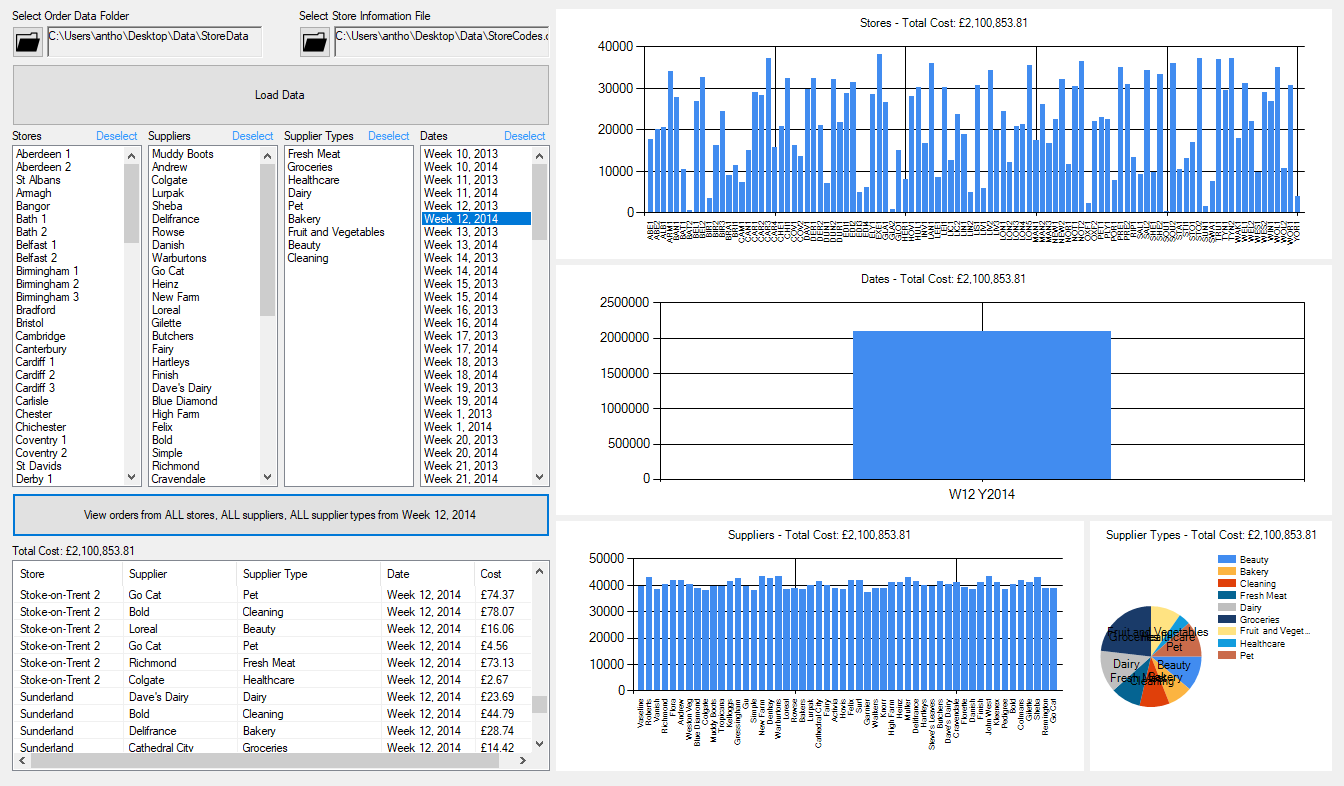
Task-Based Software Engineering Assignment Documentation

Anthony Sturdy / Staffordshire University



# Application Design

The design of the C# application was iterated upon multiple times throughout development, for a balance of speed, features and to improve usability. I wanted the application to be easy to use, and for the user to know how each feature works without having to read about it.

The initial design of the application was as follows:



This iteration was to test loading the data. The data directories were specified, then the List Boxes were populated with the information. The List Boxes are used to filter the data as the user needs to. For graphing the results, the original idea was to have another form open when the user decides to view the data. I decided against that and went with a one-window approach, as can be seen in the next iteration:



This iteration sees the addition of WinForms’ Charts on the right of the window to graph the data. When thinking about how to graph the data I was struggling to thing of a good style of chart which would allow me to present all four elements, so decided to go with four separate charts. Three of which are column bar charts, and the final is a pie chart. A pie chart would’ve better represented how the cost is split between the different options (e.g. Groceries, Dairy, Beauty, etc. Stoke-On-Trent, Brighton, Wolverhampton, etc), but with there being so many options for Stores, Suppliers and Dates, using a pie chart did not look good and was unreadable, which is why it is only used for the Supplier Type which only has a small amount of options available.

