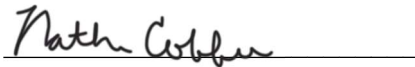


OverIT Final Report

University of Cincinnati
College of Education, Criminal Justice and Human Services
School of Information Technology

Nathan Coffee, Abhishek Goel, Dallas Blasek, John Stichweh, Takuro Fukuda



Nathan Coffee

4/22/2021

Date



John Stichweh

4/22/2021

Date



Takuro Fukuda

4/22/2021

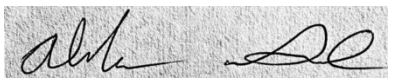
Date



Dallas Blasek

4/22/2021

Date



Abhishek Goel

4/22/2021

Date



Dyllon Dekok, Faculty Advisor

4/22/2021

Date

Contents

Project Summary	6
Problem Statement	6
Solution.....	6
Project Source	6
Project Objectives/Goals	7
Project Scope.....	7
Quick Project Timeline	8
Technologies Used.....	9
Technical Architecture Diagram	9
Network Diagram.....	10
Database Diagram	10
Use Cases	11
Use Case Diagram	16
Testing Plan	16
Overview.....	16
Methodology	16
Scope.....	17
Unit Tests	17
Ticket Request Case	17
Database Persistence	17
User Acceptance.....	17
Objectives.....	17
Test Logs and Procedures.....	17
Testing Review.....	18
User Acceptance Testing Review.....	18
Change Management Plan	19
Budget	19
Overview.....	19
Azure	19
Google Firebase Authentication	20
Problems Encountered & Analysis of Problems Solved.....	20
Conclusion.....	20
Fall Lesson Learned	20
References.....	22

List of Illustrations

Project Timeline	8
Network Diagram	10
Database Diagram	10
Use Cases - Elderly Customer	11
Use Cases - Younger Customer	12
Use Cases - IT Contractor	13
Use Cases - Application Administration/Developer	14
Use Cases - Security Administrator	15
Use Case Diagram	16
Test Logs & Procedures	18
Budget/ROI Chart	20

Abstract

The purpose of OverIT is to supply IT technical assistance to non-commercial customers without requiring a subscription while also supplying work to undergraduate and recently graduated IT students. Our group developed a website connected to a SQL database backend that will allow a user to post IT problems they have encountered and receive help in real time either in person or over a voice chat from a volunteer IT professional. This peer-to-peer system is mutually beneficial, as the customer gets IT help, and the volunteer receives experience they can add to a resume without the time commitment of a job or internship.

Introduction

Project Summary

The objective of OverIT is to provide residential digital technology consumers with a web application that they can use to find basic on-demand IT services in their area in real time.

Problem Statement

Today, there are few basic ad-hoc or on-demand IT services available for digital technology consumers in a residential space (Home&). Left unchanged, many consumers will have to resort to expensive scheduled services (*Smart-home-services-best buy*) that do not offer any level of immediate resolution for their needs. As more consumers adopt digital technologies into their homes and lifestyles, it is important and imperative that they receive the proper level of service if they need immediate assistance with a product to reduce return rates and help educate consumers on new digital solutions available to them.

Solution

In leveraging OverIT users will be able to select from a predetermined list of solutions which are available in their area and submit a formal request for either in-home or remote assistance. OverIT is a free service provided by volunteers who give their time to help individuals in their community. Once an IT contractor has picked up the request from the queue, a handshake between the user and contractor will occur, and the scheduling process will begin. Users will work with the contractor to select a time and the session will be scheduled. Upon completion of the service, the user and the contractor will mark the request as completed in the app. In terms of competition, we have found some similar services that exist like HelloTech.com which is more of a product seller that also sells services for those products. As opposed to general IT help with any product that is outside of their product list, their services are provided via phone call or a live chat. Similarly, we also have BestBuy's Geek Squad service that would be our biggest competitor in the market. They offer both remote and on-site support for any device at a yearly cost of \$200, which provides total tech support. They will work with you via call/chat/or at home visit, which will cost an additional \$50 or more. They also offer other specific services like home Wi-Fi setup, smart doorbell installations, and other smart tech installs that are all around \$100 per service.

