MyHelpDesk

by

Devin Rindler, Samon Fayaz, and Tulashi Gautam

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

Devin Rindler	*4/26/20*
Devin Rindler	Date
Samon Fayaz	*4/26/20*
Samon Fayaz	Date
Tulashi Gautam	*4/26/20*
Tulashi Gautam	Date
Ryan Moore	*4/26/20*
Ryan Moore, Faculty Advisor	Date

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

April 2020

MyHelpDesk

Prepared by:

Devin Rindler, Samon Fayaz, Tulashi Gautam

Students of
University of Cincinnati
College of Education, Criminal Justice, and Human Services
School of Information Technology
cech.uc.edu

April 2020

TABLE OF CONTENTS

ABSTRACT……1

1 Problem Statement……..2
1.1 Introduction……..2
1.2 Problem……2
1.3 Solution……....3
1.4 Project Goals……..4
1.5 Overview……4

2 Discussion……...……5
2.1 Project Concept……..5
2.2 Design Objectives……..6
2.3 User Profile……..……7
2.4 Use Case Diagram…….……10

3 Project Management……11
3.1 Budget……..11
3.2 Project Schedule……...……...12

4 Technical Approach……17
4.1 Design Requirements……...17
4.2 Procedures……18
4.3 Network Overview…….…..18
4.4 Application Overview…….….19
4.5 Database Overview……..…19
4.6 Security Overview……20

5 Technical Diagrams…….……21
5.1 Application Architecture……...…….……21
5.2 Screenshots and Examples……..…..……20

TABLE OF CONTENTS

6 Testing……..…….25
6.1 Overview……..……25
6.2 Scope……25
6.3 Objective……..26
6.4 Logging Testing and Procedures…….26
6.5 Test Results…….……27

7 Development Issues……..……29
7.1 Problems Encountered…….……29

8 Future Recommendations…….…….……..……..30
8.1 What would you do if you had more time?....…….……30
8.2 What would you do if you had to do it all over again?.....…….……30
8.3 What suggestions have you had from others?.....31
8.4 What, if any, are your plans for this project?.....31

9 Conclusion…….…….…….32
9.1 Fall Semester 2019……..32
9.2 Spring Semester 2020……..33

REFERENCES……..……34

Appendix A…….……..……..……..……..……35

Appendix B…….…….…….……36

LIST OF ILLUSTRATIONS

TABLES

Table 1: End User Profile……8
Table 2: Administrative User Profile……...9
Table 3: Budget Breakdown……..…11
Table 4: Work Breakdown Structure…….13

FIGURES

Figure 1: Use Case Diagram……..10
Figure 2: Gantt Chart…….12
Figure 3: Gantt Chart Continued……13
Figure 4: Application Architecture..…….……21
Figure 5: SimpleSolutionsCenter Example.....22
Figure 6: SimpleSolutionsCenter Example cont.....23
Figure 7: Proof of Completion.....24
Figure 8: Test List Examples……..…28
Figure 9: Test Log Examples…….28

ABSTRACT

Zendesk states that 69% of support tickets were resolved with just “one touch”, and yet many tickets require resolution times of over 24 hours[1]. Incident Management, Change Management, and Service Request management are key elements to combat with an end user's daily frustration regarding problematic or unavailable business needs.

MyHelpDesk eliminates the need for an end user to reach out to Managed Services for assistance by providing a wide selection of tools to use, including, but not limited to: training videos, trusted software solutions, access to Statement of Work’s and Client/Company process documentation, and in-app scripting all in one convenient location. MyHelpDesk allows end users of any technological experience to easily operate this centralized application for all things troubleshooting, training, workflow, and support within their network scope.

1 PROBLEM STATEMENT

1.1 Introduction

Today's world is connected globally by the use of the internet and computers. From 2002 to 2016, the US Department of Labor found a 59% increase in the amount of jobs that require the use of technology[4]. While this number may be increasing as time goes on, the number of those who are technologically literate is not growing at the same rate. Everyday there are users who are unable to complete their work because of various computer issues, and even with a dedicated help desk these issues can be left in the backburner for more priority issues. In order to alleviate this problem, an application needs to be developed to automate these fixes and guide users through proper troubleshooting techniques.

1.2 Problem

Zendesk, a customer service software company, reports that nearly 69% of level one service desk issues are solved in one click[1]. Because of this, many of these tickets have an average response time of 24.2 hours[1]. Why, if the solution is so easy, does it take so long to resolve then?

There are many factors that can be the cause of this including but not limited to: scheduling conflicts, poor communication, or maybe the end user has poor technological knowledge. Even when users have training for basic computer skills and/or how to interact with the service desk properly, over time that knowledge will fade if they aren't using it every day. There is no easy and/or cost-effective way to keep users trained, and it's difficult to complete large tasks as a service desk technician if they simply outnumber you, and priority issues take precedent.

1.3 Solution

Many various Service Desk type applications already exist, but MyHelpDesk gathers the best parts of all those apps and puts them in one, convenient location. MyHelpDesk is a desktop-based application that aims to help the level one help desk environment by having a plethora of tools and easy training readily available to all end users at just one click. There will be an easy ticketing system built into the app for users to send in tickets or call the help desk if urgent, similar to applications like LiveAgent[2].

Similar in scope to InvGate's Service Desk[3], it will give all users the power to solve basic issues on their own whether by tool's built into the application itself, or by accessing the vast knowledge base included. What's MyHelpDesk application uses to separate itself from competition is that it is custom built for every client and contains any in-house applications and documents the end user could possibly need access to. This will also contain a database of custom help/training documentation and videos, as well as a collection of safe third-party tools available for download and use to further assist users. MyHelpDesk is aimed to help with both IT Service Providers as well as smaller businesses who can't afford external IT and have just a small personal team but allowing their users to solve problems all users can resolve. MyHelpDesk assists businesses in streamlining users job functions by resolving tickets before they are even submitted.

1.4 Project Goals

The goal of MyHelpDesk is to lessens the time spent between end users and service desk technicians for level one type tickets to boost productivity as well as provide an archive for training purposes and a repository of valuable and safe web applications to assist the end user.

As a group, we all had previous experience with help desk solutions and troubleshooting. Aiming to solve end users and managed services struggles, we are developing a centralized IT solution personalized for each client. With this we will be able to alleviate issues for Managed Service providers before they are even alerted of the issue as users will be able to take action towards their goals at all hours at the convenience of their device.

MyHelpDesk will be a web-based application that will have a user-friendly interface to help users of all ages and skill level, accomplished through modern accessibility techniques and an intuitive design structure.

1.5 Overview

The remainder of this final report outlines, in detail, how MyHelpDesk was developed. The report includes the following sections: discussion and design objectives, project budget, project timeline, technical approach, technical diagrams, testing procedures and results, problems encountered, and future recommendations.

2 Discussion

2.1 Project Concept

MyHelpDesk was inspired by Devin Rindler, Samon Fayaz, and Tulashi Gautam. Devin initially pitched the idea for a Help Desk Support type app through the class's discussion board and was reached out to by both Samon and Tulashi. Both Samon and Tulashi, also having worked at a service desk position before, began brainstorming various other ways the app could be helpful to users and companies alike based off of their experience.

After the first class, the team met up to discuss the various ideas they have come across and decided over the next week they would try and narrow down all the possibilities of the app and try and figure out what issues users struggled with most and what issues were easiest to solve. Finally, after comparing all ideas and trimming them down to the most effective, MyHelpDesk was ready to be developed.

2.2 Design Objectives

The features of MyHelpDesk will include:

- End users have the ability to resolve some issues themselves without having to interact with the Service Desk by preforming various operations to check and improve device health.
- Easily send tickets to the service desk through the application by going through some brief questions to help describe the issue

    <u>NOTE:</u> This design had to be changed to a general "contact us" section of the app that is ever-present along the top bar of the application. We believe this will be easier as users can briefly explain their issue instead of going through a questionnaire.

- A client custom video help library to help explain parts of the app itself, various training videos for onboarding or instructional purposes in various subject areas (how to install a printer, how to submit a support ticket, how to perform a disk cleanup, etc.)
- A computer up-time counter so that it is easy to see when the last successful reboot was performed on the device

    <u>NOTE:</u> This design had to be changed to a single-click function that displays last successful reboot as we were unable to configure a live uptime counter for the app.

- An area that will include helpful applications with safe links leading to them so that end users do not accidentally download any malicious programs. They will be categorized for things like virus scanners, screen sharing/TeamViewer type software, browsers, helpful tools, driver installation software, among other things.

• Contains troubleshooting area that allows a user to attempt to fix the issues themselves with the ability to check for updates, links to change password, and various "one click" fixes.

• Contains logging for administrator access so that it is easy to see what has been attempted through the app so far, so that if the issue is escalated, the technician working on the issue won't have to repeat steps.

> *NOTE: We had to stop development on this feature due to time constraints. Other areas of the project proved more challenging than initially though when creating the app.*

Together, these features will provide a powerful tool that puts power in the hands of end users to solve some issues on their own. It also saves time for service desk workers to focus on problems that need more attention, therefore saving time and money for all.

2.3 User Profile

Through our work we have determined that there will only be two types of people who will interact with MyHelpDesk: the End User (EU) and the Administrative User (AU). The differences between the two are minimal, and both users will interact with the application mostly in the same way. Further information on the different User Profiles can be found in *Table 1: End User Profile* and *Table 2: Administrative User Profile*.

