RED Phone

By

Brad Stapleton, Chris Rosiak, Paul Beeson

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Brad Stapleton	4/17/17
	Date
Chris Rosiak	4/17/17
	Date
Paul Beeson	4/17/17
	Date
Brian Verkamp, Faculty Advisor	4/17/17
	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2017

TABLE OF CONTENTS

FIGURES

Acronyms and Abbreviations

AHD	Assistant Hall Director
CC	Community Coordinator
GUI	Graphical User Interface
RA	Resident Advisor
RED	Resident Education and Development
VoIP	Voice Over Internet Protocol

ABSTRACT

In 2017, 5200 students at the University of Cincinnati resided in some type of housing facility managed by the university. With students living on their own, some for the first time, these residential students may encounter issues or situations that they need some assistance with. When such emergencies occur, the appropriate personnel need to be notified to assist with the situation. Our VOIP based phone system solved problems with call routing and the exchanging of information amongst residents, resident advisors, and senior staff members. Our system increased efficiency by automating daily call flow, accountability by providing central call logging, and increased response times by eliminating call operators as well as the need for multiple contract phone numbers. By implementing a VOIP phone system for the University of Cincinnati housing units, residents breathe easier knowing someone is on the other end of the phone when they need them most.

1. Problem Statement

1.1 Introduction

Resident Education and Development (RED) is a division of the University of Cincinnati, Cincinnati, Ohio, Student Affairs office. RED is made up of six full time central office personnel, thirty fulltime residence hall leaders, and greater than 100 student resident advisors serving a student population of 5,200. RED operates out of eleven different residence halls throughout the University of Cincinnati campus, as well as three remote facilities in the surrounding community. With 5,200 students living in buildings under the watch of RED, buildings are staffed around the clock to assist with residential emergencies as they manifest. RED staff is required to be on-call following a regularly rotating schedule to properly respond and resolve residential emergencies. Depending on the type of emergency, RED staff may be required to communicate with other members of the team to acquire the information or assistance needed to properly resolve a situation.

1.2 Project Description

As a team, we will be creating and test piloting a voice over IP (VoIP) system to assist with the daily workflow of RED's communication and operations. Our system, designed and built with housing operations in mind, will be a very beneficial system to assist in RED's daily work. Some current systems do not meet all of the current requirements of RED or the University of Cincinnati because of their cloud hosting platform. Other systems require sophisticated configuration to align with the needs of RED and university housing. Our team's solution will come as an out-of-the-box solution tailored

to housing operations with the features and functions most important to RED. In addition, our system comes at a much lower cost due to the absence of license fees. Our VoIP system will allow RED to have better communication over the phone as well as help RED better serve a community of 5,200 residential students.

1.3 Problem Statement

Currently, RED utilizes a variety of tools for residential students to communicate with RED staff in the event of an emergency. The University of Cincinnati's current analog phone system is the first level of RED's phone communication usage. Front desk phones are a main point of contact for anyone outside of the hall to communicate with hall personnel. Once these full-time desk workers leave for the day, phones go unmanned or left to forward to another building where other staff members are currently positioned. Phone calls received are screened and then passed along to the appropriate staff member after referencing a Google Spreadsheet with staff contacts and calendars. Residential students can contact on-call RED staff members using a Google Voice number that is attached to student staff's personal cell phones. These numbers can sometimes fall into the wrong hands where phone calls from parents, community members, and residential students outside of the building can call in. Student staff of RED utilize a third level of phone communication by contacting a pair of Google Voice numbers to contact senior level staff when problems arise. The mixture of analog telephones, Google Voice, and personal cell phones create a very confusing process for residential students to get the help they need when they need it most. In

addition, utilizing Google Voice is not a best practice for the University of Cincinnati as data is stored in Google's environment, not the University's.

