Computer Programming

C20013

Programming Assignment #2

Brief:

You are to write a POS program in Perl which implements the logic of a shop cash register.

The product codes and other details are in a MySQL database. You will interact with this database via the API routines defined below. This is part of The Quinlough Adventure Centre and Hostel combined subjects project.

Summary:

The creation of this application will be a two-stage process.

First the main POS functions will be created; when this is working an extra layer to enable graphics will be added. We will later use the knowledge gained in this process to write our game.

The program will repeatedly loop until instructed to exit. Valid commands are:

X – Exit.....Drop everything and terminate the program immediately
N – New Sale.....Zero as required and start new sale
A barcode
E – End of Sale.....End the sale, display a receipt with an amount due on it
V – Void last Code.....Cancel last product from items purchased

You'll probably also want to use 'receipt' arrays to store the descriptions and prices of scanned products in order to be able to generate a sales receipt. This is the store for later paradigm.

In essence the purpose of this assignment is to create a loop that continually reads bar codes and matches the code with a database entry. Each correctly recognised code will yield a description and price for use/display by/on the register.

The prices will be totalled so that at the end of the sale a Total Amount Due is available to charge the customer.

After a sale is complete, the program should print the receipt and total due before remaining waiting for another sale to commence.

At any point, if a bar code is read which is not in the database, the program should display an error message on screen; no further action need be taken in the case of the unrecognised code, except to return to the waiting loop.

This methodology of accepting a barcode or a command code is the same as accepting a name, number or a 'q' to quit in previous programs: just with more options.

Presentation:

Marks are awarded for attractive presentation both of the screen output and the source code. Code indentation and commenting are ***vital***.

API:

In your program include this line:

```
require "/home/public/pos_functions.pl";
```

which will give you access to these functions:

```
sub product_numbers {
# Purpose: get a list of EAN13 codes only
}
sub product_detail {
# Purpose: get product details in response to EAN13
}
sub product_barcode {
# Purpose: get barcode image data in response to EAN13
}
sub product_image {
# Purpose: get product image data in response to EAN13
}
sub product_manufacturer {
# Purpose: provide product manufacturer name (brand)
}
sub product_stocklevel {
# Purpose: Get or set product stock level
}
```

Definition of **Expects:** and **Returns** for these functions are on the course website.

Submission Mechanism:

By paper via the submission box in Room 15, and by e-mail. Include:

- Cover sheet ('My Own Work')
- Methodology planner
- Flow chart *or* Decision tree (Read the programming website for information).
- Source code (The perl program, printed)
- Demonstration video
- Sample data used
- Screen capture(s) as required

Any other relevant supporting materials.

Due Date:

20130125, 15:15, Friday
(subject to on-line Calendar changes)