Test Plan Document

For

NATS Interactive Video Walls

1 Introduction 3

2 Business Background 3

3 Test Objectives 3

4 Scope 3

5 Test types Identified 3

6 Problems Perceived 3

7 Architecture 3

8 Environment 3

9 Assumptions 3

10 Functionality 3

11 Security 4

12 Performance 4

13 Usability 5

14 Test Team Organization 6

15 Schedule 6

16 Defects Classification Mechanism 6

17 Configuration Management 6

18 Release Criteria 6

Test Plan

# Introduction

The proposed system will be video walls to replace the old 70s looking publications displays in Life Science and Physical Science. The system will be a large touch screen or a combination of multiple screens that work together to allow students the ability to interact with the information that is displayed on the screens at the moment. For instance, there could be a rotating display of faculty and staff. If a student touches one of the faculty or staff members, their bio and a blurb regarding their recent research would appear on screen. Then the board could rotate to the disciplines and a student could touch one of those. Once a student touches a disciple, the screen(s) would show more information about that discipline, career opportunities, types of degrees offered, etc. If the system ends up being a video wall with multiple screens instead of just one huge screen, the project would be how to get these screens functioning properly together and how to display the information amongst multiple boards.

The ultimate goal of the proposed system is to modernize the NATS halls and bring them up to date with the 21st century. Our capstone team would need to creatively problem solve through cost estimating an entire package from hardware and possibly software. In addition, we would need to train the administration of this project and other faculty on how to use the system and update the software.

# Business Background

Our business background deals with the Dean of Natural Science and currently, there is not a system like this on IU Southeast’s campus. In addition to the Dean being in charge of our capstone group as the sponsor, we have been working with Dana (the Dean’s assistant) to figure out our budget and get money approved so that we can begin the process of installing and creating the software for the video boards. A business background may include company history, profiles of business principals, parent company, subsidiary, branch affiliations and operational highlights (including terms of sales, territory, company locations, number of employees and limited financial information).

# Test Objectives

For our test objectives, we are going to need to make sure the video boards work 24/7 and are able to have the touch screen interactivity the full time they are on. For our group, it is important that we ensure both hardware and software work as expected. It is important that they both work properly because they rely on each other. The hardware must be installed properly with the necessary storage needed to support our software we will be creating. Then, once we get the software created, we must make sure that is well tested so that we do not end up with performance issues or glitches. If for some reason our software wasn’t tested well and put through many test cases, we would risk the chance of the boards essentially being ineffective. The whole purpose of the boards is to simply run our software and if they do not work, these problems fall back on our capstone group to fix. Therefore, as long as we test the product thoroughly, I think we should be good to go until it is time to update.

# 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.

# Test types Identified

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 functionability of our system and be able to analyze it to see if it is where it needs to be.

# Problems Perceived

Some problems I perceive us having are touch-sensitivity issues because we are working with interaction. With hundreds of students touching the screen each day, it is inevitable that the screen will get dirty and this may hinder the touch-screen capability to detect when someone is interacting with the screen. In addition, since our system will be running 24/7, there may be times where the system starts to run slow or needs a reboot. Even though these are minor problems I expect to have, there should be a schedule created so that we have someone resetting the screen and wiping off the finger residue from the screen at least two times a week.

# Architecture

Introduction/Purpose of this Component/Entity: Architectural design is a concept that focuses on components or elements of a structure. An architect is generally the one in charge of the architectural design. They work with space and elements to create a coherent and functional structure.

Input for this Component/Entity: Our architectural design input will be very important as we will need to have a couple screens to create the video boards. Then we will take the screens and hang them in the NATS buildings to give us the architectural design we are looking for.

Output for this Component/Entity: The output for our architectural design will be a modern look composed of video boards with touch capability. The screens we use will also be composed of LED lights so it will be very vibrant. This will brighten up both the Physical Science and Natural Science halls and draw attention to students and visitors as they enter the doors.

