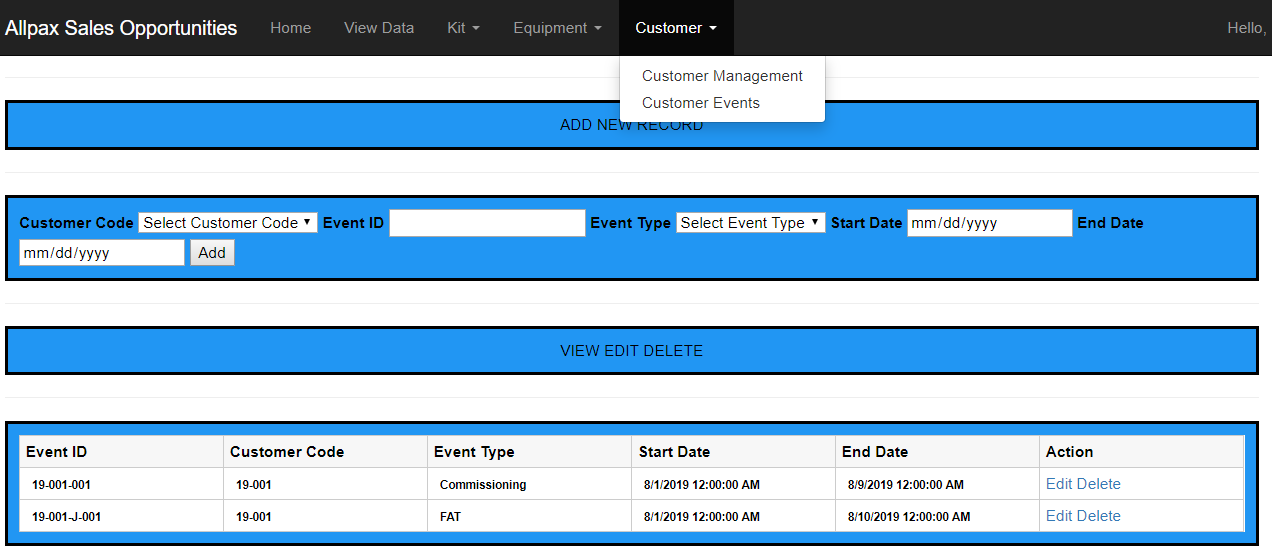
Below are turnover notes for the Allpax Sales Opportunity project.

**Banner**

I began by setting up a banner that will span across all pages. I believe this will be the final layout because it covers all topics, however at this time most of them are not attached to a view or a controller.



The logic for the banner is in the **\_Layout.cshtml** page.

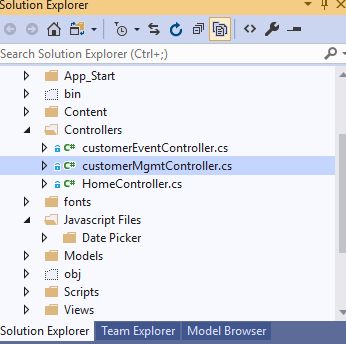


The only links on the banner that are attached to a view and a controller at this time are the **Customer Management** and **Customer Events** pages.

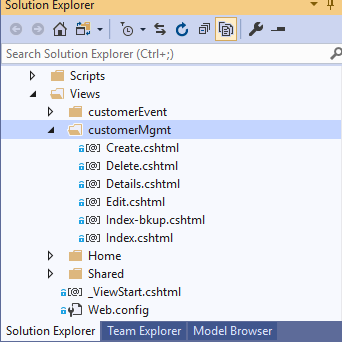
**Customer Management Page**

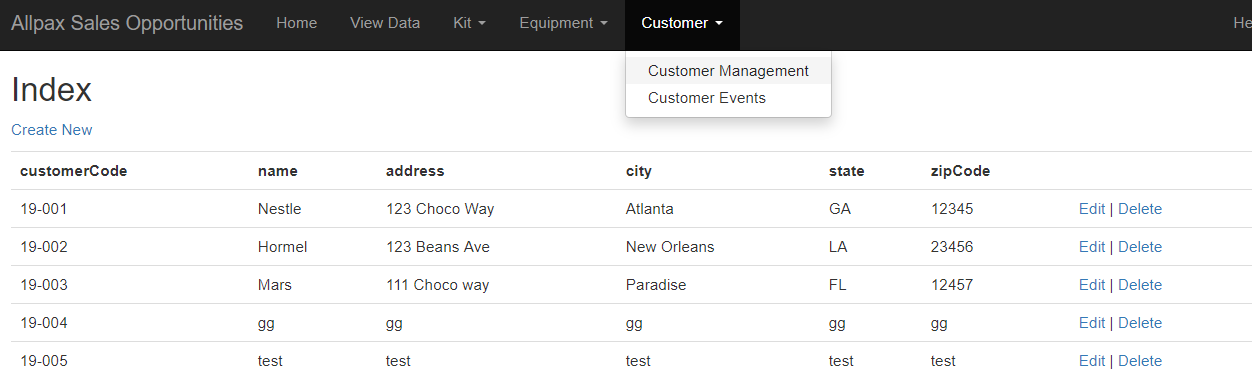
I began working with the Customer Management page to complete data entry and view, but after encountering problems, I decided to swap to another method on the Customer Events page. The Customer Management page was left as it was soley so I could refer to it as a canned go-by provided by Visual Studio to understand how information is passed between views, models, controllers and the database. This page presently allows for CRUD, just not inline CRUD.

**Controller**



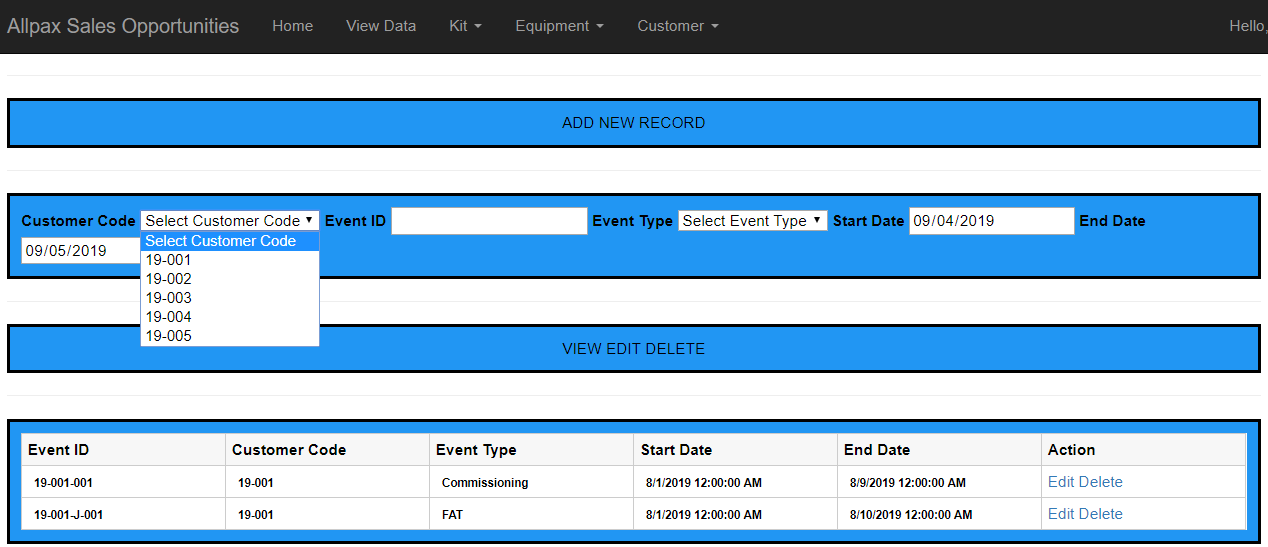
**View**





**Customer Events Page**

The Customer Events page contains all of my relevant progress for inline CRUD operations. An overview of this page follows below.



