# 0.Introduction

Sign in with 3 roles :

1. **CUSTOMER**
2. **SELLER**
3. **ADMIN**

Or register with a new customer or seller (admin is not registerable )

General function:

* View the Goods list
* Sort the Goods list (with different sorting methods)
* Search for desire goods
* Filter the desired good from the goods list

Unique function:

1. **Customer**

* Customers can add goods to their own shopping cart
* Customers can also adjust (i.e. the number of goods) and delete the goods in their shopping cart. At last, they check out
* They can then select a payment method and address for delivery

1. **Seller**

* Seller can add goods of their brand to the supermarket (addressing the problem of insufficient goods variety and promote the sales , fame of the market )
* The seller can also adjust (i.e. managing the stock , price of the goods )
* The seller can delete THEIR OWN goods (can’t delete other companies’good)

1. **Admin**

* admin can adjust, add, delete the goods in the list without any limitation
* admin can find the statistics data of the sales of the goods

# A screenshot of a computer program Description automatically generated

# 0.1 Modular Approach

Modules like csv, datetime are used to enhance the functionality of the program and also allow the easier maintenance of for the programmer and easier to debug

# 0.2 Storing of Data

**0.2.1 - goods\_data**

It stores the data for the goods

A computer screen shot of text

Description automatically generated

The data of goods is stored in csv file, which can be read in the module CSV , imported in the 0.1 , stores the **NAME,ID,COMPANY,PRICE,STOCK** of the goods available in this online platform for the customers to purchase. Only **ADMIN** and the **SELLER** of those goods can adjust the data of the goods

**0.2.2 - pickup\_data**

It stores the data for the pickup locations

A screenshot of a computer screen

Description automatically generated

The data of pickup locations is stored in csv file. **NAME,ADDRESS,PHONE,START,CLOSE** are stored in the file ( START is the opening time of the shop , while CLOSE is the closing time for the shop) which are the data of the shops the allow the customers to pickup what they purchased after the ordering online.

**0.2.3 – purchase record**

**It** stores the records of the customers’ purchase

A screen shot of a computer program

Description automatically generated

The purchase records of customers stores **TID,NAME,ID,PRICE,QUANTITY,TOTAL,CUSTOMER,COMANY,TIME**

Those are the data of the customers’ purchases, the TID is the unique transaction ID to identify different transactions. **ADMIN** can view the statistical data of those transactions and adjust the stock or price of the goods

**0.2.4 – data of users**

The login names and passwords of ALL users are stored in form of lists.

No other file is used to store that data as those data should not be frequently changed and in concern of privacy.

**0.2.4 – Main loop**

A screen shot of a computer code

Description automatically generated

This is the main loop of the whole program , where user has to log in

The menu displays the available commend for the user under his permission, while menu\_control allow users to use commends to show or change the data ( further explanation are presented in this report) and the flag\_bit indicates the start/end of the whole program.

# 1.Login of users

A screen shot of a computer program

Description automatically generated

The users can choose if they are existing users (admin, customer, seller) or register as a new user by inputting the representing short form (A/C/S/NEW)

While their login function is not correct , they have to retry until a valid input is inputted

Different identity has different login functions

A function of creating new users is used for NEW user

* 1. **– admin Login**

**A screen shot of a computer code

Description automatically generated**

The login name is inputted by the user. If the input of the users is in the list of admin login name, password is needed for the user to input to log in If both are correct , the login is successfully for the admin

If the password is not correct or the username does not exist, the user have to re-enter the username and the password again

A black screen with text

Description automatically generated

A find function is used to find the index of the input of username in the list of admin name. **matrix** (parameter) will be the admin list, as we are searching for the name of the admin , and **target** (parameter) will be the input of user , which we want to find in the name list of admin

**OUTPUT of admin login :**

**A black screen with white text

Description automatically generated**

* 1. **– seller login**

**A computer code on a black background

Description automatically generated**

The login name is inputted by the user. If the input of the users is in the list of seller login name, password is needed for the user to input to log in . If both are correct, the login is successfully for the seller

**OUTPUT of seller login:**

A black screen with white text

Description automatically generated

If the password is not correct or the username does not exist, the user has to re-enter the username and the password again

**1.3 – customer login**

**A screen shot of a computer code

Description automatically generated**

The login name is inputted by the user. If the input of the users is in the list of customer login name, password is needed for the user to input to log in . If both are correct, the login is successful for the customer

If the password is not correct or the username does not exist, the user have to re-enter the username and the password again

**OUTPUT of customer login :**

A black screen with white text

Description automatically generated

**1.4 - new user registration**

**1.4.1 - registration**

Two different identities / roles, **seller** and **customer** , can be registered ( **admin** cannot be registered as it involves permission of viewing sensitive information )

If the input name is unique (not in the seller or customer name list respectively ) , the password is needed to be created by the users , which certain rules have to be followed .

