

Contents

[Analysis](#_gjdgxs) **2**

[Project background – The Problem](#_30j0zll) 2

[Current Solution](#_1fob9te) 2

[Client Requirements](#_2et92p0) 2

[Proposed solution](#_3znysh7) 2

[Project Overview](#_7wtklng8o4mc) 3

[Data Flow Diagram](#_2afc10ut0unh) 4

[Project Objectives](#_3dy6vkm) 4

[ERD](#_5axk7whcwr3a) 5

[Data Dictionary](#_mn9yeld75c4q) 5

# Analysis

## Project background – The Problem

The client for the project is The Music Rooms Selby, located just out of Selby town centre. The shop needs an efficient system for keeping track of stock and sales. Sales may consist of both music books and other items (reeds, mugs, straps etc.) so the system needs to be able to deal with many possible combinations of items and inputs.

## Current Solution

The current solution is very poor and very unorganised. The only way that the business has any knowledge of what products they have is through a paper ledger that comes with any new stock. A member of staff then must go through that ledger and check if the numbers in there match the number of products the store has. It also only gets checked every few months, rather than being updated every time a product is sold which could allow for the possibility of theft being mistaken for a sale. It also means that if the store were to run out of a particular book or item, it may not be noticed for several months or until a customer asks to purchase it only to find that there are none left in stock.

## Client Requirements

The client needs a system that will allow them to check the quantities of products in their store. It needs to all be stored in one place and allow for easy alterations to quantities and product lists. It needs to have a user-friendly interface and not have any complicated mechanisms so that it is easy to use and does not require someone versed in the program to help use it. It needs to have a function that alerts the client when a product is out of stock, or which reorders the product when it gets to a certain level.

## Proposed solution

A solution has been proposed for a system that will allow easier updates to stock inputs and outputs. It will have a login system so that the database is only able to be altered by those with access. It will allow for products to be stored in a database and have their quantities altered as and when is needed. There should be a search function so that the quantity of a product is able to be found quickly. It needs to be user-friendly so that it is worth using rather than manually checking everything. It needs to be able to keep track of stock levels and either raise an alert when a product reaches 0 or send a reorder form when the product gets to a pre-set level.

The data needed would include: a list of all current stock consisting of names, descriptions, and a total of how many the store currently has; a list of what stock comes from which suppliers; and a list of staff that would be needing to have edit access. The shop’s systems would need to be able to run Python and Microsoft Access

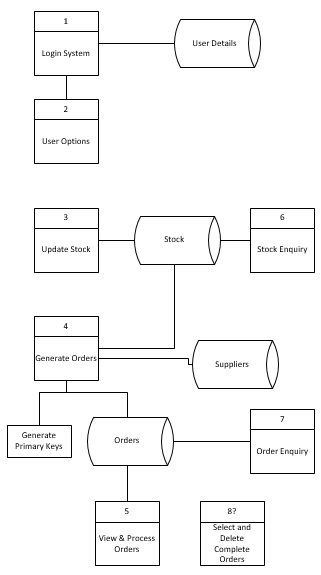
The program will check for any products that have a stock level of either “0” or a predetermined number and send out reorder forms. When it does this, it may also produce a list that is then brought up upon the next login, or directly emailed to the manager, of all of the products that were reordered.

## Project Overview

When the program is run, a user interface will appear with the option to allow users to login or continue as a guest. The only people who should be able to login should be members of staff which would give them permissions to alter stock levels, add new items, or remove items from the catalogue that are no longer going to be ordered in. If the user continues as a guest, they should just be able to view all the products that are currently being sold in the shop and how many of them are left.

There needs to be a database behind this with several connected tables: items, stock level, suppliers, customers, orders, and login details. The suppliers of the products would need to be stored so that the program could automatically send out a product reorder form when the stock level gets down to a pre-set point. The database would be made using Microsoft Access, the main programming with Python, and the user interface with the Tkinter library.

