StabiliVOIP

by

Randy Baldrick, Logan Hahn, and Brett Tierney

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

Logan Hahn _____ _____

Logan Hahn *4-27-2020*

Randy Baldrick _____ _____

Randy Baldrick *4-27-2020*

Brett Tierney _____ _____

Brett Tierney *4-27-2020*

Ryan Moore _____ _____

Ryan Moore, Technical Advisor *4-27-2020*

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

April 2020

StabiliVOIP

Prepared by

Brett Tierney, Randy Baldrick, & Logan Hahn

Students of

University of Cincinnati

College of Education, Criminal Justice, & Human Services

School of Information Technology 2020

April 27th, 2020

TABLE OF CONTENTS

List of Illustrations

FIGURES

TABLES

List of Abbreviations

VOIP – Voice Over Internet Protocol

AWS – Amazon Web Services

PBX – Private Branch Exchange

Abstract

Business communications are becoming more and more demanding as technology continues to advance in our ever-changing world. VOIP communications allow a business to stay connected with a reliable and flexible method of handling their telecommunication needs. However, VOIP must have optimal networking conditions for it to work properly and reliably. We created a solution to allow businesses to stay connected while minimizing any impacts they may have on their everyday communications. Using Dynamic routing we were able to shift network traffic from overloaded VOIP servers in order to keep communications up and running. According to *PriceItHere.com* VOIP Solutions can cost $20 to $40 per month per user, which can add up fast. We were able to keep costs low by using open-source software. The result is a quick and reliable VOIP solution that can handle large amounts of traffic and keeps client costs low.

Introduction

Problem Statement:

VOIP calls are very sensitive to different network conditions such as jitter, ping, and bandwidth requirements. These issues cause calls to be dropped and call quality to degrade. According to VOIP provider Jive, an absolute minimum for bandwidth requirements to use their services is 100kbps. This is all relative to how many concurrent calls a client would like to be able to process and depends on the codec. Ping and Jitter are also a huge factor in determining how reliable calls will be. Jive recommends a round trip latency of 100ms or less, anything over 150ms will result in choppy sounding calls. Jive says that jitter should never uncontrollably spike or be above 20ms.

Solution:

We will dynamically route calls to the appropriate switch based on network factors such as latency and jitter. Once a call is placed at the network endpoint it will automatically ping each of the switches and select the one that has the best network conditions. This reduces the potential of a call suffering in quality due to bad networking conditions at the SIP switch that processes the calls.

Project Goals:

Deploy a VOIP network that is spread out across multiple regions within the AWS cloud environment. Have PBX systems sitting in each of the regions that can process and route phone calls appropriately. This VOIP network would be able to dynamically route calls based on which region has the least amount of network hops and latency.

Purchase a domain name for our project. Use that domain to establish DNS records for the calls to flow appropriately from region to region based on DNS SVR records.

Install softphones on end user computers in order to simulate call flows from the regions based on the geolocation of the end user computer.

Have a more cost-effective model than other VOIP solutions available in today's market, which range anywhere from 20-40 dollars per user according to *PriceItHere.com*. Provide better call quality and reliability than Microsoft Teams and Vonage

Develop a secure web application that would allow us to visualize the call flows that are occurring within our VOIP network. This would allow us to shed light into the network conditions that are happening in real time. It would also display which route each of the calls are taking and why.

Overview:

The reminder of this report outlines how our project will be completed. The report includes the following sections: project concept, design objectives, methodology, problems encountered, analysis of problems solved, and then ends with a conclusion of what we learned. There are also three diagrams: user profiles for each type of user in our project, a use case diagram, and a Gannt chart.

Discussion

Project Concept:

The big picture of our project is to provide small to medium sized businesses across the United states a cost-effective and reliable solution for optimizing VOIP communications within their company. StabiliVOIP will provide reliability, transparency, and customization that most VOIP solutions on the market are unable to provide.

The idea was first introduced by Logan Hahn. The 3 of us had experience using VOIP applications on various types of networks including both work and personal. Additionally, we could recall times where VOIP calls were not the best quality. After conducting research on what impacts poor VOIP calls we learned most issues are caused by several network conditions. Being IT Networking students, we wanted to solve that issue!

Design Objectives:

Our initial goals were developing a front-end admin UI to show the optimal VOIP call, build and implement a solution to dynamically route VOIP calls within a company's network cost effectively, and to secure our solution to protect all customers data.

As we were building, testing and deploying our final project we did not accomplish all our initial goals. As mentioned above, developing a front-end admin UI was a keyway we wanted to provide transparency within our project. This was a goal that that we had to abandon. However, we still were able to provide transparency within our project via admin logs that showed the network statistics of each call.

Methodology:

Establishing design requirements was a key factor in allowing us to meet our project goals. Defining design requirements allowed us to understand what was expected and needed for each deliverable. It also gave us an idea of roughly how long it would take to complete each deliverable and kept us on track. Lastly, it set a standard for what was needed for a goal to be considered achieved.

Establishing procedures allowed us to reach our goals by providing a baseline to follow for each step in our project. It saved us time and energy knowing we had a procedure for each task. Lastly, it held members of our team accountable for the standard work that was expected.

User Profile:

Figure 1: User Profile Sending/Receiving User, illustrates one of two user profiles for the StabiliVOIP product. It establishes expected outcomes that the user will receive and will outline the basic functionality that the end user will experience. It should also be noted that the end user will not need any prior IT experience in order to utilize StabiliVOIP at this profile.

User Profile: StabiliVOIP Sending/Receiving User

User Profile Form
PROJECT: StabiliVOIP – A VOIP routing solution.
POTENTIAL USERS: Any small to medium sized business that is looking to conduct VOIP communications in a cost-effective manner that provides transparency to the end-users.
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: End-users/customers from all technical backgrounds that utilize VOIP calls within their work environment. They will use the phone itself and won't necessarily interact with the application. The customer role will not require any IT experience.
EXPERIENCE WITH SIMILAR APPLICATIONS: • Microsoft Teams • Vonage • Skype for Business

TASK EXPERIENCE:

- Manage and maintain network infrastructure
- Configure phones and deliver them to the customers.
- Monitor VOIP communication using UI via a web browser
- Ensure application is secure

FREQUENCY OF USE:

Anytime VOIP communication is necessary within a business. This can range from constant phone calls to a couple calls a day.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Optimize VOIP calls to route to the nearest server that has the least amount of network congestion which may include latency, jitter, or packet loss.
- Admin UI with high availability and performance

Figure 1. Sending/Receiving User Profile

User Profile:

Figure 2: User Profile: Admin User, displays the expected outcomes that the admin user will experience when using StabiliVOIP. It establishes the prerequisites that the user will need to know before using the system and establishes prior products that users may have used before.

