Trello: <https://trello.com/w/onesupplychainmapsandtools>

Github: https://github.com/JoshPewtress/MapsAndTools

**Date: 12/11**

**Today’s Goals:**

* Create a Figma landing page for presentation.
* Meeting with Johanna.

**How did it go:**

The landing page was accepted, color palette needs to be swapped to a brighter orange to better correlate with The Home Depot, went to HomeDepot.com and grabbed color swatches of the site to mimic in my design. Johanna wanted to add more than I was initially planning with Version 1, with the additions I decided I will be using a database to store the information, will go with MongoDB as I’ve studied it and feel confident with the methods.

**Date 12/13**

**Today’s Goals:**

* Outline necessary Models and Data Access for Project

**How did it go:**

Planned out the Models. For the call flow-chart I will store the types of calls and the “Steps” to resolve them. The Flatbed Delivery Center tab will use Fdc, I will store general information about the delivery center. Email Templates will contain the name of the template, subject line and body for easy copy pasting. Back Office will store all Team Members and display their Backoffice days. Knowledge-base articles will store the Id and title so they can be searched.

**Date 12/14**

**Today’s Goals:**

* MongoDB academy courses.

**How did it go:**

Finished 3.5 hours of MongoDB courses, In-depth tutorials in Connecting, CRUD methods and Navigation. I also got more familiar with GitBash to be more confident with Branching, committing, and merging in the console.

**Date 12/16**

**Today’s Goals:**

* C# specific MongoDB academy courses.

**How did it go:**

Finished the MongoDB C# certification. Gave more examples of using Builders to create equality filters and general CRUD examples. Solidified my knowledge of how to make classes that could be serialized to BSON for use in MongoDB’s document model.

**Date 12/18**

**Today’s Goals:**

* Code the Models and Data Access classes

**How did it go:**

I created my solution, cleaned out the Blazor preset UI and finished my Class library models. Created and finished my DbConnection class for creating my MongoDB database and hooked up my Models to it.

**Date 12/20**

**Today’s Goals:**

* Create Data Access methods

**How did it go:**

I have the basic Data Access methods done, Retrieving, Creating, Updating and Deleting for each of my models, The more fine tuned methods still need to be worked. I am having trouble figuring out how I want to code the methods that add objects to lists within objects and then update the database with that information.

**Date 12/21**

**Today’s Goals:**

* Create the Index Blazor Page

**Problems I ran into:**

While working on the Daily Expectations element of the page I discovered that Bootstrap components do not innately work from following the examples on GetBoostrap.com. After scrapping the design from my figma whiteboard I wanted to use an Accordion as it is not obvious that clicking the titles does anything and I do not want to use buttons.

**How did it go:**

Created the Header that all my pages will use, easily implemented through calling the blazor component but the nav links currently go nowhere. Added a way to easily view the weekly team video for ease of access.

**Date 12/23**

**Today’s Goals:**

* Finish the Daily Expectations element
* Complete the Figma Design for the Call Flows page
* Start work on the Call Flows page

**Problems I ran into:**

I could not begin work on the Call Flows page because I had no way to create Call flows.

**How did it go:**

Completed the Daily Expectations element, I reverted back to the original figma whiteboard design but improved it visually with directions for users. In testing I tried to create documents within MongoDB for my CallFlows to see how they would foreach onto the page but my testing failed. I created the Create.razor page for a temporary solution to creating CallFlowModels and inserting them into my MongoDB database.

**Date 12/25**

**Today’s Goals:**

* Create all of the CallFlowModel documents
* Complete the CallFlows Blazor Page

**Problems I ran into:**

I am not able to pull the self-referential lists from my CallFlowModel accurately. The child CallFlowModel within the Parent CallFlowModel is correct, but it is not pulling the grandchild CallFlowModels from the child CallFlowModels. So my expanding list of call flows is not working properly

**How did it go:**

I created all of the CallFlowModel documents that I will be releasing at the program launch, I linked the child CallFlowModels to the ChildSteps property of the parent CallFlowModels but I am running into problems displaying the children properly for ease of use. I don’t want to hard code anything so I am going through the debugging process to see where the problem originated so I can loop through the list the way I intended.

**Date 12/27**

**Today’s Goals:**

* Complete the CallFlows Blazor Page

**How did it go:**

I located the problem with my CallFlowModels, I was creating them in the correct order and I was adding each CallFlowModel to its Parent’s ChildSteps as soon as it was created, so a reference to the child was put into the ChildSteps when it had 0 ChildSteps and then I would add ChildSteps to it afterwards. I re-created all the documents and added them to the parents in reverse order. It works at this time, in the spirit of keeping it extensible I also added a “ParentId” property to each CallFlow so I will swap the CallFlow’s page to use that in future so if more call flows are added it will still function, this will not be a on release feature.

**Date 12/28**

**Today’s Goals:**

* Complete the FDCList Blazor Page

**How did it go:**

I started out by creating a Card layout for outputting the steps for each Call flow and finished with that page completely. Then I created all the FDC’s that the program would need and saved them to MongoDB, created a search function and card layout to display all of the FDC’s with an expanding section to see the escalation phone numbers as they are not always needed. After finishing the FDC list page I started on the Templates page, I kept the same search logic as there was no need to change it and created all the EmailTemplateModels that I would need for the project. Created a card layout and used a NuGet package to allow users to copy all the important information from an EmailTemplateModel to their clipboard for ease of copy pasting. Spent about 12 hours today getting everything working but it was very worth it.

**Date 12/29**

**Today’s Goals:**

* Finish Web Application

**How did it go:**

Began by removing the Backoffice section of the header, models data access and references as it is no longer requested. I created KbArticleModels and saved them to the database. My FDCList and Templates pages both revolve around using cards to display their information and I did not want to replicate that a third time, so I decided to use a table, it gets the limited information across and is easy to expand if needed. Hooked up the DataAccess and finished the page. I used Dev tools with google chrome to check breakpoints in my pages, this application is only meant for web use and not phone use, so I got it working down to the large breakpoint but did not do any mobile changes as they were not requested.