**Software Design**

**Document**

**for**

**LA County Board of Supervisors Smart Board Directory**

**Version 1.0 approved**

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

**LA County**

**4/9/2018**

Table of Contents................................................................................................................... <pg 2>

Revision History..................................................................................................................... <pg 4>

1. Introduction................................................................................................................ <pg 5>
   1. Purpose........................................................................................................... <pg 5>
   2. Document Conventions……………………................................................. <pg 5>
   3. Intended Audience and Reading Suggestions................................................ <pg 5>
   4. System Overview........................................................................................... <pg 5>
2. Design Considerations............................................................................................... <pg 6>
   1. Assumptions and dependencies..................................................................... <pg 6>
   2. General Constraints........................................................................................<pg 6>
   3. Goals and Guidelines.....................................................................................<pg 6>
   4. Development Methods................................................................................... <pg 6>
3. Architectural Strategies.............................................................................................. <pg 7>
4. System Architecture................................................................................................... <pg 8>
   1. ....................................................................................................................... <pg 8>
   2. ....................................................................................................................... <pg 9>
5. Policies and Tactics.................................................................................................,<pg 10>
   1. Specific Products Used.................................................................................<pg 10>
   2. Requirements traceability.............................................................................<pg 10>
   3. Testing the software......................................................................................<pg 10>
   4. Detailed System Design...............................................................................<pg 10>
6. Detailed System Design

6.1 Name of Module..........................................................................................<pg 11>

6.1.1 Responsibilities....................................................................................<pg 11>

6.1.2 Constraints............................................................................................<pg 11>

6.1.3 Composition.........................................................................................<pg 11>

6.1.4 Uses/Interactions..................................................................................<pg 11>

6.1.5 Resources..............................................................................................<pg 11>

6.1.6 Interface/Exports..................................................................................<pg 11>

6.2 User Interface..............................................................................................<pg 12>

6.2.1 Responsibilities....................................................................................<pg 12>

6.2.2 Constraints.......................................................................................... <pg 12>

6.2.3 Composition........................................................................................ <pg 12>

6.2.4 Uses/Interactions................................................................................. <pg 12>

6.2.5 Resources............................................................................................ <pg 13>

6.2.6 Interface/Exports................................................................................. <pg 13>

6.3 Audio Module.............................................................................................<pg 13>

6.3.1 Responsibilities....................................................................................<pg 13>

6.3.2 Constraints............................................................................................<pg 13>

6.3.3 Composition.........................................................................................<pg 13>

6.3.4 Uses/Interactions..................................................................................<pg 13>

6.3.5 Resources............................................................................................. <pg 13>

6.3.6 Interface/Exports..................................................................................<pg 13>

1. Database Design…………………………………………………………………...<pg 14>
2. User Interface.......................................................................................................... <pg 15>

8.1 Overview of User Interface…………………………………………………....<pg 15>

8.2 Screen Frameworks or Images……………………………………………..….<pg 15>

8.3 User Interface Flow Mode…………………………………………………..…<pg 17>

1. References................................................................................................................<pg 20>

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| First Draft | 04/14/18 | Needed to create the document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

This document list the functions and design details for the Smart Board Directory Project Ver 1.0. This project contains components consisting of ASP.NET, Unity, Amazon Alexa, and AWS.

**1.2 Document Conventions**

None

**1.3 Intended Audience and Reading Suggestions**

The intended audience of this document are software developers, and educational staff at CSULA. This document contains the technical overview of what the project pertains to.

**1.4 System Overview**

The system consist of a web application, which contains Unity WebGL, Amazon Alexa, ASP.NET, and a Microsoft SQL connection. There is an additional, separate component, which acts as a administrative portal. The web application will display an interactive map of the building, which will allow users to get directions to their desired destination within the building. Other basic information, such as a daily of what is being served at the cafeteria, or events that will occur on each day will also be displayed. The administrative portal will allow staff to update information regarding the employees and departments of the building.

**2. Design Considerations**

Design considerations that should be addressed are providing a database management system, Amazon Alexa Skills for the directory, a web application for the directory, and a path-finding algorithm within said web application.

**2.1 Assumptions and Dependencies**

Assumptions or Dependencies regarding the software and its use:

* Unity
* ASP.NET
* Amazon Alexa Skill
* Windows 10

**2.2 General Constraints**

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.

**2.3 Goals and Guidelines**

The software must be completed by the week of 4/16/18. The software also must be intuitive for the general population, as it will be used by the constituents of LA County.

**2.4 Development Methods**

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.

