# **C#/.NET Project Proposal – Inventory Management System**

Andrew Morrison (200463770)

Marcos de Gois Borges (200445501)

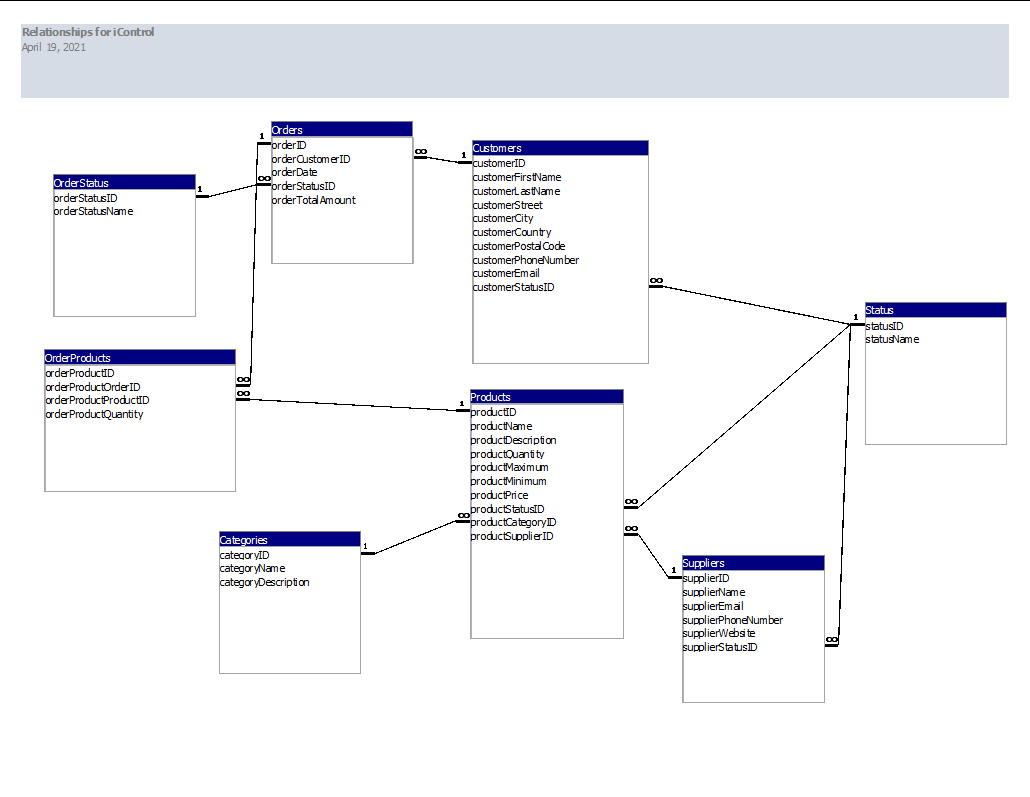
Department of Computer Studies: Georgian College - Barrie

