**LIST OF PLANNED IMPROVEMENTS**

1. Remove the ‘random’ colors’ from the pie charts to bring a more attractive and uniform appearance to the UI.
2. Add the ability drag and drop a file to load it.
3. Make it so you are able to click on any part of a row in the treeview to display the pie/bar graph.
4. Use the arrow keys to quickly switch between data points in the treeview.
5. Prevent the user from creating multiple custom sorts with the same name. We plan to implement a warning that says name is in use and to choose another name.
6. In the Error Edit Dialog, once you add data to a cell with missing data, the data cell should turn green to show it is valid data instead of staying red.
7. If you fill in some of the missing data but not all of it, you can choose use to use the data you have fixed and discard the rest.

**HOWE WE INTEND TO IMPLEMENT THE IMPROVEMENTS**

1. Remove the ‘random’ colors’ from the pie charts to bring a more attractive and uniform appearance to the UI.
   * The PieChartWidget class’s method: setData() takes a parameter of a “QVector<QColor> colors” which is randomized. Instead of passing in a randomized “colorList” from the main. Instead we will have the colours set directly in the PieChartWidget class so that all pie charts have the same uniform look.
   * The color schemes that will work best for this interface will be chosen by the Golden Ratio Color Scheme Algorithm. The colours selected are far from each other and will never repeat.
2. Add the ability drag and drop a file to load it.
   * Along with using the file picker to get the file path. We can use QT’s built in “void dropEvent(QDropEvent\* event);” in our mainwindow.cpp to support allowing the user to drop files into our program.
3. Make it so you are able to click on any part of a row in the treeview to display the pie/bar graph.
   * Currently the bar charts are only displayed when you click on a member name in the TreeView. We plan to modify our TreeVIew on\_click even to make it so no matter where you click in a row, it will display the data for that row.
4. Use the arrow keys to quickly switch between data in the treeview.
   * We also plan to modify our mainwindow.cpp to utilize QT’s QKeyEvent class to allow the user to switch between data.
5. Prevent the user from creating multiple custom sorts with the same name. We plan to implement a warning that says name is in use and to choose another name.
   * This will be added directly to the CustomSort Class and we will add a QMessageBox to tell the user that the name is in use and that they need to select another name.
6. In the Error Edit Dialog, once you add data to a cell with missing data, the data cell should turn green to show it is valid data instead of staying red.
   * We will implement this in the ErrorEditDialog class. When the user inputs data into a cell and makes the cell valid, we will use the QTableWidgetItem setBackground method to change the colour to green.
7. If you fill in some of the missing data but not all of it, you can choose to use the data you have fixed and discard the rest.
   * This will be implemented in the ErrorEditDialog class. During the on\_save\_clicked() method, instead of just immediately displaying an error when the user has not filled out all missing items. We will instead offer an option to save the changes the user has made and discard the remaining invalid items.