**3. Architectural Strategies**

For this project, we were required to use ASP.NET Core 2. This was a requirement put into place by the LA County. C# became a primary programming language since ASP.NET uses C#. Unity was chosen as the platform to host the interactive map due to its ease of use, and compatibility with C#. Python was a language was added later on in the development in order to allow for Amazon Alexa integration. The choice of this programming language was due to the fact it is the recommended programming language for writing programs for the Alexa Skills kit. For the Amazon Skills, AWS was chosen as the database since it has seamless connection with the skills. Microsoft SQL was the chosen database for the ASP.NET portions of the application, because these two products were designed to be used together.

**4. System Architecture**

The **HASBD** architecture is summarized in the Context Diagram (DFD Level 0) given below. A more complete Functional Description is given in Section 2 of this document. The Context Diagram provides the overall structure of the software modules and all its inputs and outputs. The notation used corresponds to that defined for any Data Flow Diagram (DFD).

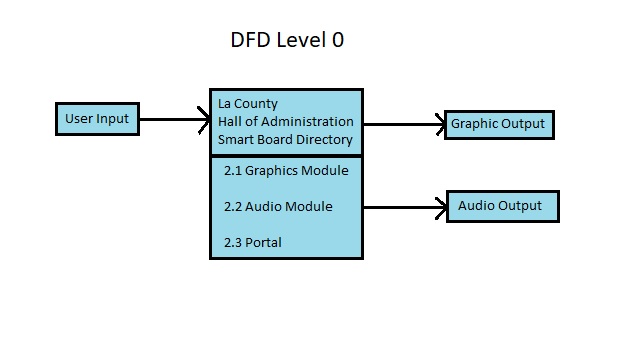


Figure 4 - 1 : DFD Level 0

The software must accept user input in order for it to work. Then it will take that input, whether it may be voice or text, in order to translate it into a set of instructions. The graphics module will provide a user interface for the user and the audio will be a response regarding the user’s input. The portal is an administrative website set up for the County for their personal use. In return, the user’s input will generate two outputs, graphically and auditory.

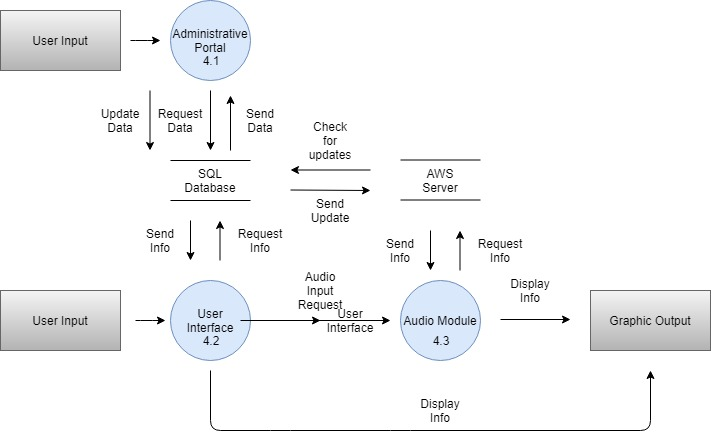


Figure 4 - 2 : DFD Level 1

The user input will go through an user interface (4.2), which will then send the input into two different places. The first being the Audio Module (4.3), which is a function that talks to the AWS Servers. It will return requested data as specified from the user input and the Lambda Function from DynamoDB. The data returned will then be displayed on the graphic output.

The second is SQL Database that sends information to the Administrative Portal (4.1) in order to display the most up to date information. With this, the portal will constantly check for updates made into the SQL Database. The SQL Database also sends information to the AWS Server with information in order to have the most up to date data.

**5. Policies and Tactics**

**5.1 Choice of which specific products used**

Microsoft Visual Studio 2017

Microsoft SQL

DynamoDB

Unity3d

**5.2 Plans for ensuring requirements traceability**

In order to ensure design traceability, we are designing the user interface to be intuitive. This way constituents will be able to easily utilize the software

**5.3 Plans for testing the software**

In order to test our software, we used the following:

* Google Chrome Dev Tools
* AWS CloudWatch Log

**6. Detailed System Design**

**6.1 Administrative Portal**

**6.1.1 Responsibilities**

The primary responsibilities and/or behavior of this component. What does this component accomplish? What roles does it play? What kinds of services does it provide to its clients? For some components, this may need to refer back to the requirements specification.

**6.1.2 Constraints**

Any relevant assumptions, limitations, or constraints for this component. This should include constraints on timing, storage, or component state, and might include rules for interacting with this component (encompassing preconditions, post conditions, invariants, other constraints on input or output values and local or global values, data formats and data access, synchronization, exceptions, etc.)