User Profile: StabiliVOIP Admin User

User Profile Form
PROJECT: StabiliVOIP – A VOIP routing solution.
POTENTIAL USERS: • Client IT staff • StabiliVOIP Staff • Special Users
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: Primarily, our solution will be managed and maintain by IT Administrators/Engineers using an admin web interface; to monitor the routes of the calls. They will also configure the phones before the customers receive them. They will also perform all other support of the application/software. They will need basic knowledge of networking and phone systems. Logs of user numbers and user interactions should be kept.
EXPERIENCE WITH SIMILAR APPLICATIONS: • Microsoft Teams Admin Interface • Vonage • Skype for Business

TASK EXPERIENCE:

- Manage and maintain network infrastructure
- Configure phones and deliver them to the customers.
- Monitor VOIP communication using UI via a web browser
- Ensure application is secure

FREQUENCY OF USE:

Anytime VOIP communication is necessary within a business. This can range from constant phone calls to a couple calls a day.
When issues with calls arise and tracking of the call will need to be performed.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Optimize VOIP calls to route to the nearest server that has the least amount of network congestion which may include latency, jitter, or packet loss.
- Admin UI with high availability and performance

Figure 2. Admin User Profile

Use Case Diagram:

Figure 3: Use Case Diagram, displays the use case for all associated users and their roles within StabiliVOIP. The diagram displays all users and their corresponding tasks when they are using the StabiliVOIP platform. The diagram also depicts how StabiliVOIP chooses which region to send a call over.

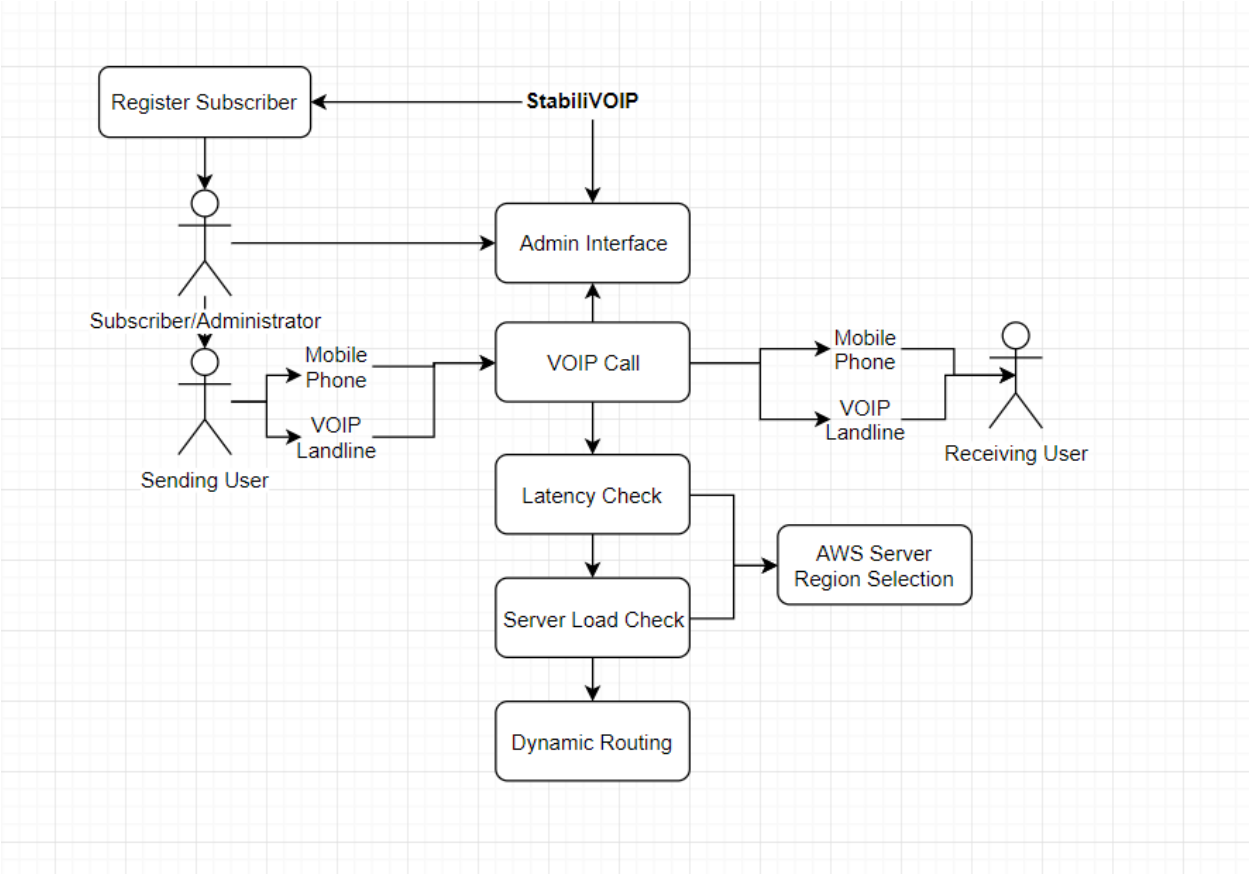


Figure 3. Use Case Diagram

Technical Discussion:

Network Overview - StabiliVOIP is a blend of multiple technologies coming together to create a streamlined solution. We use Amazon's EC2 to host our PBX servers in the N. California, N. Virginia, and Ohio regions. We use Amazon's Route53 for DNS resolution and for latency-based DNS resolution to our PBX servers from the client device. FreePBX is the operating system that is used for our PBX servers and is based off the Asterisk framework which is used in many other PBX distros. Telnyx is our VOIP trunk provider which is our connection to the public telephone network which allows us to receive and make external calls.

Application Overview - Most users will not be aware that they are interacting with StabiliVOIP as this is more of a backend type of solution. However, users will connect to the StabiliVOIP network and will automatically be placed on the best server available to them based on various network conditions such as latency, packet loss, and jitter. This connection can occur from a softphone on their laptop or smartphone or from a dedicated VOIP phone that plugs into an Ethernet port. The softphone will make the initial connect to Route53 and then forward that client to the best EC2 server.

Security Elements - Security is a major component to our project since phone calls can carry sensitive information and we want to ensure our clients have no doubts about the integrity of their calls. StabiliVOIP is whitelisting trusted IPs to access our PBX servers which includes the ability to make, receive calls, as well as authenticating into the server. This is to help stop attacks on our servers from unknown IPs who try to brute force and take down our servers. Access to the Management interface is even further restricted to only trusted IPs of individuals who would need access to these features. To authenticate a phone or "extension" into the PBX server you need a secret which is the password associated with the specific extension you are trying to authenticate. There is a forced policy to ensure that all secrets are strong and use a set of password requirements, this is designed so that no one can easily steal or guess someone's client password for StabiliVOIP.

