Test Summary Report

Contents

1. Purpose

This document explains the various activities performed as part of Testing of the 'Interactive Video Walls in NATS' application.

2. Application Overview

The 'Interactive Video Walls in NATS' is an application that students, faculty, administration, and any other visitors of the IU Southeast campus can use on a daily basis. People can walk up to the interactive video wall and click on professors to find out their office hours or just read about their recent research. The goal of our software for these boards is to simply provide a wide range of information to users. We want to answer common questions students have while providing them with helpful information along the way. It will be a great addition to the hallways as it will not only benefit students and other users, but also modernize the overall "look" of the two NATS entrances.

3. Testing Scope

a) In Scope

Primarily, the scope pertains to the interactive video walls that will go into making the two NATS buildings more modern. It focuses on IU Southeast, the students and the administration who will help us reach our product goals.

This scope is also aimed at specifying requirements of software to be developed but can also be applied to assist in the selection of in-house and commercial software products. The standards can be used to create software requirements specifications directly or can be used as a model for defining an organization or project specific standards.

b) Out of Scope

Functional testing was not done for this application as we don't have the hardware to run our software on yet.

c) Items not tested

The risk we have is that our open source software will not always be supported. A mitigation to this problem is to make sure we have at least one computer science sponsor on board as to what we did in our project software so if anything was to go wrong after we leave, there would be a way to solve any issues.

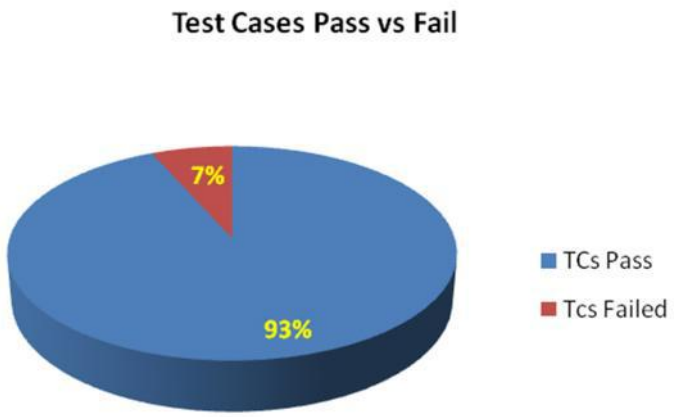
4. Metrics

<Metrics will help to understand the test execution results, status of test cases & defects etc. Required Metrics can be added as necessary.>

d) No. of test cases planned vs executed

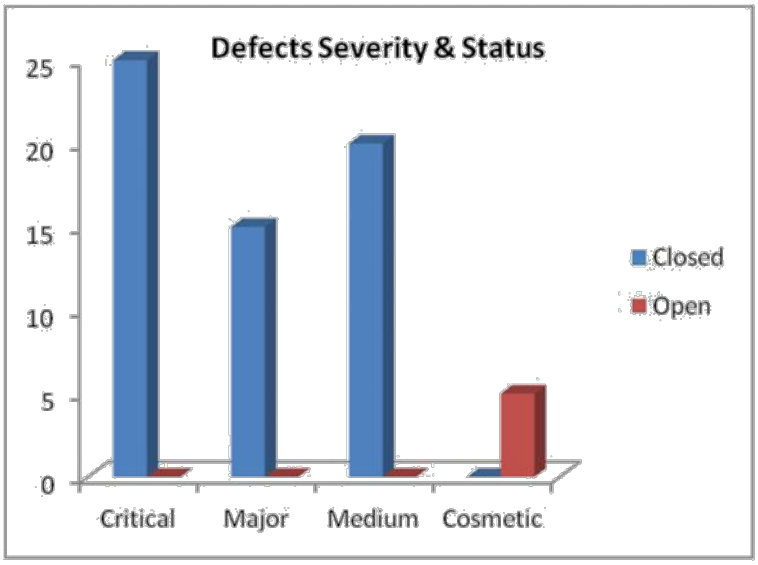
e) No. of test cases passed/failed

Test cases planned	Test cases executed	TCs Pass	Tcs Failed
80	75	70	5



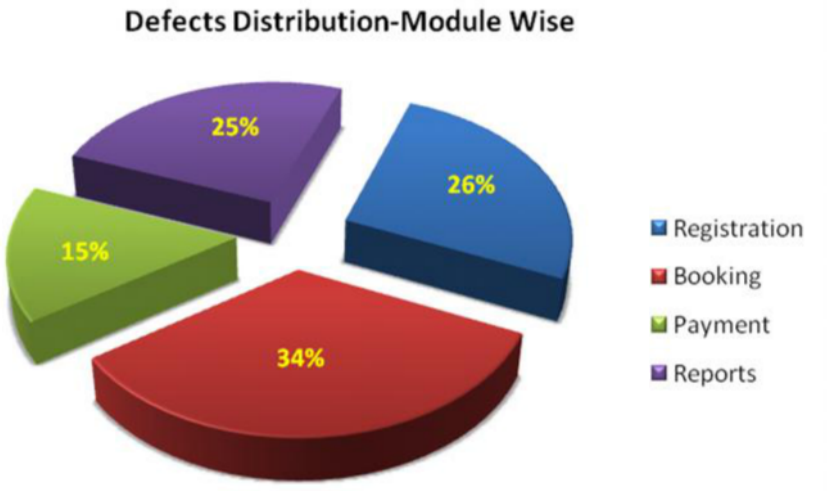
f) No of defects identified and their Status & Severity