COMP1098: C#/.NET

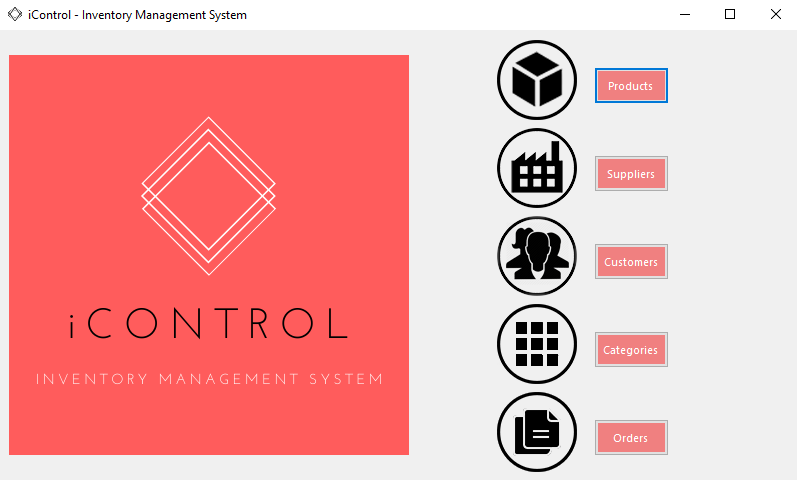
Professor Radhika Sharma

April 19th 2021

# **C#/.NET Project Proposal – Inventory Management System**



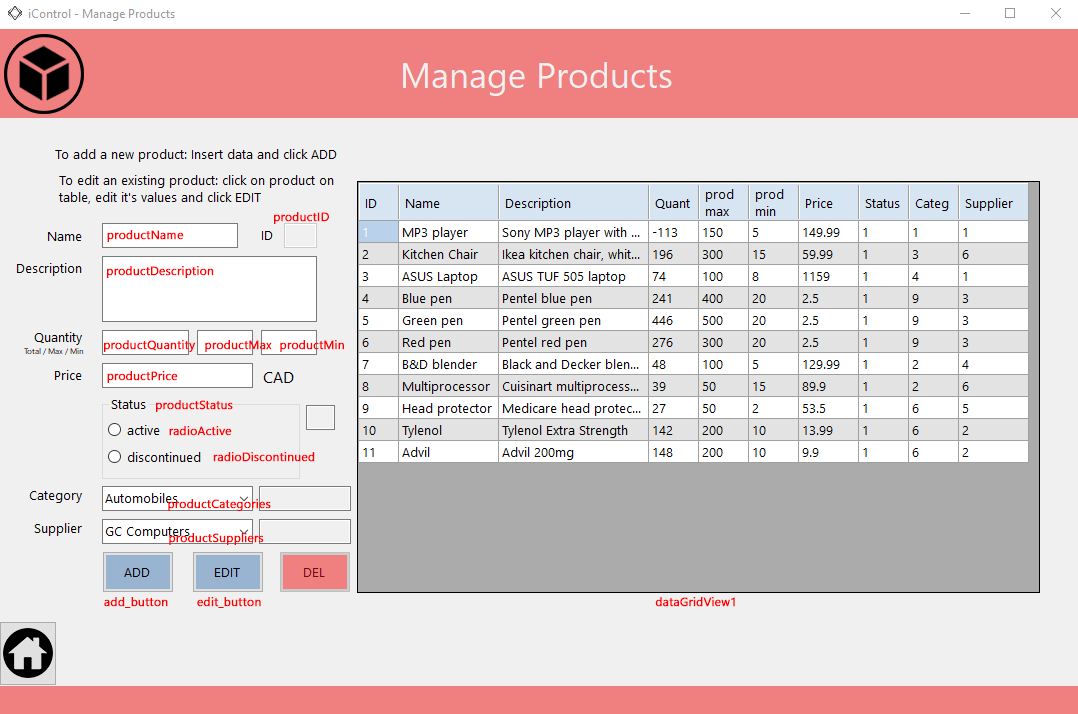
The database was created using Microsoft Access and contains 8 tables. There are multiple dependencies in between the tables. They are named as: Customers, Suppliers, Products, Categories, Orders, OrderStatus, OrderProducts, and Products.



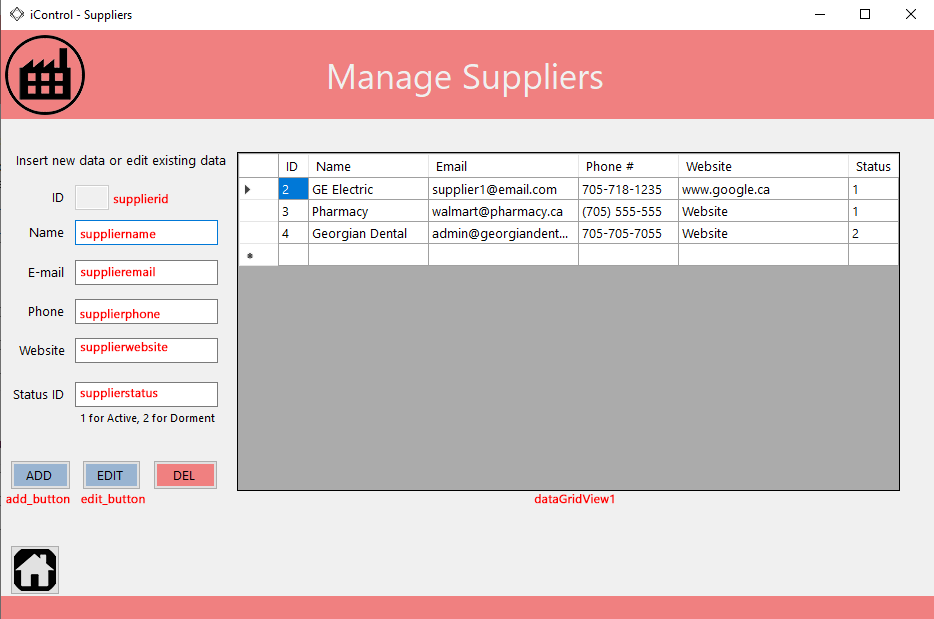
The home screen has 5 buttons that takes the user to the correspondent screens. It uses labels to hold the images and default buttons. The buttons action closes the home screen and calls an instance of the form.

