1. Create the DataHolder.class to save data.

Source code cited: https://docs.oracle.com/cd/E13211\_01/wle/javadoc/m3java/com/beasys/rmi/iiop/DataHolder.html)

* Set value to each field.
* Constructor Detail
* Get info of Customer and Order by ID
* Set string for new order and new customer
* Create two serialized files “Customer” & “Order”
* InitData Function: Read the data first, then check the biggest ID. It is helped add new customer and new order.
* Read-Write data: Saving data of customerlist and orderlist in dataholder.
* Calling get methods of DataHolder in other class.

1. Modify the Customer and Order Class

* Add StringBuilder Method. [StringBuilder](https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html)objects are like [String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html)objects, except that they can be modified. Internally, these objects are treated like variable-length arrays that contain a sequence of characters. At any point, the length and content of the sequence can be changed through method invocations. By StringBuilder Source code cited:

<https://docs.oracle.com/javase/tutorial/java/data/buffers.html>

1. Add Alert pop up window in three controllers (AddNewCustomerController, NewOrderDetailsController, OrdersController)

* By Alert Popup Window Source code cited:

<http://code.makery.ch/blog/javafx-dialogs-official/>

1. Add .clear() method into unfilled to fix the bug of order duplication.