Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

Universiti Tunku Abdul Rahman

Bachelor of Computer Science (hons)

UCCD1004 Programming Concepts and Practices

Session 2019 /05

FINAL REPORT

Lecturer Name: Dr.Jasmina Khaw Yen Min

Group Title: Shopeeee (An online shoping system)

Group : 1

No.	Name	Student ID
1	TAN JING JIE	1804560
2	JACYNTHTHAM MING QUAN	1801600
3	TAN WEI MUN	1803705
4	NG KWAN HOU	1803606

Table of Content

Introduction

In the modern era, the popularity of online shopping systems is rising among the younger generation. The convenience of buying item with just several clicks has even captured the attention of the elderly and disabled. To compete with the numerous online shopping systems out in the market, we have decided to create our own online buying and selling platform – Shopeeee.

Our online shopping system, Shopeeee, is designed to cater to all users. Whether old or young, tech-savvy or the technically-challenged, you can rest assure that that you can use our application with no issues. Shopeeee features a simple-to-use, straight to the point user interface that allow users to interact with each other through buying and selling goods.

Users can choose to have a member account or an admin account. All of the user's details are saved to our secure external database. For maximum security, all users' passwords are encrypted using the concept of ASCII code to prevent leak of data.

A member account allows users to shop in our vast product catalogue and buy items. We accept both credit or debit cards and cash on delivery as our payment methods. Here at Shopeeee, users are guaranteed to enjoy fast, smooth and safe transactions.

An admin account allows users to turn "trash" into cash by simply putting the item up for sale. Our record management system makes it extremely convenient for new users to update, add and delete products. Shopeeee also automatically generates a summary of the products sold, complete with the time of purchase and total income earned. This helps inexperienced sellers to keep track of how well their business is doing.

Shopeeee also features a feedback system. Customers can choose to provide feedback on the system, the admin's service or a particular product. These feedbacks will be saved to the system and can be viewed by the admins. The purpose of this is to allow admins to improve their service and the quality of products sold. Sellers with the admin role can also provide feedback, which will be reviewed by the system's admins to decide how Shopeeee can be improved in future updates.

Assignment Objectives

1. To transform the manual process of buying and selling to a computerized system
2. To allow members and admins to create account and enter the online shopping system
3. To allow users to recover forgotten passwords like a real account-handling system would
4. To allow users to manually update their own details when necessary
5. To allow admins to keep products' details updated at all times
6. To restrict admins to only managing their own products, and not those of other sellers
7. To allow admins to remove products that are outdated or out of stock
8. To allow admins to add and update products to the system
9. To allow admins to view feedback provided by members
10. To auto-generate a summary of sales for the convenience of admins
11. To allow members to view and purchase products put up by admins
12. To enable system to auto-generate an invoice for customers with every purchase
13. To enable system to keep all saved passwords as randomized data through ASCII code encryption.
14. To enable system to grade the name of product in a record list to output the highest possibility of product which user want to search.
15. To enable system to detect the any invalid input which will make the system corrupted or giving the wrong output content.

Job Task

Member	Modules	Roles	Description
Member 1: Tan Jing Jie	Add record	Admin	Add the information of items such as the product code, item name, stock number and unit price.
	Update record	Admin	Update the information of the items such as item name, stock number and unit price via view entire product or by search. Stock number depletes automatically upon purchase by customers.
	Log in	Admin/ Member	Displays the menu and the login page for users, whether admin or member. An option is given to users to create a new account. The login module also features an option to reset any forgotten password.
	Encrypt system		Encrypts the user's password and favorite number (a 12-character long number which assists users in resetting their password). This is to ensure that hackers that can access the external database will not be able to gain access to the users' accounts. Upon login, the system encrypts the user's input and matches it against the encrypted password stored in an external database.
	Print layout		To display layouts in external text files.
	Update file		To replace a file by another file such as temporary file to original file.
	Validate input		Ensure that the user's character, string, integer and float input is of the correct data type, length and within a given range
	Grading system		Prepare a grading system for search to grade every record related in the according to the search to give out the similarity mark. The grade will be arranged by arrange record system.

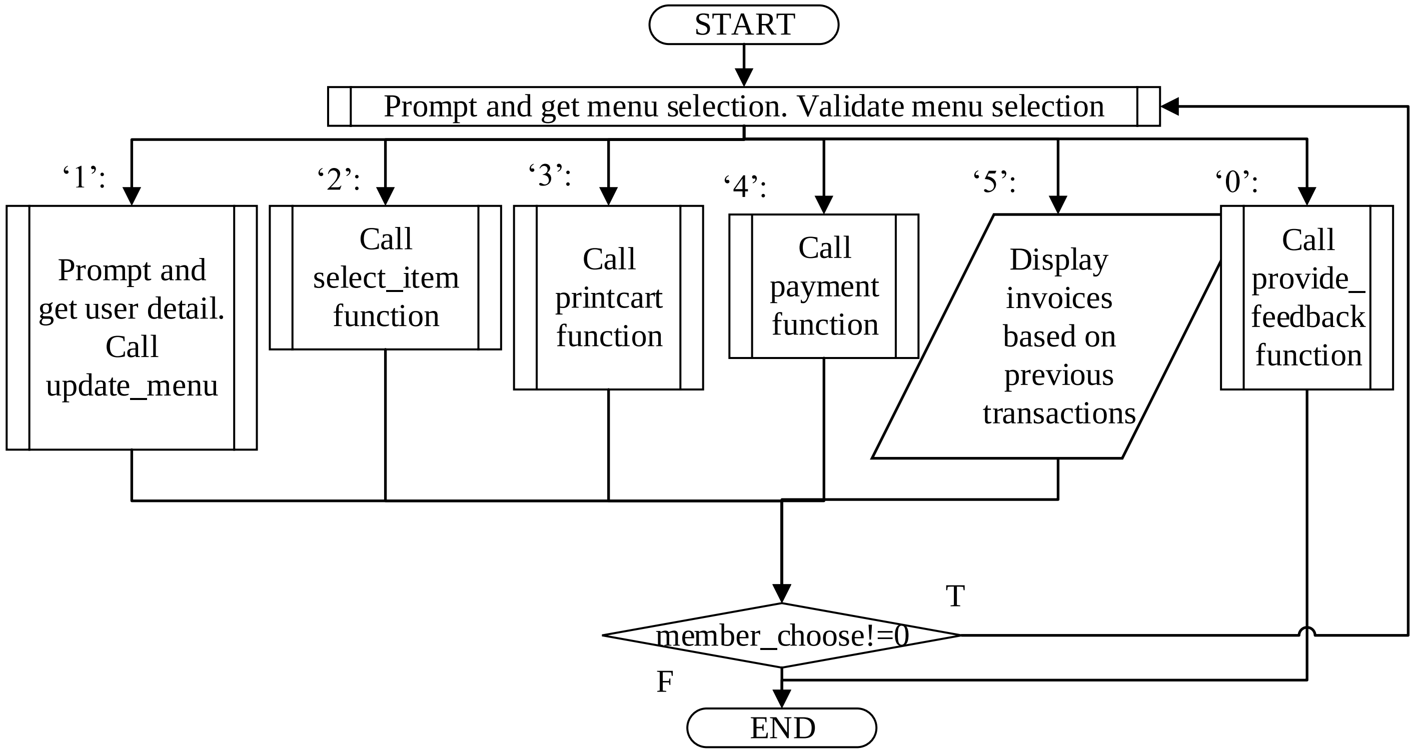
Member	Modules	Roles	Description
Member 2: Jacynth Tham Ming Quan	View Record	Admin	Displays list of products and returns total number of products to assist other self-functions such as update record and delete record
	Delete Record	Admin	Allow admins to delete certain products and all details related to it via view entire product or by search.
	Search Record	Admin	Searches for related products based on user's input and returns total number of products to assist other self-function such as update record, delete record and select item by member (buyer) which they wanted to buy.
	Arrange Searched Records		This function arranges all the search results in ascending order according to the points given by the Grading System. This is to make sure that the most similar product which wanted by the user able to list out at the top.
	Validate input		Ensure that the user's integer input is of the correct data type, length and within a given range
Member 3: Tan Wei Mun	Provide feedback	Member	Users can provide any kind of feedback, such as feedback of the system, feedback of a service and feedback of particular product.
	View feedback	Admin	Admin can view the feedback from the members such as feedback of the system, feedback of a service and feedback of particular product.
	Payment	Member	User can make payment by credit or debit card and cash delivery. An invoice will be generated, and this is saved in the system's database
	Select product, save and update the cart	Member	Members are able to add products into a list. If a product is added multiple times, the quantity to be bought will be totaled.

Member	Modules	Roles	Description
Member 4: Ng Kwan Hou	Member Sign Up	Member	Add new users to the system
	View Data	Member	View user’s data such as view the history of the payment, user’s own profile, items added by the admin (product list) and total earnings for admin.
	Update Data	Member	Update any records related to the user such as password, favorite number email, delivery address and etcetera

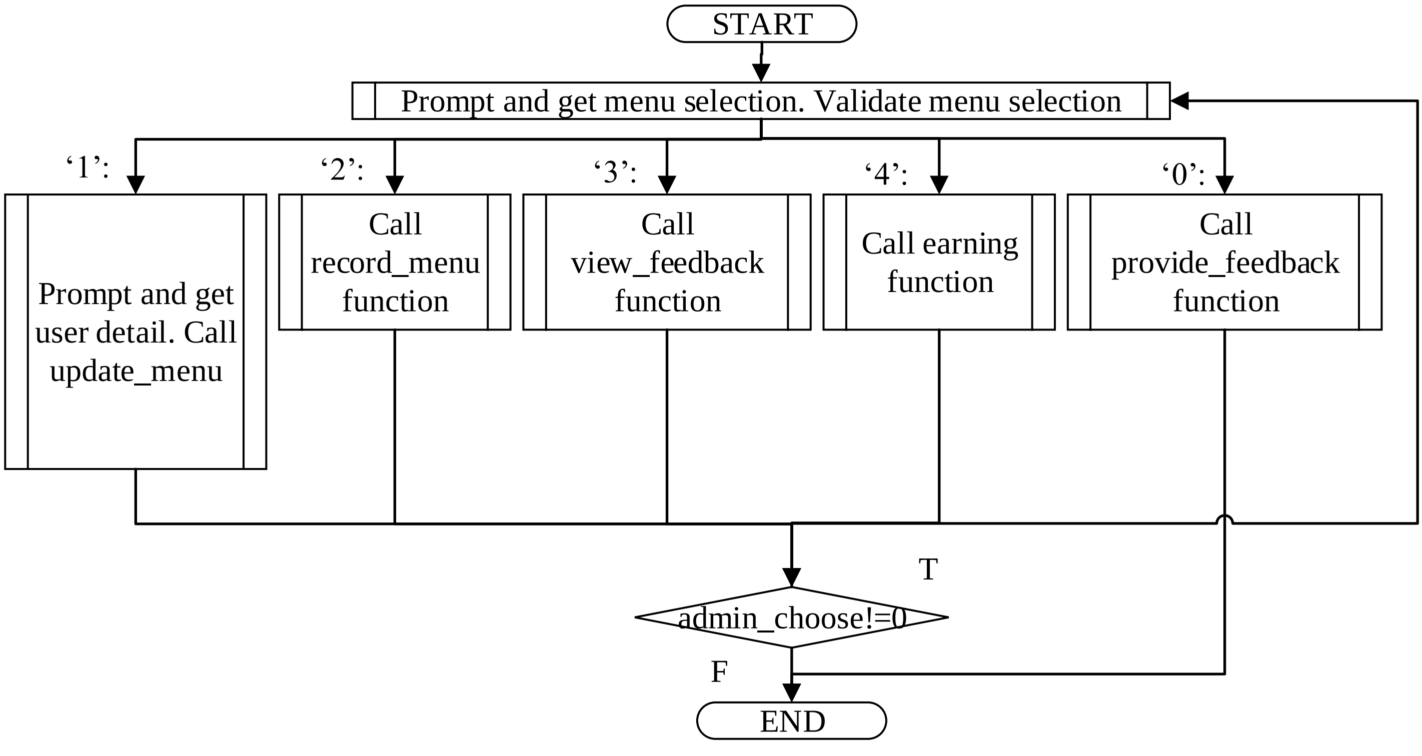
Flowchart

Main menu.cpp

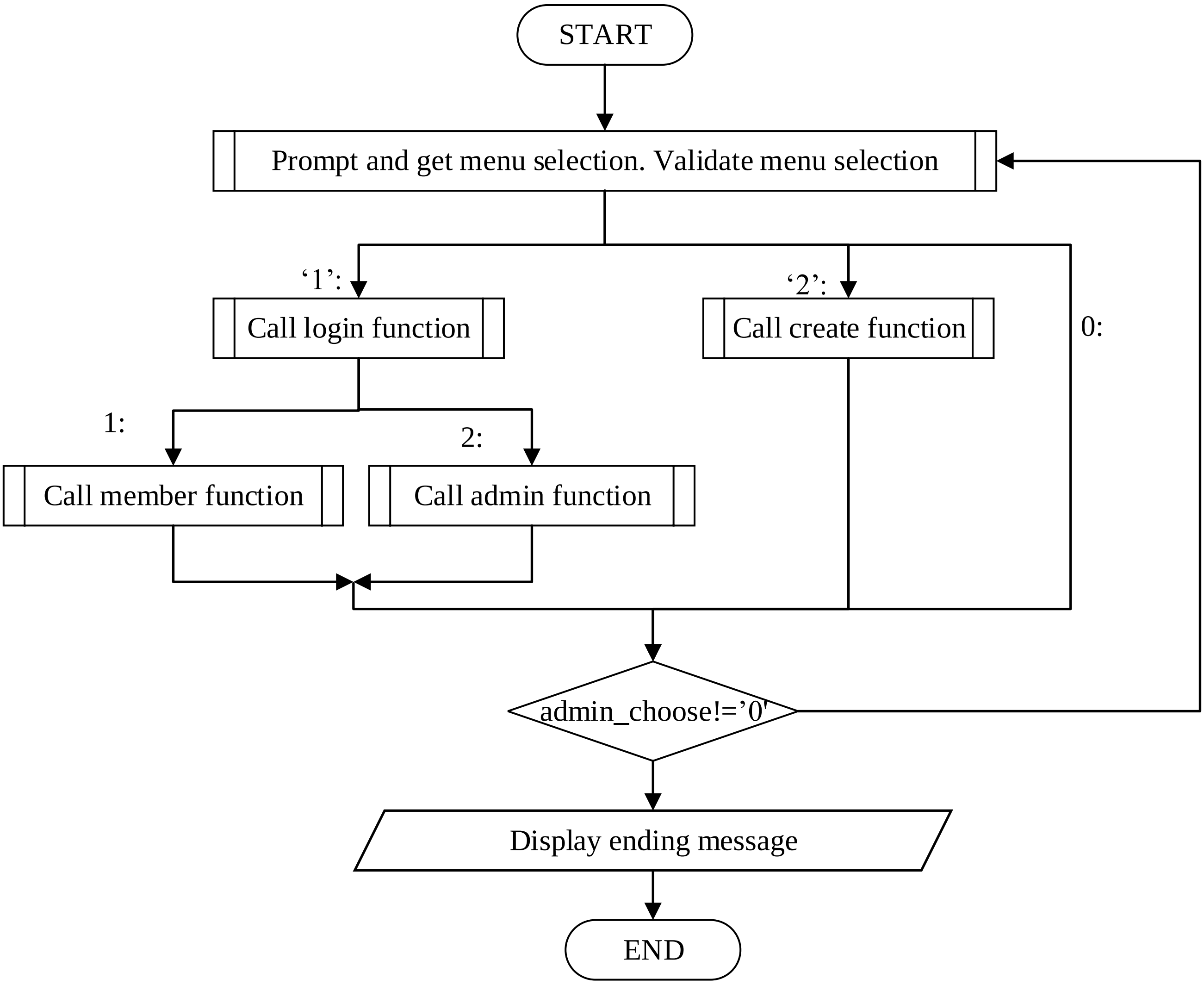
void member()


START
Prompt and get menu selection. Validate menu selection
'1':
'2':
'3':
'4':
'5':
'0':
Prompt and get user detail. Call update_menu
Call select_item function
Call printcart function
Call payment function
Display invoices based on previous transactions
Call provide_ feedback function
T
member_choose!=0
F
END

void admin()

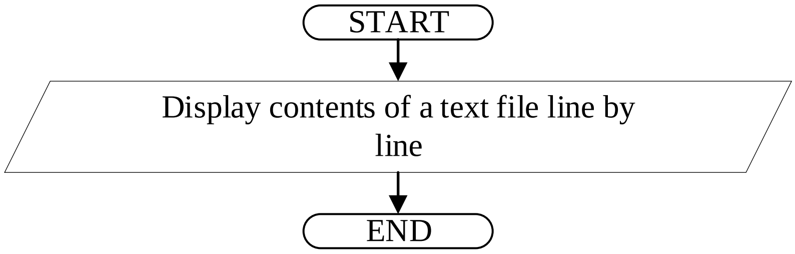

START
Prompt and get menu selection. Validate menu selection
'1':
'2':
'3':
'4':
'0':
Prompt and get user detail. Call update_menu
Call record_menu function
Call view_feedback function
Call earning function
Call provide_feedback function
T
admin_choose!=0
F
END

int main()

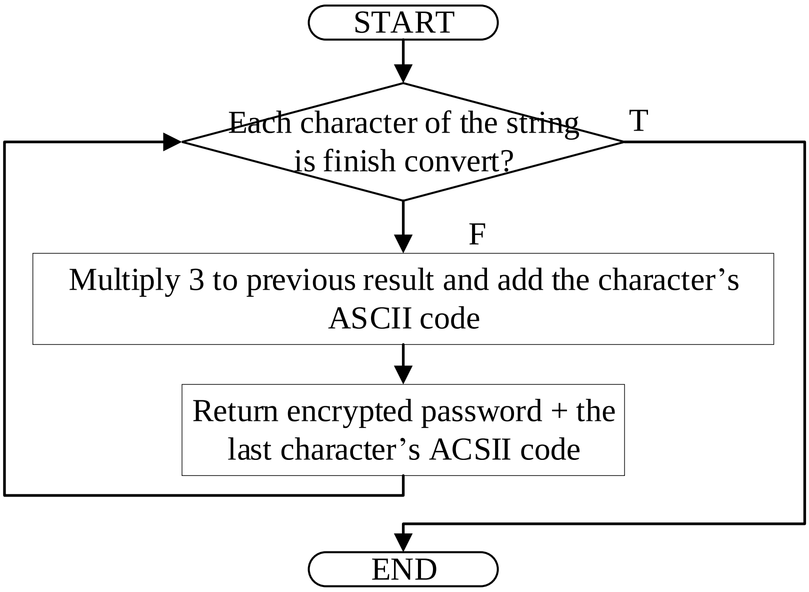

START
Prompt and get menu selection. Validate menu selection
‘1’:
Call login function
‘2’:
Call create function
0:
1:
2:
Call member function
Call admin function
admin_choose!=’0'
Display ending message
END

Feature.h

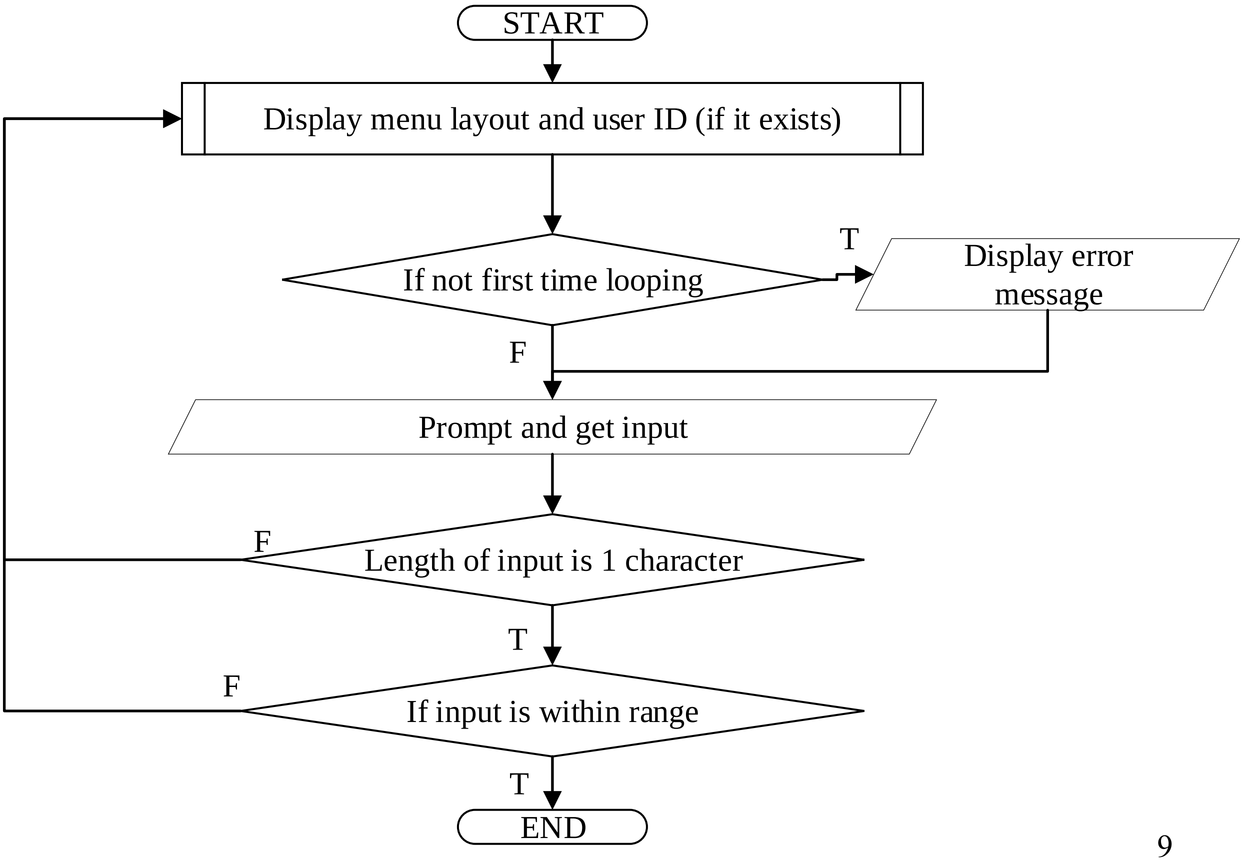
void readfile(string filename)