There are global rules for all the forms:

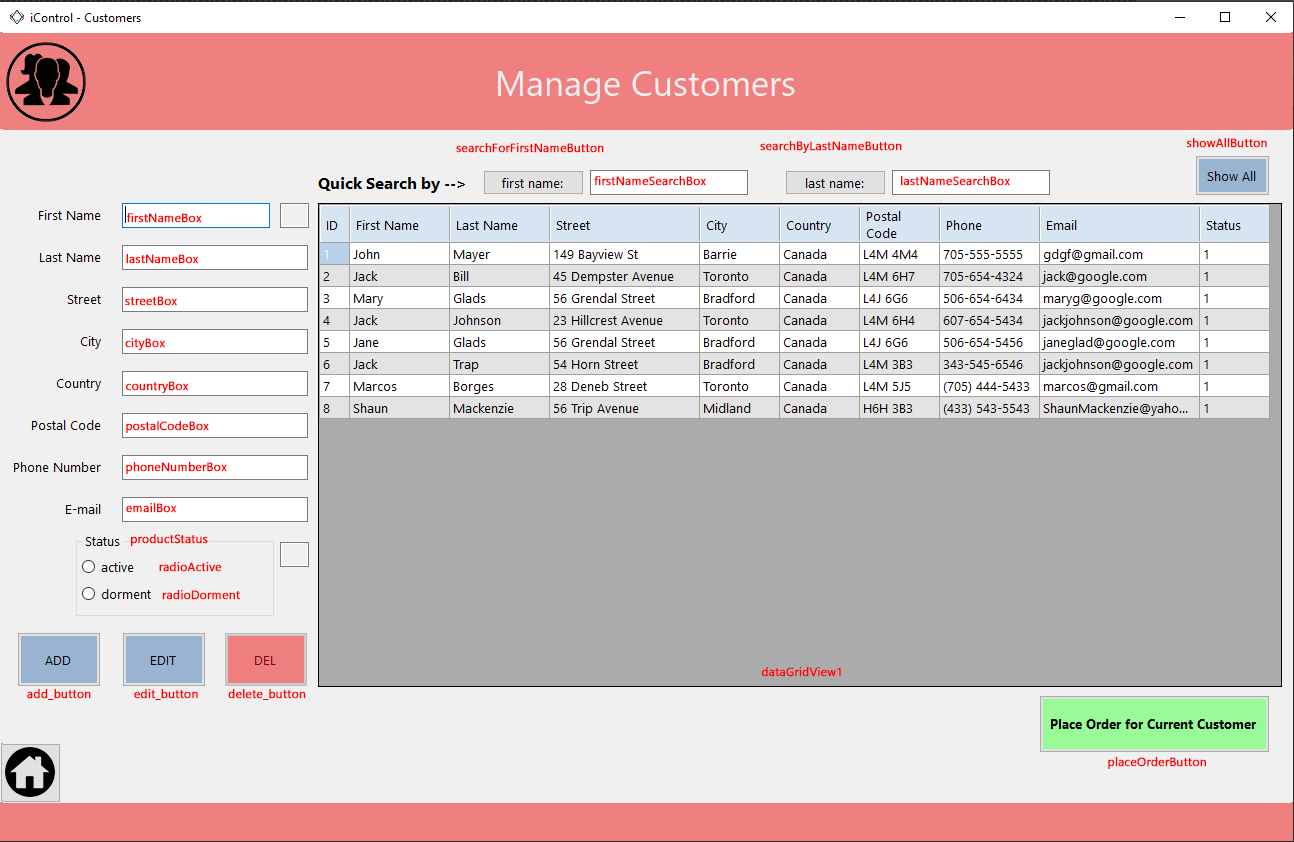
1. New form window opens centered in the screen. User can resize, maximize or close program
2. The header containing the form title label and the logo label is part of the top dock
3. The footer containing the home button and the bottom label is part of the bottom dock
4. When user resizes the screen the header and footer moves accordingly and fills the screen horizontally (responsive)
5. The DataGridView cannot be resized. Columns and rows cannot be resized by the user. The user can organize data by clicking on the column header or using the search fields. When user clicks on an entry, it populates the side fields. DataGridViews are connected to the Microsoft Access database through a OleDb connection.
6. Delete buttons are disabled as we want to avoid the database to break. They were used during the process for creation purposes.
7. After adding or editing entries, a method clear all the fields
8. There is also a method that refreshes the DataGridView after adding or editing an entry
9. Some forms has search tools that shows different info on the DataGridView



