**LA County, Hall of Administration Smart Board Directory**

**Version 1.0 approved**

**Prepared by Matthew Gerlits, Julius Macalutas, Christopher Ly, Jose Vega, Chelle Cruz, Cedric Tong**

**La County Hall of Administration**

**05/11/18**

# **Table of Contents**

Table of Contents................................................................................................................... 1

Revision History..................................................................................................................... 2

1. Introduction................................................................................................................ 3
   1. Sponsor........................................................................................................... 3
   2. Background.................................................................................................... 3
   3. Requirements................................................................................................. 3
   4. Design Principles ………………………...................................................... 4
   5. Current Solution ............................................................................................ 4
2. Related Works and Technologies............................................................................... 6
   1. Existing Works…........................................................................................... 6
   2. Techniques Used............................................................................................. 6
3. System Architecture…………................................................................................... 7
   1. DFD............................................................................................................... 7
   2. Workflow Between Components…............................................................... 7
   3. Software Development................................................................................... 8
   4. Software Implementation............................................................................... 8
4. Results and Conclusion….......................................................................................... 9
   1. Results…………………................................................................................ 9
   2. Future Work…………………....................................................................... 9
5. References………...................................................................................................... 10

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Technical Paper Final | 05/11/18 | Final Report | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **1. Introduction**

This document will introduce the sponsor, background, requirements, design principles, a description of the project, how it works, and non-functional requirements.

**Abstract**

The Hall of Administration Board of Supervisors wanted to provide a user-friendly interface and showcase of the various rooms and departments of the floor inside the building to people who visit. Amazon Alexa will provide a voice component and allow people to use and search the interface using their voice.

**1.1 Sponsor**

LA County Hall of Administration Board of Supervisors

**1.2 Background**

The LA Hall of Administration Board of Supervisors is in the process of updating their building directories, by replacing their current physical directories, with a digital directory system. In order to help facilitate this change, we have been tasked with creating an interface for the digital directory system.

**1.3 Requirements**

The contents of the software must exist within a web application. This means that any piece of software that lives on the application will need to adhere to the constraints web development. One example of this is the lack of regular Socket usage. Unity WebGL specifically has a number of limitations, the most notable being that the WebGL application exist within a different scope than the rest of the webpage. The software must also be able to handle being within a noisy environment, as the application will live on device in a public area. The UI must be able to be used for a touch interface, so the size of elements, such as buttons, will need to be designed in way to increase usability. In order to ensure that all of the Smart Board Directories will share the same information, each instance of the application must be able to connect to a central database.

Another requirement was for the Amazon Alexa Skill to fulfill the requests of

**1.4 Design Principles**

No formal development method was used. Each component of the project was divided into different teams within the group, and the project was tied together by an additional team at the end. This way, we had a high level of focus on each component of the project.

# **1.5 Current Solution**

# 

The interactive map includes the first four floors and the basement. Rooms can be currently selected by tapping on the section, which will display a sidebar consisting of the room name, description, and image relating to the room. Dijkstra’s algorithm is used to find the optimal path to a designated room. Floors can easily be navigated and switched based on buttons integrated into the web application interface. A new path is animated when a user selects a room. If the directions involve going to the elevator into another floor, the map will automatically switch floors to show the directions involving both floors (ex. Starting at first floor and going to room 493 in fourth floor).

The web application and map work coherently by providing UI elements in the web interface, which will be used to communicate with the map (exported as a WebGL build) through Unity browser scripting. The UI elements consist of a dropdown of rooms to select and find directions, categorized by floors, a section for outputting directions, which include corresponding icons for each direction (ex. Left, right, destination icons), a search bar for filtering rooms based on keywords in order to quickly find a room instead of scrolling through the dropdown, and speech output when interacting with Alexa.

For voice recognition, a custom Alexa skill is developed comprising of intents corresponding to different services users can inquire (ex. Department, service within department, employee, lunch menu). Working with AWS, lambda functions are created for handling the logic of user voice input and gathering current speech output stored in the DynamoDB cloud database, which is a part of AWS. Information on departments, employees, services, and lunch menu items are also stored as tables in DynamoDB.

For the web portal, user authentication consists of administrative access, department access, and regular user access. Databases of employees and departments are included which can be easily edited through the UI table. Powered through SQL Server, data can be manipulated through CRUD operations. The web portal is implemented using ASP.NET Core 2.0.

Challenges with existing solution is finding an efficient way of gathering speech output from Alexa in real time. Currently, speech output is gathered as a JSON response from an API call using HTTP GET requests. In order to simulate real-time responses, these HTTP GET requests are called every second. The API handles getting the current speech output from the DynamoDB table, which is updated every time Alexa gives a response in the Hall Directory skill. Plans for improving this method include using web socket technology such as AWS Appsync, which creates real-time interaction based on event changes and multi-user interaction.

