Problem Description:

The objective of this project is to design, implement and test a realistic computer software using the concepts and functionalities learned throughout the COE 528 course. The software design had to include the following:

* Minimum of 8 classes
* 2 design patterns
* UML diagrams
  + 1 class diagram
  + 1 use case diagram
  + Multiple sequence diagrams
  + Multiple state diagrams
* JUnit testing
* White and Black box testing
* Specifications for all methods for two classes
* Javadoc for all classes

To meet the above requirements the software design we chose to implement is an Ecommerce Online Store. Users of this store can login and gain access to specific roles/operations depending on their user profile stored in our text file database. From there the user may add products to their cart, remove products, search products and finally make purchases of items in their respective carts. Additional operations are given to managers to allow them to add quantity of items, or adjust prices accordingly. Along with this registered customers receive an additional 10% discount on items purchased from our store.

With the functionality described above as well as some additional features. Our project contains 13 classes, incorporating 3 design patterns; façade, strategy and singleton. In conclusion the optimization of the Ecommerce project is a realistic implementation of an online store, with great added benefits for users to gain access to.

Class Descriptions:

Categories: an interface containing all the methods implemented by the category subtypes. Only accessible through façade.

Clothing: overrides the methods of the Categories interface providing an array list for Categories to access products in the clothing category without having a direct interaction with the text file every time a product is accessed.

Electronics: overrides the methods of the Categories interface providing an array list for Categories to access products in the electronics category without having a direct interaction with the text file every time a product is accessed.

Payment: an abstract class accessible only by façade, contains the method MakePayment, called upon when user is checking out of the online store

Sports: overrides the methods of the Categories interface providing an array list for Categories to access products in the sports category without having a direct interaction with the text file every time a product is accessed.

cart: implementation of the singleton pattern, the class has an array list taking products selected by the user to add to their own designated cart. This instance is reset for every time a user logins, thus allowing everyone using the online store to have their own unique cart.

creditcard: overrides the MakePayment method of Payment, asking the user for inputs of their credit card information

customer: the customer class contains an object of the façade class, giving the user the operations accessible to unregistered customers

manager: the manager class contains an object of the façade class, giving the user the operations accessible to managers.

paypal: overrides the MakePayment method of Payment, asking the user for inputs of their paypal information

registered\_customer: the registered customer class contains an object of the façade class, giving the user the operations accessible to registered customers.

Ecommerce: contains the main method of the program, this class contains direct interaction with the user, as well as the starting point for access to the online store. It asks the user for login information and from there directs the program to the specific roles/operations whether that would be manager, registered customer or guest checkout.

façade: this class is tied to many of the design patterns, the façade pattern as well as the strategy pattern and an aggregation with the singleton class as well. The façade class acts as an interface used by the manager, customer and registered customer classes. For the user to do any of the operations in our online store they online interact with the façade class. The purpose of this is to avoid the interaction of the user and multiple classes as not allowing the user access to rep of the Categories, Payment or Cart classes.

login: this class validates user login information with the text file, whether that user is a registered customer, manager or a normal customer registering with our online store