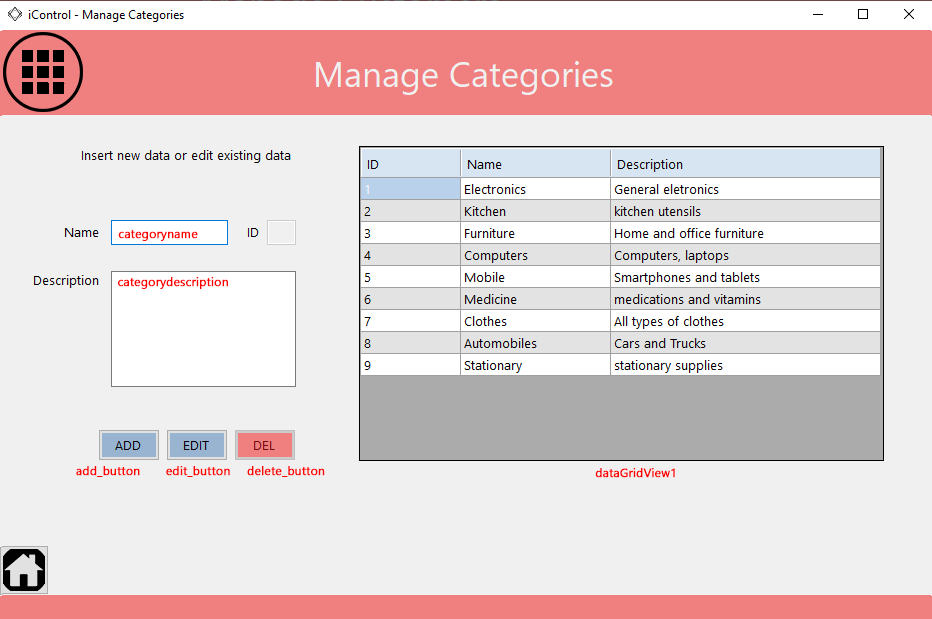
The products form shows a complete list of products on the DataGridView1. The user can create new products by completing all fields and clicking the ADD button. The ADD and EDIT buttons first checks if there is no empty field. It also checks the format of some entries (price, quantity, max and min should be numbers only). This Status Group Box allows user only to make 2 options. The Category Combo Box retrieves and shows the category names from the database. The Supplier Combo Box retrieves the suppliers on the suppliers table but only shows the active suppliers (it sorts by supplierName and filters by supplierStatusID = 1 (active only)).



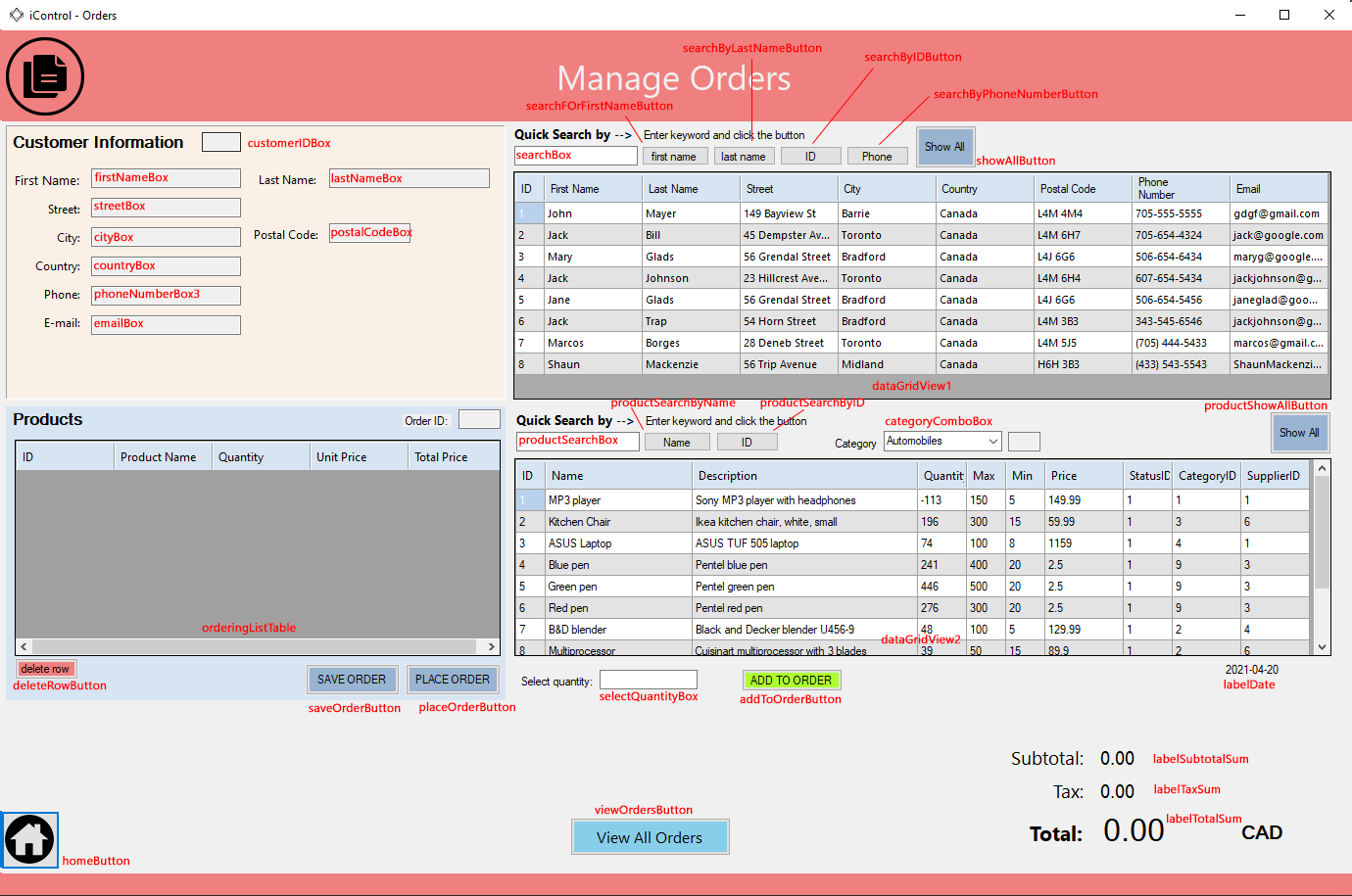
The Suppliers form shows a complete list of suppliers retrieve from the suppliers table. User can ADD or EDIT entries. There is field validation to check for phone format, e-mail format and Status ID (1 or 2 only). Also, it checks for any empty fields. User can click on an entry on the DataGridView, it populates the fields and user can modify the info.



The Customers form show a complete list of customers retrieved from the customers table. The user can add new customers. To edit an entry, the user can click on the table to select the entry, it will populate the fields automatically so user can modify existing information and click the EDIT button to send them to UPDATE the table. The form has the same field validation as other forms to check for any empty field and entry format. On this form, the user has the ability to search for first or last name using TextBoxes. The SHOW ALL button shows all customer entries again. From this form, the user can start an order. It makes it easier when adding a new customer that is currently placing an order. It closes the Customers Form and opens an instance of the Orders Form (frmOrders.cs). Also, when clicking the PLACE ORDER FOR CURRENT CUSTOMER, it collects the information of the text box fields and drags it to the new form. It automatically fills the customers information on the new Orders Forms.