Why StabiliVOIP is Better:

Price – VOIP Pricing Varies between provider and specific client needs but according to *PriceItHere.com* VOIP Solutions can cost $20 to $40 per month per user, this includes Microsoft Teams and Vonage. We use open source software which keeps our projected price much lower.

Networking – The simplest way of measuring network quality is ping. Ping checks if a host is available and gives you the time it takes for a host to respond. When we ping our StabiliVOIP servers from a personal computer in Westwood Cincinnati Ohio we get the following response times: Ohio=40ms, North Virginia= 30ms, NorCal=70ms. When pinging teams.microsoft.com we got a response time of 65ms. While this response time is serviceable, it is not ideal. Even if our primary server were to go down, our secondary server would still have a response time 30% faster than teams.

Transparency – Some VOIP providers do not disclose where the calls that you place on their network go to. StabiliVOIP aims to provide the highest amount of transparency and gives you the location to where our servers are located. Companies like Vonage do not disclose where the calls get sent and could be seen as a sign of an untrustworthy company to do business with.

Customization/Security - According to *PCMag,* Vonage poses security issues because of their compatibility requirements. Vonage requires users to turn off stateful packet inspection on some

firewalls and routers. Additionally, Vonage even requires that SIP application layer support be turned off at the router layer, which could result in customers being forced to purchase new hardware routing hardware. Vonage requires open access to the public internet instead of disclosing IPs/FQDNs this is a big security risk, most businesses put their VOIP traffic on a separate VLAN to further separate traffic and ensure only permitted traffic is allowed. Since Vonage doesn't disclose that information this cannot be done.

Technical Architecture Diagram:

Figure 4: Outbound Calling Diagram, shows the network architecture for StabiliVOIP. The Client is connected to their network and needs to make a call. They connect to StabiliVOIPs servers through their softphone. Amazon Route 53 DNS connects the user the best StabiliVOIP server depending on network conditions. The call goes through the optimal server and then through the StabiliVOIP VOIP Trunk. Then the call hits the Publicly Switched Telephone network and is routed to the user.

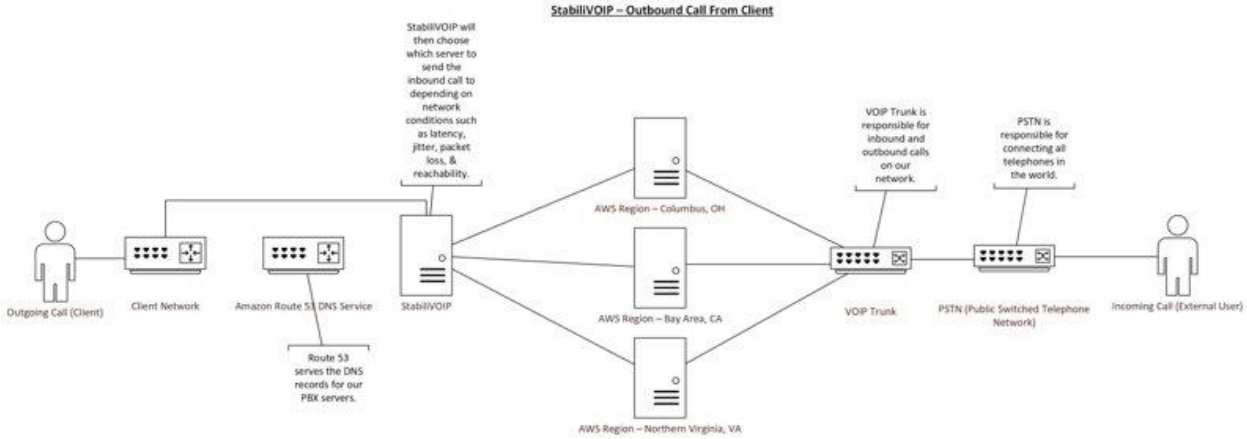


Figure 4. Outbound Calling Diagram

User Interfaces/Visuals:

Figure 5: Bria Solo Softphone, shows what the user will be seeing as they connect to StabiliVOIP using Bria Solo.

Figure 5. Bria Solo Softphone

Figure 6: FreePBX Admin Interface, shows the main FreePBX dashboard. Administrators use this to manage the PBX settings on a server.

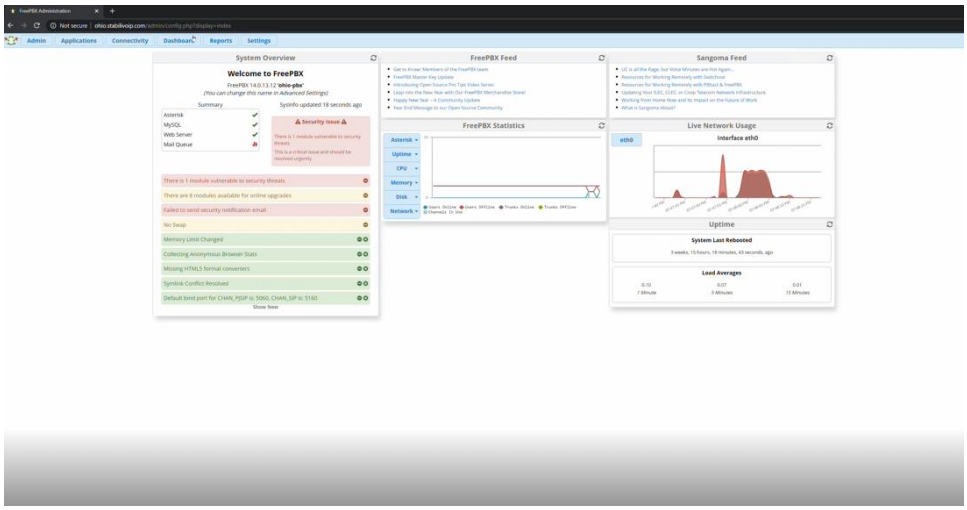


Figure 6. FreePBX Admin Interface

Testing Plan:

StabiliVOIP used a manual quality assurance testing methodology with defined test cases that will ensure our application stays online and is able to function for our end users. With all the different pieces coming together, we created test cases as each feature or function came online.

Our testing plan includes the connection to the StabiliVOIP application servers and the ability to make and receive telephone calls while also maintaining reasonable quality and latency. This ensures that our customers will be able to maintain their ability to use the StabiliVOIP application while also maintaining the quality that they expect. End user' connection to AWS which is where StabiliVOIP resides will be tested since the primary user of StabiliVOIP will be our end user or customer.

