

6. SYSTEM IMPLEMENTATION AND TESTING

6.1 CODING STANDARDS

To make the system coding easy, easy to remember and reducing the chances of errors some techniques are used at the time of coding of the application which is called coding system. The coding system which we adopted during the coding is explained as follows:

1. File names follow Pascal Casing where First characters of each word are in upper case and other characters are in lower case.

Example: -

Airport.aspx, Price.aspx

2. Method names follow Camel Casing where First characters of all words, except the first word are in uppercase and other characters are in lowercase.

Example: -

```
private void SearchGoogleAPI ()  
{  
}
```

3. Variable names follow Camel Casing where First characters of all words, except the first word are in uppercase and other characters are in lowercase.

Example:-

```
int userCount = 0;  
int index = 0;  
string temp = "";
```

4. Naming Conventions

In short examples that do not include using directives, use namespace qualifications. If you know that a namespace is imported by default in a project, you do not have to fully qualify the names from that namespace. Qualified names can be broken after a dot (.) if they are too long for a single line, as shown in the following:

Example:-

```
var currentPerformanceCounterCategory = new System.Diagnostics.  
PerformanceCounterCategory();
```

5. Layout Conventions

Use the default Code Editor Settings (smart indenting, four-character indents, tabs saved as spaces). For more information, see Options, Text Editor, and Formatting.

Write only one statement per line.

Write only one declaration per line.

If continuation lines are not indented automatically, indent them one tab stop (four spaces).

Add at least one blank line between method definitions and property definitions.

Use parentheses to make clauses in an expression apparent, as shown in the following code.

Example:-

```
If ((val1 > val2) && (val1 > val3))  
{  
    // Take appropriate action.  
}
```

6. Commenting Conventions

Place the comment on a separate line, not at the end of a line of code.

Begin comment text with an uppercase letter.

End comment text with a period.

Insert one space between the comment delimiter (//) and the comment text, as shown in the following :

Example:

```
// the following declaration creates a query. It does not run  
// the query.
```

7. LINQ Queries

Use meaningful names for query variables. The following example Compare the Airport details from Database.

Example:

```
var Airpots = (from u in OdContext.Airports
               where u.AirportCode == txtSearchAirportCode.Text select u).FirstOrDefault();
```

6.2 TESTING METHODS

Software tested from two different perspectives:

1. Internal program logic.
2. Software requirements

Testing Strategies

Steps	Description
I	The determination of the functionality that the intended application is meant to perform.
II	The creation of test data based on the specifications of the application.
III	The output based on the test data and the specifications of the application.
IV	The writing of Test Scenarios and the execution of test cases.

6.3 TEST SUITS DESIGNS

- The testing technique that we are using in the project is black box testing.
- The expected inputs to the system are applied and only the outputs are checked.

6.4 TEST CASES

Test Suites No: 1

Test Suite Detail: Checking validation over register page.

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	Register	Register with already exist username and email	Register Unsuccessful	Register successful	Fail
2	Register	Register with already exist username and email	Register Unsuccessful	Register unsuccessful	Pass
3	Register	Require field follow validation	Register successful	Register successful	Pass
4	Register	Require field not follow validation	Register Unsuccessful	Register Unsuccessful	Pass
5	Register	Registration details should mail to user	Successfully mailed the details	Successfully mailed the details	Pass

Table 6.1: Checking validation over register page

Test Suites No: 2

Test Suite Detail: Inappropriate username and password in login model.

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	Login	Username Contains White Spaces	Login Successful	Login Successful	Pass
2	Login	Login with Incorrect Username and Password	Login Unsuccessful	Login Unsuccessful and give error	Pass
3	Login	Login with Correct Username and Password	Login Successful	Login Successful and set session variable	Pass

Table 6.2: Inappropriate username and password in login model.

Test Suites No: 3

Test Suite Detail: Add/Update/Delete – (Airport/Airlines/Train Station) Details

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	Details	Add/Update/Delete Details	Successfully effected in database	Successfully effected in database	Pass

Table 6.3: Add/Update/Delete Details

Test Suites No: 4**Test Suite Detail:** Add/Update/Delete (State/Country)

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	State	Add/Update/Delete State	Successfully effected in database	Successfully effected in database	Pass
2	Country	Add/Update/Delete Country	Successfully effected in database	Successfully effected in database	pass

Table 6.4: Add/Update/Delete (State/Country)**Test Suites No: 5****Test Suite Detail:** Changing Price

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	Price	Changing price	Successfully effected in XML file	Successfully effected in XML file	Pass

Table 6.5: Changing Price

Test Suites No: 6

Test Suite Detail: Checking User Management Module.

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	User Management	Create User	User successful	User successful	Pass
2	User Management	Roles/Module Assign	User unsuccessful	User unsuccessful	Fail
3	User Management	Roles/Module Assign	User Successful	User Successful	Pass

Table 6.6 Checking User Management Module

Test Suites No: 7

Test Suite Detail: Changing location

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	Address	Finding location to given Address	Successfully effected in database	Successfully effected in database	Pass

Table 6.7: Changing Location

Test Suites No: 8

Test Suite Detail: Configuration setting & Details settings

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	Configuration setting	Admin update own details	Successfully effected in database	Successfully effected in database	Pass
2	Details settings	Admin adjusting details	Successfully effected in database Successfully effected in database	Successfully effected in database	Pass

Table 6.8: Configuration setting & Details setting

Test Suites No: 9

Test Suite Detail: Manage CMS (Content Management System)

Test Case Id	Function Name	Test Case (Condition)	Expected Results	Actual Results	Pass/Fail
1	CMS content	Update CMS content	Successfully effected in database and view	Unsuccessful	Fail
2	CMS content	Update CMS content	Successfully effected in database and view	Successfully effected in database and view	Pass

Table 6.9 Manage CMS content