Team Project – Workflow Coordination with Security Features

CSCI 4145 – Winter 2018

Brandon Lange - B00688070

Sarah Redmond - B00636646

Yaser Alkayale – B00633454

Table of Contents

URL of hosted web services 3

Resources Used 3

Code from other sources 4

Framework used to implement web services 4

Testing description 4

Employer Page 4

Mortgage Broker Page 6

Real Estate Company Page 7

Design 7

Web Services Description 7

Description of Portals and Hosting Details 8

Workflow Coordination Description 8

Databases 10

# URL of hosted web services

Serverside Hosting URL: <http://groupprojectmbr.azurewebsites.net/>

FrontEnd Hosting URL: <https://projectfrontend.azurewebsites.net/>

# Resources Used

* Macbook Pro with OS High Sierra V10.13.3
* Eclipse Oxygen.3 Release 4.7.3
* Brandon’s computer and tools
* Base model Macbook Pro with Touchpad 15 inch
* Processor 2.8 GHz Intel Core i7
* 16 GB 2133 MHz LPDDR3
* macOS High Sierra Version 10.13.3 Beta
* Visual Studio Code, Visual Studio 2017 Mac Beta
* Windows 10 Home Edition
* Notepad++
* Chrome
* Amazon Elastic Beanstalk
* Azure webapp
* Cloud Instance on Pay-As-You-Go with scale-out capabilities on the service plan.

# Code from other sources

Portions of the project were inherited from Assignment 4.

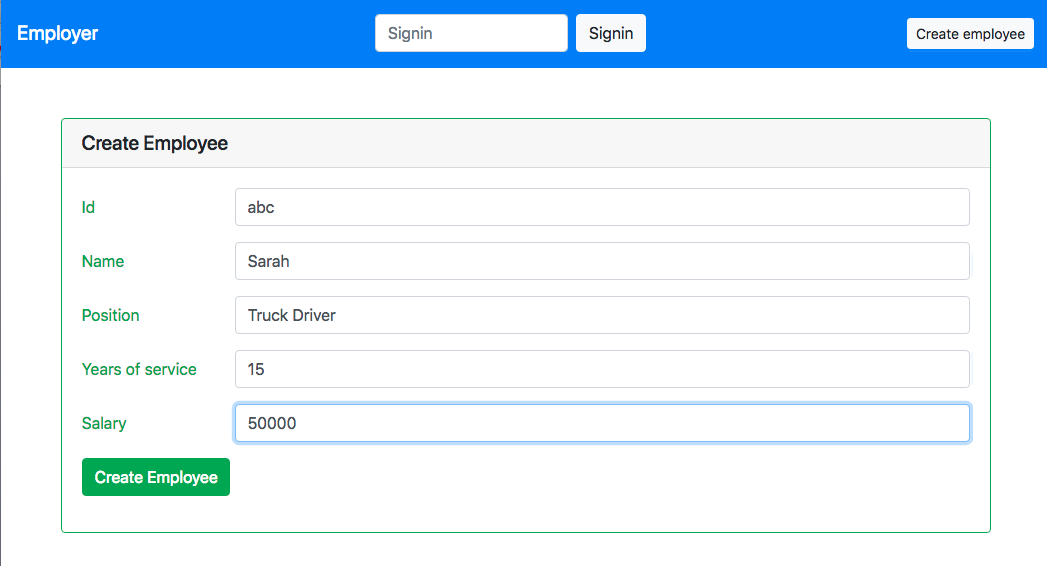
# Framework used to implement web services

The frameworks used are .NET core, jQuery, bootstrap, and Unrest.

