

**DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS**

**FACULTY OF SCIENCE AND HUMANITIES**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**VADAPALANI CAMPUS**



## **RECORD NOTE**

**NAME OF THE STUDENT : MOHAMED FIRNAS M A**

**REGISTER NUMBER : RA2232241040034**

**NAME OF THE COURSE : I - MCA / Computer Applications**

**SEMESTER & YEAR : I & I**

**SUBJECT CODE : PCA20D03J**

**SUBJECT NAME : Software Engineering Laboratory**

**APRIL – 2023**

**DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS**

**FACULTY OF SCIENCE AND HUMANITIES**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**VADAPALANI CAMPUS**



**CERTIFICATE**

**Certified to be the Bonafide record of Practical work done by**

**NAME OF THE STUDENT : MOHAMED FIRNAS M A**

**REGISTER NUMBER : RA2232241040034**

**NAME OF THE COURSE : I - MCA / Computer Applications**

**SEMESTER & YEAR : I & I**

**SUBJECT CODE : PCA20D03J**

**SUBJECT NAME : Software Engineering Laboratory**

**In SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, DEPARTMENT OF  
COMPUTER APPLICATIONS Laboratory during the Academic Year 2022-2024 and Submitted for  
M.C.A degree practical examination held on \_\_/\_\_/2023.**

**STAFF**

**HOD**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **CONTENT**

<b>S.NO</b>	<b>DATE</b>	<b>NAME OF THE EXERCISE</b>	<b>PAGE NO</b>	<b>SIGNATURE</b>
<b>1</b>	<b>24/08/2022</b>	<b>LIBRARY MANAGEMENT SYSTEM</b>		
<b>2</b>	<b>26/08/2022</b>	<b>SOFTWARE REQUIREMENT SPECIFICATION (SRS) FOR ONLINE SHOPPING SYSTEM (OSS)</b>		
<b>3</b>	<b>07/09/2022</b>	<b>TEST CASE DESIGN FOR LOGIN SCREEN</b>		
<b>4</b>	<b>23/09/2022</b>	<b>AIRLINE RESERVATION SYSTEM</b>		
<b>5</b>	<b>14/10/2022</b>	<b>TESTING FOR GMAIL REGISTRATION FORM</b>		
<b>6</b>	<b>26/10/2022</b>	<b>TESTING FOR STUDENT REGISTRATION FORM</b>		
<b>7</b>	<b>04/11/2022</b>	<b>WHITE BOX TESTING</b>		
<b>8</b>	<b>09/11/2022</b>	<b>BLACK BOX TESTING</b>		

**EX. NO: 1**

## **LIBRARY MANAGEMENT SYSTEM**

**DATE: 24/08/2022**

### **AIM:**

To write a problem statement, implement the feature of the UML diagram, scope, roles and responsibilities for the library management system.

### **PROBLEM STATEMENT:**

Library Management system is an application which is used to solve the problems that occurs. It helps to maintain the records. It's an effective communication tool that informs the admin about a potential threat.

### **ROLES AND RESPONSIBILITIES:**

- **ADMIN:** The person who creates the system and the database which helps the librarian to work on it.
- **LIBRARIAN:** The person who have access to the database and maintains the data like issue of books, checking whether the issued book is returned or not.
- **MEMBERS:** The person who reads the books, buy the books or rent it from the library.

### **PROBLEM OR CONSTRAINTS:**

There are several problems and errors that library management system should overcome, some of the major problems are:

- Duplicate records must be removed as soon as possible.
- Server may go down anytime so there should be a backup plan.
- The issued book may not be returned on time.
- The books should be re-racked on the exact spot because it helps the other to find the book easily.
- The delayed books should be returned and the collection of fees must be important.

### **SCOPE:**

- There should be a search filter that should be easy for the authorized user to find the books in the libraries.
- Addition and deletion of the data must be present because the new and old records must be updated soon.
- Each book in the library should have a unique ID that identifies can book individually when searching it in database.

### **INPUT:**

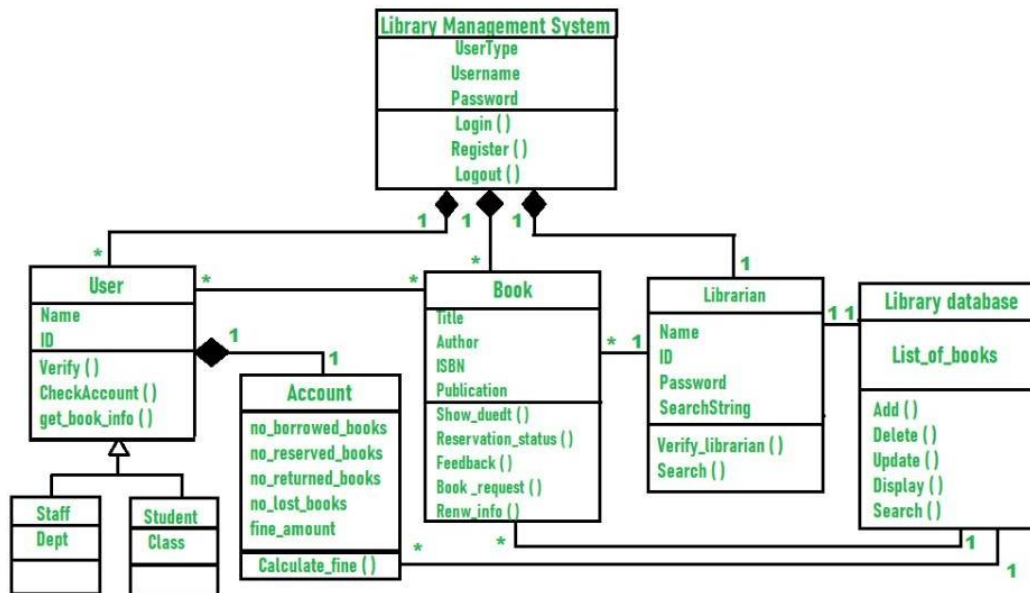
- An authorized user can sign up for the library management system and in order for them to use the application order for them to use the application, they need to be logged in.
- In the search bar, there should be option to sort the data by the price of the book, starting alphabet of the book and so on.
- If the user doesn't return the book on the due date, it should be reminded to both the member and the librarian and the fine must be added automatically.

## UML DIAGRAM:

- UML diagram is a behaviour diagram to visualize.