	Critical	Major	Medium	Cosmetic	Total
Closed	25	15	20	0	60
Open	0	0	0	5	5
					65



g) Defects distribution – module wise

	Registration	Booking	Payment	Reports	Total
Critical	6	7	5	7	**25**
Major	4	5	2	4	**15**
Medium	6	8	2	4	**20**
Cosmetic	1	2	1	1	**5**
Total-->	**17**	**22**	**10**	**16**	**65**



5. Types of testing performed

For the testing of our software, we will be using functional and non-functional testing. Functional testing will include unit testing our web-based code, integration testing, system testing, sanity testing, interface testing, regression testing, and beta/acceptance testing. Non-functional testing will include performance testing, load testing, stress testing, volume testing, security testing, compatibility testing, install testing, recovery testing, reliability testing, usability testing, compliance testing, and localization testing. After using each of these types of testing, we will know the full functionality of our system and be able to analyze it to see if it is where it needs to be.

a) End-to-End

- This testing was done several times once we developed our software fully. We wanted to make sure it ran perfectly and one of the best ways to do this is with end-to-end testing.

c) System Integration Testing

- This is the Testing performed on the Application under test, to verify the entire application works as per the requirements.
- Critical Business scenarios were tested to make sure important functionalities in the application works as intended without any errors.

d) Regression Testing

- Regression testing was performed each time a new build is deployed for testing which contains defect fixes and new enhancements, if any.
- Regression Testing is being done on the entire application and not just the new functionalities and Defect fixes.
- This testing ensures that existing functionalities works fine after defect fix and new enhancements are added to the existing application.
- Test cases for new functionalities are added to the existing test cases and executed.

e) Performance Testing

- This testing was done to check whether the system meets the performance requirements.

6. Test Environment & Tools

For our test environment, key areas to set up include system and applications, test data, database server, front-end running environment, client operating system, browser, hardware, network cables, and documentation. We also need bug reporting tools in our testing environment so that the administration knows when our software or hardware isn't working properly.

The test environment management system includes maintenance of central repository with all the updated version of test environments. It is also important for us to have an ongoing test environment because these boards will be the new staple of the two NATS buildings. If they don't work properly and have a high, ongoing error rate, we risk a bad reputation as well as not living up to the dean's expectations. Our goal is to exceed all expectations, so testing for us is crucial to success.

7. Lessons Learnt

Lessons we learned from our project is the importance of communication as well as developing a budget early on. We struggled throughout the year with getting a budget approved which made it extremely hard for our project to take off.

8. Recommendations

<Any workaround or suggestions can be mentioned here.>

- A recommendation would be to get a budget approved before starting a project.
- Clearer communication would also be a good recommendation. Our group felt as if sometimes we were out of the loop as to what was going on. We would've liked to be clued in more and given the dean's timeline.

9. Best Practices

<There will be lot of activities done by the Testing team during the project. Some of them could have saved time, some proved to be a good & efficient way to work, etc. These can be documented as a 'Value Add' to show case to the Stakeholders.

- End-to-end testing is done each day by users of our system so once we deploy it, this type of testing will be ongoing.
- Automation scripts were prepared to create new customers, where lot of records need to be created for Testing.
- Business critical scenarios are separately tested on the entire application which are vital to certify they works fine.

10. Exit Criteria

<Exit Criteria is defined as a Completion of Testing by fulfilling certain conditions>

a) All test cases should be executed – **Yes**
b) All defects in Critical, Major, Medium severity should be verified and closed – **Yes**.
c) Any open defects in trivial severity – **Action plan prepared with expected dates of closure.**

No Severity1 defects should be 'OPEN'; Only 2 Severity2 defects should be 'OPEN'; Only 4 Severity3 defects should be 'OPEN'. Note: This may vary from project to project. Plan of Action for the Open defects should be clearly mentioned with details on when & how they will be addressed and closed.>

11. Conclusion/Sign Off

As far as the conclusion/sign off is concerned, I would say that we are not currently ready to sign off on our project because we don't have the hardware to run our software on yet. However, our software does work on a regular touch-compatible screen so hopefully when we transfer it to the large screens, it works on there as well. We will know more once the large touch screens are in and we do several tests on them to ensure the software is compatible.

