Assignment No -5

Title: Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

Functional Testing Principles:

Testing:

Tests are the individual tests specified in a test plan document. Each test is typically described by:

- An initial system states.
- A set of actions to be performed.
- The expected results of the test.

Testing as a continuous process
All testing follows a pre-planned process, which is agreed to. All tests consider not only a nominal system condition but also address anomalous and recovery aspects of the system. The system is tested in a stressed environment, nominally in excess of 150 percent of its rated capacities. All test products (test cases, data, tools, configuration, and criteria) are documented in a software description document. Every test shall be described in traceable procedures and have pass-fail criteria included.

Implementation
Test cases are planned in accordance to the test process and documented with detailed test descriptions. These test cases use cases based on dissertational operational mission scenarios. The testing process also includes stress / load testing for stability purpose (i.e., at 95% CPU use, system stability is still guaranteed). The test process thoroughly tests the interfaces and modules. Software testing includes a traceable white box testing, black box testing and other test processes verifying implemented
Software against design documentation and requirements specified.

Types of testing

White Box Testing
A level of white box test coverage is specified that is appropriate for the software being tested. The white box and other testing uses automated tools to instrument the software to measure test coverage.

Black Box Testing
A black box test of integration builds includes functional, interface, error recovery, stress and out-of-bounds input testing. All black box software tests are traced to control requirements. In addition to static requirements, a black box of a fully integrated system against scenario sequences of events is designed to model operation. Performance testing for systems is integrated as an integral part of the black box test process.

Unit Testing

Unit testing is used to check the execution path of the module, function, and procedure of the system. Test is conducted with the help of normal data and abnormal data. This testing includes the different factors like statement coverage, branch coverage, loop processing, abnormality, and circulation etc. With the help of this Unit testing system check that all the statement in the code is executed or not so it avoids the dead code

statement. It checks all the branches and execution path of the code. It ensures that all the internal method of program are executed and properly integrated with program.

Integrated system

In integrated testing, all the modules are checked together to ensure that all the modules are executing together according to the program specification. Once all the modules have been tested individually, the most legitimate question can be asked is that when all the modules are working properly, why there is need of integrated testing. The answer is, though all modules are working properly problem may occur while interfacing individual module. Data can lose across an interface one module can have an adverse on another.

Testing tools and environment

- JUnit is a very good testing tool for using with unit testing. It is a java-based tool, which can go through code especially with extreme programming as it takes up less time to do.

- QTP: A testing tool for functional regression. It is an automated functional Graphic User Interface (GUI). The GUI is automated so actions can be made automatically online or offline. QTP has many functions such as letting you compare existing terns with current ones. QTP can test many applications such as Java, visual basic application, .net and many more.

- Coverage Tools: Code coverage, test case coverage, test coverage and so on Design: Caliber -RBT (No 1) uses the requirements as a basis to design the minimum number of test cases needed for full functional coverage.

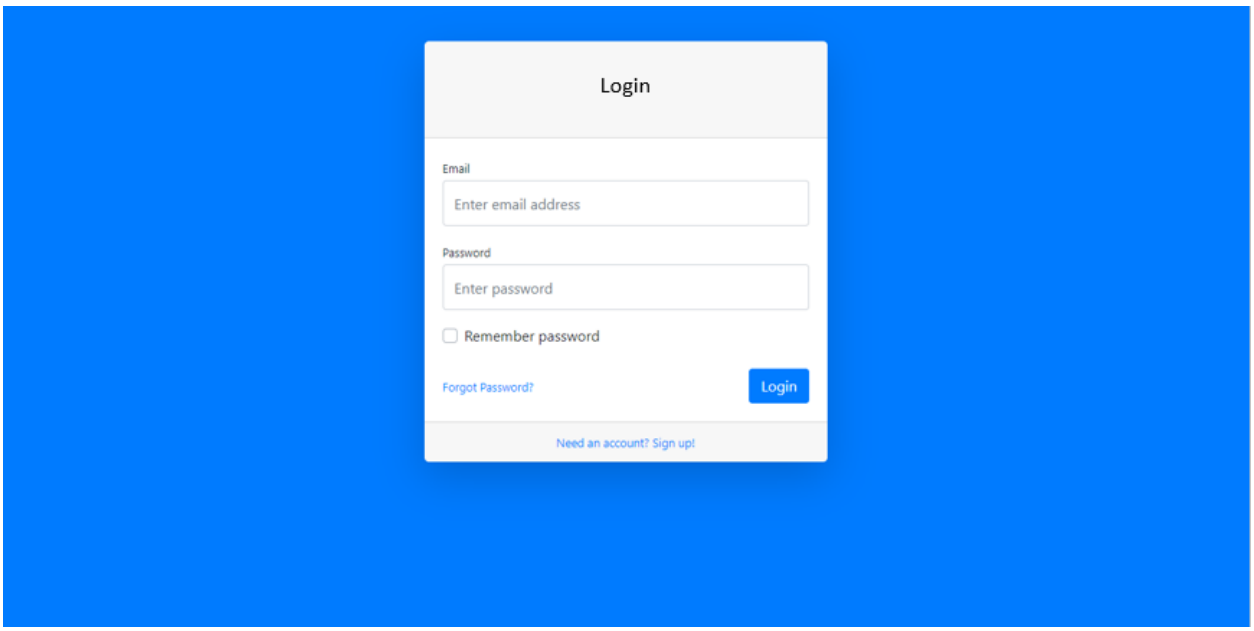
- GUI: Smalltalk Test Mentor (No 15) automatically gathers execution and method coverage metrics.

