**Roles and responsibilities.**

**Objective:** This project is for a computer parts shop. The User can login and buy computer parts. The Admin can add users, products, and perform other administrator activities.

## Admin

* Login
* Add User
* Remove User
* View Products (Override to see extra columns)
* Add Customer
* Remove Customer
* Add Special Sale
* Edit Special Sale.
* Remove Special Sale
* Same functionality as a user
* Override certain functionality if it is a user

## User (Client)

* Login
* View Products.
* View Products by Filter. Filter Products.
* Sort products. (Use Sorting Algorithm)
* View Order Information
* View Account Information
* Buy Item.
* Un-Buy Item.
* Sell Item
* Add item to Wish List.

## Products

Hard drives

Desktops

Laptops

Phones

Accessories

All

Software

Hardware

## TODO

**Done Tasks**

* Create Use Case Document
* Technical document low level. Describes each method.
* Create UML Diagram.
* Create Program Flow Diagram.
* Create PseudoCode view for each method created for easily understanding for development or PseudoCode for a particular functionality.
* Precondition: User has entered the valid conditions
* User login: with right conditions
* Remove Order
* Remove Product
* Serialize Order list for persistent usage. Static Variable.
* Create Order Table. (Add Order object to database)
* Create Customer Table (Add combination of orders and Users to table)(Dates purchased as well). Country Origin, Address, Zip Code, First Name and Last Name
* Customer/User will have FirstName and Last name
* Extend User class to make Customer Class.
* Customer would contain FirstName, LastName, Last Purchase Date, Number of Orders, Number of Products purchased, Last Date logged in
* Add Customer method to populated these fields
* Add Customer to database to populate customer database that will have same fields.
* Add Customer to Orders
* Admin can also view all the customers.
* Have Customer class with static customer list.
* Fix adding datagrid to new column
* Remove Product and remove Order functionality;
* Move Place Order functionality in AddOrderMethod
* Make getcurrentProductID functionality for selectedCell
* Make RemoveProduct for selectedCell as well
* Make RemoveOrder for selectedCell as well
* Code when Placing an order with a blank quantity.
* Make ConfirmBox for Placing an Order.
* Add Customer to table.
* Add Order to table.

**Not Done Tasks**

* **Decide Columns to display for Orders.**
* **Implement Sort for Orders**
* **View Orders dataset in another form**
* **Add Key events for Tab and Enter for all Forms.**
* **Add status update when mouse hovers over Buttons**
* **Add Menu strip items.**
* **Design what menu items should be in the menus using Microsoft Visio 2013**
* **Finish About Form**
* **Finish Account Form**
* **Think about how a user can login and register to be a customer.**
* **Clean up code and re-factor code. Group UI functionality to more methods.**
* **Think how to implement current functionality to work on any computer. (Maybe have a sample hardcode admin and Current Admin )**
* **Add try/catch blocks. And exception handling**
* **Add Sample set of Users, Orders, and Products checkboxes for sample list.**
* **Add ability for admin user to change current database settings?**
* **Incorporate ASP.NET to project.**
* **Make ConfirmBox for Removing Order**
* **Make ConfirmBox for Add/Removing Product for Admin User.**
* **Maybe create check boxes to allow for sample data to fill datagrid instead of data from ProductID table. Or maybe create a database on this system.**
* **Add ASP.NET MVC and JQUERY/Javascript to project**
* **Maybe translate Customer and Orders to XML File.**
* **Generate Receipt class?**
* **Draw Database relationship diagram or create with MS SQL**
* **If Placeorder is clicked, display red message stayed customer must login.**
* **Create Customer Login and Register button.**
* **Can login as a guest.**
* **Can login as a customer. Check customer list for login username. Add Register button to Login Form. Click button goes to registration form. After valid information entered. Create a User for current customer. Assign User to Customer. Assign Customer to database.**
* **Add usability tips to Forms(click here to this, etc..)**
* **Upload project to GitHub**
* **Add validation for Username,password,phonenumber,email,address**
* **Use Thread Class**
* **Add Middle Name to Registration form.**
* **Add Unit Test**
* **Add Unit Testing Documents**
* **Create Unit Test Documentation for specific scenarios.**

# Priority

**Unit Test (Sample Unit Test File)**

**Unit Test Documentation**

**Refactor code in Form Class**

**Try/catch Exception Handling**

**Add Key events for Tab and Enter for all Forms.**

**Key event for MouseHover**

**Menu bar layout**

**Thread Class**

**Sample Data list for User and Customers**

**Sample Defect and Code Fix with Unit Test Documentation.**

**Identify defects**

## ASP.NET

Convert Orders,Products, Receipt to HTML

Run project through ASP.NET. ASP.NET version of projects?

## Product

* Add Brands to products.
* Extend Product class for different product objects
* Add validation for adding a product.
* Change Category to Combo box in Product Form for admin and regular user.(Desktop, Laptop, Accessories, Software, Hardware, Other)
* Add Sorting of category for admin user and user.
* Add Sort by category for different categories. (Sort by category then sort by weight and other)
* Another combo box for sort by other categories.
* Define category as an Enum?
* Put products in datastore.
* Create Product Class and user as objects like User Class.
* Create Interface for Product Class.
* Create Update Product Form
* Create UpdateProduct Method In Product Class
* Create Method to obtain all products from product table and put in product objects then to List of Product objects. Call method before updating product
* Think of design of Update Product Form.
* Update Query method for Product class.
* Make method to get certain products based off of product name and category(?).
* **Search Product Forum. Auto-detect product from typing key words. (Need RegEx)**

## Orders

When a Customer buys a product. Product becomes an Order. Customer has a list of products ordered. Order has a quantity, the Product ordered, and date purchased. (gets current date), username, cost.

Create Add Order Method.

Create Add Multiple Orders.

Create Cancel Order Method.

Create Cancel all Orders.

View AllOrders (admin)

View Orders(username) – View Orders from a particular user.

Create option to update orders. (Have 2 days to update order)

Add Option to Cancel Order for Client Orders Class.

## 

## Client Orders

Buy single orders.

Buy multiple orders

Get your orders.

Update Orders Button go to Update Orders form

Create Update Order Form.

View Account Info

View Current Orders

Order will contain username, Hashtable for Products and last date purchase(For now list of products, quantity)

## Account

Username and email

View Orders Button

Change email

Change password

Contact support

About