For the customer, a birthday has to be inputted for a potential discount and the most important is to confirm the customer can legally purchase alcohol, as the customer under 18 years of age cannot buy alcohol.

**A screen shot of a computer program

Description automatically generated**

The date of the customer should be input by the function date input

Module **datetime** is used in this function for the use of type “datetime”

If the input of the user is not a integer , the customer have to re-enter a integer input ( Error handling )

**A computer code on a black background

Description automatically generated**

**1.4.2 – password**

Validation: The password should have the following rules:

* 1. Not shorter than 8 characters
  2. Not longer than 20 characters
  3. At least have 1 capital letter
  4. At least have 1 number
  5. No space in the password

If any rules is against, the re-entry of the password by the user is needed until the input is valid.

Verification: The user has to input the password twice to confirm the password, to prevent the

mis-input by the user

**A screenshot of a computer program

Description automatically generated**

**OUTPUT of new user registration:**

**A screenshot of a computer program

Description automatically generated**

# 2. Goods viewing

**2.0 – Menu**

The goods can be viewed in different ways , which users can show different data of goods in different ways by different commends

A screen shot of a computer program

Description automatically generated

permission\_stat is used to store the permission of different users ( 1 – Customer , 2 – Seller , 3 – Admin ) , customer can view their shopping cart and check out but the other two identities cannot

Seller can edit their own goods data while customer cannot. Admin can edit all the goods data without limitation and view the statistical data while the other two cannot.

Only the user with permission to use the command can see the command.

By typing the commend short form respectively, different commands can be executed to view different data

OUTPUT of menu:

A screenshot of a computer screen

Description automatically generated

\*Situation: The menu for the customer

**2.1 – Viewing of data in different ways**

The data of goods can be viewed in :

**2.1.1 - Sort by Name**

**2.1.2 - Sort by Price**

**2.1.3 - Sort by goods ID**

**A screen shot of a computer code

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

\*a line of **view(goods)** is at the end of for viewing the data in default

By inputting the commands respectively, different ways of presentation of data can be used.

The “with” commend open the file of goods\_data in read mode, which can be closed automatically at the end. The data of the goods is extracted by the module CSV, which provides a way to unpack the content of the csv file and forming a 2D array , goods , for the manipulation of the data in python .

**2.1.0 Viewing data**

**A computer code with colorful text

Description automatically generated with medium confidence**

The 2D array of the goods data can be presented formatted by the view function

The **data** (parameter) , a 2D array , should be input and a formatted table will be outputted .   
The view function will be frequently used in the display of 2D array like goods data , shopping cart.

**OUTPUT of view:**

(This output is based on the default order of the goods i.e. according to the time added into the data of the good)

A screen shot of a computer

Description automatically generated

**2.1.1 Sort by Name**

**A computer screen shot of code

Description automatically generated**

The data of goods is input to the selection sort function as the raw\_list parameter , and the index marks the **NAME** column in the file of goods\_data , in this case is **1.** The start parameter is **1** , stating the first row of the file of goods\_data should not be sorted , which is the row of heading

The Selection sort algorithm find the smallest element and swap it with the first element, getting the smallest element at its correct position. Then the smallest among remaining elements is to move it to its correct position by swapping. This process will be repeated until all the elements in this list in sorted.

**OUTPUT of SN:**

**A screen shot of a computer

Description automatically generated**

**2.1.2 Sort by Price**

**A black screen with white text

Description automatically generated**

The data of goods is input to the selection sort function as the raw\_list parameter, and the index marks the **PRICE** column in the file of goods\_data , in this case is **3.** The start parameter is **2** , stating the first row of the file of goods\_data should not be sorted , which is the row of heading

The bubble sort works by comparing the item with the next item, if the next item is smaller than the item, they swap their position. Hence the largest item will sort to the last position. The process repeats until all the items are sorted

**OUTPUT of SP:**

**A screen shot of a computer

Description automatically generated**

**2.1.3 Sort by SID**

**A screen shot of a computer program

Description automatically generated**

The data of goods is input to the selection sort function as the raw\_list parameter, and the index marks the **SID** column in the file of goods\_data , in this case is **1.** The start parameter is **2** , stating the first row of the file of goods\_data should not be sorted , which is the row of heading

The insertion sort works by comparing second element with the first element and check if the second element is smaller the first element, if so , swap them. Repeat the process for the second element and the third element , and so on . The sorting will be done .

