Project Title: **Pharmacy2U Desktop Notification and Management System for Deaconess Hospitals.**

Project Engineer(s): **Jim Allen**

**Work Completed to Date ~ Approximately 70 hours:**

The project was restarted over the Christmas 2019 break, with emphasis made on restructuring significant portions of the code. The first six weeks of this project have focused on setting up the new generic data models and implementing the Model-View-ViewModel (MVVM) design pattern as cleanly as possible. This restructuring was done to take advantage of templating, interfaces and generics that will be used for pieces of the overall project (not being completed as part of the CS495/497 sequence). A condensed list of the major accomplishments this semester so far are as follows:

1. Since restriction to the live data is not possible, a MockDataServices set was created that utilizes an IDataService interface to facilitate the creation of database data. This interface creates the database table values of the schema provided by the Deaconess Business Development Office (BDO). IDataService will be inherited by the EntityFramework (EF) which is responsible for communicating with the live databases when the live connections are made at the end of the project. The IDataService will be responsible for loading, filtering, and saving data to a specific service (such as a the MockDataService, the live database service, or even to and from a local file (as part of a backup system).
2. Setup Dependency Injection system for accessing the main application object (and its view models) from anywhere within the user interface. This eliminates the repeated need to traverse the visual and logic trees used by WPF applications when creating the bindings for the data objects between the view and view models (a major part of the MVVM design pattern philosophy).
3. New data models were recreated that merge data from the databases (P2U\_XXX) of the database schema into custom models. A wrapper class was made around the EF models, to allow templating and creating of the custom application data model objects that are used by the view models and views of the application.
4. New view models such as the OrderListViewModel were made for displaying the objects of the P2U\_Order table. Selecting an order from the OrderListView now displays the information in a Details window through the creation of the FullObjectOrderViewModel.
5. FullOrderObjectViewModel compiles the data from the different tables of the IDataService into a single order record. This view model is also responsible for computing pricing, subtotals and tax on taxable items. This object was necessary since the existing database schema has no foreign keys at this time, and a manual merging of information from the P2U\_XXX tables of the EF model was needed in several locations.
6. Clicking on objects such as customer information, pharmacy information, or specific items within the Details window launches a new edit windows that will be used for editing of the data. The structure of this display system was recreated using data templates and generics and will be reused for the various editing features not included in this CS495/497 project (such as inventory management and specific database update capabilities needed by the pharmacist).
7. Printing capabilities for multipage FlowDocuments proved to be no small feat. Both detail views and delivery manifests can be printed now! A minor issue with document scaling prior to printing needs to be revisited, but the generic printing code is completely functional.

**Changes to Project Specification**:

No significant structural changes to the project specification have been made. However, several minor items were requested and implemented:

1. Ability to print a full order record if desired by the pharmacist.
2. Ability to print a modified delivery manifest for the delivery company that does not include pricing or specific delivered item information.
3. Several requests regarding window layouts and UX experience have been made, particularly with regards to search features and the printing

**Updated Final Project Goals**:

The project goals as they pertain to CS495/497 remain unchanged. As a deliverable, the order management system and the active alert monitor will be implemented as described in the final project proposal from CS495.

**Plan for Remaining Work**:

To reach the final project goals outline in CS495, the following items still need to be addressed.

1. Final assembly of the edit window views for the different editing options of the detailed order view.
2. Search features for the order list view will be completed in early March.
3. The active alert monitor will be created and implemented in mid to late March.
4. Professional polish to improve the UX of the application will be finalized in late March / early April. These features include animations, a uniform (and more appealing) color and font palette, splash screens, and generally layout and feel of the application.