



Honors Theses

Lee Honors College

4-20-2021

OIT Web App

Stanley Ritsema

Western Michigan University, swsritsema@gmail.com

Follow this and additional works at: https://scholarworks.wmich.edu/honors_theses



Part of the Data Science Commons

Recommended Citation

Ritsema, Stanley, "OIT Web App" (2021). *Honors Theses*. 3429.

https://scholarworks.wmich.edu/honors_theses/3429

This Honors Thesis-Open Access is brought to you for free and open access by the Lee Honors College at ScholarWorks at WMU. It has been accepted for inclusion in Honors Theses by an authorized administrator of ScholarWorks at WMU. For more information, please contact wmu-scholarworks@wmich.edu.



**CS 4910 – Software and System Design II:
Implementation and Testing
Final Report – OIT Web App**

Stan Ritsema, Andrew Hazlett, Nicholas Warriner

Last Edited: 29 April 2021

Table of Contents

2. Abstract	3
3. Problem Statement.....	3
3.1 Need	3
3.2 Objectives.....	3
3.3 Terms, Acronyms, Glossary	3
4. Problem Analysis and Research	4
5. Requirements	5
6. Standards and Constraints.....	5
6.1 Applicable Standards.....	5
6.2 Constraints	5
7. System Design.....	6
8. Testing	6
9. Results	6
9.1 Realization of Requirements	6
9.2 Realization of Standards and Constraints	6
9.3 Testing Results.....	6
10. Future Work.....	7
10.1 LDAP	7
10.2 Authentication.....	7
10.3 Transition	7
11. Conclusion	7
12. References	8
Appendices:	8
A. Project Management Plan	9
B. Progress Reports	9
C. Development Costs	17

2. Abstract

The goal of this project was to create a web app to assist WMU's help desk in handling various user issues relating to Office365 and WebEx. The three issues of unblocking email, enabling live-streaming, and changing the URL of a personal meeting room all require administrative access but the OIT department wanted to empower front desk staff to handle such requests. The project was designed using a web server that takes user input from a help desk employee and executes java functions that make API calls. In the end, we were able to successfully create this proof-of-concept prototype for two of the functions. Enabling live-streaming via API is not possible. In the future, this project will be retooled into a plug-in for the helpdesk's main tool, ServiceHub.

3. Problem Statement

3.1 Need

The OIT team at WMU wanted to empower front-desk computer support staff to perform changes to end-user accounts in Microsoft 365 and Cisco Webex. The issue they were having was that the changes they had in mind required administrative access to the respective environments. A need existed to create a middle-man program to allow front-desk staff to access a controlled subset of administrative controls.

3.2 Objectives

As a proof of concept, the OIT team requested the implementation of three functions: unblocking a user's Microsoft email account, enabling live streaming for a user's Cisco Webex account, and changing the display name for a user's Cisco Webex personal meeting room URL.

3.3 Terms, Acronyms, Glossary

- Application – A program or group of programs designed for end-users.
- Azure – A service through Microsoft that allows the hosting of a variety of applications, including virtual machines, and web servers.
- Backend – The part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data.
- CAS – Central Authentication System, this system is used to authenticate users.
- Frontend – Relating to or denoting the part of a computer system or application with which the user interacts directly.
- JSP – Java Server Protocol
- JDC – Java Database Connection
- LDAP – Lightweight Directory Access Protocol, an open and cross-platform protocol used for authentication of directory services
- Office365 – An online version of Microsoft Office, that is available to WMU attendees.
- PowerShell – A cross-platform task automation and configuration management framework, consisting of a command-line shell and scripting language.

- ServiceHub – An IT based software, where users can submit tickets to receive IT help.
- VM – VM or Virtual Machine is a piece of software that can be used to run a test environment or other software, while remaining separate from the host machine.
- Webex – An online video call software developed and distributed by Cisco.
- WMU – Western Michigan University

4. Problem Analysis and Research

Nicholas came across some issues while developing the PowerShell service to unblock a user's email. The first major issue was that directly working on WMU's mail service was not an option for security reasons. Instead, he had to create a developer sandbox account through Microsoft. This service created fictitious email accounts that made testing the program possible. Also, he initially started developing the program on Linux but ran into issues automatically authenticating with Microsoft Exchange. There probably was a way to solve this on Linux, but he found a way to easily solve this on windows platforms. This is why we ended up web hosting through a windows virtual machine on Azure. Lastly, Nicholas was able to find a way to automatically connect with Microsoft Exchange via a system certificate. This way the team avoided the need to put administrator credentials in plain text on the machine.