**OUTPUT of SID:**

A screen shot of a computer

Description automatically generated

**2.1.4 Searching**

A computer screen shot of a computer code

Description automatically generated

The function “S” and “SID” using the search function

It works by first creating an empty list. Looping through the raw\_list (parameter), a 2D array, and find the item in the specific position is the same with the desired item(parameter), if so , add the whole item into the new\_list and return to show . In the case of S and SID , the index will be **0** and **1** , stand for the **NAME** and the **ID** in this case respectively.

A screen shot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

A black screen with white text

Description automatically generated

# 3.Filtering

A computer screen with text on it

Description automatically generated

A computer code on a black background

Description automatically generated

A screen shot of a computer code

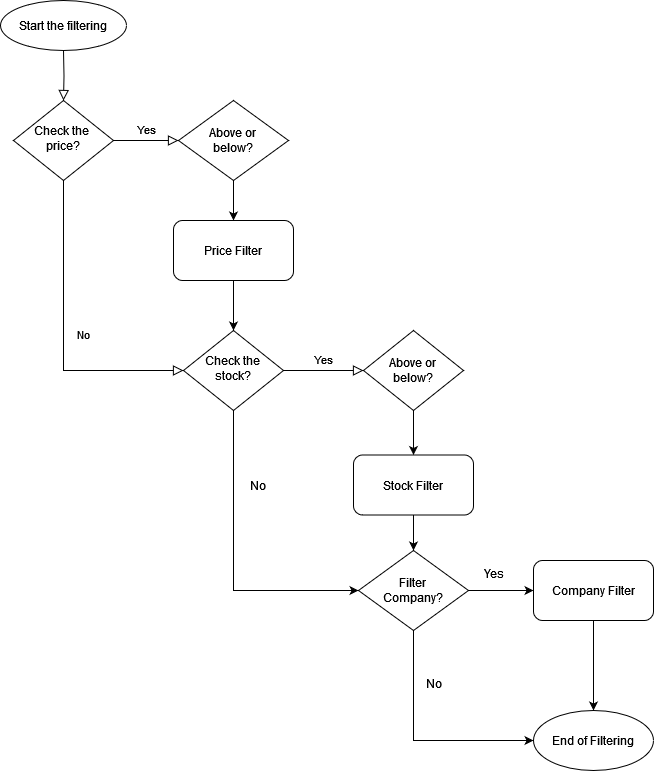
Description automatically generated

The filtering will have 3 processes: Filter of Price, Stock, Company

Each of them can choose filter or not to filter it.

Error handling is used in this case when the user does not input a float or a integer (for example string), Error message will pop and the user have to redo the whole filtering process. The filtering function is used to find the desired output of each filtering process (i.e. will do three times at most) and the compare function is used as to compare the result to the main list and only left the items that satisfy the filtering

**Illustration of the filtering process**



**OUTPUT of filtering:**

A computer screen with white text

Description automatically generated

A screenshot of a computer program

Description automatically generated

# 4.Manupulation of Goods Data

**4.1 – Add goods**

Only seller and admin can add the goods to the goods data, but not customer, by permission check function. The seller can only add goods with their company’s name while the admin can add goods with any company name without limitation.

**NAME, ID, PRICE , STOCK** of the goods are needed to fill in to add a good to the file of goods data (**COMPANY** is needed only by admin) If the inputs are less than 4 , meaning the missing of at least 1 item, will have quit to the main menu and re-enter the whole inputs of 4(Error handling). Verifications are also required as the **ID** must be unique. Also the number of stock and price should not be less than 0. A age check will be needed for the alcoholic goods, which ID of that goods will be added to the list of age checking.