**6.1.3 Composition**

A description of the use and meaning of the subcomponents that are a part of this component.

**6.1.4 Uses/Interactions**

A description of this components collaborations with other components. What other components is this entity used by? What other components does this entity use (this would include any side-effects this entity might have on other parts of the system)? This concerns the method of interaction as well as the interaction itself. Object-oriented designs should include a description of any known or anticipated subclasses, superclass’s, and metaclasses.

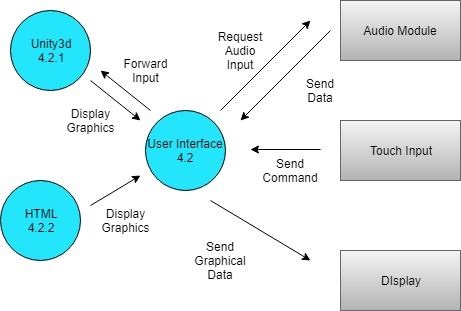
**6.1.5 Resources**

A description of any and all resources that are managed, affected, or needed by this entity. Resources are entities external to the design such as memory, processors, printers, databases, or a software library. This should include a discussion of any possible race conditions and/or deadlock situations, and how they might be resolved.

**6.1.6 Interface/Exports**

The set of services (classes, resources, data, types, constants, subroutines, and exceptions) that are provided by this component. The precise definition or declaration of each such element should be present, along with comments or annotations describing the meanings of values, parameters, etc. For each service element described, include (or provide a reference) in its discussion a description of its important software component attributes (Classification, Definition, Responsibilities, Constraints, Composition, Uses, Resources, Processing, and Interface).

**6.2 User Interface**



**6.2.1 Responsibilities**

The primary responsibilities of this user interface are to display information provided by the rest of this application. This information includes data such as the path to a location, or the output of the audio module. All of the user interaction will take place here.

**6.2.2 Constraints**

Users will be limited in the data they can access. They will be able to search for locations within the building, by only on specified floors and locations. The User Interface was created as a web application, so HTML must be used, and all constraints relating to WebGL must be followed.

**6.2.3 Composition**

Unity3d is used to render and perform the logic of a 3d map of several floors of the building.

**6.2.4 Uses/Interactions**

The audio module acts as a secondary input method for this module.

**6.2.5 Resources**

This module pulls data from an SQL database

**6.2.6 Interface/Exports**

This module sends a request to the Audio module to start recording for an audio query. This module sends graphical data to the Graphical display

**6.3 Audio Module**

**6.2.1 Responsibilities**

The primary responsibilities of this audio module is to have the user speak into the device and what the user says will go through AWS. AWS will handle the logic through Lambda Functions and the desired output will be displayed the graphic output

**6.2.2 Constraints**

Users will be limited in the data they can access. They will be able to search for locations within the building, by only on specified floors and locations. The auditory inputs are limited to what the Lambda Function and AWS Interaction Model are able to output.

**6.2.3 Composition**

Lambda Functions and AWS are responsible for generating an output.

**6.2.4 Uses/Interactions**

The audio input is the only input for this module.

**6.2.5 Resources**

This module pulls data from an noSQL database, DynamoDB

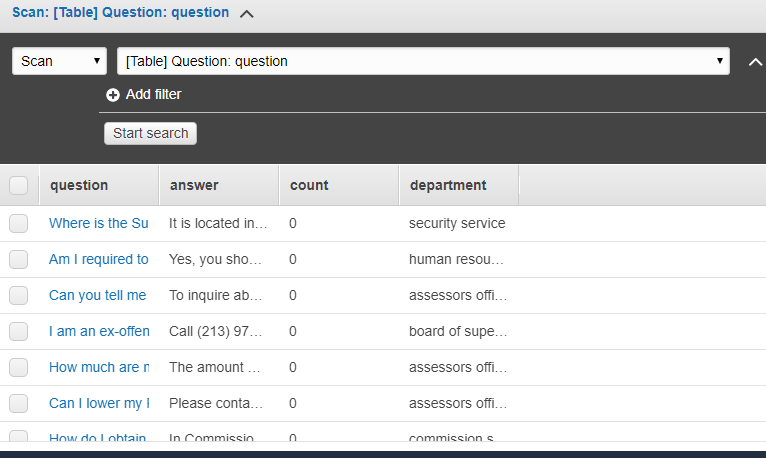
**6.2.6 Interface/Exports**

This module sends a request to AWS, which returns the information requested to be displayed onto the graphical output.