Component/Entity Process to Convert Input to Output: The process to convert this input to output will require a lot of work. We will first need to get measurements of the walls and how large our video boards can be. Hanging the screens is where the challenge comes into play because we will need to provide the correct hardware to these systems. This means that we will need to thread both the Internet and power to the walls and locate the server in the ceiling to keep it hidden from users.

Design constraints and performance requirements of this Component/Entity: For our architecture design, we run into the problem of moving the existing glass case and the air conditioning/heating wires. Currently, there is a thermostat located on the wall where we need to put the screens. This means that we will have to move the thermostat and its wiring over which will require approval from IU Southeast administration.

# Environment

The environment that our system will be in is a college campus. Therefore, our system will be placed in the open for students, faculty, visitors, and administration to use as they please. There will be no log-in required as the main goal of our system is to provide users with information.

# Assumptions

Our assumption is that our system will light up the two NATS hallways and give them a much more modern vibe as people walk in. Right now, the hallways are dated which is not helping with recruitment of students. Once we get this new technology implemented, we will be able to not only show what senior computer science students are actively doing but have a new modern display in both buildings.

# Functionality

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Importance out of 10** |
| Constraint 1 | Organization | 10 |
| Constraint 2 | Method | 4 |
| Constraint 3 | Sustainability | 9 |
| Constraint 4 | Resources | 5 |
| Constraint 5 | Risk | 10 |
| Constraint 6 | Time | 9 |
| Constraint 7 | Cost | 7 |
| Constraint 8 | Scope | 6 |
| Constraint 9 | Quality | 7 |

***Risk Identified & Mitigation Planned***

***Test Strategy***

Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements. For our testing strategy, we will do each of the following types of testing to ensure our software meets our requirements: unit testing, smoke testing, sanity testing, integration testing, white box testing, black box testing, user acceptance testing, and regression testing.

***Automation Plans***

When it comes to the automation of our functionality, we will have the interactive screens on 24/7 so a software that runs without any breaks is what we will need. This can be automated by storing the data in a server so that we don’t overwhelm our system with data so quickly.

***Deliverables***

1. User should be able to touch the screen at any point and a reaction of some sort occur to show that the touch was registered
2. User should be able to go back to a previous page
3. User should be able to navigate easily through the UI
4. Video walls should be able to change the content or display with one single touch
5. Administrators should have the ability to preset the content that will be available to students
6. Administrators should be given the ability to restrict what can be done on the video wall
7. Students should have the ability to interact with the video wall at any time of day
8. Video walls should display important event information, local news, class information such as cancelled classes, and helpful student information
9. Students should have the ability to see advising information
10. Video wall should have many “spider web” effects as you click on a piece of information
11. Video wall should return to the main screen after a certain amount of time has past without interaction
12. Video wall should have the ability to display images of any type and possibly videos

# Security

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Importance out of 10** |
| Constraint 1 | Organization | 10 |
| Constraint 2 | Method | 4 |
| Constraint 3 | Sustainability | 9 |
| Constraint 4 | Resources | 5 |
| Constraint 5 | Risk | 10 |
| Constraint 6 | Time | 9 |
| Constraint 7 | Cost | 7 |
| Constraint 8 | Scope | 6 |
| Constraint 9 | Quality | 7 |

***Risk Identified & Mitigation Planned***

There are many security risks that come to mind when I think of our interactive video walls. For one, our physical security is not that secure due to the fact that the video boards will be out in the open for anyone to use and are available 24/7. However, we can mitigate the risk of our hardware getting taken overnight by installing cameras in the hallway facing the screens. Not only will this mitigate the risk of someone taking the screens but it will also show us if someone is messing with the software on our system and trying to hack into our database.

***Test Strategy***

Access control is a security technique that regulates who or what can view or use resources in a selected environment. In our case, any student, faculty, staff or visitor on the IU Southeast campus will be able to have access to our system. The physical access control limit is that it is located within the campus so unless you are on campus, you will not have access to the information displayed on the video board. This also relates to the logical access control limit of the computer networks, system files and data. The network will be secured and directed into the ceiling where the server will be located that holds our system flies. The system files will be located on a password protected server that will be stationed in a secure place so that unauthorized users cannot get ahold of it.

