QUESTION:

Real Estate Leasing & Rent Collection – Specification Document Problem Statement:

Design and implement a Java console application for a Real Estate Leasing & Rent Collection system that manages properties, units, tenants, leases, invoices, and payments. The application should demonstrate object-oriented principles and maintain accurate occupancy and dues.

Class Requirements:

1. Property
2. Unit
3. Tenant
4. Lease
5. RentInvoice
6. Payment
7. MaintenanceRequest

Business Rules:

1. A unit can be leased only if currently vacant.
2. Lease activation updates unit occupancy immediately.
3. Rent invoices must be generated per lease cycle before payment.
4. Late payments may include penalties as per lease policy.
5. Maintenance requests must link directly to a unit and tenant.

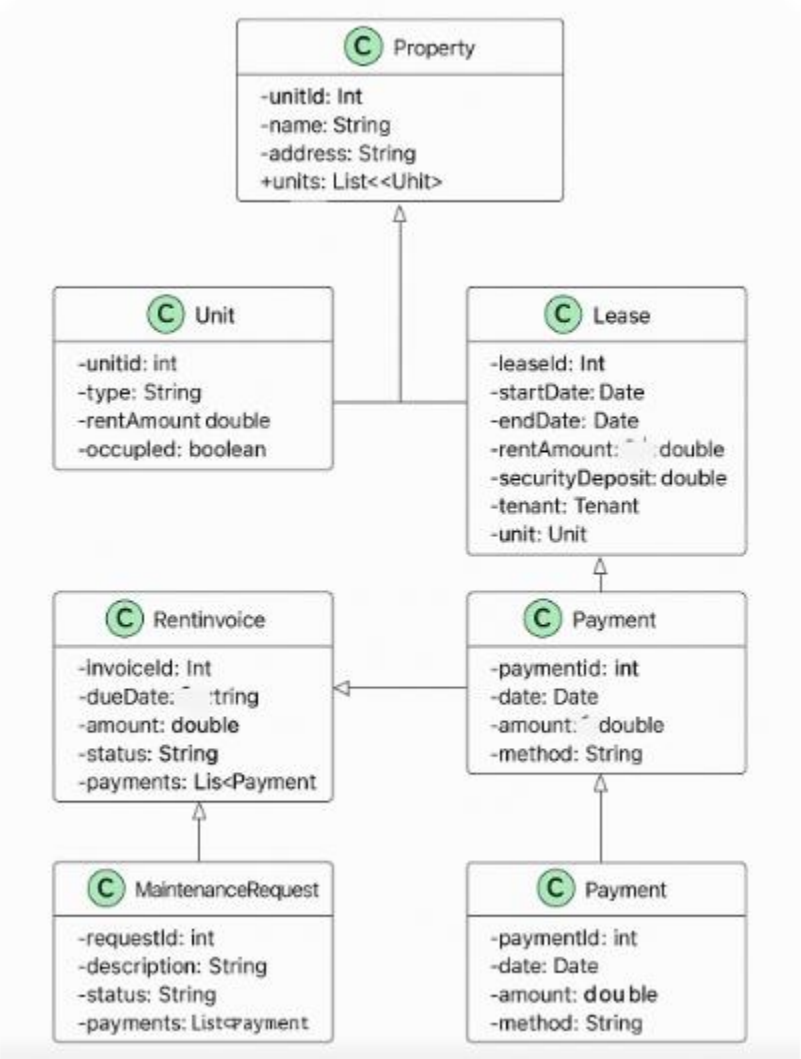
Console Interface Requirements:

1. Menu-driven program: Add Property / Add Unit / Add Tenant / Create Lease / Generate Invoice / Record Payment / Log Maintenance / Display Units / Exit
2. Input validations must be performed for all user entries.
3. Encapsulation must be followed for all attributes.

Expected Output Behavior:

1. Show fixture list and updated results.
2. Show live standings with points and tie-breakers.
3. Show disciplinary logs per team/player.

UML DIAGRAM:



SOURCE CODE:

Property.java:

```
package realestate;

import java.util.ArrayList;
import java.util.List;

public class Property {
    private int propertyId;
    private String name;
    private String address;
    private List<Unit> units;

    public Property(int propertyId, String name, String address) {
        this.propertyId = propertyId;
```

```
        this.name = name;
        this.address = address;
        this.units = new ArrayList<>();
    }

    public void addUnit(Unit unit) {
        units.add(unit);
    }

    public List<Unit> getUnits() {
        return units;
    }

    public void displayDetails() {
        System.out.println("Property ID: " + propertyId + ", Name: " + name + ", Address: " + address);
    }
}
```

Unit.java:

```
package realestate;

public class Unit {
    private int unitId;
    private String unitNumber;
    private boolean isOccupied;
    private Lease lease;

    public Unit(int unitId, String unitNumber) {
        this.unitId = unitId;
        this.unitNumber = unitNumber;
        this.isOccupied = false;
    }

    public boolean isOccupied() {
        return isOccupied;
    }

    public void markOccupied() {
        this.isOccupied = true;
    }
```

```
    public void markVacant() {
        this.isOccupied = false;
    }

    public void assignLease(Lease lease) {
        this.lease = lease;
        this.isOccupied = true;
    }

    public void displayDetails() {
        System.out.println("Unit ID: " + unitId + ", Unit Number: " + unitNumber + ", Occupied: " +
isOccupied);
    }
}
```

