CHAPTER-7

TESTING

7.1 TESTING PLAN AND STRATEGY

7.2 TESTING METHODS

7.3 TEST CASES

7.1 TESTING PLAN AND STRATEGY

This state a number of rules that can server well as testing objectives:

- Testing is a process execution of a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet undiscovered error.
- A successful test is one that uncovers an as-yet undiscovered error. These Objectives imply a dramatic change in viewpoint for some hardware developers. They move counter to the commonly held view that a successful test is one in which no errors are found, Our objective is to design tests that systematically uncover different classes of errors and to do so with minimum amount of time & effort.

There are mainly five testing principles Which are described below:

principle #1: All tests should be traceable to custorner requirernents.

The objective of software testing is to uncover errors. It follows that the most server defects are those that cause the program to fail to meet its requirements.

principle #2:Test should be planned long before the actual testing begins.

Planning can begin as soon as the analysis model is complete. Detailed definition of test cases can begin as soon as the design model has been solidified. Therefore, all tests can be planned and designed before any code has been generated.

Principle #3: The PMeto applies to hardware testing.

Stated simply, the Pareto principle implies that go percent of all errors uncovered during testing will likew be traceable to 20 percent of all program components, The of course. is to isolate these suspect components to test than.

Principle #4: Testing should begin "In the small" and progress towards testing "in the large".

The first test planned and executed generally focuses individual components. As testing progresses. focus shifts in an attempt to find errors in Clusters Of Components and ultimately in the entire System.

Principle # 5: Exhaustive is not possible.

The number cf path permutations for even a mcH3erately sized program is exceptionally large, For this reason.it is impossible to execute every combination Of paths during testing. It is possible, however. to adequately cover program logic and to ensure that all connections in the component-level design have been exercised.

7.2 TESTING METHODS

Top down testing:

In top-down testing, testing starts With the most abstract Components and works downwards.

Back-to-Back testing:

It is versions of a system are available They are tested together and their outputs are Testing Can be to Check the performance of the product.

System Testing:

System Testing typically running every input to verify that it results in the right outputs using the software as an end-user would. It is needed to improve the performance of the system so Can have fast processing of all work.

Top down testing:

In top-down testing, testing starts with the most abstract components and marks downwards. Back-to-Back testing: It is used when different versions Of a system are available They are tested together and their outputs are compared. Testing can be done to check the performance of the product.

System Testing:

System Testing typically involves running through every possible input to verify that it results in the night outputs using the software as an end-user would, It is needed to improve the performance of the system so user can have fast processing of all work.

White-Box Testing:

Whitebox testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white box testing on an application, a tester needs to know the internal workings of the code. The tester needs to have a 100k inside the source code and find out Which unit/chunk of the code is behaving inappropriately.

Grey-Box Testing:

Grey box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application. Mastering the domain Of a System always gives the tester an edge over someone with limited domain knowledge, unlike black-box testing, where the tester only tests the application's user interface in grey-box testing the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

7.3 TEST CASES

Different types of Test Cases:

PURPOSE	REQUIRED INPUT	EXPECTED OUTPUT
COMPLAINT FORM	Complainer Name	If username exceeds more than 10 characters then the textbox will ignore that another letters. If username contains whitespace then that is also not allowed. Whitespaces are also ignored by textbox.
	Email	If email address does not contain "@", display "Invalid Email Address".
	Phone No.	Any other character rather than 0-9 digits or exceeds or less than 10 digits, "Invalid Phone number" error will generate
LOGIN	Valid Username and Valid Password	User enters both things valid, then, the user will be redirected to dashboard screen.
	Invalid Username or Invalid Password	If any of the field is incorrect, then error will be generated of the incorrect field. And, until the user inserts the correct information,

		he/she will not be re-directed to the dashboard screen.
Forgot or Reset Password	Email	If email address does not contain "@", display "Invalid Email Address" and if the filled E-mail address will receives the new password by dashboard.
	Get New Password	The password will be reset if the new password matches the conditions. And if it is same as the old password, then "Password exists" error will be occurred.

CHAPTER-8

LIMITATIONS AND ENHANCEMENT

❖ Limitations:

- No cash on delivery option is available.
- User cannot contact with cook.

❖ Future ehnancement:

- Makes this application totally “Dynamic”.
- Application will be available for all over the India.

CHAPTER-9

Conclusion

❖ Conclusion:

We conclude that till now the implementation we had done in our application Was the best of all things, because, We had some what kind of advanced. We also conclude that we have learnt so many new concepts of testing methods and strategies. We finally can conclude that according the contents of the above documentation, our Application has been built.

CHAPTER-10

References

❖ Websites:

- Stack Overflow - Where Developers Learn, Share, & Build Careers
- Bootstrap · The most popular HTML, CSS, and JS library in the world. (getbootstrap.com)
- GitHub: Where the world builds software · GitHub
- YouTube