Project Source

Nathan Coffee was the contributor who conceived the problem statement as well as the project solution for our project, OverIT. We all conducted the requirement analysis as a team, with most discussions about the requirements being held as a group. All requirements went through an internal evaluation process where the team would give feedback on each requirement that was submitted in an open forum leveraging Microsoft Teams. Requirements that were approved were then added to our backlog with unapproved requirements being added to a separate backlog for future reevaluation. Our team was formed both on campus at UC as well as virtually through Microsoft Teams.

Discussion

Project Objectives/Goals

Our final goal is to have a website catering to two distinct types of users, customers, and volunteers. When a customer submits a request for IT assistance based on their needs, then any online volunteers will have the opportunity to take that request, schedule a session with the customer, and assist them with their IT problems. After the services are rendered, the volunteer will close the ticket and mark it resolved.

- Hosted Website
 - User Accounts
 - Customer
 - Volunteer
 - Services
 - Pricing
 - Location
 - Availability
 - Information
 - What we do
 - Who we are
 - Why you need our work
- Database
 - Users
 - Services
 - Pricing

Project Scope

Our team will develop a web application that allows users with personal IT issues to contact local IT operators that are available. The IT operators will be provided with information on the issue and the address of the problem by the client. Upon completion of the issue a ticket will be submitted on the site, which will indicate the job is closed.

Quick Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1	Brainstorming	~1 Month	August 23rd	September 20th
2	Wireframing & Design	1 Month	September 21st	October 22nd
3	Pre-Development Diagramming and Use Cases	1 Month	September 21st	October 22nd
4	Scrum Development Cycles	TBD	October 25th	TBD
5	Have 4 Webpages Running Locally and interlinked			November 12th
6	Unit Testing	2 Weeks	December 27th	January 7th
7	Integration Testing	2 Weeks	January 10th	January 24th
8	Integration	1 Month	January 24th	February 25th
9	Security Assessment	3 Weeks	February 25th	March 18th
10	Write- Up for publishing and Expo	3 Weeks	February 25th	March 18th
11	Final integration and launch.	1 Month	March 19th	April 19th

Figure: Project Timeline

Technologies Used

- Azure
 - Leveraging cloud services to tie our stack together, the team members will be splitting up different services throughout multiple cloud instances to ensure that costs are kept free/low. We will be leveraging the Microsoft stack not only for version control, storyboarding & project management, but for our hosting services as well.
- SQL
 - All team members have experience in SQL database schema design and use.
- React
 - All team members have experience in leveraging React to generate reactive templates for web applications.
-
- ASP.NET
 - Multiple team members have developed both desktop and web applications leveraging ASP.NET.
- Wireshark
 - Used for network testing.
 - Checking application security.
- Azure DevOps
 - Multiple team members with version control experience in Azure DevOps + project management.
- Postman
 - Used for testing microservices and API calls.

Technical Architecture Diagram

OverIT has a rather simple database and network structure, the network diagram shows how the devices themselves communicate in our network. Cellphones, iPads, and laptops all connect to the internet which they then communicate through a firewall into a DMZ and eventually into our internal network data center to process those customers' requests and IT contractors' responses to requests. The database diagram shows that we have three primary tables in use, the users table, services table, and incidents table. All of which contain their respective information.