# Testing description

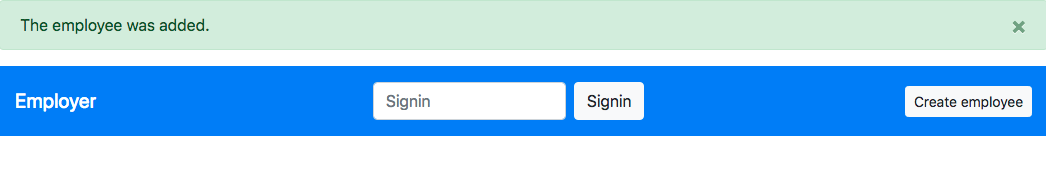
## Employer Page

#### Figure 1: Form Structure



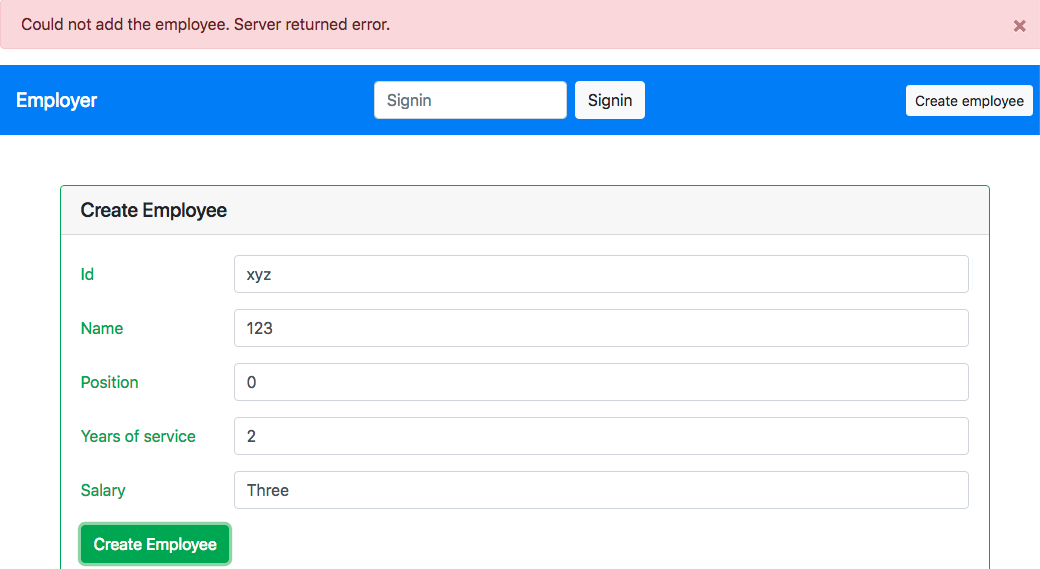
Upon entering the Employer page and clicking “Create Employee”, the page shown in Figure 1 appears. If all fields are completed with valid entries, a success message appears, as shown in Figure 2.

#### Figure 2: Success Messages



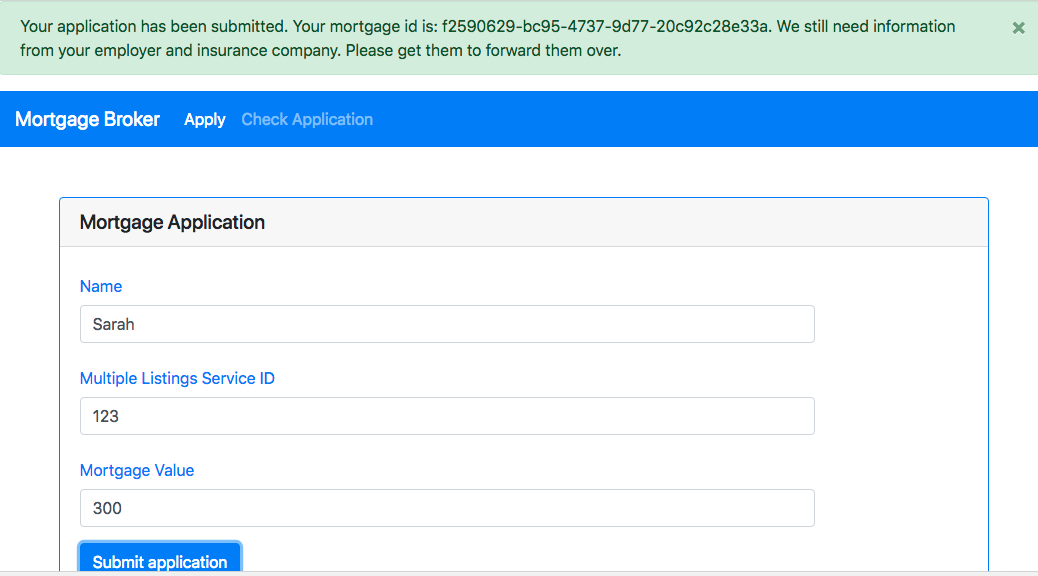
If invalid values are entered, or if a field is skipped, an error message is displayed as shown in Figure 3. Error messages such as this are shown if, on any of the pages, information is entered incorrectly.

#### Figure 3: Error Messages



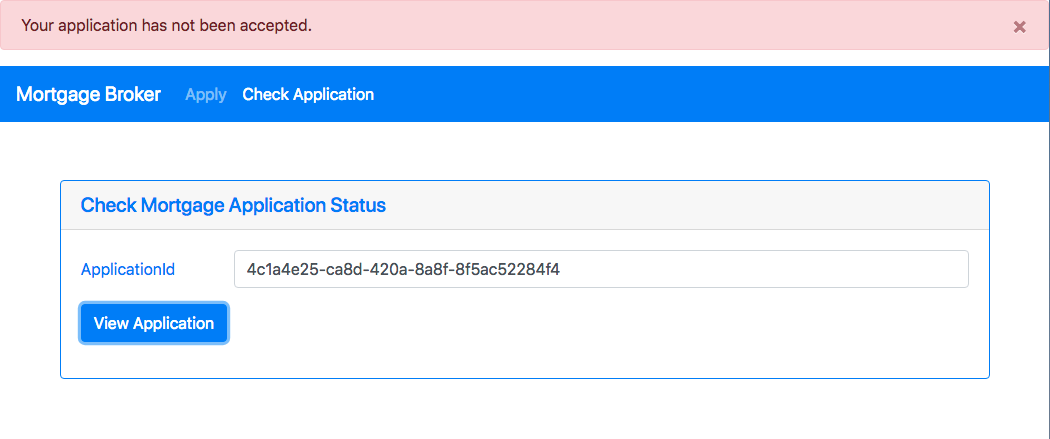
## Mortgage Broker Page

#### Figure 4: ID Creation



On the Mortgage Broker page, when all fields are filled in correctly and the button is clicked, a success message appears along with a unique Mortgage ID. At any point the “Check Application” tab on this page can be checked to see if the application is complete, as shown in Figure 5.

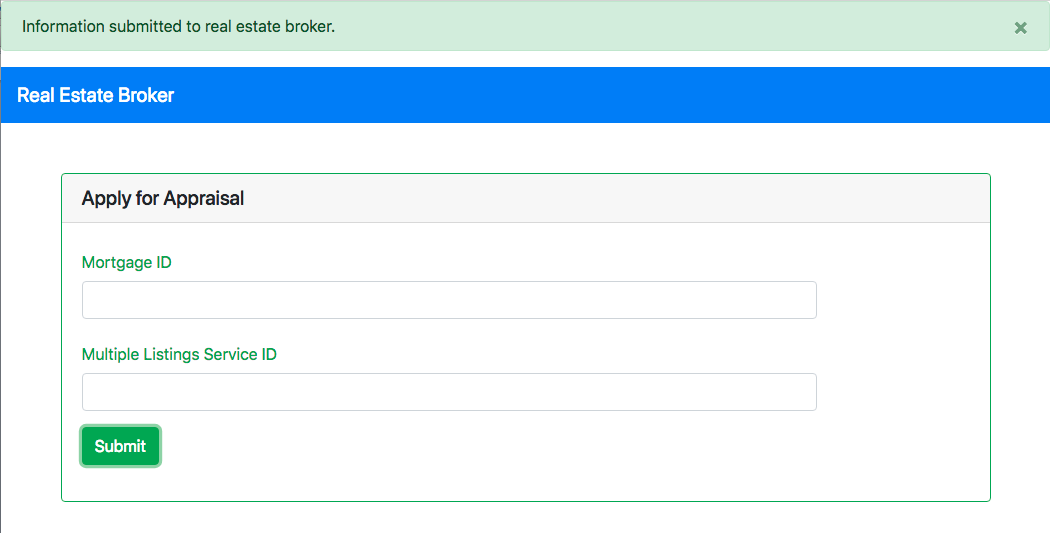
#### Figure 5: Status Check



## Real Estate Company Page

The Real Estate page has a simple form that displays a success message when all fields are filled in correctly, as shown in Figure 6.

#### Figure 6: Real Estate Company Page



# Design

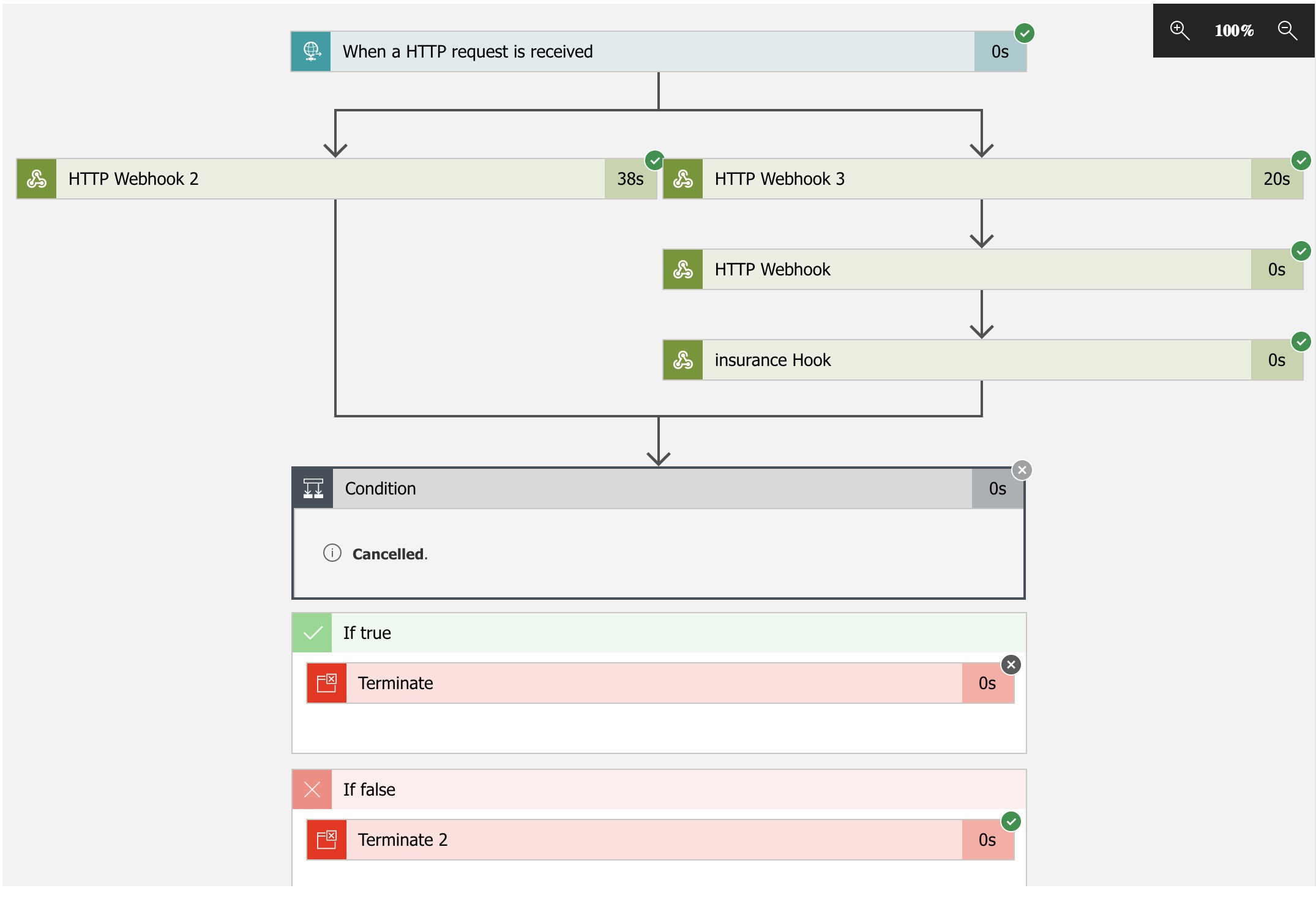
## Web Services Description

The service is pretty straight-forward. The logic app receives a request from the MBR to begin a mortgage request process. The app then initiates two other webhooks, which are used to handle the confirmations from the employer and insurer. The insurer, however, requires additional details from the municipality and real estate broker, so that webhook calls two others. When all of the details are received, the MBR database is updated to mark the user as having completed a request.

## Description of Portals and Hosting Details

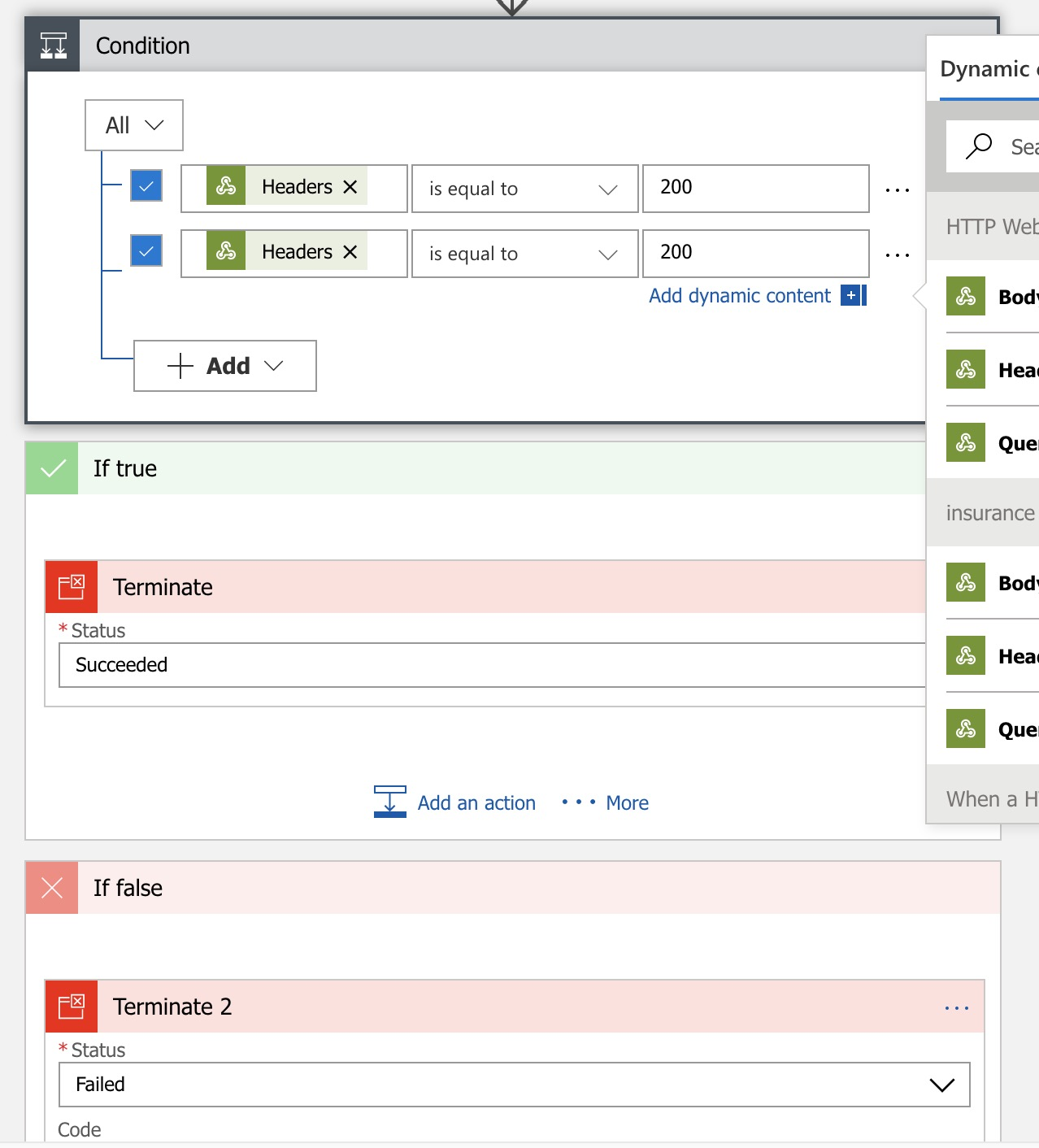
All of the web services are hosted through Azure using C# and HTML.

