Final Project Report

CSIS 4050 Project Team 7 – Electronic Shop App

Team Leader: Sadaf Sadat Riazi
Team Member: Quintus Mai, Mei Kwan Chan

Project Description and Process Model

In our project, we are proposing an app to help with the ordering process in an Electric and Electronic equipment company. Our application is used by two users:

1. The Administrator
2. The Manager

The Administrator will take the order of electric and electronic products with certain specifications. First, the admin must input customers details at the top of the form. Then the admin chooses the product ordered by the customer based on different criteria such as: category, types, voltage, current and voltage version. The app will have tables and check boxes to make the selection more specific and easier. The admin must also input the quantity of the product ordered into a text box. There will be a button to add the chosen item to a grid view which will display the product name, description, unit price, quantity to be ordered and subtotal for that ordered product. A row of certain products will be added to the grid view every time the admin inputs an item into the list. Just below the data grid view of ordered items, there will be 3 labels to display: order total amount, tax and total amount including tax. At the very bottom of the page there will be three buttons for the admin to save the order, to edit the ordered products or to close the form.

Once the admin clicks on the save button:

1. The ordered items are removed from inventory database based on their quantities and then the inventory is updated.
2. An invoice is generated for customer orders. The Invoice will be a read only file that displays customer details, the data grid-view generated, total amount, tax and total amount including tax.

The manager can view a list of all products with their total quantities within the inventory. If an item is low in quantity within the inventory, then the manager can choose the product specifications and add it to the inventory. One label will show the name of the chosen item and another label will show the description of the item. Once the manager hits the add button, if the item doesn't exist in the inventory, then an item is added to the list and if the item does exist in inventory, then the quantity is changed for that product. And once the manager hits the update button then the inventory is updated with newly added items.

Our app will also have a page for admin to restore and backup data frequently. This will protect our data within the company.

Database Model (Final design)

a) Normalization Process:

Initial Entities: (1) Manufacturer, (2) Electronic Shop Inventory, (3) Customer

Customers go to the Electronic Shop and doesn't know anything about the product, Electronic Shop Inventory as a medium/platform which allows the customer to buy inventories and make the order.

INVENTORY

InventoryId
InventoryName
InventoryQuantities
InventoryUnitInStock
InventoryUnitPrice
InventoryCategory
InventoryType
InventoryCurrent
InventoryVoltage
InventoryVersion

CUSTOMER

CustomerId
CustomerName
CustomerPhone
CustomerAddress
CustomerEmail
OrderId
OrderDate
OrderDetailsInventory
OrderDetailsQuantity
OrderDetailsTotalAmount

Functional Dependencies:

1. InventoryId -> InventoryName, InventoryQuantities, InventoryUnitPrice, InventoryManufactuers, InventoryCountryOfOrigin, InventoryType
2. CustomerId -> CustomerName, CustomerPhone, CustomerAddress, CustomerEmail
3. OrderId -> OrderDate, OrderDetailsInventory, OrderDetailsQuantity, OrderDetailsTotalAmount

Many-to-Many Relationship:

1. Inventory <-> Orders
   One order can have many inventories, and one inventory can also exist in many orders. Therefore, another entitiy OrderDetails as an association table is required.

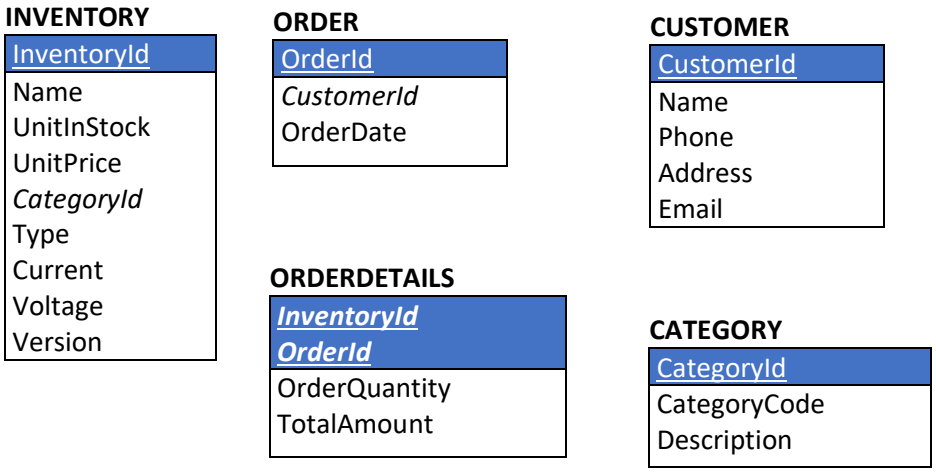
Category Entity:

1. Category entity is required for filtering, like the Department entity in StudentRegistrationDB.

After the normalization process, we have 5 main entities, they are **INVENTORY, CATEGORY, CUSTOMER, ORDER** and **ORDERDETAILS**

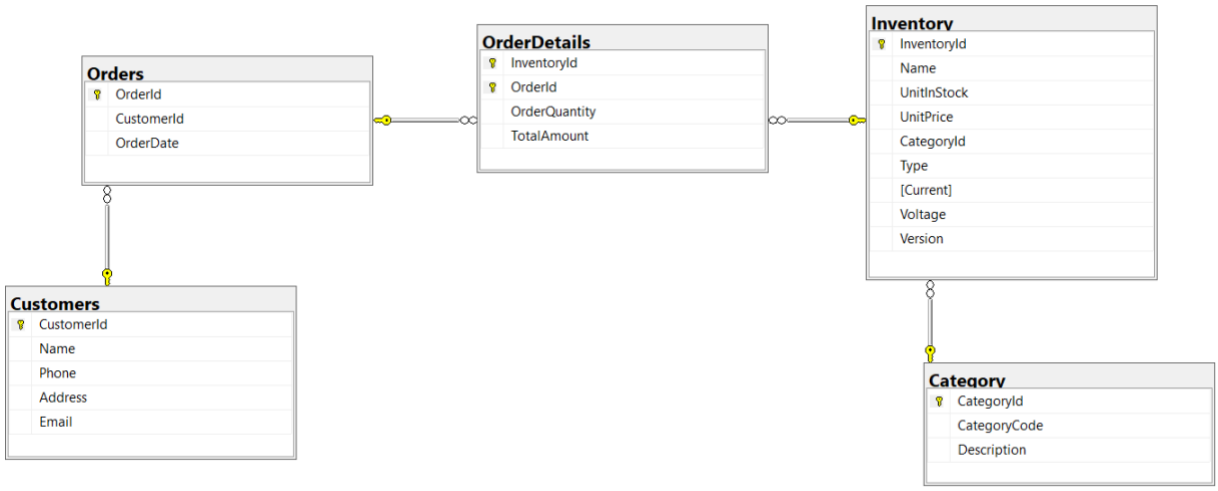
This project includes these entities:

1. Inventory (**InventoryId**, Name, UnitInStock, UnitPrice, *CategoryId*, Type, Current, Voltage, Version)
2. Category (**CategoryId**, CategoryCode, Description), code: LI, RE, EM
3. OrderDetails (***InventoryId, OrderId,*** OrderQuantity, TotalAmount), [composite primary key]
4. Orders (***OrderId***, *CustomerId**,*** OrderDate)
5. Customers (**CustomerId,** Name, Phone, Address, Email)



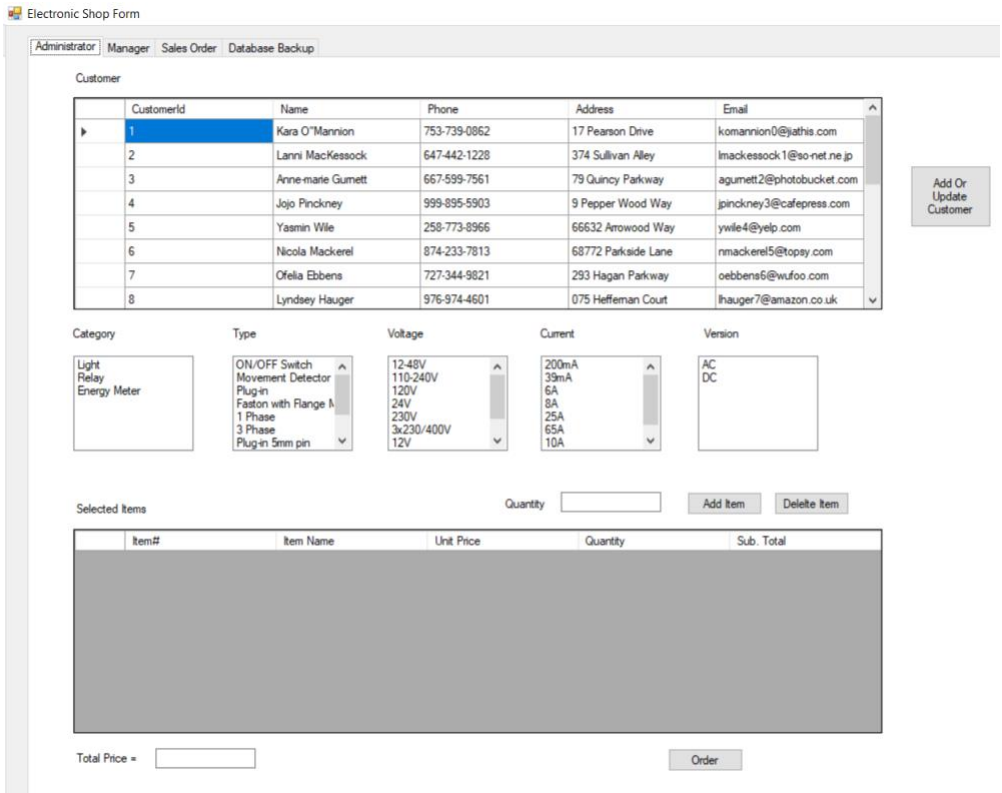
* Remark: Version refer to AC / DC power supply