**Add New Record Section**

Customer Code, Event Type, Start Date and End Date are all tied to controls (dropdowns or datepickers) that are linked to the database. New records are added from the page as shown below…

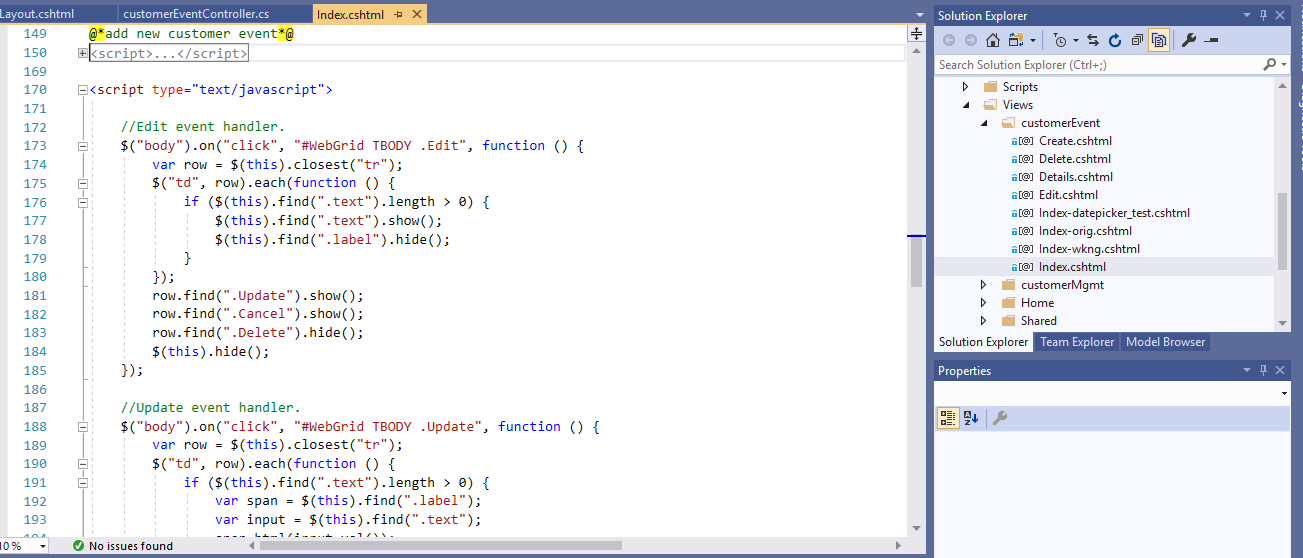


…and then passed to this controller.



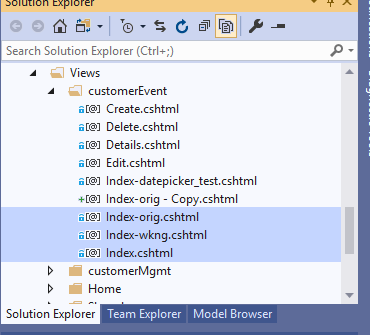
**View, Edit and Delete Section**

This section allows for inline operations on records through Jquery and AJAX. Operations on records are carried out in a fashion similar to how records are added.



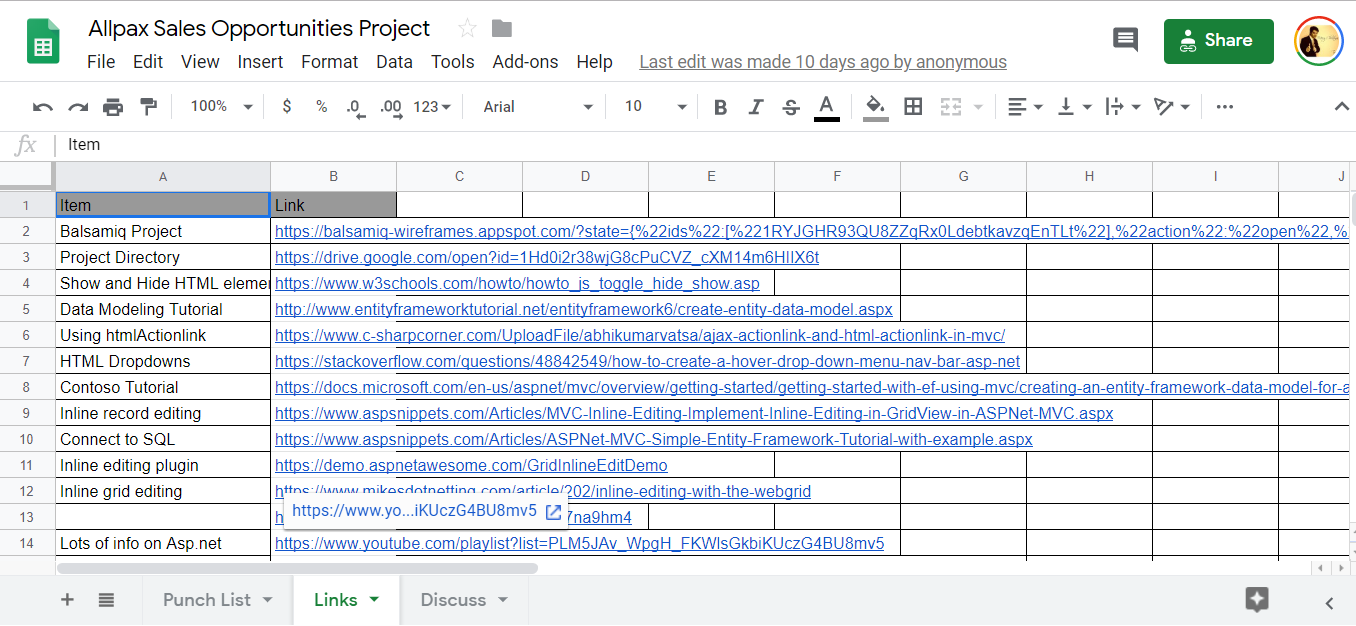
**Customer Event Views**

I am still working toward understanding exactly how the MVC model in Visual Studio hands off data between view, model, controller and the database, so I kept the basic view that was generated by scaffolding as a go-by. This view is called Index-orig. The view I am working toward perfecting for the final production version of this software that uses inline editing, AJAX and Jquery is called Index-wkng. The live view that is called up by navigation through the banner is called Index. My method has been to copy the contents of either of the first two views for testing in Index. This way, I can do my tests in a live way, but have a way to back out to a backup if something gets broken and I can’t easily find a way to fix it.