Methodology:

StabiliVOIP will be using manual quality assurance testing with defined test cases that will ensure our application stays online and is able to function for our end users. With all the different pieces coming together, we created test cases as each feature or function came online.

Testing Workflow:

- The main developer creates test cases as each new feature and function is built out for the StabiliVOIP application.
- The main developer runs these tests as features and functions are built to ensure functionality and no errors exist.
- Testers will then test the feature or function that the developers have created to ensure functionality and success in performing the specific action.
- If the test fails, the testers will notify the developer and the team.

Testing Cases:

Our testing plan, as displayed in Figure 7a: Test Case Plan, consisted of a table created in Microsoft excel. This table lists each test needed as a case and numbers them. The test is open until proven to pass under all circumstances.

Case #	Description	Status
1	Internal Call Cin to NV	Open
2	Internal Call Cin to NCal	Open
3	Internal Call NCal to NV	Open
4	Internal Call NCal to Cin	Open
5	Internal Call NV to Cin	Open
6	Internal Call NV to NCal	Open
7	Outbound call from CIN	Open
8	Outbound call from NV	Open
9	Outbound call from NCal	Open
10	Cin Server Connection	Open
11	Cin Server Connection/Latency	Open
12	NV Server Connection/Latency	Open
13	NCal Server Connection/Latency	Open
14	Extenstion Creation CIn	InDevelopment
15	Extenstion Creation NV	InDevelopment
16	Extenstion Creation Ncal	InDevelopment
17	Call Quality Assurance	InDevelopment
18	Web Interface Login	InDevelopment
19	Web Interface Call Tracking	InDevelopment
20	Cell Phone functionality	InDevelopment
21	Desk Phone Functionality	InDevelopment
22	PC Softphone functionality	InDevelopment

Figure 7a. Test Case Plan

Testing Results:

Our Testing results, as displayed in Figure 7b: Test Case Results, resulted in all passes beside the cases where we were not able to incorporate an aspect into the project.

Case #	Description	Status
1	Internal Call Cin to NV	Pass
2	Internal Call Cin to NCal	Pass
3	Internal Call NCal to NV	Pass
4	Internal Call NCal to Cin	Pass
5	Internal Call NV to Cin	Pass
6	Internal Call NV to NCal	Pass
7	Outbound call from CIN	Pass
8	Outbound call from NV	Pass
9	Outbound call from NCal	Pass
10	Cin Server Connection	Pass
11	Cin Server Connection/Latency	Pass
12	NV Server Connection/Latency	Pass
13	NCal Server Connection/Latency	Pass
14	Extenstion Creation CIn	Pass
15	Extenstion Creation NV	Pass
16	Extenstion Creation Ncal	Pass
17	Call Quality Assurance	Pass
18	Web Interface Login	NA
19	Web Interface Call Tracking	NA
20	Cell Phone functionality	Pass
21	Desk Phone Functionality	NA
22	PC Softphone functionality	Pass

Figure 7b. Test Case Results

Budget:

Table 1: Project Budget represents the associated real-world costs of deploying a project like StabiliVOIP. Labor was donated as this was our senior project and has not been sold to an outside entity.

No.	Item	Unit, Hours	Unit Price	Line Item Total	
1	Labor	750	$40	$30,000	
2	AWS EC2 (3 servers)	3	$50	$150	
3	AWS Route53 (1 zone)	1	$10	$10	
4	Domain Name Purchase	1	$10	$10	
5	FreePBX (3 servers)	3	$0	$0	
6	Bria Solo (individual licenses for 3 months)	3	$9	$27	
7	Telnyx Usage (in minutes)	100	$0.0075	$0.75	
8	Ubuntu LTS (3 servers)	3	$0	$0	
			TOTAL	**$30,197.75**	

Table 1: Project Budget

Table 2: This table represents our major Milestones achieved throughout the course of our project. Both Spring and Fall semesters are included.

Major Project Milestones (Deliverables)				
Fall 2019 Milestones				
AWS EC2 Setup Complete	9/30/19		Bria Solo Connection	11/11/19
Successful FreePBX Deployment(Ohio)	10/07/19		Successful Internal Call	11/18/19
Spring 2020 Milestones				
Virginia & Norcal PBX Deployments	01/06/20		StabiliVOIP Domain Acquired	
AWS Networking & Telnyx	01/20/20		Route 53 Dynamic Routing Successful	02/03/202
Testing Completed	3/23/20		Tech Expo	04/14/20

Table 2: Milestones/Deliverables

Project Schedule:

Figure 8a: Work Breakdown Structure, outline the timeline for our project.