***Automation Plans***

The automation of our interactive video screens will be 24/7 as users are able to use the screens anytime throughout the day/night. This can cause a bit of a security issue if someone were to try and hack or steal our system. However, with the campus security being put in place in both NATS buildings, I do not see this being an issue.

***Deliverables***

When it comes to a deliverable for the security of our system, we will need to have cameras on our system so that if someone were to tamper with our system, we would know who and what exactly they did. In addition, we will also need to have software security put into place so that students can’t access the client side of our software and mess with the display information. This security can be created through a username and password database that only allows authorized administration to get into the system to update and add/remove information.

# Performance

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Importance out of 10** |
| Constraint 1 | Organization | 10 |
| Constraint 2 | Method | 4 |
| Constraint 3 | Sustainability | 9 |
| Constraint 4 | Resources | 5 |
| Constraint 5 | Risk | 10 |
| Constraint 6 | Time | 9 |
| Constraint 7 | Cost | 7 |
| Constraint 8 | Scope | 6 |
| Constraint 9 | Quality | 7 |

***Risk Identified & Mitigation Planned***

The performance risk is that the server gets overwhelmed and doesn’t perform at proper efficiency. This can be mitigated if you remove old data and make sure your server is in a cool place so that it doesn’t overheat and quit working.

***Test Strategy***

To test this, we will run routine server checks to ensure that the server is running up to peek efficiency.

***Automation Plans***

The testing strategy will be automated, and a report will be sent each week so that it is easy to check and will be pretty much hassle-free.

***Deliverables***

Performance is crucial to the success of our project because if our project does not perform up to the user’s expectations, we risk people not using our system. However, if our performance is very quick and exceeds user’s expectations, we are likely to increase interaction and retain users.

# Usability

***Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Importance out of 10** |
| Constraint 1 | Organization | 10 |
| Constraint 2 | Method | 4 |
| Constraint 3 | Sustainability | 9 |
| Constraint 4 | Resources | 5 |
| Constraint 5 | Risk | 10 |
| Constraint 6 | Time | 9 |
| Constraint 7 | Cost | 7 |
| Constraint 8 | Scope | 6 |
| Constraint 9 | Quality | 7 |

***Risk Identified & Mitigation Planned***

This may be one of the most important aspects of our project because it is what IU Southeast students, faculty, administration, and visitors will be interacting with. To put it simply, user interface design is important because it can make or break your customer base. It creates fewer problems, increases user involvement, and perfects functionality. To close, it also creates a strong link between your customers and your product. Our products user interface must be appealing and give off a “wow” factor that draws people’s attention as they walk into the buildings.

In addition, there is a screen layout user interface constraint and that is the size of the design cannot exceed the size of the screen. We also must have the user interface easy to use and navigation links that appear on every screen with a “help” button for first time users. The color scheme of the interface must portray the IU brand and the icons, buttons, labels, fonts, and images all must fit one standard. The UI must be uniform and clean with accommodations for those who are visually impaired.

***Test Strategy***

For the testing of usability, we will need to do non-functional testing. This will include performance testing, load testing, volume testing, stress testing, security testing, installation testing, penetration testing, compatibility testing, and migration testing.

***Automation Plans***

The communication between the different parts of the system is important since they all depend on one another. However, the way in which communication is achieved is not important for our system and is handled by the underlying operating systems of both the computer and display screens.

***Deliverables***

For the deliverables of usability, we will want a system that performs without lag and is the upper 90th percentile when it comes to performance.

***Compatibility Constraints and Resolutions***

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Customer Constraints** | **Importance out of 10** |
| Constraint 1 | Organization | 10 |
| Constraint 2 | Method | 4 |
| Constraint 3 | Sustainability | 9 |
| Constraint 4 | Resources | 5 |
| Constraint 5 | Risk | 10 |
| Constraint 6 | Time | 9 |
| Constraint 7 | Cost | 7 |
| Constraint 8 | Scope | 6 |
| Constraint 9 | Quality | 7 |

***Risk Identified & Mitigation Planned***

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.