Table 1: End User Profile

End User Profile
Application: MyHelpDesk
POTENTIAL USERS: - Businesses with low/no budget for internal IT staffing - Users who like to solve problems themselves - People with basic knowledge of computers and applications
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: This application will be geared towards individuals in the workforce who have used desktops/laptops and web applications before and know the basics of how to navigate them. High technical knowledge is not needed to operate MyHelpDesk.
EXPERIENCE WITH SIMILAR APPLICATIONS: - HP Support Assist - InvGate ServiceDesk - "Installation wizards" - YouTube or other video-based websites
TASK EXPERIENCE: - Navigating desktop or web applications and clicking through menu's - Knowledge of common tech words (IP, Hard vs Soft Reset, File share, etc.) - Ability to read direction carefully - Familiarity with troubleshooting is a plus, though not necessary
FREQUENCY OF USE: This product will come into use whenever a user needs to interact with the service desk, access the knowledge base for training or assistance, or needs support with a level-one type technology issue.
KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS: - Intuitive interface for easy navigation and use, no matter the skill level - Integration with any Tech Support/Service Desk already in place - Client specific training/knowledge base built-in - Automated resolution of every day computer issues

Table 2: Administrative User Profile

Administrative User Profile

Application:

MyHelpDesk

POTENTIAL USERS:

- IT team members assisting EUs with technical issues
- Users who wish to edit the Knowledge Base
- Users who wish to edit EU Information
- People with moderate/advanced knowledge of computers and applications

SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:

This application will be geared towards individuals in the workforce who have used desktops/laptops and web applications before and know the basics of how to navigate them. High technical knowledge is not needed to operate MyHelpDesk.

EXPERIENCE WITH SIMILAR APPLICATIONS:

- HP Support Assist
- InvGate ServiceDesk
- "Installation wizards"
- Live Agent

TASK EXPERIENCE:

- Navigating desktop or web applications and clicking through menu's
- Knowledge of common tech words (IP, Hard vs Soft Reset, File share, etc.)
- Ability to read logs
- General troubleshooting

FREQUENCY OF USE:

AUs will only need to interact with MyHelpDesk whenever EUs are unable to complete computer issues themselves, or if they wish to try and use the tool to fix an EUs issue. The only other times that an AU will use MyHelpDesk is when they are editing the Knowledge Base or EU information.

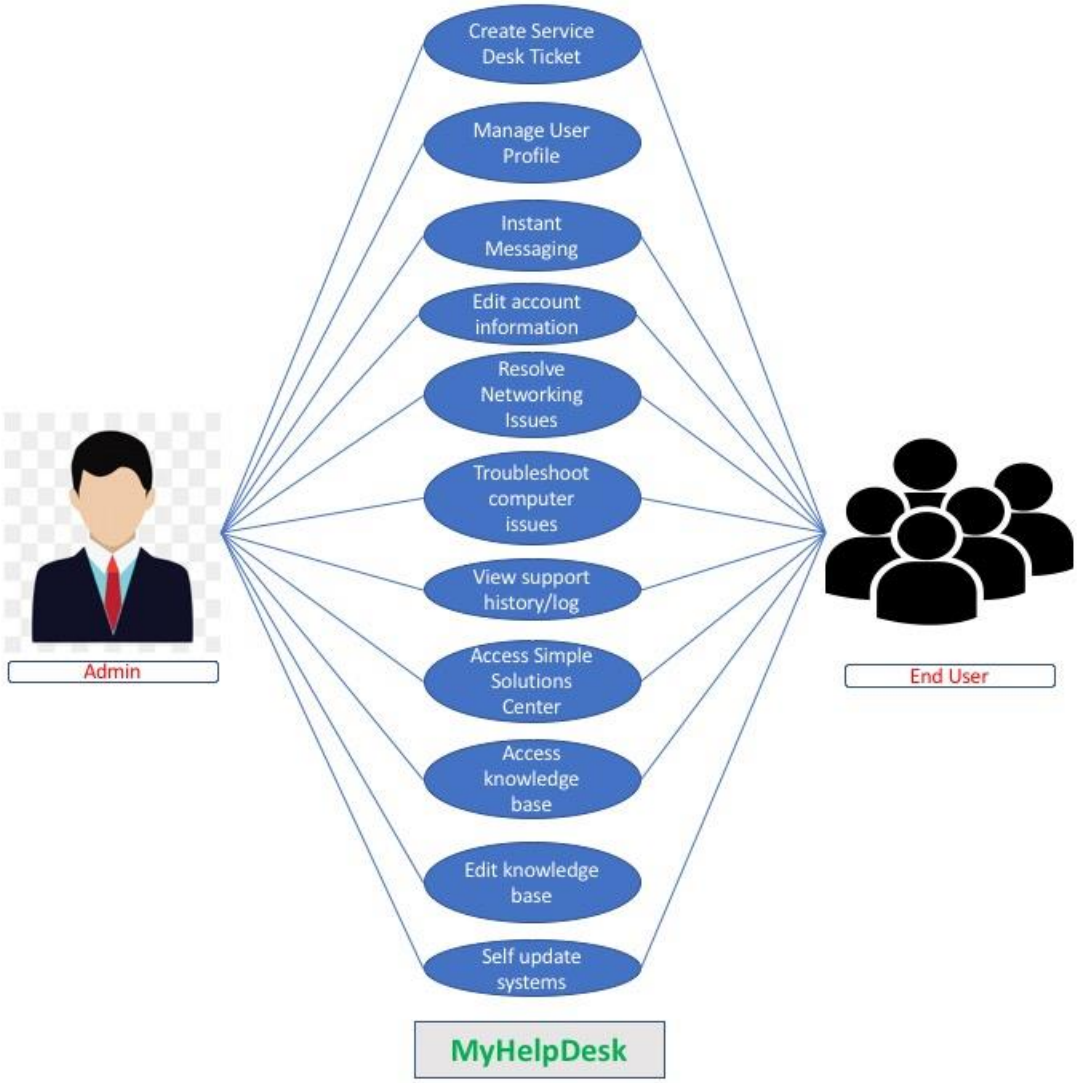
KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Intuitive interface for easy navigation and use, no matter the skill level
- Integration with any Tech Support/Service Desk already in place
- Logging of what has already been done with the application
- Automated resolution of every day computer issues

2.4 Use Case Diagram

In the illustration below, *Figure 1: Use Case Diagram* shows how both EUs and AUs will interact with the application. Both users will have full access to the applications knowledge base and solution center, as well as the various other support options of MyHelpDesk. The main differentiator is that AUs have the ability to edit the core structure of the app, such as the contents of the Knowledge Base or the User Profiles.

Figure 1: Use Case Diagram



3 Project Management

3.1 Budget

Thankfully, the tools we used to develop our app were either open-source or free to students. We were also lucky enough to have a server at our disposal to host the virtual environment in, further saving us costs. Therefore, the following *Table 3: Budget Breakdown* contains estimated numbers and values based on estimates for running the servers and programs in a corporate environment, as well as an estimation of labor and equipment costs. At the end, the entirety of the project came out to an estimated $9,982

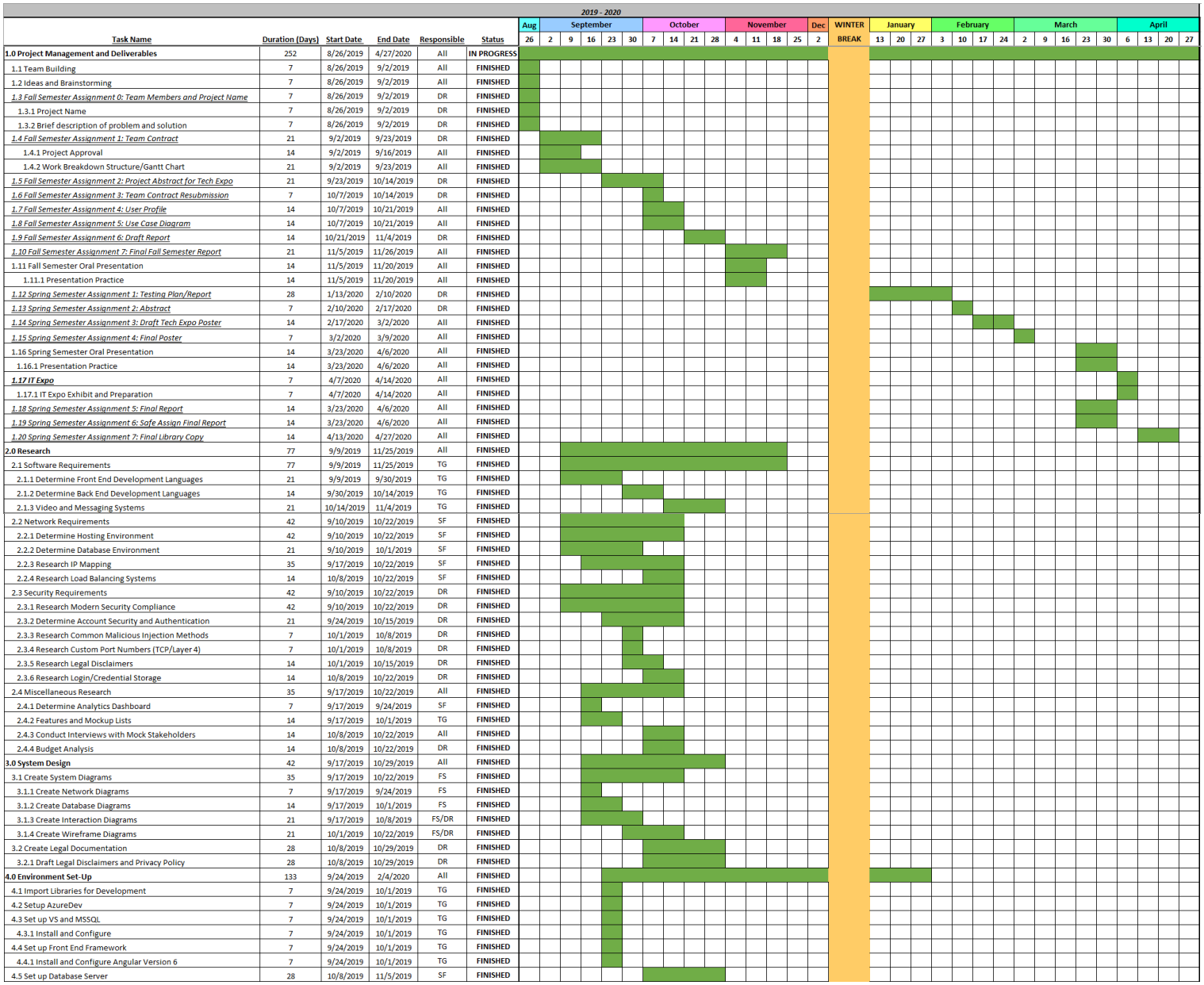
Table 3: Budget Breakdown

Project Costs	
Labor ($20/Hr for 15Hr/Week for 30 Weeks)	$9,000
Hosting Server (Dell PowerEdge T420)	$443
Visual Studio Professional (1-year subscription)	$539
Total Costs	**$9,982**