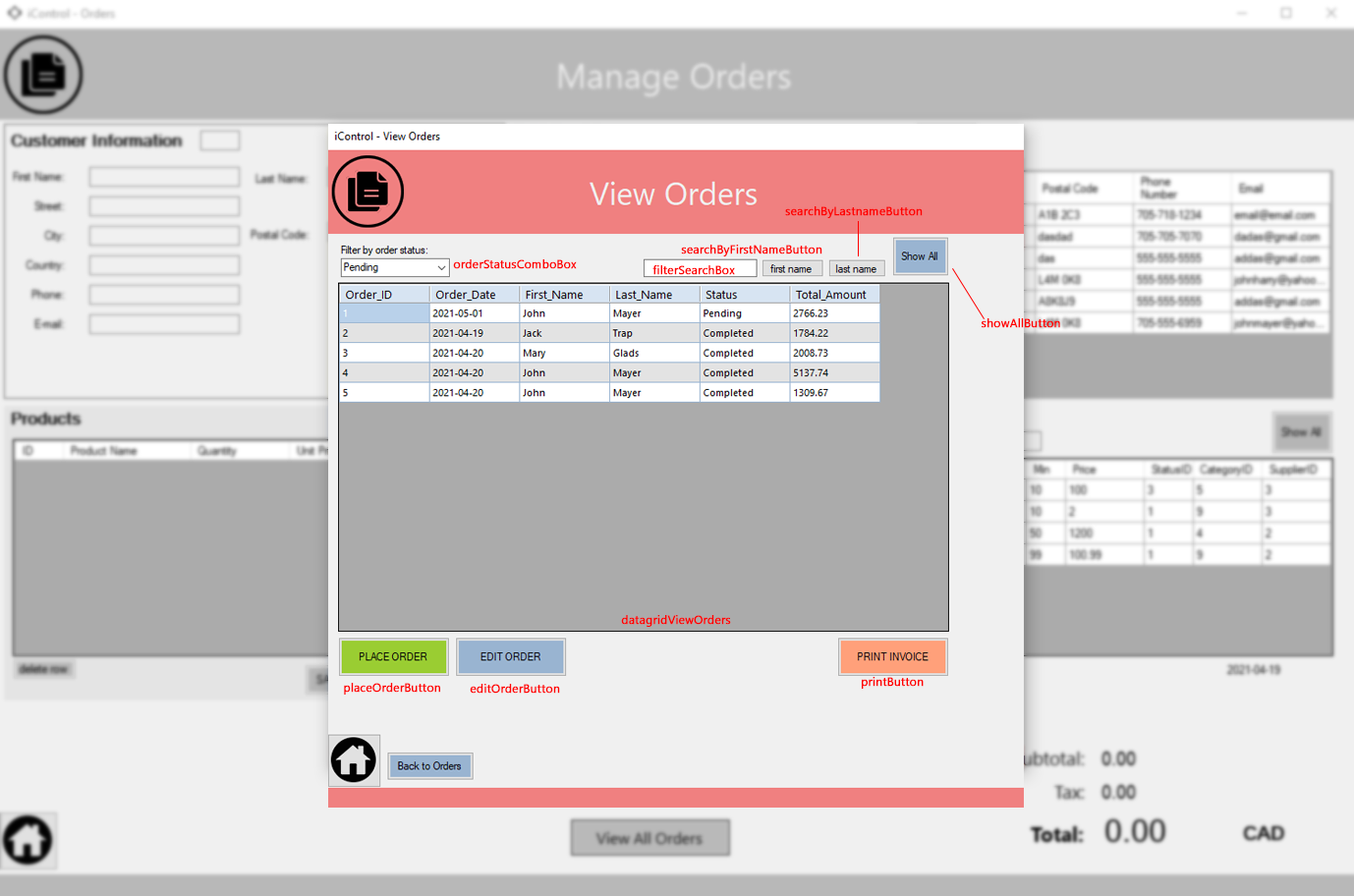
The Categories Form initially shows a complete list of categories retrieved from the categories table. The user only needs a category name and a description to add or edit the entry. All products are related to one category.



The Orders Form is responsible to unite information from a customer and products that the user selects. First, the user has to select a customer. There are search tools so the user can search by first or last name, ID, or phone number. After selecting the customer, the user can search products by category, name, or ID and select the item from the table. User has then to enter a quantity and click ADD TO ORDER. The ADD TO ORDER button checks if user selected a product and if user entered a quantity. The quantity must be lower than the product quantity on the table. Moreover, it checks if item is already added to the ordering list. It can only be added once. The action of this button checks for all this information, send the user a message if there is any error or send the information to the Ordering Table if the information is correct. The Ordering List Table is just a temporary DataGridView. From the Ordering List Table, the user can delete the entry if needed. When ordering list is done, user can SAVE or PLACE the order. Both buttons will save the products to the orders table but saving will create a “pending” order and placing will create a “completed” order. After doing this, it clears the ordering list.