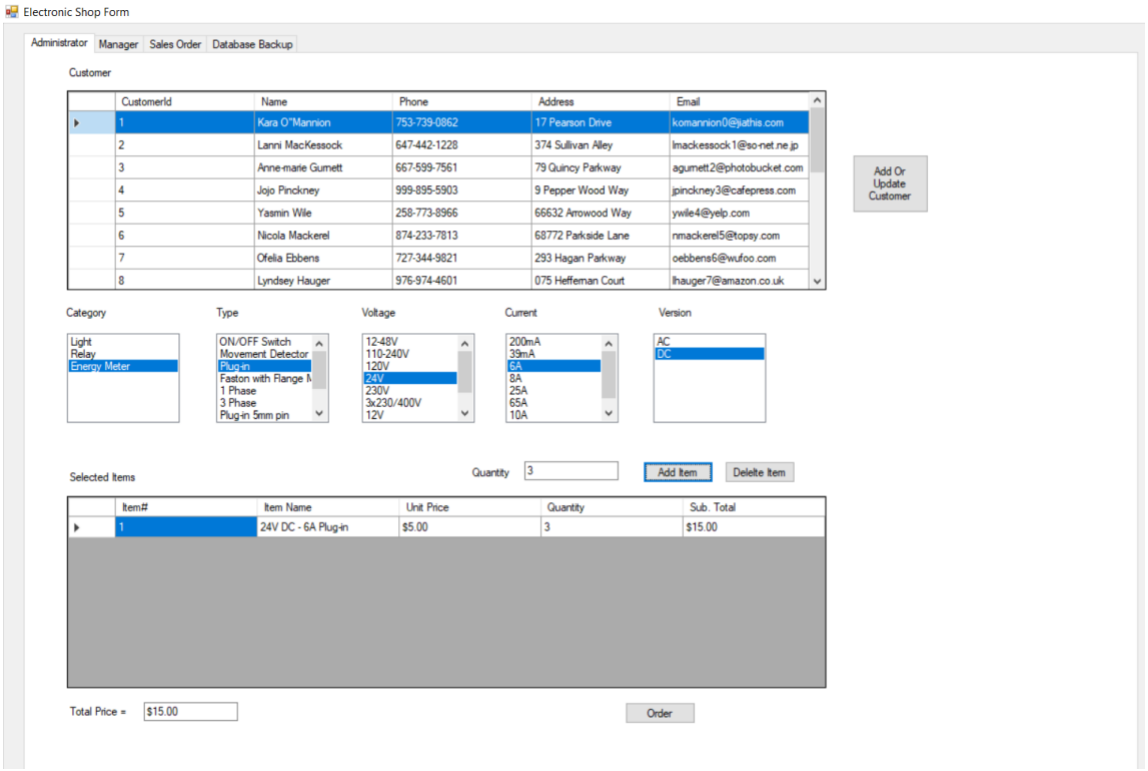
b) E-R Model from the MsSQL Server Database



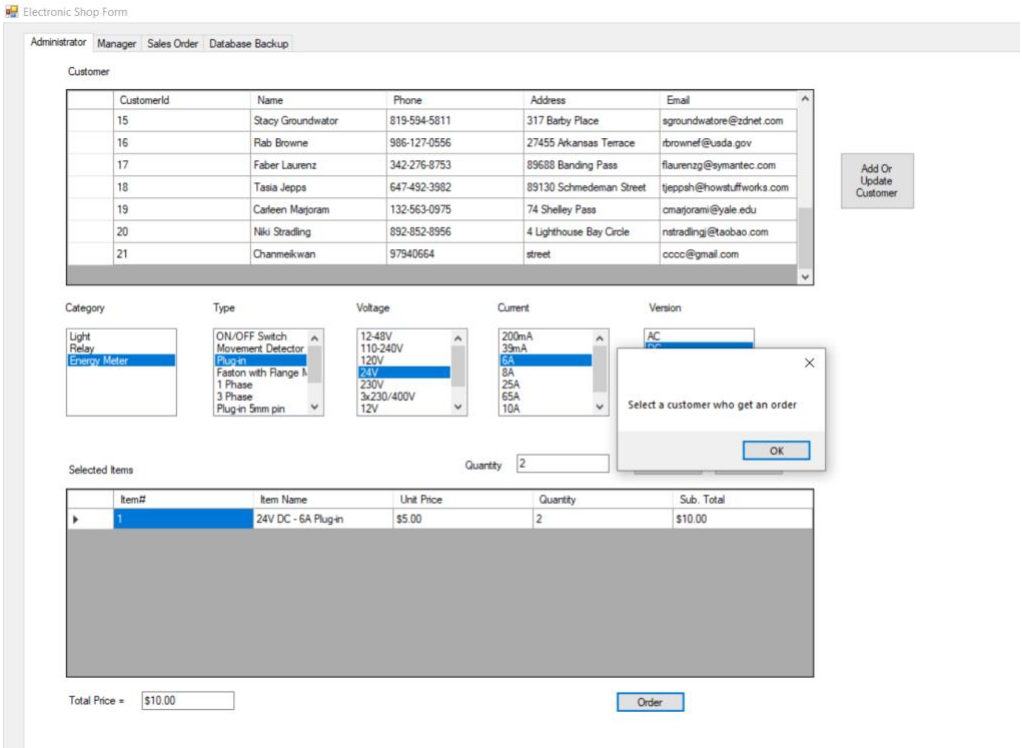
Functionality and Testing

1. Administrator Tab

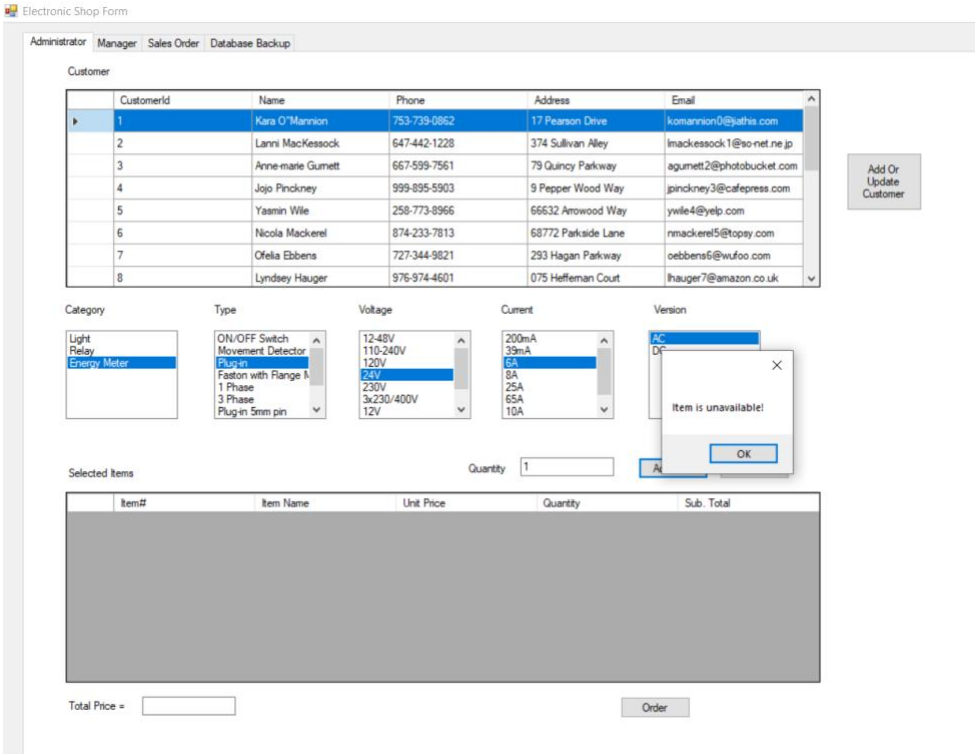




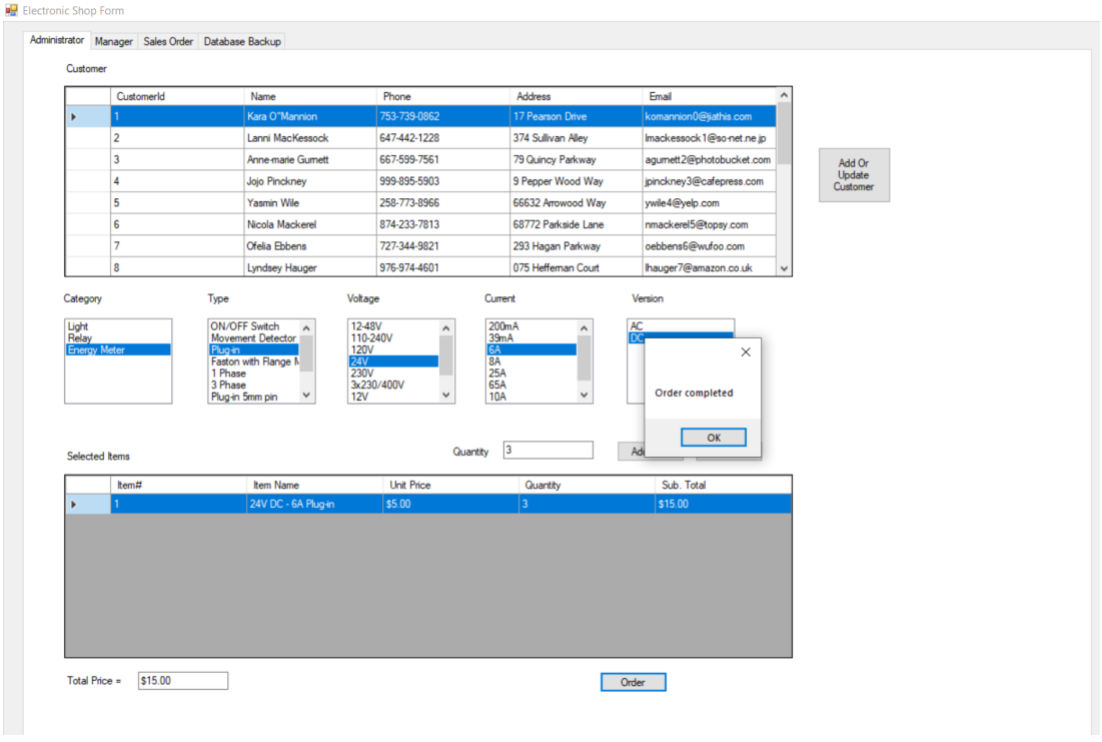
a) Select the customer and select item's category, type, voltage, current, version to add the inventory to the datagridview below.