Tenant.java:

```
package realestate;

public class Tenant {
    private int tenantId;
    private String name;
    private String phone;
    private String email;

    public Tenant(int tenantId, String name, String phone, String email) {
        this.tenantId = tenantId;
        this.name = name;
        this.phone = phone;
        this.email = email;
    }

    public void displayDetails() {
        System.out.println("Tenant ID: " + tenantId + ", Name: " + name + ", Phone: " + phone + ",
Email: " + email);
    }
}
```

Lease.java:

```
package realestate;

import java.util.Date;
```

```
public class Lease {
    private int leaseId;
    private Date startDate;
    private Date endDate;
    private double monthlyRent;
    private boolean isActive;
    private Tenant tenant;
    private Unit unit;

    public Lease(int leaseId, Date startDate, Date endDate, double monthlyRent, Tenant tenant, Unit
unit) {
        this.leaseId = leaseId;
        this.startDate = startDate;
        this.endDate = endDate;
        this.monthlyRent = monthlyRent;
        this.tenant = tenant;
        this.unit = unit;
        this.isActive = false;
    }

    public void activateLease() {
        if (!unit.isOccupied()) {
            this.isActive = true;
            unit.assignLease(this);
            System.out.println("Lease activated successfully.");
        } else {
            System.out.println("Unit already occupied!");
        }
    }

    public void terminateLease() {
        this.isActive = false;
        unit.markVacant();
        System.out.println("Lease terminated.");
    }

    public void displayDetails() {
        System.out.println("Lease ID: " + leaseId + ", Rent: " + monthlyRent + ", Active: " + isActive);
    }
}
```

RentInvoice.java:

```
package realestate;

import java.util.Date;

public class RentInvoice {
```

```
    private int invoiceId;
    private double amount;
    private Date dueDate;
    private boolean isPaid;

    public RentInvoice(int invoiceId, double amount, Date dueDate) {
        this.invoiceId = invoiceId;
        this.amount = amount;
        this.dueDate = dueDate;
        this.isPaid = false;
    }

    public void applyPenalty(double penalty) {
        this.amount += penalty;
    }

    public void markPaid() {
        this.isPaid = true;
    }

    public void displayDetails() {
        System.out.println("Invoice ID: " + invoiceId + ", Amount: " + amount + ", Paid: " + isPaid);
    }
}
```

Payment.java:

```
package realestate;

import java.util.Date;

public class Payment {
    private int paymentId;
    private double amount;
    private Date date;
    private String method;

    public Payment(int paymentId, double amount, Date date, String method) {
        this.paymentId = paymentId;
        this.amount = amount;
        this.date = date;
        this.method = method;
```

```
    }

    public void getReceipt() {
        System.out.println("Payment ID: " + paymentId + ", Amount: " + amount + ", Method: " +
method + ", Date: " + date);
    }
}
```

MaintenanceRequest.java:

```
package realestate;

import java.util.Date;

public class MaintenanceRequest {
    private int requestId;
    private String description;
    private String status;
    private Date date;

    public MaintenanceRequest(int requestId, String description, Date date) {
        this.requestId = requestId;
        this.description = description;
        this.date = date;
        this.status = "Pending";
    }

    public void updateStatus(String status) {
        this.status = status;
    }

    public void displayDetails() {
        System.out.println("Request ID: " + requestId + ", Desc: " + description + ", Status: " + status + ",
Date: " + date);
    }
}
```

Main.java:

```
package realestate;

import java.util.*;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        List<Property> properties = new ArrayList<>();
        List<Tenant> tenants = new ArrayList<>();

        while (true) {
            System.out.println("\n==== Real Estate Leasing & Rent Collection ====");
            System.out.println("1. Add Property");
            System.out.println("2. Add Unit");
            System.out.println("3. Add Tenant");
            System.out.println("4. Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter Property ID: ");
                    int pid = sc.nextInt(); sc.nextLine();
                    System.out.print("Enter Property Name: ");
                    String pname = sc.nextLine();
                    System.out.print("Enter Address: ");
                    String address = sc.nextLine();
                    properties.add(new Property(pid, pname, address));
                    System.out.println("Property added.");
                    break;

                case 2:
                    if (properties.isEmpty()) {
                        System.out.println("Add a property first!");
                        break;
                    }
                    System.out.print("Enter Property Index (0.." + (properties.size()-1) + "): ");
                    int idx = sc.nextInt();
                    System.out.print("Enter Unit ID: ");
                    int uid = sc.nextInt(); sc.nextLine();
                    System.out.print("Enter Unit Number: ");
                    String unum = sc.nextLine();
                    Unit u = new Unit(uid, unum);
                    properties.get(idx).addUnit(u);
```

```
                System.out.println("Unit added.");
                break;

            case 3:
                System.out.print("Enter Tenant ID: ");
                int tid = sc.nextInt(); sc.nextLine();
                System.out.print("Enter Tenant Name: ");
                String tname = sc.nextLine();
                System.out.print("Enter Phone: ");
                String phone = sc.nextLine();
                System.out.print("Enter Email: ");
                String email = sc.nextLine();
                tenants.add(new Tenant(tid, tname, phone, email));
                System.out.println("Tenant added.");
                break;

            case 4:
                System.out.println("Exiting...");
                sc.close();
                return;

            default:
                System.out.println("Invalid choice!");
        }
    }
}
```

OUTPUT:'

GITHUB REPOSITORY LINK:

https://github.com/717824f157-varshini/Realestate