**Checkout system for GroceryCo supermarkets.**

**Third-Party Libraries Used:**

NUnit , Log4net, Microsoft exception datablocks. Libraries are included in checkin (3rdPartyDLL folder)

All projects in this application employ version 4.5 or higher of the .NET framework.

**Running the App**

When the console gives multiple input options, the actions available can be selected by entering the letter or number surrounded in brackets and pressing enter. All the data is checkedin to the Github including user, login, product, promotions etc.

**Assumptions Made**

A grocery item cannot have more than one promotion associated with it at any given time.

A grocery item is checked out as unit price. System doesn’t support weight, grouping of different types of items.

A grocery item is not grouped into categories like Dairy, Organic, raw etc.

Promotions needs to be deleted in order to and new ones to the same product.

Sale types are not user defined. That is to say, new "types" of sales (a "combo" deal for example; buy one Apple and save on Oranges) will require new development.

On the functioning of the "Additional product discount": only one subsequent item will receive a discount after the requisite number of items has been added to the checkout. This discount will be entered as a percentage. In the case of "buy one get one free", the discount would be set to 100%.

"Basket" files (containing an unsorted list of item names) are assumed to be comma-separated .txt files. Sample files have been included in data folder.

Reciepts will be output as a plain text file, and stored in the repository folder. The resulting receipt.txt file gets overwritten with each checkout.

**Behavioral Decisions**

Promotions with no start date and no end date cannot be entered via console. However if someone goes into xml file and changes it to null then if any of start date and end date is null or both are null then promotion is considered as infinite and will not end until user manually goes to the console and end it.

Receipts will be output as a plain text file, and stored in the repository folder. The resulting receipt.txt file gets overwritten with each checkout.

**Design Decisions**

Entities will be serialized and stored in files using XML.

User needs to be login before entering into system.

All configuration must be set properly before login.

The system has 3 types of users

1. Admin 🡺 Admin user can access both settings and checkout and generate receipt. It can add promotion to an item, end promotion at any time.

Login information: Username: admin1, Password: test12

Login credentials are stored in Logins.xml. Password is encrypted in base64.

1. Marketing🡺 Marketing user can access only settings. It can add promotion to an item, end promotion at any time.

Login information: Username: marketing1, Password: test12

1. Clerk🡺Clerk user can access only checkout and generate receipt

Login information: Username: clerk1, Password: test12