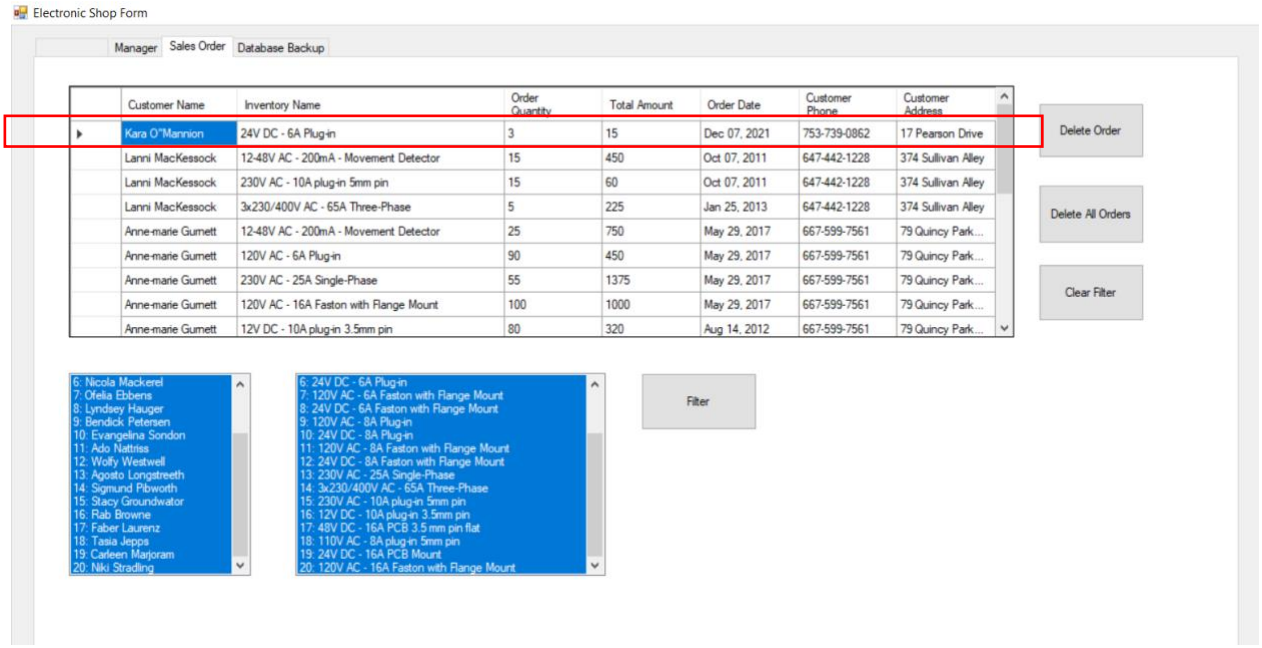
b) Error message pop up if no customer is selected.



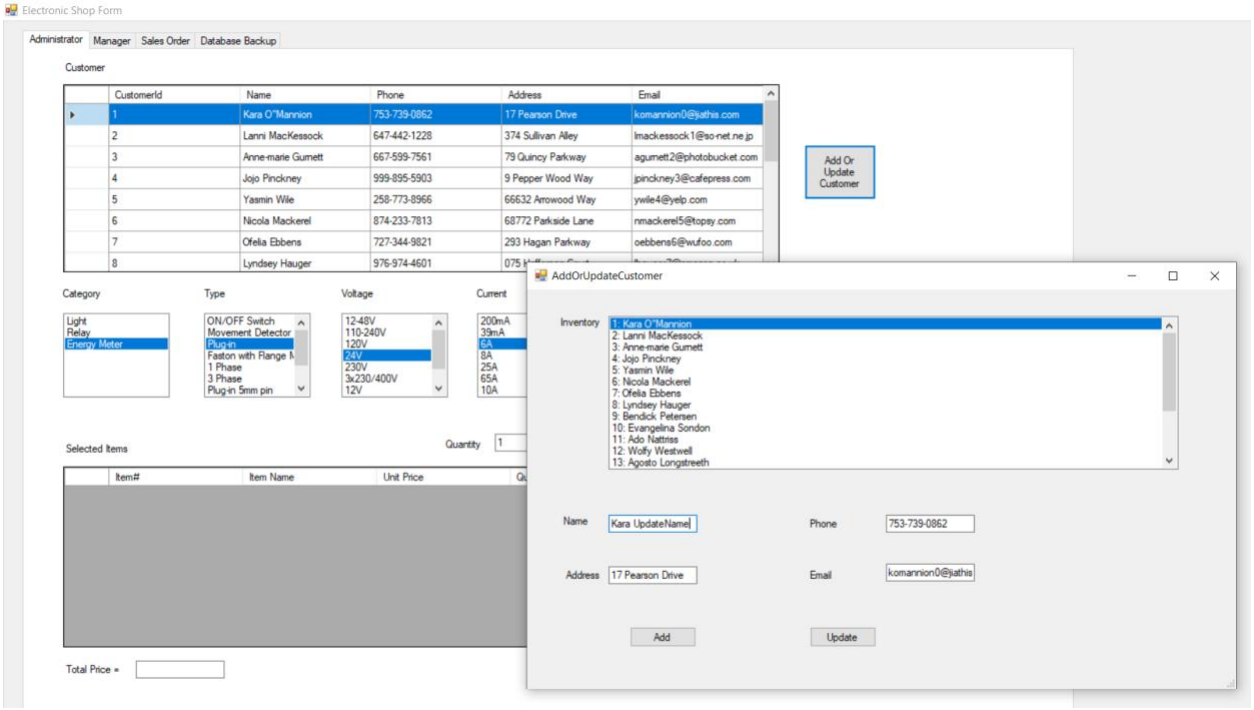
c) Error message pop up if the item is not available.



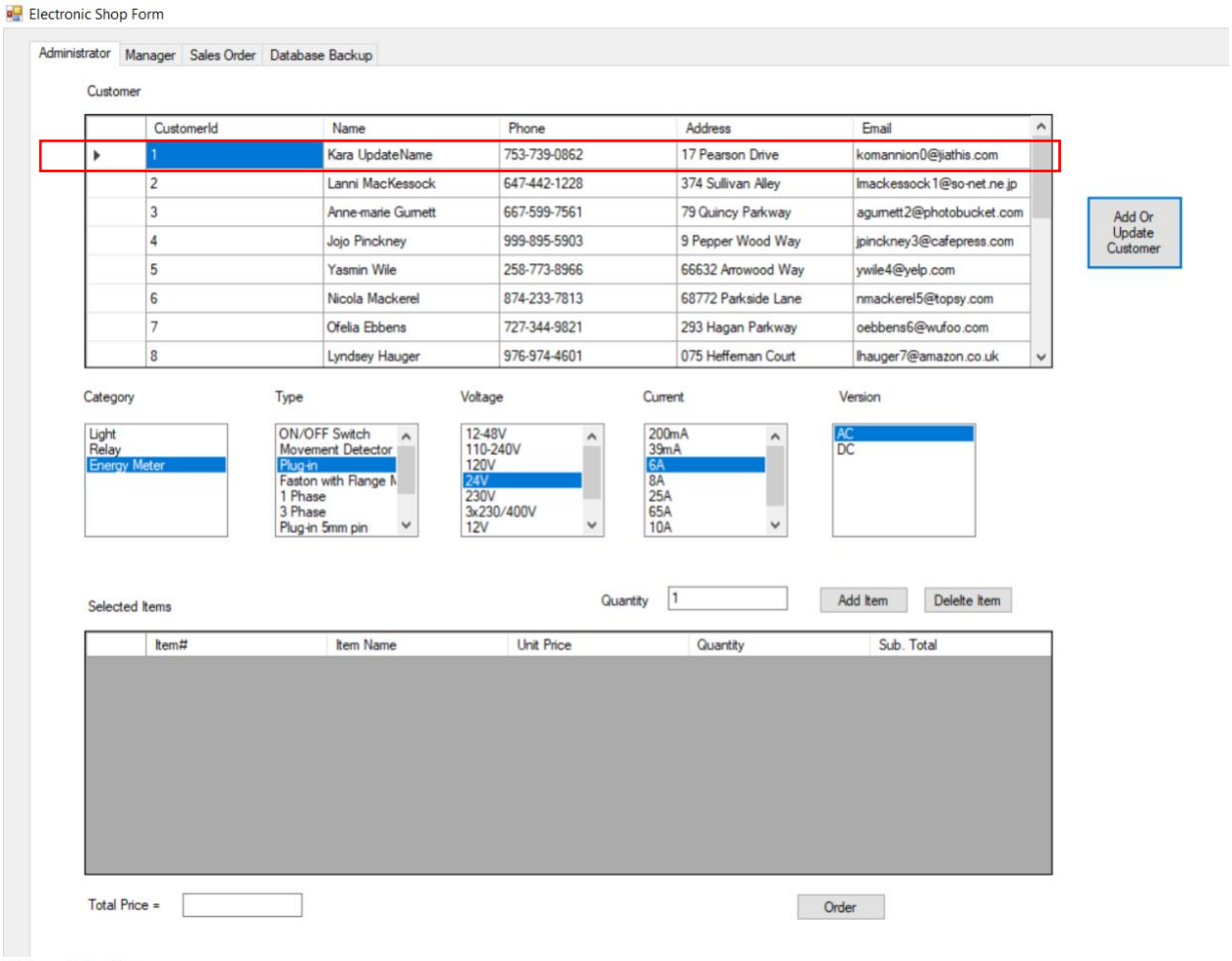
d) Press the order button to add the order, send it to the Sales Order Tab Page.

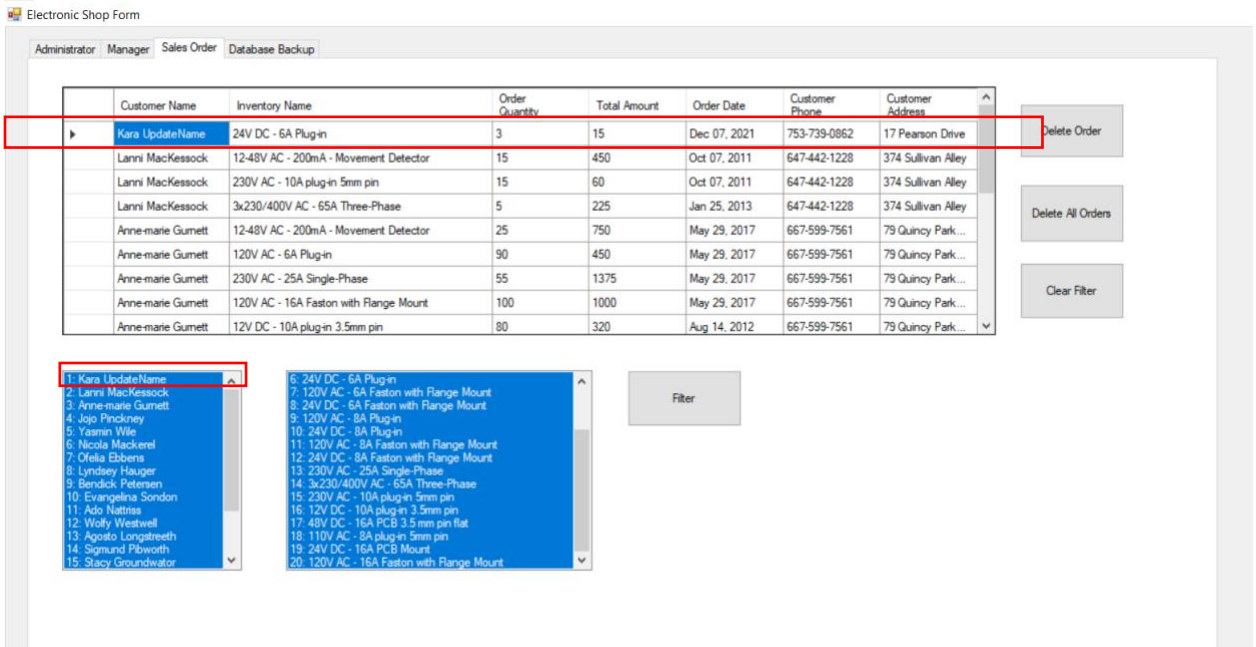


e) New Order added.

