Ryan Roper, Maxwell Kizewski, Jared DeWinne

Computer Science Capstone COM497

Dr. Nguyen

20 January 2020

Residence Life Web Application Proposal

After a discussion within the group, we have decided to create a web application for Saint Leo University’s Office of Residence Life. Two members within our group work within in the department and feel as though there could be a great need for a website where there will be additional resources for Resident Assistants and Residence Life team. As a Resident Assistant, aspects of the job should be easily accessible. There are moments where seconds count in the position; which means centralizing and integrating new tools for RAs to better do their jobs in a quick timeframe would be extremely helpful. Therefore, our group is planning to create an auxiliary website to lighten the load of the average Resident Assistant and the rest of the Residence Life staff.

Our web application will bring together all the necessary components of the Resident Assistant position into one, centralized location. Currently the position has several interfaces that are used, which results in the use of several different links that are sporadically located. Our goal is to centralize these tools that the RAs currently use and introduce new applications into their user ecosystem to make their jobs easier. This application would result in being a central location for: Current RA Website, Duty Log Calendar, Program Calendar, JAMS/Simplicity. New features that could potentially be added include: Instant Duty Swap, Duty Check-in, Anytime Roster, On-call Quick Access Phone Links.

In order to make this possible, our group will be working closely with Assistant Director of Residence Life for Student Conduct Sean Ferguson, Executive Director of Campus Life Sean VanGuilder, and Associate Vice President of Student Affairs Ken Posner to get a better idea of what they would like to see from an advisory perspective. We will also host focus groups with Residence Life staff to help define boundaries for our project. Overall, we’ll use the waterfall software development cycle to keep us organized and on track during this process.

1) STAKEHOLDER REQUIREMENTS

Max and Ryan held a Stakeholder meeting on Friday, January 24th. The meeting consisted of mainly supervisors and professional staff for the Office of Residence Life, in which we surveyed their desires for what they would like to see in a new application. It seemed like there was a lot of positive feelings towards the idea of a new application that were in conjunction with the hope that key elements of the current infrastructure are not changed too much.

The consensus seems to be that there is a need for some type of application that simplifies and mainstreams RA accessibility among all its platforms. Several applications are used simultaneously in the position, so implementing an application where these can be used together would be beneficial and efficient. Below are the do’s and don’t's from the meeting that we had with our stakeholders, as well as new features that would be desired in the application.

* Do’s and Don’ts
  + Don’t change too much of the existing infrastructure of the websites
  + Easy to maintain
  + User friendly
  + Easy accessibility online (tied to OKTA)
  + Easy Mobile User Interface and Desktop Interface for RAs & Professional Staff
  + Admin. Access for Professional Staff
* Key Features
  + Access to currently used applications: (JAMS, Residence, RA Website, TeamUp (Duty), TeamUp (Programming))
  + Duty Swapping System
  + Duty Check-in System (via Text or Email)
  + Lockout Database [NEW UPON STAKEHOLDER MEETING]
  + On-Call Quick Access Phone Links
  + RA/RAD/Prostaff Mobile Directory Access (Email & Phone #)

It should be noted that this is a list of desired features that would like to be seen by the staff. Given time restraints, we may not be able to completely implement all of these features. This list of features should give us enough work to build a solid infrastructure around our project that is adequate for capstone and meets the needs of the Office of Residence Life at Saint Leo University.

2) LIST TECHINCAL REQUIREMENTS:

Based on our desire for a web-based application, that allows for speedy communication, and unmatched organization, we need a few services, and sections of knowledge to make this project possible. The table below represents our requirements, and the summary below attempts to explain each and every one of them.

|  |  |  |
| --- | --- | --- |
| **Programming Languages** | **Online Services** | **Assistance/ Authorization from other entities** |
| Objective-C | AWS EC2 service | UTS permission to add program to Okta |
| Swift | Domain Hosting & DNS supplier (GoDaddy, Amazon Route 53) | IT permissions to setup access management for the Okta/Microsoft portal to authorize users |
| HTML | Microsoft Login Portal management |  |
| CSS |  |  |
| PHP |  |  |

Our project will feature a web-based application that authorized users can access from any platform they wish, either by downloading an app or going to our website. To make this possible, we need the above-listed services/requirements that will enable the finished project to work seamlessly with real data. First, we need to host the service. This will require server space/cloud equipment from a company such as AWS, which we are currently planning to use. After this is hosted, we need to ensure a constant DNS connection, by utilizing the services of Amazon Route 53 and GoDaddy to purchase a domain name, and static IP for our AWS instance. To meet the goal for scalability, and customization we would need to enable on-demand scaling from AWS for our specific instance, allowing any means of traffic and data to be allocated for, without overspending for a server bigger than we need. The overall program we are planning on building is predicted to only use HTML and CSS, along with PHP, for any email functionality. The application by nature is simple, but how we are implementing it requires full accessibility from all entities. Using knowledge with Swift and Objective-C, we will create iOS and Android applications that connect to the mobile version of the program, such as using a Safari App Extension for iOS. Where user management comes into play, is where we need authorization from UTS/TI^3 to change Okta and identify Microsoft User management. We want specific users (RA’s/RAD’s/etc.) to access the primary form of the platform, allowing them to use the features granted to them, while other users (their bosses/department heads) have management features where they can adjust functions such as user role management, and access permissions. This will require implementation with both Okta and Microsoft Portal since they are the two forms of user authorization used for Saint Leo’s platforms.