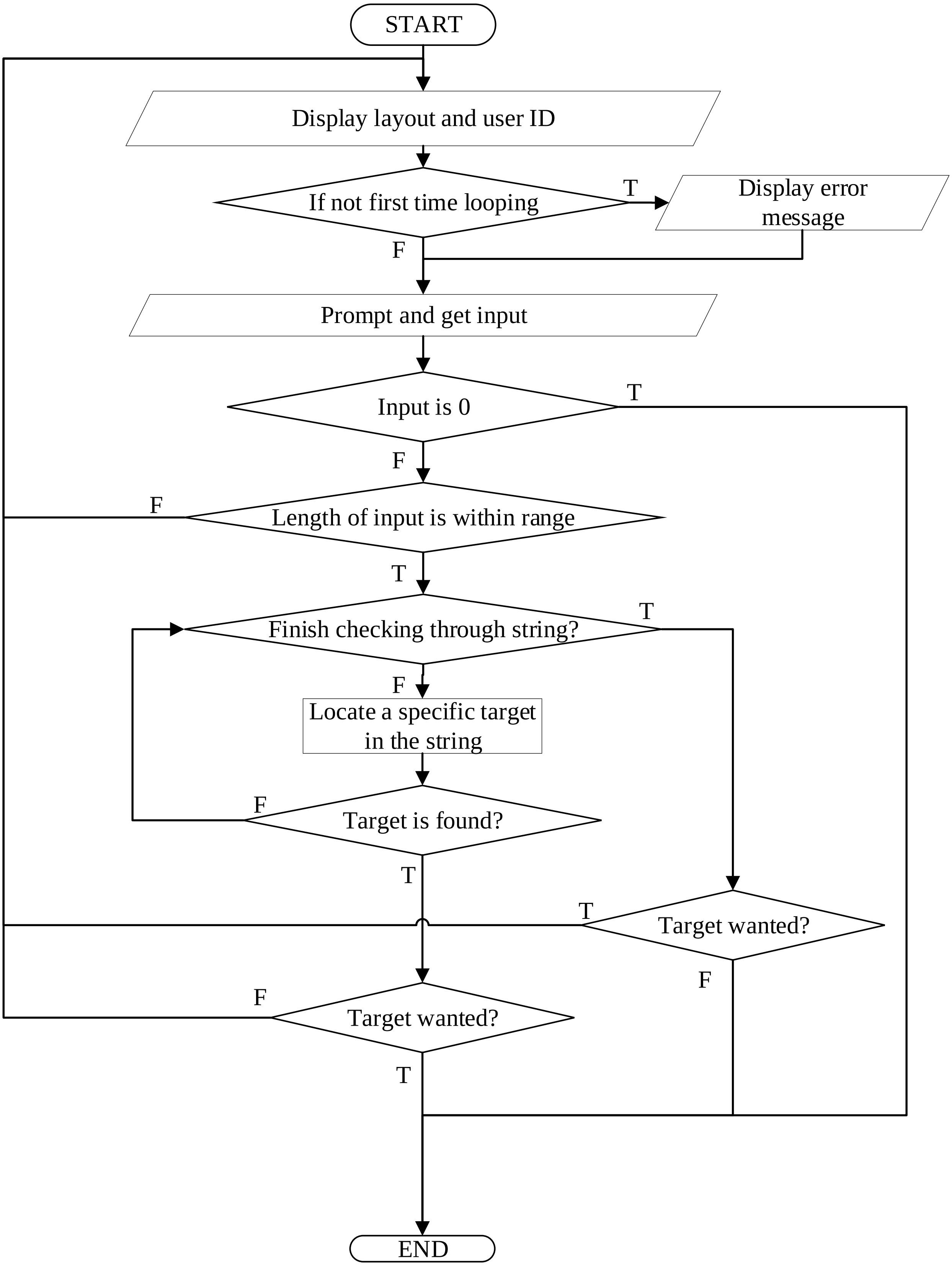
int encrypt_system(string enter)


START
Each character of the string is finish convert?
T
F
Multiply 3 to previous result and add the character’s ASCII code
Return encrypted password + the last character’s ACSII code
END

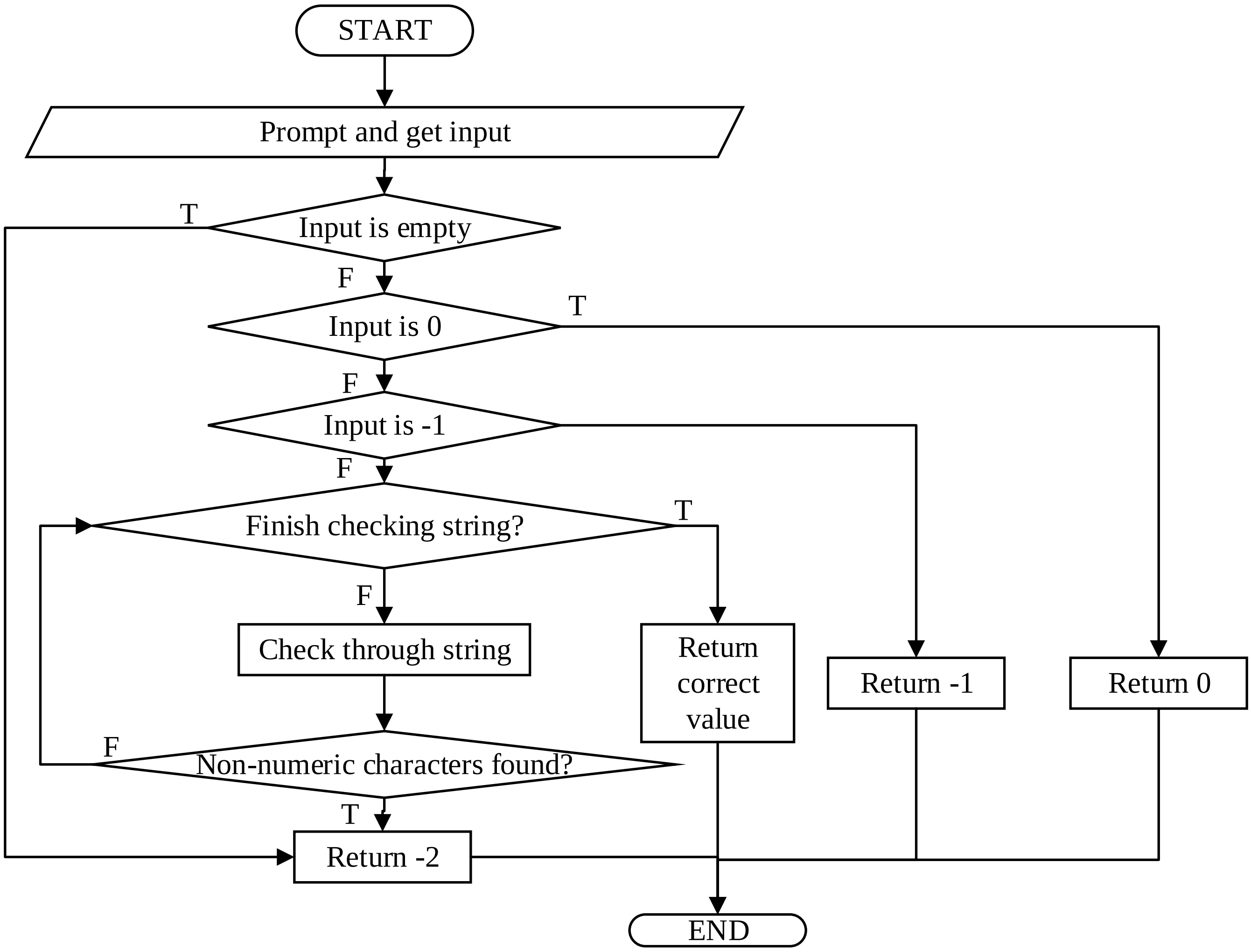
void input_char(string file1, string addi, string file2, string prompt, char& var, int indicator, char accept1, char accept2)


START
Display menu layout and user ID (if it exists)
If not first time looping
T
Display error message
F
Prompt and get input
F
Length of input is 1 character
T
F
If input is within range
T
END

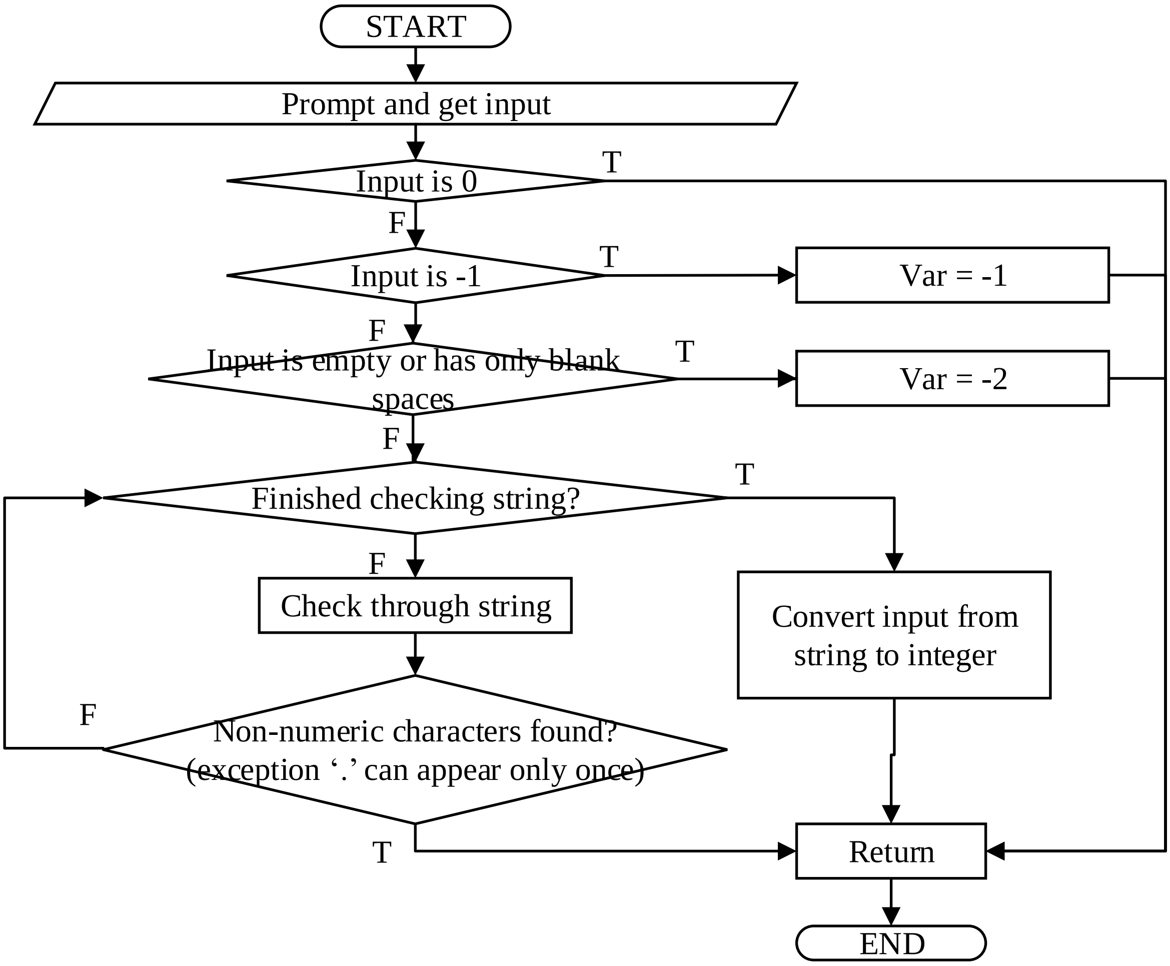
void input_string(string file1, string error, string prompt, string& var, int indicator, char target, int min, int max)


START
Display layout and user ID
If not first time looping
T
Display error message
F
Prompt and get input
Input is 0
T
F
F
Length of input is within range
T
Finish checking through string?
T
F
Locate a specific target in the string
F
Target is found?
T
T
Target wanted?
F
F
Target wanted?
T
END

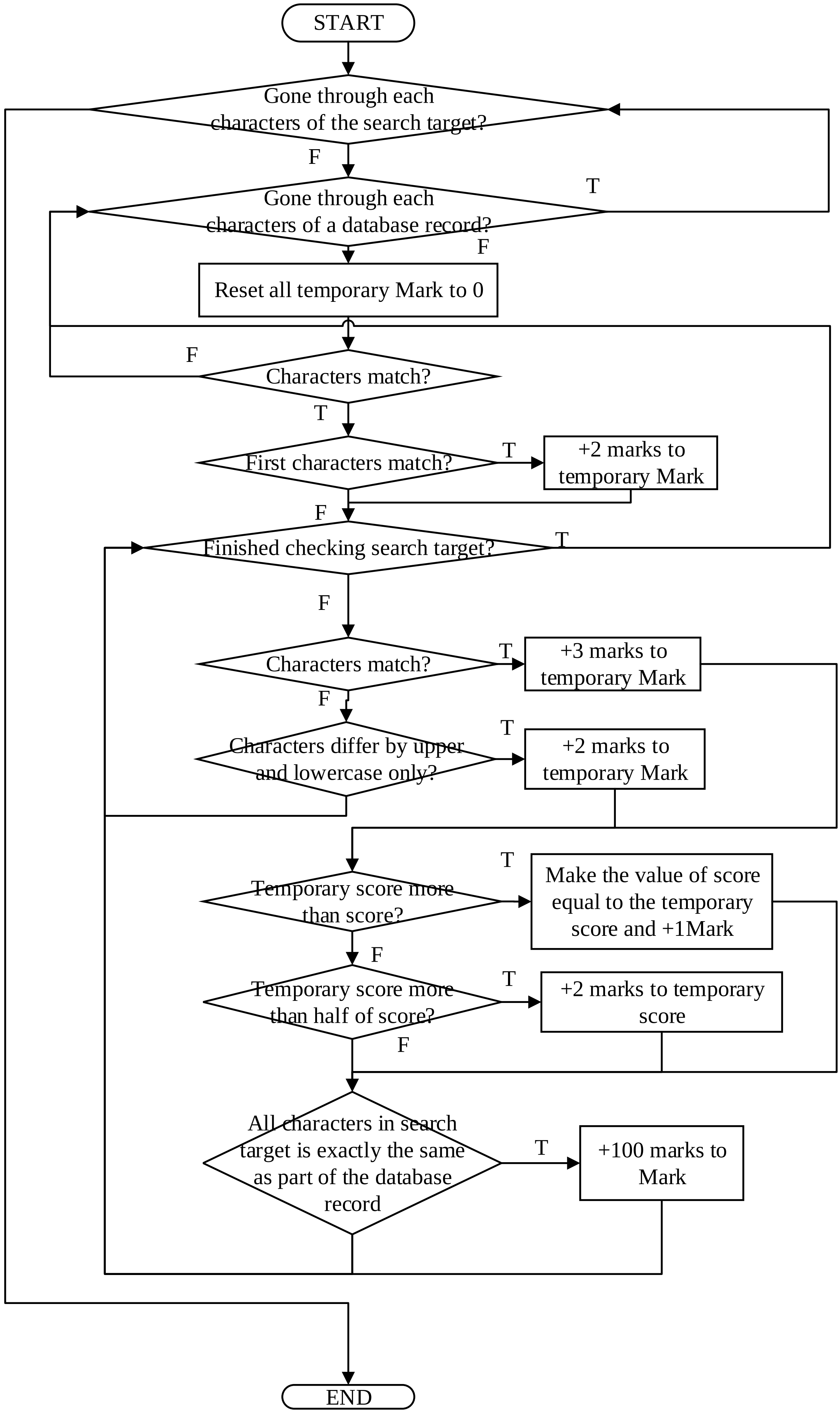
int input_integer(string show, int min, int max)



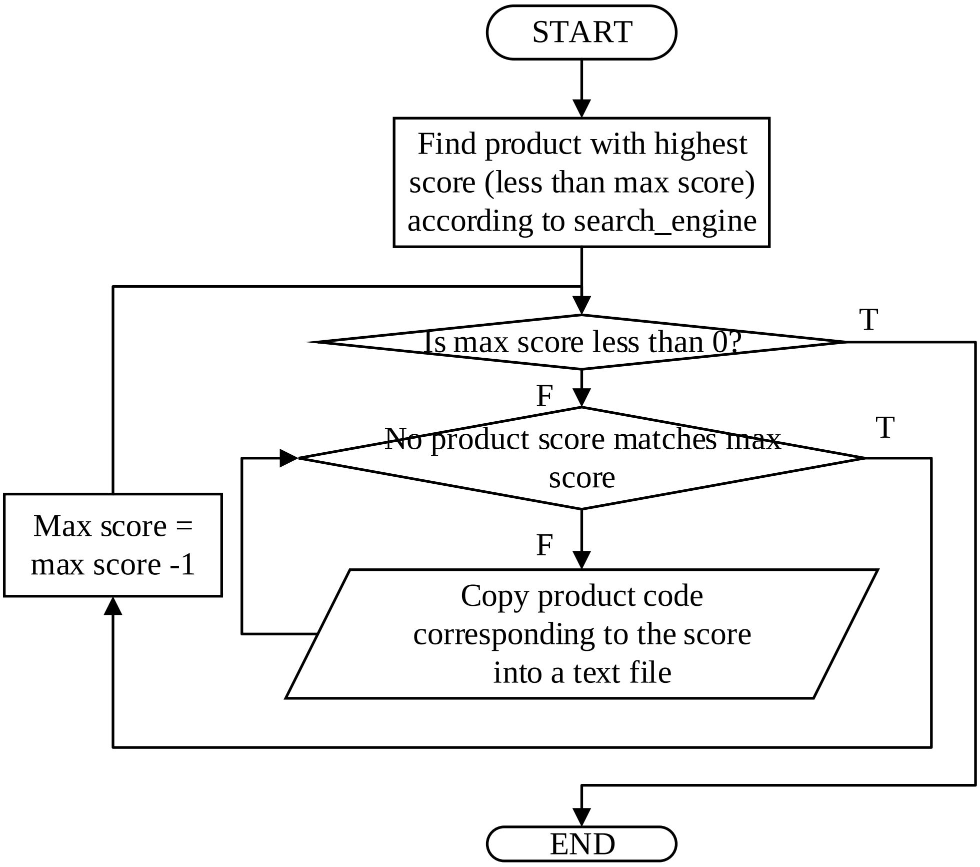
void input_float(string show, float& var, float min, float max)


START
Prompt and get input
Input is 0
T
F
Input is -1
T
Var = -1
F
Input is empty or has only blank spaces
T
Var = -2
F
Finished checking string?
T
F
Check through string
Convert input from string to integer
F
Non-numeric characters found? (exception '.' can appear only once)
T
Return
END

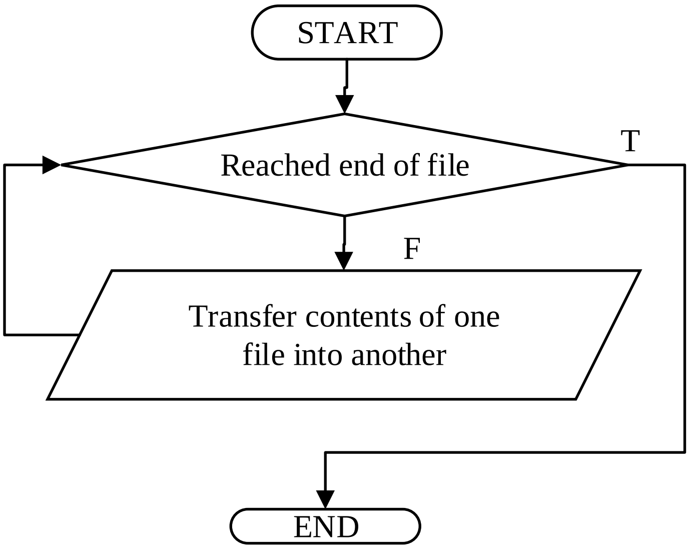
void grading_system(string target, int code, string source)


START
Gone through each characters of the search target?
F
Gone through each characters of a database record?
T
F
Reset all temporary Mark to 0
F
Characters match?
T
First characters match?
T
+2 marks to temporary Mark
F
Finished checking search target?
T
F
Characters match?
T
+3 marks to temporary Mark
F
Characters differ by upper and lowercase only?
T
+2 marks to temporary Mark
Temporary score more than score?
T
Make the value of score equal to the temporary score and +1Mark
F
Temporary score more than half of score?
T
+2 marks to temporary score
F
All characters in search target is exactly the same as part of the database record
T
+100 marks to Mark
END

void arrange_search()

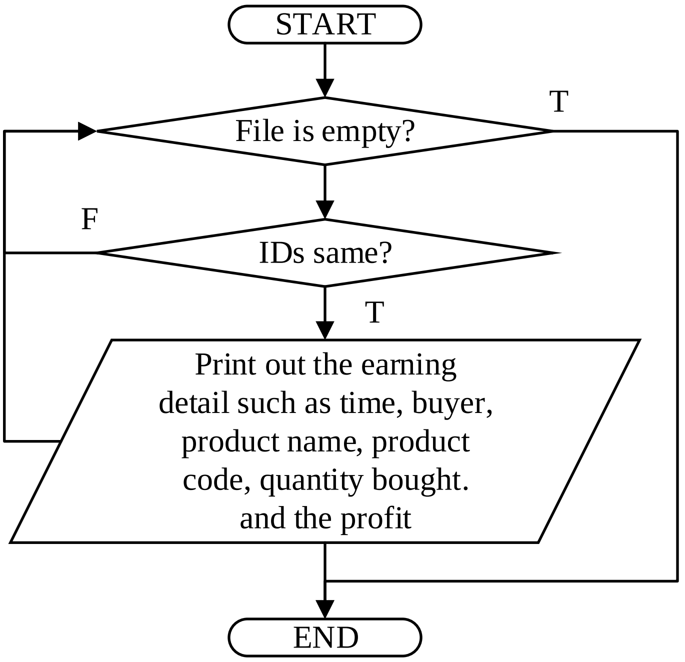

START
Find product with highest score (less than max score) according to search_engine
Is max score less than 0?
T
F
No product score matches max score
T
F
Copy product code corresponding to the score into a text file
Max score = max score -1
END

void updatefile(string file1, string file2)