1.4 Design Objectives

The phone system should have the ability to operate out of the box with minimal configuration for university housing units. The project aims to create a fully scalable VoIP system that is highly available to accommodate 24/7 emergency operations. In addition, the project aims to create an advanced web GUI to operate and adjust settings as needed by the department staff. Features such as call recording, call forwarding, voicemail, and group calling will be incorporated into the finished system. Other special feature requests by RED will be evaluated.

1.5 Methodology

The system will be developed in stages so each feature is isolated in its development. Therefore, no features will directly affect other features in the system during the development phases. Extensive testing will follow each of the development pieces to test functionality and overall stability of the system as a whole. Once all features have been built, the system will be rebuilt for consistency and rolled into beta testing. The beta will be tested in a select number of residence halls to find bugs and get feedback from users. All feedback will be taken into account for bug fixes to be applied and then rolled out again for further testing. Once final comments have been implemented into the system, the system will be rolled out as a fully finished product and implemented

into the daily workflow for RED staff and members of the University of Cincinnati Information Technologies. Final functionality will be showcased at the annual IT Expo.

1.6 User Profile

Our phone system and web interface will be utilized by three different groups of users inside RED. Resident Advisors (RAs) are the first group. This group is comprised of full time undergraduate students who live on floors with residential students. RAs work a total of 15 hours per week in their role. Residence Hall Leadership are comprised of Residence Hall Directors (RHDs), Community Coordinators (CCs), and Assistant Hall Directors (AHDs). Each of these individuals are fulltime, hourly employees that live and hold offices inside the various residence hall buildings on campus. RHDs, CCs, and AHDs work a total of 40 hours per week. Department Directors are comprised of full time employees that oversee all RHDs, CCs, and AHDs. These members work in the central RED office but do not live in any residence hall buildings or on campus.
Each of these groups have different needs and responsibilities when it comes to the VoIP phone system. These responsibilities and needs are outlined in Figures 1 through 3 and graphed in Figure 4.

User Profile 1: Resident Advisors
Application: VoIP communication system for housing operations.
Potential Users: Resident Advisors
Software and Interface Experience: The user will interact with a web GUI that will update a central database that modifies configuration of the VoIP system to correctly route calls to the appropriate destination.
Experience with Similar Applications:

The user will need to have experience with web based forms, call flow, and organizational procedures/policies.
Task Experience: The user will be presented with a list of fellow resident advisors. This list will feature checkmarks of those currently on-call and those not on-call within their unit.
Frequency of Use: The user will interact with the system on a daily basis to change the on-call rotation and update call flow based on the resident advisor duty schedule.
Key Interface Design Requirements that the Profile Suggests: The user will interact with a web interface to make changes to call flow and on-call rotation. The web interface will be customized to easily allow resident advisors to update call routing daily via web form fields.

Figure 1: User Profile 1

User Profile 2: Residence Hall Leadership
Application: VoIP communication system for housing operations.
Potential Users: Residence Hall Leadership
Software and Interface Experience: The user will interact with a web GUI that will allow reports to be generated containing received calls, call recordings, and call durations taken by resident advisors.
Experience with Similar Applications: The user will need to have experience with Microsoft Excel, web based reports, log files, and audio playback applications.
Task Experience: The user will be presented with basic call metrics and phone log analysis for reporting, archiving and compliance**.**
Frequency of Use: The user will interact with the system twice a week to verify calls are being received and submit weekly call reports to department directors.

Key Interface Design Requirements that the Profile Suggests:
The user will interact with the web interface to pull reports and call logs. The web interface will be customized to allow the user to format reports based on departmental needs.

Figure 2: User Profile 2

User Profile 3: Department Directors
Application: VoIP communication system for housing operations.
Potential Users: Department Directors
Software and Interface Experience: The user will interact with the web interface to delegate user permissions and verify all organizational policies and procedures are being followed.
Experience with Similar Applications: The user will need to have experience with access control, policy management, and policy implementation.
Task Experience: The user will manage user accounts and verify policies are being followed as set forth by the university and student affairs.
Frequency of Use: The user will interact with this system on a normal audit rotation and when user accounts and permissions need to be modified.
Key Interface Design Requirements that the Profile Suggests: The user will interact with the web interface to view and modify user accounts. The web interface will be customized to allow the user to create, modify and delete permissions.