3.2 Project Schedule

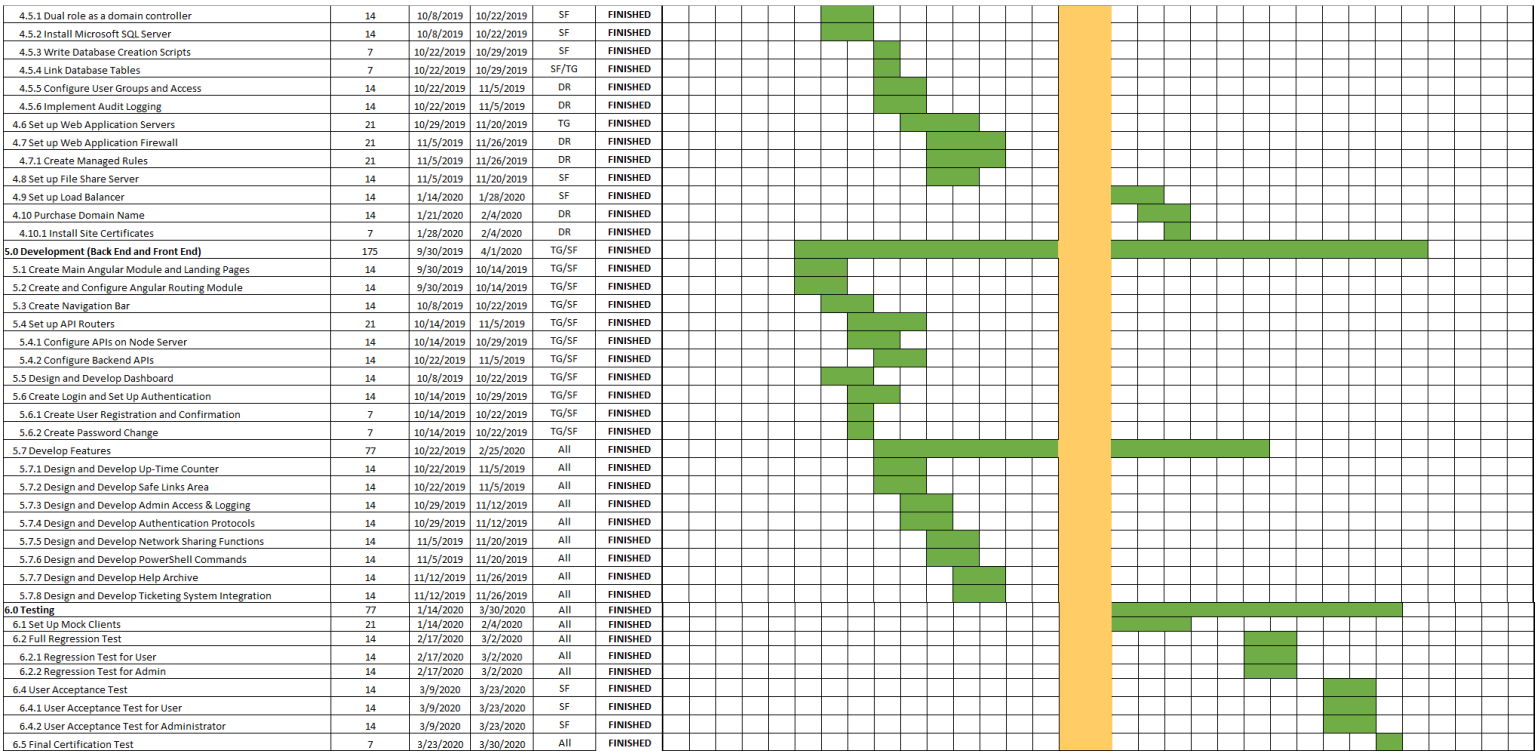
Figure 2: Gantt Chart, *Figure 3: Gantt Chart Continued*, and *Table 4: Work Breakdown Structure* show the predicted timeline and division of work for completion the MyHelpDesk application.

Figure 2: Gantt Chart



Task Name	Duration (Days)	Start Date	End Date	Responsible	Status
1.0 Project Management and Deliverables	252	8/26/2019	4/27/2020	All	IN PROGRESS
1.1 Team Building	7	8/26/2019	9/2/2019	All	FINISHED
1.2 Ideas and Brainstorming	7	8/26/2019	9/2/2019	All	FINISHED
1.3 Fall Semester Assignment 0: Team Members and Project Name	7	8/26/2019	9/2/2019	DR	FINISHED
1.3.1 Project Name	7	8/26/2019	9/2/2019	DR	FINISHED
1.3.2 Brief description of problem and solution	7	8/26/2019	9/2/2019	DR	FINISHED
1.4 Fall Semester Assignment 1: Team Contract	21	9/2/2019	9/23/2019	DR	FINISHED
1.4.1 Project Approval	14	9/2/2019	9/16/2019	All	FINISHED
1.4.2 Work Breakdown Structure/Gantt Chart	21	9/2/2019	9/23/2019	All	FINISHED
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	21	9/23/2019	10/14/2019	DR	FINISHED
1.6 Fall Semester Assignment 3: Team Contract Resubmission	7	10/7/2019	10/14/2019	DR	FINISHED
1.7 Fall Semester Assignment 4: User Profile	14	10/7/2019	10/21/2019	All	FINISHED
1.8 Fall Semester Assignment 5: Use Case Diagram	14	10/7/2019	10/21/2019	All	FINISHED
1.9 Fall Semester Assignment 6: Draft Report	14	10/21/2019	11/4/2019	DR	FINISHED
1.10 Fall Semester Assignment 7: Final Fall Semester Report	21	11/5/2019	11/26/2019	All	FINISHED
1.11 Fall Semester Oral Presentation	14	11/5/2019	11/20/2019	All	FINISHED
1.11.1 Presentation Practice	14	11/5/2019	11/20/2019	All	FINISHED
1.12 Spring Semester Assignment 1: Testing Plan/Report	28	1/13/2020	2/10/2020	DR	FINISHED
1.13 Spring Semester Assignment 2: Abstract	7	2/10/2020	2/17/2020	DR	FINISHED
1.14 Spring Semester Assignment 3: Draft Tech Expo Poster	14	2/17/2020	3/2/2020	All	FINISHED
1.15 Spring Semester Assignment 4: Final Poster	7	3/2/2020	3/9/2020	All	FINISHED
1.16 Spring Semester Oral Presentation	14	3/23/2020	4/6/2020	All	FINISHED
1.16.1 Presentation Practice	14	3/23/2020	4/6/2020	All	FINISHED
1.17 IT Expo	7	4/7/2020	4/14/2020	All	FINISHED
1.17.1 IT Expo Exhibit and Preparation	7	4/7/2020	4/14/2020	All	FINISHED
1.18 Spring Semester Assignment 5: Final Report	14	3/23/2020	4/6/2020	All	FINISHED
1.19 Spring Semester Assignment 6: Safe Assign Final Report	14	3/23/2020	4/6/2020	All	FINISHED
1.20 Spring Semester Assignment 7: Final Library Copy	14	4/13/2020	4/27/2020	All	FINISHED
2.0 Research	77	9/9/2019	11/25/2019	All	FINISHED
2.1 Software Requirements	77	9/9/2019	11/25/2019	TG	FINISHED
2.1.1 Determine Front End Development Languages	21	9/9/2019	9/30/2019	TG	FINISHED
2.1.2 Determine Back End Development Languages	14	9/30/2019	10/14/2019	TG	FINISHED
2.1.3 Video and Messaging Systems	21	10/14/2019	11/4/2019	TG	FINISHED
2.2 Network Requirements	42	9/10/2019	10/22/2019	SF	FINISHED
2.2.1 Determine Hosting Environment	42	9/10/2019	10/22/2019	SF	FINISHED
2.2.2 Determine Database Environment	21	9/10/2019	10/1/2019	SF	FINISHED
2.2.3 Research IP Mapping	35	9/17/2019	10/22/2019	SF	FINISHED
2.2.4 Research Load Balancing Systems	14	10/8/2019	10/22/2019	SF	FINISHED
2.3 Security Requirements	42	9/10/2019	10/22/2019	DR	FINISHED
2.3.1 Research Modern Security Compliance	42	9/10/2019	10/22/2019	DR	FINISHED
2.3.2 Determine Account Security and Authentication	21	9/24/2019	10/15/2019	DR	FINISHED
2.3.3 Research Common Malicious Injection Methods	7	10/1/2019	10/8/2019	DR	FINISHED
2.3.4 Research Custom Port Numbers (TCP/Layer 4)	7	10/1/2019	10/8/2019	DR	FINISHED
2.3.5 Research Legal Disclaimers	14	10/1/2019	10/15/2019	DR	FINISHED
2.3.6 Research Login/Credential Storage	14	10/8/2019	10/22/2019	DR	FINISHED
2.4 Miscellaneous Research	35	9/17/2019	10/22/2019	All	FINISHED
2.4.1 Determine Analytics Dashboard	7	9/17/2019	9/24/2019	SF	FINISHED
2.4.2 Features and Mockup Lists	14	9/17/2019	10/1/2019	TG	FINISHED
2.4.3 Conduct Interviews with Mock Stakeholders	14	10/8/2019	10/22/2019	All	FINISHED
2.4.4 Budget Analysis	14	10/8/2019	10/22/2019	DR	FINISHED
3.0 System Design	42	9/17/2019	10/29/2019	All	FINISHED
3.1 Create System Diagrams	35	9/17/2019	10/22/2019	FS	FINISHED
3.1.1 Create Network Diagrams	7	9/17/2019	9/24/2019	FS	FINISHED
3.1.2 Create Database Diagrams	14	9/17/2019	10/1/2019	FS	FINISHED
3.1.3 Create Interaction Diagrams	21	9/17/2019	10/8/2019	FS/DR	FINISHED
3.1.4 Create Wireframe Diagrams	21	10/1/2019	10/22/2019	FS/DR	FINISHED
3.2 Create Legal Documentation	28	10/8/2019	10/29/2019	DR	FINISHED
3.2.1 Draft Legal Disclaimers and Privacy Policy	28	10/8/2019	10/29/2019	DR	FINISHED
4.0 Environment Set-Up	133	9/24/2019	2/4/2020	All	FINISHED
4.1 Import Libraries for Development	7	9/24/2019	10/1/2019	TG	FINISHED
4.2 Setup AzureDev	7	9/24/2019	10/1/2019	TG	FINISHED
4.3 Set up VS and MSSQL	7	9/24/2019	10/1/2019	TG	FINISHED
4.3.1 Install and Configure	7	9/24/2019	10/1/2019	TG	FINISHED
4.4 Set up Front End Framework	7	9/24/2019	10/1/2019	TG	FINISHED
4.4.1 Install and Configure Angular Version 6	7	9/24/2019	10/1/2019	TG	FINISHED
4.5 Set up Database Server	28	10/8/2019	11/5/2019	SF	FINISHED