Network Diagram

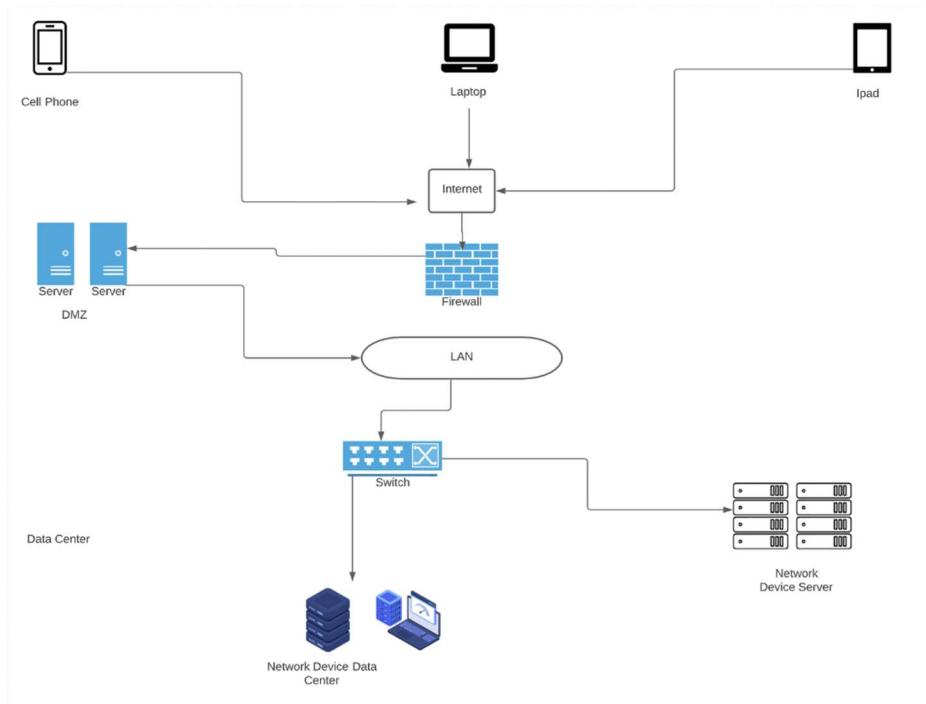


Figure: Network Diagram

Database Diagram

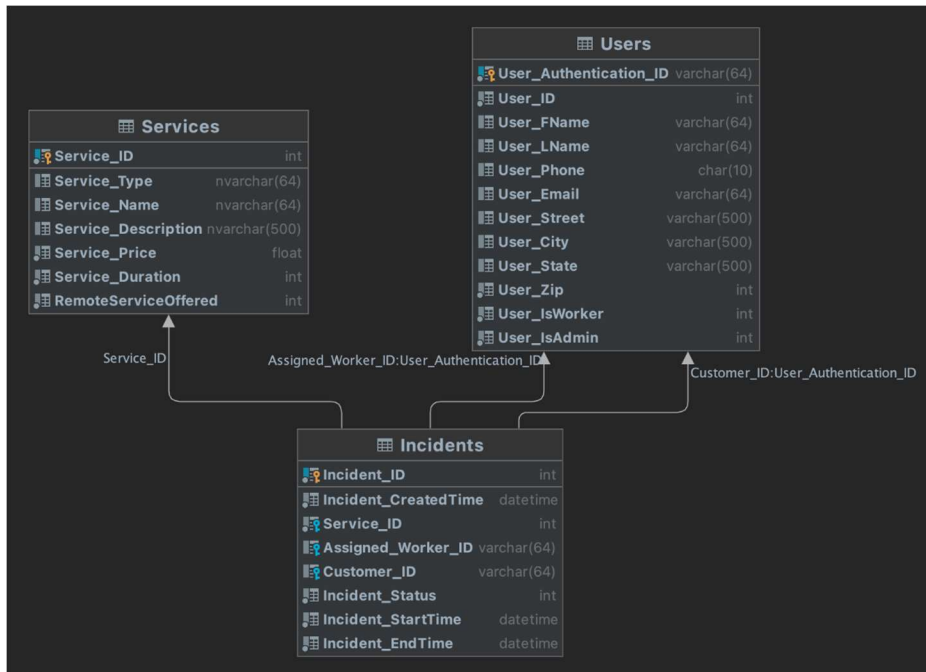


Figure: Database Diagram

Use Cases

Name	Elderly Customer
Description	The elderly customers that are typically handled in the IT industry are generally people who are just out of touch with technology, they are seeking general help with technology. Everything from setting up a printer to getting help removing a virus on their computer.
Actors	Older people who are out of touch with technology.
Frequency of Use	Going to be our most common customer we deal with, and I would estimate that 70% of customers that we receive business will be in this category.
Triggers	An elderly customer has an IT issue and calls us for support.
Preconditions	Elderly customers have IT questions or issues they need resolved.
Postconditions	Elderly customers have IT questions or issues resolved by one of our IT volunteers
Main Course	Main course of action for an elderly customer would be to go on our application and submit a request for assistance. Where they will submit what issue, they are having and how they would like to be assisted (in person, virtually, or by chat/phone call). Then they would schedule a time with an IT contractor and then complete the work and transaction.
Alternative Courses	An alternative course of action that could occur with an elderly customer would be instead of completing the work and the transaction being processed they cancel the assistance and choose not to be assisted.
Expectations	Overarching expectation for the elderly customer is going to be that they request their issue be addressed by one of our IT contractors, they schedule a time to be assisted with the contractor. Then they can get to work figuring out the best way to assist the elderly customer until the issue is resolved and then at the end of the scheduled time the customer pays the IT contractor. Then the customer's case can be closed.

Figure: Use Cases - Elderly Customer

Name	Younger Customer
Description	The younger customers that we deal with are typically someone who is not interested in learning about technology. They are likely to not have the time to set up their devices and deal with IT issues.
Actors	Younger people are too busy to deal with IT issues and setting up devices.

Frequency of Use	Via a rough estimate I expect an additional 30% of our customers to come from this category.
Triggers	Younger customers have an IT issue and access our website for support.
Preconditions	Younger customers or someone who does not have time to solve the IT questions they have.
Postconditions	Younger customers or someone who does not have time has their IT questions or issues resolved.
Main Course	The main course of action for a younger customer or someone who does not have time to set up a device would be to go onto our application and submit a request to have their issue resolved. They would then proceed to choose what kind of help they would like and how they would like to be assisted (in person, virtually, or chat/phone call).
Alternative Courses	A potential alternative course of action for the younger customer or someone who does not have time to deal with the IT issue would be to cancel the in-person meeting or virtual assistance they were going to receive. They may have found time to complete the setup or figure it out themselves.