Figure 3: User Profile 3

Figure 4 outlines the roles for each user of the VoIP system.

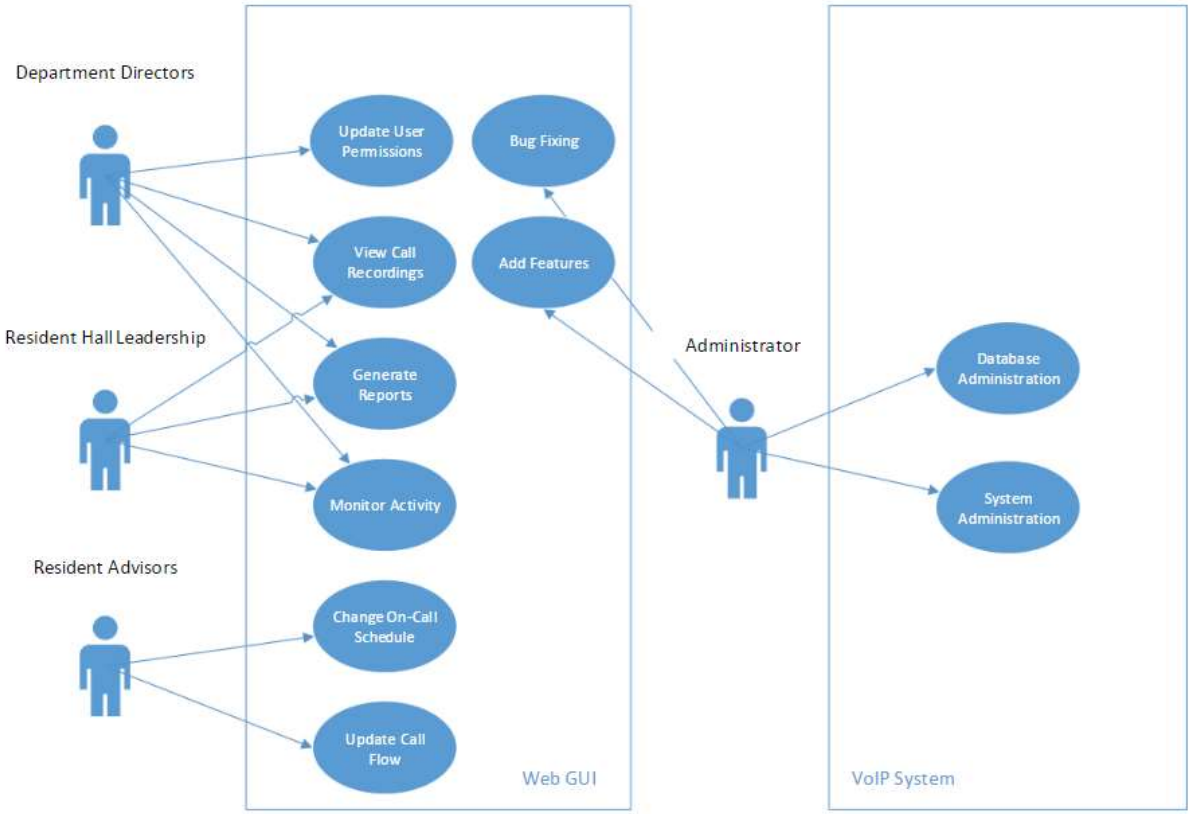

Figure 4: Use Case Diagram

2. Project Management

2.1 Budget

Figure 5 presents the project budget. It shows a real-world cost of $11,620. Due to the agreement reached with RED, the project will be given away for a cost of $0 in return for feedback from a committee of residence life professionals and the opportunity to test the system in an active residence hall.

