

**Lahore Campus**

**OOP**

**Plantation Management System**

**Submitted to:** Sir Shahid Bhatti

**Submitted by:** Hafsa & Abdullah

**Roll no:** FA24-BCS-038 & 141

**Section:** B

**Date:** 15 May,2025

**Submitted to:** Mr. Abdul-Karim

**Submitted by:** Abdullah Saleem

**Roll no:** FA24-BCS-141

**Section:** B

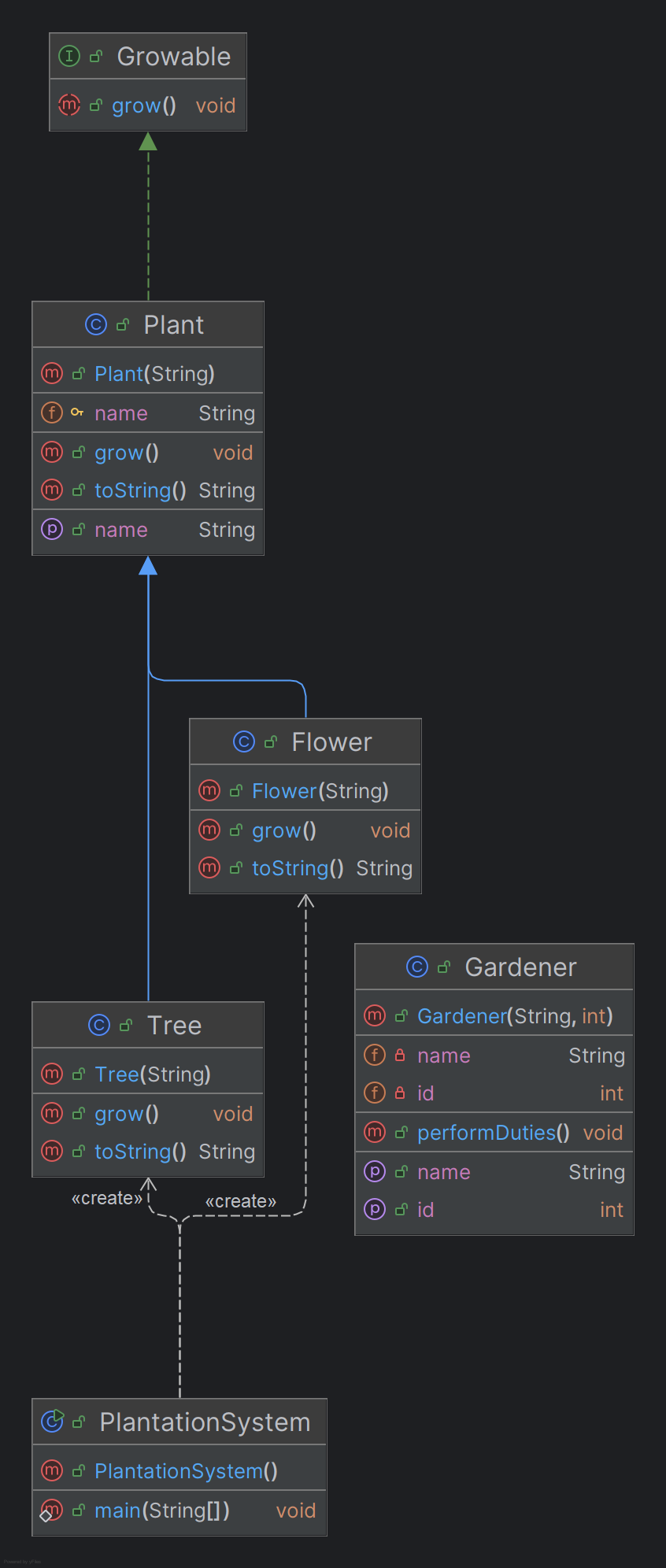
**Date:** 7 Nov,2024

**Plantation Management System**

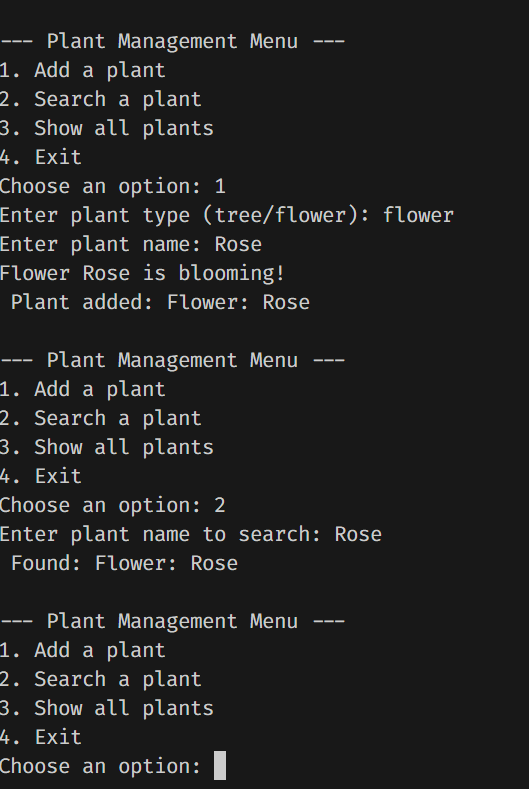
**Abstract:**

The Plant Management System is a user-friendly nursery application designed to help manage plants, users, and daily operations more efficiently. It supports three roles: Admin, Customer, and Gardener. Admins manage the system, customers can view, buy, and favorite plants and while gardeners handle plant care. The system detects plant diseases, and provides zone-specific care tips. Overall it combines plant care, smart recommendations, and easy management in one place.

**UML Diagram**

****

**Output:**

****

**Source Code**

**Tree.java**

public class Tree extends Plant {

    public Tree(String name) {

        super(name);

    }

    @Override

    public void grow() {

        System.out.println("Tree " + name + " is growing taller!");

    }

    @Override

    public String toString() {

        return "Tree: " + name;

    }

}

**Gardener.java**

public class Gardener {

    private String name;

    private int id;

    public Gardener(String name, int id) {

        this.name = name;

        this.id = id;

    }

    public void performDuties() {

        System.out.println("Gardener " + name + " is taking care of the zone.");

    }

    public String getName() {

        return name;

    }

    public int getId() {

        return id;

    }

}

**Plantation Management System.**

import java.util.ArrayList;

import java.util.Scanner;

public class PlantationSystem {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        ArrayList<Plant> plants = new ArrayList<>();

 while (true) {

   System.out.println("\n--- Plant Management Menu ---");

   System.out.println("1. Add a plant");

    System.out.println("2. Search a plant");

    System.out.println("3. Show all plants");

    System.out.println("4. Exit");

    System.out.print("Choose an option: ");

    int choice = scanner.nextInt();

       scanner.nextLine();

    if (choice == 1) {

    System.out.print("Enter plant type (tree/flower): ");

    String type = scanner.nextLine().trim().toLowerCase();

    System.out.print("Enter plant name: ");

    String name = scanner.nextLine().trim();