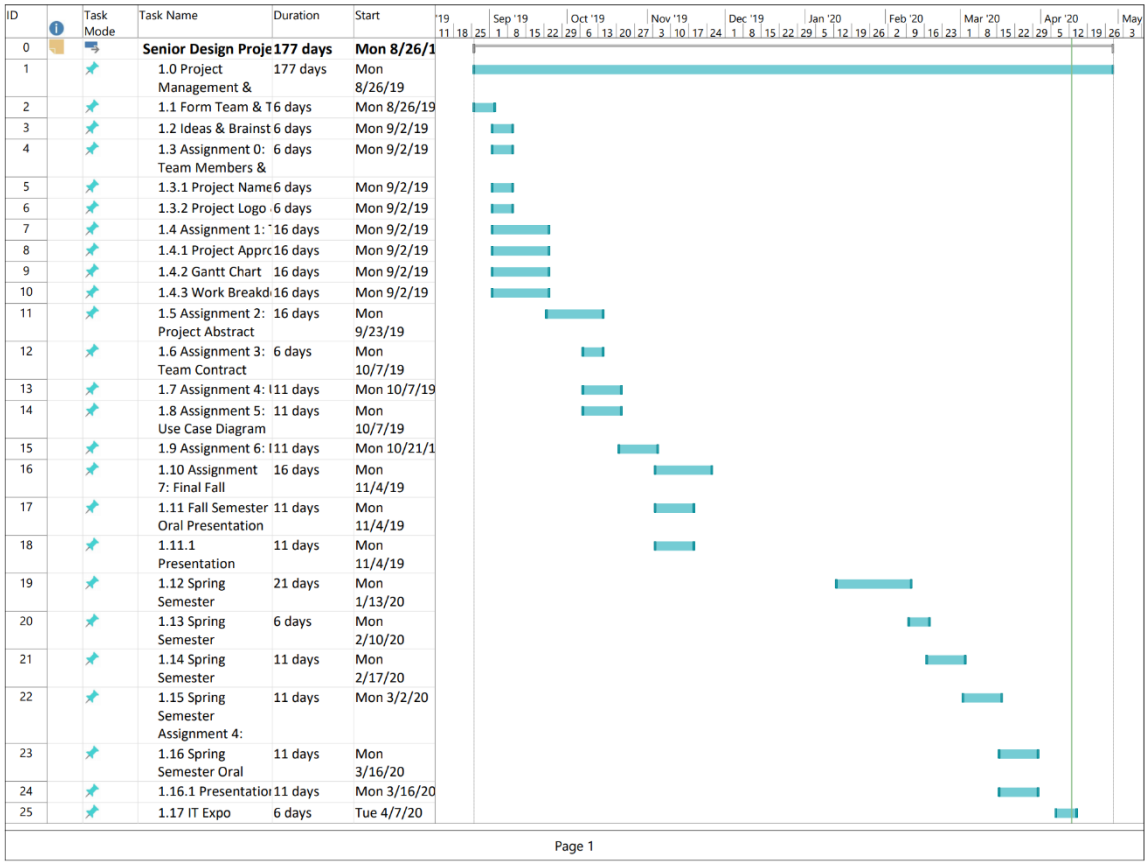
Task	Duration in Days	Start Date	End Date
1. 0 Project Management and Deliverables	252?	8/26/19	4/15/20
1.1 Form Team & Team Building	7	8/26/19	9/2/2019
1.2 Ideas and Brainstorming	7	9/2/2019	9/9/2019
1.3 Fall Semester Assignment 0: Team Members & Project Name	7	9/2/2019	9/9/2019
1.3.1 Project Name	7	9/2/2019	9/9/2019
1.3.2 Project Logo and Branding	7	9/2/2019	9/9/2019
1.4 Fall Semester Assignment 1: Team Contract	21	9/2/2019	9/23/2019
1.4.1 Project Approval	21	9/2/2019	9/23/2019
1.4.2 Gantt Chart	21	9/2/2019	9/23/2019
1.4.3 Work Breakdown Structure	21	9/2/2019	9/23/2019
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	21	9/23/2019	10/14/19
1.6 Fall Semester Assignment 3: Team Contract Resubmission	7	10/7/2019	10/14/2019
1.7 Fall Semester Assignment 4: User Profile	14	10/7/2019	10/21/2019
1.8 Fall Semester Assignment 5: Use Case Diagram	14	10/7/2019	10/21/2019
1.9 Fall Semester Assignment 6: Draft Report	14	10/21/2019	11/4/2019
1.10 Fall Semester Assignment 7: Final Fall Semester Report	21	11/4/2019	11/25/19
1.11 Fall Semester Oral Presentation	14	11/4/2019	11/18/2019
1.11.1 Presentation Practice	14	11/4/2019	11/18/2019
1.12 Spring Semester Assignment 1: Testing Plan/Report	28	1/13/2020	2/10/2020
1.13 Spring Semester Assignment 2: Abstract	7	2/10/2020	2/17/2020
1.14 Spring Semester Assignment 3: Draft Tech Expo Poster	14	2/17/2020	3/2/2020
1.15 Spring Semester Assignment 4: Final Poster	14	3/2/2020	3/16/2020
1.16 Spring Semester Oral Presentation	14	3/16/2020	3/30/2020

1.16.1 Presentation Practice	14	3/16/2020	3/30/2020
1.17 IT Expo	7	4/7/2020	4/14/2020
1.17.1 IT Expo Exhibit and Preparation	7	4/7/2020	4/14/2020
1.18 Spring Semester Assignment 5: Final Report	14	4/7/2020	4/14/2020
1.19 Spring Semester Assignment 6: Safe Assign Final Report	14	4/7/2020	4/14/2020
1.20 Spring Semester Assignment 7: Final Library Copy	14	4/14/2020	4/28/2020
2.0 Research	42	9/9/2019	10/21/2019
2.1 Software Research	42	9/9/2019	10/21/2019
2.1.1 Determine Method to Display Data	7	9/9/2019	9/16/2019
2.1.2 Pick Public Cloud Provider	7	9/16/2019	9/23/2019
2.1.3 Research Deploying Remote Switch	14	9/23/2019	10/7/2019
2.1.4 Research PBX software	14	10/7/2019	10/21/2019
2.2 Networking Research	42	9/9/2019	10/21/2019
2.2.1 Switch Connectivity	21	9/9/2019	9/30/2019
2.2.2 VOIP Routing	21	9/30/2019	10/21/2019
2.3 Security Research	42	9/9/2019	10/21/2019
2.3.1 Switch Configurations	21	9/9/2019	9/30/2019
2.3.2 Firewall Configurations	21	9/30/2019	10/21/2019
2.4 Miscellaneous	42	9/9/2019	10/21/2019
2.4.1 Budget	14	9/9/2019	9/23/2019
2.4.2 User Needs	14	9/23/2019	10/7/2019
2.4.3 Additional Features	14	10/7/2019	10/21/2019
3.0 System Design	42	10/21/2019	12/2/2019
3.1 Diagrams	35	10/21/2019	11/25/2019
3.1.1 Network Diagram	14	10/21/2019	11/4/2019
3.1.2 Interaction Diagram	7	11/4/2019	11/11/2019
3.1.3 Application Wireframe.	14	11/11/2019	11/25/2019
3.2 Legal	21	11/11/2019	12/2/2019
3.2.1 Disclaimers, EULA, Privacy Policy	21	11/11/2019	12/2/2019
4.0 Development	175	9/23/2019	3/16/2020
4.1 Set up 3 FreePBX Servers	105	9/23/2019	1/6/2020
4.1.1 Deploy PBX Server WEST	35	9/23/2019	10/28/2019
4.1.1.1 AWS PBX Server Deployment	7	9/23/2019	9/30/2019