Test cases

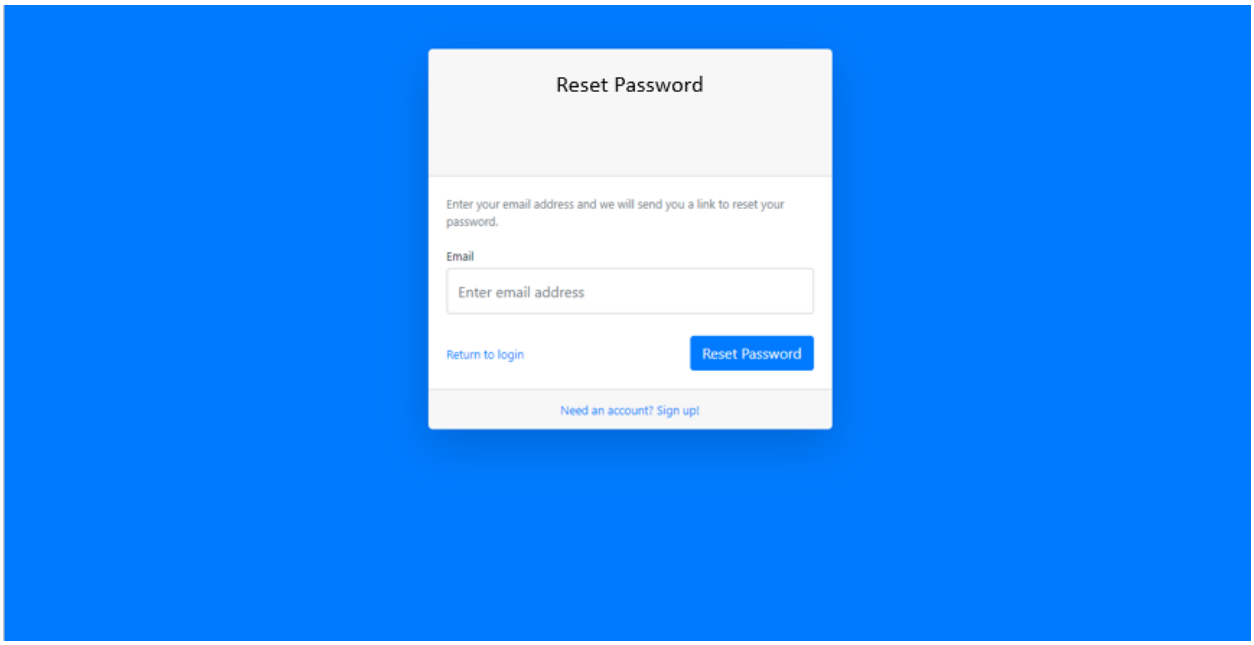
Test Case Id	Test Case	Objective	Expected Result
TC01	Login Screen- Sign up	Click on sign up button then check all required/ mandatory fields with leaving all fields blank	All required/ mandatory fields should display with symbol "*". Instruction line "*" field(s) are mandatory should be displayed
TC02	Create a Password, Confirm Password	Check the validation message for Password and Confirm Password field	Correct validation message should be displayed accordingly or "Password and confirm password should be same" in place of "Password mismatch".
TC03	Reset Password	Reset password using the email address.	Verification link should be sent on registered email address and user should able to reset password.
TC04	Store data on Cloud	Store Data on Cloud Database (Amazon DynamoDB)	The sensor data should be sent to cloud and store on Amazon DynamoDB database.
TC05	Display Data Mobile Application	Display Sensor data, Current market rates, weather condition on application	The data should be display on application in the form of charts, graphs, maps etc.

GUI:

1. **Login**



2. **Reset Password**



3. Signup

Conclusion:

Thus, we have tested our project module with different test principles. Also write different test cases of DFD and UML diagrams.