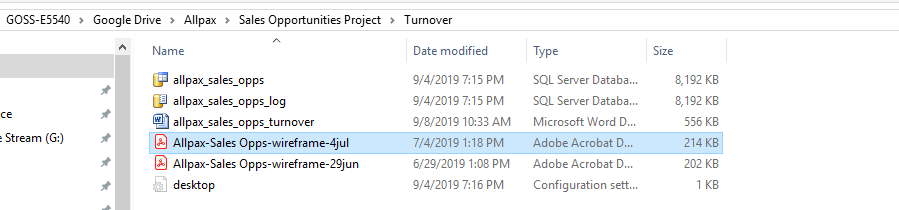


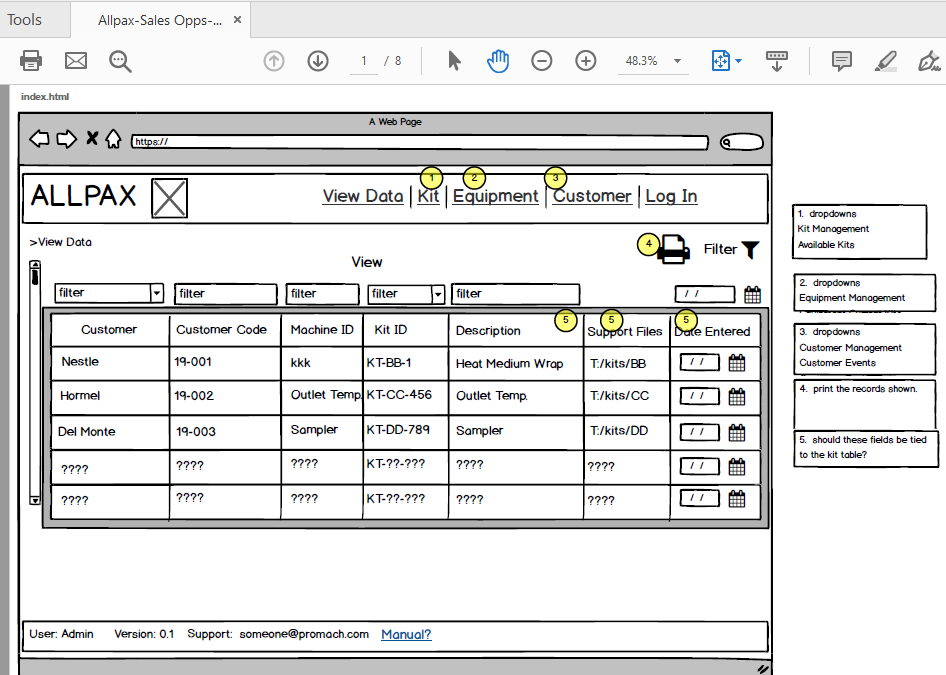
**Support Documents**

I have kept links to web resources that I used to complete what I have done so far in the Allpax Sales Opportunities Project spreadsheet. These links were VERY helpful.

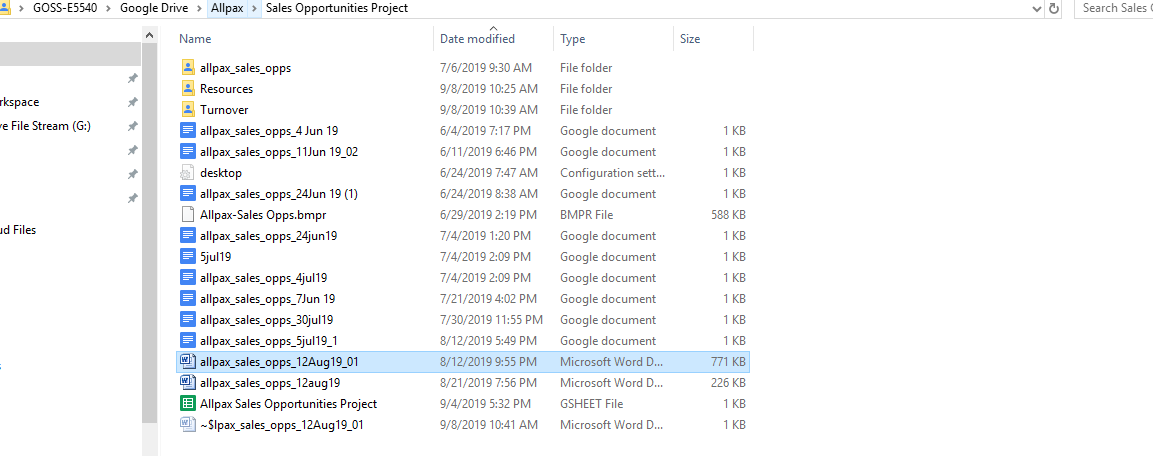


Prior to writing code in Visual Studio, a wireframe of how the OI will appear and work was prepared with Balsamiq.





Reports were submitted to Allpax on at least a weekly basis to document my progress. Contained within these reports are items that were completed and what items are left to be completed.



An example of where the project stands now follows below.

|  |  |  |  |
| --- | --- | --- | --- |
| To: | Watkins | Subject: | Sales Opportunity Project |
| From: | Goss | Type: | **Update RFI** Alert |
| Date: | 12 Aug 19 | Concerned Parties: | n/a |
| Client: | Allpax Internal |

Highlights:

-bound drop down lists on the customer events page to database values

-bound date pickers, but the result always includes time after a refresh. Have to figure the reason.

-can delete a record from the customer events page now

-created a new field called eventID in tbl\_customer\_events to tie things together

-adjusted the visibility of edit, delete and cancel on page customer events

To do:

-may have to put the project to sleep because I am now able to take 383.  Pischoff tells me 383 won’t be a time sucker, so I may be able to continue. Want to talk to you about an auxilliary.

-add JS logic that only allows for one record to be edited at a time

-set edit values to present values

-delete the cancel confirmation pop up

-add clear all new record fields button

-split up create/update/delete into separate JS scripts for readability

-need to auto refresh the contents of the db after a record addition

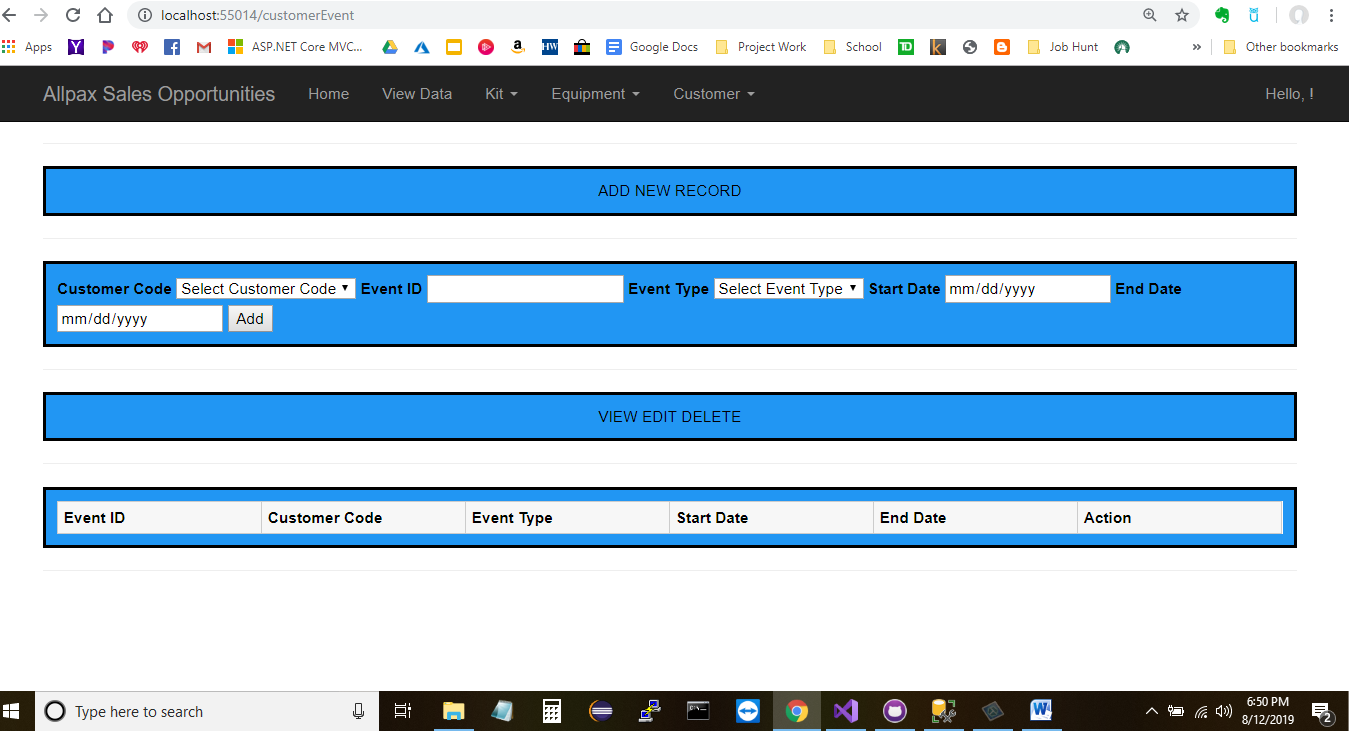
Watkins,

Find below screen shots of what I have been able to get done today.

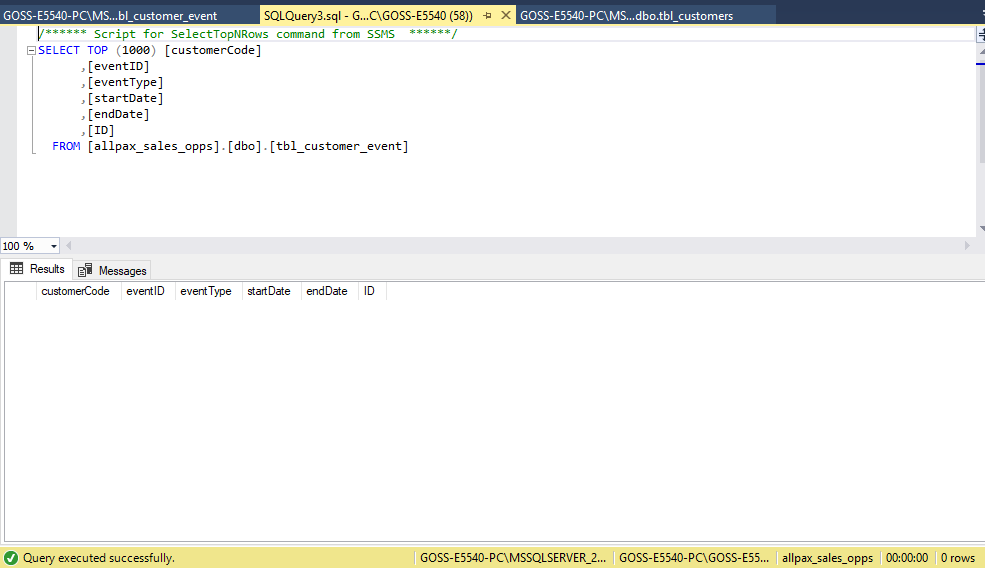
Google Docs is great, but I’m a screen shot and mark up kind of guy, and I don’t see those tools available in Google Docs yet.

Goss

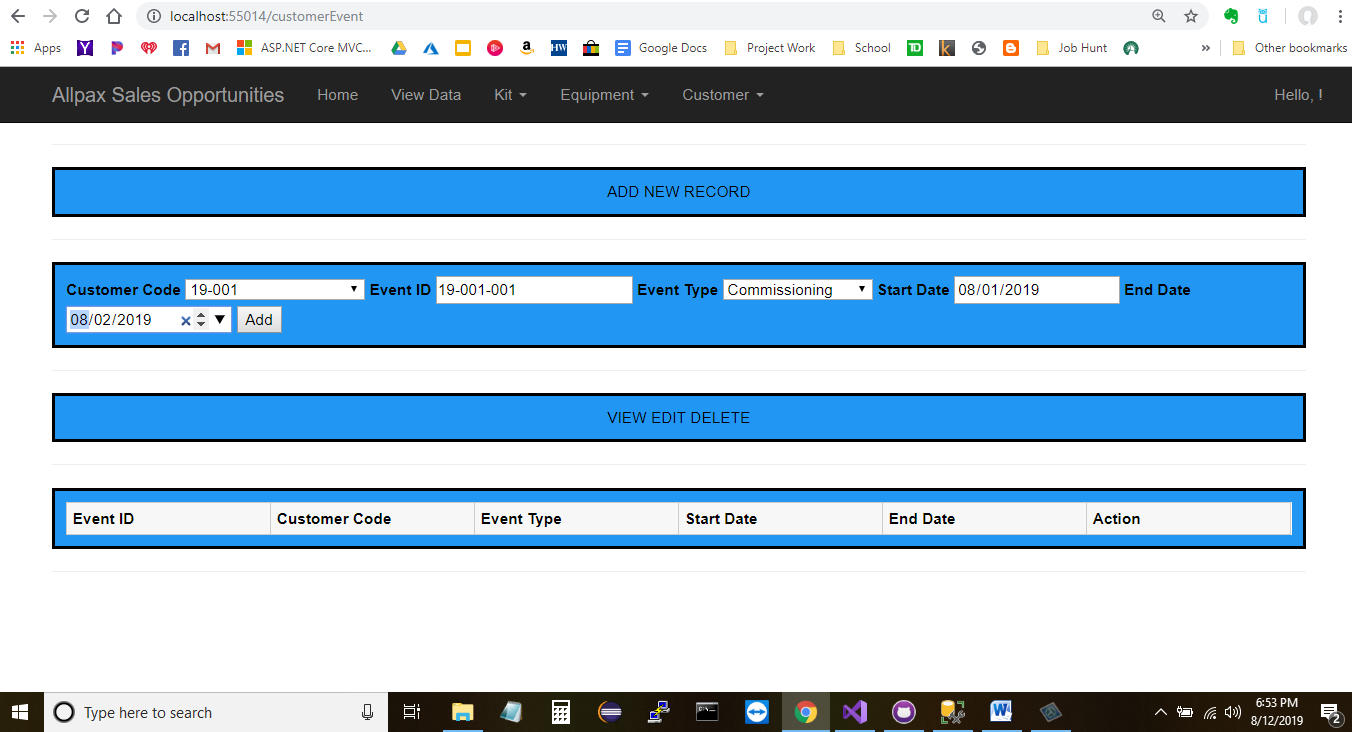
1. Create a record.