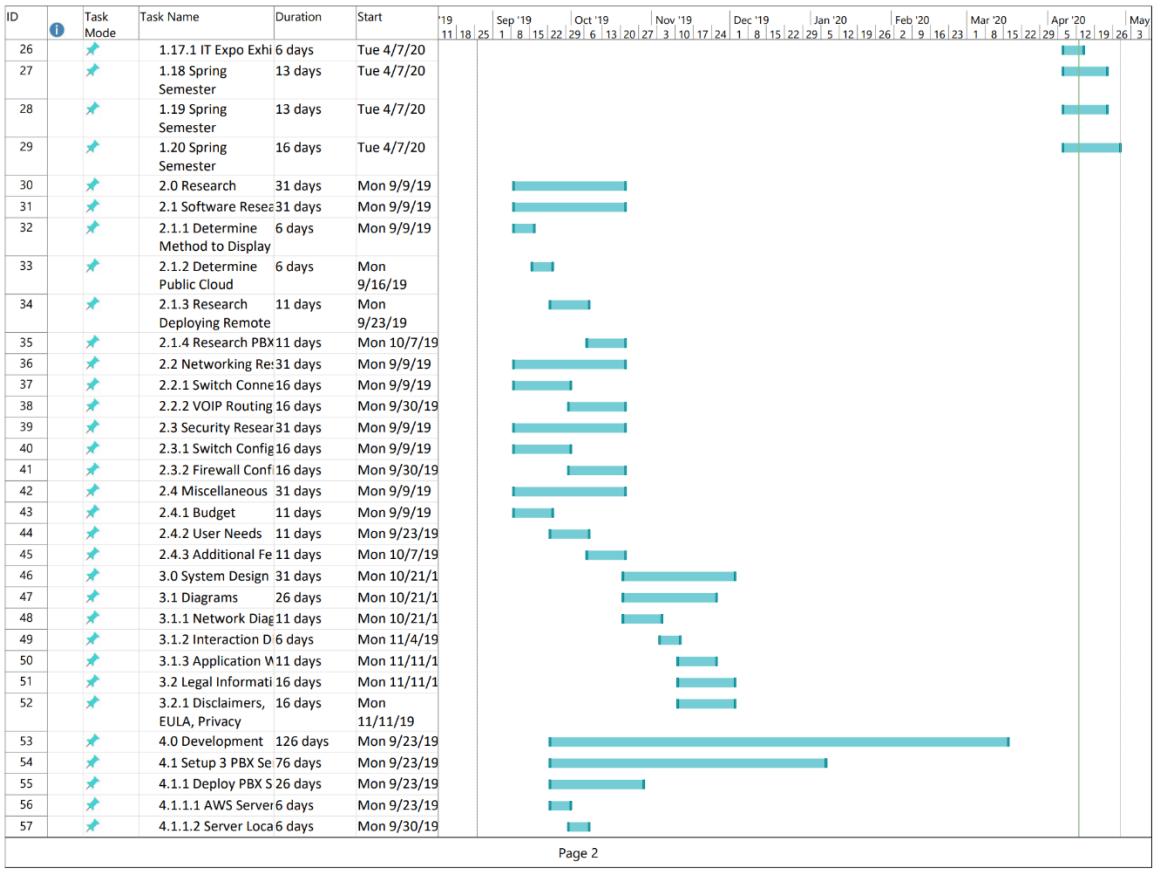
4.1.1.2 PBX Server Location	7	9/30/2019	10/7/2019
4.1.1.3 PBX Configuration	7	10/7/2019	10/14/2019
4.1.1.4 PBX Network Configuration	7	10/14/2019	10/21/2019
4.1.1.5 PBX Dynamic Routing Config	7	10/21/2019	10/28/2019
4.1.2 Deploy PBX Server CENTRAL	35	10/28/2019	12/2/2019
4.1.2.1 AWS PBX Server Deployment	7	10/28/2019	11/4/2019
4.1.2.2 AWS PBX Server Location	7	11/4/2019	11/11/2019
4.1.2.3 AWS PBX Configuration	7	11/11/2019	11/18/2019
4.1.2.4 AWS PBX Network Configuration	7	11/18/2019	11/25/2019
4.1.2.5 PBX Dynamic Routing Config	7	11/25/2019	12/2/2019
4.1.3 Deploy PBX Server EAST	35	12/2/2019	1/6/2020
4.1.3.1 PBX Server Deployment	7	12/2/2019	12/9/2019
4.1.3.2 PBX Server Location	7	12/9/2019	12/16/2019
4.1.3.3 PBX Configuration	7	12/16/2019	12/23/2019
4.1.3.4 PBX Network Configuration	7	12/23/2019	12/30/2019
4.1.3.5 PBX Dynamic Routing Config	7	12/30/2019	1/6/2020
4.2 Develop Admin Console	35	1/6/2020	2/10/202
4.2.1 Application Framework	14	1/6/2020	1/20/2020
4.2.2 Back End Data	14	1/20/2020	2/3/2020
4.2.3 Application Style and Branding	14	1/27/2020	2/10/2020
4.3 Secure Virtual Switches	35	2/10/2020	3/16/2020
5.0 Testing	77	1/6/2020	4/6/2020
5.1 Set up Clients	21	1/6/2020	1/27/2020
5.1.1 VOIP Clients	14	1/6/2020	1/20/2020
5.1.2 Mobile Clients	14	1/6/2020	1/20/2020
5.1.3 Administrators	14	1/13/2020	1/27/2020
5.2 Client Testing and Stress Test	42	1/27/2020	3/9/2020
5.2.1 Dynamic Route Testing	28	1/27/2020	2/24/2020
5.2.2 High Volume testing	14	2/24/2020	3/9/2020
5.3 Admin Console Testing	14	3/23/2020	4/6/2020

Figure 8a: WBS

Figure 8b: Gantt Chart, outlines the projected completion time for each aspect of the project broken down by deliverables.



ID	Task Mode	Task Name	Duration	Start
0		**Senior Design Proje**	**177 days**	**Mon 8/26/1**
1		1.0 Project Management &	177 days	Mon 8/26/19
2		1.1 Form Team & T	6 days	Mon 8/26/19
3		1.2 Ideas & Brainst	6 days	Mon 9/2/19
4		1.3 Assignment 0: Team Members &	6 days	Mon 9/2/19
5		1.3.1 Project Name	6 days	Mon 9/2/19
6		1.3.2 Project Logo	6 days	Mon 9/2/19
7		1.4 Assignment 1:	16 days	Mon 9/2/19
8		1.4.1 Project Apprc	16 days	Mon 9/2/19
9		1.4.2 Gantt Chart	16 days	Mon 9/2/19
10		1.4.3 Work Breakd	16 days	Mon 9/2/19
11		1.5 Assignment 2: Project Abstract	16 days	Mon 9/23/19
12		1.6 Assignment 3: Team Contract	6 days	Mon 10/7/19
13		1.7 Assignment 4: I	11 days	Mon 10/7/19
14		1.8 Assignment 5: Use Case Diagram	11 days	Mon 10/7/19
15		1.9 Assignment 6: I	11 days	Mon 10/21/1
16		1.10 Assignment 7: Final Fall	16 days	Mon 11/4/19
17		1.11 Fall Semester Oral Presentation	11 days	Mon 11/4/19
18		1.11.1 Presentation	11 days	Mon 11/4/19
19		1.12 Spring Semester	21 days	Mon 1/13/20
20		1.13 Spring Semester	6 days	Mon 2/10/20
21		1.14 Spring Semester	11 days	Mon 2/17/20
22		1.15 Spring Semester Assignment 4:	11 days	Mon 3/2/20
23		1.16 Spring Semester Oral	11 days	Mon 3/16/20
24		1.16.1 Presentatio	11 days	Mon 3/16/20
25		1.17 IT Expo	6 days	Tue 4/7/20