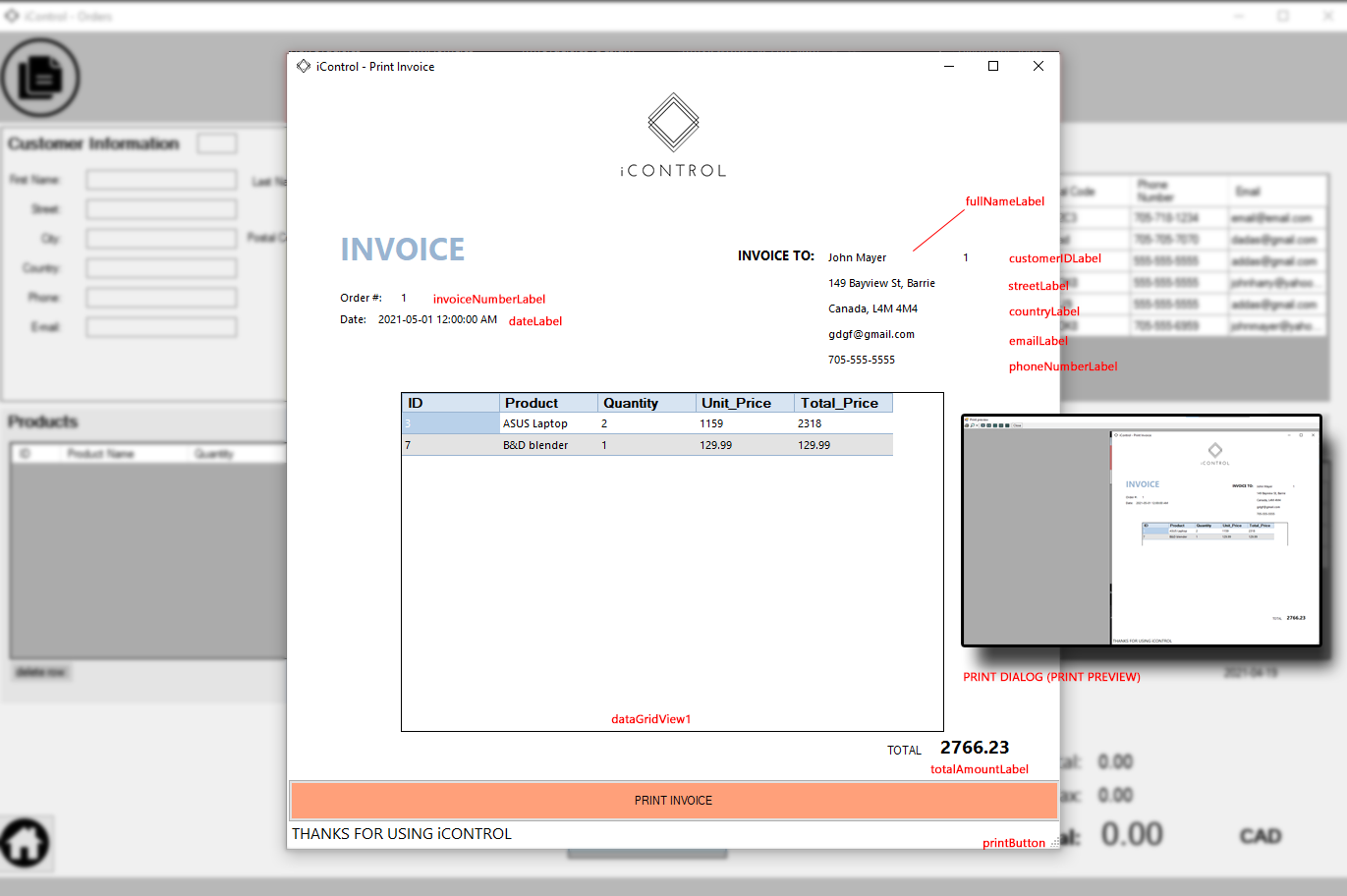
There is labelDate containing the current date. This information is used when creating a new order as we have a date field on the orders table.

The VIEW ALL ORDERS closes the Orders window. It only opens the ViewOrders form on top of the parent screen.



The View Orders screen shows all orders (pending and completed). The user can now change the status of the pending orders or see and print the invoice. To show and print the invoice the user needs to select an entry from the DataGridView and click on PRINT INVOICE. The button checks if user selected an entry. The user can also edit a pending order, which will open the order in the Orders window.

The PLACE ORDER button also checks if entry is selected but also if entry has a “pending” status. If yes, UPDATE the status to completed, If not, inform user.



The PrintScreen form shows the selected order Invoice. If user wants only to visualize info they can close the window or they can click PRINT INVOICE to send it to the printer.