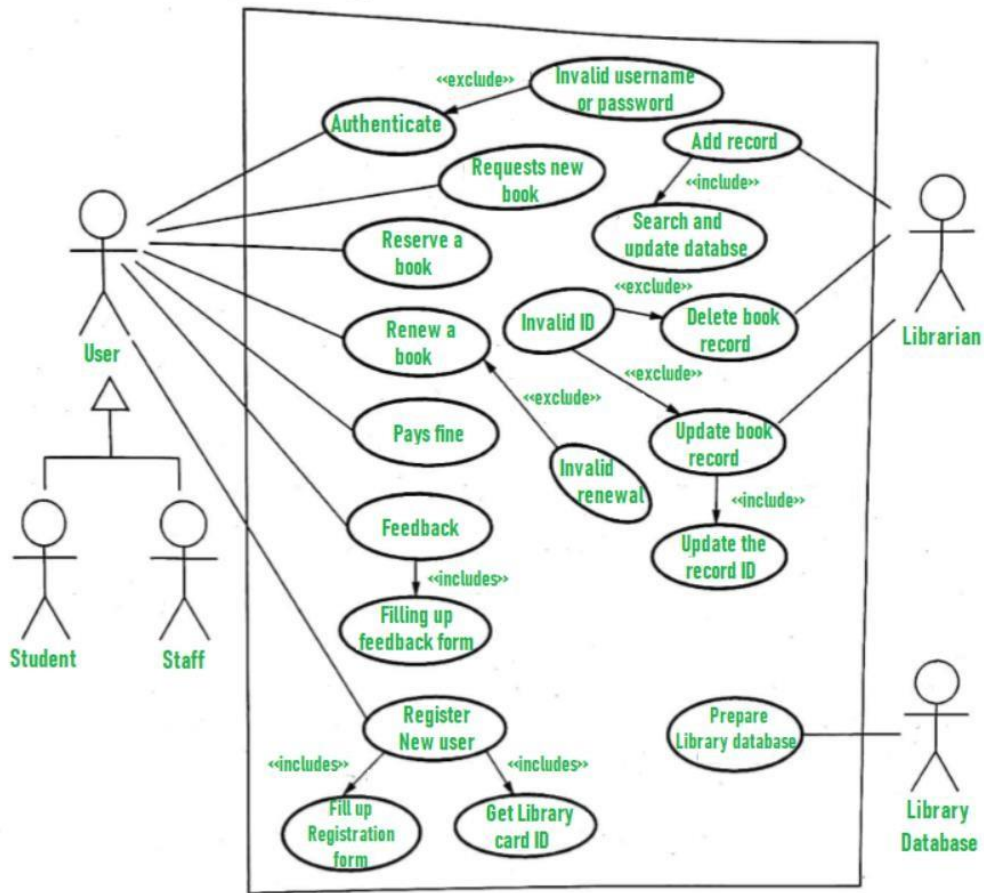
It can be categorized as:

- Class diagram
- Activity diagram
- Interaction diagram
- State chart diagram



**CLASS DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM**

## USE CASE DIAGRAM:



**RESULT:**

Thus the problem statement for the library management has been completed successfully.

**EX.NO: 2**

## **Software Requirement Specification (SRS) for Online Shopping System (OSS)**

**DATE: 26/08/2022**

### **Aim:**

To write a software requirement specification for online shopping system.

### **Purpose:**

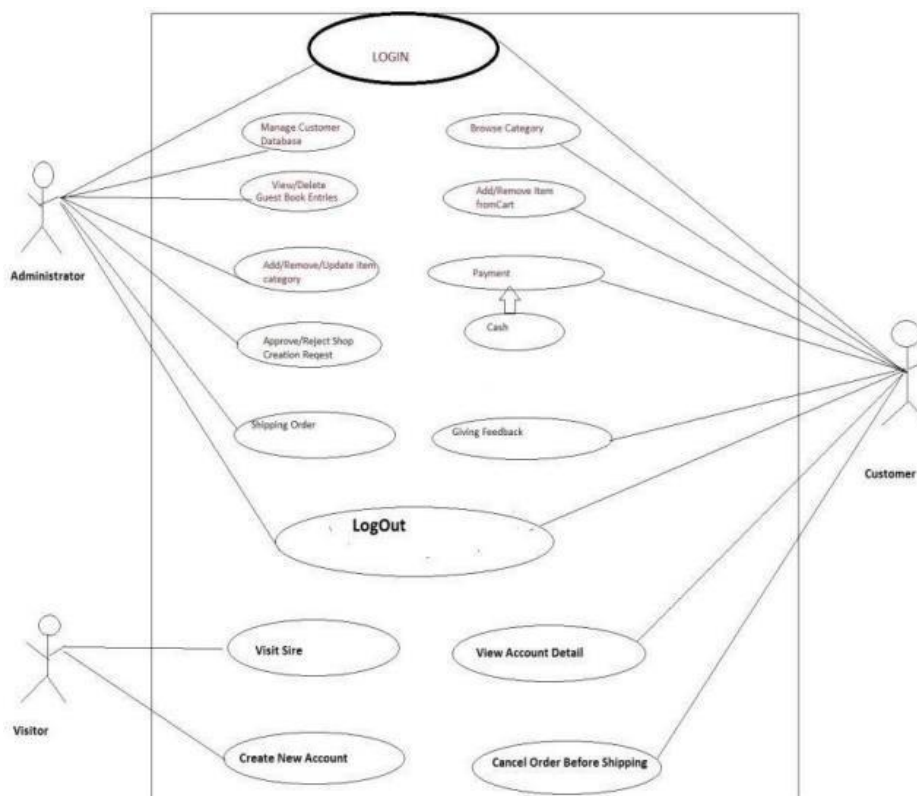
This document is meant to delineate the features of OSS, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other. The Online Shopping System (OSS) for electronics item shop web application is intended to provide complete solutions for vendors as well as customers through a single get way using the internet. It will enable vendors to setup online shops, customer to browse through the shop and purchase them online without having to visit the shop physically. The administration module will enable a system administrator to approve and reject requests for new shops and maintain various lists of shop category.

### **Scope:**

This system allows the customers to maintain their cart to add or remove the product over the internet.

### **Product Perspective:**

This product aimed toward a person who don't want to visit the shop as he might don't get time for that or might not interested in visiting there and dealing with lot of formalities.