Figure 3: Gantt Chart Continued



4.5.1 Dual role as a domain controller	14	10/8/2019	10/22/2019	SF	FINISHED
4.5.2 Install Microsoft SQL Server	14	10/8/2019	10/22/2019	SF	FINISHED
4.5.3 Write Database Creation Scripts	7	10/22/2019	10/29/2019	SF	FINISHED
4.5.4 Link Database Tables	7	10/22/2019	10/29/2019	SF/TG	FINISHED
4.5.5 Configure User Groups and Access	14	10/22/2019	11/5/2019	DR	FINISHED
4.5.6 Implement Audit Logging	14	10/22/2019	11/5/2019	DR	FINISHED
4.6 Set up Web Application Servers	21	10/29/2019	11/20/2019	TG	FINISHED
4.7 Set up Web Application Firewall	21	11/5/2019	11/26/2019	DR	FINISHED
4.7.1 Create Managed Rules	21	11/5/2019	11/26/2019	DR	FINISHED
4.8 Set up File Share Server	14	11/5/2019	11/20/2019	SF	FINISHED
4.9 Set up Load Balancer	14	1/14/2020	1/28/2020	SF	FINISHED
4.10 Purchase Domain Name	14	1/21/2020	2/4/2020	DR	FINISHED
4.10.1 Install Site Certificates	7	1/28/2020	2/4/2020	DR	FINISHED
5.0 Development (Back End and Front End)	175	9/30/2019	4/1/2020	TG/SF	FINISHED
5.1 Create Main Angular Module and Landing Pages	14	9/30/2019	10/14/2019	TG/SF	FINISHED
5.2 Create and Configure Angular Routing Module	14	9/30/2019	10/14/2019	TG/SF	FINISHED
5.3 Create Navigation Bar	14	10/8/2019	10/22/2019	TG/SF	FINISHED
5.4 Set up API Routers	21	10/14/2019	11/5/2019	TG/SF	FINISHED
5.4.1 Configure APIs on Node Server	14	10/14/2019	10/29/2019	TG/SF	FINISHED
5.4.2 Configure Backend APIs	14	10/22/2019	11/5/2019	TG/SF	FINISHED
5.5 Design and Develop Dashboard	14	10/8/2019	10/22/2019	TG/SF	FINISHED
5.6 Create Login and Set Up Authentication	14	10/14/2019	10/29/2019	TG/SF	FINISHED
5.6.1 Create User Registration and Confirmation	7	10/14/2019	10/22/2019	TG/SF	FINISHED
5.6.2 Create Password Change	7	10/14/2019	10/22/2019	TG/SF	FINISHED
5.7 Develop Features	77	10/22/2019	2/25/2020	All	FINISHED
5.7.1 Design and Develop Up-Time Counter	14	10/22/2019	11/5/2019	All	FINISHED
5.7.2 Design and Develop Safe Links Area	14	10/22/2019	11/5/2019	All	FINISHED
5.7.3 Design and Develop Admin Access & Logging	14	10/29/2019	11/12/2019	All	FINISHED
5.7.4 Design and Develop Authentication Protocols	14	10/29/2019	11/12/2019	All	FINISHED
5.7.5 Design and Develop Network Sharing Functions	14	11/5/2019	11/20/2019	All	FINISHED
5.7.6 Design and Develop PowerShell Commands	14	11/5/2019	11/20/2019	All	FINISHED
5.7.7 Design and Develop Help Archive	14	11/12/2019	11/26/2019	All	FINISHED
5.7.8 Design and Develop Ticketing System Integration	14	11/12/2019	11/26/2019	All	FINISHED
6.0 Testing	77	1/14/2020	3/30/2020	All	FINISHED
6.1 Set Up Mock Clients	21	1/14/2020	2/4/2020	All	FINISHED
6.2 Full Regression Test	14	2/17/2020	3/2/2020	All	FINISHED
6.2.1 Regression Test for User	14	2/17/2020	3/2/2020	All	FINISHED
6.2.2 Regression Test for Admin	14	2/17/2020	3/2/2020	All	FINISHED
6.4 User Acceptance Test	14	3/9/2020	3/23/2020	SF	FINISHED
6.4.1 User Acceptance Test for User	14	3/9/2020	3/23/2020	SF	FINISHED
6.4.2 User Acceptance Test for Administrator	14	3/9/2020	3/23/2020	SF	FINISHED
6.5 Final Certification Test	7	3/23/2020	3/30/2020	All	FINISHED

Table 4: Work Breakdown Structure

Task Name	Duration (Days)	Start Date	End Date
1. 0 Project Management and Deliverables	252	8/20/19	4/15/19
1.1 Team Building	7	8/20/19	8/27/19
1.2 Ideas and Brainstorming	7	8/27/19	9/3/19
1.3 Fall Semester Assignment 0: Team Members & Project Name	7	8/27/19	9/3/19
1.3.1 Project Name	7	8/27/19	9/3/19
1.3.2 Project Logo and Branding	7	8/27/19	9/3/19
1.4 Fall Semester Assignment 1: Team Contract	21	9/3/19	9/24/19
1.4.1 Project Approval	14	9/3/19	9/17/19
1.4.2 Work Breakdown Structure/Gantt Chart	21	9/3/19	9/24/19
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	21	9/24/19	10/15/19
1.6 Fall Semester Assignment 3: Team Contract Resubmission	7	10/8/19	10/15/19
1.7 Fall Semester Assignment 4: User Profile	14	10/8/19	10/22/19
1.7 Fall Semester Assignment 5: Use Case Diagram	14	10/8/19	10/22/19
1.9 Fall Semester Assignment 6: Draft Report	14	10/22/19	11/5/19
1.10 Fall Semester Assignment 7: Final Fall Semester Report	21	11/5/19	11/26/19
1.11 Fall Semester Oral Presentation	14	11/5/19	11/20/19
1.11.1 Presentation Practice	14	11/5/19	11/20/19
1.12 Spring Semester Assignment 1: Testing Plan/Report	28	1/14/20	2/11/20

1.13 Spring Semester Assignment 2: Abstract	7	2/11/20	2/19/20
1.14 Spring Semester Assignment 3: Draft Tech Expo Poster	14	2/19/20	3/4/20
1.15 Spring Semester Assignment 4: Final Poster	14	3/4/20	3/19/20
1.16 Spring Semester Oral Presentation	14	3/19/20	4/1/20
1.16.1 Presentation Practice	14	3/19/20	4/1/20
1.17 IT Expo	7	4/1/20	4/9/20
1.17.1 IT Expo Exhibit and Preparation	7	4/1/20	4/9/20
1.19 Spring Semester Assignment 5: Final Report	14	4/1/20	4/15/20
1.20 Spring Semester Assignment 6: Safe Assign Final Report	14	4/1/20	4/15/20
1.20 Spring Semester Assignment 7: Final Library Copy	14	4/15/20	4/29/20
2.0 Research	42	9/10/19	10/22/19
2.1 Software Requirements	35	9/10/19	10/15/19
2.1.1 Determine Front End Development Languages	7	9/10/19	10/1/19
2.1.2 Determine Back End Development Languages	14	9/10/19	10/1/19
2.1.3 Video and Messaging Systems	21	10/1/19	10/15/19
2.2 Network Requirements	42	9/10/19	10/22/19
2.2.1 Determine Hosting Environment	42	9/10/19	10/22/19
2.2.2 Determine Database Environment	21	9/10/19	10/1/19
2.2.3 Research IP Mapping	35	9/17/19	10/22/19
2.2.4 Research Load Balancing Systems	14	10/8/19	10/22/19
2.3 Security Requirements	42	9/10/19	10/22/19
2.3.1 Research Modern Security Compliance	42	9/10/19	10/22/19
2.3.2 Determine Account Security and Authentication	21	9/24/19	10/15/19
2.3.3 Research Common Malicious Injection Methods	7	10/1/19	10/8/19
2.3.4 Research Custom Port Numbers (TCP/Layer 4)	7	10/1/19	10/8/19
2.3.5 Research Legal Disclaimers	14	10/1/19	10/15/19
2.3.6 Research Login/Credential Storage	14	10/8/19	10/22/19
2.4 Miscellaneous Research	35	9/17/19	10/22/19
2.4.1 Determine Analytics Dashboard	7	9/17/19	9/24/19
2.4.2 Features and Mockup Lists	14	9/17/19	10/1/19
2.4.3 Conduct Interviews with Mock Stakeholders	14	10/8/19	10/22/19
2.4.4 Budget Analysis	14	10/8/19	10/22/19
3.0 System Design	42	9/17/19	10/29/19
3.1 Create System Diagrams	35	9/17/19	10/22/19
3.1.1 Create Network Diagrams	7	9/17/19	9/24/19
3.1.2 Create Database Diagrams	14	9/17/19	10/1/19
3.1.3 Create Interaction Diagrams	21	9/17/19	10/8/19
3.1.4 Create Wireframe Diagrams	21	10/1/19	10/22/19
3.2 Create Legal Documentation	28	10/8/19	10/29/19