Challenges with the web portal include deployment to a server since ASP.NET Core 2.0 is a new technology. We are currently working with HoA administrators to find a solution to deploy the web application to their server.

# 

# **2. Related Works and Technologies**

**2.1 Existing Works**

For this project, we got inspiration from a number of existing works. The most notable include digital directories at malls, government websites, and common UI layouts for digital kiosks.

The web application was integrated using Vue.js, a front-end Javascript framework for creating scalable and interactive user interfaces through real-time state management of data and variables and binding of DOM elements during rendering. Different components are updated and re-rendered based on state changes in the application. Other technologies for creating a enhanced experience are CSS Grid, a built-in CSS tool to create 2d grid-layouts, and Greensock Animation Platform (GSAP), a Javascript tool for creating HTML5 animations.

With the Amazon Alexa Skill, there were many sample applications listed on Amazon’s Alexa repository on their GitHub. We used this as inspiration to build and shape our Alexa Skill to meet the requirements given by the Hall of Administration.

**2.2 Techniques Used**

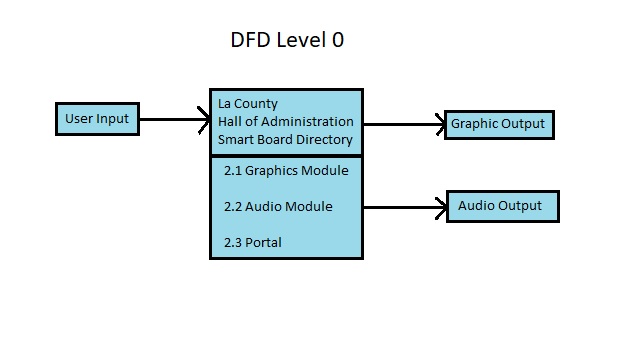
The most notable technique we used was the implementation of Dijkstra's Algorithm in order to set up a path finding solution.

For the Amazon Alexa, the most notable technique we used was fuzzy finding, which is approximate string matching, in order to ask the Alexa Skill any question. Then the Alexa Skill will search the database for that question, approximately matching what the user said. We used a Python library to do this.

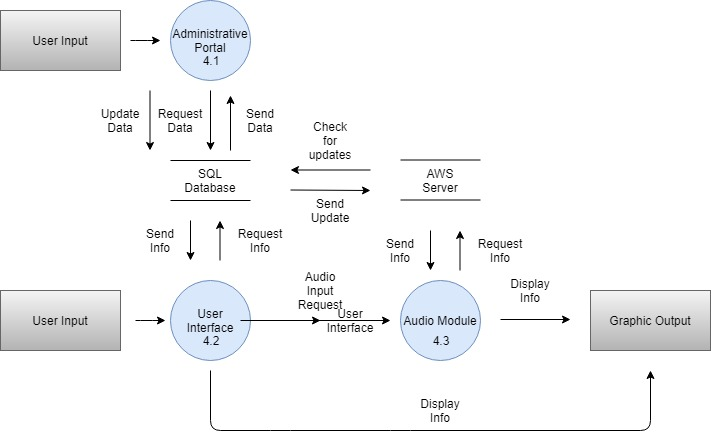
# **3. System Architecture**

# 

# **3.1 DFD**



# **3.2 Workflow Between Components**



# 

# **3.3 Software Development**

In order to get the path finding portion of this software working, we had to implement several components. We had to implement a Node structure, Dijkstra's Algorithm, a priority queue, and Edges.

# 

# **3.4 Software Implementation**

In order to implement Dijkstra’s Algorithm, we had to create a priority queue, a Node structure, and Edges to pair with the nodes. C# does not have a native priority node object, so in order implement one we had to create a tuple, with its comparison operators overloaded. Next, we created Node objects which contain information such as a GameObject, Edge connections, and a list of neighbors. The Edges contained a start and end GameObject, and a weight.

# 

# **4. Results and Conclusions**

# 

# **4.1 Results**

The county liked our project, Keenan was happy. All is well. We got Pamula a contract with LA County. Also had the best poster at the Expo.

# 

# **4.2 Future Work**

For future work, we would like to use Amazon Web Service’s AppSync in order to receive the speech outputs in real-time. Currently, we are using http requests to gather the speech outputs on a request by request basis.

We would also like to have a more refined output on the speech, because as of right now, it is a giant block of text.

Additionally, we want a more robust interaction with the pathfinding and the map.

# 

# **5. References**

<https://docs.unity3d.com/ScriptReference/>

<https://aws.amazon.com/documentation/>

<https://developer.amazon.com/docs/ask-overviews/build-skills-with-the-alexa-skills-kit.html>

<https://vuejs.org/v2/api/>

<https://greensock.com/docs>

<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout>