Project 0

**Presentation Day: Dec 23rd**

Objects:

* Customer
* Products
* Orders
* Stores

Customer:

* view their past purchases
* view available store locations
* must be able to purchase 1 or more products
* must be able to view current cart
* must be able to checkout
* must be able to cancel a purchase

Products:

* have a name
* a price
* a description

Order

* must be able to compute its total cost
* must be able to contain at least 1 product
* has a order limit of 50 items
* must be able to limit its total cost to no more than $500

Store:

* must be able to view past sales
* has store location
* [stretch goal] must be able to manage product inventory (add, edit, delete any product)

Architecture:

* StoreApplication (solution - sln template)
* Client (executable - console template)
* Domain (library - classlib template)
* Storage (library - classlib template)
* Testing (unit testing - xunit template)

Other Requirements:

* a solution containing 4 projects (Client(FE), Domain(Business Layer), Storage(Repository Layer), Testing) should be implemented (10pts)
* test coverage of at least 30% should be implemented (10pts)
* data storage using a DB with ADO.NET should be implemented (10pts)
* runtime monitoring with logging should be implemented (10pts)

Scoring - 50 pts total

|  |  |
| --- | --- |
| Points | Description |
| 2 pts | customer being able to view a list of store locations and select one |
| 2 pts | customer being able to view available products to purchase at a store |
| 2 pts | customer able to purchase at least 1 product, max 50 items or max $500 |
| 2 pts | customer able to view their past purchases |
| 2 pts | customer able to view past purchases of a store |
| 10 pts | solution containing 4 projects (Client(FE), Domain(Business Layer), Storage(Repository Layer), Testing) should be implemented |
| 10 pts | test coverage of at least 30% |
| 10 pts | data storage using a DB with ADO.NET |
| 10 pts | runtime monitoring with |

Notes:

Cart features – what happens if two people are shopping for the same thing at the same time? Give a person a warning

Login features- use the same thing that we used in the RPS

Database – use the same thing we did in the RPS

In RPS you can start a new game at the end, use that to continue shopping when you finish one. If they don’t want to play a new game they have to log out and then it goes back to login. So do the same with the app, if they don’t want to continue shopping they can log out and then it goes to the log in prompt

Clarification, you can have one store different locations or each store is each location