**User Characteristics:**

User should be familiar with the terms like login, register, order system etc.

**Principle Actors:**

2 Principle Actors are Customer and Administrator.

**General Constraints:**

A full internet connection is required for OSS.

**Assumptions and Dependencies:**

Working of OSS need Internet Connection.

**Functional Requirements:**

This section provides requirement overview of the system.

- Registration
- Login
- Changes to Cart
- Payment
- Logout
- Report Generation

**Non-Functional Requirements:**

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Compatibility
- Resource Utilization

**Technical Issues:**

This system will work on client-server architecture. It will require an internet server and which will be able to run PHP application. The system should support some commonly used browser such as IE, Mozilla Firefox, chrome etc.

**Interface Requirement:**

Various interfaces for the product could be

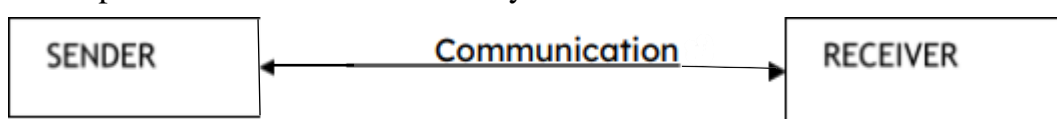
- Login Page
- Registration Form
- There will be a screen displaying information about product that the shop having.
- If the customers select the buy button, then another screen of shopping cart will be opened.
- After ordering for the product, the system will sent one copy of the bill to the customer's Email-address.

**Software Interface:**

1. Operating System: Windows7 Ultimate which supports networking.
2. JAVA development toolkit.

**Communication Interfaces:**

The two parties should be connected by LAN or WAN for the communication purpose.



**Result:** Thus the SRS document for the online shopping system has been successfully written.

**EX NO: 03**

## **TEST CASE DESIGN FOR LOGIN SCREEN**

**DATE: 07/09/2022**

**Aim:**

To write Test Cases in Manual Testing for Check Login Functionality:

**PROBLEM DEFINITION:**

- Test Case 1: Check results on entering valid User Id & Password.
- Test Case 2: Check results on entering Invalid User ID & Password.
- Test Case 3: Check response when a User ID is Empty & Login Button is pressed.
- Test Case 4: Verify that TAB functionality is working properly or not.
- Test Case 5: Verify that Enter/Tab key works as a substitute for the Sign-in button.
- Test Case 6: Verify that the reset button clears the data from all the text boxes in the login form.

**Purpose:**

In any application, 'logging in' is the process to access an application by an individual who has valid user credentials. 'Logging in' is usually used to enter a specific page. Login page is very important for any application in terms of the security aspect.

**Scope:**

- We have to write test cases on each object in the login form.
- We have to write both positive and negative test cases.
- We have to write both functional and non-functional test cases.

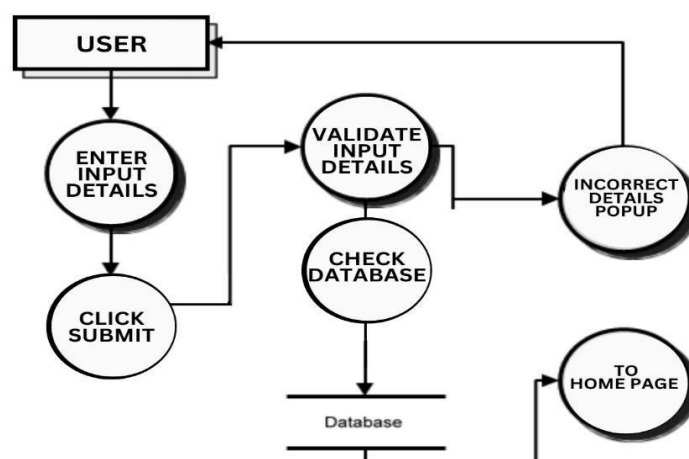
**Hardware interfaces:**

- Hard disk
- Standard output device
- Keyboard and mouse

**Software interfaces:**

- Back End: MS Access 2007
- Front End: Microsoft Visual Basic 6.0

**DATA FLOW DIAGRAM:**



**Test Cases:**

Test ID	Test Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Customer Login with valid Data	→Go to site (URL) →Enter Username →Enter Password →Click Submit	Username=soft123  Password=ware456	User should Login into application	Expected	Pass
TU02	Check Customer Login with invalid Data	→Go to site (URL) →Enter Username →Enter Password →Click Submit	Username=soft@123  Password = ware@456	User should not Login into application	Expected	Pass
TU03	Check response when Username is empty	→Go to site (URL) →Enter only Password →Click Submit	Username=  Password=ware456	Username Invalid	Expected	Pass
TU04	Check response when Password is empty	→Go to site (URL) →Enter only Username →Click Submit	Username=soft123  Password=	Password incorrect	Expected	Pass
TU05	Verify that TAB functionality is working properly or not	→Go to site (URL) →Enter Username →Click TAB →Enter password →Click Submit	Username=soft123  Password=ware456	TAB functionality is working	Expected	Pass
TU06	Verify that TAB functionality is working properly or not	→Go to site (URL) →Enter Username →Click TAB →Enter password →Click Submit	Username=soft123  Password=ware456	TAB functionality is not working.	Unexpected	Fail
TU07	Verify that Enter/Tab key works as a substitute for the Sign-in button.	→Go to site (URL) →Enter Username →Click TAB →Enter password →Click TAB →Click Enter →Click Submit	Username=soft123  Password=ware456	Enter/Tab key works as a substitute for the Sign-in button.	Expected	Pass
TU08	Verify that Enter/Tab key works as a substitute for the Sign-in button.	→Go to site (URL) →Enter Username →Click TAB →Enter password →Click TAB →Click Enter	Username=soft123  Password=ware456	Enter/Tab key does not work as a substitute for the Sign-in button.	Unexpected	Fail

		→Click Submit				
TU09	Verify that the reset button clears the data from all the text boxes in the login form.	→Go to site (URL) →Enter Username →Enter password →Click Reset	Username= soft123  Password= ware456	The reset button clears the data from all the text boxes in the login form.	Expected	Pass
TU10	Verify that the reset button clears the data from all the text boxes in the login form.	→Go to site (URL) →Enter Username →Enter password →Click Reset	Username= soft123  Password= ware456	The reset button does not clears the data from all the text boxes in the login form.	Unexpected	Fail

## OUTPUT:

ALREADY REGISTERED?

