**APPLICATIONS DEVELOPER: Richard Spector**

**PROJECT MANAGER: Cassie Revis**

**SYSTEMS ANALYST: David Noble**

**TEST ENGINEER: Priya Vemala**

**TECHNOLOGY STACK:**

- Cold Fusion version 9  - selected because this is legacy framework for applications for the GOAST application and for speed of development.

-jQuery Datatables plugin – selected for speed of retrieving queries in comparison to Cold Fusion

-Oracle 12c -selected because it is the legacy back end database

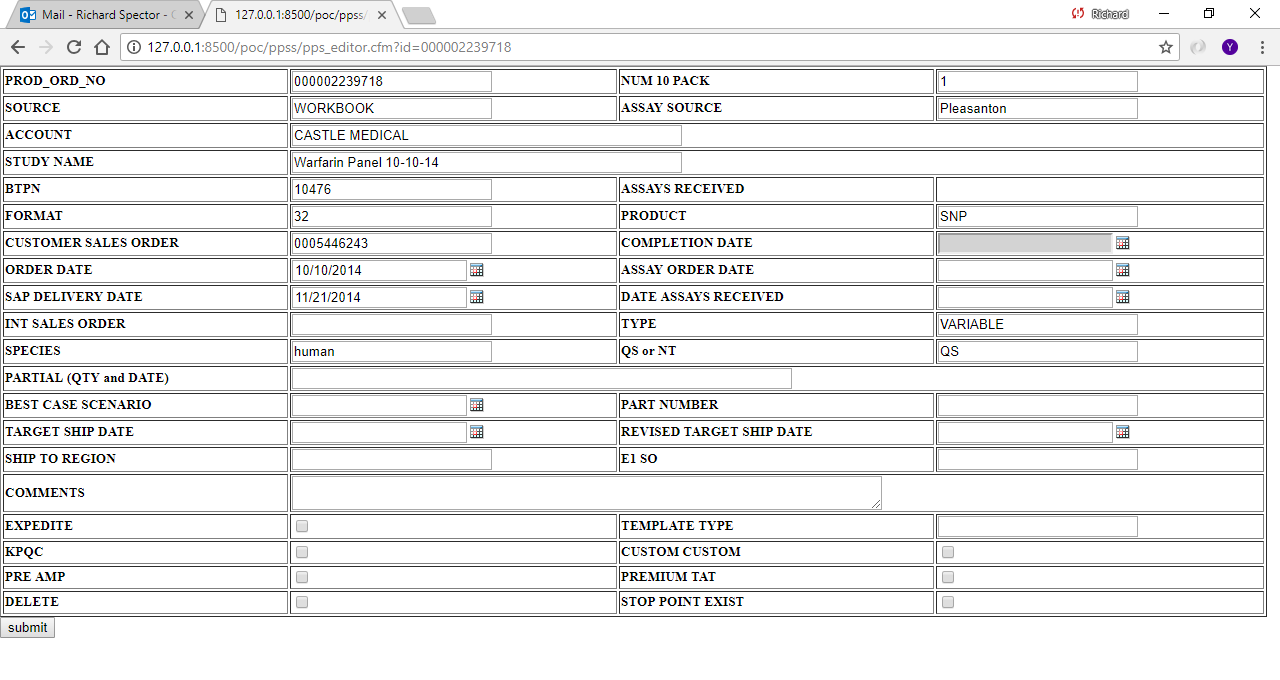
-Microsoft Excel – all output from Jquery Searches is downloadable to Microsoft Excel

**USER STORY HISTORY:** The PPSS application was developed and presented to a department at Life Technologies in Pleasanton in 2015. The purpose of the demonstration was to suggest a way of transitioning out of the current work book excel based PPSS application. A set of production orders came in to the GOAST Oracle Cold Fusion application via a web methods service from SAP. Another set of orders were just ‘workbook’ orders meaning someone entered into the PPSS workbook from customer contact and then capturing a large set of entry fields into the workbook record.

This prototype was suggested as a way to bridge the gap between the two applications, using Cold Fusion to retrieve from the Oracle backend data and also replace the excel workbook as a way of entering the other set of orders.

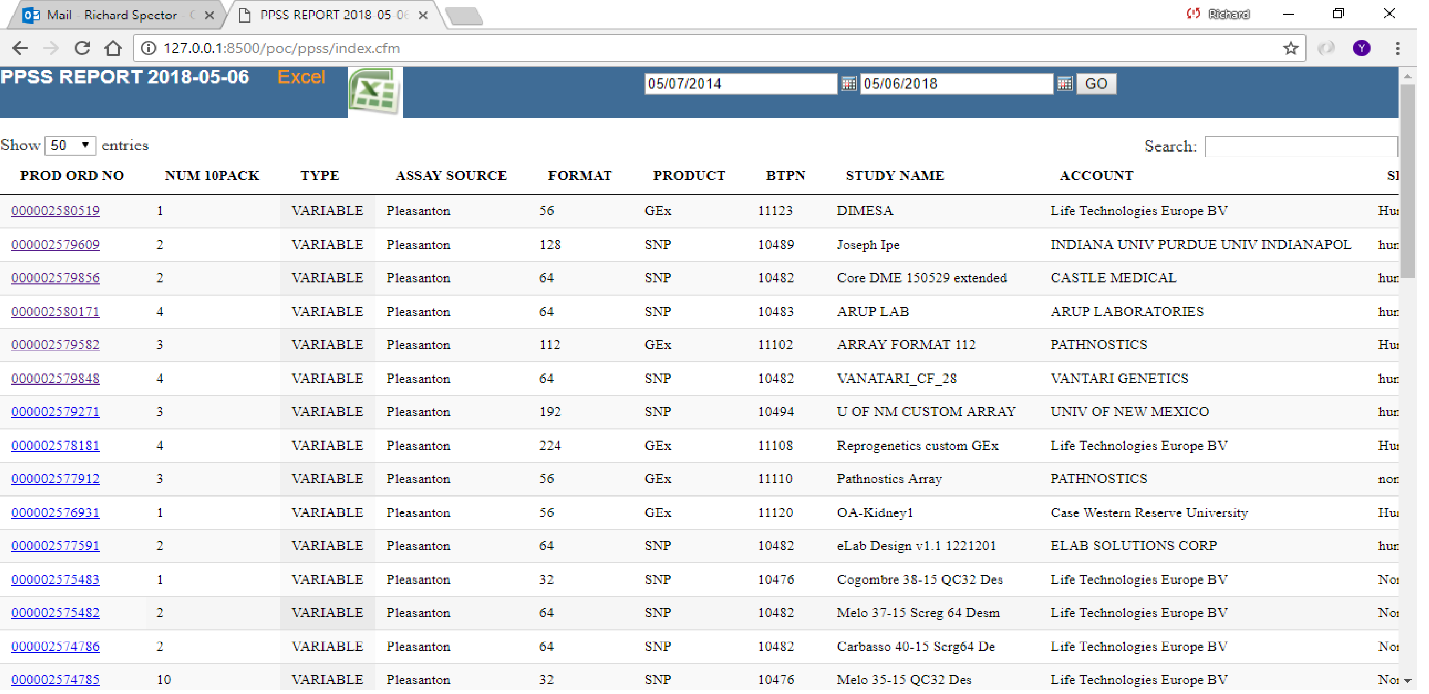
**ORDER ENTRY/ EDIT**

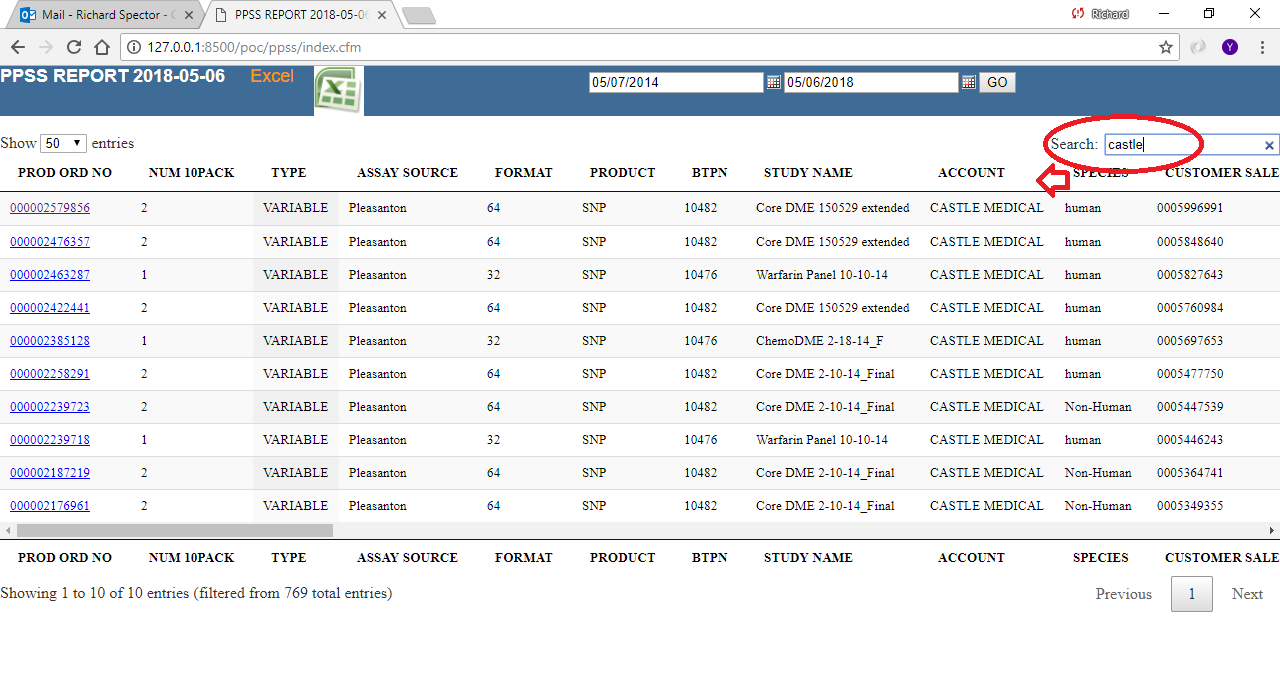
The order entry and edit forms are the same module. Below is the form in edit mode.