There are no records shown in the OI.

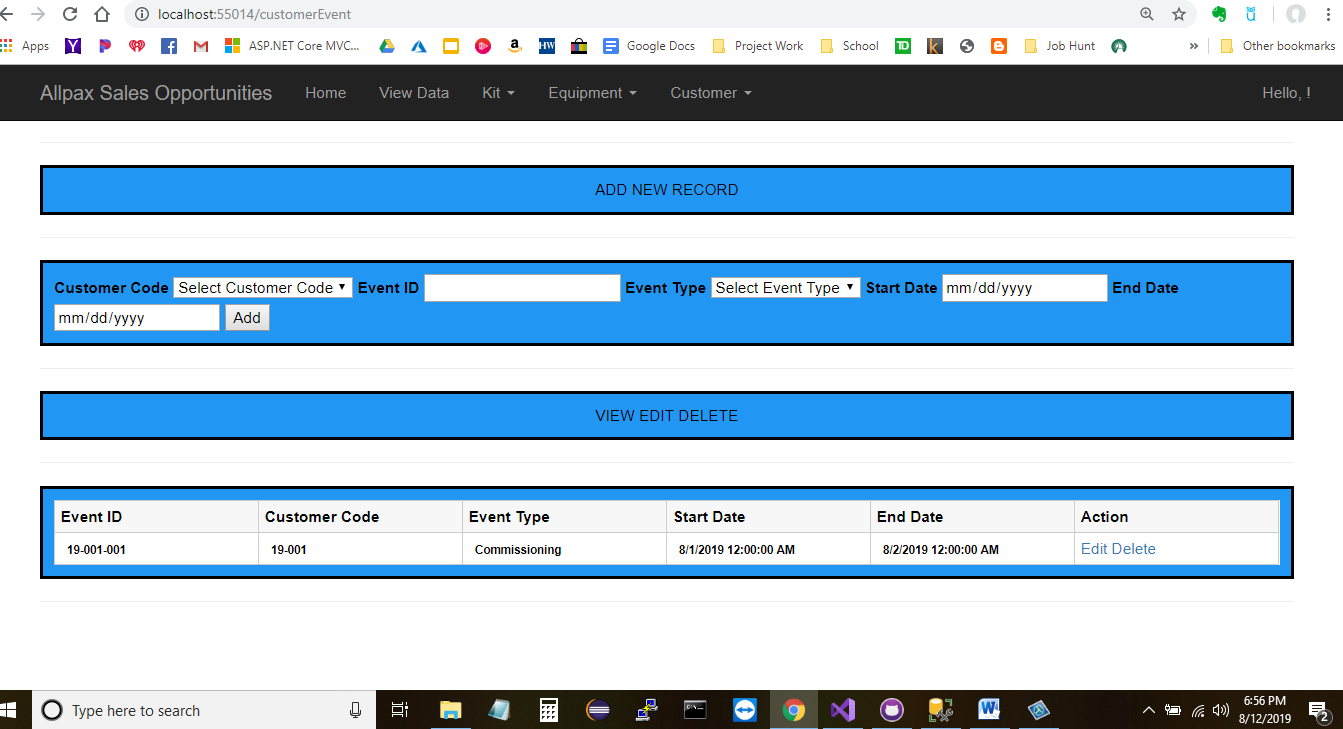


There are no records in the database.

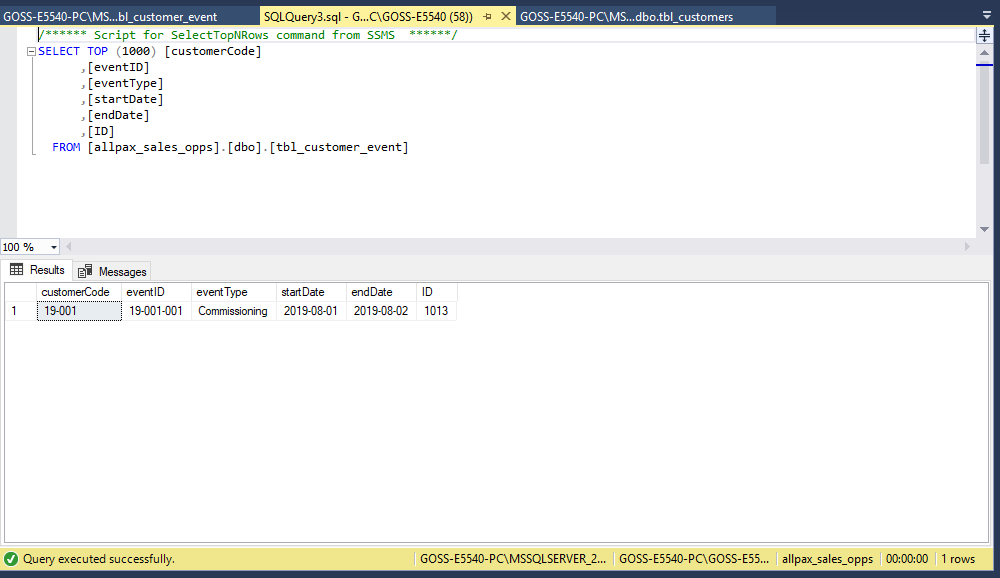


Press Add and then reload the page.

Added a new record with an event ID of 19-001-001.

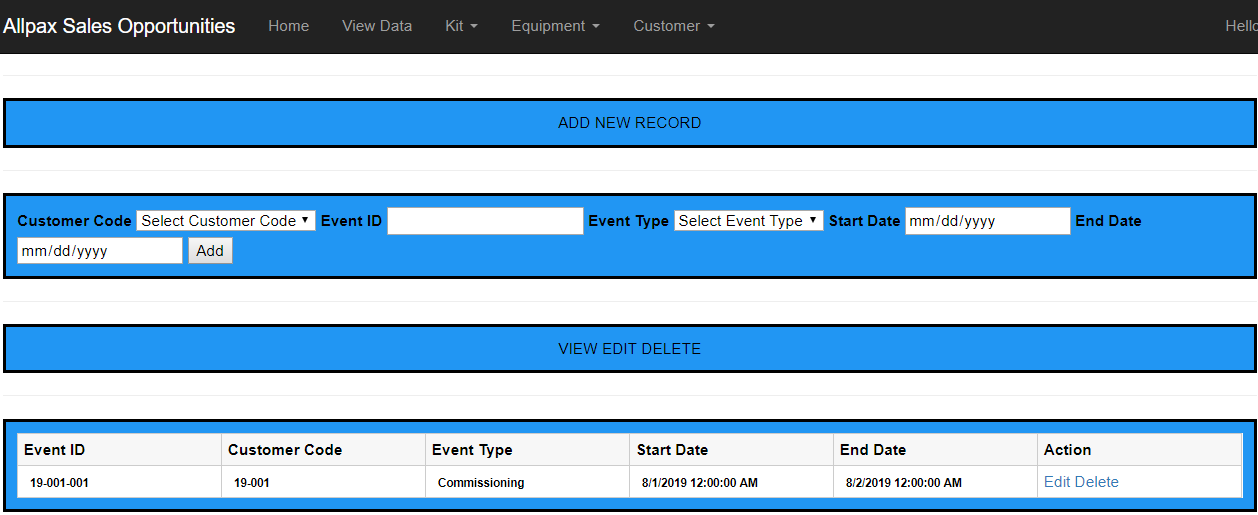


The new record is present in the OI…

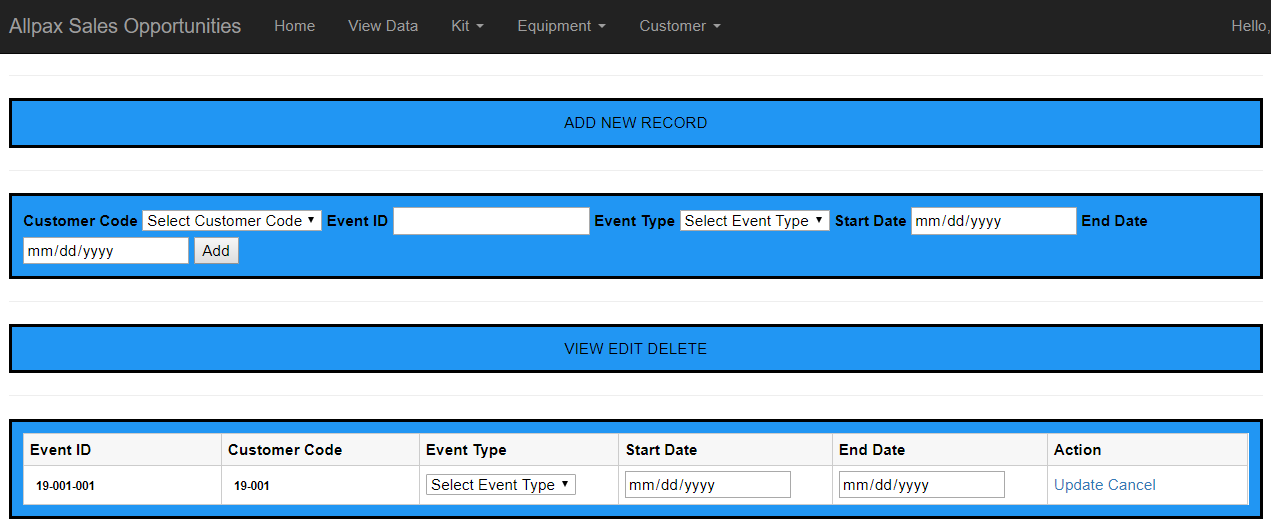


The new record is present in the database.

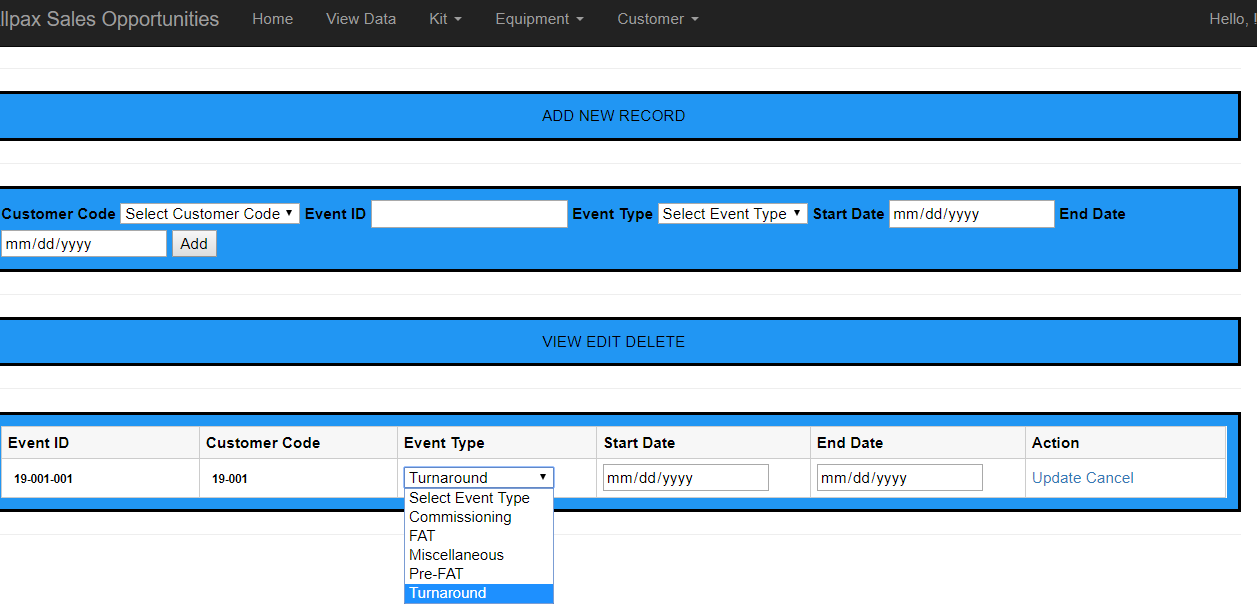
1. Update a record



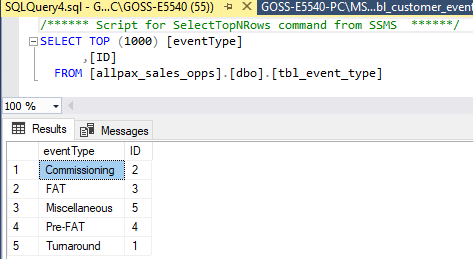
Presss edit.

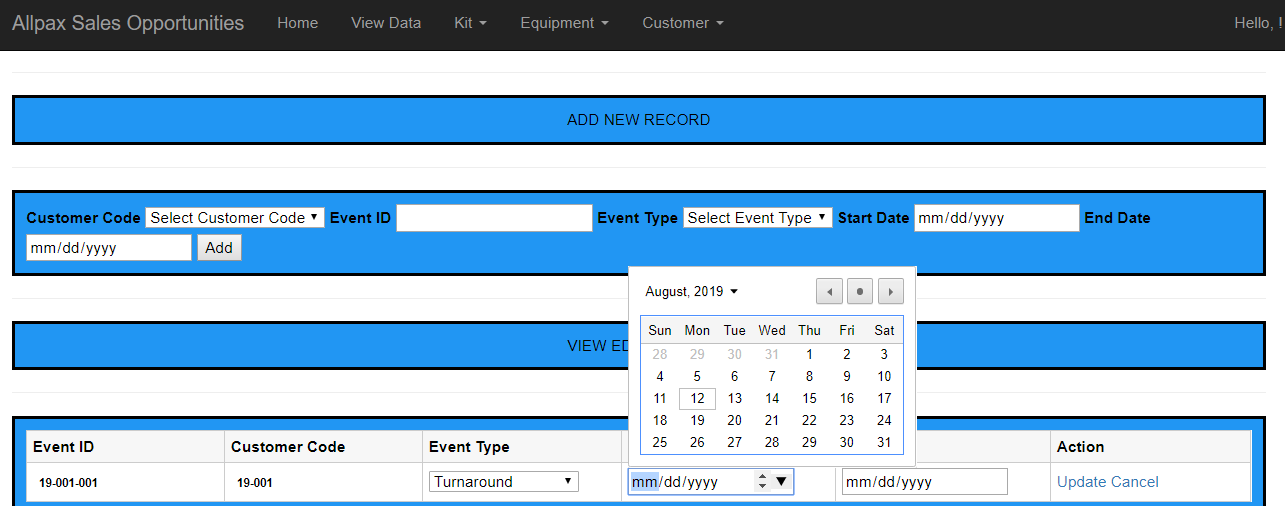


3 fields become alive due to jquery. Didn’t liven all 5 because this is an edit.

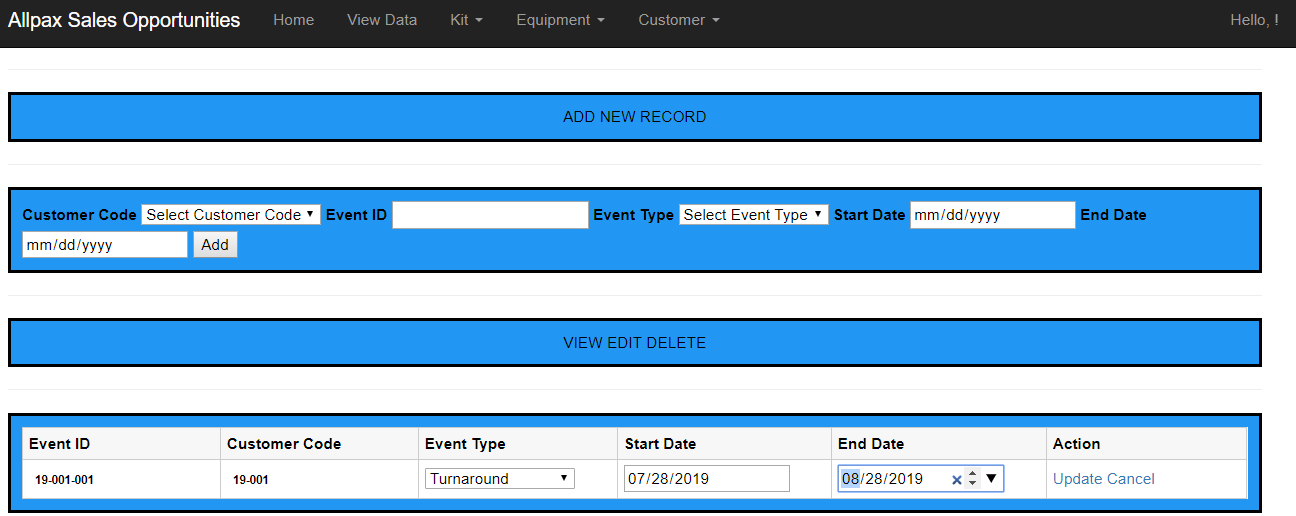


Select an event type from a drop down list bound to a table.





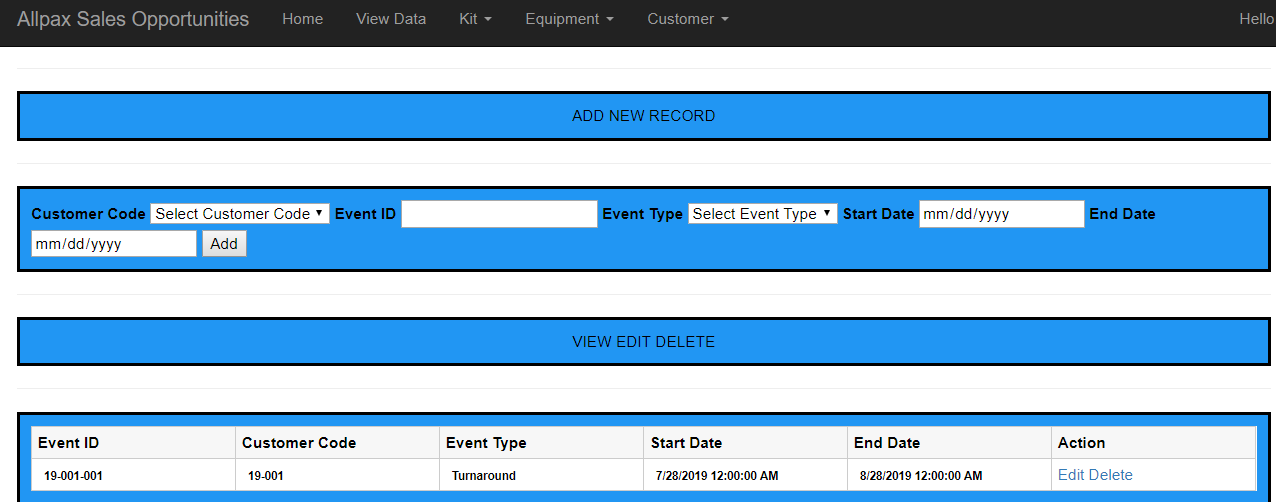
Change the start date.

