16JPeach

[Email address]

Created for Furnival Press Co by Jude

Style for Code

AQA NEA Coursework

A quote production software for printing presses

Table of Contents

[Analysis 2](#_Toc118363839)

[***The initial Problem*** 2](#_Toc118363840)

[***Existing Systems:*** 2](#_Toc118363841)

[***Problems with Existing Software*** 3](#_Toc118363842)

[***Program requirements:*** 4](#_Toc118363843)

[***Client interview for stock management system*** 4](#_Toc118363844)

[***Business materials*** 6](#_Toc118363845)

[Design 7](#_Toc118363846)

[***Computational Methods*** 7](#_Toc118363847)

[***Initial Data Flow Diagram Drafts:*** 8](#_Toc118363848)

[***Initial Entity-Relationship Diagram drafts:*** 8](#_Toc118363849)

[***Calculating the most cost effective print method*** 9](#_Toc118363850)

[***Initial Screen/GUI design*** 9](#_Toc118363851)

Referencing:

AQA A level Computer science NEA programming project

A Quoting system for Furnival Press Printing Firm

Language – Python

# Analysis

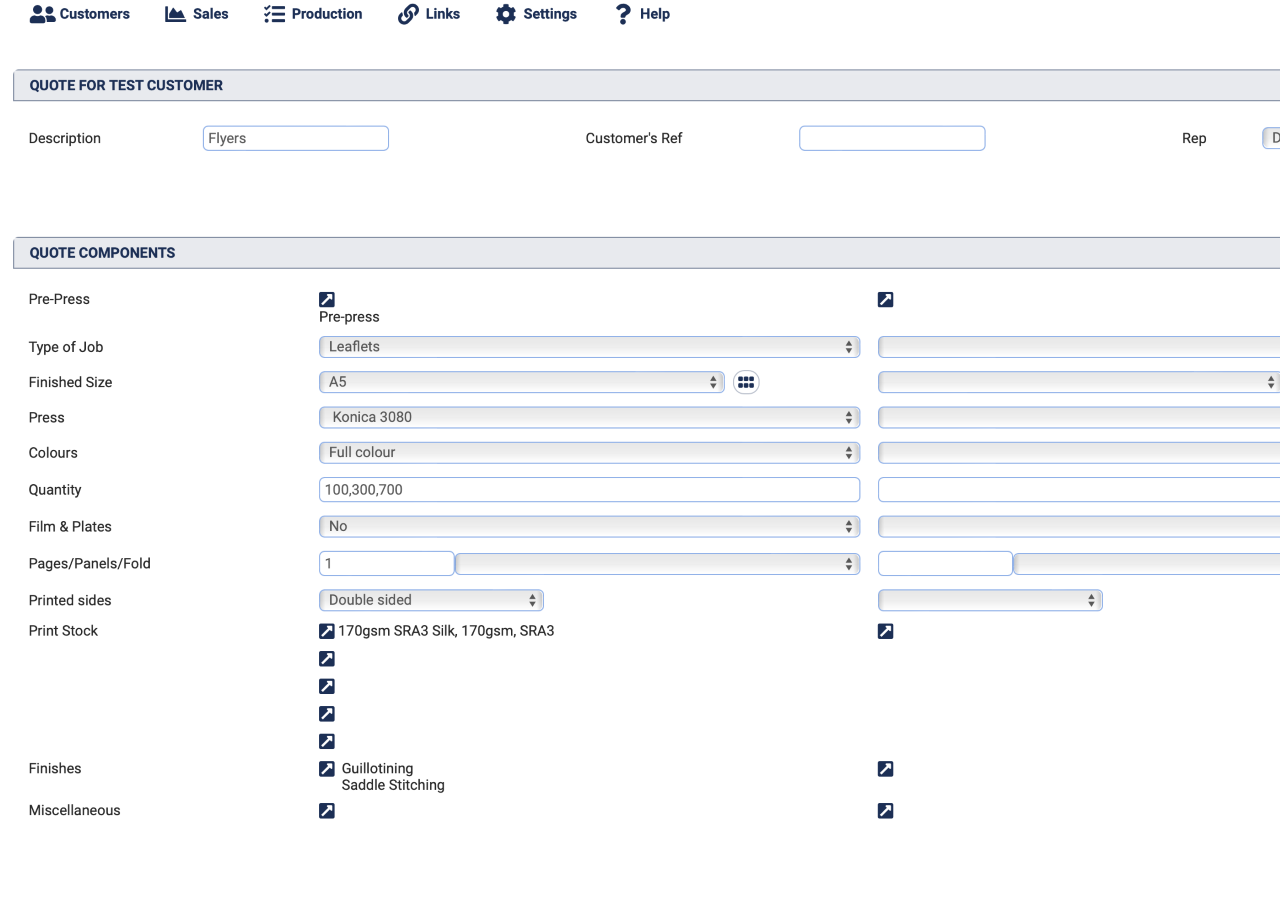
# The initial Problem

My father is the owner of a printing company named Furnival Press. They are based in Stockwell, London. They were established in 1862 and my dad has been working there for over 20 years now. Printing requires lots of complex calculations to generate pricing quotes for printing jobs, meaning that my dad pays lots of money annually for a quoting system which runs on a web server. This system is called PrintLogic and is not very functional and is very expensive. Due to printing becoming more and more of a rare trade in modern times, there are less and less quoting systems that are not perfectly tailored toward printing. For example, they do not contain all of the prices for paper types used by Furnival Press meaning certain quotes have to be done by hand.

Therefore I aim to provide my father with a program that is specifically tailored to his company, using all of his paper types and time rates.

# Existing Systems:

PrintLogic



PrintLogic is a bespoke print management software with over 1000 customers.

It calculates costs for print jobs and directly provides my dad with pdf work tickets and invoices. Work tickets are descriptions of how the job should be made, essentially instructions given to workers when printing jobs.

Work tickets and Invoices will be essential to my program since they are essential to running print jobs, therefore the program must create pdfs.

# Problems with Existing Software

Although PrintLogic is functional and accessible, it does lack in some areas

Print Logic is hugely expensive, and takes lots of profit away from the business, it’s starting cost is $115 monthly. My dad runs the business year round, so multiplying this by 12, means that $1380 yearly take home is being lost.