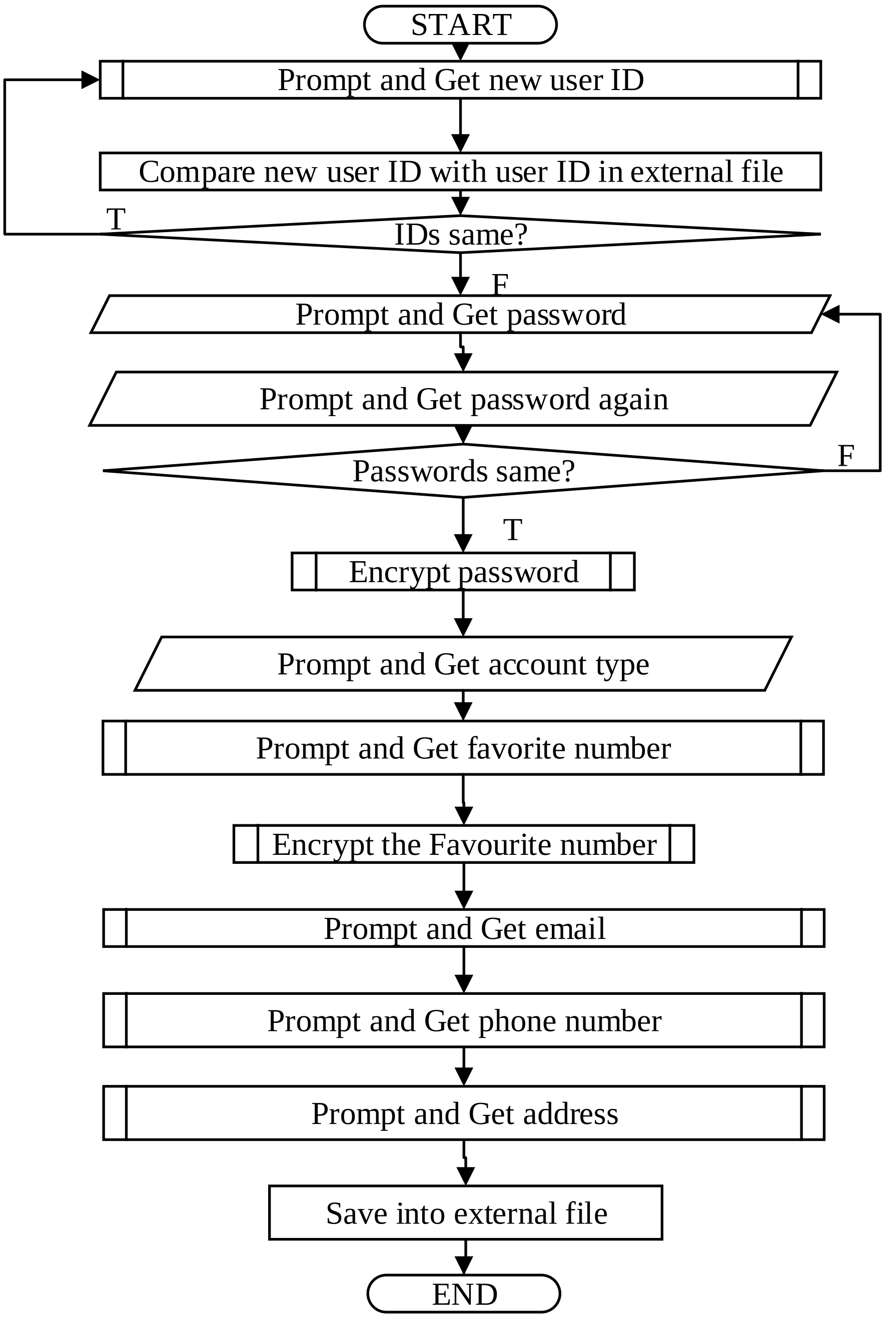


Account.h

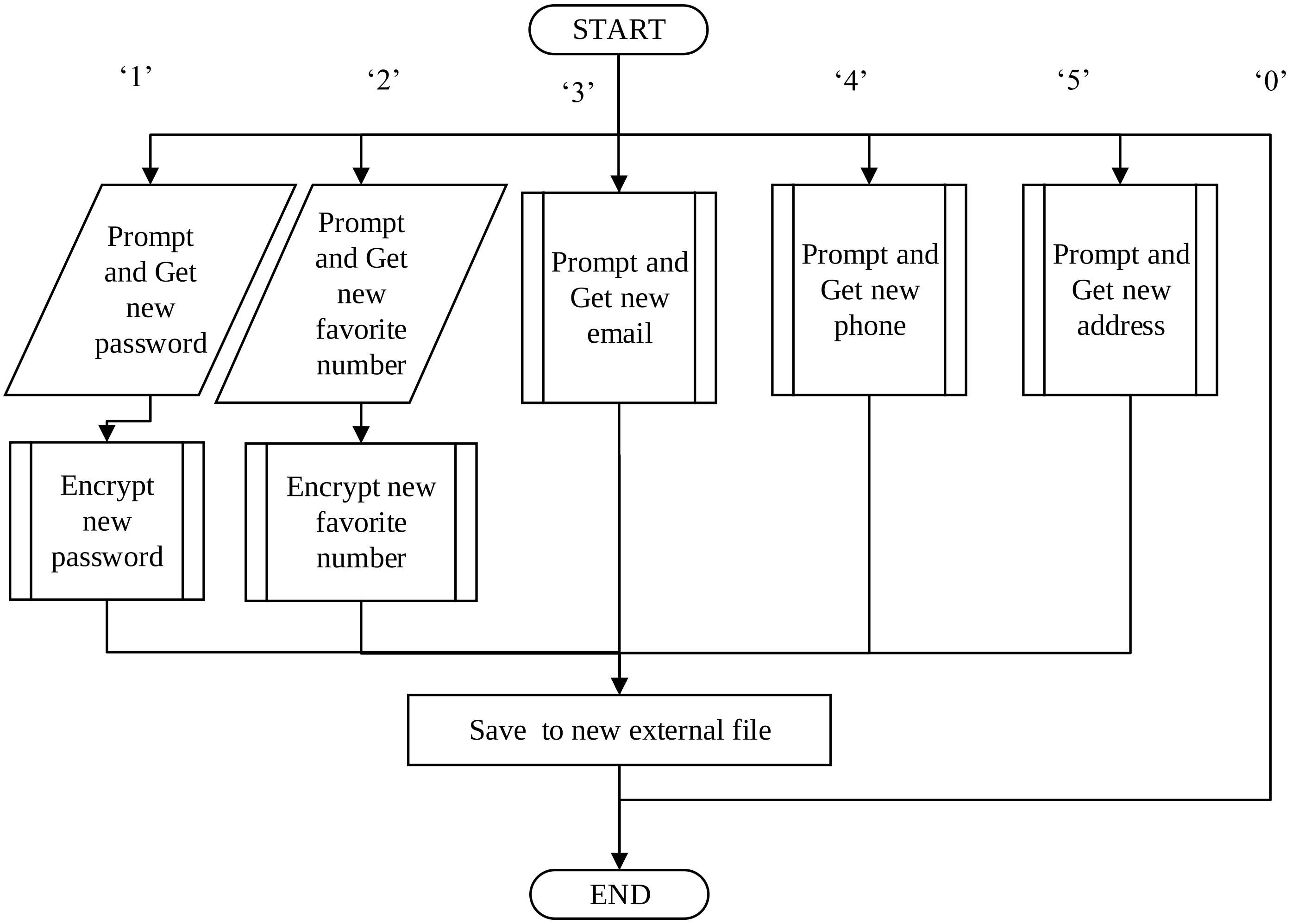
void earning()


START
File is empty?
T
F
IDs same?
T
Print out the earning detail such as time, buyer, product name, product code, quantity bought. and the profit
END

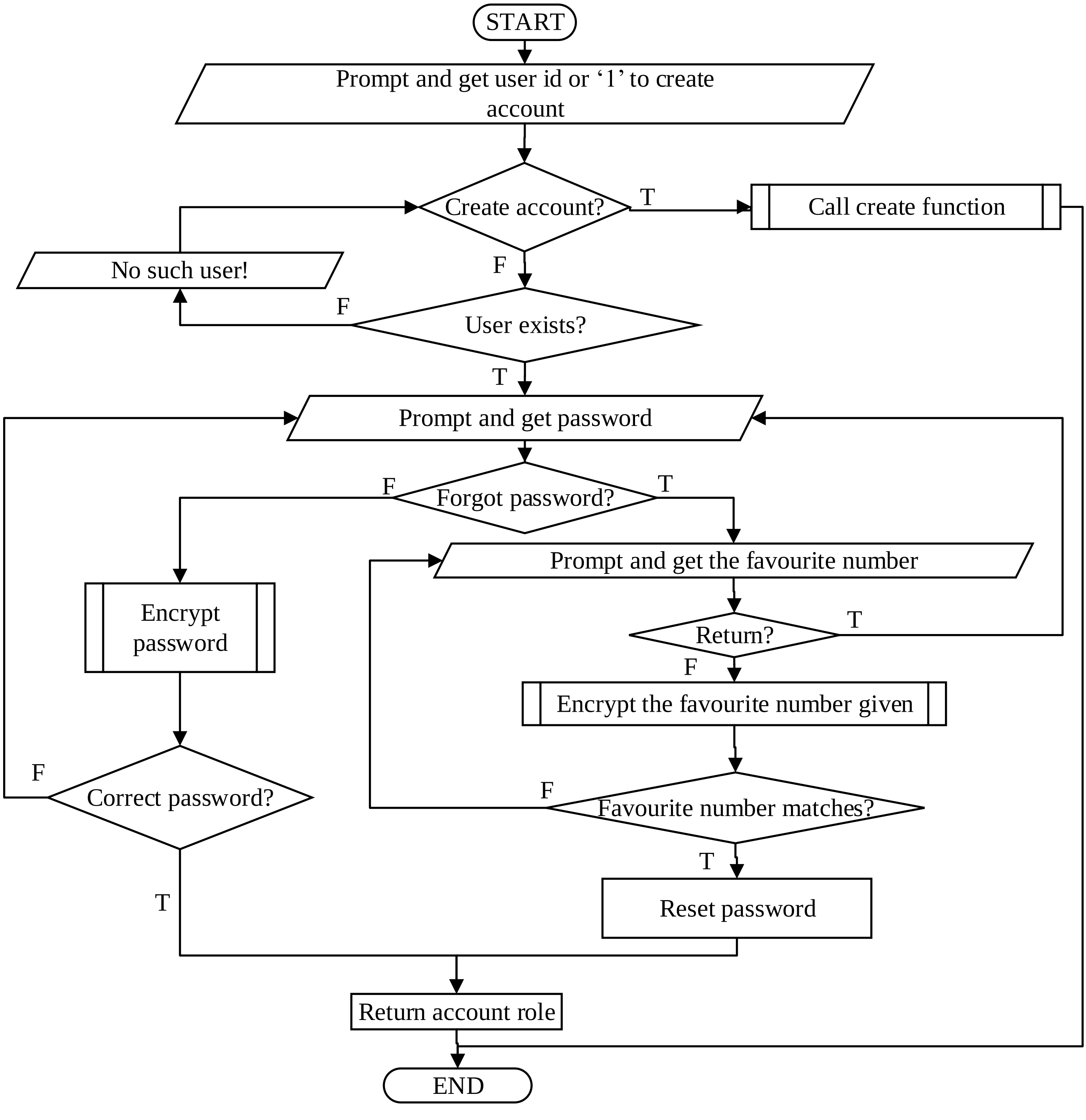
void create()


START
Prompt and Get new user ID
Compare new user ID with user ID in external file
T
IDs same?
F
Prompt and Get password
Prompt and Get password again
Passwords same?
F
T
Encrypt password
Prompt and Get account type
Prompt and Get favorite number
Encrypt the Favourite number
Prompt and Get email
Prompt and Get phone number
Prompt and Get address
Save into external file
END

int update_menu(char selection)

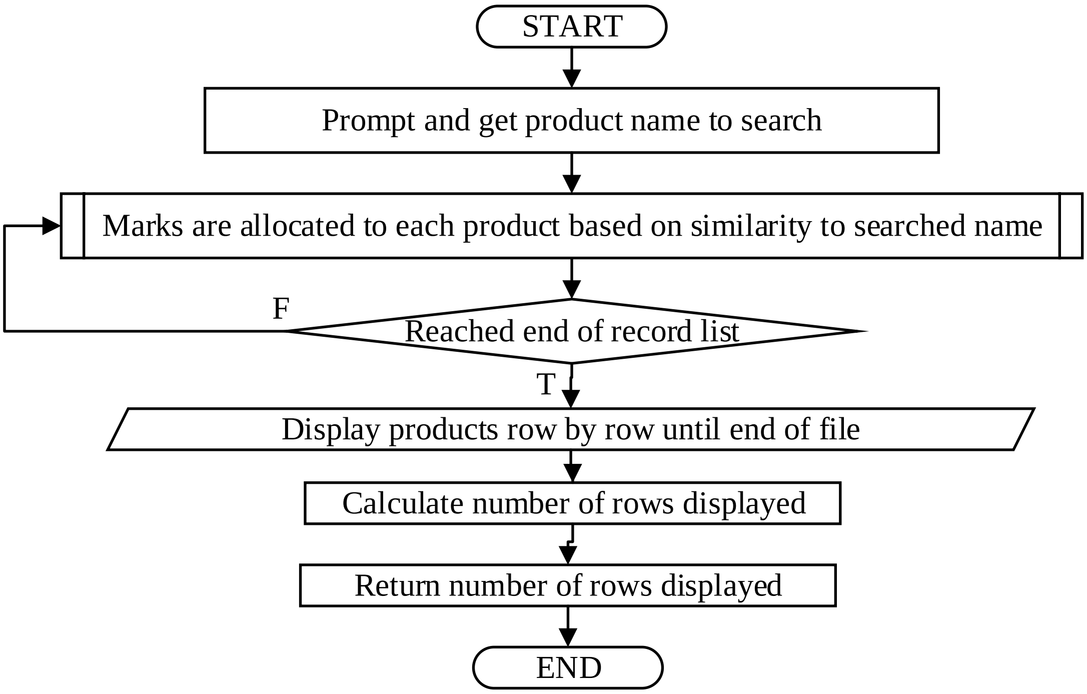

START
'1'
'2'
'3'
'4'
'5'
'0'
Prompt and Get new password
Prompt and Get new favorite number
Prompt and Get new email
Prompt and Get new phone
Prompt and Get new address
Encrypt new password
Encrypt new favorite number
Save to new external file
END

int login()

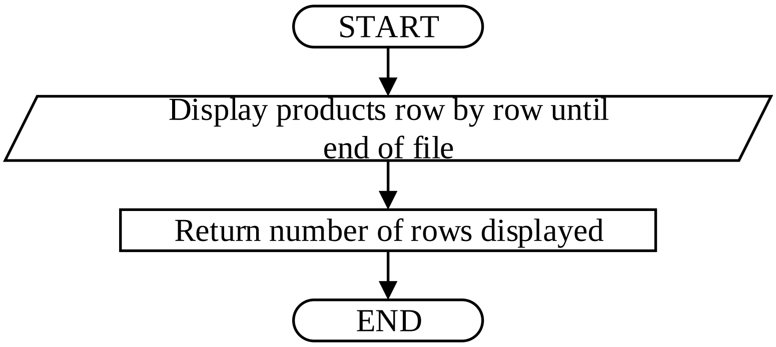

START
Prompt and get user id or '1' to create account
Create account?
T
Call create function
F
No such user!
F
User exists?
T
Prompt and get password
F
Forgot password?
T
Prompt and get the favourite number
Encrypt password
Return?
T
F
Encrypt the favourite number given
F
Correct password?
F
Favourite number matches?
T
T
Reset password
Return account role
END

Record.h

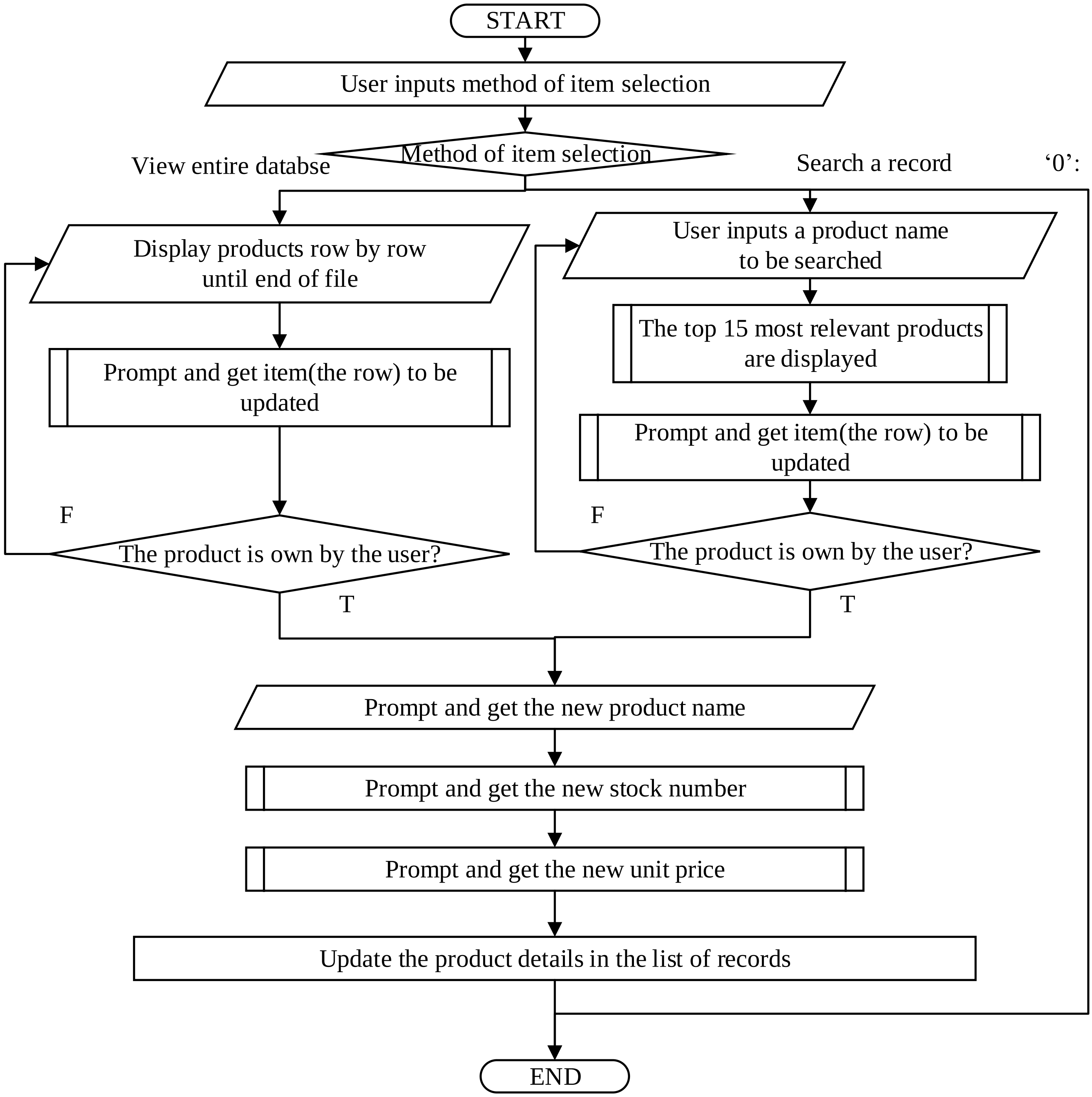
int search_record(string name_search)


START
Prompt and get product name to search
Marks are allocated to each product based on similarity to searched name
F
Reached end of record list
T
Display products row by row until end of file
Calculate number of rows displayed
Return number of rows displayed
END

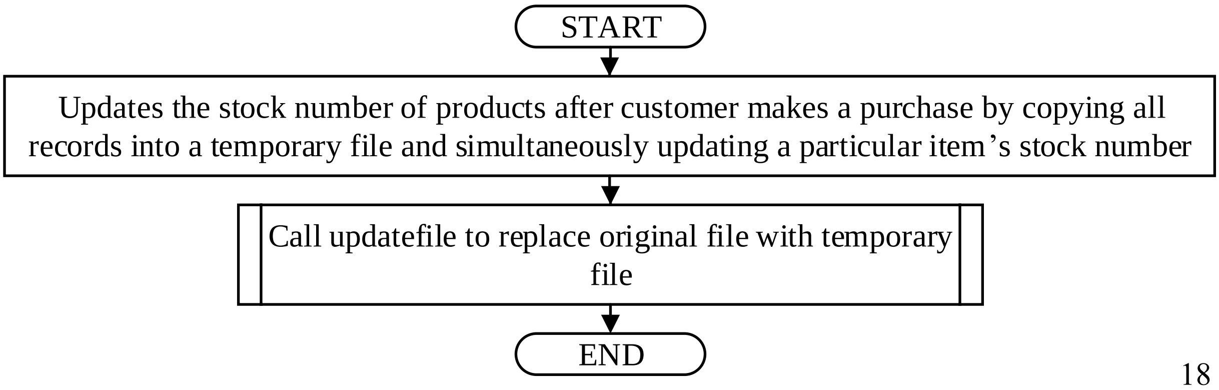
int view_record()


START
Display products row by row until end of file
Return number of rows displayed
END

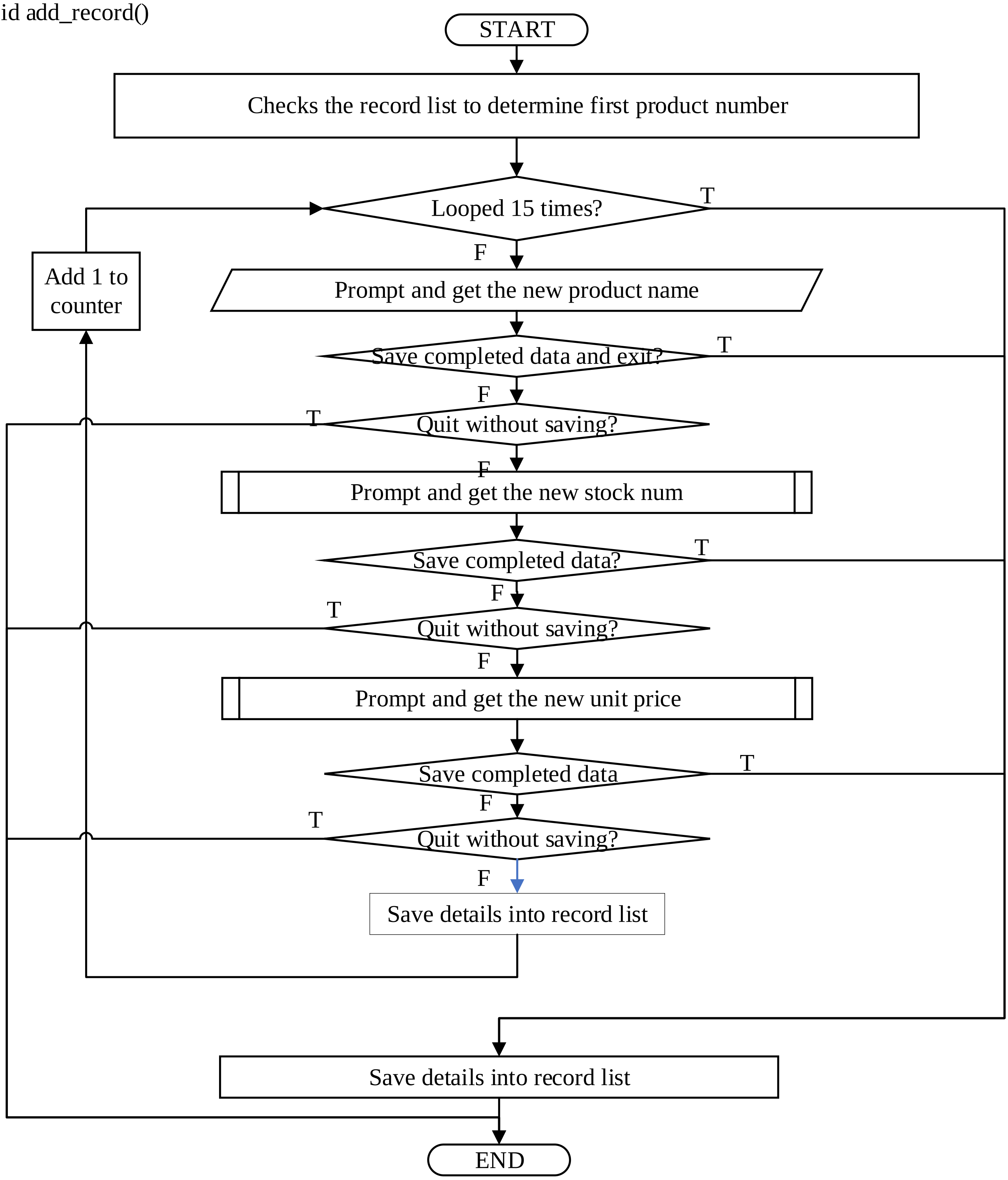
void update_record()


START
User inputs method of item selection
Method of item selection
View entire databse
Search a record
'0':
Display products row by row until end of file
Prompt and get item(the row) to be updated
F
The product is own by the user?
T
User inputs a product name to be searched
The top 15 most relevant products are displayed
Prompt and get item(the row) to be updated
F
The product is own by the user?
T
Prompt and get the new product name
Prompt and get the new stock number
Prompt and get the new unit price
Update the product details in the list of records
END