**BROWSE GRID INTERFACE**

The front end uses jQuery with the dataTables plug in to present over 10 thousand records of data, rapidly searchable through the AJAX methods.





Downloading the excel through the icon returns the same data in .xlsx format.

The Jquery data grid proved to be superior in performance to native coldfusion cfgrid

For following reasons

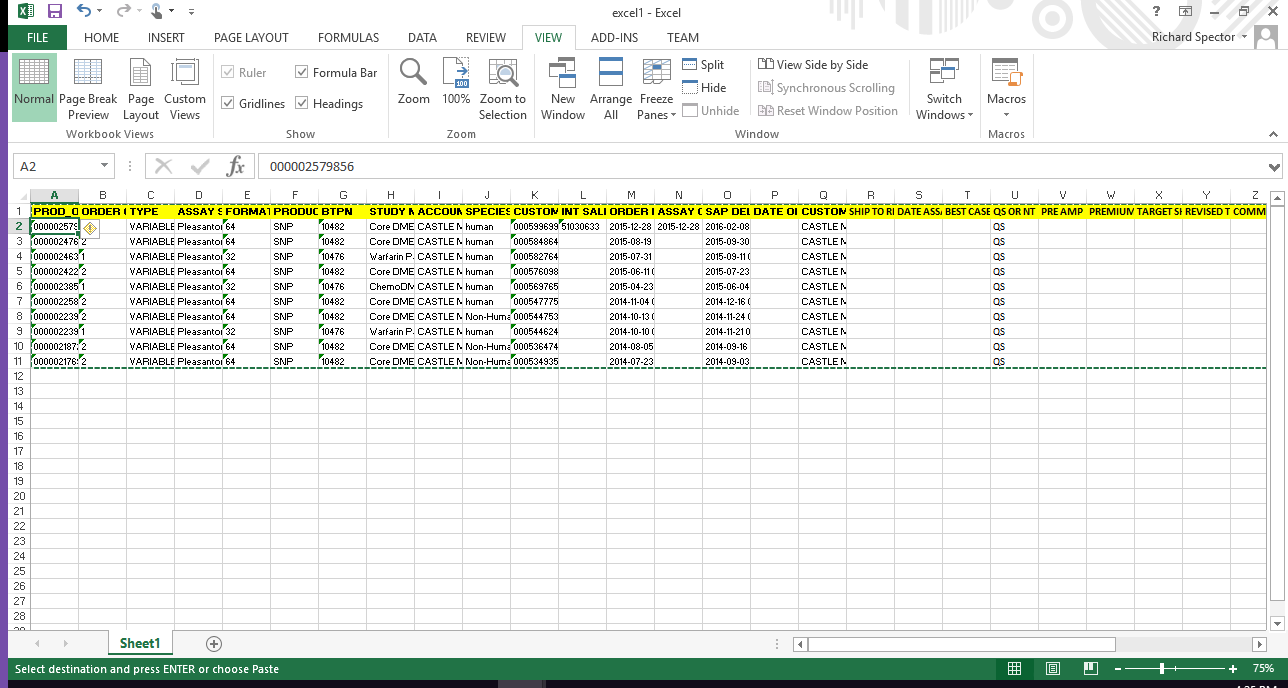
1] speed in retrieving all records.

2] searches all fields in the returned object for requested string.

3] jquery javascript responds to alphanumeric characters returning search results as characters entered.

The edit record passes the form structure contents to the objects edit data method which then iterates through the form structure object and initializes the checkbox and date field fields (through cfparam) to null in case no data was sent in the form. (The edit screen is illustrated on Page 2 previous)

The Excel download is WYSYWG of the currently filtered data. For example if data were to be filtered down to the SEARCH string Castle, 10 records are quickly returned which could then be saved into an excel workbook through the interface.



Code is attached (spreadsheetinc.cfm) that illustrates the *spreadsheetnew* and associated cfscript functions to populate rows of data. Using the spreadsheet function *spreadsheetformatrows ,* the format of the spreadsheet output can be configured to freeze rows and columns, colorize the background and font color of specific rows, bold and set font size to the designated rows.

Additionally a user defined function written in cfscript is employed to force specified columns To text format and override the default behavior of interpreting columns that are numeric.

The code that is attached is the version 1 which employs coldfusion modules. Version 2 was introduced to convert all modules to components and 1 snippet of that code (ppss.cfc) is attached.

Version 2 is also improved by caching all component modules in the application scope, thereby creating an object in the singleton design pattern.