Expectations	The general expectation for the younger customer or someone who does not have time to set themselves up is going to be that they request their issue be addressed by one of our IT contractors. They schedule a time to be assisted with the contractor, then they can get to work figuring out the best way to assist the customer until the issue is resolved. Then at the end of the scheduled time the contractor will close their case and submit a ticket indicating the job is closed.

Figure: Use Cases - Younger Customer

Name	IT Volunteer
Description	Our IT volunteers are going to be IT experts who have a wider range of skills assisting customers in different IT fields, everything from setting up Wi-Fi to removing a virus. The services will be available for our customers, they will be able to pick which jobs they want to work on based on a list of available jobs that are listed out for them. They will choose based on their experience level and what they feel comfortable working on.
Actors	Multiple IT experts that skills vary widely across multiple fields within the technology scale from printers to virus removal and system errors assistance.
Frequency of Use	IT volunteers will be one of our biggest actors in our application as they will deal with every single customer that makes a request. They will be dealing with 100% of the requests that come in.

Triggers	IT volunteers reach out to customers to set up a time to assist them. Customers will have various options available on how they would like to meet. (In person, virtually, or chat/phone call).
Preconditions	An IT expert is looking for side work to do and signs up to join our IT contractors list, then signs up to select different issues to help resolve.
Postconditions	Our IT expert completes the work they selected to help our customers and gets paid based on the work that was completed.
Main Course	The IT volunteers' main course of action would be signing up to become one of our verified volunteers that can assist customers with their IT issues. After becoming a verified IT volunteer, they can begin assisting customers. They will select which issues they would like to handle and which ones they feel comfortable assisting people with. They will be determining if they are willing to travel to people's homes to assist them in person or if they want to assist them via virtual, phone call/chat only.
Alternative Courses	With the IT volunteers an alternative course of action that occurs for them would be in the event they have a cancellation on one of their calls that they had scheduled. They would go back to square one where they select a customer to assist and begin scheduling a time to assist them.
Expectations	The expectations for our IT volunteers are that they select their availability for if they are taking in person visits or only virtual and chat/phone call customers. Once they have selected that, they can go through the list of available customers and select which customer they are willing/able to assist and begin scheduling times. Once they have a time scheduled, they begin assisting customers until their issue is resolved and then once resolved they are paid, and that customer's case is closed.