Email address

Password

check Login  
Functionality

**Result:** Thus the test cases for the login screen functionality has been executed successfully.

**EX NO: 04**

## **AIRLINE RESERVATION SYSTEM**

**DATE: 23/09/2022**

### **PROBLEM DEFINITION:**

Ticket reservation system for airlines has to be developed. The system developed should contain the following features:

1. The system should contain the following features
2. Search for information about the flight by means of flight number and destination
3. While displaying information about the flight it has to provide availability of seats.
4. While reserving tickets the system obtains following information from the user
  - Passenger Name,
  - Sex,
  - Age,
  - Address.
  - Credit Card Number,
  - Bank Name.
  - Flight number,
  - Flight name,
  - Date of Journey and
  - Number of tickets to be booked.
5. Based on the availability of tickets, the ticket has to be issued. The ticket issued should contain the following information –ticket number, flight no, flight name, date of journey, number of passengers, sex, age and departure time.
6. Cancellation of booked tickets should be available.

### **SRS DOCUMENT FOR AIRLINE RESERVATION SYSTEM:**

#### **INTRODUCTION:**

##### **Purpose:**

The purpose of this SRS is to describe the requirements involved in developing a Airline Reservation system (ARS). The intended audience is any person who wants to reserve or cancel tickets or tocheck the availability of Airline tickets.



**Scope:**

- The product is titled Airline Reservation system (ARS).
- The product will perform the following task.
- The software that is being developed can be used to check the availability of the flight tickets for the specified flight, destination and date of journey.
- If the tickets are available to the user's needs and specification, then the software provide a facility to book the tickets.
- If the passengers want to cancel the tickets, he can use the cancellation module of the Airline Reservation System.

**Hardware interfaces:**

- Hard disk
- Standard output device
- Keyboard and mouse

**Software interfaces:**

- Back End: MS Access 2007
- Front End: Microsoft Visual Basic 6.0

**Operations:**

The user mode enables the end-users to do the end user operations like checking the availability, reserving and canceling of flight tickets.

**Viewing Flight Details:**

The user must have the access up-to-date information about the flights including ,

- Flight number
- Flight Name
- Flight route (Start and Destination stations)
- Flight timings
- Seat availability.

**Reserving Tickets:**

The user must be able to reserve tickets after selecting,

- Flight number
- Flight Route

**Canceling Tickets:**

The user must be able to cancel tickets that he has earlier reserved by quoting the ticket number, credit card number and bank name.

### Constraints:

At the time of reservation, each user is provided a unique ticket number that must be used for further operation like cancellation. Hence the user is required to remember or store this number carefully.

### SPECIFIC REQUIREMENTS:

#### Logical Database Requirements:

- The system should contain databases that include all necessary information for the product to function according to the requirements. These include relations such as flight details, reservation details, and cancellation details.
- The user details refer to the information such as flight number and name, start and destination stations, seat availability.
- Reservation details refer to personal information that is obtained from the user  
At the time of reservation, the passenger is provided a unique ticket no that could be used at the time of cancellation.
- While displaying any information about the flight it has to provide the following information.

Flight no and name

- Availability of seats for the particular flight  
The flight timing  
The passenger personal details should be obtained for reserving the tickets.

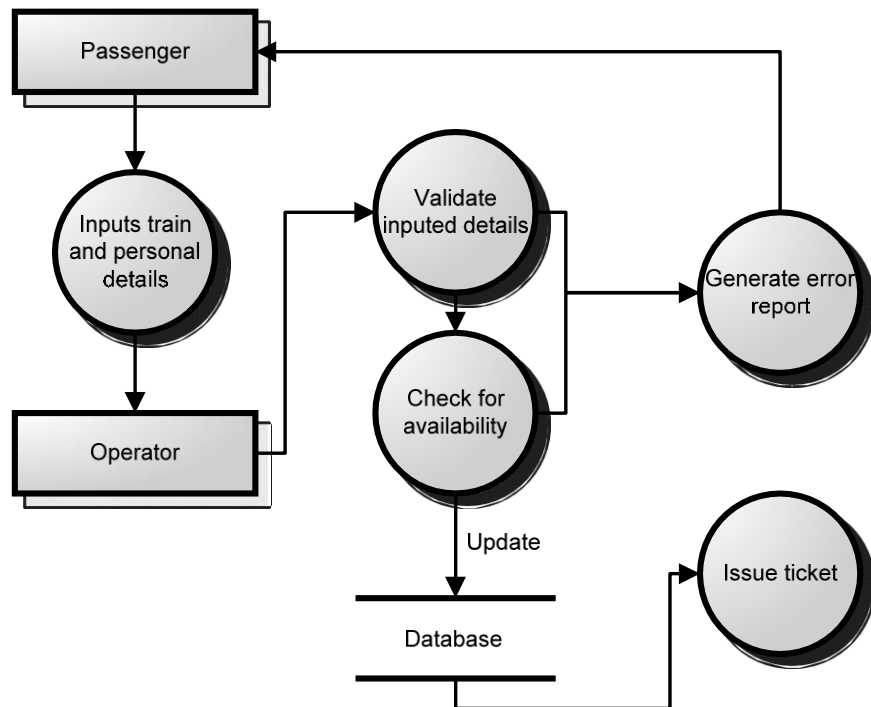
### DATA STRUCTURES:

#### FLIGHT DETAILS:

FIELD NAME	TYPE	CONSTRAINTS
ROUTE_NAME	TEXT	NOT NULL
FLIGHT_NO	NUMBER	NOT NULL
SEATS_AVAIL	NUMBER	
JOURNEY_DATE	DATE/TIME	
DEP_TIME	DATE/TIME	
ARR_TIME	DATE/TIME	
COST	NUMBER	

