

# Car Sales and Inventory Store

## Table of Contents

CASE STUDY .....	Error! Bookmark not defined.
Introduction:.....	Error! Bookmark not defined.
Brief description: .....	Error! Bookmark not defined.
FUNCTIONS: .....	Error! Bookmark not defined.
1.    login (string Pin):.....	Error! Bookmark not defined.
2.    Product (double Amount): .....	Error! Bookmark not defined.
3.    Void Buyer_id(string name, double pin, stringemail):: .....	Error! Bookmark not defined.
BLACK BOX TESTING .....	Error! Bookmark not defined.
Boundary Value Analysis: .....	Error! Bookmark not defined.
Function 1: int login Check (string Pin): .....	Error! Bookmark not defined.
Function 2: Void Product (double Amount):.....	Error! Bookmark not defined.
Function 3: Void Buyer_id(string name, double pin, stringemail): .....	Error! Bookmark not defined.
Robust BVA:.....	Error! Bookmark not defined.
Function 1: int login Check (string Pin): .....	Error! Bookmark not defined.
Function 2: Void Product (double Amount):.....	Error! Bookmark not defined.
Function 3: Void Buyer_id string name, double pin, stringemail):.....	Error! Bookmark not defined.
Worst Case BVA testing:.....	Error! Bookmark not defined.
Function 1: int login Check (string Pin): .....	Error! Bookmark not defined.
Function 2: Void Product (double Amount):.....	Error! Bookmark not defined.
Function 3: Void Buyer_id string name, double pin, stringemail):.....	Error! Bookmark not defined.
Robust Worst Case BVA: .....	Error! Bookmark not defined.

**Function 1: int login check(string Pin): ..... Error! Bookmark not defined.**

**Function 2: void Product (string Amount): ..... Error! Bookmark not defined.**

**Function 3: Void Buyer\_id string name, double pin, stringemail): ..... Error! Bookmark not defined.**

Strong Robust Equivalence Class Partitioning: ..... 8

**Function 1: int login Check (string Pin): ..... 8**

**Function 2: Void Product (double Amount): ..... 8**

**Function 3: Void Buyer\_id (string name, double pin, stringemail): ..... 8**

This is an online car and car parts store that has listings of various cars along with their features. It also consists of car parts and accessories. This system allows user to buy car and inventory online. Users can check various car stats including car engine, mileage, tank capacity and other factors. Payments can be made through cash or credit cards. Car booking has other methods for booking and registration and even a test drive registration. The visitor who visits the system must register himself by filling up personal details. After registration user can login to the system with his username and password in order to access the system. User can check various car listing and can view each car features as well as inventory parts. User may select the product and can add the product to shopping cart. Test drive facility requires advance booking. The system also provides users a facility of car loan where users can apply for loans for their car. This application is a combination of both sales and inventory management of the car and car parts. User can easily purchase car or car parts by using this system user does not have to come manually to shop to purchase the product. He can view the car and car parts in effective Graphical User Interface. User can view features of each product and can compare the products in order to purchase a better product.

Functions 1:

void Login (String username, int password);

This function verifies the username and password entered by the user. And if the username and password does not match with the username and password stored in the database then he/she will have to re enter the username and password till he enters the correct username and password.

void get product (String name, Date date);

In this function the user will request for product . He will enter date for that product. Then the product will be reserved for the specific date.

void car-buyer-id (String name, int password, int id);

This function allows user to register himself in the system. He will have to enter name, password and id. Then he will get registered in the system.

Black box Testing:

Worst BVA:

Function 1:

Total cases =  $5^1 = 5$

Min: 1

Min+1: 2

Normal: 200

Max -1: 339

Max: 400

Case	password	Expected output
1	0	Invalid
2	99	Valid
3	258	Valid
4	321	Valid
5	401	Invalid

Function 2:

Total cases =  $5^1 = 5$

Min: a

Min + 1: b

Normal: h

Max-1: y

Max: z

Case	name	Expected output
1	Haider	Valid
2	z	Invalid
3	a	Invalid
4	saad	Valid
5	haider	valid

Function 3:

Input values of name:

Min: a

Min + 1: b

Normal: h

Max-1: y

Max: z

Input values of pin:

Min: 1

Min+1: 2

Normal: 200

Max -1: 399

Max:40

Input values of email:

Min: a

Min + 1: b

Normal: h

Max-1: y

Max: z

Case	name	pin	email	Expected output
1	saad	000	Bse173112@cust.pk	Invalid
2	saad	002	Bse173112@cust.pk	Valid
3	saad	003	Bse173112@cust.pk	Valid
4	saad	004	Bse173112@cust.pk	Valid
5	saad	005	Bse173112@cust.pk	Valid
6	saad	006	Bse173112@cust.pk	Valid
7	saad	007	Bse173112@cust.pk	Valid
8	saad	008	Bse173112@cust.pk	Valid
9	saad	009	Bse173112@cust.pk	Valid
10	saad	010	Bse173112@cust.pk	Valid
11	saad	011	Bse173112@cust.pk	Valid
12	saad	012	Bse173112@cust.pk	Valid
13	saad	013	Bse173112@cust.pk	Valid