Figure: Use Cases - IT Contractor

Name	Application Administrator/Developer
Description	Application administrators are our own team of developers that will be making quality of life updates as well as a general hotfix that will be required for the application if any issues occur during the application's runtime.
Actors	Our team of administrators who develop and make changes to the application as needed are the original developers for the application and know every detail of the applications process and creation.
Frequency of Use	Our application administrators will be always working during the life of the app to reassure everything is working as it should and making quality of life improvements as the applications' popularity continues to grow.

Triggers	Application administrators are always initiated in the process during the life cycle of the app as they will always be working on the app to reassure it runs correctly and has any issues resolved immediately.
Main Course	The application administrators' main course of action would be working on developing the app to further the capabilities of the application as it continues to grow in popularity, as well as making any fixes in the application that need to be performed when issues are discovered.
Alternative Courses	With the application administrator an alternative course of action that occurs for them would be when they also perform elimination of IT contractors who do not sign up to help customers or have too many bad reviews, they will be booted from the OverIT system as a contractor. They will not be able to sign up to work for us again.
Expectations	The expectations for our application administrators/developers are to continuously make hotfixes and quality of life updates on the application. As well as maintaining the overall IT contractor pool by eliminating complacent contractors and negatively reviewed contractors from the pool of available IT contractors for customers to select.

Figure: Use Cases - Application Administration/Developer

Name	Security Administrator
Description	The security administrator will oversee creating incident response plans as well as overseeing checking application security and helping clear the IT contractor pool from complacent and negatively reviewed contractors.
Actors	Security administrator is one actor who oversees the applications security, incident response plans that are set up with the application administrator/developer teams and helping with the IT contractor pool.
Frequency of Use	Our security administrator is always monitoring the network to reassure no unusual traffic is occurring, as well as creating and updating incident response plans that help the OverIT team recover in the event an attack on the application occurs, as well as constantly working with the application administrator/developer team to manage the IT contractor pool.
Triggers	The security administrator is mainly initiated into their primary directive when any suspicious activity occurs within the OverIT network that is detected by the intrusion detection system that is put in place by the security administrator.
Main Course	The security administrators' main course of action would be always monitoring the OverIT application network for suspicious activity that could be someone attempting an attack on the application. As well as always updating and creating new incident response plans in accordance with the latest laws and regulations as well as the latest updates that occur within the application itself.
Alternative Courses	With the security administrator an alternative course of action that occurs for them would be when they also perform elimination of IT contractors who do not sign up to help customers enough or have too many bad reviews, they will be booted from the OverIT system as a contractor and will not be able to sign back up to work for us.
Expectations	The expectations for our security administrators are to continuously make updates on the OverIT incident response plans, monitoring the OverIT network, as well as maintaining the overall IT contractor pool by eliminating complacent contractors and negatively reviewed contractors from the pool of available IT contractors for customers to select.

Figure: Use Cases - Security Administrator

Use Case Diagram

The following use case diagram displays a general account of a user going through expected simple actions and how other actors interact with the actions and users to deliver help with OverIT.