Item	Units	Unit Price	Total
Labor	120	$ 65.00	$ 7,800.00
Server Infrastructure	3	$ 1,000.00	$ 3,000.00
Phone Minutes	2000	$ 0.01	$ 20.00
Miscellaneous Expenses	1	$ 800.00	$ 800.00
		Grand Total:	$ 11,620.00

Figure 5: Projected Project Budget

2.2 Objectives/Deliverables

Project deliverables and deadlines for Fall Semester 2016 are presented in Figure 6.

Project Deliverables	Start	End	Duration (days)
Research and development	**8/22/2016**	**10/2/2016**	**41**
Finalize group for project			6
Finalize choice of group project topic			12
Investigate software packages for this project			15
Pick an organization to work with and focus the project on			6
Meet with UCIT and RED to talk about need for this project			2
Configure Base Asterisk Server	**10/3/2016**	**10/17/2016**	**14**
Download, Compile and install Asterisk			2
Build Extensions.conf			4
Build SIP.conf			4
Create users and test functionality			4
Build Chef Recipes	**10/18/2016**	**10/31/2016**	**13**
Look at automation tools for compiling and installing Asterisk			6
Build chef recipes and look at GIT or SVN for version control of config files			7
Test VOIP Communication and Build a working demo for presentation	**11/1/2016**	**11/13/2016**	**12**
Make sure 2 way audio works			1
Make sure several different types of phones can connect			4
Build and chef a production server for presentation and test it			7

Figure 6: Project Deliverables & Deadlines Fall Semester 2016

Project deliverables and deadlines for Spring Semester 2017 are presented in Figure 7.

Project Deliverables	Start	End	Duration (days)
Enable 3rd party access for real-world communication	**12/12/2016**	**12/26/2016**	**14**
Select a 3rd party provider			2
Create account with provider			2
Build integration with 3rd party provider			10
Tweak chef recipes	**12/27/2016**	**1/9/2017**	**13**
Make recipes aware of new modules and features			13
Database Features	**1/10/2017**	**1/30/2017**	**20**
Make more features database driven for more flexibility			10
Have call logs and transaction details put in database so website can read that info			10
Build a Find Me function	**1/31/2017**	**2/20/2017**	**20**
Ring multiple external numbers and have the quickest one answer the phone			20
Prepare for IT expo	**2/20/2017**	**4/10/2017**	**49**
Build final working demo			21
Test Chef Recipes and testing on AWS / Azure, etc			21
Pilot in UC Residence Hall			7

Figure 7: Project Deliverables & Deadlines Spring Semester 2017

2.3 Project Schedule

Figure 8 outlines the project timeline for Fall Semester 2016.