The next iteration has only a small change but improved the data loading speed drastically.

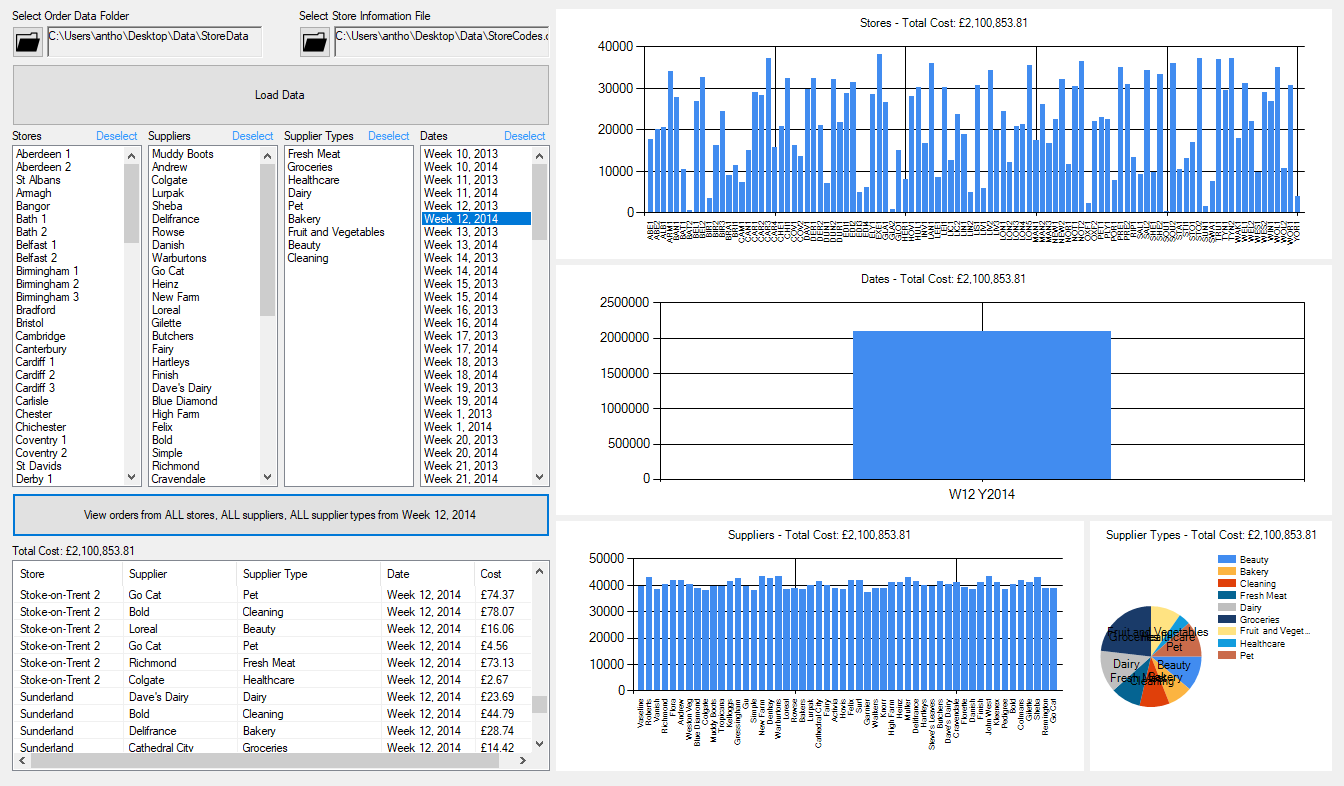


The removal of the loading bar enabled me to increase the speed of data loading from an average of 25.876 seconds to an average of 8.808 seconds, meaning the time to load the data decreased by 65.96%. The reason for this increase was that updating then refreshing a UI element for each loop caused it to slow down considerably. Although I think the loading bar was a great addition to the application, with such an increase of speed I did not think it was worth it.

Lastly, the final iteration of the application allows the user to see a List View of the filtered data alongside the charts. The charts were good to quickly see where the money spent is being split up but didn’t show individual orders for a more detailed view.



This extra feature completes the application, allowing the different filters to be applied and graphed straight away, and for each order to be individual inspected by the user. The above design can be compared to the final application:



In this screenshot, the user is viewing all data from Week 12 in 2014. When different filters are selected, the ‘Show Filtered Data’ button text updates to show what filters will be applied.