ID	Task Mode	Task Name	Duration	Start
26		1.17.1 IT Expo Exhi	6 days	Tue 4/7/20
27		1.18 Spring Semester	13 days	Tue 4/7/20
28		1.19 Spring Semester	13 days	Tue 4/7/20
29		1.20 Spring Semester	16 days	Tue 4/7/20
30		2.0 Research	31 days	Mon 9/9/19
31		2.1 Software Resea	31 days	Mon 9/9/19
32		2.1.1 Determine Method to Display	6 days	Mon 9/9/19
33		2.1.2 Determine Public Cloud	6 days	Mon 9/16/19
34		2.1.3 Research Deploying Remote	11 days	Mon 9/23/19
35		2.1.4 Research PBX	11 days	Mon 10/7/19
36		2.2 Networking Re	31 days	Mon 9/9/19
37		2.2.1 Switch Conne	16 days	Mon 9/9/19
38		2.2.2 VOIP Routing	16 days	Mon 9/30/19
39		2.3 Security Resear	31 days	Mon 9/9/19
40		2.3.1 Switch Config	16 days	Mon 9/9/19
41		2.3.2 Firewall Conf	16 days	Mon 9/30/19
42		2.4 Miscellaneous	31 days	Mon 9/9/19
43		2.4.1 Budget	11 days	Mon 9/9/19
44		2.4.2 User Needs	11 days	Mon 9/23/19
45		2.4.3 Additional Fe	11 days	Mon 10/7/19
46		3.0 System Design	31 days	Mon 10/21/1
47		3.1 Diagrams	26 days	Mon 10/21/1
48		3.1.1 Network Diag	11 days	Mon 10/21/1
49		3.1.2 Interaction D	6 days	Mon 11/4/19
50		3.1.3 Application V	11 days	Mon 11/11/1
51		3.2 Legal Informati	16 days	Mon 11/11/1
52		3.2.1 Disclaimers, EULA, Privacy	16 days	Mon 11/11/19
53		4.0 Development	126 days	Mon 9/23/19
54		4.1 Setup 3 PBX Se	76 days	Mon 9/23/19
55		4.1.1 Deploy PBX S	26 days	Mon 9/23/19
56		4.1.1.1 AWS Server	6 days	Mon 9/23/19
57		4.1.1.2 Server Loca	6 days	Mon 9/30/19

Page 2

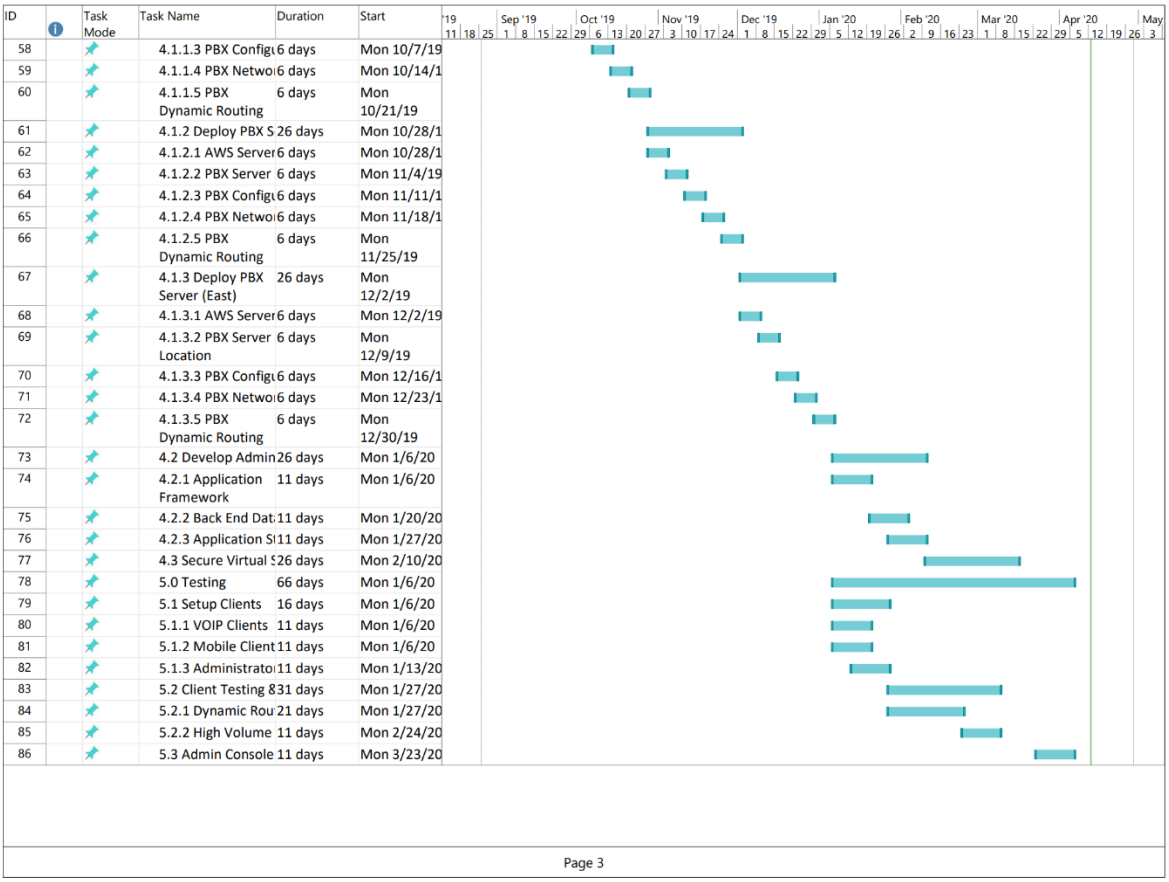


Figure 8b. Gantt Chart

Problems Encountered and Analysis of Problems Solved:

We have faced several issues as we build our project. Our first problem was installing and configuring FreePBX. We initially setup a CentOS VM inside of AWS. When we attempted to install and setup FreePBX within our CentOS box for some reason it would not install properly (we received a generic unable to install error). We tried a few troubleshooting steps recommended by google but none were successful. Eventually, we setup an Ubuntu box. From the Ubuntu box we were able to setup and configure FreePBX and access it from the web interface. Another issue we faced is bugginess within the internal calls inside of FreePBX.

