# Design Overview for Product Sales Enquiry

Name: Paul Watts

Student ID: 969987

# Summary of Program

This program will facilitate the creation of a product item. It with include an enquiry option that will display information helpful for a person selling the product based on the entered values. It will also allow the user to change any of the entered Product details and redisplay the calculated values based on the changes.

The calculated values helpful for Sales People include the Gross Profit in dollars and the Gross Profit as a percentage (i.e. the proportion of the sale that is profit) and the gross markup % (i.e. the selling price compared to the cost price expressed as a percentage). Note these are all real world values used in product sales.

The CR level task would include the ability for a Sales Person to enter a Selling price or a discount off the stored selling price and display the calculated information with appropriate warnings if it exceeds the maximum discount or the minimum markup on cost. However it will still display the calculated information with appropriate warnings .

These figures are all exclusive of any local taxes, e.g. GST, as this would be normally calculated based on the relevant tax (e.g. GST or VAT) percentage or amount at both the selling and buying stages of the Product. Handling GST or local taxes is beyond the scope of this project.

The HD level task would include the ability to store the Product objects in a **dynamic** Array with ability to add, change, or delete Product objects in the Array. This would be the precursor to using local file storage as explained below. There would be additional Properties added to the Product class at this time, e.g. creation date, last modified date, (using the Java Date() class), and the user identifier of the person who created or changed the Product, etc.

A higher level project, adding on to this one, would be the ability to store the Products in local file storage with a simple file structure (e.g. CSV or XML) with the ability to add, change or delete a Product while still using all the methods and classes (i.e. all the functionality) of the lower level Project.

# Required Data Types

Table : Product details

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| code | String | Unique identifier |
| description | String | Product description |
| sellPrice | Double | Recommended selling price |
| cost | Double | Cost price |
| productSize | Size | Product size |
| packSize | Size | Required Packaging size |
| maxDiscountPercent | Double | Max discount % off selling price |
| minMarkupPercent | Double | Min allowed markup on cost |
| grossProfitAmount | Double | Gross Profit Margin in $ |
| grossProfitPercent | Double | Gross Profit Margin as a % |
| grossCostAmount | Double | Gross markup in $ |
| grossCostPercent | Double | Gross markup % on cost |

Table 2: Size details, declared in Product

|  |  |
| --- | --- |
| Value | Notes |
| SMALL |  |
| MEDIUM |  |
| LARGE |  |
| EXTRA LARGE |  |

# Overview of Program Structure

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Name | Return type | Description |
| SellProduct | Main | n/a | Main controlling logic (Menu) for the Program |
| sellProduct | calculateSales | Boolean | Calculates sales values and returns True if figures within acceptable parameters. |
| sellProduct | addProduct | Boolean | True if Product successfully added |
| sellProduct | maintainProduct | Boolean | True if Product successfully changed |
| sellProduct | deleteProduct | Boolean | True if Product successfully deleted |
| sellProduct | EnquireProduct | Void | Displays Product allowing Selling Price to be entered (not stored) for comparison purposes |
| Product | Product | Product | Instantiates a new product |

# Sample Output

Product Sales Enquiry

1. Add Product
2. Change Product
3. Delete a Product
4. Enquire on a Product
5. Sales calculator

9. Quit

<Choice 1>

Will display all fields and allow values to be entered

<Choice 2>

Will display all current field values and allow them to be changed

<Choice 3>

Will allow entry of a Product code and with appropriate warnings, delete it from the Array

(This is a HD level task)

<Choice 4>

Will display all current value of fields in a way that highlights the calculated values and cost and price fields

<Choice 5>

Ask for a Product code, display the product, and then allow a discount (off list price) to calculate the selling price or they can enter the desired selling price.

Appropriate warnings will be displayed if selling price is too low.

# Notes

Thought will be put into how to structure the program in such a way that facilities the maximum reuse of methods.

The 11.1HD project could be to research the best form of local storage for the dynamic array in this program and then modify the existing program to implement the solution.