Stan encountered two significant issues while developing against the API for Cisco Webex: lack of documentation and unimplemented functionality. The Webex API is RESTful, meaning interacting with it occurs via messages in XML format (such messages could also be in JSON format, but for the purposes of this project being a proof-of-concept, the older XML format was used). This API style is easy to program against, but requires each message send and received to be large and clunky relative to the amount of actual data being transmitted. Stan spent arguably too much time attempting to bypass this issue by investigating Cisco's community-developed SDK's, especially their Java SDK. However, the documentation for each SDK, Java included, was woefully incomplete, as was the functionality of the SDK's. Stan had to return to working with the Webex API directly. Furthermore, only after directly conferring with Cisco Webex developer staff was it revealed that live streaming could not be enabled for end-users via the API. Nowhere in the documentation Stan reviewed was this constraint mentioned.

Andrew created the hosting environment for the web application. In doing so, he encountered issues with setting up the host machine, web server, database, and the front-end itself. For the issues with the host machine documentation provided by Microsoft proved to be a valuable resource in making the correct environment for our purposes. The database issues were relatively minor, as the main issues involved trying to connect the database to the web application itself. This was solved with the addition of additional libraries to the needed Java functions and the use of JDC. The largest issue that Andrew had to manage was the web front-end. The front-end proved to be quite a challenge as Andrew was not as experienced with Java compared to other front-end languages. The maintaining of this front-end and the addition of the other two APIs cause issues within some of the Java functions regarding compatibility. This was solved by using backwards compatibility settings to allow the use of some legacy style code to make all the APIs able to communicate properly. This may not have been the best solution but was the one that is in the current implementation.

5. Requirements

The following are the project's requirements as they stand at the project's completion:

- The project will follow the principle of least privilege to avoid giving help desk staff access to information they are unauthorized to see.
- The program will have a front-end web page where help desk employees can access and execute the following functions:
 - Unblock a user's Microsoft email account
 - Enable live streaming for a user's Webex account
 - Change the display name for a user's Webex personal meeting room URL
- The program will be written in Java to easily integrate this module with existing OIT programs.

6. Standards and Constraints

6.1 Applicable Standards

Standards that applied to this project are as follows:

- WMU web design standards
 - The interface, even if temporary, must be recognizable as associated with WMU.
- Java as primary programming language for the backend
 - This program may eventually be refactored as a plug-in for an existing tool written in Java.

6.2 Constraints

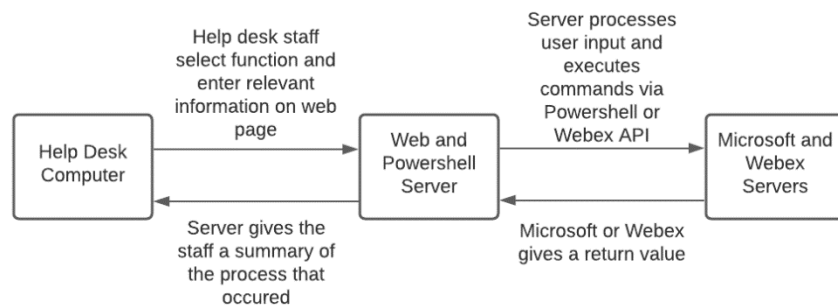
The OIT team requested a distinct split between front and backends. The eventual goal, though not of this project, is to refactor this program into a plug-in for the tool that helpdesk staff currently uses, ServiceHub. To that end, they requested that the project confine as much of the heavy-lifting code to the backend so that the interface could be as flexible as possible once the transition from stand-alone program to plug-in began.

This program is designed to operate more on WMU's intranet than the Internet at large. It would be hosted on WMU's servers and require CAS for authorized users to access it. Users would only be able to interact with the web page of the frontend. The backend, which contains the code to emulate administrative access, would not be accessible to users.

The safety risk of this program is minimal. Users will not have access to administrative functionality beyond what the frontend specifically allows, and the only private information revealed for users making requests of this program are their respective email addresses.