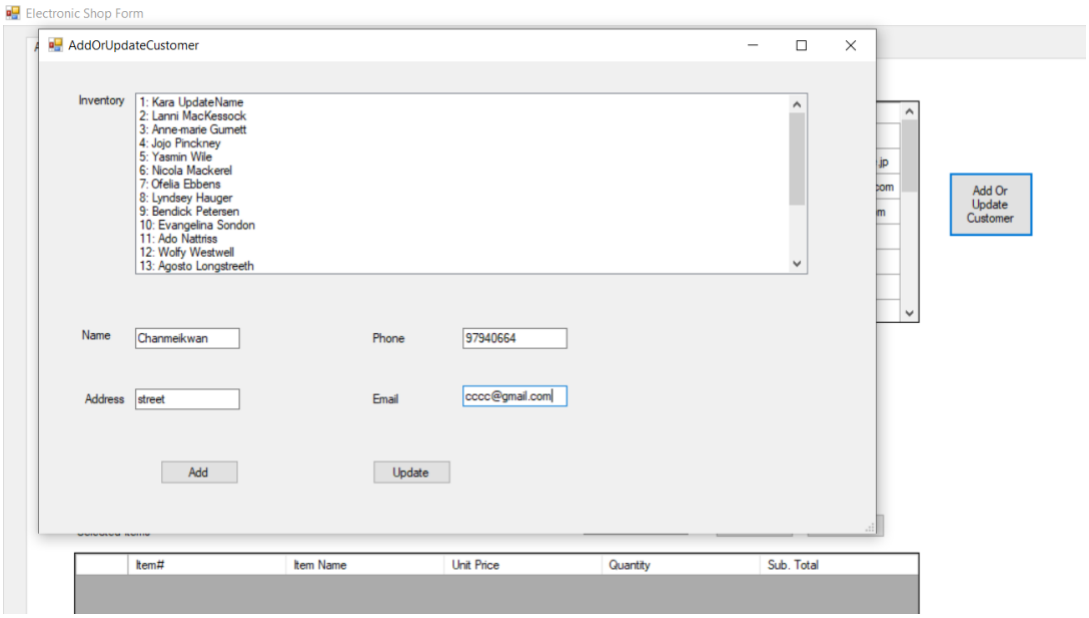


f) Update the customer information, the changes reflected in both Admin and Order Tab.

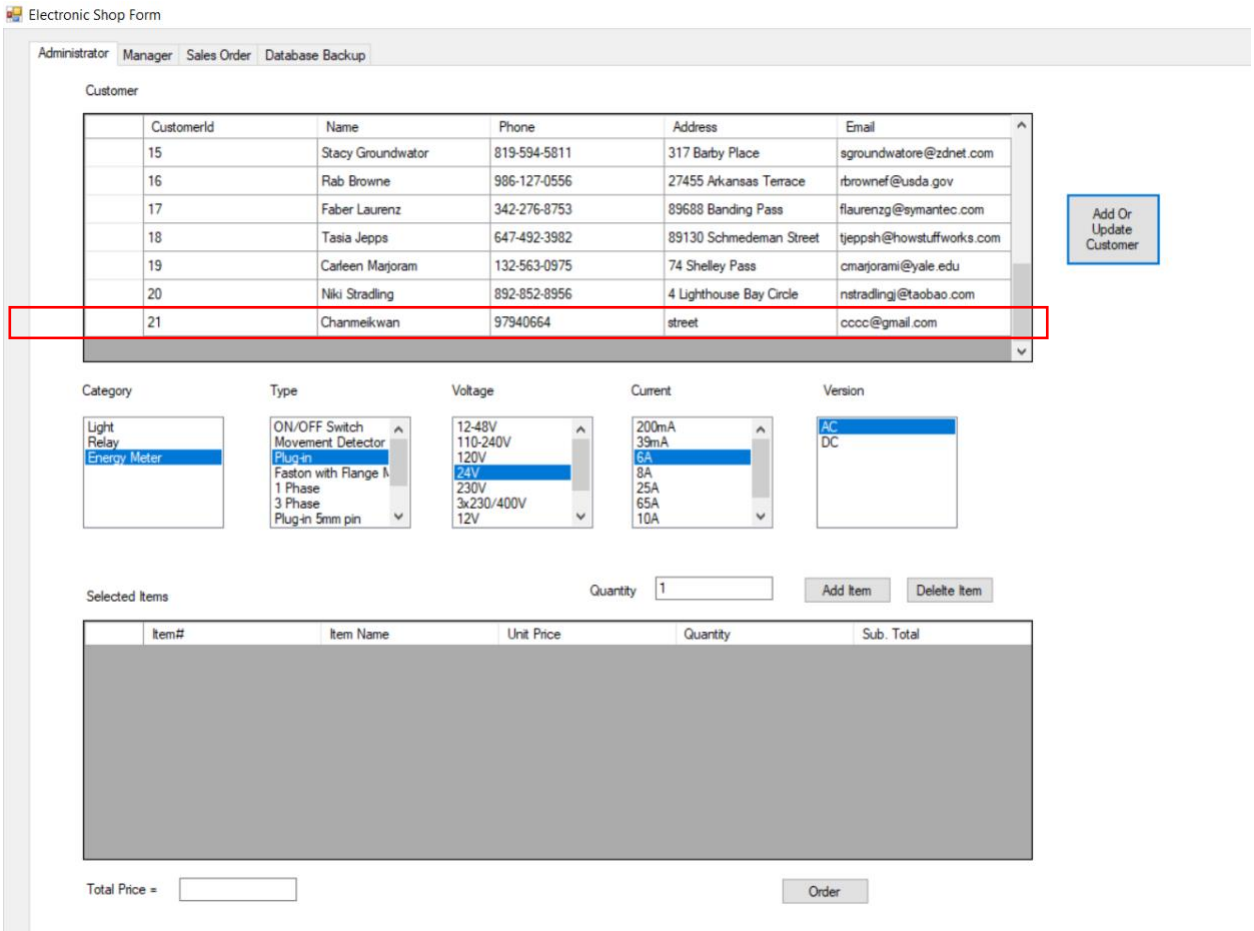




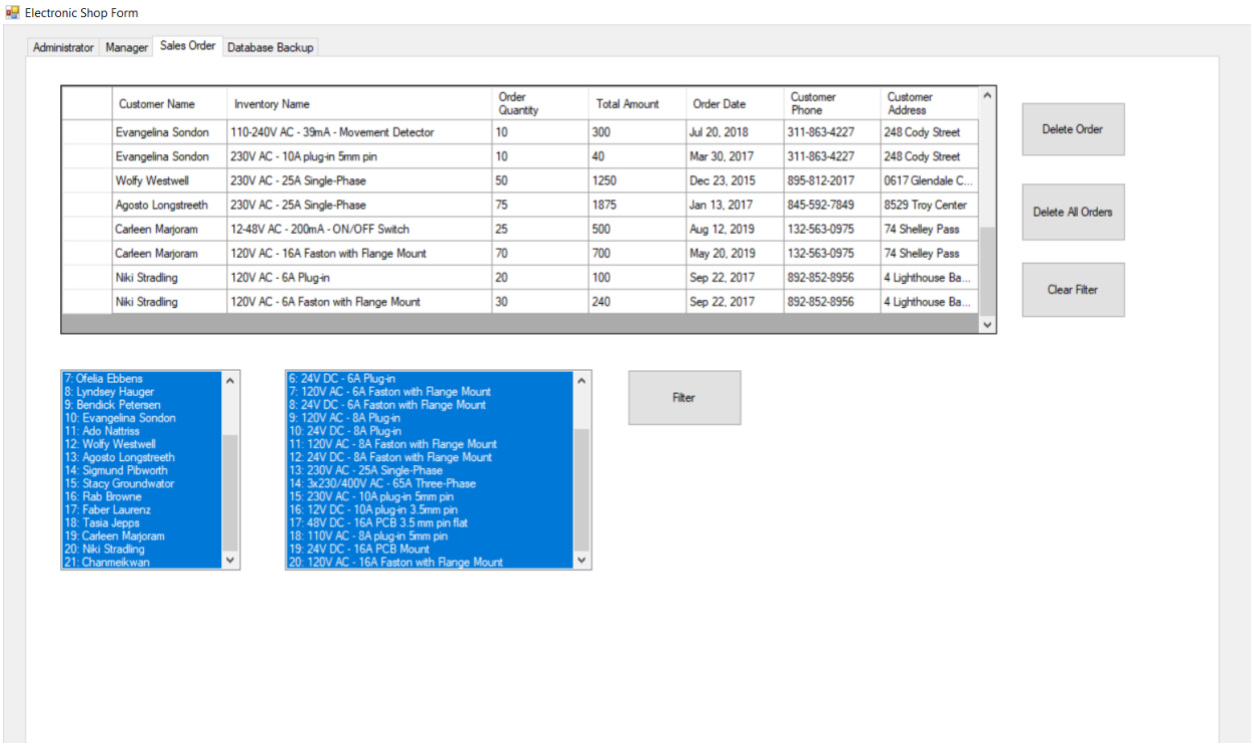
g) Both admin tab and Sales order tab are updated with customer information



h) Add a new customer, input information for new customer and press add button



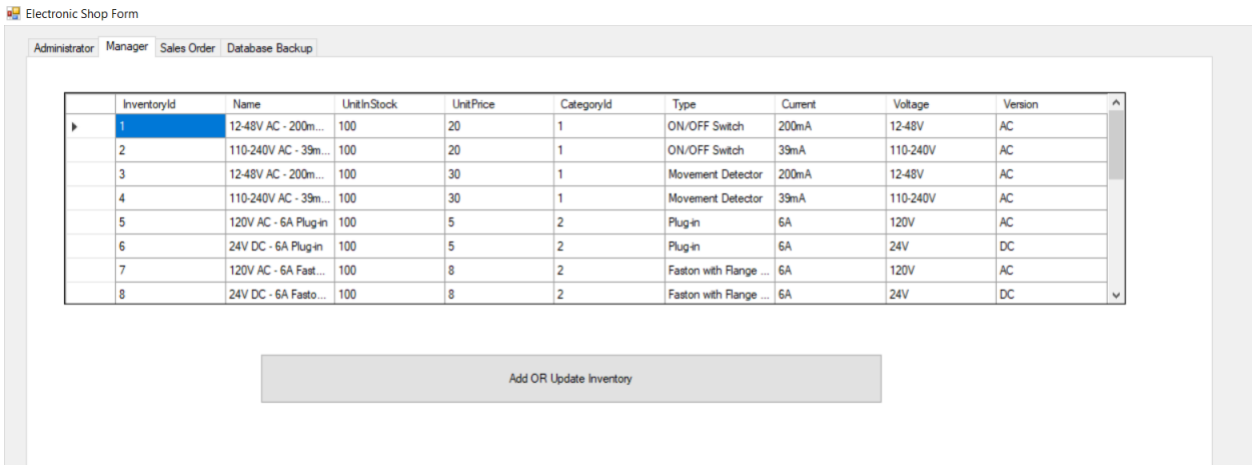
i) New customer is added to customer table in Admin tab



j) Now the new customer can place new order.