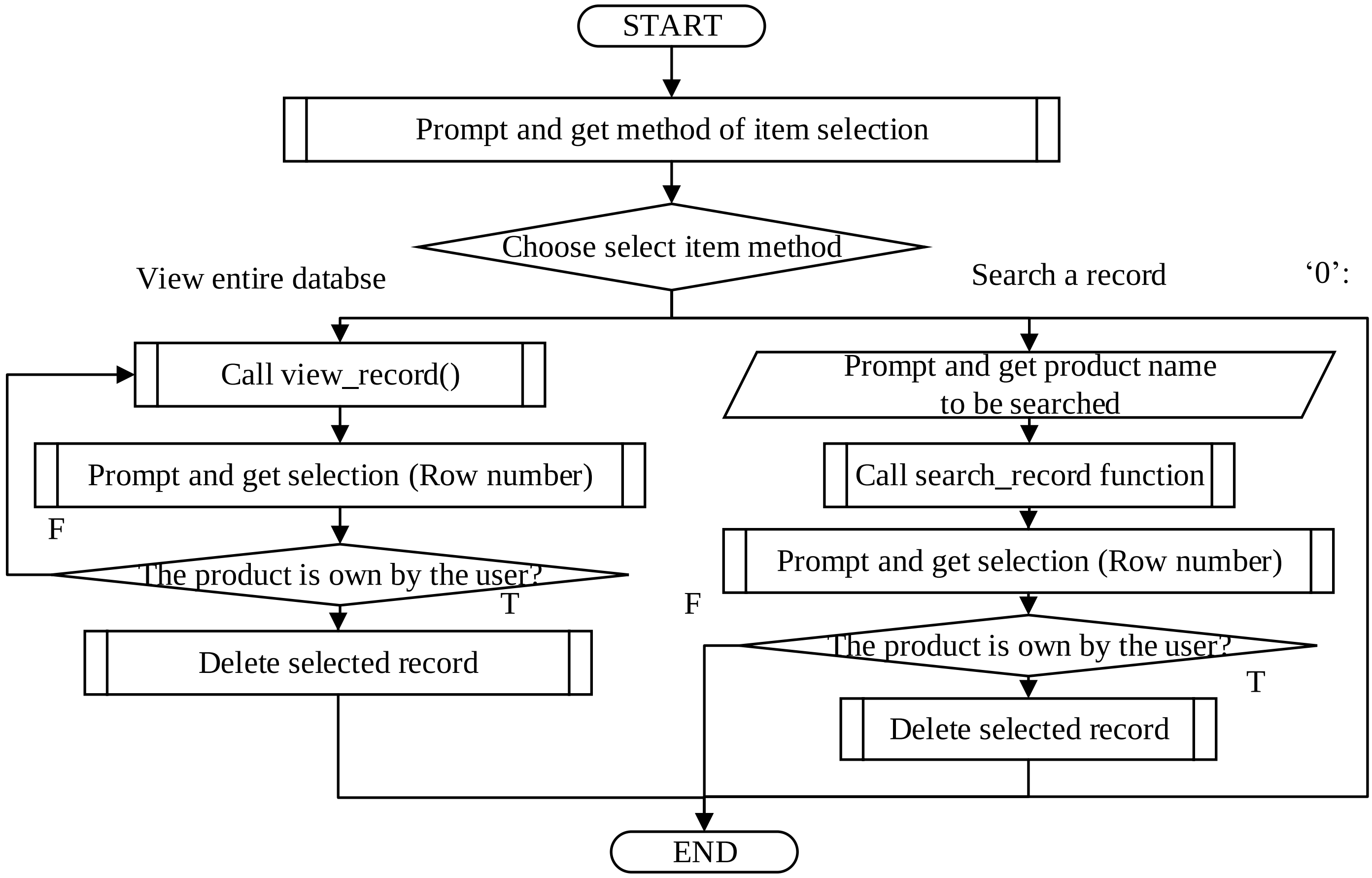
void update_buy(int code, int quantity)


START
Updates the stock number of products after customer makes a purchase by copying all records into a temporary file and simultaneously updating a particular item's stock number
Call updatefile to replace original file with temporary file
END

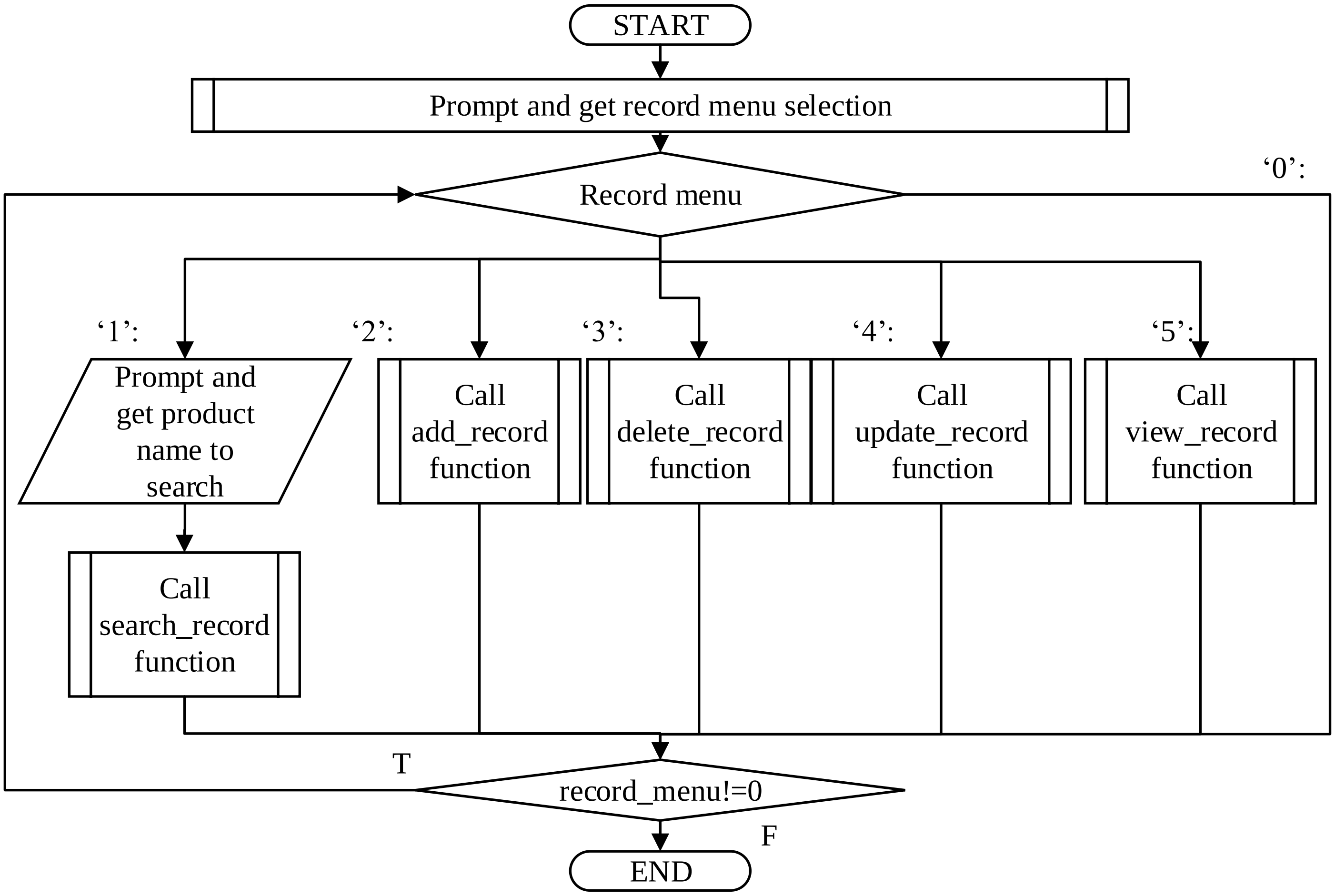
void add_record()


START
Checks the record list to determine first product number
Looped 15 times?
T
F
Add 1 to counter
Prompt and get the new product name
Save completed data and exit?
T
F
Quit without saving?
T
F
Prompt and get the new stock num
Save completed data?
T
F
Quit without saving?
T
F
Prompt and get the new unit price
Save completed data
T
F
Quit without saving?
T
F
Save details into record list
Save details into record list
END

void delete_record()

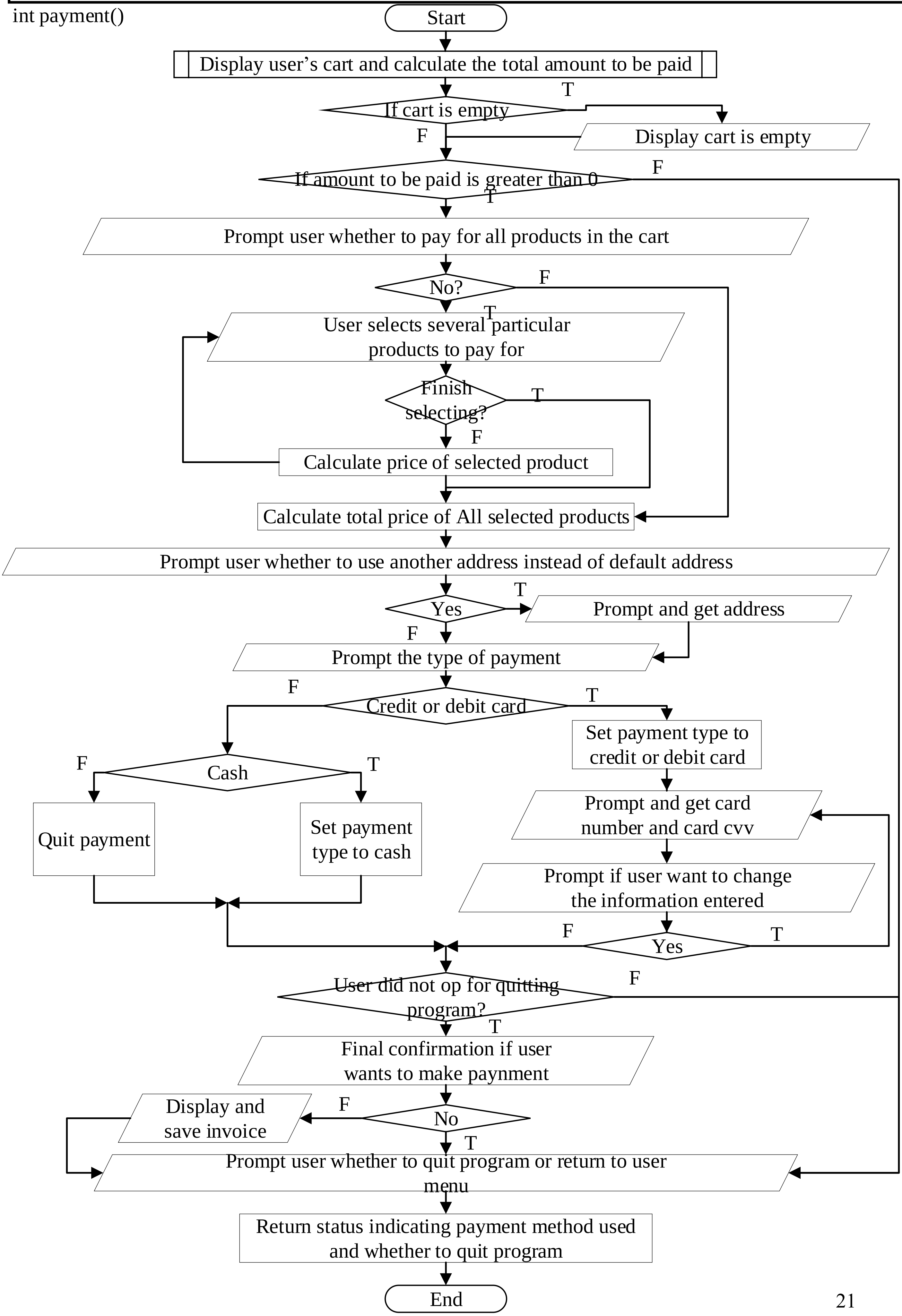

START
Prompt and get method of item selection
Choose select item method
View entire databse
Search a record
‘0’:
Call view_record()
Prompt and get selection (Row number)
F
The product is own by the user?
T
Delete selected record
Prompt and get product name to be searched
Call search_record function
Prompt and get selection (Row number)
F
The product is own by the user?
T
Delete selected record
END

void record_menu()

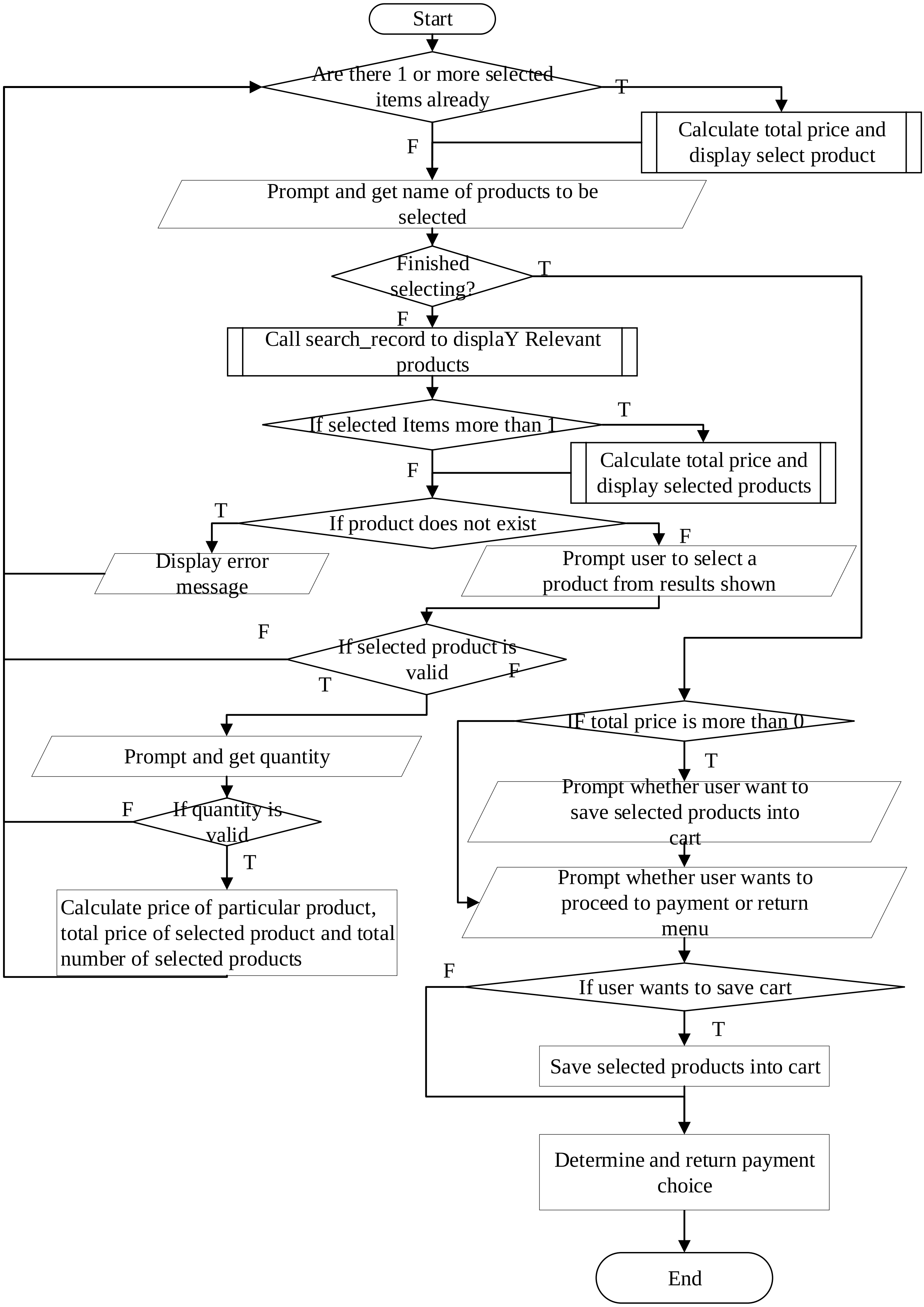

START
Prompt and get record menu selection
Record menu
‘0’:
‘1’:
Prompt and get product name to search
Call search_record function
‘2’:
Call add_record function
‘3’:
Call delete_record function
‘4’:
Call update_record function
‘5’:
Call view_record function
T
record_menu!=0
F
END

Payment.h

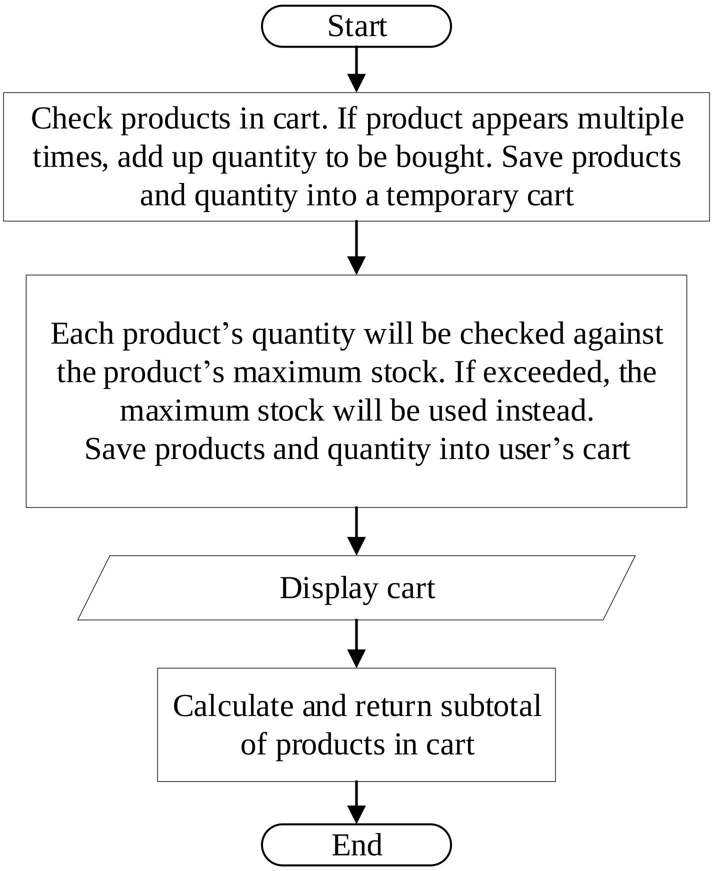
int payment()


Start
Display user’s cart and calculate the total amount to be paid
If cart is empty
T
Display cart is empty
F
If amount to be paid is greater than 0
F
T
Prompt user whether to pay for all products in the cart
No?
F
T
User selects several particular products to pay for
Finish selecting?
T
F
Calculate price of selected product
Calculate total price of All selected products
Prompt user whether to use another address instead of default address
Yes
T
Prompt and get address
F
Prompt the type of payment
Credit or debit card
F
T
Set payment type to credit or debit card
Cash
F
T
Quit payment
Set payment type to cash
Prompt and get card number and card cvv
Prompt if user want to change the information entered
Yes
F
T
User did not op for quitting program?
F
T
Final confirmation if user wants to make paynment
No
F
Display and save invoice
T
Prompt user whether to quit program or return to user menu
Return status indicating payment method used and whether to quit program
End

int select_item()

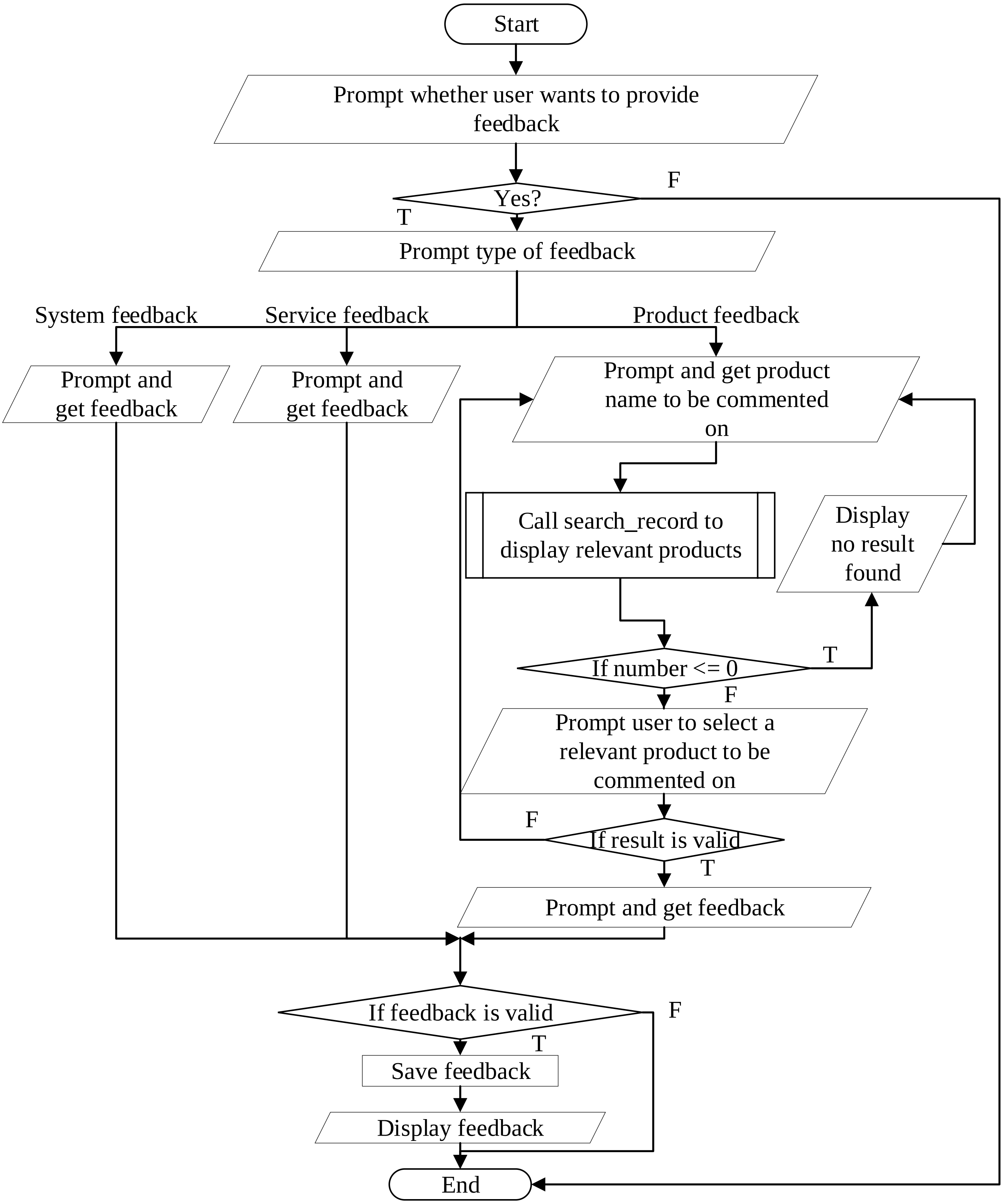

Start
Are there 1 or more selected items already
T
Calculate total price and display select product
F
Prompt and get name of products to be selected
Finished selecting?
T
F
Call search_record to displaY Relevant products
If selected Items more than 1
T
Calculate total price and display selected products
F
If product does not exist
T
F
Display error message
Prompt user to select a product from results shown
F
If selected product is valid
F
T
Prompt and get quantity
F
If quantity is valid
T
Calculate price of particular product, total price of selected product and total number of selected products
IF total price is more than 0
T
Prompt whether user want to save selected products into cart
Prompt whether user wants to proceed to payment or return menu
F
If user wants to save cart
T
Save selected products into cart
Determine and return payment choice
End

double printcart(string file,int &total_item)