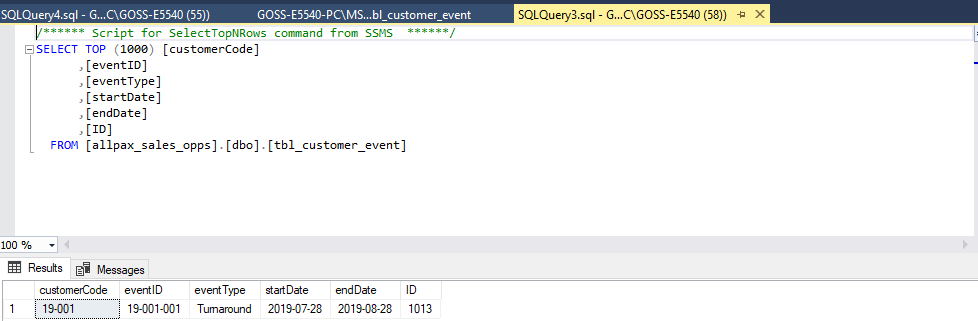


Change the end date.

Press update after the 3 fields have been changed.

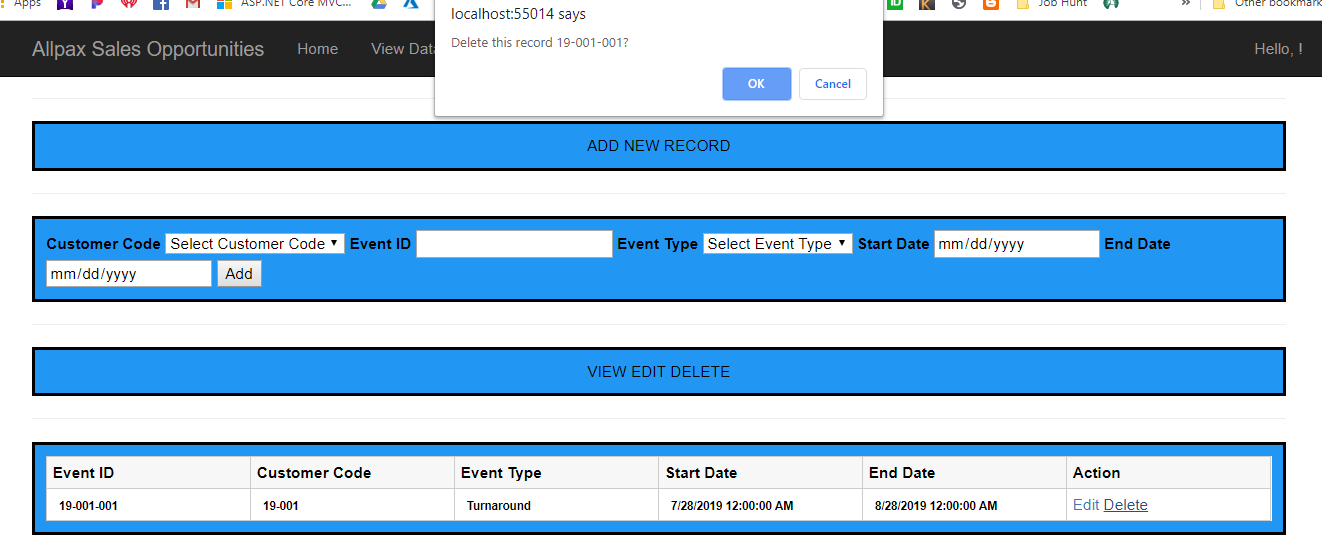
19-001-001 has been updated in the OI.



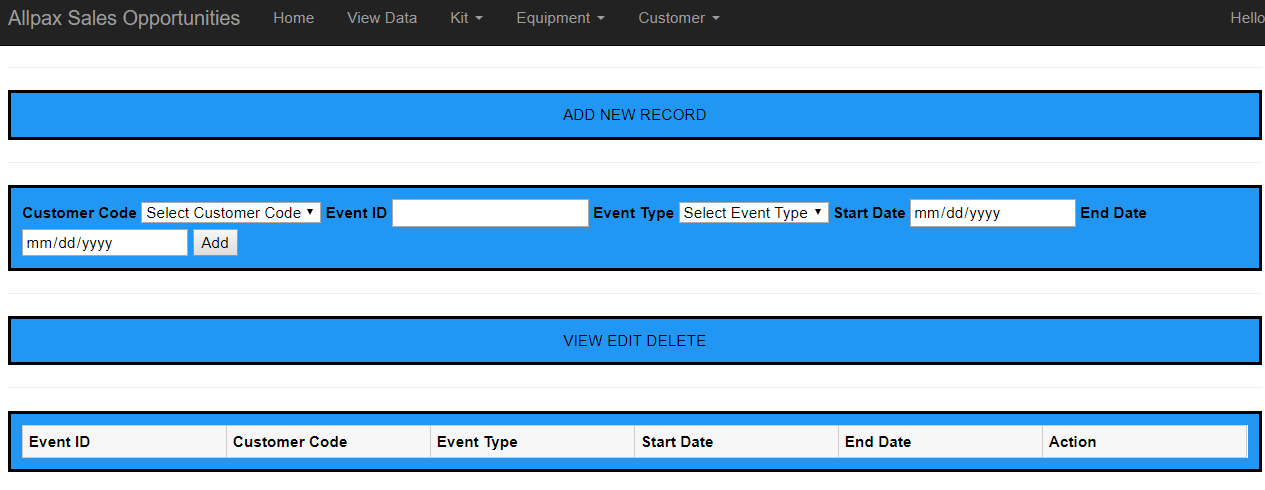


…and the database.

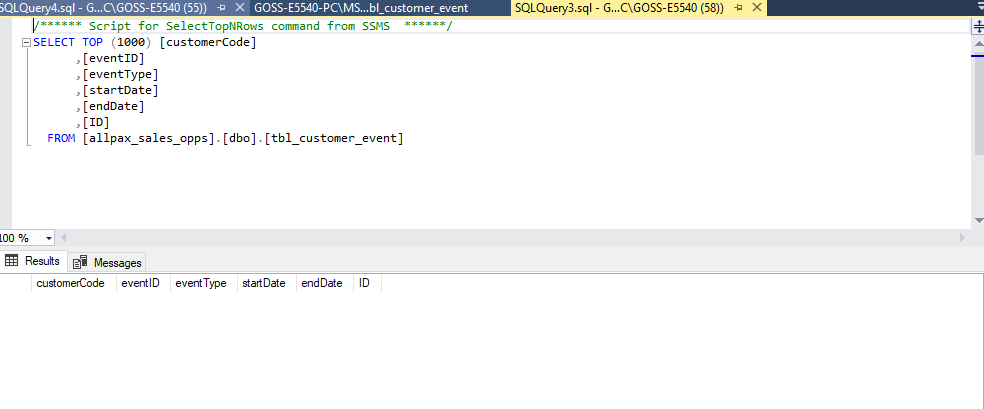
1. Delete a record



OK or cancel.



Pressed Ok. The record is no longer present in the OI….



…or the database.