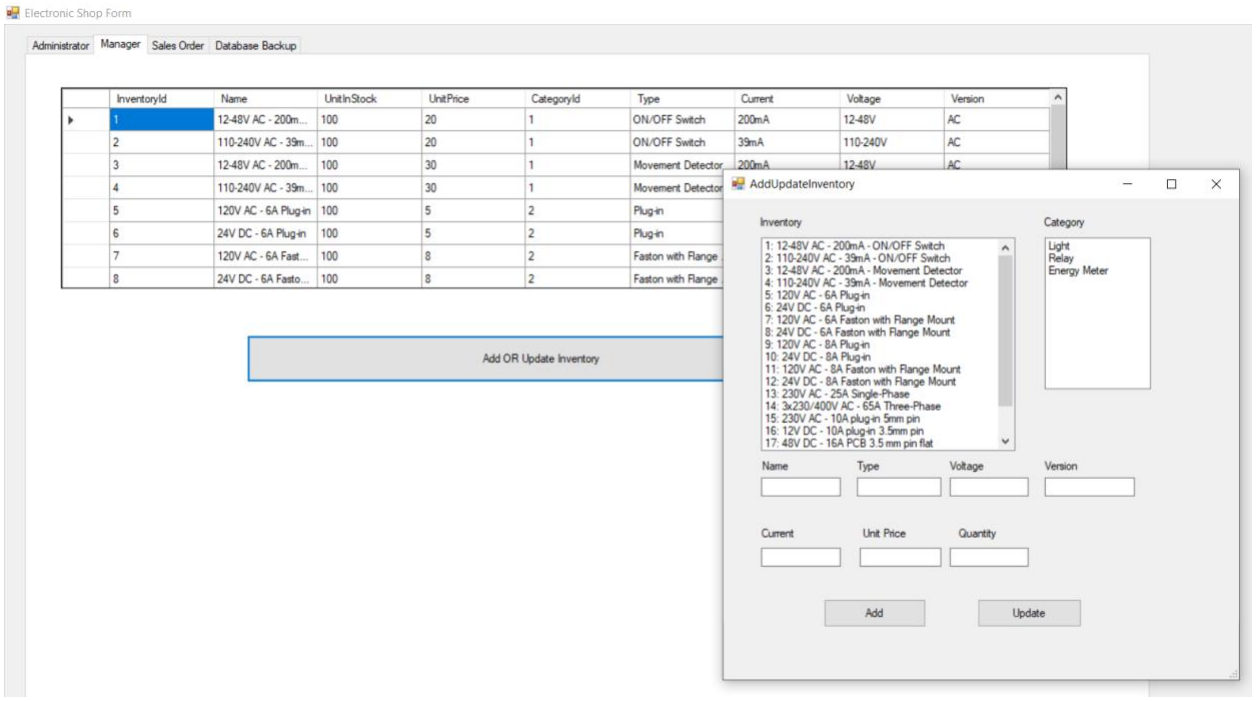
2. Manager Tab

Main Form



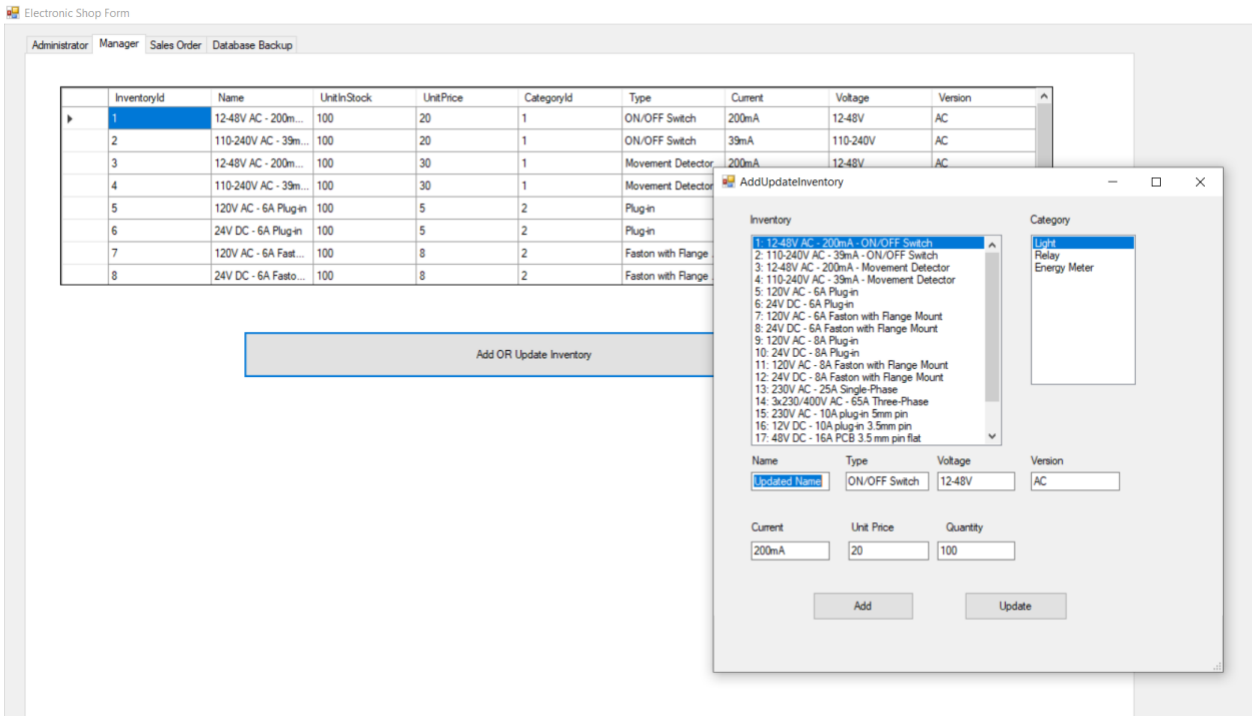
a) dataGridView displays the items in inventory with their unit price and quantity in stock. Press Add OR Update button to add or update inventory. This will open inventory child form.

Add or update inventory child form

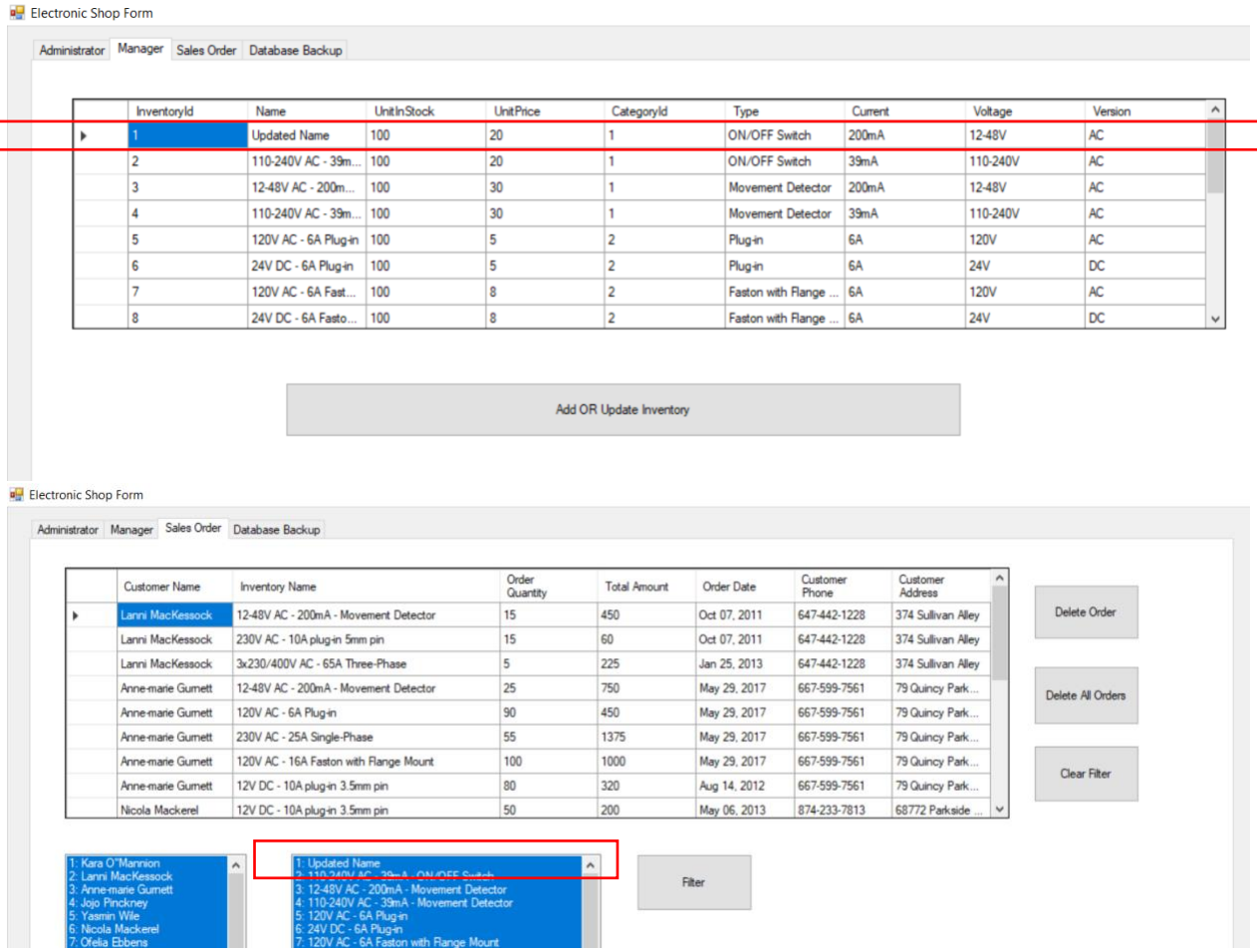


b) Child form inventory display all the items within the inventory in a list box, user can add or update the inventory using Category list box and all related text boxes.