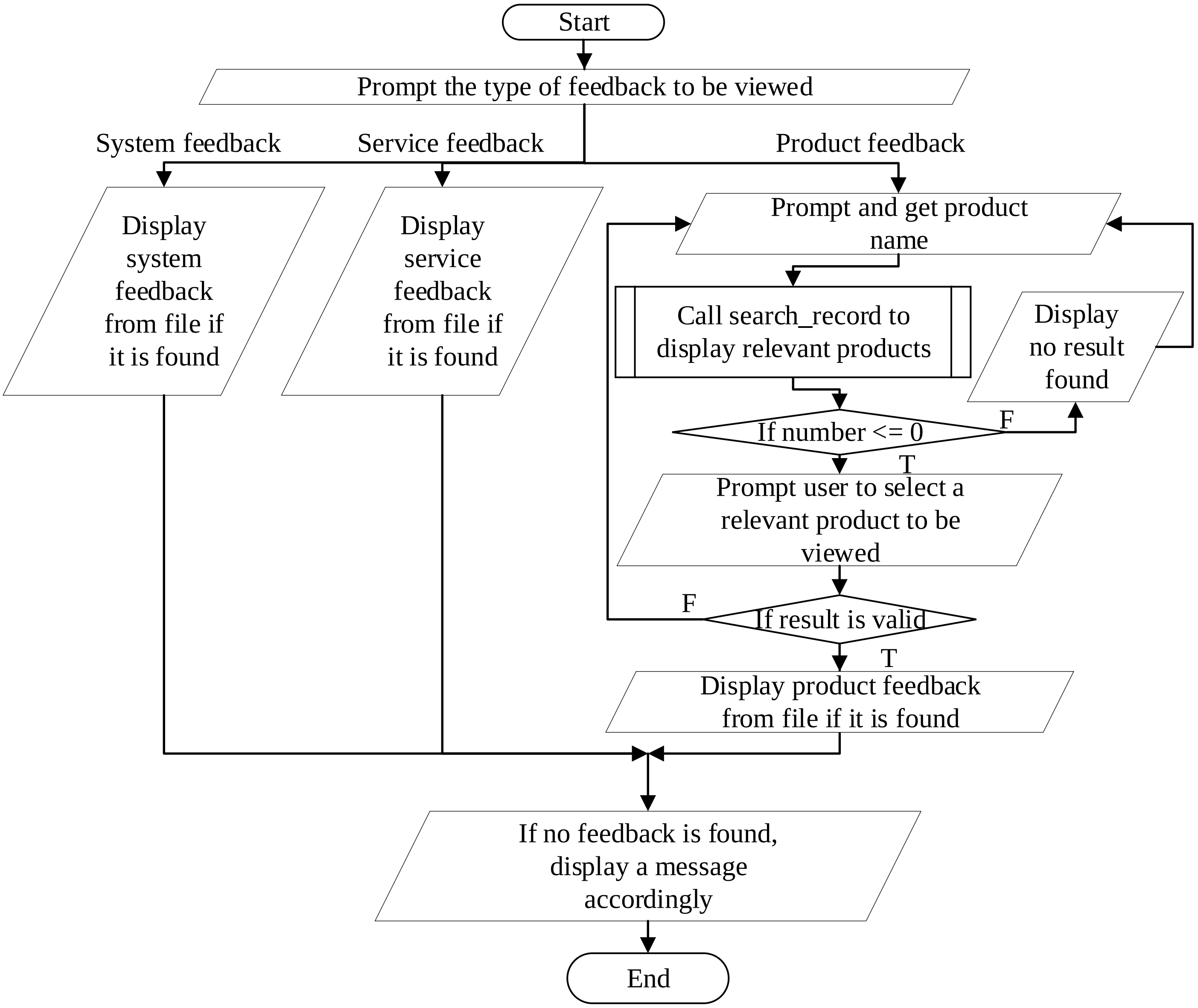

Start
Check products in cart. If product appears multiple times, add up quantity to be bought. Save products and quantity into a temporary cart
Each product's quantity will be checked against the product's maximum stock. If exceeded, the maximum stock will be used instead.
Save products and quantity into user's cart
Display cart
Calculate and return subtotal of products in cart
End

Feedback.h

int provide_feedback()



int view_feedback()


Start
Prompt the type of feedback to be viewed
System feedback
Service feedback
Product feedback
Display system feedback from file if it is found
Display service feedback from file if it is found
Prompt and get product name
Call search_record to display relevant products
Display no result found
If number <= 0
F
T
Prompt user to select a relevant product to be viewed
F
If result is valid
T
Display product feedback from file if it is found
If no feedback is found, display a message accordingly
End

Pseudocode Code

Main menu.cpp

void member ( )

```
Prompt and get member menu selection
Validate member menu selection
    Menu selection is '1' – Prompt and get user detail to be updated
                            Call function to update user details
    Menu selection is '2' – Call function to browse products
    Menu selection is '3' – Call function to view shopping cart
    Menu selection is '4' – Call function to pay for items in cart
    Menu selection is '5' – Displays invoices based on previous transactions made
    Menu selection is '0' – Call function to allow user to provide feedback then exit
```

void admin ( )

```
Prompt and get admin menu selection
Validate admin menu selection
    Menu selection is '1' – Prompt and get user detail to be updated
                            Call function to update user details
    Menu selection is '2' – Call function to manage records
    Menu selection is '3' – Call function to view feedback
    Menu selection is '4' – Call function to summarise total profit earned
    Menu selection is '0' – Call function to allow user to provide feedback then exit
```

int main ( )

```
Prompt and get menu selection
Validate menu selection
    Menu selection is '1' – Login into member or admin account
    Menu selection is '2' – Create new account
    Menu selection is '0' – Display ending message and quit program
```

Feature.h

void readfile (string filename)

Display contents of a text file line by line

int encrypt_system (string enter)

```
Multiply the previous result by 3.
Converts each character into ASCII code and add the  previous result.
Loop until each character of the string is finish.
Return encrypted password.
```

void input_char (string file1, string addi, string file2, string prompt, char& var, int indicator, char accept1, char accept2)

```
Display menu layout and user ID (if it exists)
First time looping?
    Yes – Prompt and get input
          Validate input and make sure it is 1 character long
          Is input within range?
              Yes – Return to caller
              No – Prompt and get input again
    No – Display error message
         Prompt and get input again
```

void input_string (string file1, string error, string prompt, string& var, int indicator, char target, int min, int max)

```
Display layout and user ID
First time looping?
    Yes – Prompt and get input
          Input is 0?
            Yes – Return to caller
            No – Is length of input within range?
                     Yes – Locate specific target in string until target is found
                           If target is found, check is target is wanted.
                               Yes – Return to caller
                               No – Repeat function again
                     No – Repeat function again

    No – Display error message
         Prompt and get input again
```

int input_integer (string show, int min, int max)

```
Prompt and get input
Is input 0?
    Yes – Return to caller
    No – Is input -1?
        Yes – Return var with a value of -1 to caller
        No – Is input empty or has only blanks?
            Yes – Return var with a value of -2 to caller
            No – Finish checking string?
                Yes – Convert input from string to integer and return -2 to caller
                No – Check through string
                    If non-numeric characters are found, return correct input
                    Else check if finished checking string
```

void input_float (string show, float& var, float min, float max)

```
Prompt and get input
Is input 0?
    Yes – Return to caller
    No – Is input -1?
        Yes – Return the variable with a value of -1 to caller
        No – Is input empty or has only blanks?
            Yes – Return variable with a value of -2 to caller
            No – Finish checking string?
                Yes – Convert input from string to integer and return
                No – Check through string
                    If non-numeric characters are found (except '.', which can
                    appear only once), return to caller
                    Else check if finished checking string
```

void grading_system (string target, int code, string source)

```
Gone through all characters in the search target?
    Yes – Return to caller
    No – Gone through all characters in the database?
            Yes – Return to caller
            No – Reset all temporary score to 0
                 Do the characters match?
                    Yes – Do the first characters match or the previous mark>3?
                            Yes - Add 2 marks to the temporary score
                    Check whether gone through all characters in search target
                    Yes – Return to the next character of the database
                    No – If continuous with same character, +3 mark
                            Else if continuous with uppercase and lowercase different +2 mark
                         If temperory score > mark, renew the mark with temperory score
                             Else if temperory score is half of the mark, +2 mark
                         If the search target is fully same with part of character in database, +100 mark
                No – Check whether gone through all characters in the search target
```

void arrange_search ()

```
Determine which product has the highest score according to grading system function
Is the score less than 0?
    Yes – Return to caller
    No – No product score matches max score?
            No – Copy product code corresponding to the score into a text file
                  Check to see if score is found
            Yes – subtract 1 from the max score
                  Check if max score is less than 0, then continue from there
```

void updatefile (string file1, string file2)

```
Check if end of file is reached
    Yes – Return to caller
    No – Transfer contents of one file into another
```

Account.h

void earning ()

```
Is file empty?
    Yes – Return to caller
    No – If IDs are same, print out user’s details such as time of purchase, buyer, product name, product code, quantity bought and profit earned
         Return to caller
```

void create ()

```
Prompt and get new user ID
Is the input same as any of the user IDs in the external file?
    Yes – Prompt and get new user ID again
    No – Prompt and get password
            Prompt and get password to double confirm
            Are the passwords the same?
                No – Prompt and get password again
                Yes – Encrypt password
                      Prompt and get account type
                      Validate account type
                      Prompt and get favourite number
                      Validate favourite number
                      Encrypt favourite number
                      Prompt and get email
                      Validate email
                      Prompt and get phone number
                       Validate phone number
                       Prompt and get address
                       Validate address
                       Save details to external file
                       Return to caller
```

int update_menu(char selection)

```
Get user’s selection from calling function
Selection is ‘1’: Prompt and get new password
                  Encrypt new password
Selection is ‘2’: Prompt and get new favourite number
                  Encrypt new favourite number
Selection is ‘3’: Prompt and get new email
Selection is ‘4’: Prompt and get new phone
Selection is ‘5’: Prompt and get new address
Selection is ‘0’: Return to caller
For selections 1 to 5, save updated user details in external file
```

int login ( )

```
Prompt and get user id or ‘1’ to create account
Create account?
    Yes - Call function named create
    No - Does user exist?
            No – Call function to create account
            Yes – Prompt and get password
                Forget password?
                    Yes - Prompt and get favourite number
                        Encrypt favourite number
                        Does favourite number match?
                            Yes – Reset password
                                Return account role (member or admin)
                            No – Prompt and get favourite number and check again
                    Input equals 0 – Return to previous step
                    No - Encrypt the password
                        Is the password correct?
                            Yes – Return account role (member or admin)
                            No – Prompt and get password again
```

Record.h

int search_record (string name_search)

```
Prompt and get product name to search
Marks are allocated based on similarity to search name
Check to see if end of record list is reached
    Yes – Display products row by row until end of file is reached
          Calculate number of rows displayed
          Return number of rows displayed
    No – Continue allocating marks to each product based on similarity to searched name
```

int view_record ()

```
Display products row by row until end of file is reached
Return number of rows displayed
```

void update_record ()

```
User inputs method of item selection
    User input is '0' – Return to caller
    Show all records – All products are displayed row by row
                      Prompt and get item to be updated
    Search for a particular record – User keys in product name to be searched
                                 The top 15 most relevant products are shown
                                 Prompt and get item to be updated
    If item does not belong to the user
        Repeat item selection process
    Knowing which item is to be updated
        Prompt and get new product name
        Validate new product name
            If the input is 0 skip the step (product name did not change)
        Prompt and get new stock number
        Validate new stock number
            If the input is 0 skip the step (stock number did not change)
        Prompt and get new unit price
        Validate new unit price
            If the input is 0 skip the step (unit price did not change)
        Update the product details in the list of records
        Return to caller
```

void update_buy (int code, int quantity)

```
Updates the stock number of products after customer makes a purchase by copying all records
into a temporary file and simultaneously updating a particular item's stock number
Replace the  original file with temporary file
Return to caller
```

void add_record ()

```
Checks the record list to determine first product number
Check if function has been looped 15 times?
    Yes – Save details to completed list and return to caller
    No – Prompt and get new product name
         Validate new product name
            Save completed data and exit?
                Yes – Save details to completed list and return to caller
                No – Quit without saving
                Other – treat as input of product name
        Prompt and get new stock number
            Validate new stock number
            Save completed data and exit?
                Yes – Save details to completed list and return to caller
                No – Quit without saving
                Other – treat as input of stock num
        Prompt and get new unit price
            Validate new unit price
            Save completed data and exit?
                Yes – Save details to completed list and return to caller
                No – Quit without saving
                Other – treat as input of unit price
Add 1 to counter
If the loop is over 15 or user press save and exit during data key-in, add the result into it
```

void delete_record ()

```
User inputs method of item selection
    User input is '0' – Return to caller
    Show all records – All products are displayed row by row
                       Prompt and get item to be deleted
    Search for a particular record – User keys in product name to be searched
                                     The top 15 most relevant products are shown
                                     Prompt and get item to be deleted
    If item to be deleted does not belong to the user
        Repeat process of item selection
    Knowing which item is to be deleted
        Delete item
        Return to caller
```

void record_menu ( )

```
Prompt and get record menu selection
Selection is ‘1’: Prompt and get product name to search
                  Call function to search for product
Selection is ‘2’: Call function to add a new product
Selection is ‘3’: Call function to delete an existing product
Selection is ‘4’: Call function to update a product’s details
Selection is ‘5’: Call function to view all products
Selection is ‘0’: Return to admin main menu
```

Payment.h

int payment ()

```
Display user’s cart and calculate the total amount to be paid
Is cart empty?
    Yes – Display message to indicate cart has no products
    No – Is the total amount to be paid greater than 0?
            No - Prompt user whether to quit program or return to user menu
                Return status indicating payment method used and whether to quit program
            Yes – Does user want to pay for all items in cart?
                    Yes – Calculate total price of all selected products
                    No – User selects several particular products to pay for
                         Calculate total price of selected products
                    Does user want to user want to enter a new delivery address?
                        Yes – Prompt and get new address
                            Validate new address
                            Continue with payment below
                        No – Does user want to pay by credit card?
                                Yes – Prompt and get card number and cvv
                                      Validate card number and cvv
                                      Display and save invoice
                                      Return status indicating payment method used
                                No – User wishes to pay by cash?
                                        No – Quit payment
                                        Yes – Display and save invoice
                                               Return status indicating payment method used
```

int select_item ()

```
Are there 1 or more selected items already?
    Yes – Calculate total price and display item
    No – Prompt and get product name to search
         Relevant products are displayed
         User selects item to be added into cart
         User inputs quantity of item to be purchased
         Calculate total price and display item
          Finished selecting products?
            Yes – Does user want to save items into cart?
                     Yes – Save items into cart and continue below
                      No – Prompt user whether to proceed to payment or return to menu
                           Return payment choice
```

double printcart (string file,int &total_item)

Check products in cart. If product appears multiple times, add up quantity to be bought. Save products and quantity into a temporary cart
Each product’s quantity will be checked against the product’s maximum stock. If exceeded, the maximum stock will be used instead.
Save products and quantity into user’s cart
Display cart
Calculate and return subtotal of products in cart

Feedback.h

int provide_feedback ()

```
Does user want to provide feedback?
    No – Return to caller
    Yes – Prompt type of feedback:
            System feedback: Prompt and get feedback
            Service feedback: Prompt and get feedback
            Product feedback: Prompt and get name of product
                              Search for product in record list
                              Prompt and get feedback for that particular product if it exists
    Validate feedback
    Display and save feedback
```

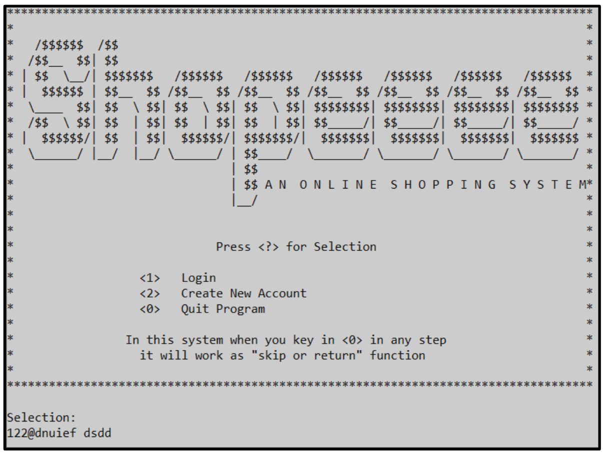
int view_feedback ()

```
Prompt and get type of feedback to be viewed
    System feedback: Display system feedback
    Service feedback: Display service feedback
    Product feedback: Prompt and get name of product
                      Search for product in record list
                      Display feedback on that particular product
If no feedback found, display a message to indicate so
Return to caller
```

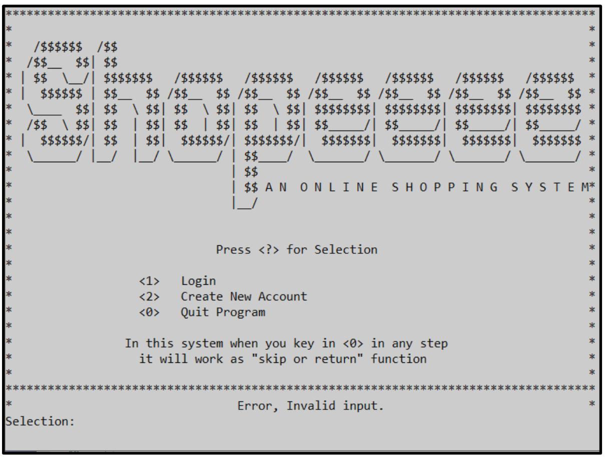
Test Case:

System can detect any of the error input.

Error input:

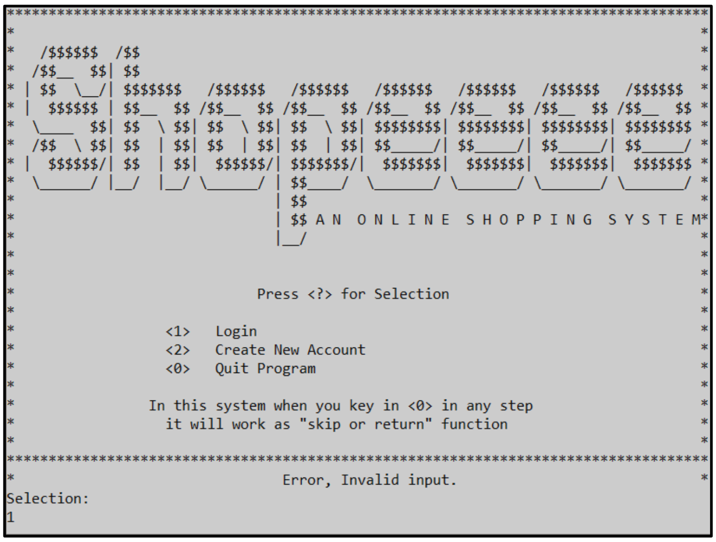


```
********************************************************************************
*                                                                              *
*   /$$$$$$  /$$                                                               *
*  /$$__  $$| $$                                                               *
* | $$  \__/| $$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$  *
* |  $$$$$$ | $$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ *
*  \____  $$| $$  \ $$| $$  \ $$| $$  \ $$| $$$$$$$$| $$$$$$$$| $$$$$$$$ *
*  /$$  \ $$| $$  | $$| $$  | $$| $$  | $$| $$_____/| $$_____/| $$_____/ *
* |  $$$$$$/| $$  | $$|  $$$$$$/| $$$$$$$/|  $$$$$$$|  $$$$$$$|  $$$$$$$ *
*  \______/ |__/  |__/ \______/ | $$____/  \_______/ \_______/ \_______/ *
*                               | $$                                     *
*                               | $$ A N  O N L I N E  S H O P P I N G  S Y S T E M*
*                               |__/                                     *
*                                                                        *
*                                                                        *
*                         Press <?> for Selection                        *
*                                                                        *
*                   <1>   Login                                          *
*                   <2>   Create New Account                             *
*                   <0>   Quit Program                                   *
*                                                                        *
*                 In this system when you key in <0> in any step         *
*                   it will work as "skip or return" function            *
*                                                                        *
**************************************************************************

Selection:
122@dnuief dsdd
```

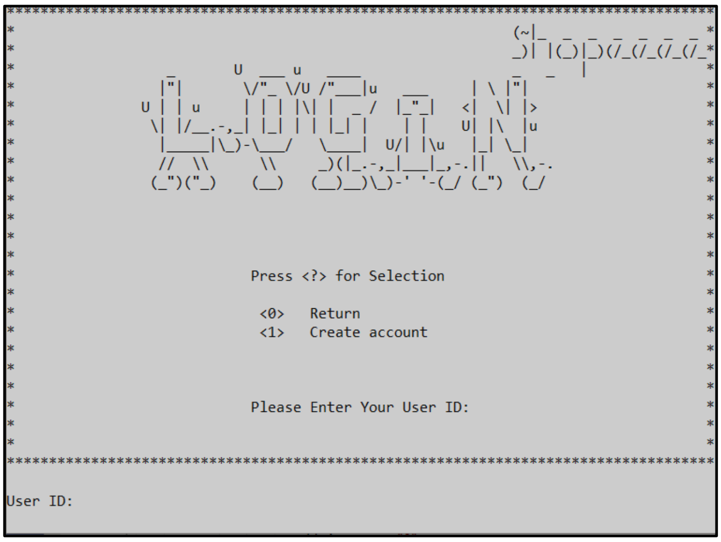


```
********************************************************************************
*                                                                              *
*   /$$$$$$  /$$                                                               *
*  /$$__  $$| $$                                                               *
* | $$  \__/| $$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$  *
* |  $$$$$$ | $$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ *
*  \____  $$| $$  \ $$| $$  \ $$| $$  \ $$| $$$$$$$$| $$$$$$$$| $$$$$$$$ *
*  /$$  \ $$| $$  | $$| $$  | $$| $$  | $$| $$_____/| $$_____/| $$_____/ *
* |  $$$$$$/| $$  | $$|  $$$$$$/| $$$$$$$/|  $$$$$$$|  $$$$$$$|  $$$$$$$ *
*  \______/ |__/  |__/ \______/ | $$____/  \_______/ \_______/ \_______/ *
*                               | $$                                     *
*                               | $$ A N  O N L I N E  S H O P P I N G  S Y S T E M*
*                               |__/                                     *
*                                                                        *
*                                                                        *
*                         Press <?> for Selection                        *
*                                                                        *
*                   <1>   Login                                          *
*                   <2>   Create New Account                             *
*                   <0>   Quit Program                                   *
*                                                                        *
*                 In this system when you key in <0> in any step         *
*                   it will work as "skip or return" function            *
*                                                                        *
**************************************************************************
*                            Error, Invalid input.                       *
Selection:
```