3.2.1 Draft Legal Disclaimers and Privacy Policy	21	10/8/19	10/29/19
4.0 Environment Set-Up	133	9/24/19	2/4/20
4.1 Import Libraries for Development	7	9/24/19	10/1/19
4.2 Setup GitHub	7	9/24/19	10/1/19
4.3 Set up VS and MSSQL	7	9/24/19	10/1/19
4.3.1 Install and Configure	7	9/24/19	10/1/19
4.4 Set up Front End Framework	7	9/24/19	10/1/19
4.4.1 Install and Configure Angular Version 6	7	9/24/19	10/1/19
4.5 Set up Database Server	28	10/8/19	11/5/19
4.5.1 Dual role as a domain controller	14	10/8/19	10/22/19
4.5.2 Install Microsoft SQL Server	14	10/8/19	10/22/19
4.5.3 Write Database Creation Scripts	7	10/22/19	10/29/19
4.5.4 Link Database Tables	7	10/22/19	10/29/19
4.5.5 Configure User Groups and Access	14	10/22/19	11/5/19
4.5.6 Implement Audit Logging	14	10/22/19	11/5/19
4.6 Set up Web Application Servers	21	10/29/19	11/20/19
4.7 Set up Web Application Firewall	21	11/5/19	11/26/19
4.7.1 Create Managed Rules	21	11/5/19	11/26/19
4.8 Set up File Share Server	14	11/5/19	11/20/19
4.9 Set up Load Balancer	14	1/14/20	1/28/20
4.10 Purchase Domain Name	14	1/21/20	2/4/20
4.10.1 Install Site Certificates	7	1/28/20	2/4/20
5.0 Development (Back End and Front End)	175	10/1/19	4/1/20
5.1 Create Main Angular Module and Landing Pages	14	10/1/19	10/15/19
5.2 Create and Configure Angular Routing Module	14	10/1/19	10/15/19
5.3 Create Navigation Bar	14	10/8/19	10/22/19
5.4 Set up API Routers	21	10/15/19	11/5/19
5.4.1 Configure APIs on Node Server	14	10/15/19	10/29/19
5.4.2 Configure Backend APIs	14	10/22/19	11/5/19
5.5 Design and Develop Dashboard	14	10/8/19	10/22/19
5.6 Create Login and Set Up Authentication	14	10/15/19	10/29/19
5.6.1 Create User Registration and Confirmation	7	10/15/19	10/22/19
5.6.2 Create Forgot Password	7	10/15/19	10/22/19
5.6.3 Set Up Redis to handle sessions	14	10/15/19	10/29/19
5.7 Develop Features	77	10/22/19	2/25/20
5.7.1 Design and Develop Up-Time Counter	14	10/22/19	11/5/19
5.7.2 Design and Develop Safe Links Area	14	10/22/19	11/5/19
5.7.3 Design and Develop Admin Access & Logging	14	10/29/19	11/12/19
5.7.4 Design and Develop Authentication Protocols	14	10/29/19	11/12/19

5.7.5 Design and Develop Network Sharing Functions	14	11/5/19	11/20/19
5.7.6 Design and Develop PowerShell Commands	14	11/5/19	11/20/19
5.7.7 Design and Develop Help Archive	14	11/12/19	11/26/19
5.7.8 Design and Develop Ticketing System Integration	14	11/12/19	11/26/19
5.7.9 Design and Develop Messaging System	21	1/14/20	2/4/20
5.7.10 Design and Develop Screen Capture System	28	1/21/20	2/11/20
5.8 Release Bugfixes	28	2/25/20	4/1/20
5.8.1 Patches for Alpha (Regression) Testing	14	2/25/20	3/11/20
5.8.2 Patches for Beta (User Acceptance) Testing	14	3/19/20	4/1/20
6.0 Testing	77	1/14/20	4/1/20
6.1 Set Up Mock Clients	21	1/14/20	2/4/20
6.1.1 Set Up Desktop Clients	14	1/14/20	1/28/20
6.2 Full Regression Test	14	2/19/20	3/4/20
6.2.1 Regression Test for Client	14	2/19/20	3/4/20
6.2.2 Regression Test for User	14	2/19/20	3/4/20
6.2.3 Regression Test for User	14	2/19/20	3/4/20
6.2.4 Regression Test for Administrator	14	2/19/20	3/4/20
6.3 User Acceptance Test	14	3/11/20	3/25/20
6.3.1 User Acceptance Test for Client	14	3/11/20	3/25/20
6.3.2 User Acceptance Test for User	14	3/11/20	3/25/20
6.3.3 User Acceptance Test for User	14	3/11/20	3/25/20
6.3.4 User Acceptance Test for Administrator	14	3/11/20	3/25/20
6.4 Final Certification Test	7	3/25/20	4/1/20

4 Technical Approach

4.1 Design Requirements

One of the main goals of MyHelpDesk is allowing the application to be accessible for users of all ages, and of all skill levels. The World Wide Web Consortium (W3C) provides several accessibility guidelines to help make this possible, including the Web Content Accessibility Guidelines (WCAG), User Agent Accessibility Guidelines (UAAG), and the Author Tool Accessibility Guidelines (ATAG)[5]. MyHelpDesk bases its design features around these guidelines to allow the app to be intuitive for use and easily read and understood.

The app has a minimalist design, in that most pages of the app only have a few various options for users to interact with to keep it simple. All options are clearly labeled, and most pages provide a brief description of the options listed.

There are four main pages of the MyHelpDesk app:

- MyHelpDesk Home – this is the startup/home page of the app. It will contain sections for interacting with the Help Desk.
- KnowledgeBase –A SQL based library for finding training videos, wiki-type walkthroughs, other self-help type solutions. Includes a specific area for how to operate the MyHelpDesk application
- SimpleSolutionsCenter –where many of the “one-click” solutions are stored. Will have a collection of scripts, downloadable tools, and more to help solve issues with minimal user interaction
- Account settings –contains options for changing your email, first name, last name, or password.

4.2 Procedures

MyHelpDesk incorporates many methodologies from both the Web Application Development Process procedure and the Agile Web Application Development procedure to help keep the project and application organized and on track. This means researching our projected audience and what their needs are, researching third-party vendors for the purchasing of things like SSL certificates, and continual development and testing to ensure the app is functioning properly and effectively [6].

4.3 Network Overview

MyHelpDesk is integrated using Azure for a database management and storage, while the backbone of the application is hosted on a server local to whatever Windows environment it is deployed to. Folders and drives have been integrated into this database to allow the users to pulled directly from a corporate trusted source. This application can be web hosted or server hosted via IIS or other means based on the client's environment and needs. Currently, a physical server is set up to host a virtual environment with ESXi for one Windows Server 2016 server and one Windows 10 workstation for testing and building purposes.

4.4 Application Overview

The application was built using C# with Visual Studio on the back end, and using HTML, CSS, and JavaScript for the front end. The application is mainly powered through scripts of varying languages (PowerShell, VB, etc.) stored on the Azure database, which are called upon in the application to be downloaded and then ran on the end users' machine. With C# we can provide this one-click access to all a user will need to complete daily functions, even when needing to deal with PowerShell or other extensions required to access the specific media.

4.5 Database Overview

The application uses Microsoft Structured Query Language (SQL) on a Microsoft SQL Server Database for data storage. This was picked based on the project being developed largely in a Microsoft based environment for the ease of use and connectivity. As we are based with C# and using Visual studio to base our application we are using another Microsoft product as it integrates with the rest of the components of our application. The database is backed up locally as well as on the Azure portal on a scheduled basis. With these Microsoft product integrations, we can retrieve and update, save, and retrieve data the user may require within the application.

4.6 Security Overview

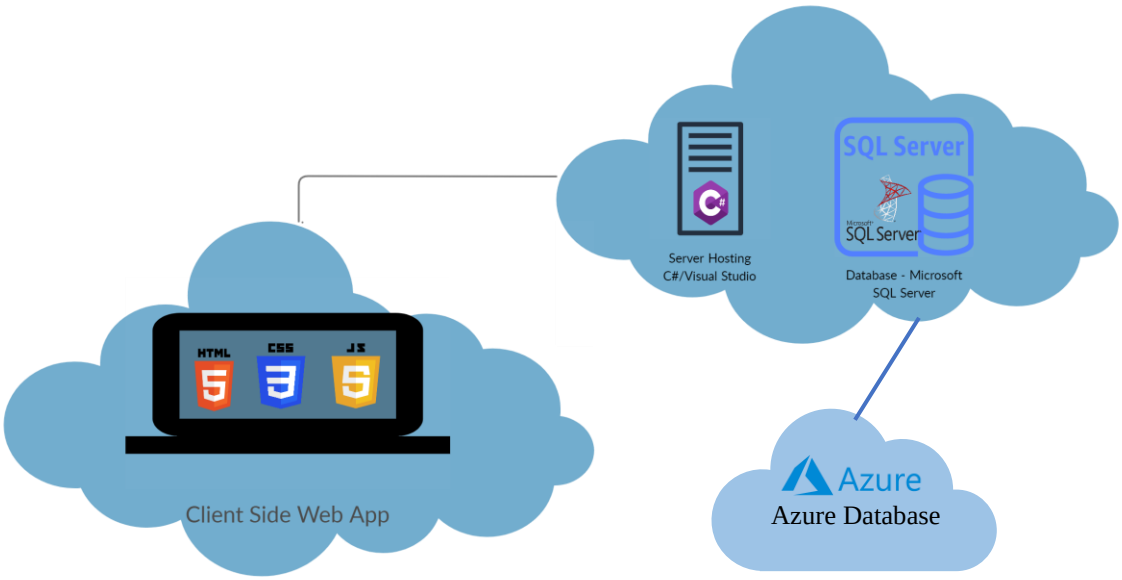
This application was secured using proper communication protocols and built to have limited access to the Internet. MyHelpDesk sits locally in the corporate environment and does not have features available to allow it to be publicly hosted. It is meant only to be an in-house application, and therefore does not have many of the risks associated with common website applications. Secure downloads were also in mind when creating the application, and MyHelpDesk encourages users to only download the applications and programs found within to lower confusion for end users so that they know which programs are company approved and which are not.

