Advanced Programming Assignment 2 – Design Document

# Changes to the Design

To accommodate the additional features of assignment 2, an additional data access class was added that is instantiated through the driver class. This class read in the person and relationships into an array list of person objects and relationship objects.

For efficiency, one of the changes made was that the relationship constructor stored two people objects and a string to describe their relationship (e.g. friends, parents, couple, colleagues and classmates). By storing the person in the relationship, we are able to access their profile characteristics as well.

The UI class which creates the user interface was designed to have a border pane which contains a horizontal box at the top with the menu, a vertical box down the left which contains the list of users to reference when selecting users to make it an easier user experience. All the menu items are mouse click, with a button which is set on mouse click to send the information to the driver class, to complete the attached method in the console, and return a successful or non-successful message to the UI. The gridpane in the centre is where the user types and interacts with.

Each menu item has it’s own method to be called as they need a different scene layout, depending on how many fields have to be entered. Any scene with a text field needs a submit and clear button to send the option to the driver, or clear the values in the scene.

To ensure consistent formatting, a lot of the features such as labels, text, buttons have their own create methods with specific formatting so it all has the same text.

# How are the new classes organised

The new UI class is instantiated through the MiniNet main argument method. This has a private instantiation of the driver class, which has a private instantiation of the database access class. The UI calls the driver class through menu options to complete the specified activity (e.g. adding new user, showing relationship) and the driver class calls the database to access the array list of person and relationship objects to manipulate them.

The relationship exception classes are thrown in the driver class and have try/catch clauses in UI class. As adding each type of relationship has it’s own method in the driver, only the relevant exception to the that type of relationship is thrown, e.g. when adding a colleague, the not available exception is not thrown.

# How does the Program interact with the external data source

The database access class reads all lines from the people text file stored inside the project directory, each item separated by a comma is treated as separate item. These are written to an array list of person objects. The same is done for the relationship text file, however these are written to an array list of relationship objects.

The database access class also then connects to the HSQL server and creates a connection, this connection occurs whether the people file is found or not. If the people file is not found, then the MiniDB is accessed and the result set is read into the array list of person objects.

As the connection is active throughout, the add new person can access the data base and insert a new person row through the driver class instantiation of the access database class.

When the user clicks the exit MiniNet option in the menu, this calls the exit method in the driver class, before completing the system exit, this stops the connection to the server.