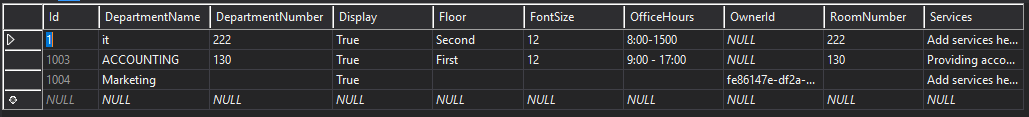
**7. Database Design**

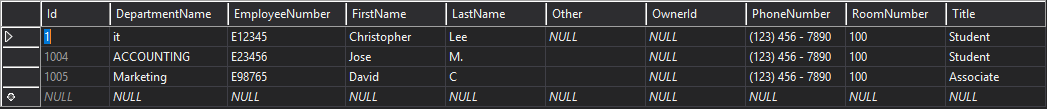
DynamoDB and Microsoft SQL Server 2016 are being used.

DynamoDB is used by the Amazon Alexa Skill. It takes the department information and displays them onto the Skill. It also has the frequently asked questions stored and allows the Skill to pull data from it to display onto the board. There are also tables for the lunch menu and employees. In the future, the employees information will be stored on the Microsoft SQL Server.



Microsoft SQL Server 2016 is used in the database management system application to add, update, and delete information department information and employee information.