In addition to this, the software is not specific to Furnival press. The company is extremely old and well established. They are not a commercial printer and are more known for bespoke printing jobs, such as the Lincolns Inn annual review, which is read by Queen Elizabeth. This means that the company uses only the finest materials and offer endless hours of work to perfect each job. Therefore some materials are not available on PrintLogic, and the software provide inaccurate working hour rates due to certain jobs requiring much more attention to detail then others.

Essentially I aim to create a software that Is more suitable to bespoke printing jobs, which is more customisable than Print Logic covering all areas of Furnival Press, especially those areas that Print logic does not include.

# ***Client interview for stock management system***

(Analysis in red)

· Question 1 – What are the good things about your current stock management / invoicing system?

This system has saved so much time , it offers a complete package from the initial quote, creating work tickets and invoicing customers. It has helped us to win more customers, being able to offer quotes within minutes, that are accurate and above all profitable.

· Question 2 – What are the drawbacks of this system?

The system could have a easier to use modern interface, the system does look dated.

It would also be better if it linked to Xero and other systems more easily.

It would be good to do more as a user without having to ask the help of the programmers

· Question 3 – What is the price of this system and could you include a receipt if possible?

The cost is £89.99 monthly with no contract or upfront payment

· Question 4 – What functionalities would you desire to be added to the system?

To be able to add artwork files to the system and create a complete job folder.

Also to be able to use on a phone/tablet - Using SMART targets, this addition is unrealistic and unachievable within the time I have to complete the job

Easier integration with delivery/account software

· Question 5 – Could you describe the absolute basic requirements the program must absolutely include?

The quoting system is what drew me to the software.

· Question 6 – Could you generally describe the different things the system will need to consider when generating quotes? – eg – Machine costs, paper, number of copies and so on

How many items can be printed on a sheet, cost of paper, cost to use machine, and man hours + margin/profit - These are the main factors I am going to need to include within the database

The program will need to take into account (see ‘Business materials’ section for in depth explanations):

* The cost of each paper, dependant on paper size (A3, A2 etc..)
* The dimensions of a singular copy of a job, and therefore will figure out how many copies can fit on one paper sheet
* It should work out the most cost effective way to print a job
* The pay for workers who print the job
* The running costs of the printing machines used
* Any design charges (if the job is not already in a pdf)
* When the client needs the job printing by (may cost more to print faster)
* Special finishes like lamination or foiling

# Program requirements:

Now that the clients needs and programs basic variables have been established these are my final requirements:

