**Project A: Customer Maintenance Application**

Carlos Igreja

Project A

Development Phase

This program will allow a user to store data of Customers information. The Customer class can create and store the customer information in a customer object. The Customers information will be the first name, the last name, and an email address. The CustomerMaintApp class will have all the methods that are called by the CustomerFrame class for all the main options of the program.

The CustomerFrame class inherits the JFrame class used to create a GUI. The user can enter data into text field boxes. Buttons correlate to each option that the user has to work with the data and the application. The buttons will include a button called "list" that will list all the customers stored on the file. An "add" button will allow the user to add a customer to the file. A "delete" button will let the user remove a customer from the file. An "update" button will enable the user to edit existing customer information stored on the file. A "help" button will provide a description for all the buttons to inform the user of what functions they each perform. Finally, the "exit" button will let the user close the program when they are finished. I am considering on adding the feature as an applet in the later stages of the programs production.

The CustomerDAO interface inherits three other interfaces. The CustomerReader interface that is used to read data from the file. The CustomerWriter interface is used to write or update data to the file. A CustomerConstants interface is used to associate specific customer data stored as string objects such as name, and email address with integer values to correspond to the string objects length.

The CustomerTextFile class implements the CustomerDAO interface. The DAOFactory class maps the CustomerDAO interface to the CustomerTextFile class that serves as the appropriate data storage mechanism. The information will be stored on either text or xml.

The data input by the user will have to be validated by the Validator class. If invalid data is input then the program will display the appropriate error message to the user. I will be using message boxes for any error messages. The data will also have to display properly so spaces will be added where they are needed using the StringUtils class.

I tried to illustrate as much information about the project with the following UML diagram.