****

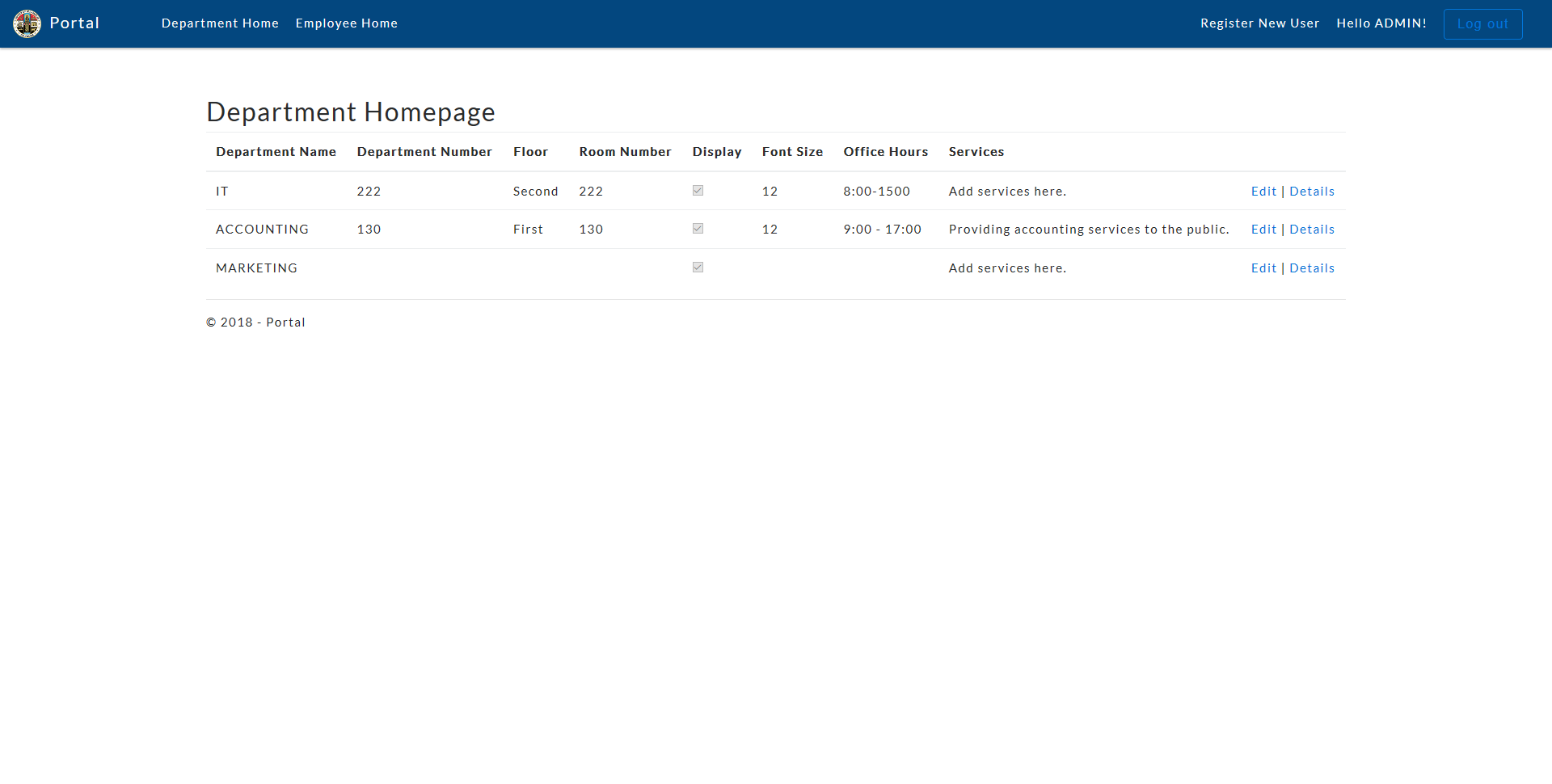
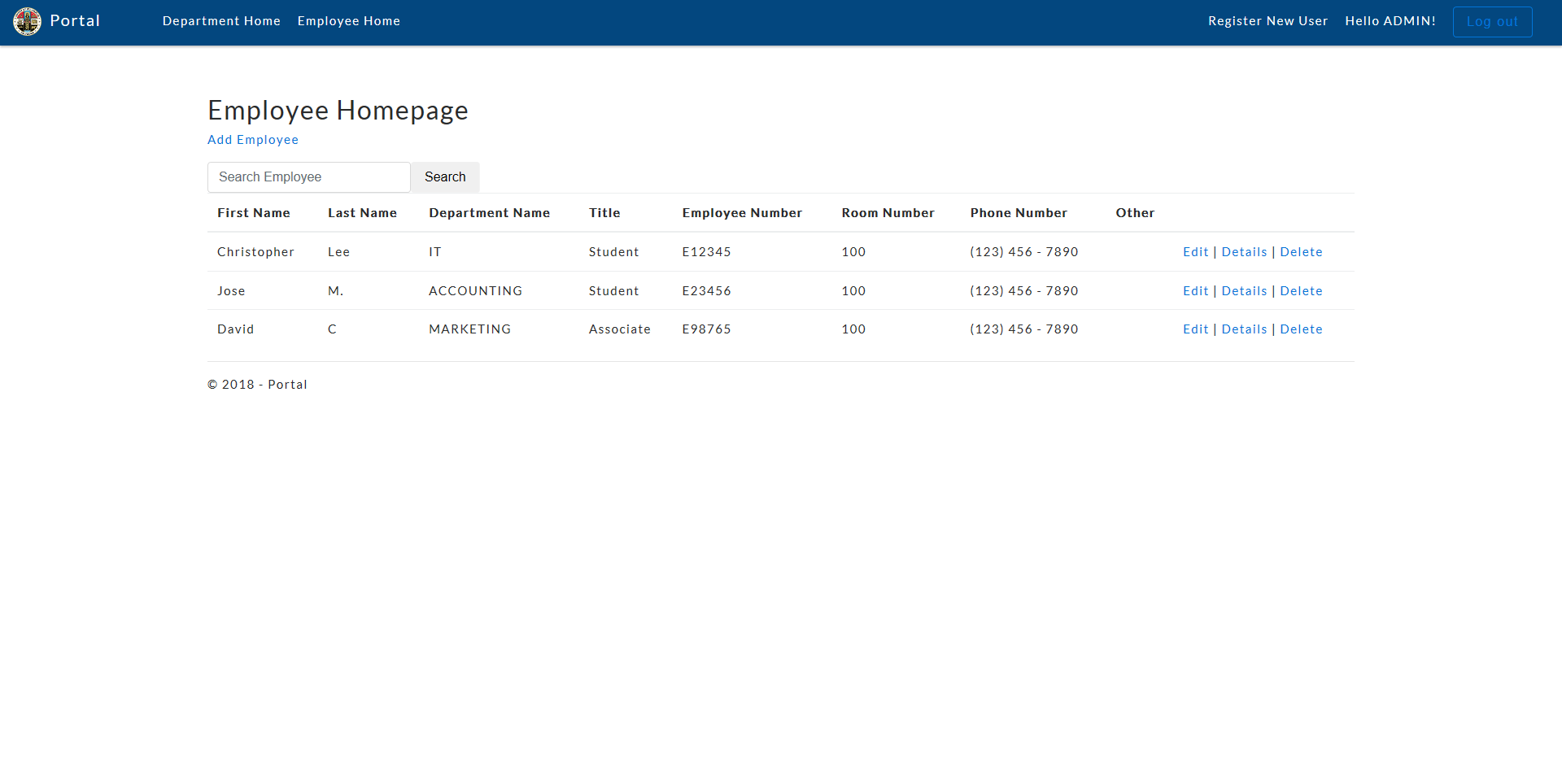
**8. User Interface**