1. The program should start up to an initial log in page
2. Users can log in, and the admin user must be able to add, remove and edit users and their usernames and passwords
3. Then the program should open to a main screen where each part of the program can be accessed from
4. One button will lead to quote history where previous quotes can be viewed and also exported as pdfs, this menu should also allow for edits to be made to the quotes, as well as allowing for deletion of quotes
5. Another button should lead to a screen used to create new quotes
6. The new quotes screen will allow for input of the following:

* Number of copies of the job
* The size of the job
* The colour options
* The paper type
* Any finishes
* Post and pre press charges
* Any machines or printing presses that will be used
* Any delivery charges

1. The program will store all of this information as variables and use the database to carry out any geometrical calculations – the most important factor is calculating the most cost effective printing method
2. From here it should return the calculations back into a new screen which displays the quote and what it is made up of
3. Then the user can check it over, make any edits before saving and printing
4. Another button should lead to window regarding the database
5. This window should simplify database changes into buttons and text boxes so users can easily amend the database perhaps using SQL statements
6. The database itself should be complex and relational including tables for:

* Products
* Customers
* Current jobs/orders
* Paper types
* Machines and presses

This will be done using sqlite3, I will create a separate py file and use OOP to more easily create and amend tables.

1. Each window should have a save and exit button which returns the user back to the main menu where they can also log out
2. The most important requirement is the ability to print pdfs and work tickets of all past and present quotes
3. Past quotes should be stored on a separate database
4. User access levels may be an option but since it is a self-ran business this may not be necessary until the future

# ***Business materials***

Through conversation with my dad I have created an in depth list of all the materials and processes the application will need to cover:

* **Paper types/sizes-** Within furnival press, hundreds of types and sizes are used each week. Most of the time, due to the small storage space at the office, paper is ordered in as and when it is needed rather than being stocked in-house. This is from a large paper company named Antalis:



Furnival press use mostly sizes that are referred to as **sra**(size) rather than a(size). These formats are slightly larger than the standard A formats and are the common use within printing. For example, sra3 is the same as a3 except sra is 320 by 450 mm whereas a3 is only 297 by 420 mm.

As well as size, each paper type also comes in different types classified by the weight. This is measured in grams per square metre (GSM), in office printing the GSM ranges from normally 80-100 gsm however in industrial printing like Furnival press it can range from 80 all the way to 450gsm which is normally used for things such as business cards. The higher the gsm, the thicker and stiffer the paper. Obviously, the gsm and size are both positively correlated to the price per sheet.

Paper is normally either coated or uncoated, with coated being more ink absorbent and smooth/glossy so is regularly used for magazines and books and uncoated being untreted paper is used In things like letters so is used less in the business.

During the design stage I will develop a database with tables for paper types and sizes/dimensions and will input the businesses entire paper list into the database including costs per sheet or per pack, all of the information will be taken from the Antalis website. This table can then be used to calculate the ways of printing the job A useful addition would be to perhaps add a section with links to the paper type on Antalis’ website.

This is where I will have to also implement/create an algorithm to calculate the most cost effective way to fulfil printing jobs, for example 100 business cards could potentially be printed on two sra3 sheets, but it may be more cost effective to print them on a singular sra2 sheet. This will become one of the most complex parts of the program as it will have to use geometrical calculations to work out the most efficient method of printing. Also the program should allow for manual entry of paper size, weight and type as sometimes there may be spare paper in the office to use rather than ordering a new pack.

* **Printing presses and ink –** The business currently uses around 6 industrial printing machines ranging from smaller printers for books and business cards to large printers specialised for banner prints and vinyl graphics. Each printer uses its own ink and toners and sometimes multiple jobs will be used per job. For example, when printing books the pages will be printed on one printed, then it may be laminated on another, scored and folded on the scoring machine then finally binded on the ring binding machine. Using SMART targets I have gathered that including very single ink type will be unrealistic as there are thousands and multiple will be used. Instead I will generalise and calculate hourly rates/price per print on each machine and implement it into the database as a table. Then my dad will be able to select each machine that will be used and enter how much time/sheets were printed/laminated/folded/binded on each machine.
* **Man hours –** For each job man hours can vary. Some jobs may be already in pdf format wit artwork complete so only need to be printed. Some jobs however will need artwork and design beforehand on adobe, this adds a cost to the quote depending on the magnitude of graphical work done, this is called Pre Press and will be a significant inclusion.

