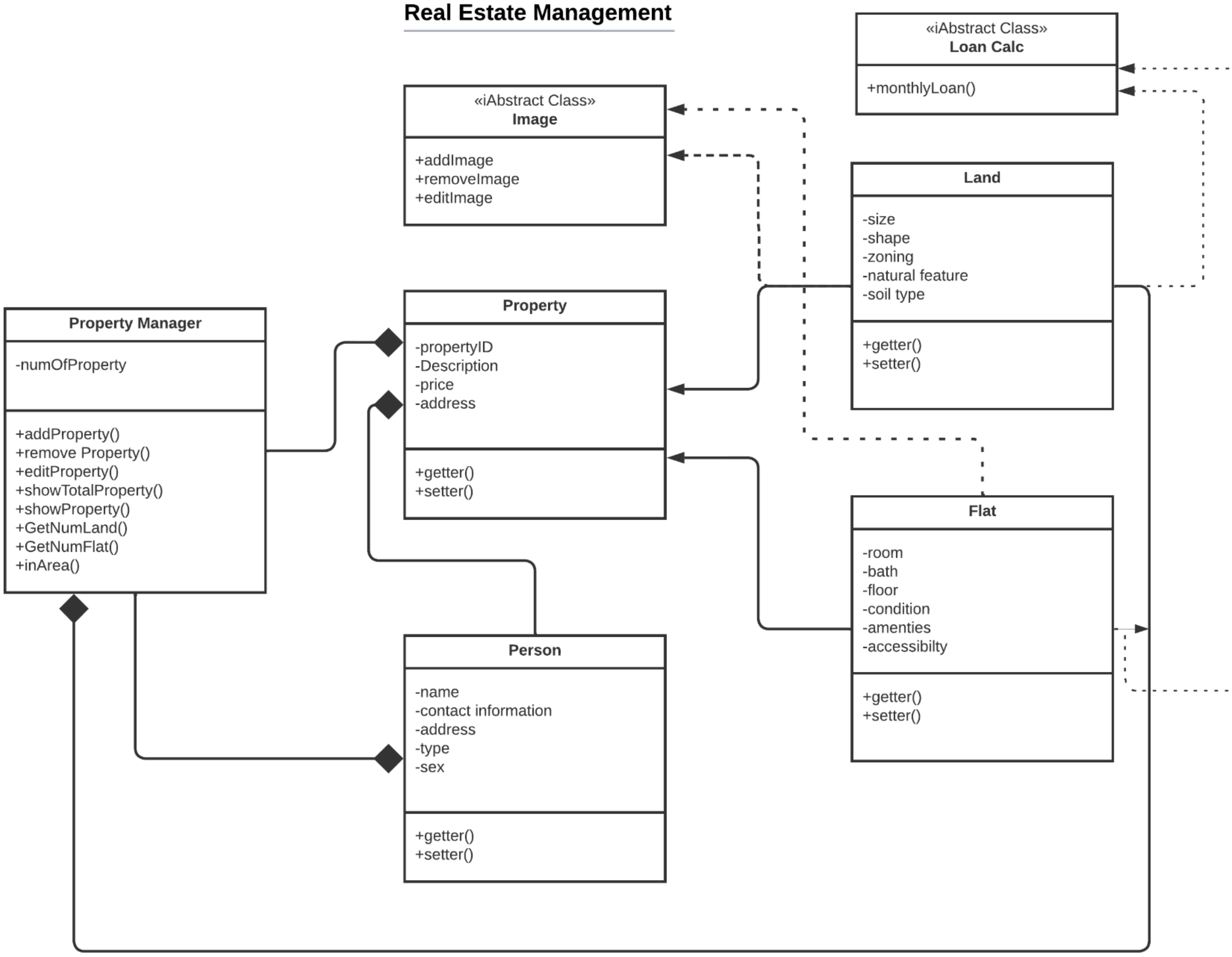
In conclusion, the Real Estate Management System serves as a valuable tool for real estate agents to efficiently manage their property listings and customer interactions. The project successfully addresses the initial requirements and provides essential features such as login, signup, property management, customer management, and a loan calculator.
However, due to the limited time frame, there are potential areas for future work and enhancements to consider:

1. Enhanced Search Functionality: Implement advanced search filters, such as price range, amenities, property size, or location proximity, to help agents find specific properties more effectively.
2. Integration with External APIs: Integrate with external APIs to fetch real-time property data, such as market trends, property valuations, or nearby amenities, to provide more comprehensive information to agents and customers.
3. Reporting and Analytics: Develop reporting and analytics capabilities to generate insights on property performance, customer preferences, and sales trends. This can assist agents in making informed business decisions.
4. Appointment Scheduling: Add functionality to schedule appointments with potential buyers, enabling agents to efficiently manage property viewings and client meetings.
5. Document Management: Introduce a document management system to store essential documents related to properties, such as contracts, agreements, or inspection reports, ensuring easy access and organization.
6. Integration with Email and Notifications: Implement email integration and notifications to keep agents informed about new property inquiries, updates, or important tasks.
7. Mobile Application: Extend the system's functionality by developing a mobile application to provide agents with on-the-go access to property listings, customer details, and essential features.

By incorporating these future enhancements, the Real Estate Management System can become an even more comprehensive and user-friendly solution, enabling real estate agents to streamline their operations, enhance customer experiences, and make informed business decisions.


Real Estate Management Application
Agent
Search Property
For Rent
For Sale
Type
Buyer
Seller
Property Listing
Add
Edit
Remove
Show Listed Property
Property
Monthly Loan Calculator
For Rent
For Sale
Flat
Land

Real Estate Management


«iAbstract Class»
Loan Calc
+monthlyLoan()
«iAbstract Class»
Image
+addImage
+removeImage
+editImage
Land
-size
-shape
-zoning
-natural feature
-soil type
+getter()
+setter()
Property Manager
-numOfProperty
+addProperty()
+remove Property()
+editProperty()
+showTotalProperty()
+showProperty()
+GetNumLand()
+GetNumFlat()
+inArea()
Property
-propertyID
-Description
-price
-address
+getter()
+setter()
Flat
-room
-bath
-floor
-condition
-amenties
-accessibilty
+getter()
+setter()
Person
-name
-contact information
-address
-type
-sex
+getter()
+setter()