**PASSENGER DETAILS:**

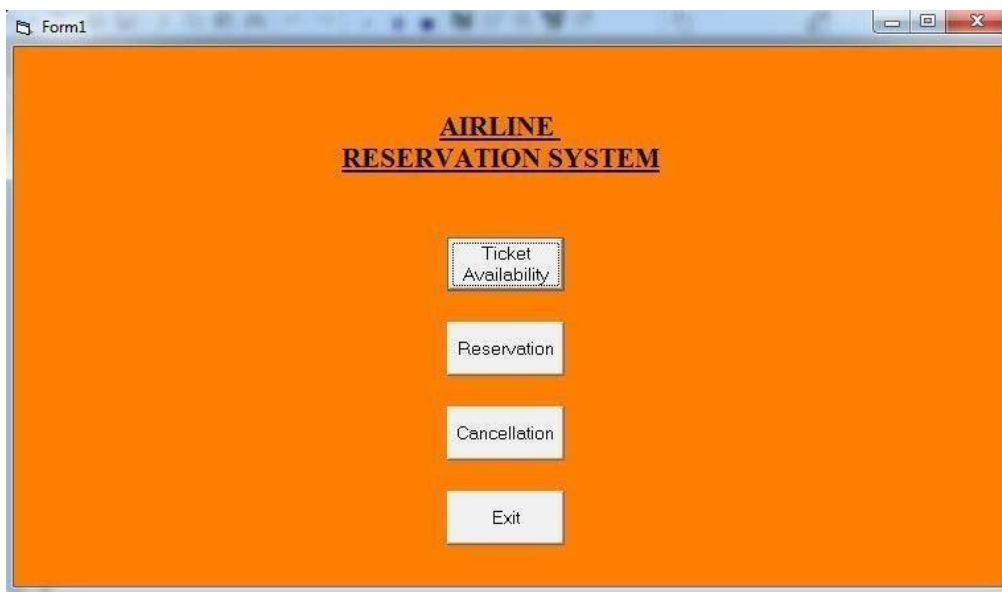
FIELD NAME	TYPE	CONSTRAINTS
TICKET_NO	AUTONUMBER	NOT NULL
NAME	TEXT	NOT NULL
GENDER	TEXT	
ADDRESS	TEXT	
CC_NO	NUMBER	NOT NULL
BANK_NAME	TEXT	
NO_OF_TICKETS	NUMBER	

**DATA FLOW DIAGRAM:****TESTING:**

FORM NAME	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
MAIN MENU FORM	Menu Option	Required Form must be displayed	Required form was displayed.	Pass
TICKET AVAILABILITY FORM	Flight route or Flight name	Flight seats availability must be displayed	Flight seats availability are displayed.	Pass
RESERVATION FORM	Personal details were entered.	Ticket must be booked and database updated.	Ticket was booked and database was updated.	Pass
CANCELLATION FORM	Ticket number was entered.	Ticket must be cancelled and database must be updated.	Ticket was cancelled and database was updated..	Pass

## SAMPLE FORMS:

### MAIN MENU FORM:



The screenshot shows a window titled "Form1" with an orange background. At the top center, the text "AIRLINE  
RESERVATION SYSTEM" is displayed in a bold, black, serif font. Below this title, four buttons are arranged vertically in the center: "Ticket Availability", "Reservation", "Cancellation", and "Exit". Each button has a thin black border and a light gray fill.

### TICKET AVAILABILITY:



The screenshot shows a window titled "Form2" with an orange background. At the top center, the text "TICKET  
AVAILABILITY" is displayed in a bold, black, serif font. Below the title, there is a "Select route" label followed by a dropdown menu. A "Submit" button is positioned below the dropdown. Further down, there are six pairs of input fields, each with a label on the left and a text box on the right: "Flight No", "No of Seats available", "Date of Journey", "Departure Time", "Arrival Time", and "Cost of Ticket". At the bottom center, there is a button labeled "Back to Main Menu".

## RESERVATION FORM:

Form3

**TICKET RESERVATION**

Route

Submit

Passenger Name

Age

Sex

Address

Credit Card No

Bank Name

No of tickets

Book Tickets Cancel

## CANCELLATION FORM:

Form4

**CANCELLATION**

Ticket No

Cancel Ticket

Back to Main Menu

**RESULT:**

Thus the online Airline Reservation System was implemented.

**EX NO: 05                      TESTING FOR GMAIL REGISTRATION FORM**  
**DATE: 14/10/2022**

**Aim:**

To write Test Cases in Manual Testing for Gmail Registration Form.

**Problem Definition:**

Test cases have to be developed for Gmail Registration page. With Gmail, user email is stored safely in the cloud. User can get to messages from any computer or device with a web browser. You can also quickly organize and find important email, as well as read and draft email without an internet connection.

For Successful Registration, the user will have to provide the following information -

- First Name
- Last Name
- Username
- New Password
- Confirm Password
- Phone Number
- Gender
- Recovery E-mail

**Software Requirement Specification:**

**Purpose:**

This Document describes the requirements of Gmail Registration webpage. Gmail is used for sending and receiving e-mail message worldwide.

**Scope:**

To allow new users to quickly create a Gmail account. The main Function for creating Gmail is to help users emailing, chatting, and archiving your mail and conversations very easy and convenient.

## Definitions:

**User:** User is a person who access website for emailing, chatting, and archiving your mail and conversation. User can also be named as Customer.

**Account:** Account term basically refers to an identity on Gmail website. A user can easily make an account on Gmail website by just giving some credentials like name, address, E-mail id, phone number.

## Functional Requirements:

It includes the features like signup, login, etc.

Login or Signup -

- This functionality is required to create an account or if you have one then, you can enter the Gmail by logging in.
- The user must create an account by giving some of their personal details.
- Input the details like name, gender, password, mobile no., location to create an account.
- The details will be stored in the database with other users.
- After creating the account or if user already have an account, the user should login to the Gmail by giving email id and password which was entered during signing up.
- The user is now entered in Gmail and can send and receive mails.