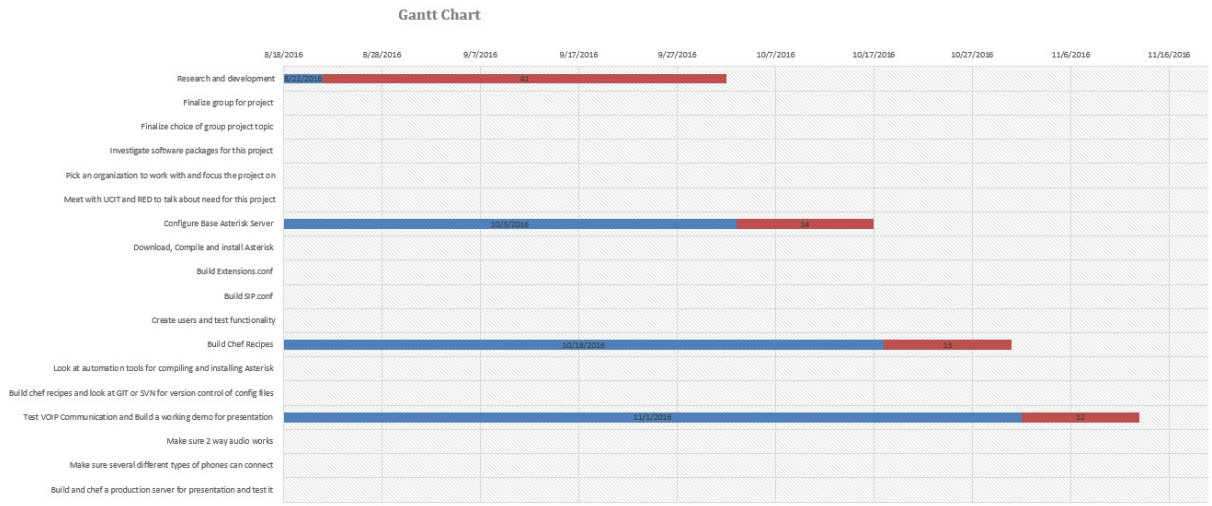

Figure 8: Gantt Chart Fall Semester 2016

Figure 9 outlines the project timeline for Spring Semester 2017.

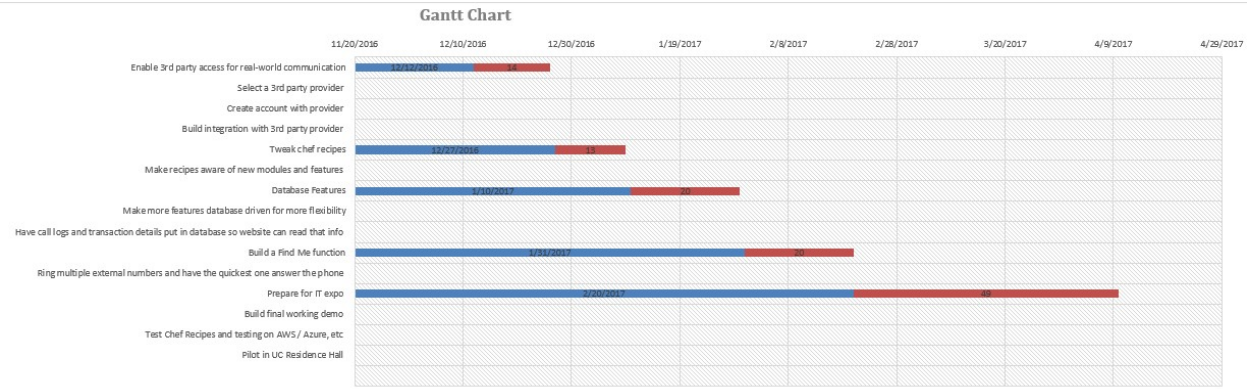

Figure 9: Gantt Chart Spring Semester 2017

3. Technical Elements

3.1 Infrastructure

The VoIP system will be run in a virtual machine environment either as a cloud hosted or on-site solution. The virtually hosted environment will be deployed with a cloud provider such as Amazon AWS or Microsoft Azure. The VoIP system will be powered by Asterisk, a free and open source VoIP PBX system. All Asterisk configuration will be stored in a MySQL database. Asterisk and MySQL will be configured with Chef to be deployed on demand. A third Linux machine will be introduced as a webserver to serve connect to the web. This webserver will host a frontend to the database utilizing the apache application. All configuration files will be checked out from a versioning repository such as GIT or subversion.

3.2 Programming

The VoIP system will be managed and controlled by users utilizing a web frontend to make changes to the MySQL database. The web interface will be developed to allow users to make changes in a graphical user interface (GUI). Changes made in the GUI will run MySQL commands in the background to make the requested changes in the database. These changes will then be reflected in the user's GUI session. The web interface will be programmed in PHP, which allows for easy MySQL integration and a simple design for the user.

3.3 Provider Integration

The VoIP system will peer with a 3rd party SIP provider in order to allow this system to interface with the phone system and the rest of the world. Our team plans to utilize Flowroute as the SIP provider for this project. The provider will interface with the Asterisk box to properly route calls that come into our phone numbers, and route them to the proper extension, and for outgoing calls.