On top of Pre Press, there is also post press work that almost always must be done, for example banners may need mounting onto foam backing called Foamex or business cards may need guillotining. One real example of this comes from one of the companies’ clients ‘The Food Show’, this company orders prints to be wrapped around ready meals and these require double sided tape to be hand taped onto each print so they can be stuck around the meal. This will simply be added as a manual entry and my dad can manually enter an appropriate charge for both pre and post press work.

* **Finishes –** Finally any additional extras like foiling or lamination will also be include as an entry box for users to manually enter the extra charge.

# ***Data***

Diagram, engineering drawing

Description automatically generated

This is my initial data flow diagram. It shows the general process that the data will go through.

Firstly a customer will make an order, normally through emailing Furnival Press, or in the future, through the website. From this email the client will absorb all the important information that the customer has provide like size, colour, job type and so on. Then they will acknowledge the customers order by replying to the email, and also requesting any extra information that is needed from the customer.

At this point the client will first open the quoting software and log in. From here they will select new quote and start entering the information from the customer. This will create a new customer in the customer database including all of their information. Also the program will refer to the printing jobs database tables to create a new job in the job table, return the dimensions of sizes and rates of machines that will be used etc..

Once the customer is created on the database, all of the requested information from the database will be returned to the program. Then the algorithm will calculate the job cost and methodology, here a pdf quote will be accessible so that the customer can review the cost and agree to continue with the job. Finally, this info will be inputted into pdf work tickets and invoices. Then the job will be fulfilled by the print operators and the invoice will be sent to the customer for payment and the job itself will be mailed to the customer usually via DPD.

Also there will be a very small one table database for the login screen so that users can be added and removed, and no unauthorised users can enter:

Diagram

Description automatically generated

# ***Data Volumes***

***Paper*** – As stated in ***Business materials*** all of the paper is sourced from the company Antalis. I will gather their selection of paper from my dad which will include all of the paper weights, types and also the cost per sheet. This will include around 100 different entries within the database table, Antalis source thousands, but Furnival press uses around 100 types.

***Paper sizes*** -These can be gathered by a simple google search. The company only uses A or SRA formatted sizes. This table will therefore consist of around 8 sizes (sra1-sra7 and a1-a7). Each size will have its x and y dimensions in millimetres.

***Customers*** – Furnival Press has an active customer base of around 50, and occasionally gets new one time customers that need smaller jobs being printed. Therefore the customer tables will originally have around 50 customers, each normally having multiple jobs being fulfilled at a time. Any one time customers will be added when they enquire. The main customers include Lincolns Inn and The Food Show

A picture containing graphical user interface

Description automatically generatedGraphical user interface, website

Description automatically generated

***Machines*** – Printing presses are very versatile and most of the time a job can be fulfilled using just one, however some specialized jobs require multiple. The business has 8 machines from brands like Konica and Xerox, these include printing presses, foil machines, laminators, guillotines, scoring machines (folding) as well as binding machines for books. I will also gather the in depth information for these from my dad including the exact models names, running costs and time per print etc..

A red and white logo

Description automatically generated with low confidence

# ***Tools***

***Pycharm –*** This will not be my first time using pycharm, this is an IDE that will allow me to centralise a vast set of tools used in python all in one app. It works will all .py files so is compatible. It lays all of the files used in my project in one place giving an excellent project overview. This will allow me to work much more efficiently as I can view all my files easily without creating a messy desktop with lots of windows open. Also this will save lots of time as I will not have to configure python to allow for imports of certain functions.

***Tkinter*** – This is the main way that graphical user interfaces are created in python and I will be using this for all of my UI. It is effective because it very easy to use and allows for easy manipulation of things like buttons and labels so I can make a unique looking application using functions like sticky and fonts.

***SQLite3*** – I have experience using this within my A level computer science lessons and so I am choosing this to implement my database as I am already fairly proficient in using the module.

Finally, any information and data will mostly be gathered directly via email or chats with my dad, as well as simply using google to find prices, paper sizes etc..