5 Technical Diagrams

5.1 Application Architecture

In *Figure 4: Application Architecture*, it shows how all facets of the application are connected. The figure demonstrates how C# and Microsoft SQL Server are used on the backend to perform all the core actions of the application. These actions are determined by user input on the front end from the web application. The user requests an action like viewing a file or running a script, and the backend servers will retrieve accurate results from the SQL server or Database according to the actions performed by the user.

Figure 4: Application Architecture



5.2 Screenshots and Examples

In *Figures 5: SimpleSolutionsCenter Example*, *Figure 6: SimpleSolutionsCenter Example cont.*, and *Figure 7: Proof of Completion*, there is an example of how an end user would use MyHelpDesk to try and connect to a predefined corporate printer.

Figure 5: SimpleSolutionsCenter Example

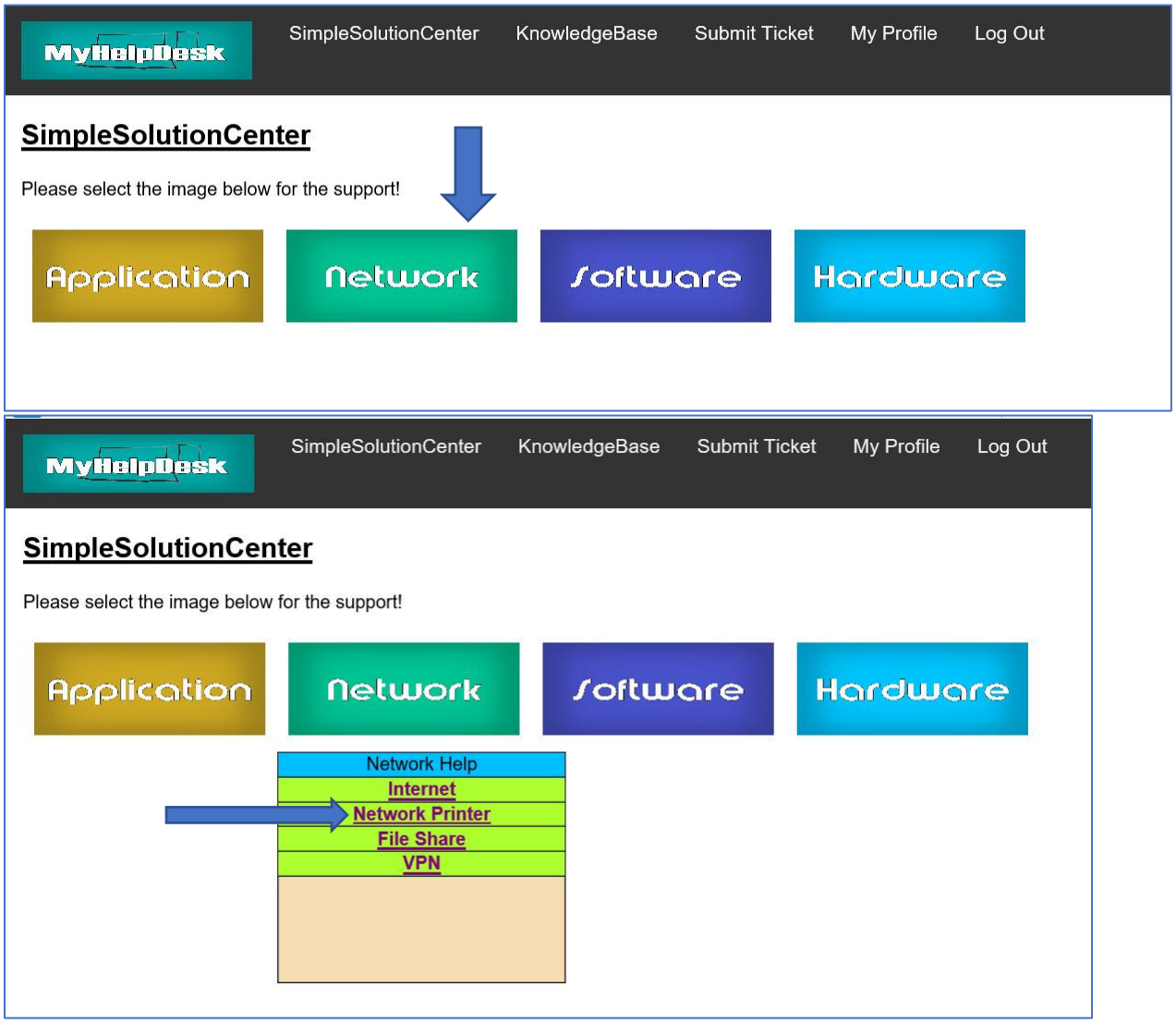
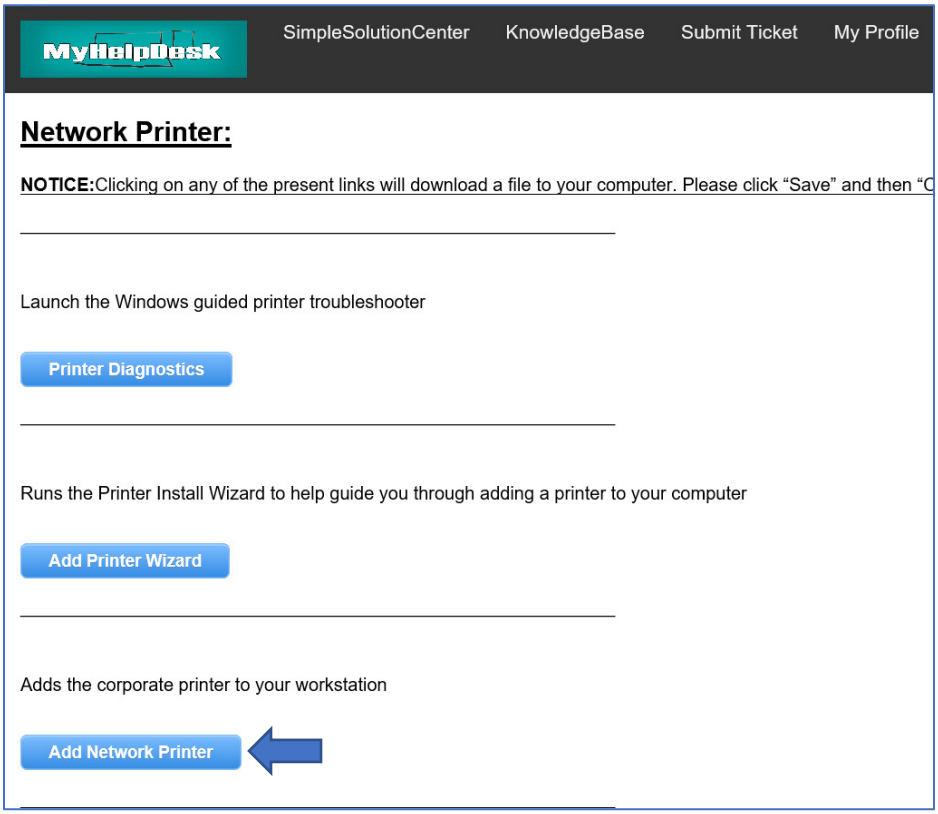


Figure 6: SimpleSolutionsCenter Example cont.