A computer screen shot of text

Description automatically generated A screen shot of a computer code

Description automatically generated

A screen shot of a computer

Description automatically generated

**OUTPUT of add\_goods:**

**A screenshot of a computer program

Description automatically generated**

**Before adding goods:**

**A computer screen shot of text

Description automatically generated**

**After adding goods:**

a new good is added to the good list

**A screen shot of a computer program

Description automatically generated**

**4.2 Modify goods**

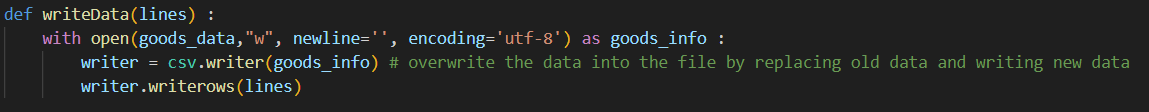
**A computer code on a black background

Description automatically generated**

**\*This is a part in the menu function for modification of goods**

**A screen shot of a computer program

Description automatically generated**

****

The modify function is only available to seller and admin, which sellers can only change the goods that belong to their company while admin has no limitation. After the opening of the goods\_data file and store data into the lines(variable), if the ID input can be found in the goods\_data , the **PRICE** , **NAME** ,**COMPANY** the **STOCK** can be changed ( **ID** is not allowed to change in order to keep the consistency of data) , the input for **PRICE and STOCK** must be larger than **1.** The data of that good in lines is changed, and then writeData function rewrite the whole goods\_data file with the data stored in lines for modification of the data.

**OUTPUT of modify :**

**A screen shot of a computer

Description automatically generated**

**Before the modification :**

**A screen shot of a computer program

Description automatically generated**

**After the modification :**

**A screen shot of a computer program

Description automatically generated**

**4.3 Delete of goods :**

**A black screen with colorful text

Description automatically generated**

**\*This is a part in the menu function for delete of goods**

**A screen shot of a computer code

Description automatically generated**

The delete function is only available to seller and admin, which sellers can only delete the goods that belong to their company while admin has no limitation. After the opening of the goods\_data file and store data into the lines(variable), if the ID input can be found in the goods\_data , the good of that ID is delete in the data of lines.ASthe data of that good in lines is changed, writeData function rewrite the whole goods\_data file with the data stored in lines for deleting of the data.

**Output of delete :**

**A black screen with white text

Description automatically generated**

\* Situation: The company cannot delete the goods belonging to other companies

**A black screen with white text

Description automatically generated**

**\*** Situation: The goods cannot be deleted if the good does not exist

**A screen shot of a computer screen

Description automatically generated**

**\*Successful delete**

**Before delete :**

**A screen shot of a computer program

Description automatically generated**

**After delete :**

**A screen shot of a computer program

Description automatically generated**

# 5.Shopping cart

Shopping cart can only be used by the customer , but not seller or admin .

Shopping cart shows the **NAME,ID,PRICE,Quantity,Total cost** of the goods , the total cost is calculated by the price multiplied by the quantity, customer can go check out after adding item into the shopping cart

**5.1 – Add cart**

**A computer code on a black background

Description automatically generated**

**\*This is a part in the menu function for add cart**

Customer needs to input the ID and the desired quantity. Error handling for invalid input here.

**A screen shot of a computer code

Description automatically generatedA black background with colorful text

Description automatically generated**

Shopping cart is a 2D array which stores the heading at first. Customer has to input a ID and a quantity to purchase. The add\_cart function can only be used for the goods that are not in the shopping cart yet, if the customer wants to add quantity of the goods that are already in the shopping cart, modify\_cart function should be used. After opening of the goods\_data file, the data of the desired good with given ID will be found (if it exists) and the age\_check function checks the customer is above 18 years old if the good has age limitation, an ERROR message will be given if the customer does not satisfy that. If the stock of that good is more than the quantity needed

OUTPUT of add\_cart:

A screen shot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

\* Situation: Customer under the age of 18 cannot buy alcoholic goods

A screen shot of a computer

Description automatically generated

\* Situation: The ID does not exist in the goods\_data

A black screen with white text

Description automatically generated

\*Situation: The stock of the good “001” is not enough for the customer to purchase

**5.2 View cart**

**A black screen with colorful text

Description automatically generated**

**\*This is a part in the menu function for view cart**

The shopping cart is viewed by the function view which showed at the previous part of this report

Only customer can access the shopping cart but not the admin and seller

OUTPUT of view cart:

A black screen with white text

Description automatically generated

**\*** Situation: The goods 001,002,003 are added to the shopping cart with quantity 1,2,3 respectively

And the shopping cart is viewed.