We faced an issue with our VOIP trunk provider as we were building out our inbound routing solution which would be from an external caller calling into a user on the StabiliVOIP network. We weren't able to route the inbound calls the way that we had hoped. Since there are so many variables with that end of the infrastructure, we weren't able to send them to the best available server for that particular call. Inbound routing was done by our VOIP trunk and it had to be routing either in a sequential or round-robin manner. The way that this routing works is that if the first server is unreachable then it will move down the list, it will only follow the routing method if that first server is unreachable. Since this was the case, all calls would've had to hit the primary server first no matter what and then fail over to the others. Outbound calling was not affected by this issue since we had control on this end.

Future Recommendations:

Overall, as a team we were able to adjust well to all speed bumps and roadblocks. If we were to start the entire project over, we would begin with researching AWS. AWS was our management console across all our systems. Throughout the semester we spent a good amount of time researching AWS forums when we came across certain issues. Next, we would be sure to use backups from the start. We had to rebuild our PBX server 3 separate times. Had we created backups it would have saved a lot of time and stress. Lastly, we would implement security (ACLs, etc.) to our VMs on AWS right away.

If we had more time, we would have focused on completing the front-end admin UI. With the security issues we faced at the end of the first semester and beginning of the second, we were unable to complete the admin UI. Another idea we discussed was making our project global. If we had more time that is something we would have liked to accomplish.

We received great recommendations from several people throughout the semester. During our last meeting with Ryan, we explained how our project essentially broke and he suggested adding security to our project because we may have been hacked. He was spot on. Unfortunately, it took us rebuilding the project twice before we figured out the root cause which was a DDoS attack. Another recommendation we received was from Logan's previous manager. He suggested that we use Telynx as our VOIP trunk and it worked out great.

Conclusion

During the fall semester, we have learned to work effectively as a team, understanding VOIP calls and how they are routed within a network, how to dynamically route VOIP traffic, and the fundamentals of securing an application and website.

As a team we learned a lot from our Project and Senior Design as a whole. Some specific skills we learned were how to build and deploy a VOIP solution, the fundamentals of VOIP at an architectural level, network security, managing and deploying VMs within AWS, what dynamic routing is and how to implement it.

We finished a lot of the heavy lifting in the Spring semester. We deployed 2 additional PBX servers one in California and one in Northern Virginia, implemented an ACL to handle the security issues we faced, modified our PBX config to allow NAT, and we conducted testing for our project.

References

Thomas, Rob. “Installing FreePBX 14 on Ubuntu 18.04.” FreePBX, https://wiki.freepbx.org/display/FOP/Installing FreePBX 14 on Ubuntu 18.04.

Thomas, Rob. “Example Systemd Startup Script for FreePBX.” FreePBX, https://wiki.freepbx.org/display/HTGS/Example systemd startup script for FreePBX.

Team, Jive, and Casmin Wisner. “VoIP Network Requirements.” The Jive Blog, 8 Aug. 2019, https://blog.jive.com/voip-network-requirements/

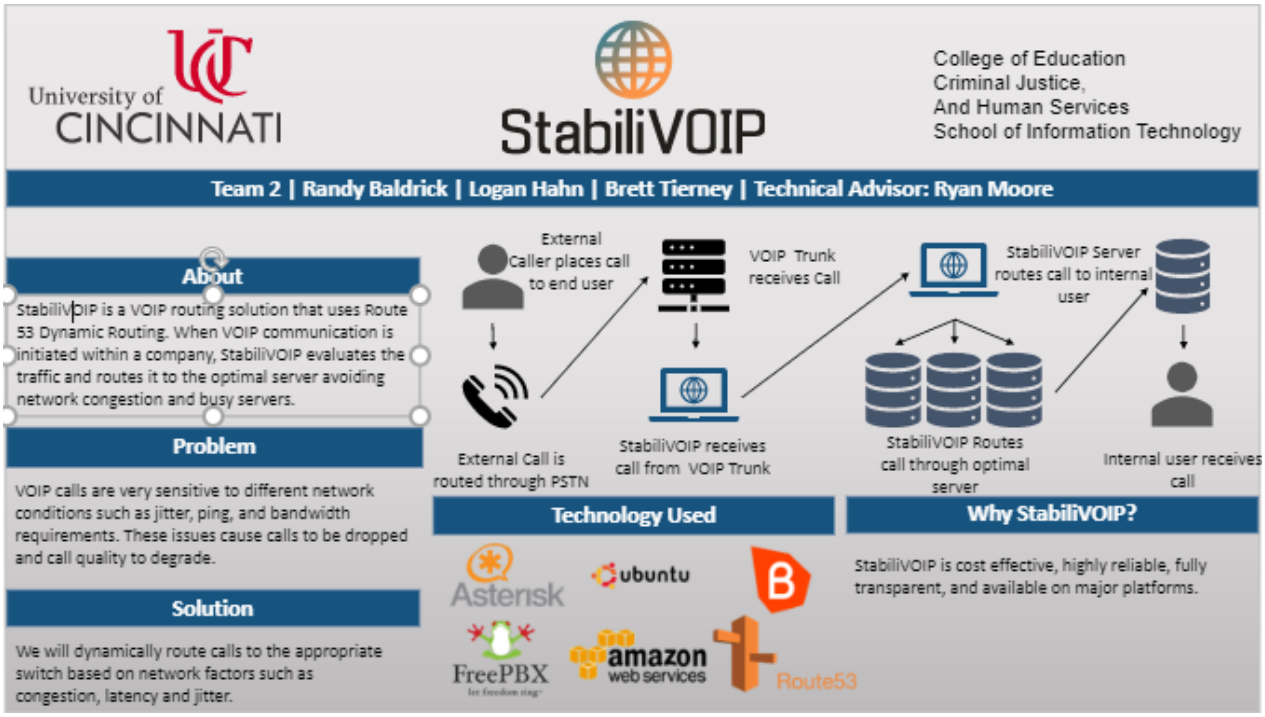
Bryant, Russell. Home - Asterisk Project - Asterisk Project Wiki, https://wiki.asterisk.org/wiki/display/AST/Home.

Rash, Wayne and Schwartz, Jeffrey. “Vonage Business Cloud Review.”28 June 2019, https://www.pcmag.com/reviews/vonage-business-cloud

“VoIP Market Stats – A Growing Industry.” *IronPaper*, 11 July 2016, https://www.ironpaper.com/webintel/articles/voip-market-stats/

http://ftp.airnet.ne.jp/mirror/mysql/Downloads/Connector-ODBC/5.3/

Appendix A -



Tech Expo Poster