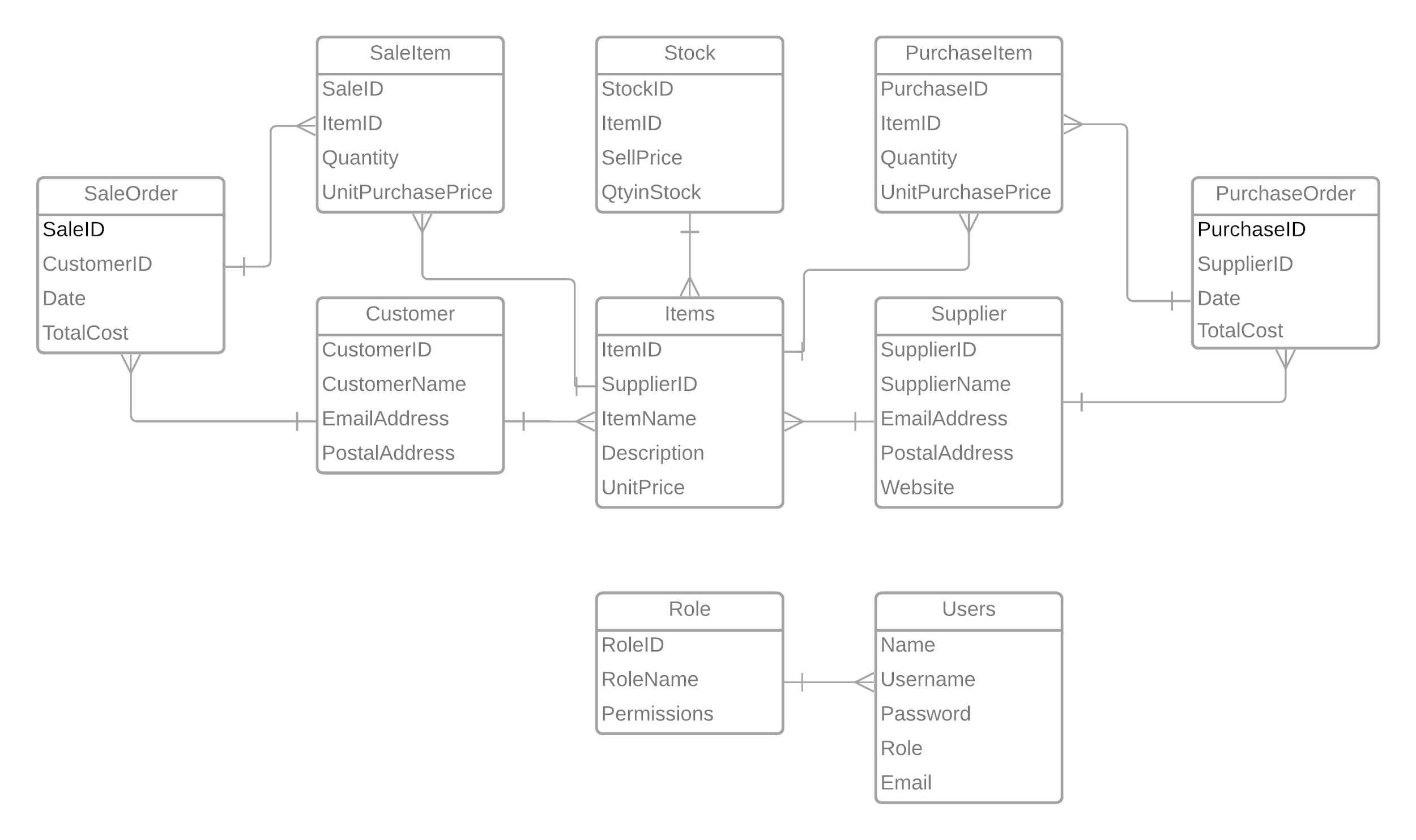
## Data Flow Diagram



## Project Objectives

1. Get authorised access
   1. Display tkinter welcome screen
   2. Be able to either login or continue as guest
      1. Login is only able to be performed by staff
         1. Asks for username
         2. Asks for password
         3. Encrypts password with hashlib
         4. Checks the inputs with data in the login table
         5. Prints error message if there are no matching entries
         6. Prints welcome message if there is a match
      2. Clicking to continue as guest will take user to the stock screen
2. See stock levels
   1. Have stock name, quantity and description printed out on each row
   2. Have a search function that allows the user to find a specific item easier
   3. Do this by printing the stock table onto the screen
   4. If logged in, there are extra functions
      1. Edit stock levels
         1. Users should be able to click a plus or minus button to add or remove stock
      2. Add new items
      3. Remove items that will no longer be restocked or sold
      4. For adding or removing items, there could be a drop-down menu that allows members of staff to choose what they want to do and then taking them to the appropriate screen or creating an appropriate pop-up menu
3. Automated reorder forms
   1. Once a product reaches a set level, send out a reorder form to the correct supplier
      1. Supplier and product tables will be linked
   2. Perform this either once every day or once every week
4. Have a working database
   1. Link the necessary tables
      1. E.g. link products to suppliers, customers to orders, etc.

## ERD



## Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | **Field Name** | **Data Type** | **Description** | **Validation** |
| Customer | CustomerID | Long Integer | Unique ID for each customer | Primary Key |
| CustomerName | Short Text |  |  |
| Email | Short Text | Customer email address for receipts or issues with order |  |
| PostalAddress | Short Text | If the order needs to be shipped |  |
|  | | | | |
| Items | ItemID | Long Integer | Unique ID for each item | Primary Key |
| SupplierID | Long Integer | Links each item back to the supplier | Foreign Key |