Correct input

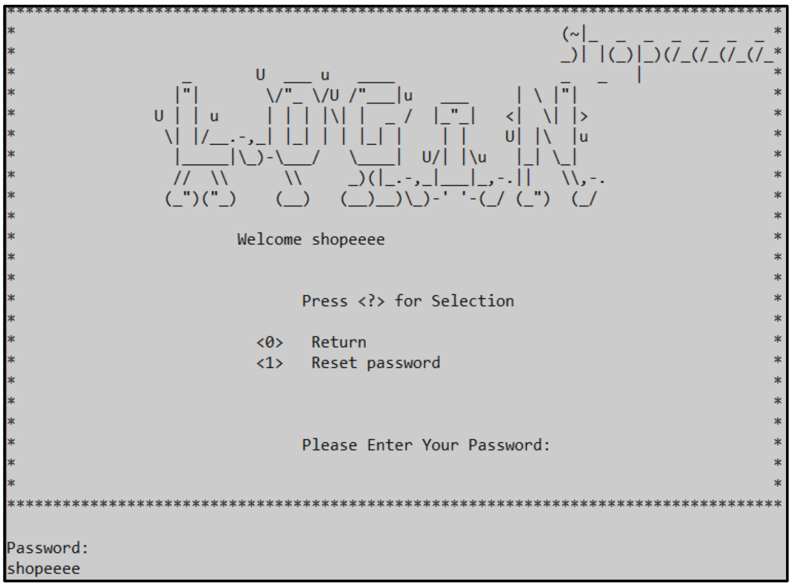


```
********************************************************************************
*                                                                              *
*   /$$$$$$  /$$                                                               *
*  /$$__  $$| $$                                                               *
* | $$  \__/| $$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$   /$$$$$$  *
* |  $$$$$$ | $$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ /$$__  $$ *
*  \____  $$| $$  \ $$| $$  \ $$| $$  \ $$| $$$$$$$$| $$$$$$$$| $$$$$$$$ *
*  /$$  \ $$| $$  | $$| $$  | $$| $$  | $$| $$_____/| $$_____/| $$_____/ *
* |  $$$$$$/| $$  | $$|  $$$$$$/| $$$$$$$/|  $$$$$$$|  $$$$$$$|  $$$$$$$ *
*  \______/ |__/  |__/ \______/ | $$____/  \_______/ \_______/ \_______/ *
*                               | $$                                     *
*                               | $$ A N  O N L I N E  S H O P P I N G  S Y S T E M*
*                               |__/                                     *
*                                                                        *
*                                                                        *
*                         Press <?> for Selection                        *
*                                                                        *
*                   <1>   Login                                          *
*                   <2>   Create New Account                             *
*                   <0>   Quit Program                                   *
*                                                                        *
*                 In this system when you key in <0> in any step         *
*                   it will work as "skip or return" function            *
*                                                                        *
**************************************************************************
*                            Error, Invalid input.                       *
Selection:
1
```

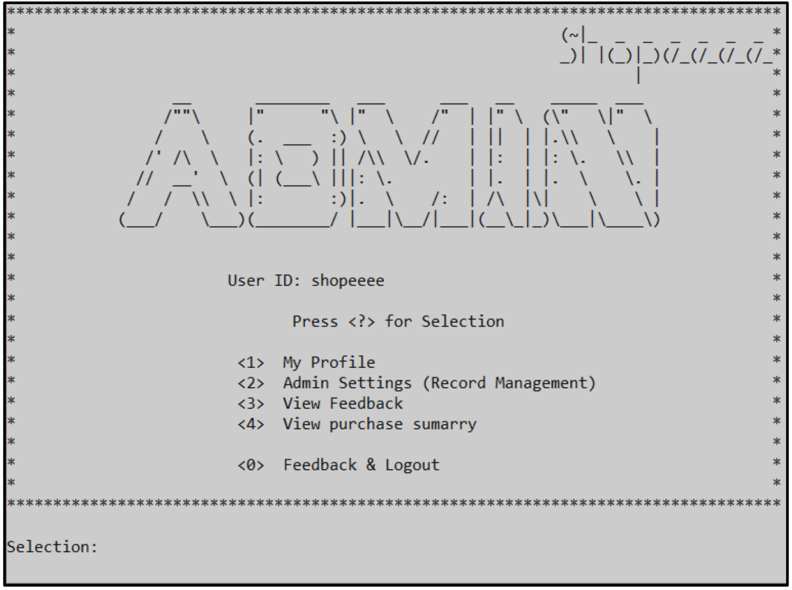


```
**************************************************************************
*                                                   (~|_  _  _  _  _  _  _ *
*                                                   _)| |(_)|_)(/_(/_(/_(/_*
*                 _        U  ___ u  ____             _   _  |             *
*                |"|        \/"_ \/U /"___|u   ___    | \ |"|              *
*              U | | u      | | | |\| |  _ /  |_"_|  <|  \| |>             *
*               \| |/__.-,_| |_| | | |_| |    | |    U| |\  |u             *
*                |_____|\_)-\___/   \____|  U/| |\u   |_| \_|              *
*                //  \\      \\     _)(|_.-,_|___|_,-.||   \\,-.           *
*               (_")("_)    (__)   (__)__)\_)-' '-(_/ (_")  (_/            *
*                                                                          *
*                                                                          *
*                                                                          *
*                                                                          *
*                         Press <?> for Selection                          *
*                                                                          *
*                          <0>   Return                                    *
*                          <1>   Create account                            *
*                                                                          *
*                                                                          *
*                                                                          *
*                         Please Enter Your User ID:                       *
*                                                                          *
*                                                                          *
****************************************************************************

User ID:
```

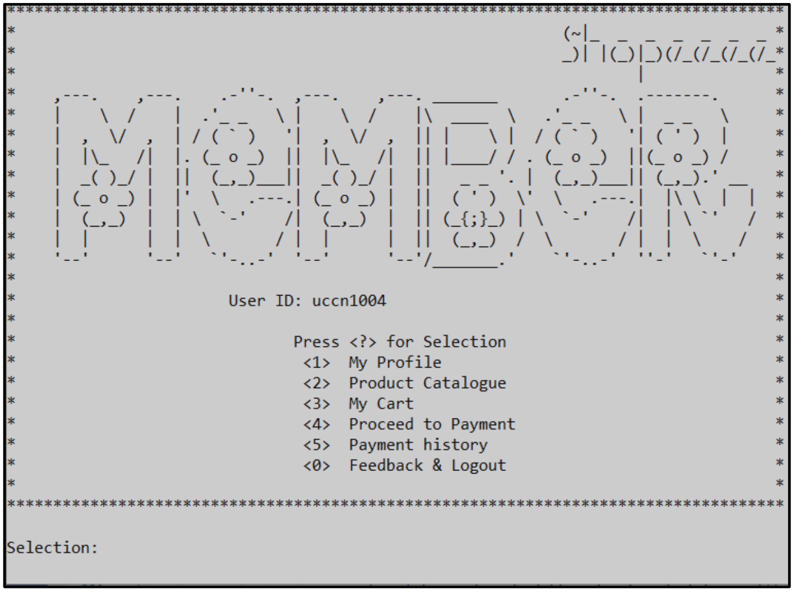
User can login into corresponding account type, system will automatically determine which account type they are.



```
Welcome shopeeee

        Press <?> for Selection

    <0>   Return
    <1>   Reset password


        Please Enter Your Password:

Password:
shopeeee
```

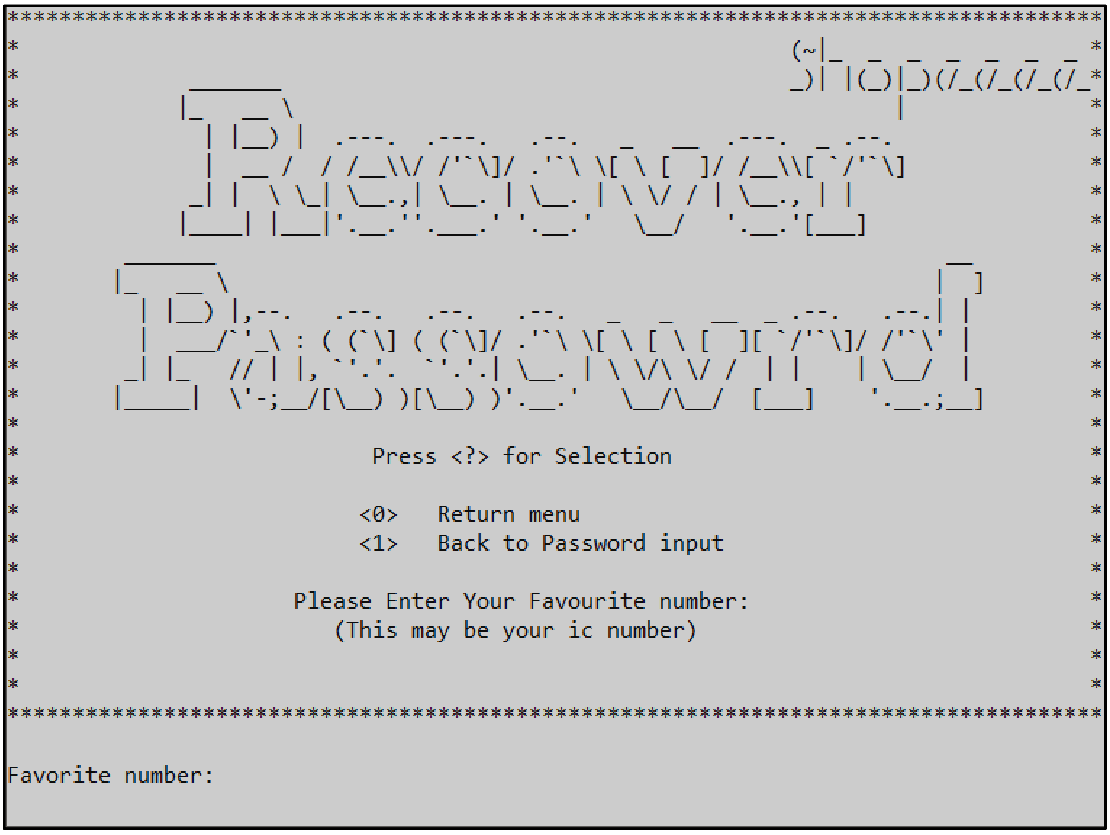


```
User ID: shopeeee

        Press <?> for Selection

<1>  My Profile
<2>  Admin Settings (Record Management)
<3>  View Feedback
<4>  View purchase sumarry

<0>  Feedback & Logout

Selection:
```

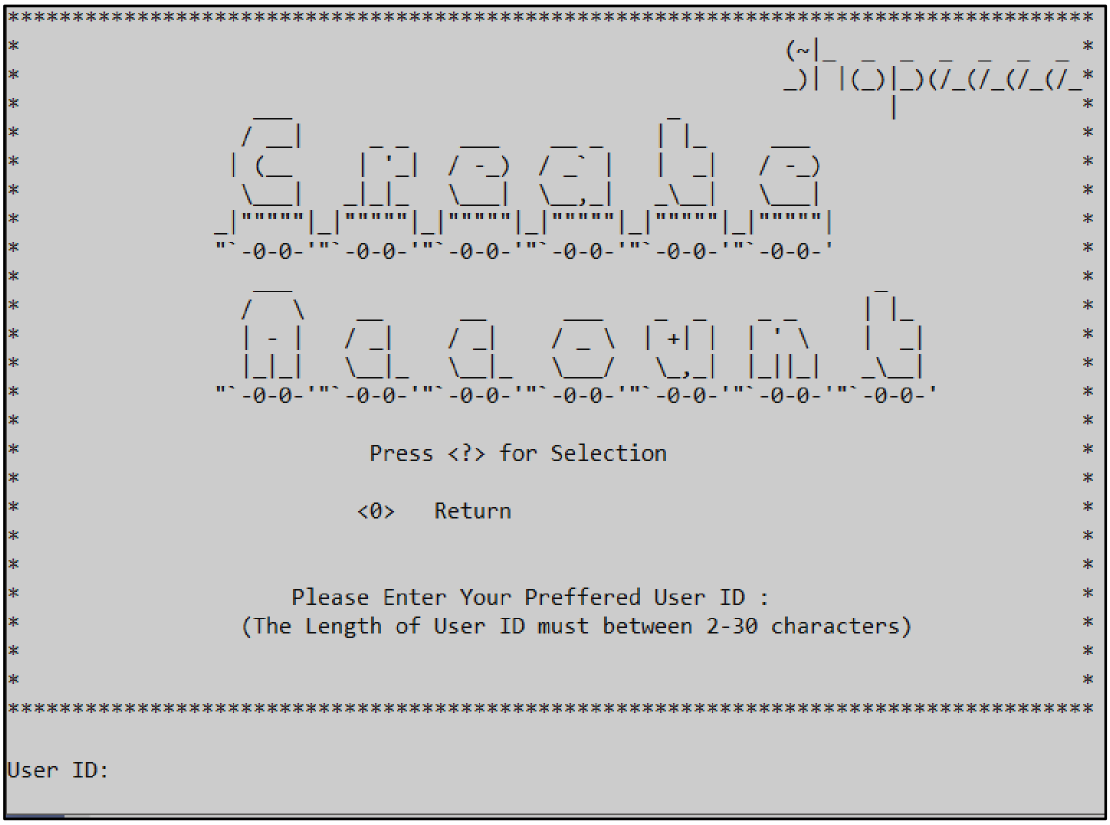


```
User ID: uccn1004

        Press <?> for Selection
         <1>  My Profile
         <2>  Product Catalogue
         <3>  My Cart
         <4>  Proceed to Payment
         <5>  Payment history
         <0>  Feedback & Logout

Selection:
```

User also can make selection while in the key-in section such as create account and forget password



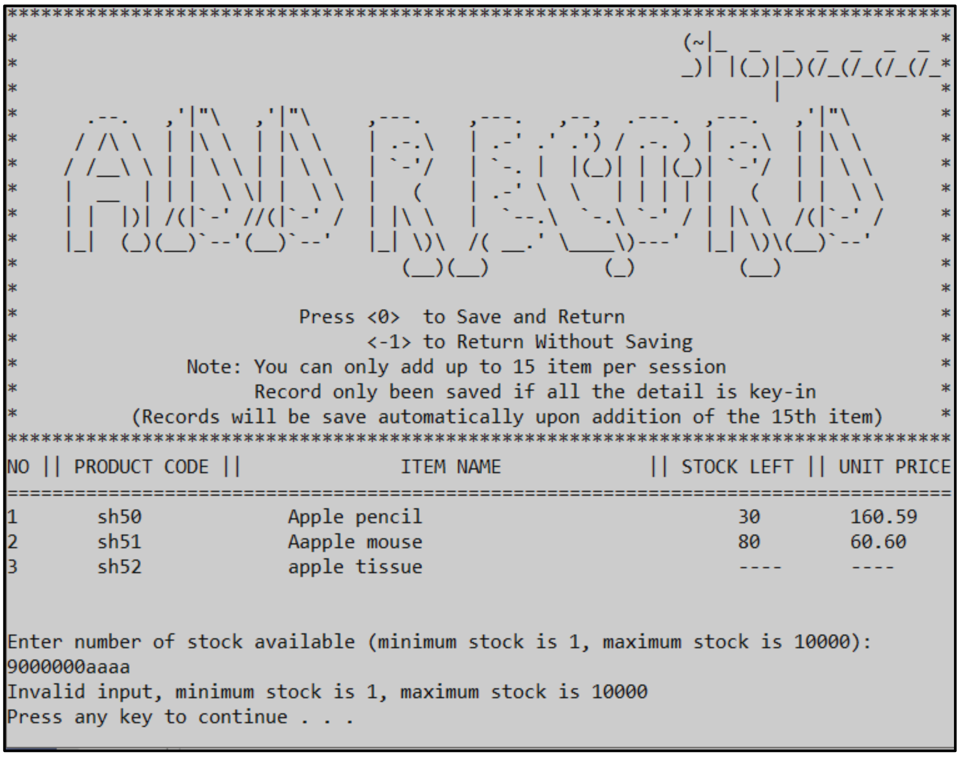
```
                    Press <?> for Selection

                      <0>    Return menu
                      <1>    Back to Password input

                 Please Enter Your Favourite number:
                    (This may be your ic number)

Favorite number:
```



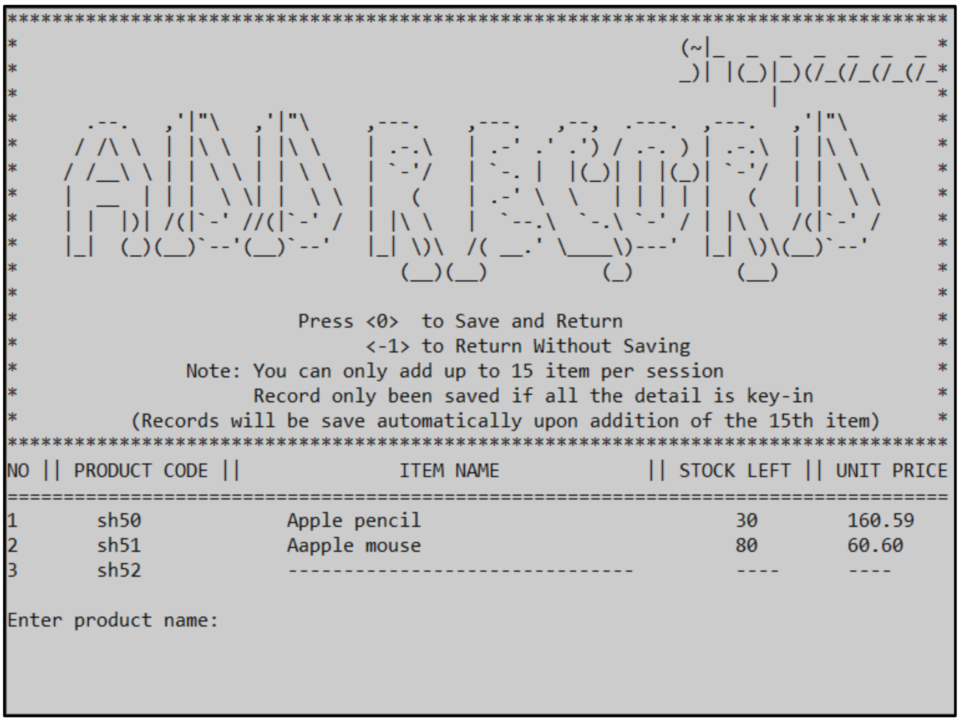
```
                    Press <?> for Selection

                      <0>    Return


                Please Enter Your Preffered User ID :
            (The Length of User ID must between 2-30 characters)

User ID:
```

User can add record

System Can detect the error if any invalid input..



```
***********************************************************************************
*                                                          (~|_  _  _  _  _  _  _ *
*                                                          _)| |(_)|_)(/_(/_(/_(/_*
*                                                                  |              *
*     .--.   ,'|"\   ,'|"\    ,---.    ,---.  ,--,  .---.  ,---.   ,'|"\          *
*    / /\ \  | |\ \  | |\ \   | .-.\   | .-' .' .') / .-. ) | .-.\  | |\ \        *
*   / /__\ \ | | \ \ | | \ \  | `-'/   | `-. |  |(_)| | |(_)| `-'/  | | \ \       *
*   |  __  | | |  \ \| |  \ \ |   (    | .-' \  \   | | | | |   (   | |  \ \      *
*   | |  |)| /(|`-' //(|`-' / | |\ \   |  `--.\  `-.\ `-' / | |\ \  /(|`-' /      *
*   |_|  (_)(__)`--'(__)`--'  |_| \)\  /( __.' \____\)---'  |_| \)\(__)`--'       *
*                                 (__)(__)          (_)        (__)               *
*                                                                                 *
*                          Press <0>  to Save and Return                          *
*                                <-1> to Return Without Saving                    *
*                    Note: You can only add up to 15 item per session             *
*                          Record only been saved if all the detail is key-in     *
*           (Records will be save automatically upon addition of the 15th item)   *
***********************************************************************************
NO || PRODUCT CODE ||              ITEM NAME             || STOCK LEFT || UNIT PRICE
===================================================================================
1       sh50             Apple pencil                            30        160.59
2       sh51             Aapple mouse                            80        60.60
3       sh52             apple tissue                            ----      ----


Enter number of stock available (minimum stock is 1, maximum stock is 10000):
9000000aaaa
Invalid input, minimum stock is 1, maximum stock is 10000
Press any key to continue . . .
```



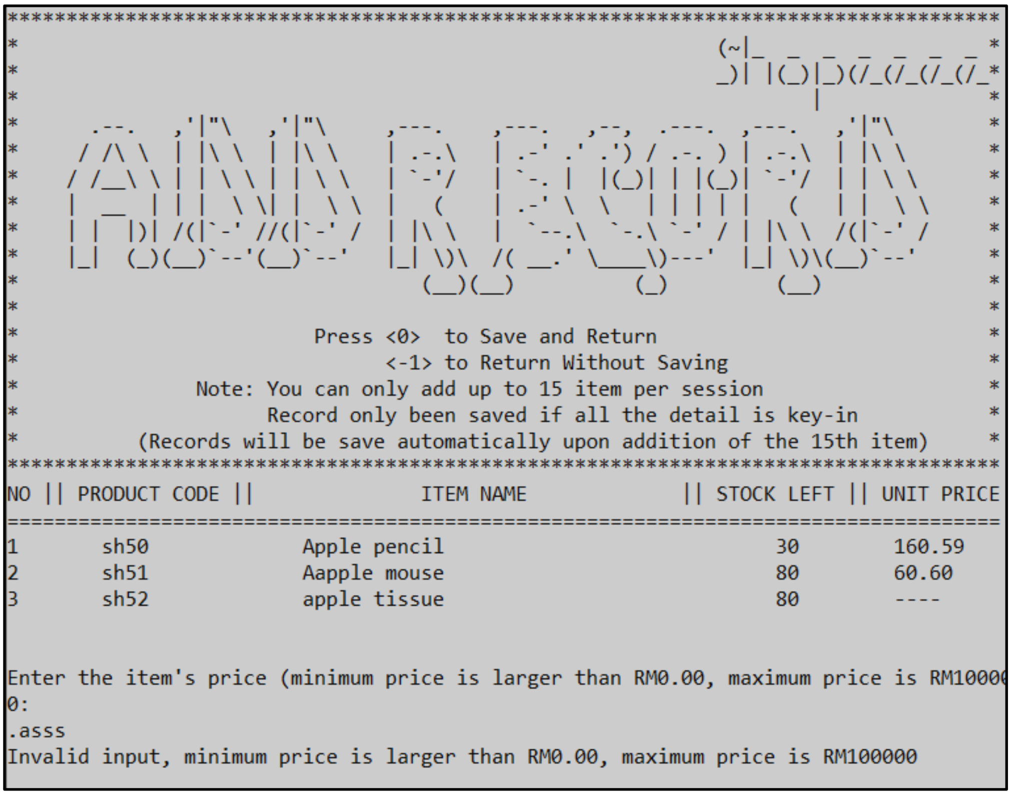
```
***********************************************************************************
*                                                          (~|_  _  _  _  _  _  _ *
*                                                          _)| |(_)|_)(/_(/_(/_(/_*
*                                                                  |              *
*     .--.   ,'|"\   ,'|"\    ,---.    ,---.  ,--,  .---.  ,---.   ,'|"\          *
*    / /\ \  | |\ \  | |\ \   | .-.\   | .-' .' .') / .-. ) | .-.\  | |\ \        *
*   / /__\ \ | | \ \ | | \ \  | `-'/   | `-. |  |(_)| | |(_)| `-'/  | | \ \       *
*   |  __  | | |  \ \| |  \ \ |   (    | .-' \  \   | | | | |   (   | |  \ \      *
*   | |  |)| /(|`-' //(|`-' / | |\ \   |  `--.\  `-.\ `-' / | |\ \  /(|`-' /      *
*   |_|  (_)(__)`--'(__)`--'  |_| \)\  /( __.' \____\)---'  |_| \)\(__)`--'       *
*                                 (__)(__)          (_)        (__)               *
*                                                                                 *
*                          Press <0>  to Save and Return                          *
*                                <-1> to Return Without Saving                    *
*                    Note: You can only add up to 15 item per session             *
*                          Record only been saved if all the detail is key-in     *
*           (Records will be save automatically upon addition of the 15th item)   *
***********************************************************************************
NO || PRODUCT CODE ||              ITEM NAME             || STOCK LEFT || UNIT PRICE
===================================================================================
1       sh50             Apple pencil                            30        160.59
2       sh51             Aapple mouse                            80        60.60
3       sh52             ------------------------------          ----      ----

Enter product name:
```