Economic constraints are likewise minimal. The programming tools and interfaces required to give this program the functionality it needs are free. The only real constraint is currently hypothetical: if Microsoft or Cisco change their API's in the future and do not maintain legacy API's, this tool will need to be updated accordingly.

7. System Design



We designed this system to take in user input from a web-based interface before passing along those commands to our server backend. Once the command had been passed, we would then process it and make the applicable changes. This can be seen in the diagram above.

8. Testing

The process for testing this application involved the following:

- Creation of test accounts for both Webex and Office365.
- Setting the test accounts to the appropriate state, i.e. blocked for Office365.
- Getting the authentication token from Webex
- Signing on to the web application.
- Completing the form for either Webex or Office365.
- Sending the form.
- Confirming the results within the test environments.

9. Results

9.1 Realization of Requirements

The team was able to accomplish all the requirements that were put forth save one. Despite the team's best efforts, direct communication with the development staff for Cisco Webex revealed that one of the requirements, enabling live streaming, could not be implemented via Webex API. That required had to be dropped.

9.2 Realization of Standards and Constraints

After receiving a copy of WMU's design guidelines, the team created an interface for this project and it was approved by the clients. The backend of the project was implemented entirely in Java and fully contains the two functionalities of unblocking a user's Microsoft 365 email and changing a Cisco Webex personal meeting room URL.

9.3 Testing Results

The testing results proved fruitful in that all requirements barring the enabling of streaming, were able to be complete via the web forms.

10. Future Work

Ongoing development for this project will consist of three distinct areas: adding LDAP, implementing permanent authentication, and transitioning the project from independent program to ServiceHub plug-in.

10.1 LDAP

LDAP implantation was a requested feature brought forth by our clients once we had already hammered out the implementation plan for the project. Thus we were unable to implement it. Future versions of the project would have LDAP serve as the 'messenger' between the front and backends. The frontend would put a 'request' into the LDAP queue with the needed information for the program to change a user account. Then the backend would 'receive' the request from LDAP and communicate with either Microsoft's or Cisco's servers.

10.2 Authentication

There are two aspects of authentication that need to be settled going forward: permanent Webex authentication and WMU authentication.

In its current state, the program requires a new Webex developer token to be manually entered into the code to access the Webex API every twelve hours. The process for rendering the program as a fully-fledged Webex Integration, capably of acquiring access tokens automatically, required that the project be in a more complete state than we were able to get it.

As we were unable to develop against WMU's actual systems, a live version of this project will also need to implement WMU's CAS, or something like it, to ensure that only designated helpdesk staff will be able to access it.

10.3 Transition

This project was envisioned from its inception as being stand-alone only until it could be transitioned into a plug-in for the helpdesk's existing tool, ServiceHub. The concentration of requested functionality in the backend was done specifically to facilitate this. The idea was that a new 'plug-in interface' would eventually replace the current interface for ServiceHub, and the backend would remain mostly unchanged.

11. Conclusion

The OIT department at WMU wanted to create a middle-man program. End users were coming to the front-end helpdesk staff with requests that required administrative access within their respective systems (Microsoft 365 and Cisco Webex). Since the OIT administrators did not believe that these requests required their explicit approval, they elected to commission a sort of pseudo-admin program that would selectively allow frontend helpdesk staff access to what would normally be administrator-only functions.

The project requirements consisted of creating a program that could accomplish three functions without using Microsoft or Cisco's direct administrative tools: unblock a user's email address, enable live streaming on a user's Webex account, and change the URL for a user's personal meeting room. Due to

security concerns, access could not be granted to WMU's systems for direct testing. Thus, this project was understood by all parties to be a proof of concept. Once developed, it would be handed off to the OIT department for further development.

The team used Microsoft Azure to host the front-end web service. This web page allows WMU's help desk staff to access and execute the back-end functions. The back-end function for unblocking a user's email account is done via Windows PowerShell. The back-end function for changing the display name for a user's Webex personal meeting room URL is done via XML. It was found that the functionality enabling live streaming for a user's Webex account is not feasible. We tested the program by entering fictitious email addresses created in the developer sandboxes.

The team was able to successfully create a prototype program that allows a WMU help desk employee to either unblock a user's Microsoft email account or change the display name for a user's Webex personal meeting room URL. We were able to successfully perform our implemented functions when testing these email addresses. This means that we were able to successfully create a proof-of-concept prototype for WMU's IOT department.

