```
package Phase1Projects;

import java.util.ArrayList;
import java.util.Scanner;

class Camera {
    private String model;
    private boolean available;

    public Camera(String model) {
        this.model = model;
        this.available = true;
    }

    public String getModel() {
        return model;
    }

    public boolean isAvailable() {
        return available;
    }

    public void setAvailable(boolean available) {
        this.available = available;
    }
}

class CameraRentalSystem {
    private ArrayList<Camera> cameraInventory = new ArrayList<>();

    public CameraRentalSystem() {
```

```
    initializeInventory();
}
private void initializeInventory() {
    cameraInventory.add(new Camera("Monitech"));
    cameraInventory.add(new Camera("Canon"));
    cameraInventory.add(new Camera("Wikico"));
    cameraInventory.add(new Camera("Nikon"));
    cameraInventory.add(new Camera("Vjianger"));
    cameraInventory.add(new Camera("Sony"));
    cameraInventory.add(new Camera("Kodak"));
    cameraInventory.add(new Camera("Saneen"));
    cameraInventory.add(new Camera("G-Anica"));
    // Add more cameras as needed
}


public void displayMenu() {
    System.out.println("\n1. Display Available Cameras for the Rental Purpose");
    System.out.println("2. Camera for Rent");
    System.out.println("3. Return a Camera that have been rented");
    System.out.println("4. Display Rented Cameras");
    System.out.println("5. Exit from the purchase");
    System.out.print("Enter your choice to purchase or rent or return camera: ");
}

public int getChoice() {
    Scanner scanner = new Scanner(System.in);
    return scanner.nextInt();
}

public void displayAvailableCameras() {
```

```
        System.out.println("\n--- Available Cameras for the rental purpose ---");
        for (Camera camera : cameraInventory) {
            if (camera.isAvailable()) {
                System.out.println(camera.getModel());
            }
        }
    }

    public void rentCamera() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the model of the camera to rent: ");
        String model = scanner.nextLine().trim();  // Trim to remove leading and trailing spaces

        boolean found = false;

        for (Camera camera : cameraInventory) {
            if (camera.getModel().equalsIgnoreCase(model)) {
                if (camera.isAvailable()) {
                    camera.setAvailable(false);
                    System.out.println("Successfully rented the camera: " + model);
                    found = true;
                    break;
                } else {
                    System.out.println("The camera " + model + " is already rented. Please choose another
camera.");
                    return;
                }
            }
        }
```

```
        if (!found) {
            System.out.println("The model " + model + " is not available or not found. Please try again.");
        }
    }


    public void returnCamera() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the model of the camera to be returned: ");
        String model = scanner.nextLine();

        for (Camera camera : cameraInventory) {
            if (camera.getModel().equalsIgnoreCase(model) && !camera.isAvailable()) {
                camera.setAvailable(true);
                System.out.println("Successfully you have returned camera!");
                return;
            }
            else
            {
            System.out.println("The model you have entered is not available or not found. Please try
again.");
            }
            }
        }

    public void displayRentedCameras() {
        System.out.println("\n--- Cameras that are rented ---");
        for (Camera camera : cameraInventory) {
```

```
            if (!camera.isAvailable()) {
                System.out.println(camera.getModel());
            }
        }
    }
}

public class CameraRentPurchase {
    public static void main(String[] args) {
        CameraRentalSystem rentalSystem = new CameraRentalSystem();

        while (true) {
            rentalSystem.displayMenu();
            int choice = rentalSystem.getChoice();

            switch (choice) {
                case 1:
                    rentalSystem.displayAvailableCameras();
                    break;
                case 2:
                    rentalSystem.rentCamera();
                    break;
                case 3:
                    rentalSystem.returnCamera();
                    break;
                case 4:
                    rentalSystem.displayRentedCameras();
                    break;
                case 5:
                    System.out.println("Exiting  from the purchase. Thank you!");
                    System.exit(0);
```

```
                break;
            default:
                System.out.println("you have entered an invalid choice. Please try again.");
        }
    }
  }
}
```