## Hardware interfaces:

- Hard disk
- Standard output device
- Keyboard and mouse

## Software interfaces:

- Back End: MS Access 2007
- Front End: Microsoft Visual Basic 6.0

## Data Structures:

Field Name	Type	Constraints
First Name	Text	Not Null
Last Name	Text	Not Null
Username	Text	Not Null
Password	Text, Number	Not Null
Confirm Password	Text, Number	Not Null
Phone No	Number	
DOB	Date	Not Null
Gender		

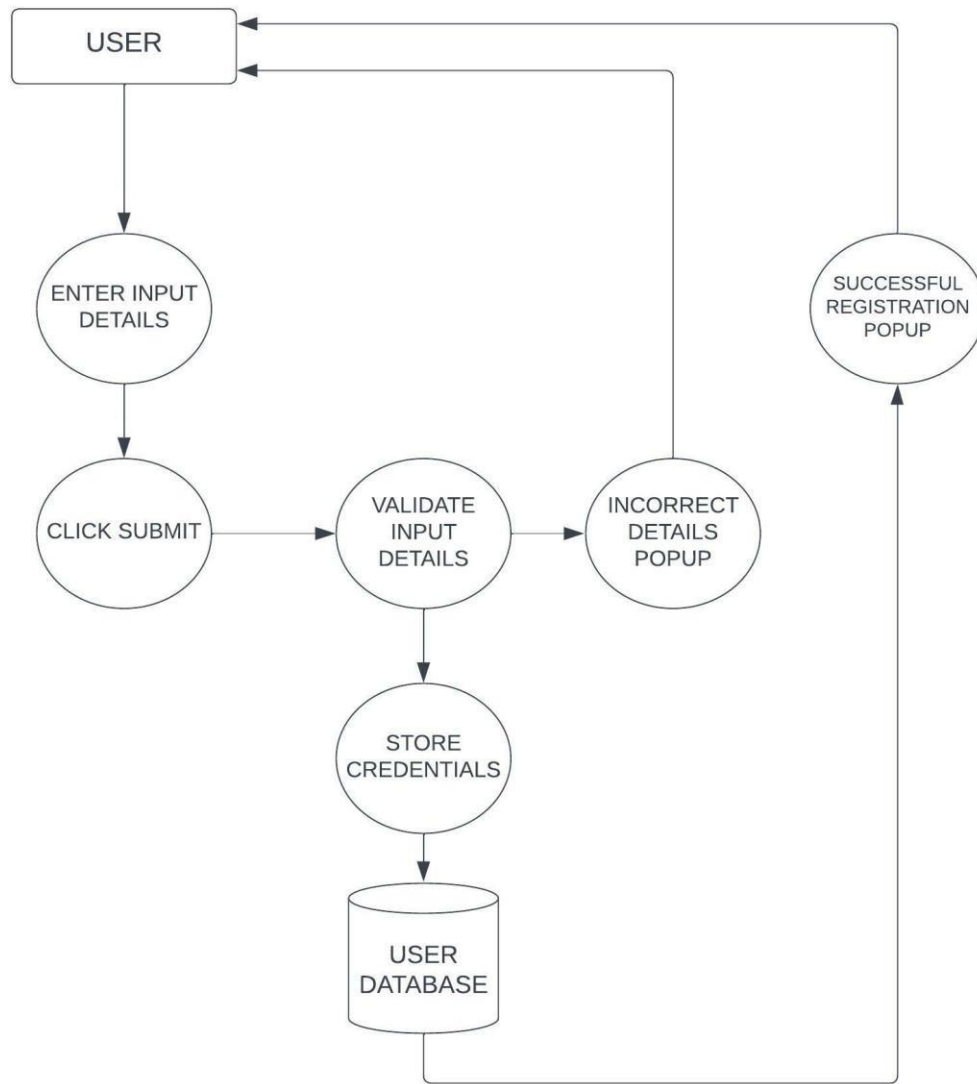


**Test Cases:**

Test ID	Test Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check if First Name and Last Name contain only alphabets	1. Enter Valid First and Last Name (alphabets) 2.Enter all required fields. 3.Click on Register Button	First Name = Deepak Last Name = Anand	Successful Registration	Expected	Pass
TU02	Check if First Name and Last Name contain only alphabets	1. Enter Invalid First and Last Name (numbers) 2.Enter all required fields. 3.Click on Register Button	First Name = Deepak Last Name = 7	Successful Registration	Unexpected	Fail
TU03	Check for Valid DOB (ie Day, Month, Year should be selected from Drop Down Menu)	1. Select Date, Month and Year 2.Enter all required fields. 3.Click on Register Button	Day = 07 Month = February Year = 2001	Successful Registration	Expected	Pass
TU04	Check for Valid DOB (ie Day, Month, Year should be selected from Drop Down Menu)	1. Select Date, Month and Don't select Year. 2.Enter all required fields. 3.Click on Register Button	Day = 07 Month = February Year =	Successful Registration	Unexpected	Fail
TU05	Verify if Valid Pincode is entered i.e has 6 digits	1. Enter Valid Pincode 2.Enter all required fields. 3.Click on Register Button	Pincode = 600028	Successful Registration	Expected	Pass
TU06	Verify if Valid Pincode is entered i.e has 6 digits	1. Enter Invalid Pincode 2.Enter all required fields. 3.Click on Register Button	Pincode = 60002832	Successful Registration	Unexpected	Fail

TU07	Verify if the length of the phone number is correct i.e. equal to 10.	1. Enter phone number less than 10 digits. 2.Enter all required fields. 3.Click on Register Button	Mobile Number = 8937592	Successful Registration	Unexpected	Fail
TU08	Verify if the length of the phone number is correct i.e. equal to 10.	1. Enter phone number more than 10 digits. 2.Enter all required fields. 3.Click on Register Button	Mobile Number = 893846723923	Successful Registration	Unexpected	Fail
TU09	Verify if the length of the phone number is correct i.e. equal to 10.	1. Enter phone number more than 10 digits. 2.Enter all required fields. 3.Click on Register Button	Mobile Number = 8073627839	Successful Registration	Expected	Pass
TU10	Check if any one radio button is selected for Courses applied	1. Select either BCA or B.Com or B.Sc or B.A. 2.Enter all required fields. 3.Click on Register Button	N/a	Successful Registration	Expected	Pass

**DFD:**



## Output:

# Register

**Username**

Username

**Surname**

Surname

**Given name**

Given name

**Email**

Email

**Mobile**

Mobile

**Phone**

Phone

**Password**

Password

Password

Register

**Result:** Thus the test cases for the login screen functionality has been executed successfully.

**EX NO: 06**

## **TESTING FOR STUDENT REGISTRATION FORM**

**DATE: 26/10/2022**

### **Aim:**

To write Test cases in Manual Testing for a Student Registration Form.

### **Problem Definition:**

Any Institution such as a college or a School may request the students to register themselves through an online registration portal for various purposes. Information about each student such as Name, Course, DOB need to be collected.

The Registration portal will allow students to provide their personal data and the personal data provided will be stored in a user database for future references. The students will have to provide the following data for successful registration -

- First Name
- Last Name
- DOB
- E-mail ID
- Mobile Number
- Gender
- Address
- City
- Pin code
- State
- Country
- Qualification
- Course

### **Software Requirement Specification:**

#### **Purpose:**

The purpose of this SRS is to describe the requirements involved in developing a Student Registration System.

