package CameraRentalApplication;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

import java.util.Scanner;

//create camera class

class Camera {

private String id;

private String brand;

private String model;

private double rentalAmountPerDay;

//create constructor of camera

public Camera(String id, String brand, String model, double rentalAmountPerDay) {

this.id = id;

this.brand = brand;

this.model = model;

this.rentalAmountPerDay = rentalAmountPerDay;

}

public String getId() {

return id;

}

public String getBrand() {

return brand;

}

public String getModel() {

return model;

}

public double getRentalAmountPerDay() {

return rentalAmountPerDay;

}

}

//create class camera user

class User {

private String id;

private double walletAmount;

public User(String id) {

this.id = id;

this.walletAmount = 0.0;

}

public String getId() {

return id;

}

public double getWalletAmount() {

return walletAmount;

}

public void addAmountToWallet(double amount) {

walletAmount += amount;

}

public void ViewAmountToWallet(double amount) {

double view = 0;

walletAmount=view;

}

}

//create class rental application

class RentalApplication {

private List<Camera> cameras;

private List<User> users;

private List<Password> passwords;

private User currentUser;

public RentalApplication() {

this.cameras = new ArrayList<>();

this.users = new ArrayList<>();

this.passwords=new ArrayList<>();

this.currentUser = null;

}

public void addCamera(Camera camera) {

cameras.add(camera);

}

public void addUser(User user) {

users.add(user);

}

public void addPassword(Password password) {

passwords.add(password);

}

public void setCurrentUser(User user) {

this.currentUser = user;

}

public void listCameras() {

//check the camera list empty

if (cameras.isEmpty()) {

System.out.println("No Data Present at This Moment.");

return;

}

//print the rental application

System.out.println("Available Cameras:");

for (Camera camera : cameras) {

System.out.println("ID: " + camera.getId());

System.out.println("Brand: " + camera.getBrand());

System.out.println("Model: " + camera.getModel());

System.out.println("Rental Amount Per Day: $" + camera.getRentalAmountPerDay());

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

}

}

public void rentCamera(String cameraId) {

Camera selectedCamera = findCameraById(cameraId);

if (selectedCamera != null) {

if (selectedCamera.getRentalAmountPerDay() <= currentUser.getWalletAmount()) {

currentUser.addAmountToWallet(-selectedCamera.getRentalAmountPerDay());

System.out.println("Camera rented successfully.");

} else {

System.out.println("Insufficient wallet amount. Please deposit more funds.");

}

} else {

System.out.println("Invalid camera ID.");

}

}

public void addAmountToWallet(double amount) {

currentUser.addAmountToWallet(amount);

System.out.println("Amount added to wallet successfully.");

}

public void ViewAmountToWallet(double amount) {

currentUser.ViewAmountToWallet(amount);

System.out.println("Amount viewed to wallet successfully.");

}

public void navigateToMainContext() {

currentUser = null;

System.out.println("Navigated to the main context.");

}

public void closeApplication() {

System.out.println("Closing the application...");

System.exit(0);

}

private Camera findCameraById(String cameraId) {

for (Camera camera : cameras) {

if (camera.getId().equals(cameraId)) {

return camera;

}

}

return null;

}

public Object getCurrentUser() {

// TODO Auto-generated method stub

return null;

}

}

//create method of rental application

public class CameraRentalApplicationExamples {

public static void main(String[] args) {

RentalApplication rentalApp = new RentalApplication();

rentalApp.addCamera(new Camera("C001", "Canon", "EOS 5D Mark IV", 25.0));

rentalApp.addCamera(new Camera("N001", "Nikon", "D850", 30.0));

rentalApp.addCamera(new Camera("C002", "Canon", "EOS 5D Mark IV", 25.0));

rentalApp.addCamera(new Camera("N002", "Nikon", "D850", 30.0));

rentalApp.addCamera(new Camera("C003", "Canon", "EOS 5D Mark IV", 25.0));

rentalApp.addCamera(new Camera("N003", "Nikon", "D850", 30.0));

//create login page

System.out.println("Welcome to Camera Rental App:");

System.out.println("Please login to continue");

// create username & password

String Username;

String Password;

Scanner scanner = new Scanner(System.in);

System.out.println("UserName");

Username=scanner.nextLine();

System.out.println("Password\n");

Password=scanner.nextLine();

//check username & password

if(Username.equals("VINEET")&& Password.equals("12345")){

System.out.println("Login Successfully");

}

else {

System.out.println("Invalid Username or Password");

}

boolean isApplicationRunning = true;

while (isApplicationRunning) {

System.out.println("Please select an option:");

System.out.println("1. View Cameras");

System.out.println("2. Rent a Camera");

System.out.println("3. Add Amount to Wallet");

System.out.println("3. View Amount to Wallet");

System.out.println("4. Navigate to Main Context");

System.out.println("5. Close Application");

int option = scanner.nextInt();

scanner.nextLine(); // Consume newline character

switch (option) {

case 1:

rentalApp.listCameras();

break;

case 2:

if (rentalApp.getCurrentUser() != null) {

System.out.println("Enter the camera ID:");

String cameraId = scanner.nextLine();

rentalApp.rentCamera(cameraId);

} else {

System.out.println("Please log in to rent a camera.");

}

break;

case 3:

if (rentalApp.getCurrentUser() != null) {

System.out.println("Enter the amount to add to the wallet:");

double amount = scanner.nextDouble();

scanner.nextLine(); // Consume newline character

rentalApp.addAmountToWallet(amount);

} else {

System.out.println("Please log in to add amount to the wallet.");

}

break;

case 4:

if (rentalApp.getCurrentUser() != null) {

System.out.println("View amount to the wallet:");

double amount = scanner.nextDouble();

scanner.nextLine(); // Consume newline character

rentalApp.ViewAmountToWallet(amount);

}else {

System.out.println("Please log in to add amount to the wallet.");

}

break;

case 5:

rentalApp.navigateToMainContext();

break;

case 6:

rentalApp.closeApplication();

isApplicationRunning = false;

break;

default:

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

break;

}

}

//it close the scanner() methods

scanner.close();

}

}