3.4 System Diagram

Figure 10 below shows the overall flow of information through the VoIP system. Users will interact with the system via a web GUI that acts as a friendly frontend to a MySQL database. The MySQL database will be the primary controller acting as a liaison between the main Asterisk system and the user. Call recordings, call logs, call flow operations, and configurations will be accessed and modified using the MySQL database. MySQL commands will, in turn, control the main Asterisk system that interfaces with stationary and cellular phones as well as the SIP provider.

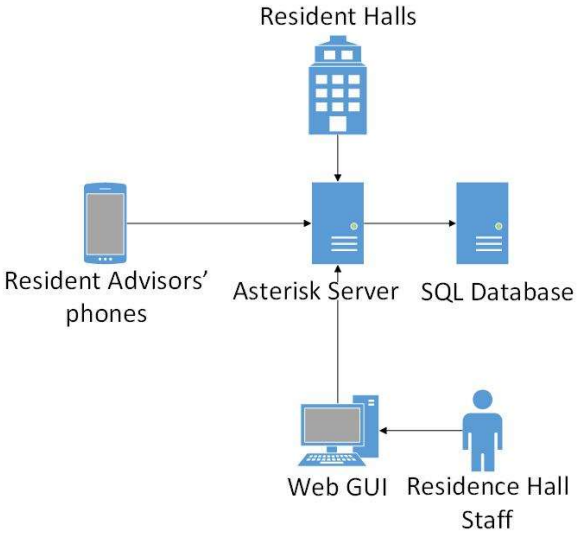

Figure 10: System Diagram

4. Test Plan

4.1 Overview

This section will discuss the methodology in which RED Phone's core system and web frontend will be tested. This should be used as a guide to identify, evaluate, and respond to potential failed tests. The following individuals will be involved in the testing phase:

- Systems Engineer
- Frontend Developer
- Project Manager
- RED Administrators
- Resident Advisors

4.2 Scope

The scope of testing RED Phone will include both the phone system's functionality as well as the web interface. Testing will be based after the needs and requirements of the stakeholders.

4.3 Objective

The objective of testing RED Phone is to verify stability and usability in the event of an emergency situation. Two areas of the application will be tested, the phone system and the web interface. Both will be tested and corrected separately in initial testing. Once initial testing is complete, both pieces of the project will be tested in conjunction with

each other. Tests will be conducted on isolated components first, then as a complete unit. During development, components will be tested for basic functionality by the developer.

4.4 Logging/Test Reporting

During testing, any issues observed on either system will be documented. The document will include the following information:

- What happened?
- What was the intended result?
- Where did the issue occur?
- When did the issue occur?
- Steps to re-create the issue?

At the conclusion of testing, all errors reported as well as feedback from test subjects will be sent to the project team via email to evaluate. The team will meet to discuss the tests to determine if the error is intended, a problem with the system, or a component that is deemed outside the scope of the system. Once the error has been assigned to an appropriate category (enhancement or problem), it will be added to a running checklist and assigned to the appropriate team member. All enhancements will be considered outside the current scope of the system. Problems will be considered inside the current scope of the system.

4.5 System Testing

RED Phone will first be tested on a per component basis before being tested as a single unit. This will allow the project team to make small adjustments to individual components before looking at the system large scale. By the end of the testing phase, it is intended to test the solution as one working unit, not individual components.

4.6 Testing Procedure

The following steps will be completed in order to test the system:

- Generate test scenarios based on the requirements of RED
- Create guides to complete the different scenarios with the intended result
- Report errors by answering the questions outlined above

Before testing is conducted, verification that the system is running will be done to ensure proper testing results. The following tests will be conducted:

- Web Interface UI – This test will ensure that the RED Phone web interface is designed in a way that is easy to navigate and use from an end user perspective.
- Web Interface Functionality – This test will ensure that the RED Phone web interface is fully functional and providing the intended results.
- Phone System Functionality – This test will ensure the RED Phone system is stable, operating correctly, and aligns with the project requirements.