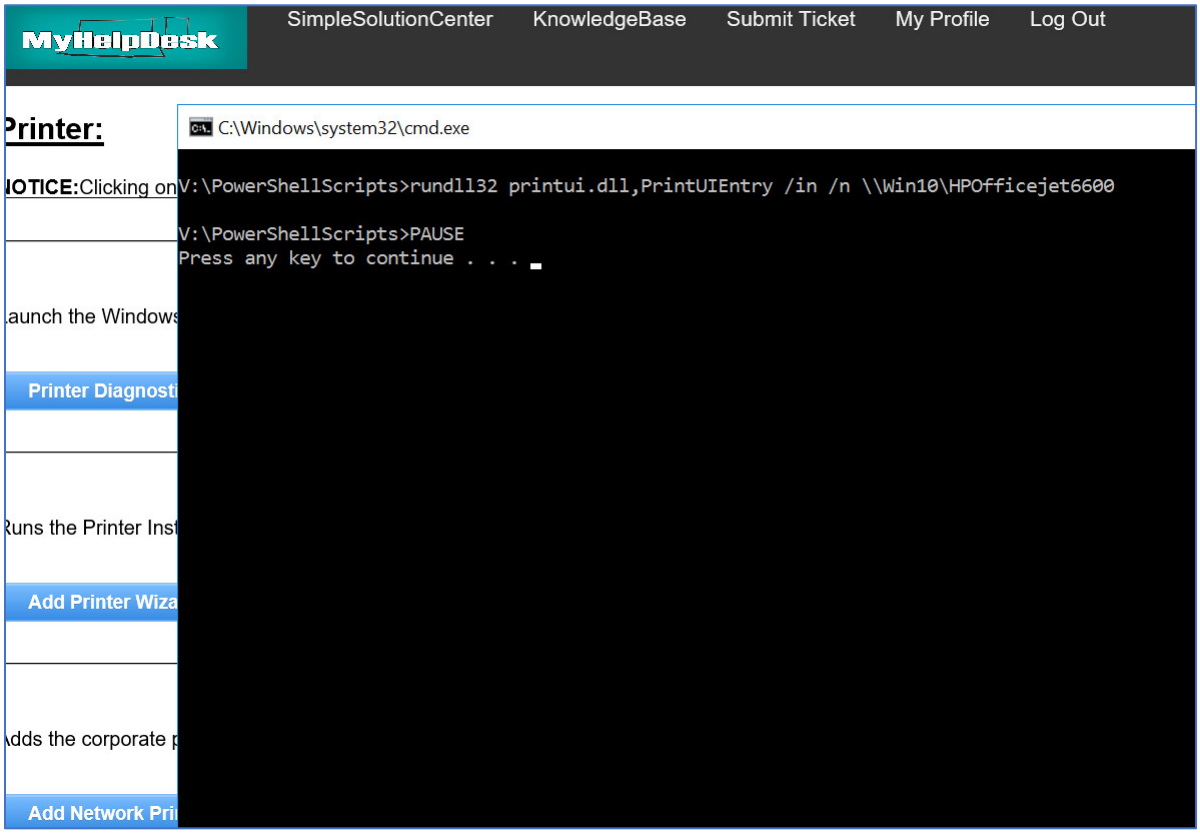
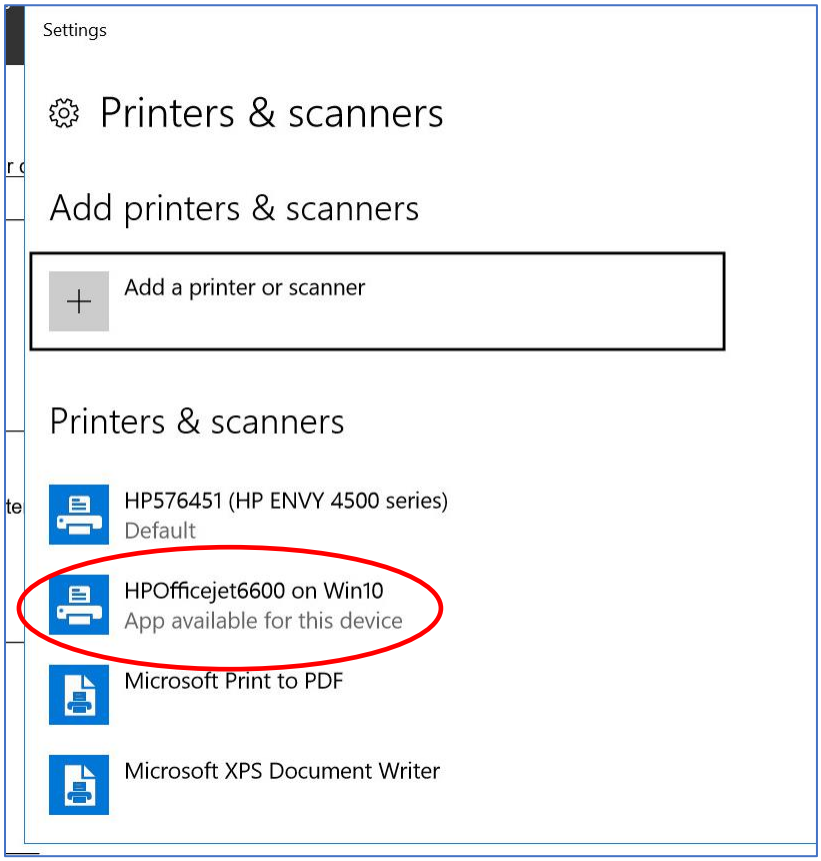


Figure 7: Proof of Completion



6 Testing

6.1 Overview

This document goes over how testing throughout the development of MyHelpDesk was performed in detail. The testing is to ensure application stability and that the main functions of MyHelpDesk, such as single-click solutions or content search, work as expected. Each function of MyHelpDesk was tested as thoroughly as possible, by both high- and low-level technologically experienced end users.

The testing structure used during development of the MyHelpDesk application followed an Agile methodology. The Agile method allows continual testing and improvement of the application and allows flexibility throughout development because of its cyclic nature. The core of MyHelpDesk is hosted in Azure, and so it is heavily reliant upon code; namely SQL, C#, and PowerShell. This means that Agile development was a must so that errors can be fixed as soon as they appear – eliminating most of the rework that is normally done at the end of development.

6.2 Scope

The Test Cases created go over every core function of the MyHelpDesk application. As new features were added, so too were new Test Cases created. Continual testing throughout the beginning stages of development was conducted by the developers, but as the app neared completion it was performed by end users who would typically have use for an app like MyHelpDesk. Testing done by both experienced and inexperienced users was required to ensure the app is accessible to all skill levels.

6.3 Objective

The primary goals of the testing conducted are as follows:

- Each core function of the app must be tested by the team to return concise and accurate results as expected of the function.
- Each core function of the app must be tested by at least two external users of different skill levels.
    - These tests must show that the users are able to easily operate and navigate the app, making note of any difficulties the user has.
- Any failed/incomplete tests will go back into development, having a page created in the Testing section of the Team OneNote explaining the issue as well as possible fixes.
- All testing must pass before presentation at the IT Tech Expo.

6.4 Logging Testing and Procedures

Throughout testing, any issue that is found will be immediately documented in its own page of the “Testing” section of the project OneNote notebook. Here, notes can be included as well as screen shots to help document the issues that arise.

Test Cases and Test Results are also stored within OneNote, as with a majority of the project notes, ideas, and visuals. The flexibility of OneNote allows us to easily track the progress of testing and issue remediation.

Each team member is expected to regularly test the application as new features are implemented and provide feedback to all team members. Whenever external testing is run, the team member running the testing will take notes on how the user interacts with the app, what path a user takes to perform a function of the app, how long it takes for a user to perform that function, if the user successfully performs that function or not (pass/fail), if the app runs as expected, and any other information the team member finds useful. At this time, no external testing has been scheduled.

6.5 Test Results

With Agile, testing is continuous throughout the entire production cycle. Waterfall projects usually have "exit criteria" to go from one step to the next, whereas with Agile the project is continually improved upon. With Agile testing for MyHelpDesk, we would develop a function of the app, test the function and report bugs, fix those bugs, and test again until the function runs as expected. To count the project as complete, every test must run successfully.

Figure 8: Test List Examples contains a screenshot from our Testing section of the project OneNote displaying a list of various test cases for different functions of MyHelpDesk. As stated, test conditions were added to the list as they were added to the site.

Figure 8: Test List Examples

List of Testing

Friday, February 7, 2020 6:03 PM

Item#	Title	Outcome	Test Runner	NOTES
1	Menu limited to login/signup until user signs in	Pass	Tulsi	
2	Account Creation will not work unless an email address is used	Pass	Tulsi	
3	Menu items populate after user successfully logs in	Pass	Samon	Site kept defaulting to homepage instead of sign in, resulting in a view of the entire site instead of just login/logout. Has been fixed.
4	Can successfully switch from login to signup page without being logged in	Pass	Tulsi	
5	Update password from Profile page	Pass	Samon	
6	Update username from Profile page	Pass	Samon	
7	Update First/Last name from Profile page	Pass	Samon	
8	SimpleSolutionCenter Categories link to appropriate page	Pass	Devin	Category links are way too large, need reduced for easier reading
9	SimpleSolutionCenter disk clean utility launches PS command	Fail	Devin	Instead of running command, links to Hardware Help page
10	Consistent "Contact Us" works for every page	Pass	Tulsi	
11	Log Out tab signs out user	Pass	Tulsi	

Figure 9: Test Log Example shows a closer look of the notes taken when a test does not go as expected. Screenshots are encouraged along with a brief explanation of the issue at hand, but mainly they contain brief information of the steps performed to solve the issues.

Figure 9: Test Log Examples

Test #12

Tuesday, March 31, 2020 10:42 AM

Download and Run SimpleSolutionCenter Scripts
Clicking to download the script results in a file with blank file type.
Tested on multiple browsers without success.
Attempting to download from cloud-share instead of MHD repository
Setting href = file:///V:/PowerShellScript/EnterScriptNameHere/
Solution did not work on chrome.
Solution did work on Internet Explorer, and downloads as correct file type.

7 Development Issues

7.1 Problems Encountered

While working on the project, as with all projects, we came across our fair share of issues. Initially, the first issue we ran into was that we all worked remotely from each other and so one of the only times we met was during class time. This led to some communication and accountability issues. For example, the tree of us all had slightly different ideas of what the app was supposed to look like and how it was to function at first. Going forward, we knew that constant communication and checking in on each other is the most effective way for the project to stay on task.

Other issues encountered were trying to figure out how to host the web app. We were unsure of what would not only be easiest to work on, but also would be easy for transportation to the IT Expo and be easily accessible for all to work on. We had access to several physical servers and had many different online options available to us, and ultimately went with hosting our own virtual environment to host the app in. This was because there is no cost to host on our own software, and we were able to do remote work through TeamViewer on the server.

The biggest issue we came across was the core functionality of our app, the "click-to-run" solutions we needed. We were having struggles getting the scripts we created to run at the click of a button, as PowerShell was not interacting with C# in the way we expected. Eventually, we did find a work-around for this issue, however. We hosted the scripts in an external file share and now the buttons in the application will download the scripts and run at completion. The only issue with this is that we were only able to get this functionality to be consistent in Internet Explorer.

8 Future Recommendations

8.1 What would you do if you had to do it all over again?

Working throughout this project, we learned several things after the fact. For starters, we would have built our authentication around Azure instead of trying to develop our own form. This would have saved much time as we could have focused our effort elsewhere. Similarly, we would have built the project as a desktop application instead of a website application. This is because there would have been less limitations on how we could implement PowerShell and other scripting into our application and would not have had as much issue with the click-to-run solutions we put in place.

8.2 What would you do if you had more time?

With more time to work on the project, we could have implemented several of the features that we had to end up getting rid of. We would have had more time to research and implement a live-chat function running inside the application; this would have greatly increased the usability of the app as end users could have communicated with a support team member at any step of the troubleshooting process. With more time, there also would have been time to further flesh out the accessibility of the application. More time to add extra help sections within the app, descriptions on mouse-hover for all options in app, and a more friendly and flowing interface.