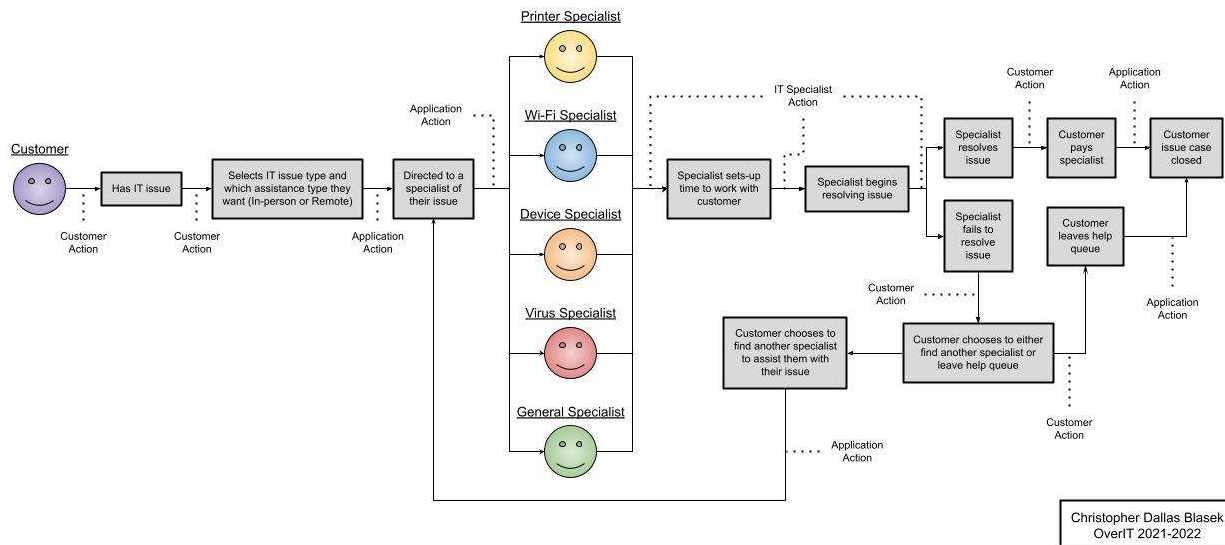


Figure:1 Use Case Diagram

Testing Plan

Overview

This section of the Over IT report contains our plans for testing from before the tests had been created and run, the documentation of our testing procedures, and a retrospective of the tests themselves. In the planning phase we will have our test methodology, the scope for the test and what will be tested, and the objectives of the tests themselves.

Methodology

The initial methodology for our testing was to test any functional part of our application that may need to process user input to create an output. This was done by creating unit tests with asserted correct values that the application could test against allowing us to fine tune functions and classes until we had reached the desired output. Additionally, we ran integration tests on the connection of the database to the microservice layer, and the connection of the microservice layer to the front end. This was necessary to ensure that not only were the individual methods of the application functional but also that they worked together to serve their desired purpose.

After integration testing, we ran user acceptance testing on the finished product by asking individuals that fit into our target demographic to attempt to navigate and use the application without guidance.

Scope

Unit Tests

Do we get the expected output from functions, are classes being constructed properly following the tenants of object-oriented programming and secure web design? Unit tests are written to assess functions and classes.

Ticket Request Case

Can the system accept user input to generate a ticket that requests help from an operator and notify operators that a ticket is available? This is important to test because this is what the most standard function of our system is required to do thus making this a benchmark for success.

Database Persistence

Does the database persist and allow the application functions to query and update it as necessary? This is important to test because access to live mutable data is necessary to provide our desired service.

User Acceptance

We will allow users that fit our target demographic to attempt to use our system to create ticket requests without direct guidance. This is important to the success of our project because we need to know how accessible our system is to unfamiliar users. We will measure the success of these tests using an exit poll about the satisfaction and ease of use of our system, as well as timing the user to make sure they are able to complete the tasks in a timely manner.

Objectives

Our testing strategy was to ensure that the building blocks of our system (individual functions and classes) were fine-tuned thus ensuring a smoother transition to integration. We wanted each feature to have both predictable and productive output before moving on to the next feature. After small features were accounted for, wanted to make sure that the integrated and deployed project would have demonstrable function by the date of the IT expo. This goal was achieved by testing an integrated system against our use cases and fine tuning the system for expected and productive output.