Update the name of inventory, update the name in Manager tab and Order tab as well

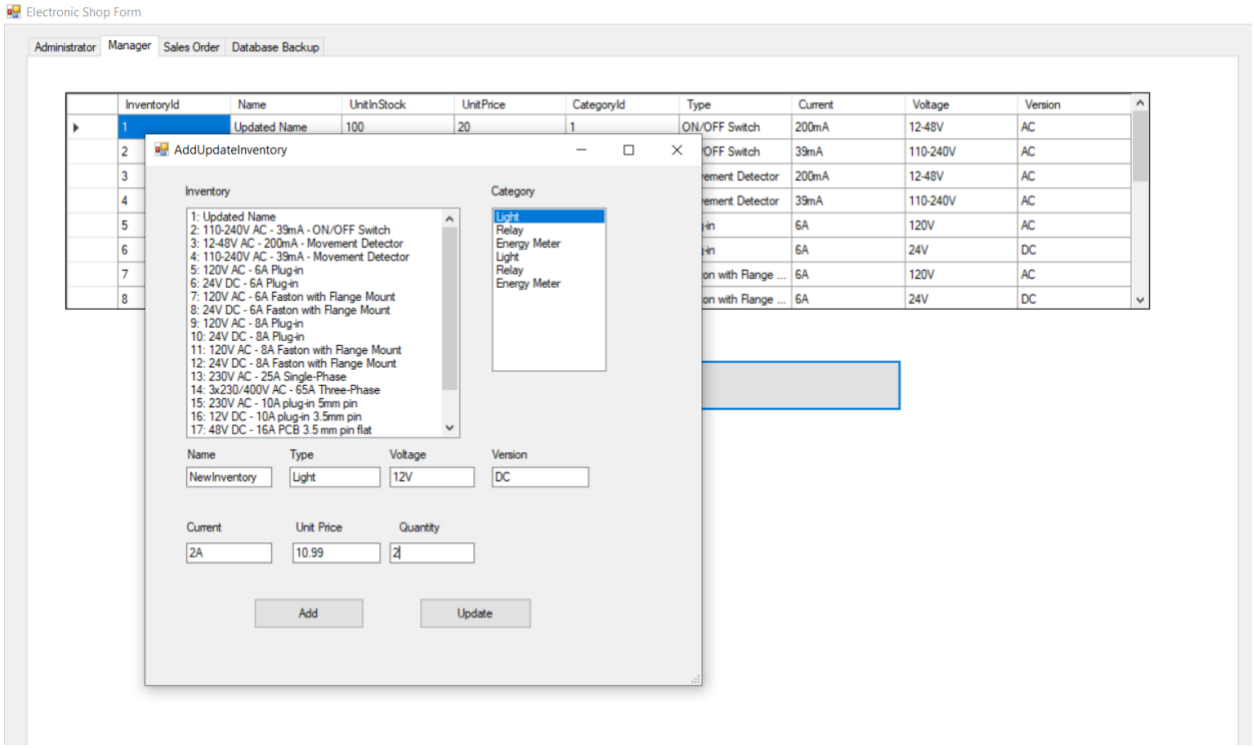


c) To update an inventory, select an item from list box inventory, the information for that product is auto populated into the category list box and all the related text boxes. You can change the value for any of the products and hit update button.



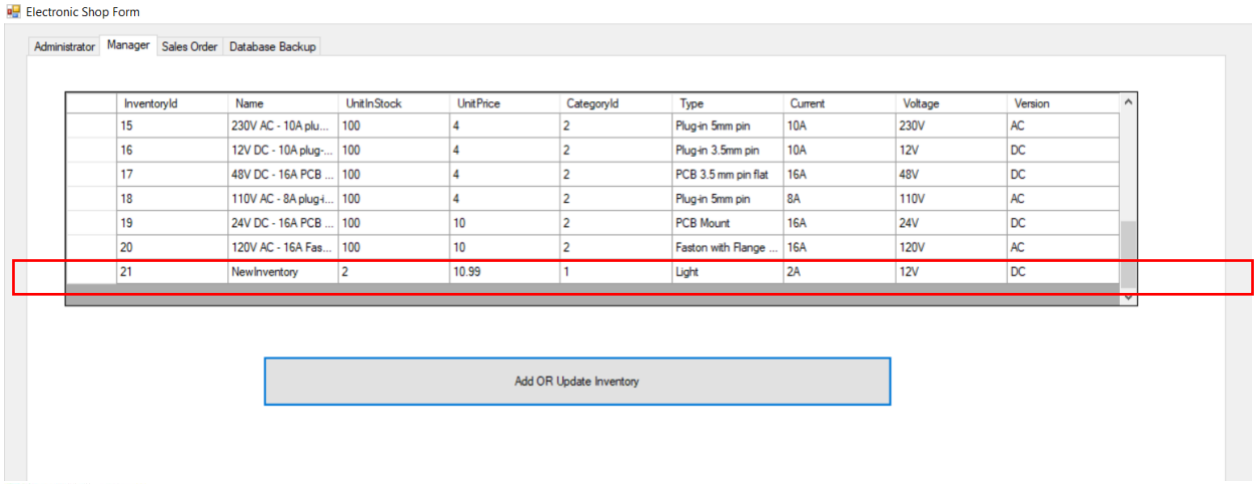
d) Once you hit update button, the specific product information is updated with new value in the inventory table. This reflects product information in both manager tab and sales order tab.

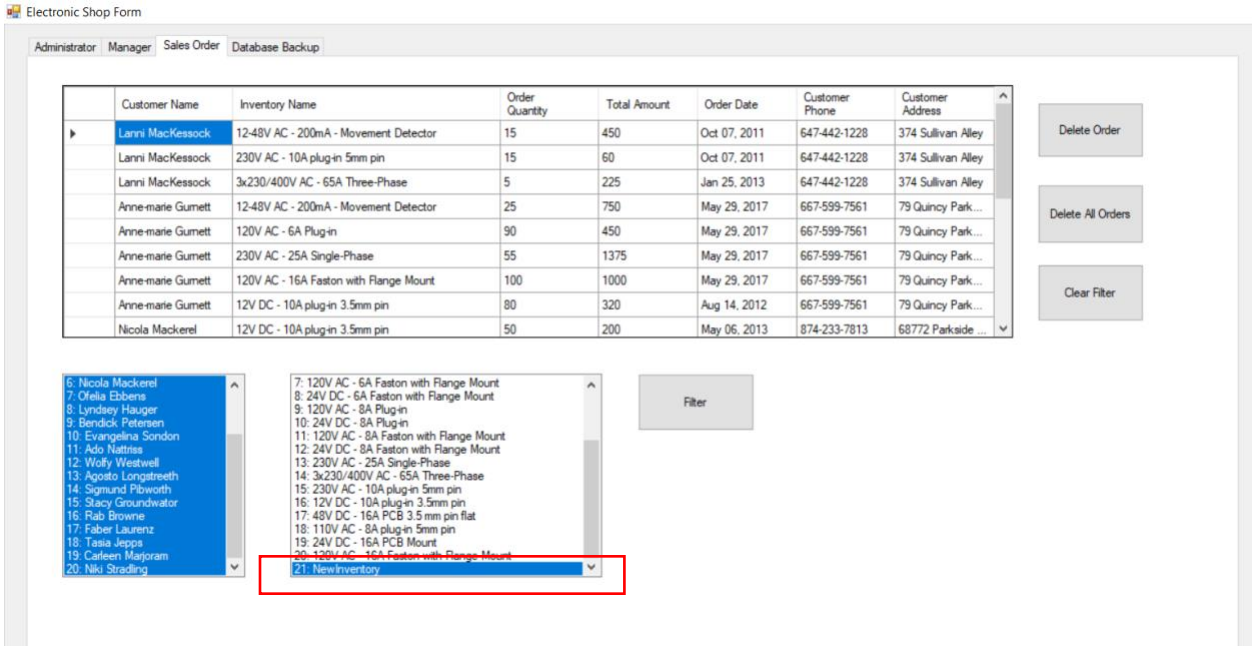
Select the Category from the list box, enter the data in textbox, then Add the New Inventory



e) You can choose a category from list box and input new product information within the related text boxes and press add button to add a new product to the inventory table.

Added the new inventory in Manager and Sales Order tab.

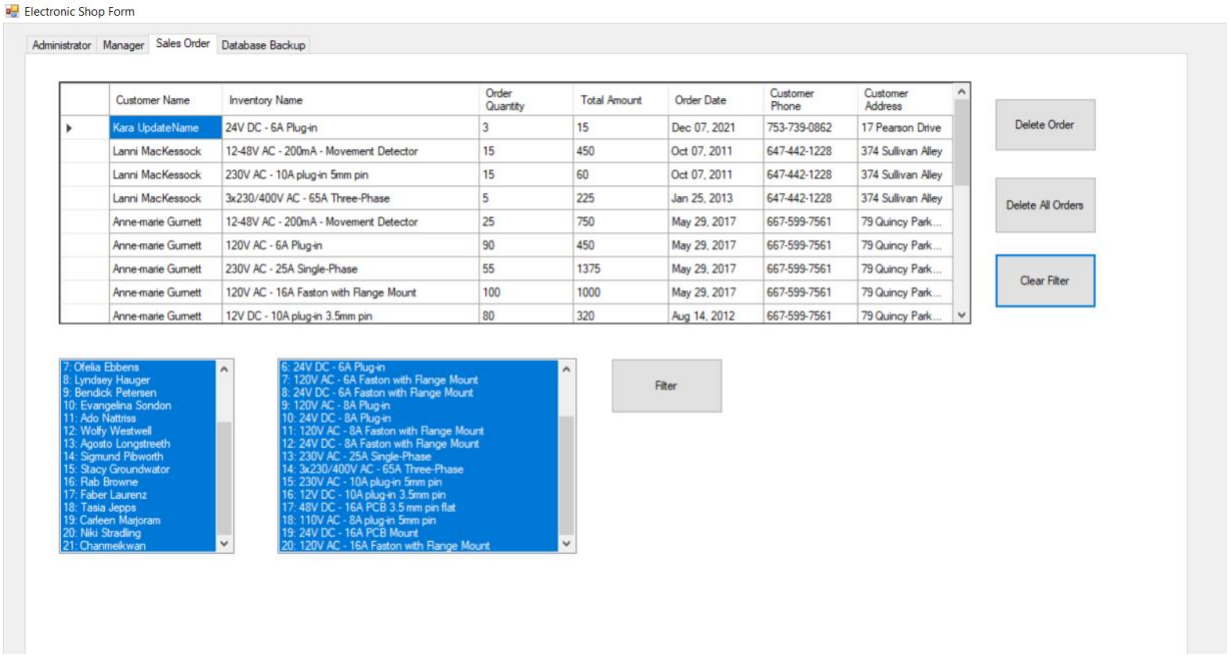




f) Once you hit add button, new product is added to the inventory table, and this will reflect both manager tab and sales order tab. Now you have new product in the inventory.