12. References

The following are references for the PowerShell script that unblocks a user's email:

[ExchangePowerShell Module](#)

[Connecting to the Module](#)

[View blocked sender address list](#)

[Unblock a user's email address](#)

Azure references:

[Setting up an Azure VM](#)

[Deploying a website using Azure VM](#)

Apache Tomcat References:

[Apache Tomcat 9 Documentation](#)

Cisco Webex references:

[XML API Reference Guide](#)

[IntegrationsIntegrations](#)

[Admin API](#)

Appendices:

A. Project Management Plan

Currently the project management plan is being worked on with the clients.

B. Progress Reports

Report 1: 25 January 2021 – Drafted by Nicholas

Project Information:

Team Members: Nicholas Warriner, Andrew Hazlett, Stan Ritsema

Client: OIT

Advisor: Not assigned

Report Date: 01/25/2021

Team Activity Report:

We are currently on the proof-of-concept stage. We held a meeting where we discussed how we are going to start programming this project. Before that though, we need to have a meeting with the OIT people to see what permissions we need to setup our server. We also came up with a couple questions to ask for clarification.

Client Interaction Report:

We did not meet last week but a meeting with OIT has been scheduled for this Friday.

Milestone Review:

Our first milestone to complete is to setup and ping the server. We are a little behind schedule because we need clearance from OIT. We anticipate being able to catch up quickly.

Issues (or stories):

No issues were accomplished this week, but we will have some next week.

Problems and Risks:

We need elevated permissions to run this server, so we hope the OIT people can give us some insight as to what security risks we need to be aware of.

Report 2: 1 February 2021 - Drafted by Andrew

Project Information:

Team Members: Nicholas Warriner, Andrew Hazlett, Stan Ritsema

Client: OIT

Advisor: Not assigned

Report Date: 2/2/2021

Team Activity Report:

Our team has met with the client to further go over the requirements for the WebApp. With what we learned in the meeting we have made new issues on our GitHub.

Client Interaction Report:

We met with our client on Friday the 29th. We discussed starting to create a prototype of the WebApp, which we are hoping to start soon. The client provided feedback about our plans to implement the WebApp, in the form of a new idea. We did not demonstrate a prototype.

Milestone Review:

Our first milestone to complete is still to setup and ping the server. This milestone has been updated due to some new information given by OIT. I believe we are still on schedule.

Issues (or stories):

We accomplished our final preliminary stages for research. There is still more to be done though.

Problems and Risks:

We will not be allowed onto a server to set up a test environment, this can be solved by switching up our methodology. Given new information by OIT we can use an intermediary system to process commands.

Report 3: 8 February 2021 - Drafted by Stan

Project Information:

Team Members: Nicholas Warriner, Andrew Hazlett, Stan Ritsema

Client: OIT

Advisor: Not assigned

Report Date: 2/8/2021

Team Activity Report:

Our team has met once more and further discussed what direction to take going forward and what our immediate next steps for the project and next questions for our clients are.

Client Interaction Report:

We have not met with our client since our last report. We plan to post questions in our client chat channel going forward regarding some specifics of the project and will plan another meeting if necessary. We have not yet demonstrated a prototype.

Milestone Review:

Our first milestone to complete is still to setup and ping the server. This milestone has been updated due to some new information given by OIT. I believe we are still on schedule. Additionally, we plan to have a basic webpage to serve as a front-end up soon.

Issues (or stories):

We accomplished our final preliminary stages for research. There is still more to be done though.

Problems and Risks:

We will not be allowed onto a server to set up a test environment, but our clients believe that we will be able to acquire a test environment(s) directly from Cisco Webex and Microsoft 365. We will investigate these possibilities going forward.

Report 4: 8 March 2021 - Drafted by Nicholas

Team Activity Report:

This past week we had our first faculty advisor meeting with Professor Yang. We discussed the progress we made with our web frontend. We also discussed what we need to get started on our backend.

Client Interaction Report:

We reached out to the clients this week asking for access to the LDAP servers we will use for the project. Project client Derek Diget sent us a sample LDAP provisioning entry to help us get started.