#### **Scope:**

The network-based registration system will be used by students whom may be familiar or not to the online registration process thus the scope of the project must be user friendly for both students and admin. Once the registration process is completed for a student, the registered details will be stored in the User database.

## Definitions:

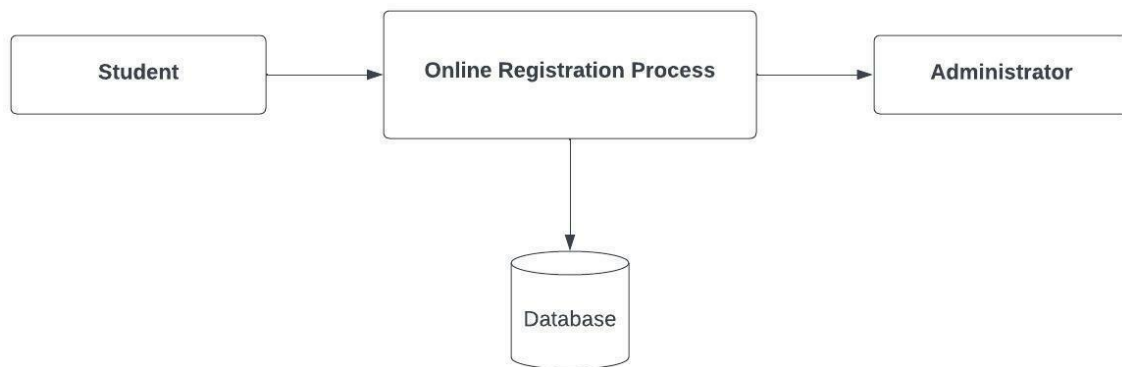
**Student:** Student is the user who accesses the registration form and provide their details. They are the ones who make use of the registration portal.

**System Administrator:** Students need to be admitted once they finish the registration process, this is done by the administrator. They are also responsible for changing and updating the registration form. They validate the registration process, add or delete fields and maintain the registration process.

## Product Perspective:

This subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software.

A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.



## Hardware interfaces:

- Hard disk
- Standard output device
- Keyboard and mouse

## Software interfaces:

- Back End: MS Access 2007
- Front End: Microsoft Visual Basic 6.0
- HTML & CSS
- Operating system: Any OS like Windows, Linux, MacOS

**Data Structures:**

Field Name	Type	Constraints
First Name	Text	Not Null
Last Name	Text	Not Null
DOB	Drop Down	Not Null
Email Id	Alphanumeric	Not Null
Mobile Number	Number	Not Null
Gender	Radio Button	Not Null
Address	Text	Not Null
City	Text	Not Null
Pincode	Number	Not Null
State	Text	Not Null
Country	Text	Not Null
Hobbies	Text	
Qualification	Text	Not Null

