import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Camera {

private int id;

private String brand;

private String model;

private double perDayPrice;

private boolean available;

public Camera(int id, String brand, String model, double perDayPrice, boolean available) {

this.id = id;

this.brand = brand;

this.model = model;

this.perDayPrice = perDayPrice;

this.available = available;

}

public int getId() {

return id;

}

public String getBrand() {

return brand;

}

public String getModel() {

return model;

}

public double getPerDayPrice() {

return perDayPrice;

}

public boolean isAvailable() {

return available;

}

public void setAvailable(boolean available) {

this.available = available;

}

}

public class CameraRentalApp {

private static List<Camera> cameraList = new ArrayList<>();

private static double walletAmount = 10000.0;

public static void main(String[] args) {

addPredefinedCameras();

Scanner scanner = new Scanner(System.in);

System.out.println("WELCOME TO CAMERA RENTAL APPLICATION");

System.out.println("PLEASE LOGIN TO CONTINUE");

System.out.print("USERNAME: ");

String username = scanner.nextLine();

System.out.print("PASSWORD: ");

String password = scanner.nextLine();

if (login(username, password)) {

System.out.println("LOGIN SUCCESSFUL!");

boolean exit = false;

while (!exit) {

System.out.println("\nOPTIONS:");

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();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

myCameraMenu(scanner);

break;

case 2:

rentCamera(scanner);

break;

case 3:

viewAllCameras();

break;

case 4:

myWalletMenu(scanner);

break;

case 5:

exit = true;

break;

default:

System.out.println("INVALID CHOICE! PLEASE TRY AGAIN.");

break;

}

}

} else {

System.out.println("LOGIN FAILED! INVALID USERNAME OR PASSWORD.");

}

scanner.close();

}

private static boolean login(String username, String password) {

// Add your login logic here

return username.equals("bharath") && password.equals("bharath123");

}

private static void addPredefinedCameras() {

// Add some predefined cameras

Camera camera1 = new Camera(1, "Samsung", "DS123", 500.0, true);

Camera camera2 = new Camera(2, "Sony", "HD214", 500.0, true);

Camera camera3 = new Camera(3, "Panasonic", "XC", 500.0, true);

Camera camera4 = new Camera(4, "Canon", "XLR", 500.0, true);

Camera camera5 = new Camera(5, "Fujitsu", "J5", 500.0, true);

Camera camera6 = new Camera(6, "Nikon", "D750", 600.0, true);

Camera camera7 = new Camera(7, "Olympus", "OMD", 600.0, true);

cameraList.add(camera1);

cameraList.add(camera2);

cameraList.add(camera3);

cameraList.add(camera4);

cameraList.add(camera5);

cameraList.add(camera6);

cameraList.add(camera7);

}

private static void myCameraMenu(Scanner scanner) {

System.out.println("\nMY CAMERAS:");

if (cameraList.isEmpty()) {

System.out.println("YOU HAVE NOT RENTED ANY CAMERAS.");

} else {

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

System.out.println("| ID | BRAND | MODEL | PER DAY PRICE (INR)| AVAILABLE |");

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

for (Camera camera : cameraList) {

System.out.printf("| %-2d | %-7s | %-15s | %-17.2f | %-9s |\n",

camera.getId(), camera.getBrand(), camera.getModel(), camera.getPerDayPrice(),

(camera.isAvailable() ? "YES" : "NO"));

}

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

boolean exit = false;

while (!exit) {

System.out.println("\nOPTIONS:");

System.out.println("1. ADD CAMERA");

System.out.println("2. REMOVE CAMERA");

System.out.println("3. GO BACK");

System.out.print("ENTER YOUR CHOICE: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

addCamera(scanner);

break;

case 2:

removeCamera(scanner);

break;

case 3:

exit = true;

break;

default:

System.out.println("INVALID CHOICE! PLEASE TRY AGAIN.");

break;

}

}

}

}

private static void addCamera(Scanner scanner) {

System.out.println("\nADD CAMERA:");

System.out.print("ENTER THE ID OF THE CAMERA: ");

int id = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

System.out.print("ENTER THE BRAND OF THE CAMERA: ");

String brand = scanner.nextLine();

System.out.print("ENTER THE MODEL OF THE CAMERA: ");

String model = scanner.nextLine();

System.out.print("ENTER THE PER DAY PRICE OF THE CAMERA: ");

double perDayPrice = scanner.nextDouble();

scanner.nextLine(); // Consume the newline character

Camera newCamera = new Camera(id, brand, model, perDayPrice, true);

cameraList.add(newCamera);

System.out.println("CAMERA ADDED SUCCESSFULLY!");

}

private static void removeCamera(Scanner scanner) {

System.out.println("\nREMOVE CAMERA:");

System.out.print("ENTER THE ID OF THE CAMERA TO REMOVE: ");

int cameraId = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

boolean removed = false;

for (Camera camera : cameraList) {

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

cameraList.remove(camera);

removed = true;

break;

}

}

if (removed) {

System.out.println("CAMERA REMOVED SUCCESSFULLY!");

} else {

System.out.println("INVALID CAMERA ID. CAMERA NOT FOUND.");

}

}

private static void viewAllCameras() {

System.out.println("\nALL CAMERAS:");

if (cameraList.isEmpty()) {

System.out.println("NO CAMERAS AVAILABLE.");

} else {

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

System.out.println("| ID | BRAND | MODEL | PER DAY PRICE (INR)| AVAILABLE |");

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

for (Camera camera : cameraList) {

System.out.printf("| %-2d | %-7s | %-15s | %-17.2f | %-9s |\n",

camera.getId(), camera.getBrand(), camera.getModel(), camera.getPerDayPrice(),

(camera.isAvailable() ? "YES" : "NO"));

}

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

}

}

private static void rentCamera(Scanner scanner) {

System.out.println("\nRENT A CAMERA:");

viewAllCameras();

System.out.print("ENTER THE ID OF THE CAMERA TO RENT: ");

int cameraId = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

Camera selectedCamera = null;

for (Camera camera : cameraList) {

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

selectedCamera = camera;

break;

}

}

if (selectedCamera != null && selectedCamera.isAvailable()) {

System.out.print("ENTER THE NUMBER OF DAYS TO RENT: ");

int numDays = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

double totalRent = selectedCamera.getPerDayPrice() \* numDays;

if (totalRent <= walletAmount) {

selectedCamera.setAvailable(false);

walletAmount -= totalRent;

System.out.println("CAMERA RENTED SUCCESSFULLY!");

System.out.printf("TOTAL RENT: %.2f INR\n", totalRent);

System.out.printf("WALLET BALANCE: %.2f INR\n", walletAmount);

} else {

System.out.println("INSUFFICIENT FUNDS IN WALLET!");

}

} else {

System.out.println("INVALID CAMERA ID OR CAMERA UNAVAILABLE.");

}

}

private static void myWalletMenu(Scanner scanner) {

System.out.println("\nMY WALLET:");

System.out.printf("WALLET BALANCE: %.2f INR\n", walletAmount);

System.out.print("ENTER THE AMOUNT TO ADD TO WALLET: ");

double amount = scanner.nextDouble();

scanner.nextLine(); // Consume the newline character

if (amount > 0) {

walletAmount += amount;

System.out.println("AMOUNT ADDED TO WALLET SUCCESSFULLY!");

System.out.printf("UPDATED WALLET BALANCE: %.2f INR\n", walletAmount);

} else {

System.out.println("INVALID AMOUNT!");

}

}

}