Edward Demkowicz-Duffy

Crossley Heath School  Halifax

Computer Science COursework 2017

# Analysis

## Brief

My application will be built for my client (my friend Phil Fowler), who produces novelty clocks and coasters from vinyl records. He wants an online storefront that can clearly display all of his products to the public (i.e. anyone who visits his website), and allows them to purchase any set of products that they wish; so long as they are in stock. He also wishes that their purchases be stored, and that they be able to put products in their “cart” and browse the shop further before they “checkout”.

He would like the website to be able to track purchases and show the data related to them in user-friendly, easy to understand visualisations to assist with sales strategies and business model tweaks.

Furthermore, the website should have functionality for employees to manage products, view past purchases and manually enter non-electronic purchases and transactions so that displayed statistics remain as accurate as possible. This is primarily because the client sells his products at a stall at the Manchester Christmas markets, and requests that the website/webapp be able to assist with this. Towards this, I suggested that the website be able to track what products are sold and are on sale there, and be able to help using that data by printing catalogues and other material to assist him and his employees there and he agreed.

## Project Parameters

I have assembled the following goals for the application according to the client’s needs. The application must:

* **Store**:
  + Products in inventory, along with all the information relevant to them: names, prices, stock, images and descriptions.
  + Users registered with the site – there should be two types of users, customers and employees. For both types, a username and a hashed password should be stored. For customers, personal and contact information should be stored, primarily full name, address and phone number. For employees, their full name should be stored. Different employees will have different permissions, so some should be able to edit products or manage user accounts and some should only be able to view it.
  + Purchases made by customers and the data attached to them (date, amount received, customer)
  + User interaction with the application, in a log (text) file. All entries should be timestamped and be labelled with what session they occurred under. The log should include:
    - Account control events: “Logins”, “logouts”, registrations, password changes and account deletions
    - Product changes: manual changes to products by employees and purchases (in addition to the database table, for the sake of debugging)
    - Exceptions thrown by the application and the data attached to them
* **Show**:
  + To customers:
    - A list of all products available to buy on an online storefront
    - A list of all products the customer has selected to buy – their “shopping cart”
    - Their transaction/purchase history
  + To employees:
    - A list of all products available to buy, with metadata included
    - A list of all registered accounts, of both types
    - A detailed breakdown of sales statistics (including visual representations) to employees with data pertaining to:
      * A time period divided basis including options for months, days and years
      * Comparisons of different selling statistics:
        + Sales by user
        + Sales by date/month/season
        + Predict the most popular times of year and outlets for sales from interpolation of visible data
* **Allow**:
  + Employees to:
    - Manually enter sales for those they make outside of the website, at the market or otherwise
    - Print a customizable catalogue to display at the market stall
    - Change product information and add products if they are authorized
    - Change information stored with customers and other employees if they are authorized
* Provide a web-based UI for all these functions with intuitively and clearly displayed information and interactive elements; including but not limited to graphs for statistics with provision for sorting data and the ability to “zoom in” to view individual data and compare specific points

## Data Management and Storage

I will use a relational database to store all the relevant data. The tables will be roughly as follows:

|  |  |
| --- | --- |
| Table Name | Contents |
| Employees | Usernames, hashed passwords and a Boolean “admin” variable which will determine whether the employee has access to certain functions when logged in |
| Customers | Usernames, personal info (i.e. telephone number, address etc.) and a (hashed) password; the credentials will be used for logging in |
| Products | Generic records of all products the site is selling and all the information pertaining to them; their name, price, amount in stock, an image, a description, what product type they are etc. |
| Orders | A record for every different type of product bought by a different customer at a different time; it will store the username of the customer and the name of the product sold referentially, and a timestamp, amount and transaction amount as ordinary entries |

I plan to use no macros or any other function built into the software I choose, instead I will manage all changes to the database, and queries, within the application itself. This will keep all code and functionality in one place, making development quicker and keeping my program transparent to me.

## Notes

-Things I did in development:

* Tried using pure code to handle database but this didn’t work so I used datasets
* Got a custom SQL sanitization class working
* Tried using HTML verification instead of sessions
* Made a custom logging class for development and debugging
* Found that generics in C# can’t be filtered

## Initial Notes (Please ignore)

Stock:

-Records to be turned into clocks

-Stickers

-Blank Coasters

-Completed coasters (# of sets (6 in a set))

-Clocks – limited selection of records which are ordered frequently:

-Tins

-Costs of stock?

Core features:

-Separate mode for Christmas market or other special occasion:

-Separate inventory

-Printable catalogue

-Inventory management:

-Overview of completed stock

-Separate screen for total inventory/total components?

-Developmental features “add x amount of y”

-Tables:

-Orders

-Deliveries of components

-Production of completed products

-Clients

-Inventory

Extra features:

-Interface with online outlet’s APIs

-Centralize stock values/inventory across all outlets

-Authentication system

-Infographics

-Product comparisons and other data aggregation

Advantages:

-Volume sold; profit and gross

Notes:

-Postage is paid by client

[phil@philfowler.co.uk](mailto:phil@philfowler.co.uk)