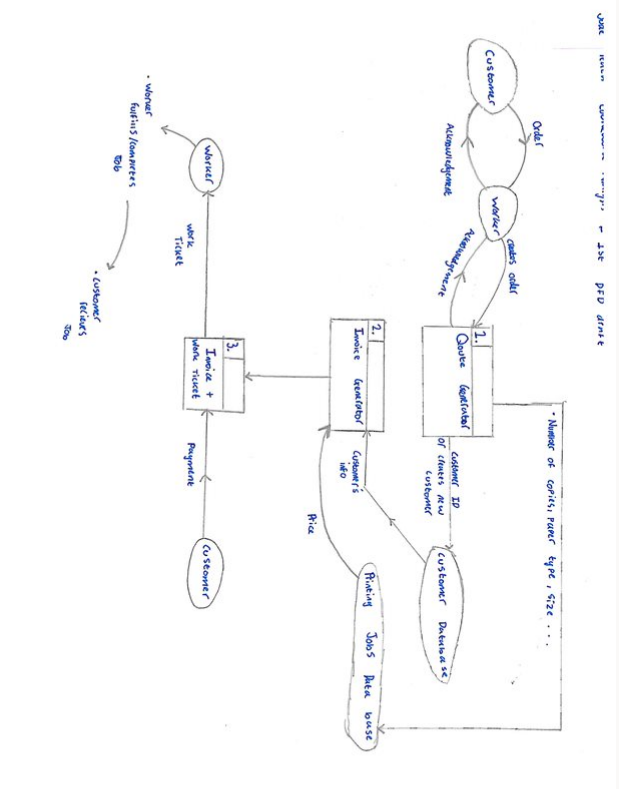
# Design

# Computational Methods

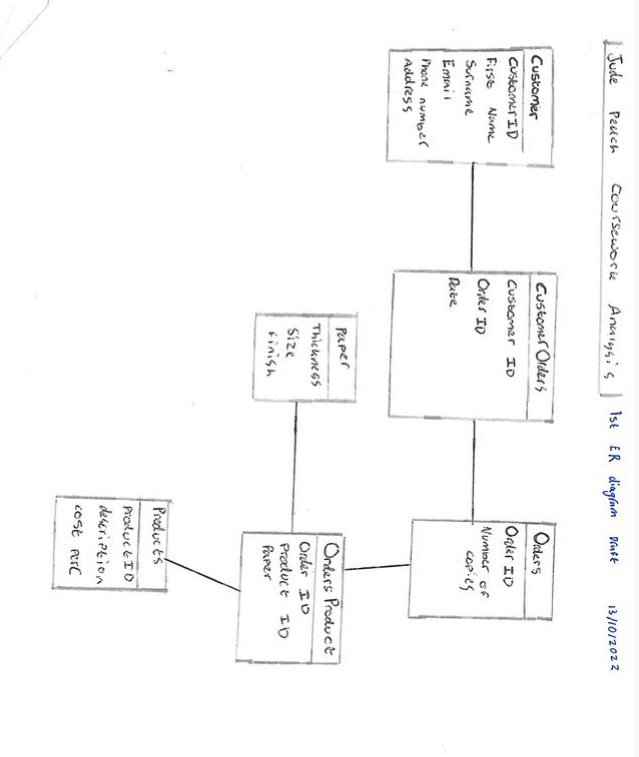
For the main GUI element I will be using object orientated programming, defining a class for the main interface, this should consist of multiple screens/windows for different parts such as the login menu and receipt menu.

I will also implement a complex inter linked data base with multiple tables so that receipts and past orders can be viewed, edited and amended. I will do this with a python module called SQLite which allows for database implementation.