***Test Strategy***

As said above, we will have training on our system as well as an information session for those who need to know the client-side of our software. By doing this in the testing strategy, we allow for software support if issues occur.

***Automation Plans***

If there are any problems in our software or hardware that is detected by our software running in the background, an alert will be sent to the administration of our software so they know what is going on and can work to fix the issues as soon as possible.

***Deliverables***

Open source is mainly protected by copyright law, and the license provides the basis for the rights you have on the software. To be classified as open source, any product needs to adhere to the four freedoms: to use, to study, to redistribute, and to improve. However, licenses often include other terms that might be incompatible with certain business models.

An open source software that we have considered is Wirewax. Wirewax is an interactive video platform, empowering users to add clickable hotspots, or “tags” to any moving person or object in the video. With up to 9x conversion in-video than on site, these tags encourage viewers to explore extra content, opportunities to buy or help create immersive experiences.

The terms of Wirewax are stated as “In the event that any licensed items are made available to the customer, Wirewax grants the customer non-exclusive, non-transferable, royalty free license to use the licensed items as follows: use of the APIs shall be limited to integrating or using the same with the customer system in order to activate the functionality of the platform within the customer system and thereby enabling the customer system to interact with the rest of the platform or the customer video, use of the off-platform functionality shall be limited to using within the deliverables and files as supplied by Wirewax and always only in conjunction with and to enable the playback of the customer video, and in the event any third party items are incorporated into or are used in conjunction with the platforms, or are used for the purposes of providing the services, then Wirewax grants (or shall procure the grant) to the customer a non-exclusive, non-transferable, royalty-free right to use such third party items for the purposes receiving the services.”

# Test Team Organization

Currently, Amanda has been working on each of the reports but as soon as we get our hardware ordered and in, Brandon and Will will begin working on the software for our system.

# Schedule

Implementation is the main priority on our schedule right now since we already have written our requirements. Once we get a budget approved and our hardware in, we will begin the implementation process. After finishing that, we will move to testing which will take about two months to ensure our system is working properly.

# Defects Classification Mechanism

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type of Defects | Functionality | Performance | Security | Usability | Compatibility |
| Critical | Works without errors | Want peek efficiency | Physical | Display | Software must run on hardware |
| Major | Software runs properly | Server getting bogged down | Software password protected | Calibration | Interactivity must work |
| Minor | Hardware is up to date | Small lag time with big tasks | Damage to screen | Sensitivity | Calibration must be set to work together |
| Cosmetics | Part of the screen stops working | Glitching of screen | Public display | Color Compliance | Database not accessible to public |

***Defects Logging and Status Changing Mechanism***

Within our system, we will have a performance checking system that tracks the defects logging and status changing of our system. This will help us easily tell when our system isn’t preforming up to standards.

***Turn Around Time for defect fixes***

A defect is measured on the basis of the turnaround time, that is, actual time taken to fix the defect divided by the planned time taken to fix the defect. When a defect occurs, the development team tries to analyze it's impact and decides on a target fix date, by when it shall be resolved. Our turnaround time unfortunately will be slower once we graduate because any software issues that the Dean runs into will have to be handled by UITS or the computer science department.

# Configuration Management

Configuration management (CM) is a system’s engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life. The way in which we will establish and maintain consistency is making sure each of the video boards function the same. The physical science building will have the same board as the Life Science building, but the contents will vary in terms of our software. For example, the software in Life Science will reflect the professors who teach in that building and the same for the professors in Physical Science. However, their displays will be consistent and the performance and functionality will be the same.

# Release Criteria

When releasing our final product, we will need to make sure our boards meet the resolution requirements, have the proper brightness settings for daytime and night time, fill the space requirements, fit the budget we have been given, works with multiple users and has been functionally and non-functionally tested, and works with the software we created.