User can exit the add record by using sentinel value

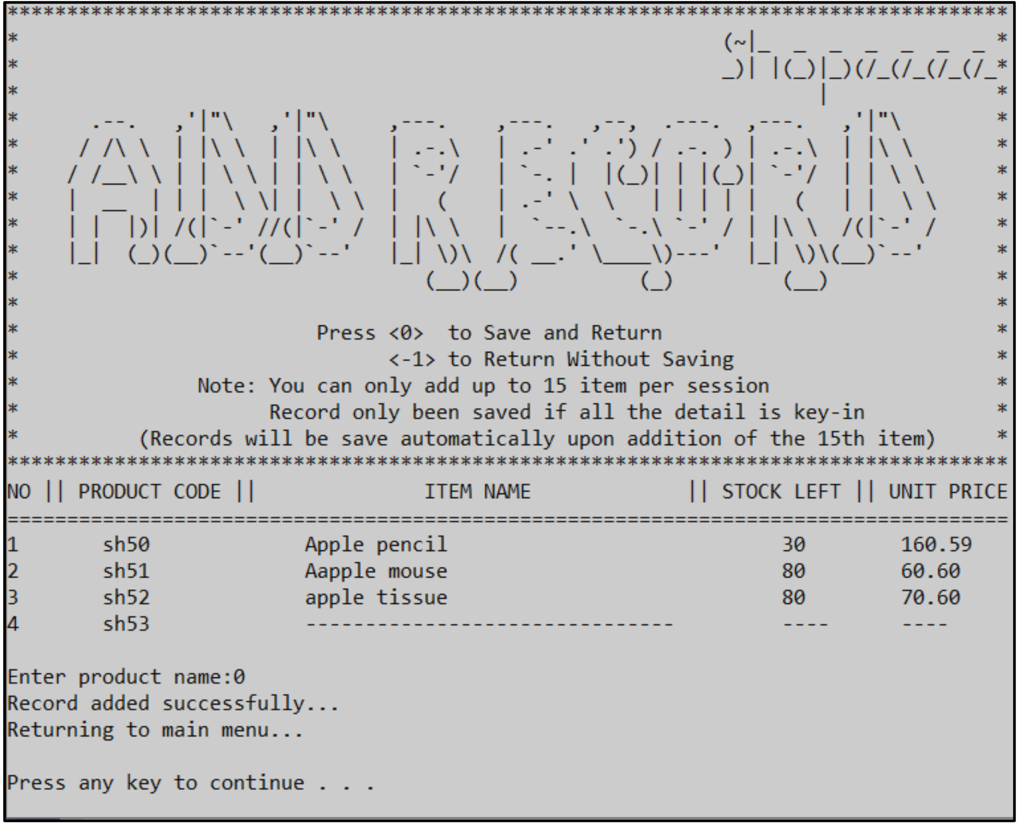
They record will be save when user want to by key-in only “0”.

When the detail is completely, the record will save, otherwise not.

User also can choose not save by key-in “1”

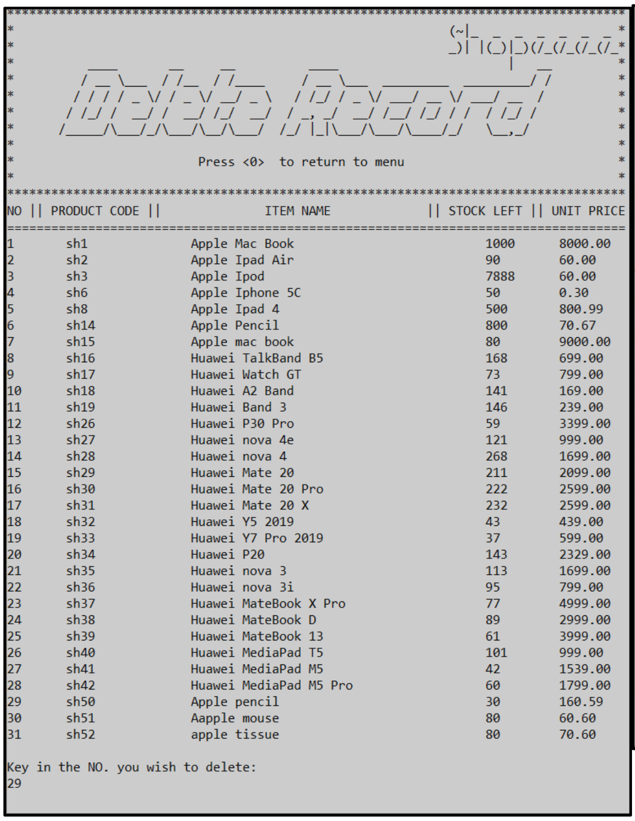


```
************************************************************************************
*                                                           (~|_  _  _  _  _  _  _  *
*                                                           _)| |(_)|_)(/_(/_(/_(/_*
*                                                                   |               *
*      .--.   ,'|"\   ,'|"\     ,---.    ,---.   ,--,  .---.  ,---.    ,'|"\       *
*     / /\ \  | |\ \  | |\ \    | .-.\   | .-' .' .') / .-. ) | .-.\   | |\ \      *
*    / /__\ \ | | \ \ | | \ \   | `-'/   | `-. |  |(_)| | |(_)| `-'/   | | \ \     *
*    |  __  | | |  \ \| |  \ \  |   (    | .-' \  \   | | | |   (     | |  \ \    *
*    | |  |)| /(|`-' //(|`-' /  | |\ \   |  `--.\  `-.\ `-' / | |\ \  /(|`-' /    *
*    |_|  (_)(__)`--'(__)`--'   |_| \)\  /( __.' \____\)---'  |_| \)\(__)`--'     *
*                                   (__)(__)         (_)          (__)             *
*                                                                                   *
*                          Press <0>  to Save and Return                           *
*                                <-1> to Return Without Saving                     *
*                Note: You can only add up to 15 item per session                  *
*                      Record only been saved if all the detail is key-in          *
*           (Records will be save automatically upon addition of the 15th item)    *
************************************************************************************
NO || PRODUCT CODE ||              ITEM NAME             || STOCK LEFT || UNIT PRICE
====================================================================================
1       sh50            Apple pencil                               30        160.59
2       sh51            Aapple mouse                               80        60.60
3       sh52            apple tissue                               80        ----


Enter the item's price (minimum price is larger than RM0.00, maximum price is RM10000
0:
.asss
Invalid input, minimum price is larger than RM0.00, maximum price is RM100000
```

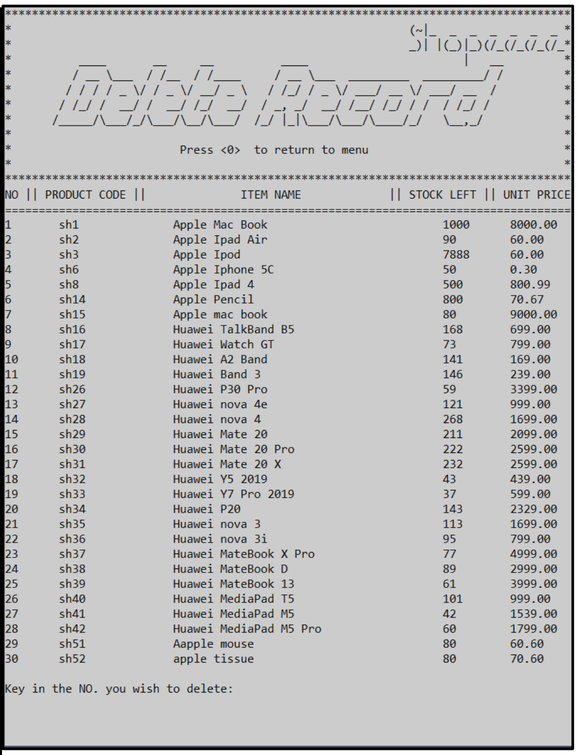


```
************************************************************************************
*                                                           (~|_  _  _  _  _  _  _  *
*                                                           _)| |(_)|_)(/_(/_(/_(/_*
*                                                                   |               *
*      .--.   ,'|"\   ,'|"\     ,---.    ,---.   ,--,  .---.  ,---.    ,'|"\       *
*     / /\ \  | |\ \  | |\ \    | .-.\   | .-' .' .') / .-. ) | .-.\   | |\ \      *
*    / /__\ \ | | \ \ | | \ \   | `-'/   | `-. |  |(_)| | |(_)| `-'/   | | \ \     *
*    |  __  | | |  \ \| |  \ \  |   (    | .-' \  \   | | | |   (     | |  \ \    *
*    | |  |)| /(|`-' //(|`-' /  | |\ \   |  `--.\  `-.\ `-' / | |\ \  /(|`-' /    *
*    |_|  (_)(__)`--'(__)`--'   |_| \)\  /( __.' \____\)---'  |_| \)\(__)`--'     *
*                                   (__)(__)         (_)          (__)             *
*                                                                                   *
*                          Press <0>  to Save and Return                           *
*                                <-1> to Return Without Saving                     *
*                Note: You can only add up to 15 item per session                  *
*                      Record only been saved if all the detail is key-in          *
*           (Records will be save automatically upon addition of the 15th item)    *
************************************************************************************
NO || PRODUCT CODE ||              ITEM NAME             || STOCK LEFT || UNIT PRICE
====================================================================================
1       sh50            Apple pencil                               30        160.59
2       sh51            Aapple mouse                               80        60.60
3       sh52            apple tissue                               80        70.60
4       sh53            -------------------------------            ----      ----

Enter product name:0
Record added successfully...
Returning to main menu...

Press any key to continue . . .
```

User can delete record.

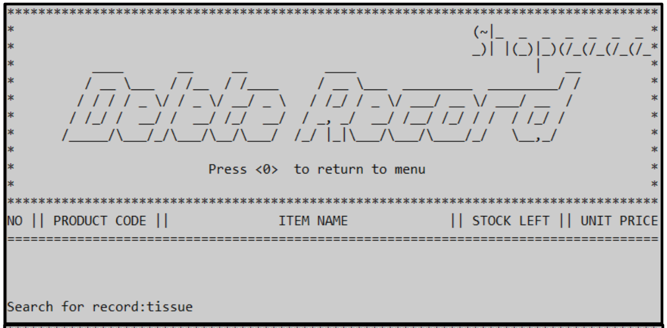


```
*******************************************************************************
*                                                    (~|_  _  _  _  _  _  _   *
*                                                    _)| |(_)|_)(/_(/_(/_(/_ *
*       ____       __     __          ____                  |       __        *
*      / __ \___  / /__  / /____     / __ \___  _________  _________/ /        *
*     / / / / _ \/ / _ \/ __/ _ \   / /_/ / _ \/ ___/ __ \/ ___/ __  /         *
*    / /_/ /  __/ /  __/ /_/  __/  / _, _/  __/ /__/ /_/ / /  / /_/ /          *
*   /_____/\___/_/\___/\__/\___/  /_/ |_|\___/\___/\____/_/   \__,_/           *
*                                                                             *
*                         Press <0>  to return to menu                        *
*                                                                             *
*******************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
==============================================================================
1       sh1             Apple Mac Book                          1000       8000.00
2       sh2             Apple Ipad Air                          90         60.00
3       sh3             Apple Ipod                              7888       60.00
4       sh6             Apple Iphone 5C                         50         0.30
5       sh8             Apple Ipad 4                            500        800.99
6       sh14            Apple Pencil                            800        70.67
7       sh15            Apple mac book                          80         9000.00
8       sh16            Huawei TalkBand B5                      168        699.00
9       sh17            Huawei Watch GT                         73         799.00
10      sh18            Huawei A2 Band                          141        169.00
11      sh19            Huawei Band 3                           146        239.00
12      sh26            Huawei P30 Pro                          59         3399.00
13      sh27            Huawei nova 4e                          121        999.00
14      sh28            Huawei nova 4                           268        1699.00
15      sh29            Huawei Mate 20                          211        2099.00
16      sh30            Huawei Mate 20 Pro                      222        2599.00
17      sh31            Huawei Mate 20 X                        232        2599.00
18      sh32            Huawei Y5 2019                          43         439.00
19      sh33            Huawei Y7 Pro 2019                      37         599.00
20      sh34            Huawei P20                              143        2329.00
21      sh35            Huawei nova 3                           113        1699.00
22      sh36            Huawei nova 3i                          95         799.00
23      sh37            Huawei MateBook X Pro                   77         4999.00
24      sh38            Huawei MateBook D                       89         2999.00
25      sh39            Huawei MateBook 13                      61         3999.00
26      sh40            Huawei MediaPad T5                      101        999.00
27      sh41            Huawei MediaPad M5                      42         1539.00
28      sh42            Huawei MediaPad M5 Pro                  60         1799.00
29      sh50            Apple pencil                            30         160.59
30      sh51            Aapple mouse                            80         60.60
31      sh52            apple tissue                            80         70.60

Key in the NO. you wish to delete:
29
```

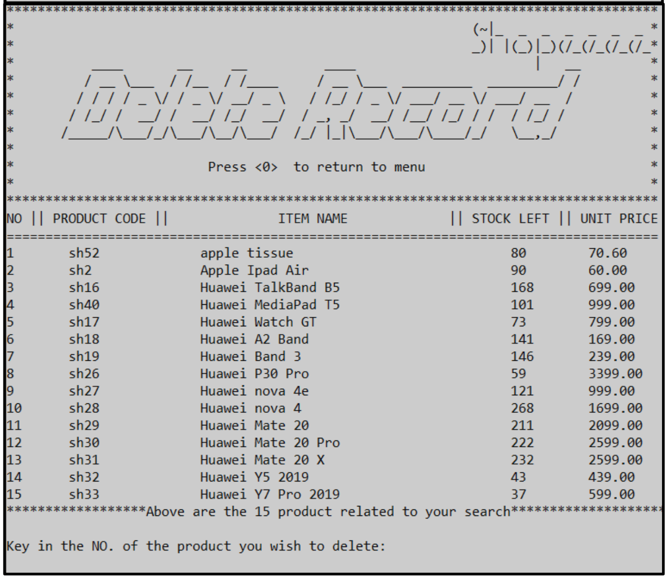


```
*******************************************************************************
*                                                    (~|_  _  _  _  _  _  _   *
*                                                    _)| |(_)|_)(/_(/_(/_(/_ *
*       ____       __     __          ____                  |       __        *
*      / __ \___  / /__  / /____     / __ \___  _________  _________/ /        *
*     / / / / _ \/ / _ \/ __/ _ \   / /_/ / _ \/ ___/ __ \/ ___/ __  /         *
*    / /_/ /  __/ /  __/ /_/  __/  / _, _/  __/ /__/ /_/ / /  / /_/ /          *
*   /_____/\___/_/\___/\__/\___/  /_/ |_|\___/\___/\____/_/   \__,_/           *
*                                                                             *
*                         Press <0>  to return to menu                        *
*                                                                             *
*******************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
==============================================================================
1       sh1             Apple Mac Book                          1000       8000.00
2       sh2             Apple Ipad Air                          90         60.00
3       sh3             Apple Ipod                              7888       60.00
4       sh6             Apple Iphone 5C                         50         0.30
5       sh8             Apple Ipad 4                            500        800.99
6       sh14            Apple Pencil                            800        70.67
7       sh15            Apple mac book                          80         9000.00
8       sh16            Huawei TalkBand B5                      168        699.00
9       sh17            Huawei Watch GT                         73         799.00
10      sh18            Huawei A2 Band                          141        169.00
11      sh19            Huawei Band 3                           146        239.00
12      sh26            Huawei P30 Pro                          59         3399.00
13      sh27            Huawei nova 4e                          121        999.00
14      sh28            Huawei nova 4                           268        1699.00
15      sh29            Huawei Mate 20                          211        2099.00
16      sh30            Huawei Mate 20 Pro                      222        2599.00
17      sh31            Huawei Mate 20 X                        232        2599.00
18      sh32            Huawei Y5 2019                          43         439.00
19      sh33            Huawei Y7 Pro 2019                      37         599.00
20      sh34            Huawei P20                              143        2329.00
21      sh35            Huawei nova 3                           113        1699.00
22      sh36            Huawei nova 3i                          95         799.00
23      sh37            Huawei MateBook X Pro                   77         4999.00
24      sh38            Huawei MateBook D                       89         2999.00
25      sh39            Huawei MateBook 13                      61         3999.00
26      sh40            Huawei MediaPad T5                      101        999.00
27      sh41            Huawei MediaPad M5                      42         1539.00
28      sh42            Huawei MediaPad M5 Pro                  60         1799.00
29      sh51            Aapple mouse                            80         60.60
30      sh52            apple tissue                            80         70.60

Key in the NO. you wish to delete:
```

User can select the record by search or view entire database

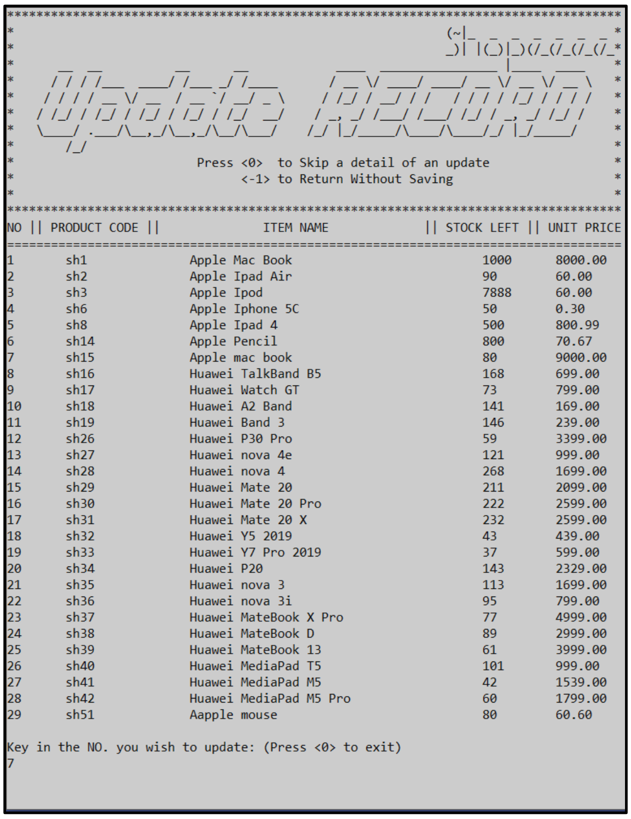


```
*******************************************************************************
*                                                    (~|_  _  _  _  _  _  _   *
*                                                    _)| |(_)|_)(/_(/_(/_(/_ *
*       ____       __     __          ____                  |       __        *
*      / __ \___  / /__  / /____     / __ \___  _________  _________/ /        *
*     / / / / _ \/ / _ \/ __/ _ \   / /_/ / _ \/ ___/ __ \/ ___/ __  /         *
*    / /_/ /  __/ /  __/ /_/  __/  / _, _/  __/ /__/ /_/ / /  / /_/ /          *
*   /_____/\___/_/\___/\__/\___/  /_/ |_|\___/\___/\____/_/   \__,_/           *
*                                                                             *
*                         Press <0>  to return to menu                        *
*                                                                             *
*******************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
==============================================================================


Search for record:tissue
```

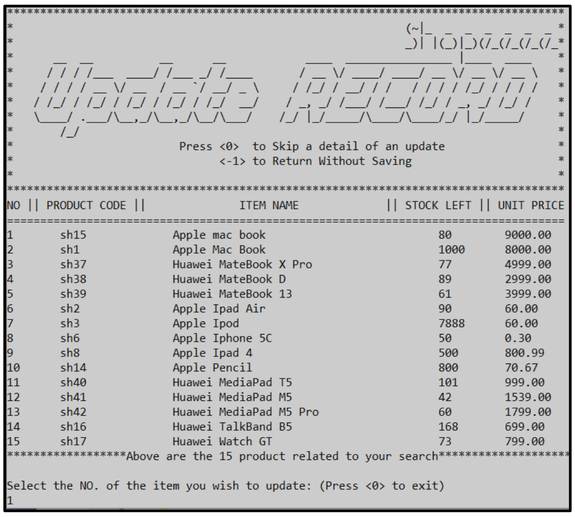


```
*******************************************************************************
*                                                    (~|_  _  _  _  _  _  _   *
*                                                    _)| |(_)|_)(/_(/_(/_(/_ *
*       ____       __     __          ____                  |       __        *
*      / __ \___  / /__  / /____     / __ \___  _________  _________/ /        *
*     / / / / _ \/ / _ \/ __/ _ \   / /_/ / _ \/ ___/ __ \/ ___/ __  /         *
*    / /_/ /  __/ /  __/ /_/  __/  / _, _/  __/ /__/ /_/ / /  / /_/ /          *
*   /_____/\___/_/\___/\__/\___/  /_/ |_|\___/\___/\____/_/   \__,_/           *
*                                                                             *
*                         Press <0>  to return to menu                        *
*                                                                             *
*******************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
==============================================================================
1       sh52            apple tissue                            80         70.60
2       sh2             Apple Ipad Air                          90         60.00
3       sh16            Huawei TalkBand B5                      168        699.00
4       sh40            Huawei MediaPad T5                      101        999.00
5       sh17            Huawei Watch GT                         73         799.00
6       sh18            Huawei A2 Band                          141        169.00
7       sh19            Huawei Band 3                           146        239.00
8       sh26            Huawei P30 Pro                          59         3399.00
9       sh27            Huawei nova 4e                          121        999.00
10      sh28            Huawei nova 4                           268        1699.00
11      sh29            Huawei Mate 20                          211        2099.00
12      sh30            Huawei Mate 20 Pro                      222        2599.00
13      sh31            Huawei Mate 20 X                        232        2599.00
14      sh32            Huawei Y5 2019                          43         439.00
15      sh33            Huawei Y7 Pro 2019                      37         599.00
*****************Above are the 15 product related to your search*******************

Key in the NO. of the product you wish to delete:
```