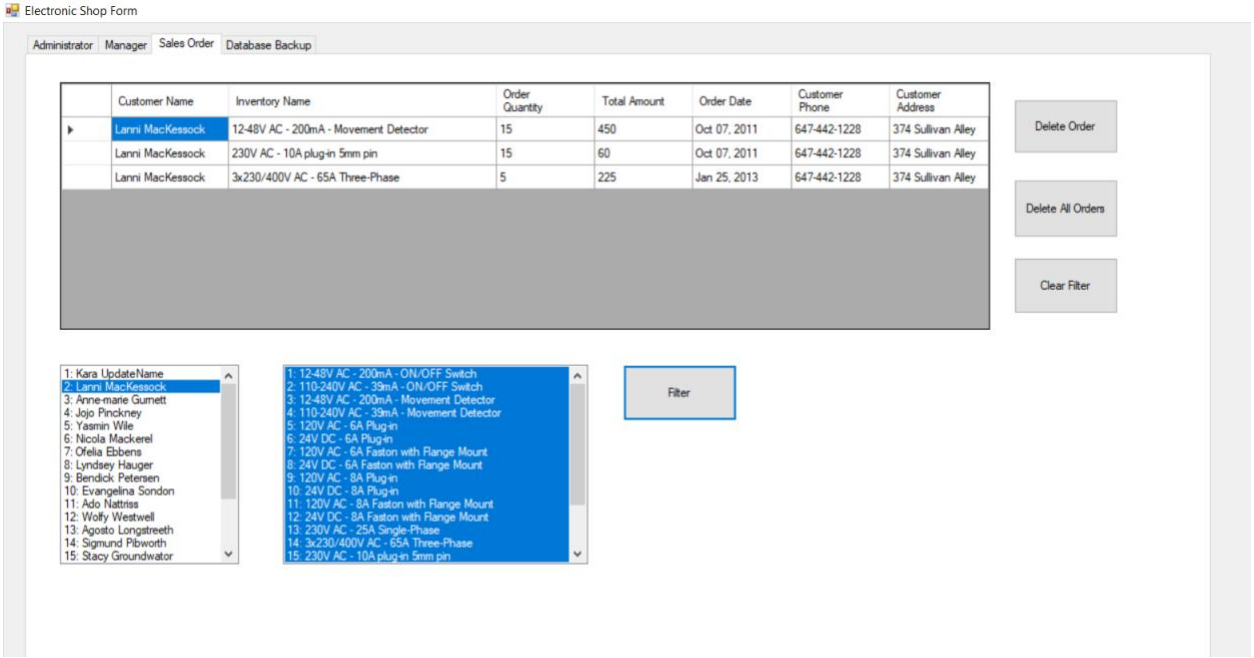
3. Sales Order Tab

Initially, all the customer and inventory in listbox are selected, so it shows all the order



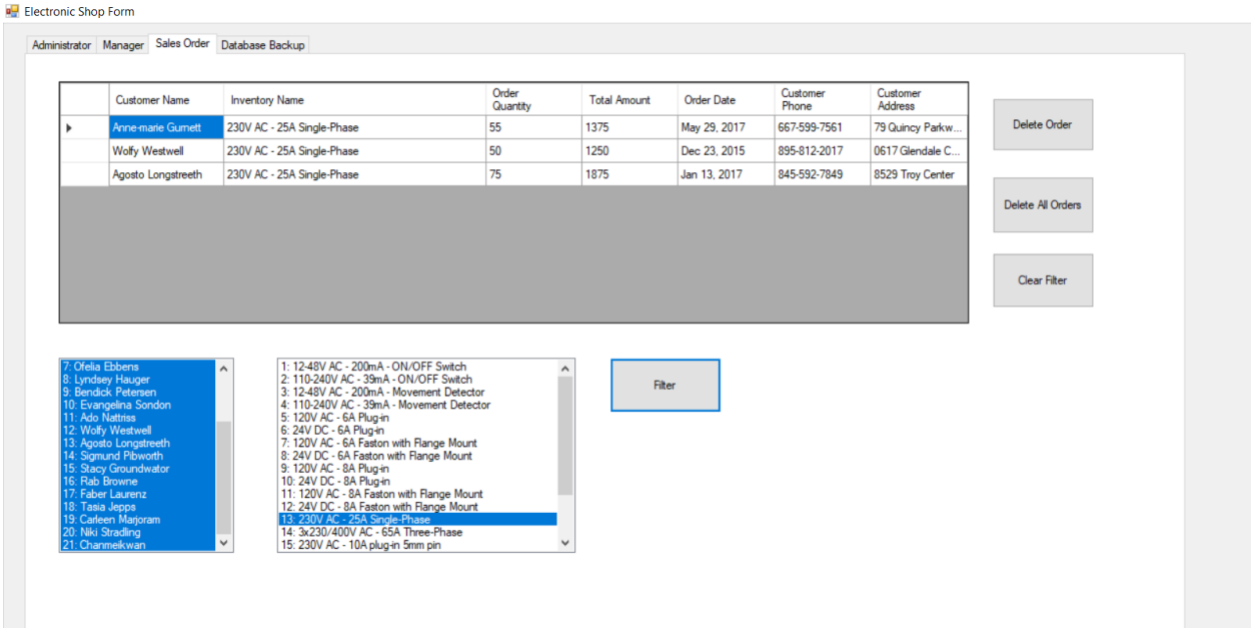
a) Sales order tab previews the information of all customers and their orders, there are 2 list boxes, one for the customer and one for the inventory. Initially all customers and products of inventory are selected to show all the customers and their orders.

Select one customer, then press Filter. (It is allowed to select more than one customer)



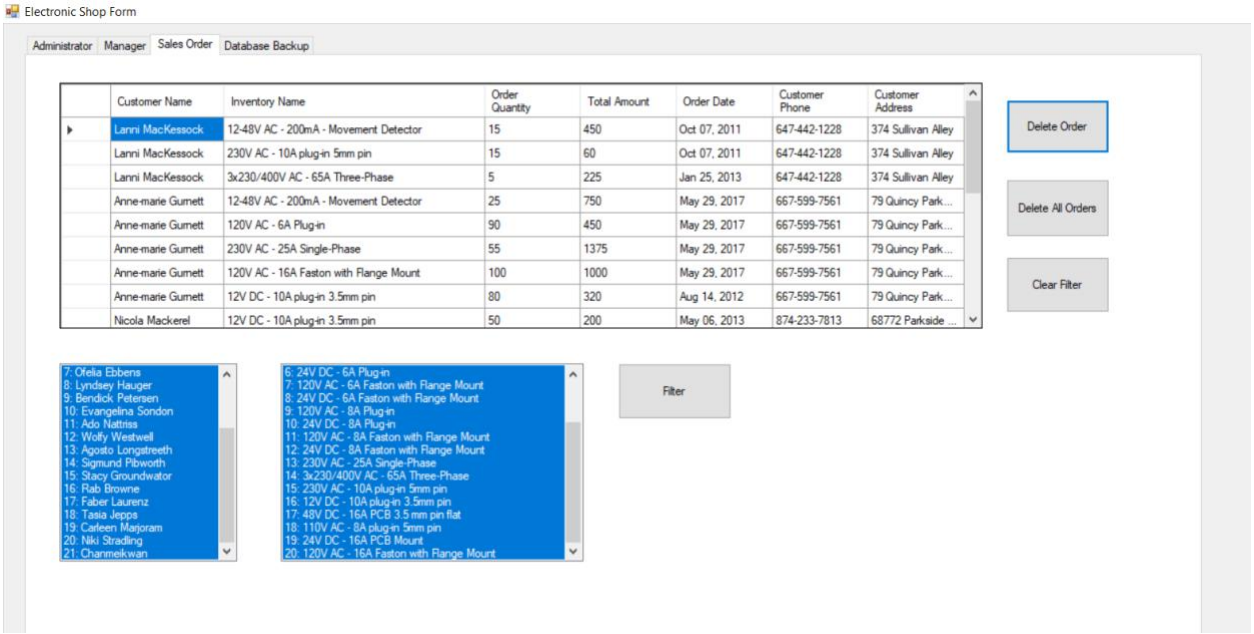
b) You can click on one or more customer, then if you press filter, you may view selected customers order and all their orders for all the selected products in inventory. You can clear filter to reselect all customers and inventory products list boxes.

Select one inventory, then press Filter. (It is allowed to select more than one inventory)



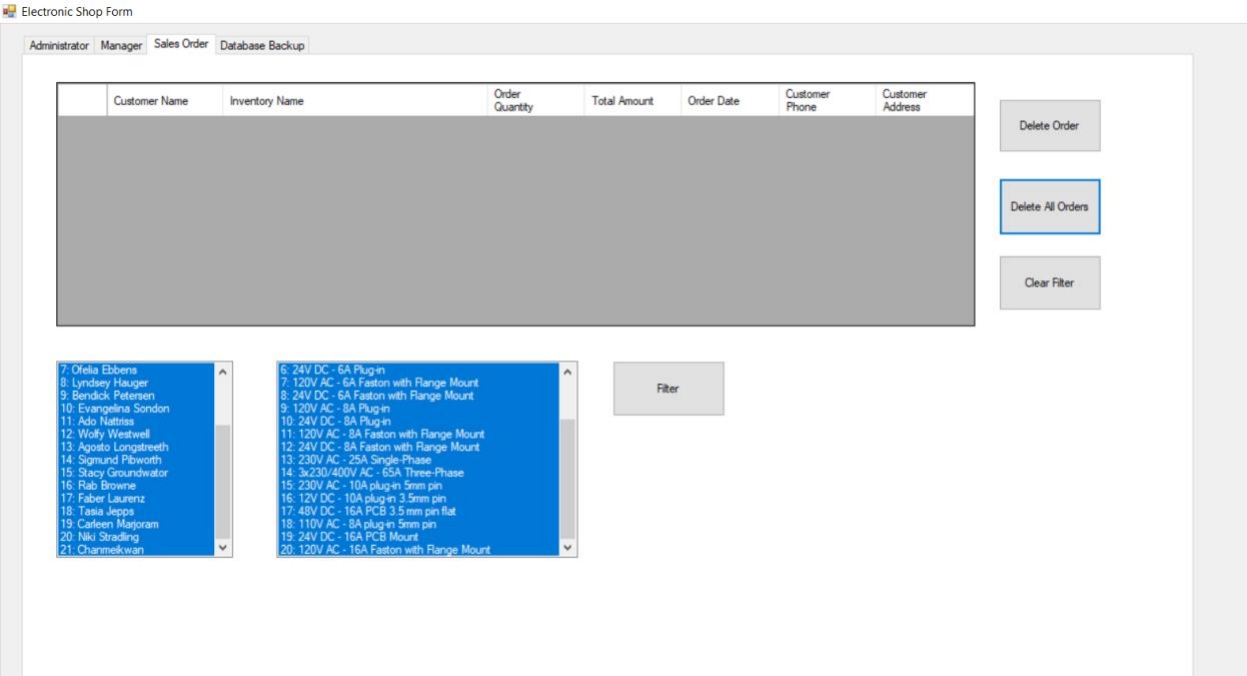
c) You can click on one or more inventory item, then if you press filter, you may view sales orders of all customers for the selected products. You can clear filter to reselect all customers and inventory products list boxes.