4.7 Pass/Fail Conditions

It is expected for RED Phone's web interface and phone system to pass all tests deemed "critical" to be successful. Tests deemed "non-critical" will be taken into account but evaluated after "critical" errors have been addressed, corrected, and tested.

4.8 Testing Schedule

Beginning March 1, 2017, the project team began testing individual components within the phone system and web interface. Once all individual components have passed, testing on the entire system will commence. Testing will be conducted weekly but subject to adjustment based on the project team's evaluation of errors being reported. Unit testing should begin no later than March 20, 2017. All testing and patching should be concluded by April 2, 2017.

4.9 Risks

The following items could potentially impact the timeline listed above:

- Delay in error correcting
- Test subject availability
- Feature requests deemed "critical"

All changes in the testing schedule will be made by the project team and announced via the project manager.

4.10 Test Report

The first test is focused on the functionality of the Phone System.

What happened?	What was the intended result?	Where did the Issue occur?	When did the issue occur?	Steps to re-create the issue?
Destination showing wrong number in call log.	The number of the building they are trying to call needs to be listed.	Database	March 3, 2017	Make a call to a resident hall.
System continues to call after the RA hangs up the phone.	The call needs to be ended when either party ends the call.	Asterisk	March 3, 2017	Call an RA on duty and have them hang up.
911 being asked if they want to answer the call.	Direct answer.	Asterisk	March 3, 2017	Test call 911.
A call was not recorded.	The entire call needs to be recorded.	Asterisk	February 27, 2017	Have a call last more than 5 minutes.
A call never connected.	The call reaches the Auto-attendant.	Cell phone	February 27,2017	Try the call again after disabling Wi-Fi.

Figure 11: System Testing

The second test is focused on the front end Web GUI.

What happened?	What was the intended result?	Where did the Issue occur?	When did the issue occur?	Steps to re-create the issue?
Wrong number showing in call log.	The caller's number showing up in the log	Call Log Page.	February 20, 2017	Have the Auto-attendant transfer a call to the RA on duty.
Permissions not set correctly	Resident Advisors should only see On-Call rotation.	Front end Website	February 27, 2017	Login as different users.
Calls not showing up in Call log.	The call log should pull all call history.	Call Log Page.	February 20, 2017	Make calls to the system
Logout timer set too high.	Logging out should take you too login page within a second.	Logout Page.	March 1, 2017	Log out of the website.

Figure 12: Web Front End Testing

5. Conclusion

5.1 Fall Semester 2016

During the 2016 fall semester, the team managed to make internal phone calls local to the system, implement voicemail, call recording, and basic call flow. Testing conducted on these elements have proven their stability in a development environment.

5.2 Spring Semester 2017

During the 2017 spring semester, the team completed the following tasks:

1. Database driven operation
2. External call forwarding feature
3. Find Me functionality
4. Call logging feature
5. Web app creation
6. Receive feedback from stakeholders on features
7. Test system in residence hall

5.3 Presentation 2017

The system was presented on Tuesday, April 11, 2017 at the University of Cincinnati Campus Recreation Center. The system won the Infrastructure and Networking Solutions Category and was crowned the overall IT Expo winner.

Works Citied

Bollmann, Jens. "Installing MariaDB 10 on CentOS 7 / RHEL 7." MariaDB. Accessed April 12, 2017. https://mariadb.com/resources/blog/installing-mariadb-10-centos-7-rhel-7.

Davenport, Malcolm. "Realtime Database Configuration." Asterisk Project. May 23, 2013. Accessed April 12, 2017. https://wiki.asterisk.org/wiki/display/AST/Realtime Database Configuration.

"How to Install Asterisk 13 and PJSIP on CentOS 6." Digium. February 24, 2015. Accessed April 12, 2017. http://blogs.digium.com/2015/02/24/install-asterisk-13-pjsip-centos-6/.