Test Logs and Procedures

Unit and Integration Testing

Item	Role	Expected Output	Actual Output	Pass/Fail	Reason for Pass/Fail	Success Yes/No
1	Implementation	Creation of home splash page	Homepage was created with some formatting done	Pass	Objective complete	Yes
2	Implementation	Addition of content pages	Additional content pages created for customer issues	Pass	Objective complete	Yes

3	Implementation	Drop down menu for customer problem	Drop down menu was implemented with basic issues like printer issue added	Pass	Objective Complete	Yes
4	Implementation	Page content refresh buttons	Refreshes the data tables present on current page/component	Pass	Objective Complete	Yes
5	Data Validation	Requests for assistance are added to the Database as input from frontend	Database updates successfully when code is executed	Pass	Output conformed to expected output	Yes

Figure: Test Logs & Procedures

Testing Review

Regarding the review of our testing, it became apparent that the ASP.NET/SQL back end in conjunction with the React front end may not be the best tech stack for our solution's needs. It also became apparent that our experience with React.JS may be sub-par as our developers were having to spend additional time learning to develop with the platform to meet our product's needs. Finally, we discovered that more time needs to be specifically dedicated to writing test for already developed infrastructure outside of currently dedicated efforts, this will allow us to create a more holistic test log and procedures and testing review writeup in future iterations.

User Acceptance Testing

Item	Tester Initials	Age	Time to Complete Ticket	Satisfaction with layout 1-5	Satisfaction with readability 1-5	Satisfaction with website overall 1-5
1	J.S.A.	52	1:58.6	3.5	5	2.5
2	L.B.S.	50	2:39.5	3	5	4

Figure: User Acceptance Test Logs

User Acceptance Testing Review

For user acceptance testing we introduced users to the home page of our application with no prior exposure, then we gave all of them the same scenario of "Your printer has been damaged and you want to use this website to create a ticket to ask for help." The users were then timed on their ability to create a ticket with no instruction. After the ticket was created, the users were asked to rate our website on three different criteria on a scale of one to five.

From user acceptance testing we learned about design choices that users with no prior experience found confusing or wording that they found unclear. This has been invaluable in planning and implementing updates to layout and descriptions within the web app.

Change Management Plan

Any member of the OverIT development team or adviser as defined by having signed their names on the project contract may propose a change to the project by writing a proposal and posting it in our Microsoft Teams group chat.

A change proposal must consist of the following:

- Change Description
- Justification for added value from the change
- Brief description of workflow change

The change can then be discussed at the next in person standup or class meeting. After the change has been presented & discussed the group will hold an open vote wherein a simple majority is required to confirm the change be added to the project.

If the change proposal passes the discussion will immediately move to where in the project timeline the change should be worked. The developer or developers responsible for implementing the change into the system shall propose a timeline implementation that they believe works best given the current circumstances of the project. Should that not be seen as satisfactory by a majority of the development team, a majority consensus must be reached to alter the development timeline.

Budget

Overview

This section of the OverIT report contains our projected costs for development and deployment of the app.

Azure

On an application our size we will use Azure Pipeline and Azure Artifacts. In our determination we found that using pipeline and artifacts would allow us to have a broad range of tools for use. We determined that by using 2 Microsoft hosted jobs which can run \$40 a job both jobs will cost us \$80. It will also allow us to have unlimited minutes of use. Azure artifacts will allow us to use 7 gigabytes which will cost an additional \$10 a month.

Google Firebase Authentication

Average Monthly Costs	
<u>Monthly Costs at 10k Users</u>	
Azure Pipeline	\$80
Azure Artifacts	\$10
Firebase Authentication Test I	\$60
Firebase Phone Authenticatio	\$0
Total Cost	\$150
<u>Monthly Costs at 17k Users</u>	
Azure Pipeline	\$80
Azure Artifacts	\$10
Firebase Authentication Test I	\$60
Firebase Phone Authenticatio	\$75
Total Cost	\$225
<u>Monthly Costs at 32k Users</u>	
Azure Pipeline	\$80
Azure Artifacts	\$10
Firebase Authentication Test I	\$60
Firebase Phone Authenticatio	\$225
Total Cost	\$375
<u>Monthly Costs at 43k Users</u>	
Azure Pipeline	\$80
Azure Artifacts	\$10
Firebase Authentication Test I	\$60
Firebase Phone Authenticatio	\$325
Total Cost	\$475
<u>Monthly Costs at 50k Users</u>	
Azure Pipeline	\$80
Azure Artifacts	\$10
Firebase Authentication Test I	\$60
Firebase Phone Authenticatio	\$400
Total Cost	\$550