Select one row from the datagridview, then press Delete Order



d) You can delete a specific sales order if you select an item from dataGridView in Sales order tab and press on Delete Order button.

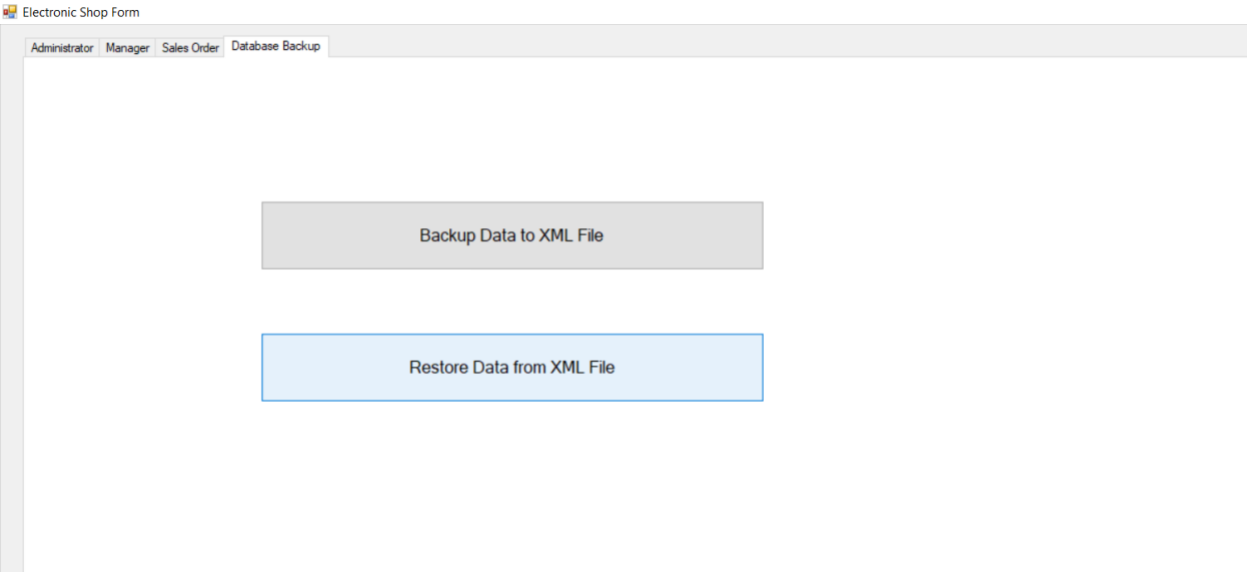
Press Delete All Orders

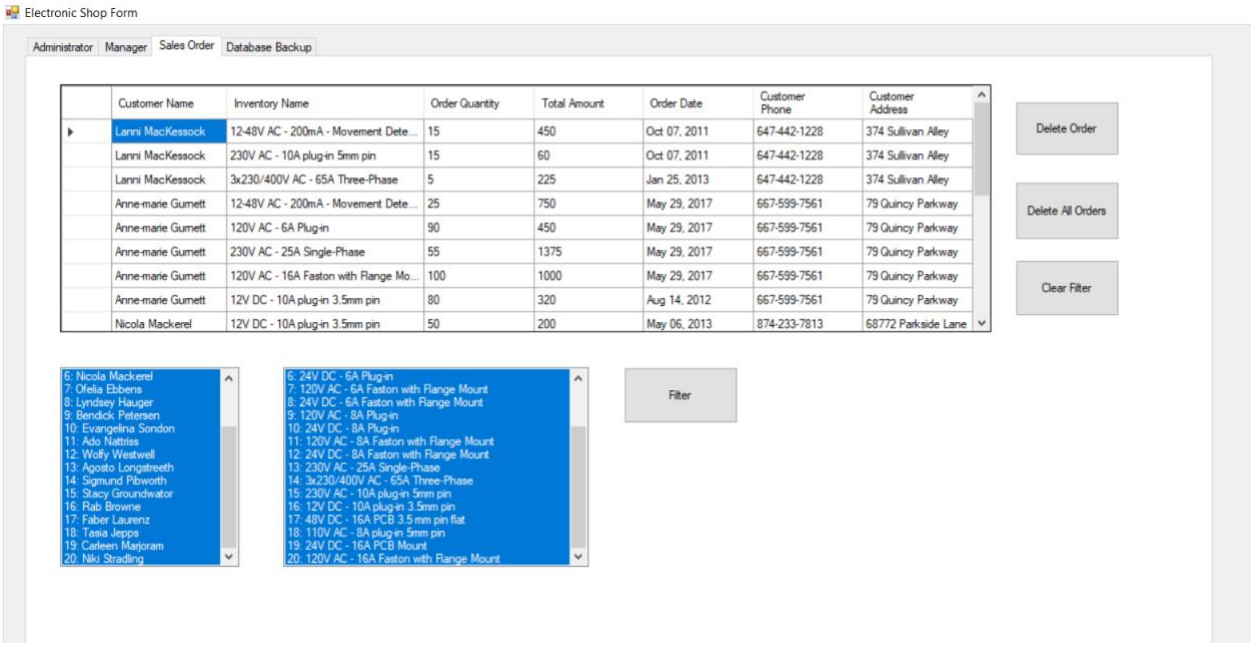


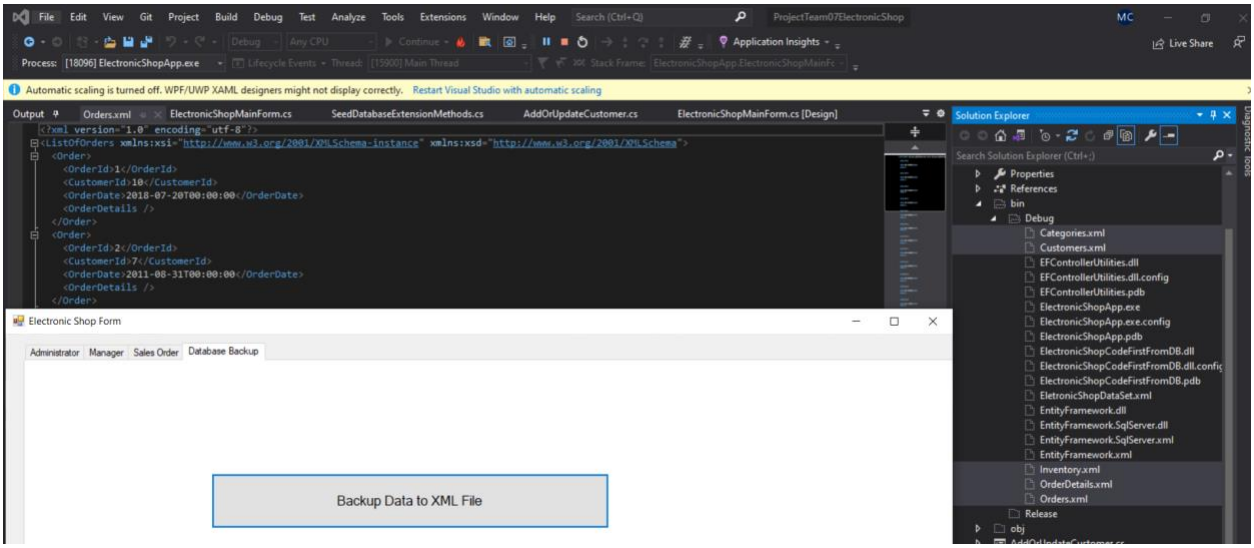
e) You can delete all the sales orders if you simply press on Delete All Orders button in Sales Order Tab.

4. Database Backup Tab

All Orders gone, go to the Database Backup Tab and retrieve all the data back to the WinForm







Press Backup Data to XML File, 5 Entities xml files will be generated and saved at (./bin/Debug).

You can backup data to XML file to be able to restore the same data in case errors happens and something goes wrong, for example all sales orders is deleted by mistake. This way you can safely retrieve all the deleted data.

User Interface

Administrator Screen A screen that the user can check and select available items and send the order to the company.	
Component	**Description**
Customer Details Field	Customers can fill in their information, such as Name, E-mail, Phone Number, Address…
Category ListBox	Listing and selecting all products of the company (Electrical & Lighting Accessories)
Type ListBox	Listing and selecting all types of each product from the category
Voltage ListBox	Listing and selecting all types of voltage from the product category
Pole & Current ListBox	Listing and selecting all the available poles and current from the product category
AC & DC Checkbox	Customer can select either AC or DC power
Add Item Button	A button that users can add selected item from the ListBoxex to the DataGridView table
Added Items DataGridView, includes 5 columns, which are: ItemId, ItemName, UnitPrice, Quantity, and SubTotal	Table that shows added items and subtotal of the item from the Listboxex
Calculate Order Button & TextView	A button to calculate the price of total of added items and print final result/number to the nearby TextView
Order Button	A button to send the order to the Manager
Close Button	A button to close the application

Manager Screen A screen that the user can manage the inventory of all available products. The manager can add, and update items within the inventory.	
Component	**Description**
Filter Item TextBox	Search for specific item bases on item Name or Id
Inventory List DataGridView	Listing all the items in the inventory
Quantity & Price TextBox	Provide values to update quantity and price of the item in the inventory
Add Item Button	A button to add new item to the inventory
Update Item Button	A button to update available item (such as price, quantity…) in the inventory
Delete Item Button	A button to delete item from the inventory

Sales Order Screen A screen that the user can manage all the sales order of customers	
Component	**Description**

Filter Order Textbox	Search for specific order bases on Customer Name or Item Id
Order List DataGridView	Listing the ordered items as well as customer information
Delete Order Button	Delete selected order from the list
Clear All Order	Delete all available orders from the list

Database Backup Screen	
Backup Data Button	A button to backup database to XML file
Restore Data Button	A button to restore database from XML file

Project Plan (Timelines)

Key Dates	Benchmarks to finish	Participants
Nov 9	Proposal 1st draft submission	All
Nov 16	Create the Database and Entity Framework, Insert the data in database, create the publish.xml	All
Nov 23	Admin Screen, Manager Screen, SalesOrder Screen, Database Backup Screen	All
Nov 30	Testing the final application	All
Dec 8	Final Report submission	All

Task assign:

Database and data access	Sadaf, Mei, Quintus
Testing	Mei
Reporting	Sadaf
Integration	Quintus
Sales + Backup Page	Mei
Admin Page	Quintus
Manager Page	Sadaf