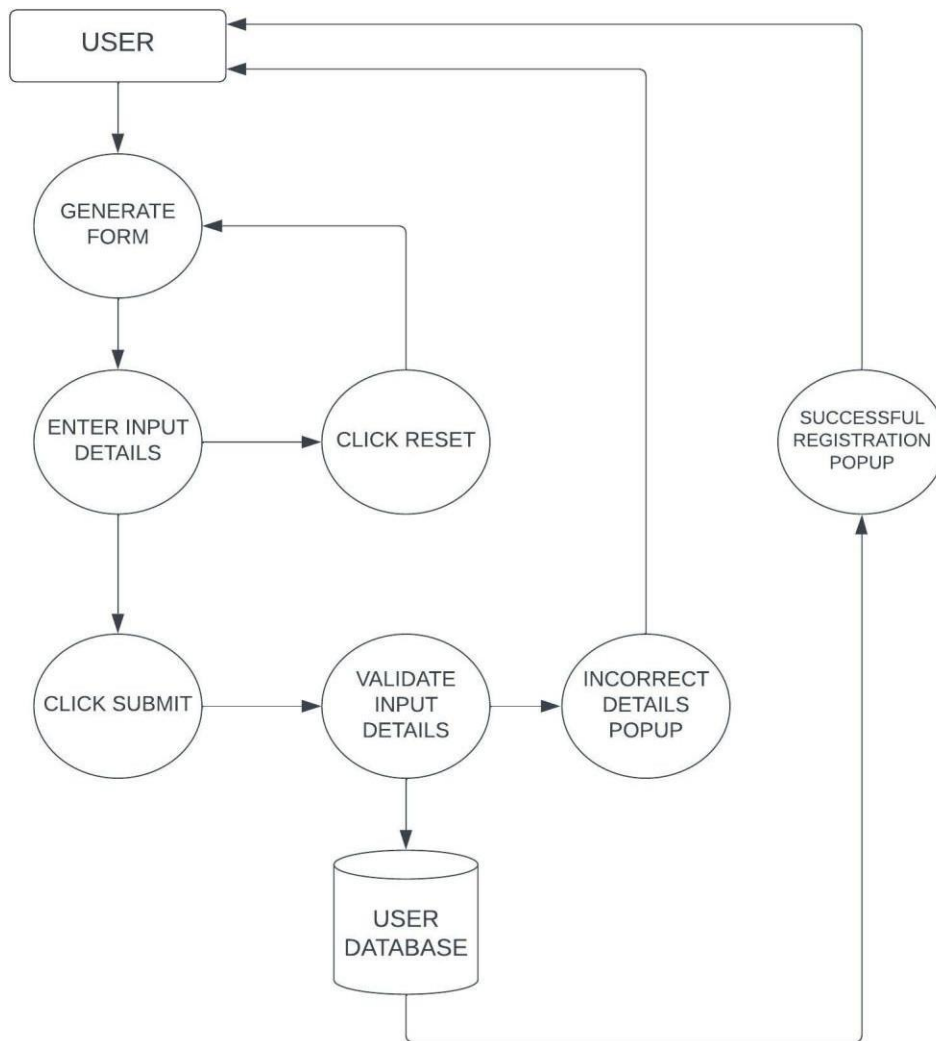


**Test Cases:**

Test ID	Test Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check if First Name and Last Name contain only alphabets	1. Enter Valid First and Last Name (alphabets) 2.Enter all required fields. 3.Click on Register Button	First Name = Deepak Last Name = Anand	Successful Registration	Expected	Pass
TU02	Check if First Name and Last Name contain only alphabets	1. Enter Invalid First and Last Name (numbers) 2.Enter all required fields. 3.Click on Register Button	First Name = Deepak Last Name = 7	Successful Registration	Unexpected	Fail
TU03	Check for Valid DOB (ie Day, Month, Year should be selected from Drop Down Menu)	1. Select Date, Month and Year 2.Enter all required fields. 3.Click on Register Button	Day = 07 Month = February Year = 2001	Successful Registration	Expected	Pass
TU04	Check for Valid DOB (ie Day, Month, Year should be selected from Drop Down Menu)	1. Select Date, Month and Don't select Year. 2.Enter all required fields. 3.Click on Register Button	Day = 07 Month = February Year =	Successful Registration	Unexpected	Fail
TU05	Verify if Valid Pincode is entered i.e has 6 digits	1. Enter Valid Pincode 2.Enter all required fields. 3.Click on Register Button 2.Enter all required fields. 3.Click on Register Button	Pincode = 600028	Successful Registration	Expected	Pass
TU06	Verify if Valid Pincode is entered i.e has 6 digits	1. Enter Invalid Pincode 2.Enter all required fields. 3.Click on Register Button 2.Enter all required fields. 3.Click on Register Button	Pincode = 60002832	Successful Registration	Unexpected	Fail

TU07	Verify if the length of the phone number is correct i.e. equal to 10.	1. Enter phone number less than 10 digits. 2.Enter all required fields. 3.Click on Register Button	Mobile Number = 8937592	Successful Registration	Unexpected	Fail
TU08	Verify if the length of the phone number is correct i.e. equal to 10.	1. Enter phone number more than 10 digits. 2.Enter all required fields. 3.Click on Register Button	Mobile Number = 893846723923	Successful Registration	Unexpected	Fail
TU09	Verify if the length of the phone number is correct i.e. equal to 10.	1. Enter phone number more than 10 digits. 2.Enter all required fields. 3.Click on Register Button	Mobile Number = 8073627839	Successful Registration	Expected	Pass
TU10	Check if any one radio button is selected for Courses applied	1. Select either BCA or B.Com or B.Sc or B.A. 2.Enter all required fields. 3.Click on Register Button	N/a	Successful Registration	Expected	Pass
TU11	Check the required fields	1. Fill all the required fields with valid data. 2. Click on Register Button	N/a	Successful Registration	Expected	Pass
TU12	Check the required fields	1. Fill some of the required fields with valid data and leave some out. 2. Click on Register Button	N/a	Successful Registration	Unexpected	Fail

## Data Flow Diagram:



Output:

STUDENT REGISTRATION FORM

FIRST NAME

(max 30 characters a-z and A-Z)

LAST NAME

(max 30 characters a-z and A-Z)

DATE OF BIRTH

Day:

Month:

Year:

EMAIL ID

MOBILE NUMBER

(10 digit number)

GENDER

Male

☐

Female

☐

ADDRESS

CITY

(max 30 characters a-z and A-Z)

PIN CODE

(6 digit number)

STATE

(max 30 characters a-z and A-Z)

COUNTRY

India

HOBBIES

Drawing

☐

Singing

☐

Dancing

☐

Sketching

☐

Others

☐

QUALIFICATION

Sl.No.	Examination	Board	Percentage	Year of Passing
1	Class X			
2	Class XII			
3	Graduation			
4	Masters			

(10 char max)

(upto 2 decimal)

COURSES APPLIED FOR

BCA

☐

B.Com

☐

B.Sc

☐

B.A

☐

Submit

Reset

**Result:** Thus, the test cases for the Student Registration Form have been executed successfully.

**EX NO: 07**

## **WHITE BOX TESTING**

**DATE: 04/11/2022**

**Aim:**

To perform white box testing on Palindromic Array.

**White Box Testing:**

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code.

**Working process of white box testing:**

- **Input:** Requirements, Functional specifications, design documents, source code.
- **Processing:** Performing risk analysis for guiding through the entire process.
- **Proper test planning:** Designing test cases so as to cover entire code. Execute and repeat until error-free software is reached. Also, the results are communicated.
- **Output:** Preparing final report of the entire testing process.

**White Box Testing Techniques:**

- **Statement Coverage** - This technique is aimed at exercising all programming statements with minimal tests.
- **Branch Coverage** - This technique is running a series of tests to ensure that all branches are tested at least once.
- **Path Coverage** - This technique corresponds to testing all possible paths which means that each statement and branch is covered.

**Code:**

```
import java.util.Arrays;
import java.util.Scanner;

public class Palindrome {
    public static void main(String args[]) {
        System.out.println("Enter");
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        char[] myArray = s.toCharArray();
        int size = myArray.length;
        char [] original = Arrays.copyOf(myArray,myArray.length);

        for (int i = 0; i < size / 2; i++) {
            char temp = myArray[i];
            myArray[i] = myArray[size-i-1];
            myArray[size-i-1] = temp;
        }
        if(Arrays.equals(myArray, original)) {
            System.out.println("1");
        } else {
            System.out.println("0");
        }
    }
}
```

**Output 1:**

Enter  
123111321  
1

**Output 2:**

Enter  
123455666  
0

**Result:**

Thus, white box testing on Palindromic Array is performed.



**Aim:**

To perform black box testing on Sum of Array.

**Black box testing:**

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications.

**Black Box Testing Techniques:**

- **Equivalence Class Testing:** It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.
- **Boundary Value Testing:** Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.
- **Decision Table Testing:** A decision table puts causes and their effects in a matrix. There is a unique combination in each column.

**Code:**

```
import java.util.*;
public class SumOfElementsOfAnArray {
    public static void main(String args[]){
        System.out.println("Enter the size of the array : ");
        Scanner s = new Scanner(System.in);
        int size = s.nextInt();
        int myArray[] = new int [size];
        int sum = 0;
        System.out.println("Enter the elements");
        for(int i=0; i<size; i++){
            myArray[i] = s.nextInt();
            sum = sum + myArray[i];
        }
        System.out.println("Sum =" +sum);
    }
}
```

**Output:**

Enter the size of the array : 2

Enter the elements

10 10

Sum =20

**Result:**

Thus, black box testing on Sum of Array is performed.