Milestone Review:

Andrew is currently looking at implementing small tweaks to the frontend such as adding a filter. Stan is looking into specific XML commands for the WebEx API. Nicholas is currently trying to understand LDAP and write a starter provisioning entry

Problems and Risks:

LDAP (Lightweight Directory Access Protocol) is something that none of us team members are familiar with. Do you have any experience with it Professor Yang? We will make sure to ask the clients questions about how it works, but we would appreciate if you could help us too, if you can.

Report 5: 15 March 2021 - Drafted by Andrew

Team Activity:

This week we got the basis of the Database for the web application set up and put up on the GitHub. We also made progress by obtaining a developer sandbox for Office365 and are close to obtaining a development environment for WebEx.

Client Interaction:

We currently have a meeting with our client scheduled for Friday at 1 pm. We also received more information regarding LDAP from the clients, including more information on logging into our testing area.

Milestone Review:

Andrew is currently looking at updating more of the frontend and adding more to the database. Stan is setting up the WebEx development sandbox. Nicholas is working on LDAP formulation and the Office365 Sandbox.

Problems and Risks:

Currently the Database has passwords as plaintext, this is strictly for testing, but we will hopefully be able to implement further security. We still need further information from the clients about features and LDAP usage, which we hope to solve this Friday with the meeting.

Report 6: 22 March 2021 – Drafted by Stan

Team Activity:

We have obtained developer sandboxes for both Office365 and Webex, and have made progress with a database for testing purposes.

Client Interaction:

We met with the clients at 1pm on Friday for two hours. We obtained more information on LDAP and how we might integrate our project into the OIT toolset.

Milestone Review:

Andrew is currently looking at updating more of the frontend and adding more to the database. Stan is working on the Webex developer sandbox. Nicholas is working on LDAP formulation and the Office365 Sandbox. We are preparing to develop a prototype within the next two weeks, and will leave fully integrating it into WMU's existing systems as the next version.

Problems and Risks:

We have a plan to develop a prototype, functional version of our program, but will need access to WMU's systems to fully integrate it. Last semester it was made clear to us that they could not grant us this access, so we are doing what we can to make it 'integration-ready'.

Report 7: 5 April 2021 - Drafted by Nicholas

Team Activity:

The team met today to discuss what we need to do to get our prototype working before the presentation on 4/20. Nick was able to make a PowerShell script that unblocks a user given an email address. He also created a java function to execute this script from the project's backend. Andrew created a windows server via Azure that is hosting our project. Stan has been going through Cisco's Webex API tutorial and exploring the sandbox environment.

Client Interaction:

No meeting with the client was made this week, however, Nick sent an update post on his work to the clients via Teams on 3/29. We are planning on meeting with the client this week for what is likely to be the last meeting before the presentation.

Milestone Review:

Stan is working on developing the Webex features for our project. Andrew is working on the database our project will be using. Nick is going to revise the Requirements Document and work on other materials the team needs to complete for this class. Another important milestone the team needs to complete is to merge our individual parts to yield a functioning project. We feel confident we are schedule to have a working prototype ready to present.

Problems and Risks:

Stan mentioned that he is having trouble understanding the OAuth protocols necessary for Webex development. Andrew may come across some issues trying to integrate JSP into the server. Nick currently has his portion of the project done, so the team can mitigate potential issues by asking him for help.

Report 8: 13 April 2021 - Drafted by Andrew

Team Activity:

Andrew has been working on getting the backend and frontend finalized before the project presentation on 4/20. Nicholas has been working on updating the requirements documentation and project report. Stan is continuing work on WebEx integration.

Client Interaction:

We had a quick client meeting today to go over the progress on the project and to let the client know that we would not be able to implement LDAP, at least until after the presentation. The client was receptive and was willing to adjust to this new information. Going forward we hope to implement these features before graduation.

Milestone Review:

The main milestone that we are pushing for now is to have the project operational by the presentation on 4/20. To that end, we are continuing work on the project, with Andrew working with the connection of the front and backend, Stan working on finalizing the WebEx integration, and Nicholas updating documents.

Problems and Risks:

Currently there are no problems and/or risks that we are aware of.

C. Development Costs

There were no developmental costs for this project. Nicholas and Stan were able to get developer sandbox accounts from Microsoft and WebEx for free. Web hosting from azure was also free. However, these were only free for a limited time so the college might have to pay for some of these services if they decide to implement this project.