# Initial Data Flow Diagram Drafts:



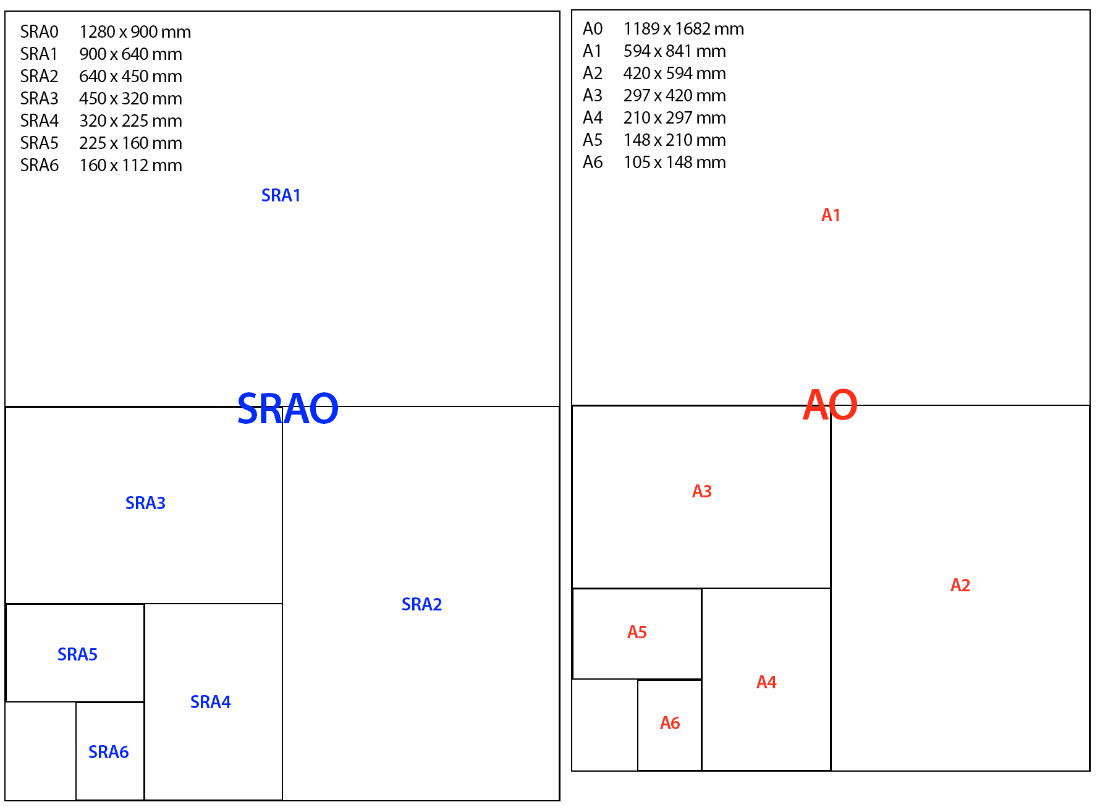
# Initial Entity-Relationship Diagram drafts:



# ***Calculating the most cost effective print method***

Paper Calculations:

* We can provide the user the option to chose the size they will be printing on, and the size that the finished print will be
* Then we can find the maximum amount of copies that can fit onto one sheet of paper.
* For example, if we are printing onto an a3 sheet and they are leaflets of size a4 then we must calculate: A4 sheet measures 210mm x 297mm making its shortest edge 210mm long. An A3 sheet measures 420mm x 297mm. so to move up a size 210 x 2 = 420. This gives an A3 sheet the flat size equivalent of exactly **2 A4 sheets**.
* We will need to make the program be able to complete these calculations automatically
* This means I will implement a Paper size table in the DB so that all of the dimensions are stored ready for use in calculation
* This calculation is fairly simple when using similar sizes, however when printing large scale jobs, especially large jobs which are strange sizes the calculations will become very complex.
* For example, if the business needs to print ten thousand round edged business cards the minimum paper amount must be used otherwise profits will be lost via waste cuts of paper.



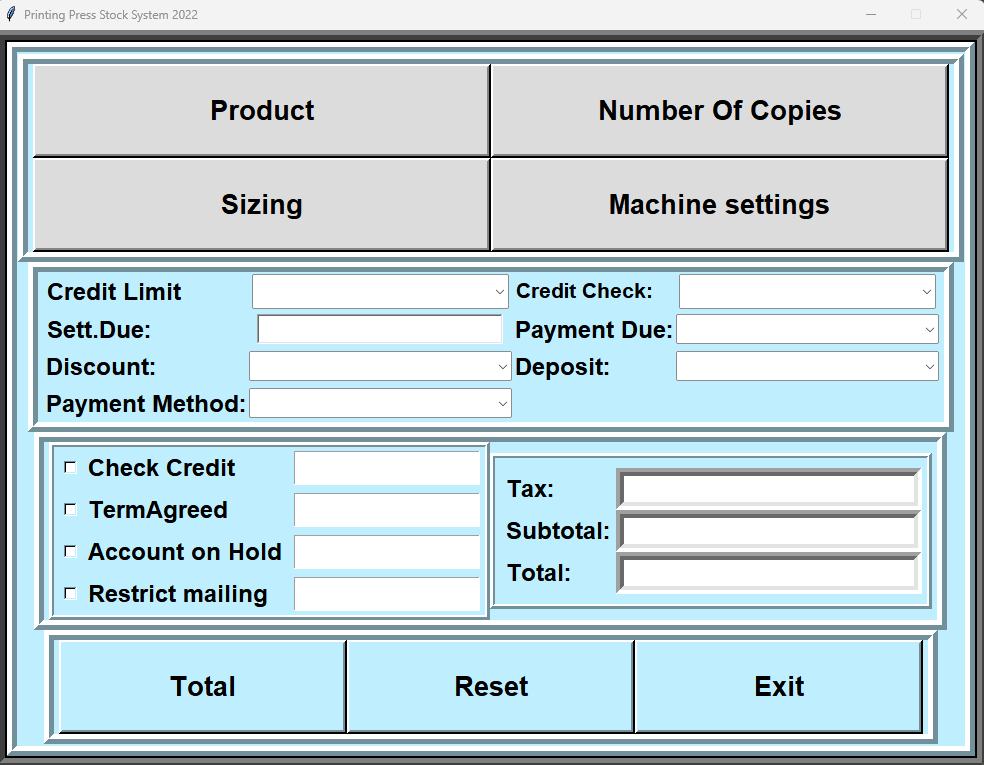
Machine costs:

* I will implement a table within the database which specifies the cost of each machine by either cost per sheet or cost per hour of run time.
* Then for optional extras like lamination, gold foil etc, I will add an option for workers to add this manually since this is normally a fixed extra charge rather than a rate per hour etc.

# ***Initial Screen/GUI design***

Bar chart

Description automatically generated with low confidence



Graphical user interface, application

Description automatically generated

# ***First database design and sql implementation***

**Data Tables:**

* For the foreign keys to be implemented, the pragma keys setting must be set to ON, since sqlite3 does not use foreign keys automatically:

A picture containing logo

Description automatically generated

**Customers:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Key | Data type | Validation | Notes/Comments |
| CustomerID | Primary key | String | -------------- |  |
| Firstname |  | String | 1-20 |  |
| Surname |  | String | 1-40 |  |
| Email |  | String | 1-50 |  |
| Phone |  | Integer | 9 or 10 digits |  |

**Users:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Key** | **Data type** | **Validation** | **Notes** |
| **UserID** | **Primary key** | **String** | **3 characters** |  |
| **UserName** |  | **String** | **------------** |  |
| **Password** |  | **String** | **One capital letter, one lower case, one number** |  |
| **AccessLevel** |  | **String** | **1-3** | 1. **Lower access users** 2. **Mid access** 3. **Admin/manager** |

**OrderItem:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Key** | **Data Type** | **Validation** | **Notes** |
| **OrderitemID** | **Primary key** | **String** |  |  |
| **OrderID** | **Foreign key** | **String** |  | **From Ordertable** |
| **ProductID** | **Foreign key** | **String** |  | **From Products** |
| **Quantity** |  | **Integer** |  |  |

**Ordertable:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Key** | **Data Type** | **Validation** | **Notes** |
| **OrderID** | **Primary key** | **String** |  |  |
| **CustomerID** | **Foreign key** | **String** |  | **From Customers** |
| **Date** |  | **String** | **Allows for easier database searches when looking for past orders** |  |

**Products:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Key** | **Data Type** | **Validation** | **Notes** |
| **ProductID** | **Primary key** | **String** |  |  |
| **ProductName** |  | **String** |  |  |
| **Description** |  | **String** |  |  |
| **MachineID** | **Foreign key** | **String** |  | **From Machines** |
| **PaperID** | **Foreign key** | **String** |  | **From Paper** |

**Machines:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Key** | **Data Type** | **Validation** | **Notes** |
| **MachineID** | **Primary key** | **String** |  |  |
| **Name** |  | **String** |  |  |
| **Description/brand** |  | **String** |  |  |
| **Automated** |  | **Boolean** | **To determine whether it requires charge for physical labour** |  |
| **CostPerPrint** |  | **Float** | **Will be used in quote generation** |  |

**Paper:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Key** | **Data Type** | **Validation** | **Notes** |
| **PaperID** | **Primary key** | **String** |  |  |
| **Supplier** |  | **String** |  |  |
| **PackSize** |  | **Integer** | **All of the below will be used in quote generation** |  |
| **CostPerPack** |  | **Float** |  |  |
| **Weight(GSM)** |  | **Float** |  |  |
| **Size** |  | **String** |  |  |
| **Finish** |  | **String** |  |  |

# **Entity relationship diagrams**