8.3 What suggestions have you had from others

Throughout demonstrations and presenting to the class, the suggestions we have received have mainly been around app design. At first, our application had too much information about what the app could do and how it could work on the home page, and we were told it looked too much like either an advertisement or a 1990's AOL homepage. We've taken this information and trimmed our app down heavily to have a more basic and simplistic design, and we know that this has greatly increased usability and understanding of the application.

8.4 What, if any, are your plans for the project?

There are no current plans for the project, aside from adding in the features that were originally left out due to time restraints.

9 Conclusion

9.1 Fall Semester 2019

Throughout this first semester of working on this project, many lessons have been learned. The biggest lesson would have to be that when everyone is working remotely, scheduling and communication are integral to completing tasks as expected and on time. At the beginning of the project, we all had different ideas of what we wanted the application to be and how it was to look. We had to spend some extra time meeting with each other and sharing ideas before we could finally settle on a final design idea, which ended up being better than anything any of us originally thought of.

A majority of this semester was focused on planning what features the app would have and drafting what it would look like and how it would function. Not much technical work has been done aside from figuring out which platform we would use to build our webserver and determining the technical features of the application. All of us have learned more about hosting with Azure DevOps and have looked into various PowerShell commands for automating whatever level-one tech support issues we could.

We are looking forward to the Spring when we can begin building this project in full. With all of the groundwork we have laid down this Fall, we are expecting an easy Spring semester of building the web server and designing the user interface and functionality.

9.2 Spring Semester 2020

Spring Semester 2020 had brought along lots of hardships and new challenges. We spent the entirety of the semester building our product from the ground up and learning the limits of our skill sets, along with learning new skills along the way to assist where we could on the project. A large portion of this semester was spent trying to figure out the click to run solution we wanted to put in place. It ended up causing us more issues than expected and a large amount of time was spent trying to get it functional. Something small that we improved upon is that many of the scripts after running would give no indication if they were successful or even if they ran, so we managed to figure out for all scripting types (.bat, .ps1, .vbs) how to provide some kind of verification to users that the script had run and prompted user input to continue on.

At the end, the application doesn't have the exact look we had envisioned from the start. The flow from one page to another required a bit more HTML & CSS work than any of us had time for between work, school, and the pandemic. The app has a much simpler appearance now and looks like it is older, but it still functions quite well.

All in all, we learned a lot about how to run a project and how much work goes into it. We have received a massive amount of knowledge about working remotely and running a project coherently from different cities. This has been a fantastic learning experience for us, and we are happy with the outcome we have produced given the circumstances that were present throughout this year.

REFERENCES

Source [1] – "The Zendesk Benchmark". https://d16cvnquvjw7pr.cloudfront.net/resources/whitepapers/Zendesk_WP_benchmark.pdf. 2013.

Source [2] – "LiveAgent Features". https://www.liveagent.com/features/. 2019.

Source [3] – "InvGate Service Desk". https://www.invgate.com/service-desk/. 2019.

Source [4] – "Technology is dramatically invading nearly all US jobs, even lower skilled occupations". https://www.cnbc.com/2017/11/15/technology-is-dramatically-invading-nearly-all-us-jobs-even-lower-skilled-occupations.html. November, 11th, 2017.

Source [5] – "Web Content Accessibility Guidelines (WCAG) Overview". https://www.w3.org/WAI/standards-guidelines/wcag/. June 5th, 2018.

Source [6] – Kohan, Bernard. "Guide to Web-Application Development". https://www.comentum.com/guide-to-web-application-development.html. 2019.

Source [7] – Guru99. "Web Application Testing: 8 Step Guide to Website Testing". https://www.guru99.com/web-application-testing.html.

Source [8] – Admin. "The True Cost of Not Providing Employee Training". https://www.shiftelearning.com/blog/the-true-cost-of-not-providing-employee-training. April 19th, 2018.

Appendix A: IT Expo Poster

College of Education, Criminal Justice, and Human Services
School of Information Technology
Advisor: Ryan Moore

MyHelpDesk

Team 39: Devin Rindler, Samon Fayaz, Tulsi Gautam

University of CINCINNATI

Problem

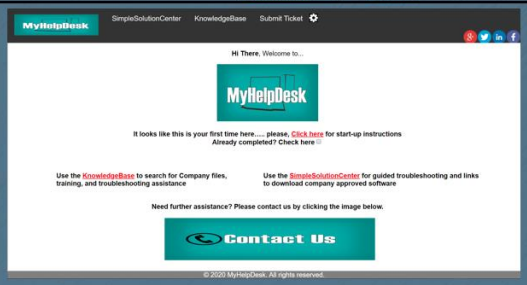
69% of support tickets are resolved in "one touch", and yet many have resolution times of 24+ hours

Most end users would solve these issues themselves, but they either lack troubleshooting skills or are technically illiterate.

Solution

Single-click solution to printer management, file shares, troubleshooting, and more

All in one location for accessing company files, submitting support tickets, training, and handling level-one support issues



What Is It?

MyHelpDesk completes all your application and helpdesk needs. End users learn how to navigate and operate their computers while learning how to use this intuitive app. Contained within are resources that allow end users to solve a plethora of level-one support issues such as adding a network printer, joining a file share, or cleaning up a full disk drive – guiding the user every step of the way. MyHelpDesk gives users a safe and consistent way to access:

- **Process Documentation**
- **Support Tickets**
- **Continual Training**
- **Company Files**
- **Scripting**
- **Trusted Software**

Technology

Visual Studio · **C#** · **Azure** · **MS SQL** · **CSS3** · **JavaScript** · **HTML5** · **PowerShell**

Appendix B: Source Code

ADDMHDFileShare.ps1

```
$connectTestResult = Test-NetConnection -ComputerName
mhdfs01.file.core.windows.net -Port 445
if ($connectTestResult.TcpTestSucceeded) {
    # Save the password so the drive will persist on reboot
    cmd.exe /C "cmdkey /add:`"mhdfs01.file.core.windows.net`"
/user:`"Azure\mhdfs01`"
/pass:`"hh99CR76sYhoNemx98VQH1nN7CvV1+WzXhgZcB5O9uPG4nVpGcaRsiBhtaj6XQYi71qnbp5
NBZZ1++1u4UTEWQ`""
    # Mount the drive
    New-PSDrive -Name V -PSProvider FileSystem -Root
"\\mhdfs01.file.core.windows.net\mhdfs01"-Persist
} else {
    Write-Error -Message "Unable to reach the Azure storage account via port
445. Check to make sure your organization or ISP is not blocking port 445, or
use Azure P2S VPN, Azure S2S VPN, or Express Route to tunnel SMB traffic over a
different port."
}
```

AddPrinterWizard

```
#Runs Add Printer Wizard
rundll32 printui.dll,PrintUIEntry /ii /f C:\Windows\INF\ntprint.inf
```

AudioDiagnostics

```
msdt.exe /id AudioPlaybackDiagnostic
```

Get Computer Info

```
Get-ComputerInfo
Write-Output " "
Write-Output "Operatoin complete. Please exit when finished."
Start-Sleep -Seconds 100
```

AddXDrive

```
net use X: \\WIN2016\Clients
Write-Output "Operation complete. Please exit when finished."
Start-Sleep -Seconds 100
```

DiskDefrag

```
defrag C: /U /V
```

dxdiag

```
dxdiag
```

WindowsUpdateDiagnostics

```
msdt.exe /id WindowsUpdateDiagnostic
```

SoundSettings

```
control mmsys.cpl sounds
```

SFCScan

```
sfc /scannow
```

DiskCleanup

```
#### Variables ####

    $objShell = New-Object -ComObject Shell.Application
    $objFolder = $objShell.Namespace(0xA)
    $temp = get-ChildItem "env:\TEMP"
    $temp2 = $temp.value
    $WinTemp = "c:\Windows\Temp\*"


#Remove temp files located in "C:\Users\USERNAME\AppData\Local\Temp"
    write-Host "Removing Junk files in $temp2." -ForegroundColor Magenta
    Remove-Item -Recurse  "$temp2\*" -Force -Verbose

#Empty Recycle Bin # http://demonictalkingskull.com/2010/06/empty-users-
recycle-bin-with-powershell-and-gpo/
    write-Host "Emptying Recycle Bin." -ForegroundColor Cyan
    $objFolder.items() | %{ remove-item $_.path -Recurse -Confirm:$false}

#Remove Windows Temp Directory
    write-Host "Removing Junk files in $WinTemp." -ForegroundColor Green
    Remove-Item -Recurse $WinTemp -Force

#Running Disk Clean up Tool
    write-Host "Running Windows disk Clean up Tool" -ForegroundColor Cyan
    cleanmgr /sagerun:1 | out-Null

    $([char]7)
    Sleep 1
    $([char]7)
    Sleep 1

    write-Host "Finished Cleanup Task " -ForegroundColor Yellow
##### End of the Script #####
```

DeleteOST

```
$users = Get-ChildItem c:\users


foreach ($user in $users){

$folder = "C:\users\" + $user +"\AppData\Local\Microsoft\Outlook"
$folderpath = test-path -Path $folder

if($folderpath)
{

Get-ChildItem $folder -filter *.ost | where-object {($_.LastWriteTime -gt (Get-
Date).AddDays(-14)) -and ($_.Length /1GB -gt 1)} | remove-item
Write-Output "Deleted OST file for $user"
}

else{

Write-Output "OST file doesn't exist or meet criteria for $user"

}
}
```

ViewNetworkInfo

```
ipconfig /flushdns
ipconfig /all
Write-Output "--------------------"
Write-Output "<<Operation complete. Please close when finished. Automatic
closeout after 100 seconds>>"
Start-Sleep -Second 100
```

```
Googletracert
tracert google.com
Write-Output "           "
Write-Output "Trace route completed. Please exit when finished"
Start-Sleep -Seconds 100

InternalTraceRoute
tracert 192.168.50.1
Write-Output "Please exit when you are finished"
Start-Sleep -Seconds 100

NetworkDiagnostics
msdt.exe /id NetworkDiagnosticsWeb


AudioDiagnositcs
msdt.exe /id AudioPlaybackDiagnostic
```