As this is a startup it will not have many customers but will need to be able to support many customers eventually down the road. Firebase will be free until we need to surpass 10 thousand overall users. Any additional user in the US will cost an additional \$0.01 per user. We estimated that if we have a continuous growth of customers, we could see around 50 thousand overall users in the US after about a year's time. This would make the cost \$400 a month to be able to use Firebase. I would also say that we should have 3 hours per day available to us for testing the application on virtual devices which would mean that the overall base price of firebase would be \$60 a month plus any additional costs we incur based on the number of phone authentications that we have in the US. Overall cost for Firebase will be \$60 a month until we hit anything over 10k verified app users. Firebase is \$60 per month and increases once the app exceeds 10k users by \$0.01 per user. Total Thus Far per month for OverIT - \$150+\$0.01 per user above 10k users.

Figure: 2Budget/ROI Chart

Problems Encountered & Analysis of Problems Solved

Problem Encountered	Analysis of Solution
Database migrations were not automated causing some team members to be unable to properly configure project on their local machines.	Startup.cs was updated which allowed for initial run to not only setup DB but configure tables on initial start.
Issues setting up DB client on Mac OS caused delay in development from primary developer.	Leveraged Microsoft KB articles to get localDB setup on Mac which allowed for setup of application.
Page refresh button would cause the page to crash as it was incorrectly updating the component state	Leveraged online React documentation to correctly format state refreshes to allow for on timer/on button press state updates which would then update the data on the page.

Conclusion

Fall Lesson Learned

In conclusion, some of our lessons learned after the fall semester would be:

- Plan more time for tasks and assignments as needed. It is better to be done early than rush them at the last minute.
- Begin the development process earlier in the semester. It resulted in a demo that did not fully meet all our expectations for the fall presentation

- During the fall semester we did communicate through Teams, but we must do a better job communicating difficulties each of us are phasing during the development process.

Fall Conclusion:

After the fall semester, our team plans to focus on developing the final pieces of the OverIT application that we did not get to demo during our fall presentation. This will include implementing changes to our backend features to include firebase. To meet our objectives, we will continue to meet 3 times a week during the spring semester. Towards the middle to end of next semester our team will conduct penetration testing, user testing, and general feedback. Lastly then we will begin preparing and practicing for the IT expo at the end of spring semester.

References

- Smart-home-services-best buy*. BestBuy.com. (n.d.). Retrieved September 27, 2021, from https://www.bestbuy.com/site/promo/smart-home-services?ref=212&loc=1&ds_rl=1262686&gclid=Cj0KCQjw18WKBhCUARIsAFiW7JxZItiBFJS487_psXArQWPpo3HFF9dRRqXAEcvLwsXvSs5uFxeKZF4aAoVhEALw_wcB&gclsrc=aw.ds.
- “How to Create an Effective Technical Architectural Diagram?” *Alibaba Cloud Community*, https://www.alibabacloud.com/blog/how-to-create-an-effective-technical-architectural-diagram_596100.
- Balosin, Ionut. “The Art of Crafting Architectural Diagrams.” *InfoQ*, InfoQ, 4 Aug. 2017, <https://www.infoq.com/articles/crafting-architectural-diagrams/>.
- “What Is a DMZ and Why Would You Use It?” *Fortinet*, <https://www.fortinet.com/resources/cyberglossary/what-is-dmz>.
- Home& Business Tech Support, smart home set up, installations, and IT services near you. HelloTech. (n.d.). Retrieved February 18, 2022, from <https://www.hellotech.com/us>