## Workflow Coordination Description

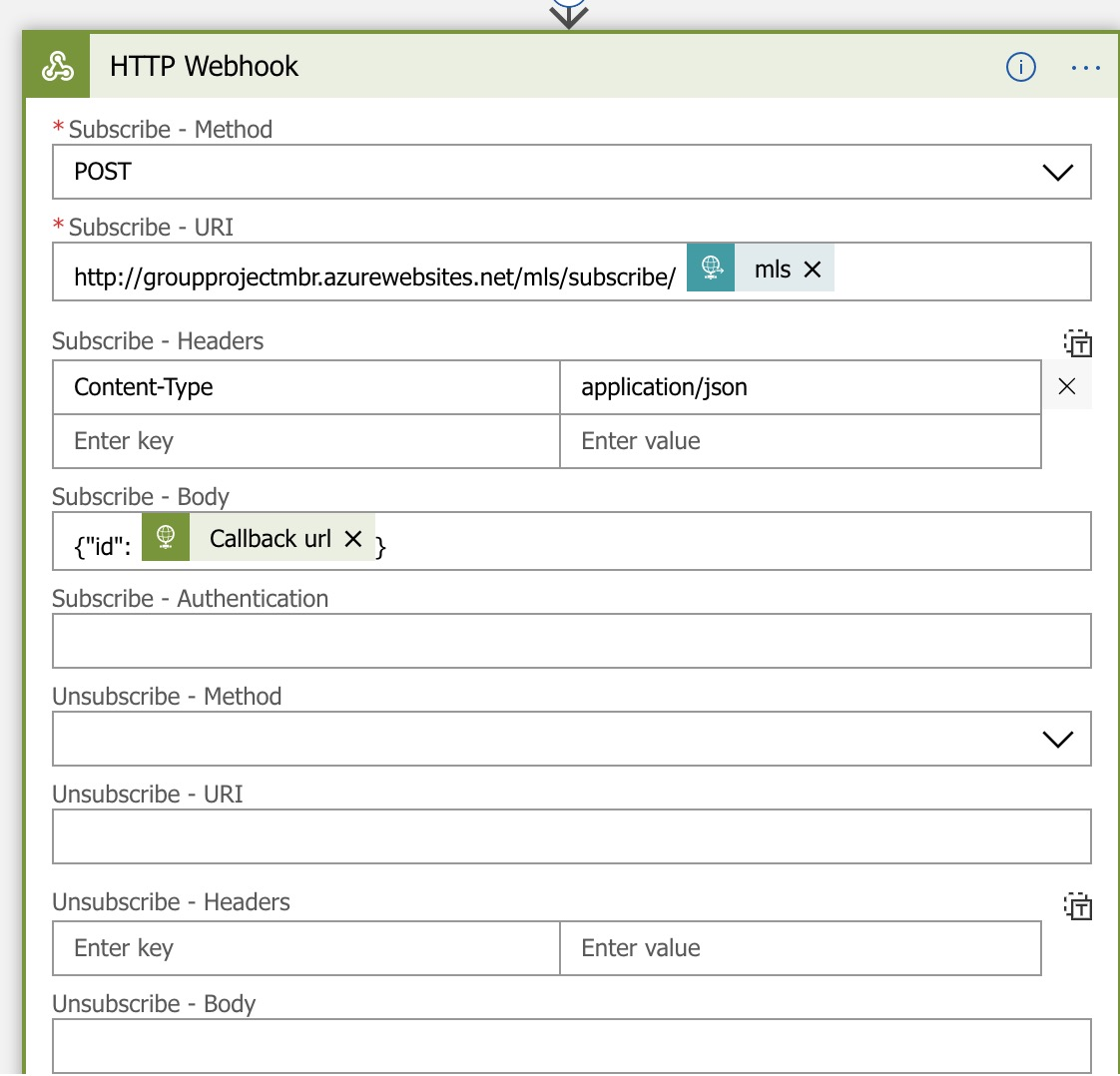
The workflow as a whole. Here you can see the webhook order of operation.

The condition statement ensures that a response code of 200 (good) is received.

This makes sure that all of the calls have been made correctly.



The layout of the webhooks are all basically the same, handing each other an ID that is used to track a user.



## Databases

* Employee Callback URLs Table:
  + String id
  + String applicationId
  + String callbackurl
* Employees table:
  + String id
  + String employee id
  + String name
  + String position
  + Long years
  + Long salary
* Broker Customer Table:
  + String id
  + String name
  + String MLS\_Id
  + Int value
  + Bool insurance approved
  + Bool employee approved
* Municipal Property Table:
  + String id
  + String MLS\_id
  + String police
  + String schools
  + String water
  + String sewage
* Insurance Property Table
* Real Estate Table:
  + String id
  + String MLS\_id
  + String MortId
* Real Estate Callback URL:
  + String id
  + String MLS\_id