14	saad	014	Bse173112@cust.pk	Valid
15	saad	015	Bse173112@cust.pk	Valid
16	saad	016	Bse173112@cust.pk	Valid
17	haider	017	Bse181063@cust.pk	Valid
18	haider	018	Bse181063@cust.pk	Valid

19	haider	019	Bse181063@cust.pk	Valid
20	haider	020	Bse181063@cust.pk	Valid
21	haider	021	Bse181063@cust.pk	Valid
22	haider	022	Bse181063@cust.pk	Valid
23	haider	023	Bse181063@cust.pk	Valid
24	haider	024	Bse181063@cust.pk	Valid
25	haider	025	Bse181063@cust.pk	Valid
26	haider	026	Bse11063@cust.pk	Valid
27	Haider	027	Bse181063@cust.pk	Valid
28	Haider	028	Bse181063@cust.pk	Valid
29	Haider	029	Bse181063@cust.pk	Valid
30	Haider	030	Bse181063@cust.pk	Valid
31	Haider	031	Bse181063@cust.pk	Valid
32	Haider	032	Bse181063@cust.pk	Valid
33	Haider	300	Bse181063@cust.pk	Valid
34	Haider	311	Bse181063@cust.pk	Valid

35	Haider	323	Bse181063@cust.pk	Valid
36	Haider	367	Bse181063@cust.pk	Valid
37	Haider	343	Bse181063@cust.pk	Valid
38	saad	127	Bse173112@cust.pk	Valid
39	saad	160	Bse173112@cust.pk	Valid
40	saad	361	Bse173112@cust.pk	Valid
41	saad	362	Bse173112@cust.pk	Valid
42	saad	363	Bse173112@cust.pk	Valid

43	saad	364	Bse173112@cust.pk	Valid
44	saad	365	Bse173112@cust.pk	Valid
45	saad	366	Bse173112@cust.pk	Valid
46	saad	367	Bse173112cust.pk	Valid
47	saad	368	Bse173112@cust.pk	Valid
48	saad	369	Bse173112@cust.pk	Valid
49	saad	310	Bse173112@cust.pk	Valid
50	saad	130	Bse173112@cust.pk	Valid
51	saad	090	Bse173112@cust.pk	Valid
52	saad	091	Bse173112@cust.pk	Valid
53	haider	092	Bse181063@cust.pk	Valid
54	haider	093	Bse181065@cust.pk	Valid
55	Haider	094	Bse181063@cust.pk	Valid

56	haider	095	Bse181063@cust.pk	Valid
57	haider	096	Bse181063@cust.pk	Valid
58	haider	097	Bse181063@cust.pk	Valid
59	haider	197	Bse181063@cust.pk	Valid
60	haider	401	Bse181063@cust.pk	Invalid

### **Strong Robust Equivalence Class Partitioning:**

It is used to form groups of test inputs of similar behaviour or nature. Test cases are based on classes, not on every input, thereby reduces the time and efforts required to build large number of test cases.

#### **Function 1: int passwordCheck (string password):**

In this case, only one class exists.

password >= 0 && password <= 10

Test cases will be:

password = (normal value, upper robust value, minimum robust value)

password = 5, 8, 9

#### **Function 2: Void Name(double Amount):**

In this case, only one class exists.

Name >= 5 && Name <= 15

Test cases will be:

Name = (normal value, upper robust value, minimum robust value)

Name = 6, 11, 14

#### **Function 3: Void upload buyerid (stringname, double pin string email):**

In this case, there are three variables with 1 input class.

buyerid >= 50 && buyerid <= 150

Test cases will be:



buyerid = (normal value, upper robust value, minimum robust value)

Amount = 40, 170, 10

Pin  $\geq 0$  && pin  $\leq 20$

Test cases will be:

Pin = (normal value, upper robust value, minimum robust value)

Pin= 1, 30, 5

**email** $\geq 0$  && email  $\leq 30$

email= (normal value, upper robust value, minimum robust value)

email=14, , 35, 0.

### **BY COMBINING ALL:**

(Pin normal, Amount normal, buyerid Normal)

(5, 8 , 9))

(Pin normal, Amount normal, buyerid lower robust)

14, , 35, 0.

(Pin normal, Amount lower Robust, buyerid Normal)

40, 170, 10

(Pin lower robust, Amount normal, buyerid normal)

(5, 8 , 9))

(Pin lower robust, Amount lower robust, buyerid lower robust)

14, , 35, 0.

(Pin normal, Amount normal, buyerid upper robust)

(5, 8 , 9))

(Pin normal, Amount upper robust, buyerid normal)

40, 170, 10

(Pin upper robust, Amount normal, buyerid normal)

14, , 35, 0.

(Pin upper robust, Amount upper robust, buyerid upper robust)

(5, 8 , 9))

(Pin normal, Amount lower robust, buyerid upper robust)

40, 170, 10

(Pin lower robust, Amount normal, buyerid lower robust)

14, , 35, 0.

(Pin lower robust, Amount upper robust, buyerid lower robust)

(5, 8 , 9))