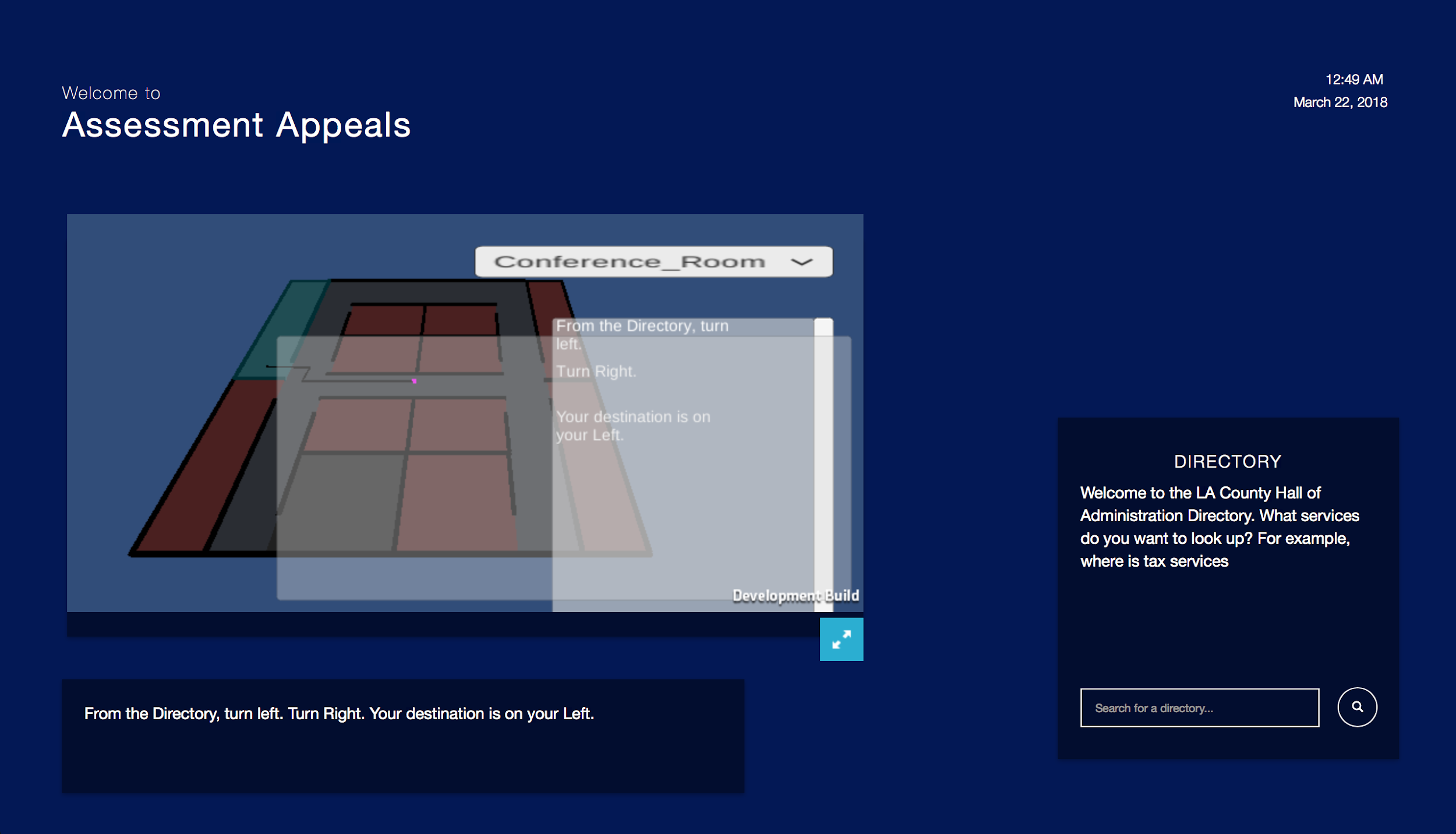
**8.1 Overview of User Interface**

The interface of the map is the main component of the web page will be displayed on the left-center position of the screen. The speech/search output results are displayed on the bottom-right corner. In the map interface, the user can choose what room they want to go to by choose from a dropdown list of rooms. Once selected, the map will animate a path from the starting point of the touch screen to the destination of the selected room. The section where the room is located will highlight using a different color. The user can also select a room to get more information about the room. A sidebar will be displayed on the left side showing it. For the search module, the user can speak to the Amazon Echo device connected to the smart board. When launching Alexa, a welcome greeting is displayed as text output in the search module. All the speech output from Alexa will be converted and displayed as text in real time as the user interacts with the device.

