Edward Demkowicz-Duffy

Crossley Heath School  Halifax

GCE Computer Science Coursework Component

Contents

[Analysis 1](#_Toc510096449)

[Description of the Problem 1](#_Toc510096450)

[Stakeholders 1](#_Toc510096451)

[Can the Problem be Solved by Computational Methods? 2](#_Toc510096452)

[Brief 2](#_Toc510096453)

[Project Parameters 3](#_Toc510096454)

[Data Management and Storage 4](#_Toc510096455)

[Notes 4](#_Toc510096456)

[Initial Notes (Please ignore) 4](#_Toc510096457)

# Analysis

## Description of the Problem

My client is a personal friend and the owner of a business that is based around the sale of novelty clocks and coasters manufactured from or based on vinyl records. They buy (often) second hand vinyl records from various sources, cut out the album art in the middle and use that to mount a clock mechanism on, then package it in it’s original sleeve modified to become a box as a clock to be hung or place on a surface. They also manufacture sets of coasters which are artificially manufactured to appear as vinyl records, sold in sets of matching bands or contemporary albums. Their present sales solution is in three parts:

* Online sales through third party storefronts Etsy and eBay
* Face-to-face sales at the client’s stall at the regular Manchester Christmas markets
* Other face-to-face sales at any other events the client or their employees may wish to attend, irregularly

This solution is inconvenient for multiple reasons. The two storefronts provide quite different experiences and tools for people wanting to sell their products on their platforms and thus my client often finds it difficult to deduce trends and critical values like net profits in their overall online sales. Sales at the Christmas markets are difficult to keep track of as the square they are held in get crowded and the client/their employees frequently lose track of sales, which can be found at the end of the day by examining existing records, money gained and remaining stock; but is still an obstacle to clear analysis of the client’s sales. Stalls at other events are usually set up at short notice, and similar logistical problems tend to arise as with the Christmas markets, but with less pre planning time and available information they also tend to be more difficult to solve.

The client would like a web app which provides a quality sales experience to customers, and powerful management tools and information for employees. It must have an online storefront, a (modifiable) storage method to track products and all their related information (most importantly stock and price) and sales of said products. Users should have to register to make purchases. Employees should be to view and modify products. There should be easy tools that specialize in managing stock and sales both at the beginning and the end of a day when the client sells their products at an event, be it unexpected or scheduled. The client also specifically requests that employees be able to view statistics and basic analysis of sales displayed in an easy-to-understand way. Employees with special permission should also be able to modify the accounts of both customers and other employees and delete or add products to the program’s storage. Furthermore, the website must be secure, with both customers’ and employees’ data appropriately protected and must appear professional and easy to comprehend.

## Stakeholders

Stakeholders are identified here as being people who must use, or will be directly affected by, the program. There are three primary stakeholders, who must be born in mind during the creation of the web app:

* Customers. These are the most important stakeholders as they as the client’s source of income. They must have the easiest and most convenient experience possible, to retain their attention on the products the client wishes to sell to them. They will benefit from the project by having a quick and easy way to exchange money for goods they wish to buy. As such, the interface they use should be carefully and logically laid out, with no unneeded features or artefacts on it. Products should be described clearly and concisely so the customers know exactly what they are buying and all other information relevant to it.
* Staff. These are the people employed by the client who manage their stock and make sales in person. Their side of the web app will be used for swift and easy viewing of stock and statistics – they may also need to modify stock data. Therefore it must provide the quickest and most powerful ways to manage current stock and sales at physical stalls, and helpful processed sales data.
* Administrators. These are senior employees who have all the responsibilities of other employees but have more power over the company’s assets. They have the same requirements as regular employees, but also need to be able to view the data of customers and ordinary employees, be able to add whole new products to the set (and delete existing ones), be able to change the passwords of and delete accounts (be they employee or customer) and be able to view technical data like logs.

## Can the Problem be Solved by Computational Methods?

The problem can be solved by computational methods, as detailed below.

The client requires a centralised service for distributing their products en masse to customers who may be international – a website or web app is perfect for this purpose as it can be accessed anywhere, can be easily translated, and requires little to no uncommon knowledge to use and access. It can be accessed from many places at once and run in parallel in all these places without compromising at all in any.

The client requires a service that will supply their company with processed data that will help them analyse their sales strategies and improve them. Computers excel at this as it is a repetitive numerical task which can be completed by a processor much faster than a human analyst. In addition to this, computers can scale easily to very large amounts of data, whereas other methods (aforementioned analyst) would likely struggle to cope in comparison.

The client requires a method for organising all their stock in a central location, with the ability for it to be accessed and changed from anywhere at any time. An always-on, internet-connected data source is an exceptional solution for this problem because besides matching all the required criteria since it can be accessed from anywhere by anyone with the appropriate credentials, it can add useful functionality to them by logging all information transfers and enforcing logical rules like relationships between data and correct formatting. The same data source can store information about both the customers and employees who use it for increased availability of data. This ties in with the previous requirement as all the data needing to be processed being stored in one (digital) location is extremely helpful for any computerised process trying to access it, keeping time taken to do that processing short.

## Research

## Features of the Proposed Solution

## Software and Hardware Requirements

As this software will be interacted with in different ways to how it is run at it’s actual location, I have included two sets of hardware requirements; one for the end user (be they employee or customer) and one for the server hosting the software:

### Client Computers

* The latest version of a modern browser. I recommend Google Chrome, Microsoft Edge, Mozilla Firefox, Opera or Safari.
* A computer which can run that browser. For reference, I have taken the requirements for Mozilla Firefox, which is regarded as an industry standard:
  + Pentium 4 or newer processor that supports SSE2
  + 512MB of RAM / 2GB of RAM for the 64-bit version
  + 200MB of hard drive space
* An internet connection with download speed equal to or exceeding 5MB/s (megabits per second)

### Host Server

* >5GB of available secondary storage with a high read/write rate
* A modern high performance server processor with >4 cores clocked at >3GHz
* >16GB of RAM
* An internet connection with download and upload speeds equal to or exceeding 1GB/s (gigabits per second)

## Success Criteria

# Old 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 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)