**5.3 Delete cart**

**A computer screen shot of text

Description automatically generated**

**\*This is a part in the menu function for delete cart**

Only customer can access the shopping cart but not the admin and seller.

The ID of the good which is desired to be removed from the shopping cart has to be input.

If that ID is found in the shopping cart , the good is removed from the shopping cart by pop()

**OUTPUT of Delete cart :**

Before delete :

A black screen with white text

Description automatically generated

\*Situation: The goods 001,002,003 are added to the shopping cart with quantity 1,2,3 respectively

And the shopping cart is viewed.

**A screen shot of a computer

Description automatically generated**

\*The delete cart commend

A black screen with white text

Description automatically generated

\*Situation: The viewing of shopping cart after the delete cart commend is executed

# 6.Check Out and collection of goods

Customer can check out after their satisficing shopping and plenty of goods are added to the shopping cart . The check out can choose two different methods, delivery or pickup ,for collection of purchased goods.

**6.0 – Check out**

**A screen shot of a computer program

Description automatically generated**

**A computer screen with text

Description automatically generated**

The flag\_bit indicated the end of the whole program when it changes to False.

The sum of all the goods are calculated in the check\_out function ( excluding the heading in the shopping list to avoid error) by adding all the amount of the total\_cost of every goods in the shopping cart.Customer is not allowed to check out if there is nothing in the shopping cart .

**OUTPUT of CO :**

**A black background with white text

Description automatically generated**

**6.1 Delivery**

**A computer code on a black background

Description automatically generated**

In the delivery function, the customer has to first input their location. The date\_input function (presented at the previous part of the report) is called for the user to input a date for the delivery. Datetime module is used to calculate the available range for the delivery (in between 7 days after the checkout and 30 days after the checkout , based on the real time of the checkout has been done) And verify the customer's input of date is in the range or not, if not, then an error message will be showed. If the input of date is valid, the customer can choose the time of delivery (AM or PM) . If the input of time of delivery is valid , the payment function will be called.

A screenshot of a computer

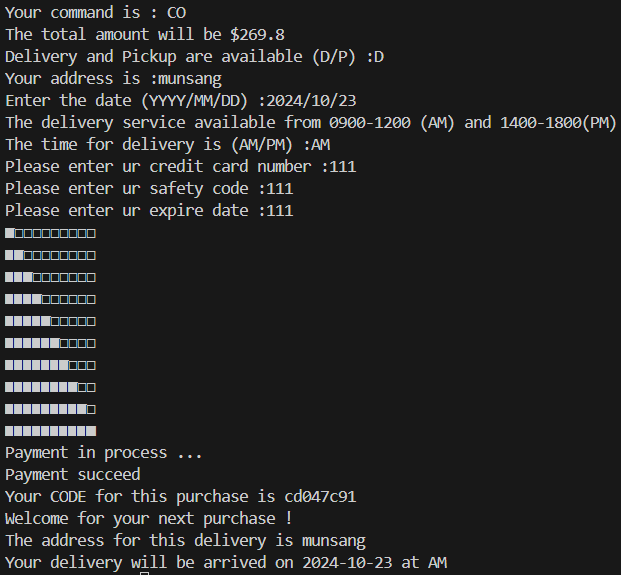
Description automatically generated

A computer code on a black background

Description automatically generated

Several information of the credit of the customer are needed to be input. An animation of the payment will be displayed for a demonstration of the payment process. Finally, a transaction ID will be created by the transaction function , in which uuid module is used for generating a 8- characters purchase code for the customer to act as a receipt ( it will be used in pickup , which will be explained in the later part) and the confirmation message of the delivery will be displayed to the customer for reference.

OUTPUT of the Delivery:



**6.2 – Pickup**

**A screen shot of a computer code

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

The pickup locations / shops available for the customer to collect their goods will be shown by the view\_pickup function, which opens the pickup\_data file and present the 2D array in a formatted way. The pickup date will be input by the customer by the date\_input function. The date should in the range of the day after the checking out and the 30 days after the day of checking out. The customer has to choose the location that they want to pickup at. The time of the pickup needs to be input and it should be within the opening hours of the shop, which will be shown by the view\_pickup function. Transaction will be called afterwards to complete a time of purchase

**OUTPUT of Pickup:**

**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

# 7.Data visualization

To be completed