         Plant newPlant = null;

        if (type.equals("tree")) {

             newPlant = new Tree(name);

             } else if (type.equals("flower")) {

                  newPlant = new Flower(name);

                } else {

                    System.out.println(" Invalid type. Try 'tree' or 'flower'.");

                    continue;

                }

                newPlant.grow();

                plants.add(newPlant);

                System.out.println(" Plant added: " + newPlant);

            } else if (choice == 2) {

                System.out.print("Enter plant name to search: ");

                String searchName = scanner.nextLine().trim();

                boolean found = false;

                for (Plant p : plants) {

                    if (p.getName().equalsIgnoreCase(searchName)) {

                        System.out.println(" Found: " + p);

                        found = true;

                        break;

                    }

                }

                if (!found) {

                    System.out.println(" Plant not found.");

                }

            } else if (choice == 3) {

                if (plants.isEmpty()) {

                    System.out.println("No plants in the system.");

                } else {

                    System.out.println(" All Plants:");

                    for (Plant p : plants) {

                        System.out.println("- " + p);

                    }

                }

            } else if (choice == 4) {

                System.out.println(" Exiting system. Goodbye!");

                scanner.close();

                return;

            } else {

                System.out.println(" Invalid option. Try again.");

            }

        }

    }

}

**Growable.java**

public interface Growable {

    void grow();

**Plant.java**

public class Plant implements Growable {

    protected String name;

    public Plant(String name) {

        this.name = name;

    }

    public String getName() {

        return name;

    }

    @Override

    public void grow() {

        System.out.println(name + " is growing.");

    }

    @Override

    public String toString() {

        return "Plant: " + name;

    }

}

**Tree.java**

public class Tree extends Plant {

    public Tree(String name) {

        super(name);

    }

    @Override

    public void grow() {

        System.out.println("Tree " + name + " is growing taller!");

    }

    @Override

    public String toString() {

        return "Tree: " + name;

    }

}