| ItemName | Short Text |  |  |
| Description | Short Text |  |  |
| UnitPrice | Currency | How much it costs to buy from supplier |  |
|  | | | | |
| PurchaseItem | PurchaseID | Long Integer | Links the items to the unique purchase | Foreign Key |
| ItemID | Long Integer | Foreign Key |
| Quantity | Number | How many were purchased |  |
| UnitPurchPrice | Currency | How much the item cost at the time of the purchase |  |
|  | | | | |
| PurchaseOrder | PurchaseID | Long Integer | Unique ID for each purchase | Primary Key |
| SupplierID | Long Integer | Links the Supplier to the item that the business is buying | Foreign Key |
| DatePurchased | Date | When was the purchase made |  |
| TotalCost | Currency | The cost of the whole purchase |  |
|  | | | | |
| Role | RoleID | Long Integer | Unique ID for each role | Primary Key |
| RoleName | Short Text |  |  |
| Permissions | Short Text | What a user with that role is allowed to do |  |
|  | | | | |
| SaleItem | SaleID | Long Integer | Links the sale to what has been sold | Foreign Key |
| ItemID | Long Integer | Foreign Key |
| Quantity | Number | How many were sold |  |
| UnitPurchPrice | Currency | How much was it sold for at the time |  |
|  | | | | |
| SaleOrder | SaleID | Long Integer | Unique ID for each sale | Primary Key |
| CustomerID | Long Integer | Links to which customer made the sale | Foreign Key |
| DatePurch | Date | When was the sale made |  |
| TotalCost | Currency | How much was the entire sale |  |
|  | | | | |
| Stock | StockID | Long Integer | Unique ID for each stock | Primary Key |
| ItemID | Long Integer | Links to which item is in stock | Foreign Key |
| SellPrice | Currency | How much the stock item is sold for |  |
| QtyInStock | Number | How many are currently in stock |  |
|  | | | | |
| Supplier | SupplierID | Long Integer | Unique ID for each supplier | Primary Key |
| SupplierName | Short Text |  |  |
| Email | Short Text | For contacting the supplier and for queries with orders |  |
| PostalAddress | Short Text |  |
| PhoneNo | Number |  |
|  | | | | |
| Users | UserID | Long Integer | Unique ID for each user | Primary Key |
| Username | Short Text |  |  |
| Password | Short Text |  |  |