Camera.java

```
package jsp.camera.rental;

public class Camera {
    private String brand;
    private String model;
    private double rentalAmount;
    private boolean isRented;

    public Camera(String brand, String model, double
rentalAmount) {
        this.brand = brand;
        this.model = model;
        this.rentalAmount = rentalAmount;
    }

    public String getBrand() {
        return brand;
    }

    public String getModel() {
        return model;
    }

    public double getRentalAmount() {
        return rentalAmount;
    }

     public boolean isRented() {
          return isRented;
     }

     public void setRented(boolean isRented) {
          this.isRented = isRented;
     }


}
```

Wallet.java

```
package jsp.camera.rental;

public class Wallet {
    private double balance;

    public Wallet() {
        this.balance = 0.0;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public void deposit(double amount) {
        balance += amount;
    }
}
```

CameraRentalApp.java

```
package jsp.camera.rental;

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

public class CameraRentalApp {
   private List<Camera> availableCameras;
   private Wallet userWallet;

   public CameraRentalApp() {
      availableCameras = new ArrayList<>();
      userWallet = new Wallet();
   }

   public void addCamera(Camera camera) {
      availableCameras.add(camera);
   }

   public void removeCamera(int cameraIndex) {
      if (cameraIndex >= 1 && cameraIndex <= availableCameras.size()) {
         availableCameras.remove(cameraIndex - 1);
         System.out.println("Camera removed successfully.");
      } else {
         System.out.println("Invalid camera selection.");
      }
   }

   public void displayAvailableCameras() {
      if (availableCameras.isEmpty()) {
         System.out.println("No Data Present at This Moment.");
      } else {
         System.out.println("Available Cameras:");
         for (int i = 0; i < availableCameras.size(); i++) {
            Camera camera = availableCameras.get(i);
            System.out.println((i + 1) + ". Brand: " + camera.getBrand() +
                  ", Model: " + camera.getModel() +
                  ", Rental Amount: $" + camera.getRentalAmount() + " per day "+
checkAvailability(camera));
         }
      }
   }
   private String checkAvailability(Camera camera) {
```

```
            return camera.isRented()? "Rented":"availavle";
    }

    public void rentCamera(int cameraIndex) {
        if (cameraIndex >= 1 && cameraIndex <= availableCameras.size()) {
            Camera selectedCamera = availableCameras.get(cameraIndex - 1);
            if (userWallet.getBalance() >= selectedCamera.getRentalAmount()) {
                userWallet.setBalance(userWallet.getBalance() -
selectedCamera.getRentalAmount());
                selectedCamera.setRented(true);
                System.out.println("Camera rented successfully!");
            } else {
                System.out.println("Insufficient Wallet Amount. Please deposit money.");
            }
        } else {
            System.out.println("Invalid camera selection.");
        }
    }

    public void depositMoney(double amount) {
        userWallet.deposit(amount);
        System.out.println("Deposit successful. Current wallet balance: $" +
userWallet.getBalance());
    }

    public double checkWalletBalance() {
        return userWallet.getBalance();
    }


}
```

CameraMain.java

```
package jsp.camera.rental;

import java.util.Scanner;

public class CameraMain {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("-----------------------------");
        System.out.println("|WELCOME TO CAMERA RENTAL APP|");
        System.out.println("-----------------------------");
        System.out.println("USERNAME ");
        String s1 = scanner.nextLine();
        System.out.println("PASSWORD ");
        String s2 = scanner.nextLine();

        CameraRentalApp rentalApp = new CameraRentalApp();

        // Sample camera
        rentalApp.addCamera(new Camera("Canon", "EOS R5", 50.0));
        rentalApp.addCamera(new Camera("Sony", "A7 III", 45.0));

        while (true) {
            System.out.println("1.MY CAMERA");
            System.out.println("2.RENT A CAMERA");
            System.out.println("3.VIEW ALL CAMERAS");
            System.out.println("4.MY WALLET");
            System.out.println("5.EXIT");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.println("1.ADD");
                    System.out.println("2.REMOVE");
                    System.out.println("3.VIEW MY CAMERAS");
                    System.out.println("4.GO TO PREVIOUS MENU");
                    int st = scanner.nextInt();
                    switch (st) {
                        case 1:
                            System.out.print("Enter the brand of the new camera:
");
                            String brand = scanner.next();
                            System.out.print("Enter the model of the new camera:
");
                            String model = scanner.next();
                            System.out.print("Enter the per-day rental amount: ");
                            double rentalAmount = scanner.nextDouble();
                            rentalApp.addCamera(new Camera(brand, model,
rentalAmount));
                            System.out.println("New camera added successfully.");
                            break;

                        case 2:
                            rentalApp.displayAvailableCameras();
                            System.out.print("Enter the number of the camera to
remove: ");
                            int removeIndex = scanner.nextInt();
                            rentalApp.removeCamera(removeIndex);
```

```
                        break;

                    case 3:
                        rentalApp.displayAvailableCameras();
                        break;

                    default:
                        break;
                }
                break;
            case 2:
                System.out.print("Enter the number of the camera you want to
rent: ");
                int cameraIndex = scanner.nextInt();
                rentalApp.rentCamera(cameraIndex);
                break;
            case 3:
                rentalApp.displayAvailableCameras();
                break;
            case 4:
                double balance = rentalApp.checkWalletBalance();
                System.out.println("Current wallet balance: " + balance);
                System.out.println("DO YOU WANT TO DEPOSIT MORE AMOUNT TO YOUR
WALLET ?(1.YES  2.NO)");
                int ans = scanner.nextInt();
                if (ans == 1) {
                    System.out.print("Enter the amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    rentalApp.depositMoney(depositAmount);
                }
                break;
            case 5:
                System.out.println("THANK YOU FOR USING. Goodbye!");
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
                break;
        }
    }
}
```