Team 17 – NATS Interactive Video Walls

Amanda Goodridge, Brandon Baugh, Will Schottler

Last updated: October 24, 2019

Software Detailed Design

1. **Project Definition:**

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.

1. **Data Design**

Introduction/Purpose of this Component/Entity: The data design describes structures that reside within the software. Attributes and relationships between data object dictate the choice of data structures.

Input for this Component/Entity: This components input will be entered through an electronic database that will be located on Microsoft Excel. We will first ask the NATS staff and faculty what data they want displayed and then from there, record the data necessary to complete their request. Once we record the data, we will input it into the software to be displayed on the screen.

Output for this Component/Entity: The output for this entity will be the information stored in the database. However, in order to display this information, as we would like, we will have to style it with HTML and CSS software.

Component/Entity Process to Convert Input to Output: The process that we will use to convert this database information into the correct interface is through programming and CSS styling. If we decide to purchase software that is designed specifically for video boards, we will have to learn how to use this software and implement our information. We will also be responsible for training future users on the software so they can modify/update the boards over time. These users will most likely be the Dean of Natural Science and Dana Hope.

Design constraints and performance requirements of this Component/Entity: Some design constraints and performance requirements are the size and shape of the boards and the information we are allocated. We must make sure we have the proper amount of information available for each board and we have the interface and design laid out before we began programming.

Process (pseudo-code algorithm): The algorithm for which this process will use has not yet been decided. Once we figure out the layout and interface we want, we will be able to select a proper algorithm along with pseudo-code that best suits our design.

1. **Architecture Design**

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 the multiple screens to create our video board. Then we will take each of these screens and place them together to give us the architectural design we are going for.

Output for this Component/Entity: The output for our architectural design will be a modern look composed of multiple 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. Then, we will need to order the screens that make up each video wall and hang them. 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 existing hanging boards and 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.

1. **Interface Design**

Introduction/Purpose of this Component/Entity: The [*interface design*](https://en.wikipedia.org/wiki/Interface_design) describes internal and external program interfaces, as well as the design of human interface. Internal and external interface designs are based on the information obtained from the analysis model

Input for this Component/Entity: As said above, the interface for our project is going to be based on the information we provide in our database. Once we have the information, displaying this information will be the most important part of our project. It may also be the most challenging because we will have to take the raw data and make an appealing user interface. Not only will the user interface need to be attention-grabbing, but it will also need to have a good flow so that users are provided with a good user experience.

Output for this Component/Entity: The output for this project is a screen that has the user interface we specified based on our design requirements.

Component/Entity Process to Convert Input to Output: The entity process that converts the input to output is the software that we chose for our system. With this software, we will use it to place the information from the database onto the screen.

Design constraints and performance requirements of this Component/Entity: The constraints and performance requirements that are needed is a fast UI that functions with a very minimal error rate.

Process (pseudo-code algorithm):

The algorithm that will be used to get the input into output is one that we don’t have specified yet. Once we figure out what we want our output to look like, we will be able to purchase software that follows our requirements.

1. **Procedural Design**

Introduction/Purpose of this Component/Entity: The [*procedural design*](https://en.wikipedia.org/wiki/Procedural_design) describes structured programming concepts using graphical, tabular and textual notations. These design mediums enable the designer to represent procedural detail, that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering work.

Input for this Component/Entity: The input for procedural design is the structure that we create based on the flowchart of the process to complete the project.

Output for this Component/Entity: The output will be the finished flowchart and the diagrams we create for the multiple video screens that will work together.

Component/Entity Process to Convert Input to Output: The entity process we use to convert the input to output will be various software tools such as Visio. Visio gives us the ability to create high level diagrams and graphs that will best aid us in our design and implementation process.

Design constraints and performance requirements of this Component/Entity: The design constraints for procedural design is to ensure that our design works well with the multiple screens. The performance requirements are that the screen works well with little to no errors.

1. **Breakdown of Individual Contributions**

Our team is composed of three members by the names of Amanda Goodridge, Will Schottler and Brandon Baugh.

Amanda Goodridge is the team lead. She provides guidance, instruction, direction, and leadership to the group for the sole purpose of achieving a complete project by the end of the academic year. She monitors the quantitative and qualitative achievements of the team and reports these results to the project managers (Dr. Finkbine and the Dean). She often works within the team, as a member, carrying out the same roles but with additional ‘leader’ responsibilities such as added documentation and reports that are to be completed weekly.

Will Schottler is the technical lead. He is responsible for leading the development of the project and the quality of the technical deliverables. His goal is to establish a technical vision and then turn that vision into a reality. He is also responsible for doing research and preparing prototypes of proof of concepts along with ensuring proper security for the technical aspects of our system.

Brandon Baugh is the system architect. His role is to analyze and recommend the right combination of IT components to achieve the project’s goal. He is also to help define and decide on the right IT strategy and approach that will best support long-term business plans and goals. He advises Will with the best tools, frameworks, hardware, software, and other IT elements to achieve the functional objectives.

1. **Key Personnel Information**

Our team is composed of three members: Amanda Goodridge is the team lead, Will Shottler is the technical lead, and Brandon Baugh is the system architect.