For the web portal, staff members of a department can put in information that will update content from the web page, Unity map, and Alexa skill. First, the user must login in order to access it. Once logged in, the administrator can register new accounts for the portal. The department is the first page where the user is directed. If empty, the user can add a department to the table, which corresponds to the database. Each department is displayed on the table where the user can edit and view details. The nav bar includes a link to the employee homepage. Similar to the department page, the user can add new employees to the database based on a certain department. The table of employees is shown and updated according to edits made by users. For each account, a user can go to account settings and update information such as email address and phone number.

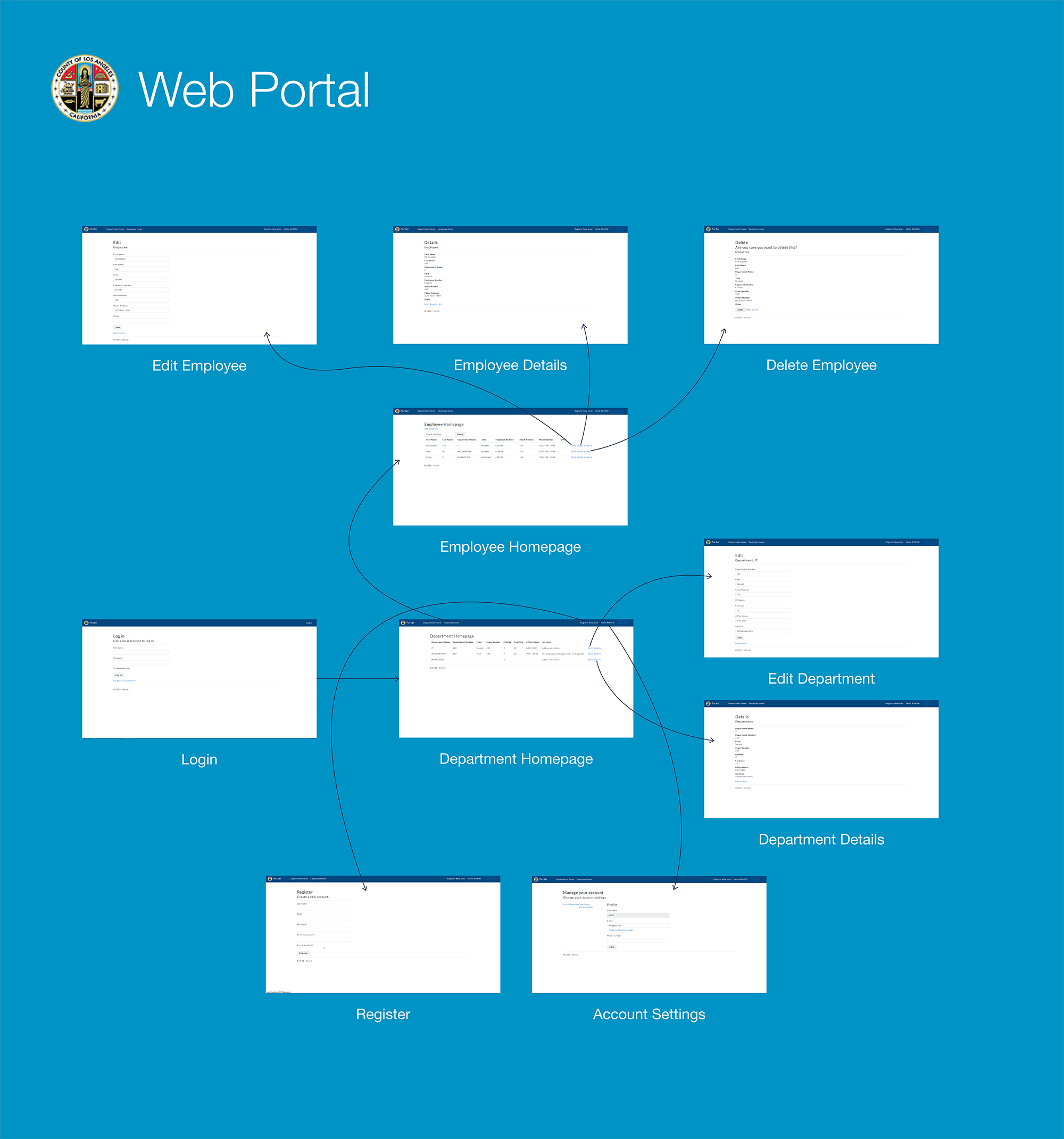
**8.2 Screen Frameworks or Images**

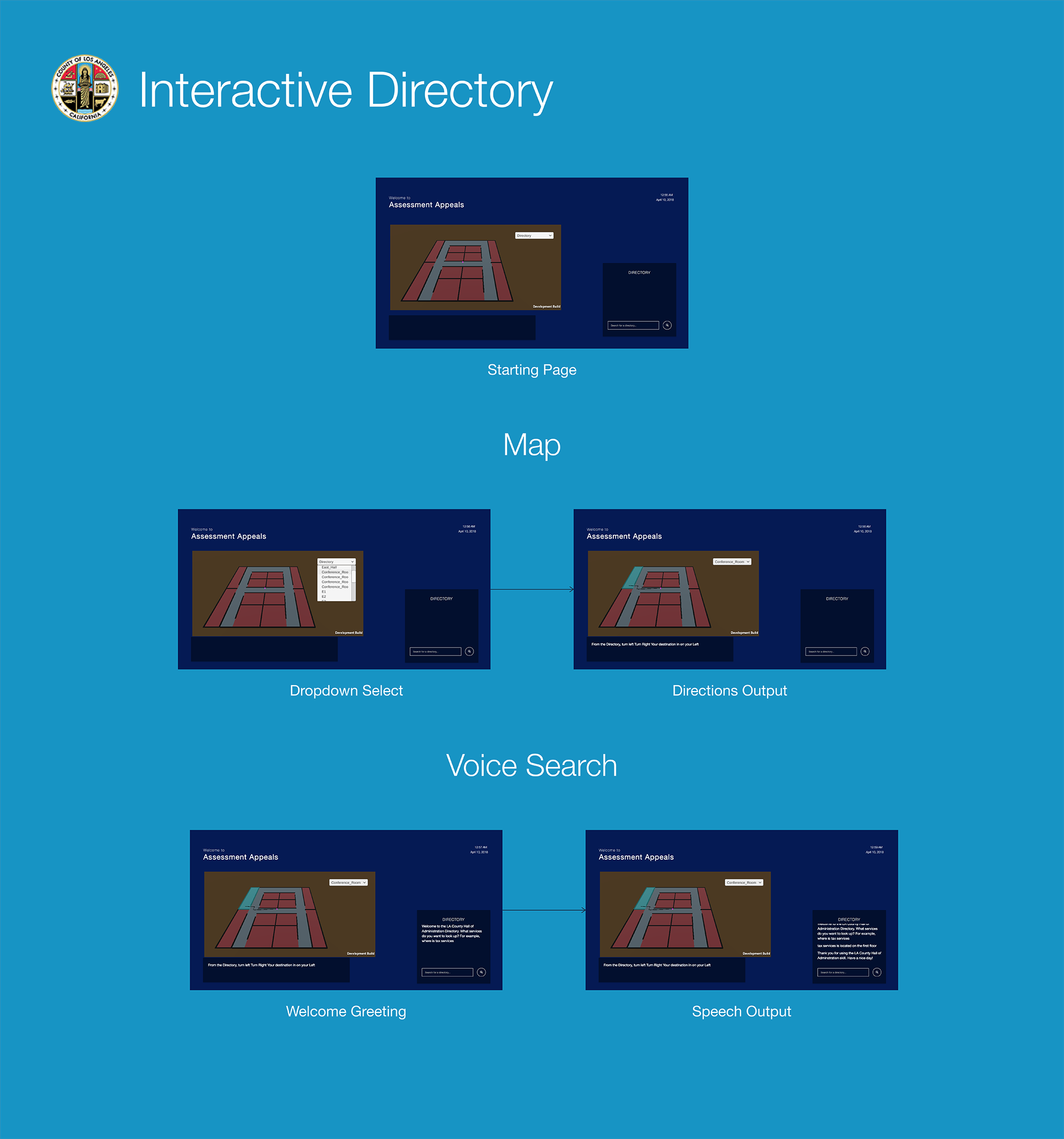
****



**8.3 User Interface Flow Model**

A discussion of screen objects and actions associated with those objects. This should include a flow diagram of the navigation between different pages.





**10. References**

<List any other documents or Web addresses to which this SDD refers. These may include other SDD or SRS documents, user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Brad Appleton <brad@bradapp.net>  <http://www.bradapp.net>

<https://www.cs.purdue.edu/homes/cs307/ExampleDocs/DesignTemplate_Fall08.doc>