User can update the record by search or view entire database

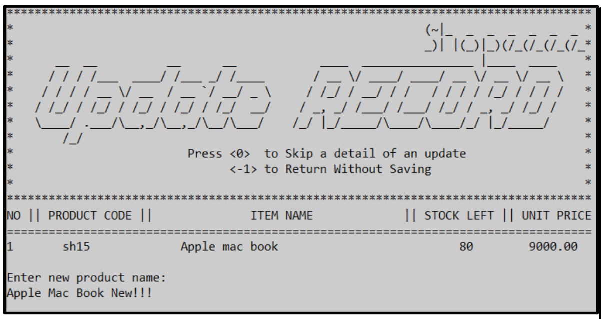


```
Press <0>  to Skip a detail of an update
 <-1> to Return Without Saving

NO || PRODUCT CODE ||              ITEM NAME              || STOCK LEFT || UNIT PRICE
======================================================================================
1       sh1             Apple Mac Book                         1000       8000.00
2       sh2             Apple Ipad Air                         90         60.00
3       sh3             Apple Ipod                             7888       60.00
4       sh6             Apple Iphone 5C                        50         0.30
5       sh8             Apple Ipad 4                           500        800.99
6       sh14            Apple Pencil                           800        70.67
7       sh15            Apple mac book                         80         9000.00
8       sh16            Huawei TalkBand B5                     168        699.00
9       sh17            Huawei Watch GT                        73         799.00
10      sh18            Huawei A2 Band                         141        169.00
11      sh19            Huawei Band 3                          146        239.00
12      sh26            Huawei P30 Pro                         59         3399.00
13      sh27            Huawei nova 4e                         121        999.00
14      sh28            Huawei nova 4                          268        1699.00
15      sh29            Huawei Mate 20                         211        2099.00
16      sh30            Huawei Mate 20 Pro                     222        2599.00
17      sh31            Huawei Mate 20 X                       232        2599.00
18      sh32            Huawei Y5 2019                         43         439.00
19      sh33            Huawei Y7 Pro 2019                     37         599.00
20      sh34            Huawei P20                             143        2329.00
21      sh35            Huawei nova 3                          113        1699.00
22      sh36            Huawei nova 3i                         95         799.00
23      sh37            Huawei MateBook X Pro                  77         4999.00
24      sh38            Huawei MateBook D                      89         2999.00
25      sh39            Huawei MateBook 13                     61         3999.00
26      sh40            Huawei MediaPad T5                     101        999.00
27      sh41            Huawei MediaPad M5                     42         1539.00
28      sh42            Huawei MediaPad M5 Pro                 60         1799.00
29      sh51            Aapple mouse                           80         60.60

Key in the NO. you wish to update: (Press <0> to exit)
7
```

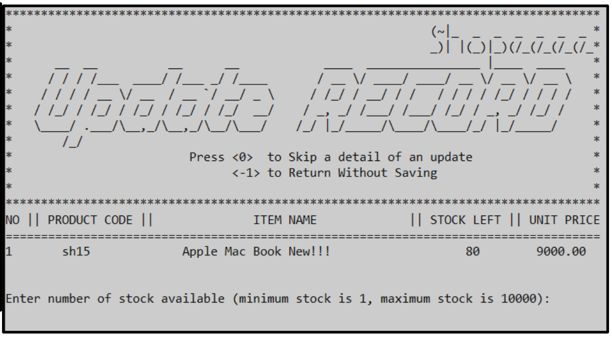


```
Press <0>  to Skip a detail of an update
 <-1> to Return Without Saving

NO || PRODUCT CODE ||              ITEM NAME              || STOCK LEFT || UNIT PRICE
======================================================================================
1       sh15            Apple mac book                         80         9000.00
2       sh1             Apple Mac Book                         1000       8000.00
3       sh37            Huawei MateBook X Pro                  77         4999.00
4       sh38            Huawei MateBook D                      89         2999.00
5       sh39            Huawei MateBook 13                     61         3999.00
6       sh2             Apple Ipad Air                         90         60.00
7       sh3             Apple Ipod                             7888       60.00
8       sh6             Apple Iphone 5C                        50         0.30
9       sh8             Apple Ipad 4                           500        800.99
10      sh14            Apple Pencil                           800        70.67
11      sh40            Huawei MediaPad T5                     101        999.00
12      sh41            Huawei MediaPad M5                     42         1539.00
13      sh42            Huawei MediaPad M5 Pro                 60         1799.00
14      sh16            Huawei TalkBand B5                     168        699.00
15      sh17            Huawei Watch GT                        73         799.00
*****************Above are the 15 product related to your search*******************

Select the NO. of the item you wish to update: (Press <0> to exit)
1
```

When a record is selected user can update the record

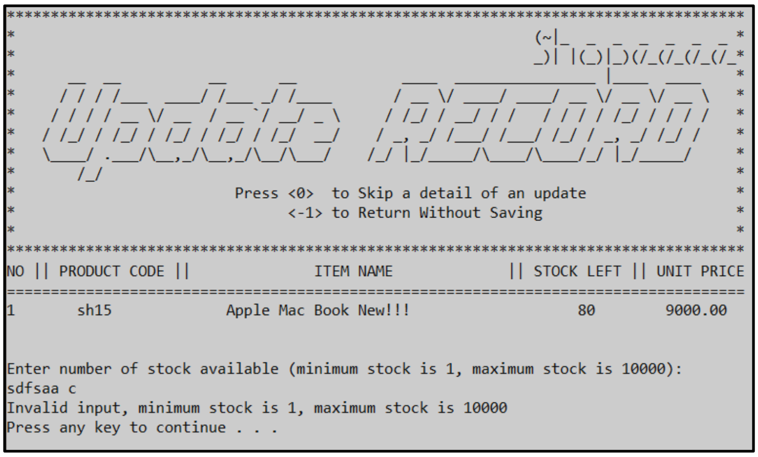


```
Press <0>  to Skip a detail of an update
 <-1> to Return Without Saving

NO || PRODUCT CODE ||              ITEM NAME              || STOCK LEFT || UNIT PRICE
======================================================================================
1       sh15            Apple mac book                         80         9000.00

Enter new product name:
Apple Mac Book New!!!
```



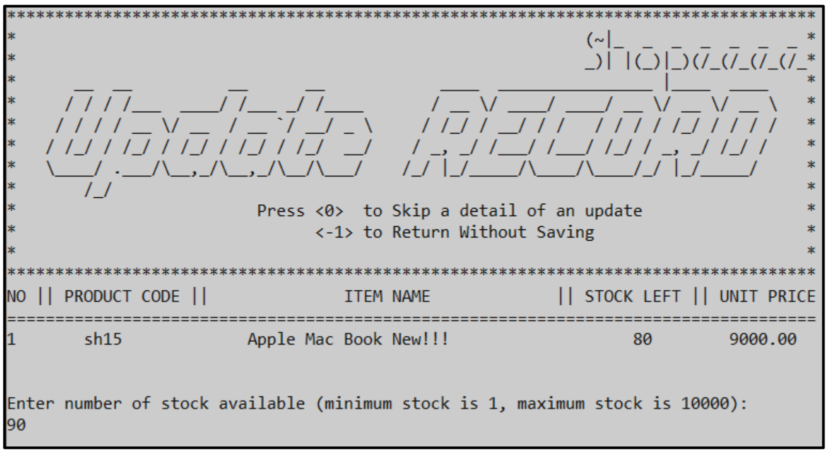
```
Press <0>  to Skip a detail of an update
 <-1> to Return Without Saving

NO || PRODUCT CODE ||              ITEM NAME              || STOCK LEFT || UNIT PRICE
======================================================================================
1       sh15            Apple Mac Book New!!!                  80         9000.00

Enter number of stock available (minimum stock is 1, maximum stock is 10000):
```

System can check the input

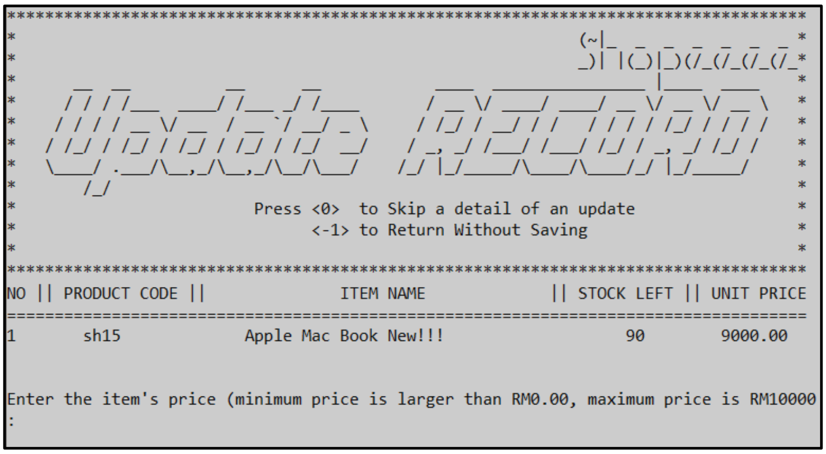


```
Press <0>  to Skip a detail of an update
 <-1> to Return Without Saving

NO || PRODUCT CODE ||              ITEM NAME              || STOCK LEFT || UNIT PRICE
======================================================================================
1       sh15            Apple Mac Book New!!!                  80         9000.00


Enter number of stock available (minimum stock is 1, maximum stock is 10000):
sdfsaa c
Invalid input, minimum stock is 1, maximum stock is 10000
Press any key to continue . . .
```

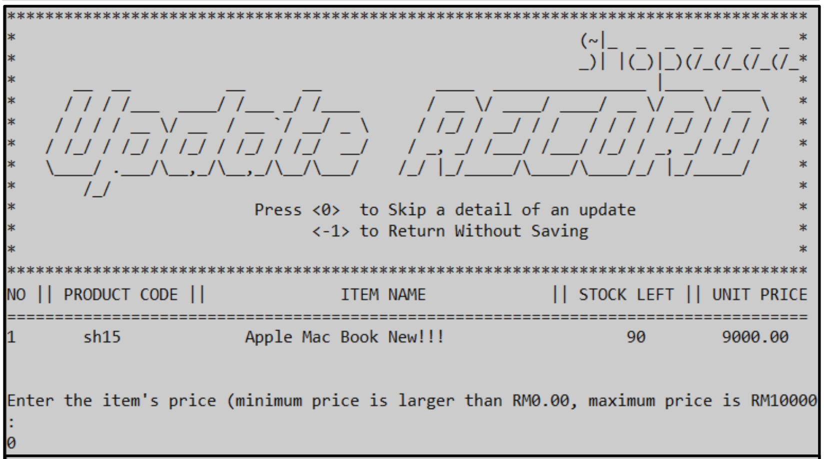
System can check user want to return or treat the input as update new record



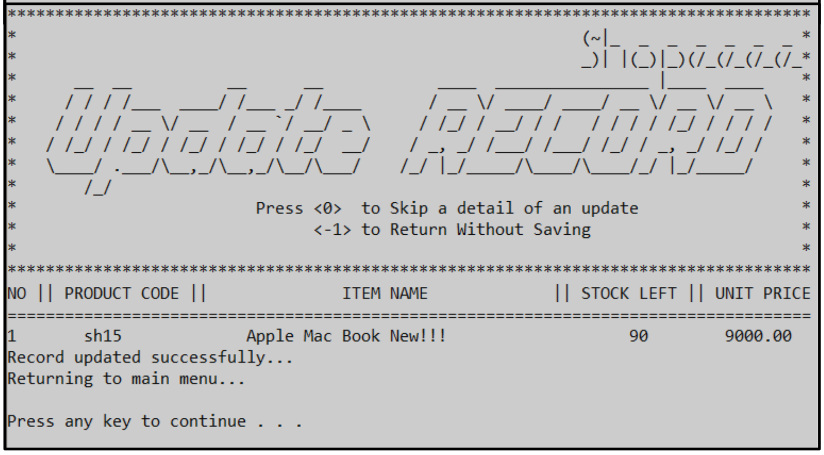
```
***********************************************************************************
*                                                           (~|_  _  _  _  _  _  _ *
*                                                           _)| |(_)|_)(/_(/_(/_(/_*
*      __  __          __      __          ____  _________________|____  ____     *
*     / / / /___  ____/ /___ _/ /____     / __ \/ ____/ ____/ __ \/ __ \/ __ \    *
*    / / / / __ \/ __  / __ `/ __/ _ \   / /_/ / __/ / /   / / / / /_/ / / / /    *
*   / /_/ / /_/ / /_/ / /_/ / /_/  __/  / _, _/ /___/ /___/ /_/ / _, _/ /_/ /     *
*   \____/ .___/\__,_/\__,_/\__/\___/  /_/ |_/_____/\____/\____/_/ |_/_____/      *
*       /_/                                                                       *
*                         Press <0>  to Skip a detail of an update                *
*                               <-1> to Return Without Saving                     *
*                                                                                 *
***********************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
===================================================================================
1       sh15             Apple Mac Book New!!!                    80        9000.00


Enter number of stock available (minimum stock is 1, maximum stock is 10000):
90
```



```
***********************************************************************************
*                                                           (~|_  _  _  _  _  _  _ *
*                                                           _)| |(_)|_)(/_(/_(/_(/_*
*      __  __          __      __          ____  _________________|____  ____     *
*     / / / /___  ____/ /___ _/ /____     / __ \/ ____/ ____/ __ \/ __ \/ __ \    *
*    / / / / __ \/ __  / __ `/ __/ _ \   / /_/ / __/ / /   / / / / /_/ / / / /    *
*   / /_/ / /_/ / /_/ / /_/ / /_/  __/  / _, _/ /___/ /___/ /_/ / _, _/ /_/ /     *
*   \____/ .___/\__,_/\__,_/\__/\___/  /_/ |_/_____/\____/\____/_/ |_/_____/      *
*       /_/                                                                       *
*                         Press <0>  to Skip a detail of an update                *
*                               <-1> to Return Without Saving                     *
*                                                                                 *
***********************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
===================================================================================
1       sh15             Apple Mac Book New!!!                    90        9000.00


Enter the item's price (minimum price is larger than RM0.00, maximum price is RM10000
:
```

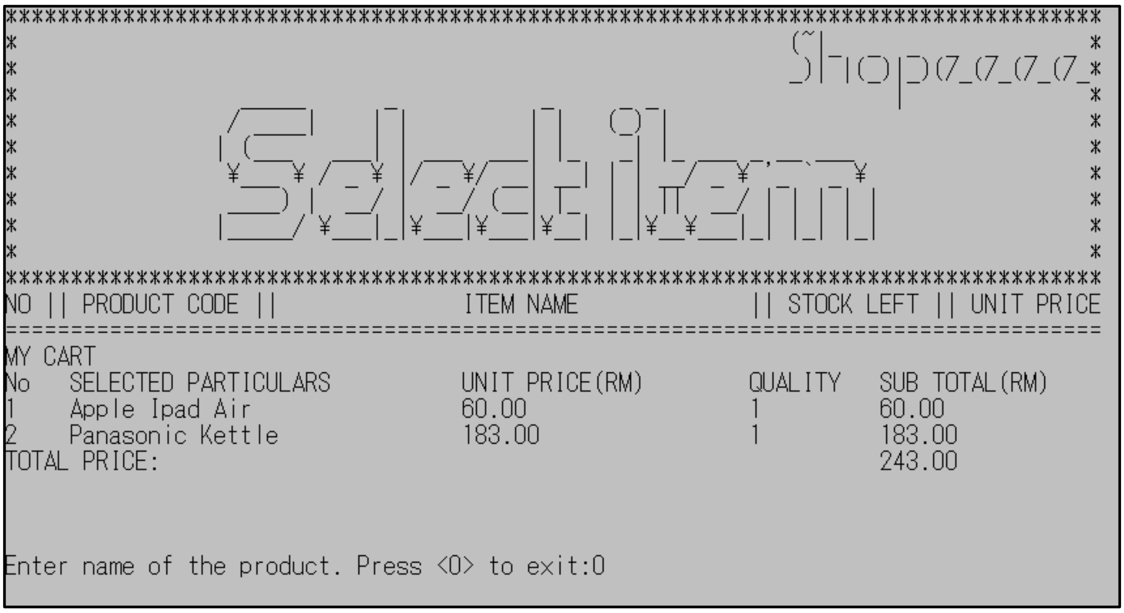


```
***********************************************************************************
*                                                           (~|_  _  _  _  _  _  _ *
*                                                           _)| |(_)|_)(/_(/_(/_(/_*
*      __  __          __      __          ____  _________________|____  ____     *
*     / / / /___  ____/ /___ _/ /____     / __ \/ ____/ ____/ __ \/ __ \/ __ \    *
*    / / / / __ \/ __  / __ `/ __/ _ \   / /_/ / __/ / /   / / / / /_/ / / / /    *
*   / /_/ / /_/ / /_/ / /_/ / /_/  __/  / _, _/ /___/ /___/ /_/ / _, _/ /_/ /     *
*   \____/ .___/\__,_/\__,_/\__/\___/  /_/ |_/_____/\____/\____/_/ |_/_____/      *
*       /_/                                                                       *
*                         Press <0>  to Skip a detail of an update                *
*                               <-1> to Return Without Saving                     *
*                                                                                 *
***********************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
===================================================================================
1       sh15             Apple Mac Book New!!!                    90        9000.00


Enter the item's price (minimum price is larger than RM0.00, maximum price is RM10000
:
0
```

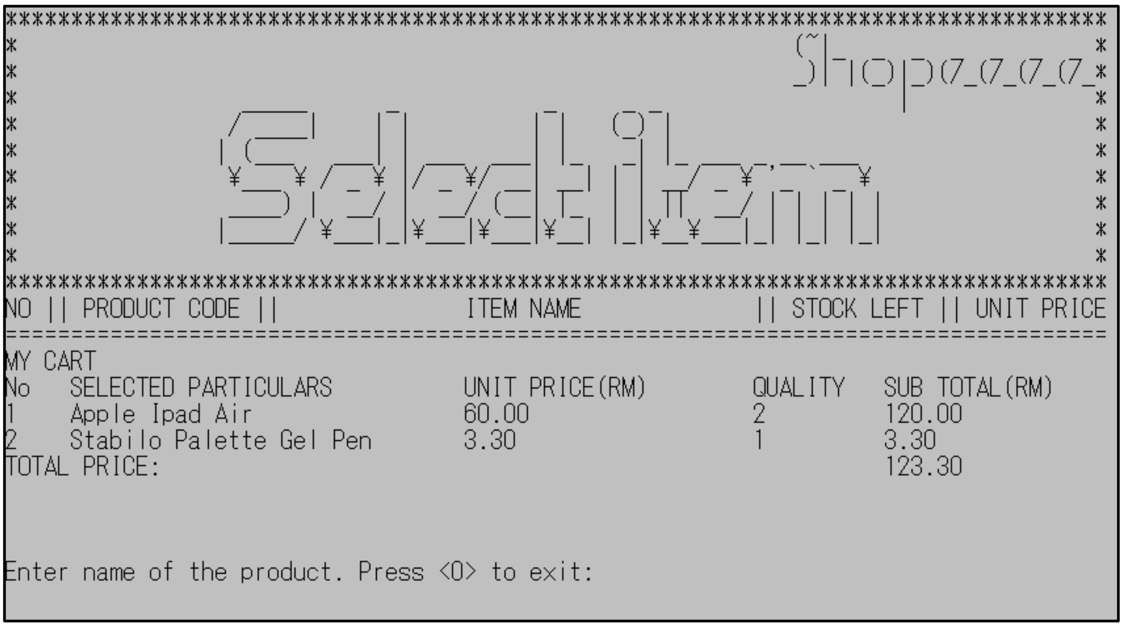


```
***********************************************************************************
*                                                           (~|_  _  _  _  _  _  _ *
*                                                           _)| |(_)|_)(/_(/_(/_(/_*
*      __  __          __      __          ____  _________________|____  ____     *
*     / / / /___  ____/ /___ _/ /____     / __ \/ ____/ ____/ __ \/ __ \/ __ \    *
*    / / / / __ \/ __  / __ `/ __/ _ \   / /_/ / __/ / /   / / / / /_/ / / / /    *
*   / /_/ / /_/ / /_/ / /_/ / /_/  __/  / _, _/ /___/ /___/ /_/ / _, _/ /_/ /     *
*   \____/ .___/\__,_/\__,_/\__/\___/  /_/ |_/_____/\____/\____/_/ |_/_____/      *
*       /_/                                                                       *
*                         Press <0>  to Skip a detail of an update                *
*                               <-1> to Return Without Saving                     *
*                                                                                 *
***********************************************************************************
NO || PRODUCT CODE ||               ITEM NAME               || STOCK LEFT || UNIT PRICE
===================================================================================
1       sh15             Apple Mac Book New!!!                    90        9000.00
Record updated successfully...
Returning to main menu...

Press any key to continue . . .
```

Users are able to add products into cart

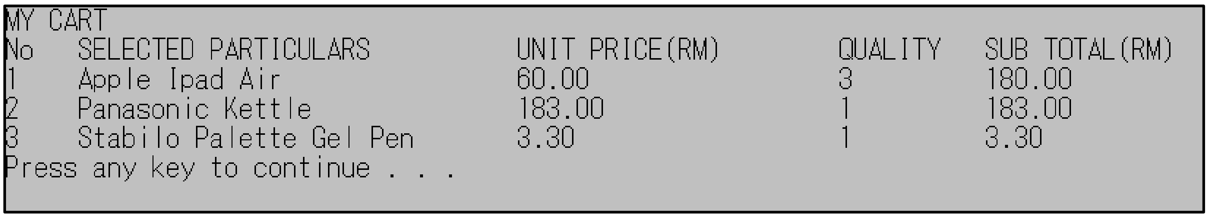


```
************************************************************************************
NO || PRODUCT CODE ||           ITEM NAME          || STOCK LEFT || UNIT PRICE
====================================================================================
MY CART
No   SELECTED PARTICULARS          UNIT PRICE(RM)       QUALITY   SUB TOTAL(RM)
1    Apple Ipad Air                60.00                1         60.00
2    Panasonic Kettle              183.00               1         183.00
TOTAL PRICE:                                                      243.00



Enter name of the product. Press <0> to exit:0
```



```
************************************************************************************
NO || PRODUCT CODE ||           ITEM NAME          || STOCK LEFT || UNIT PRICE
====================================================================================
MY CART
No   SELECTED PARTICULARS          UNIT PRICE(RM)       QUALITY   SUB TOTAL(RM)
1    Apple Ipad Air                60.00                2         120.00
2    Stabilo Palette Gel Pen       3.30                 1         3.30
TOTAL PRICE:                                                      123.30



Enter name of the product. Press <0> to exit:
```

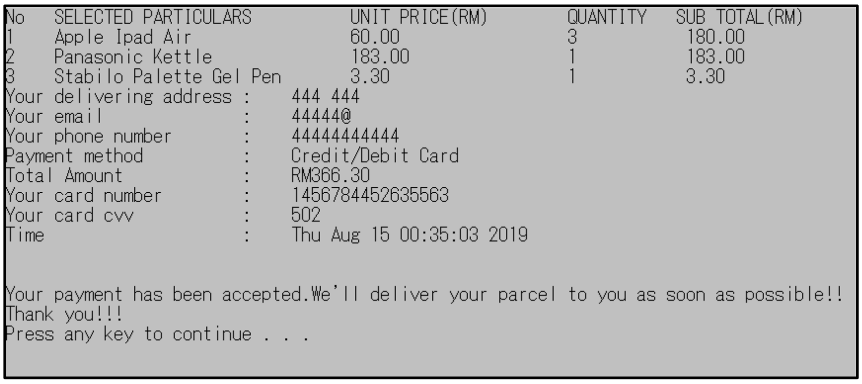
All product added into cart will be totaled up



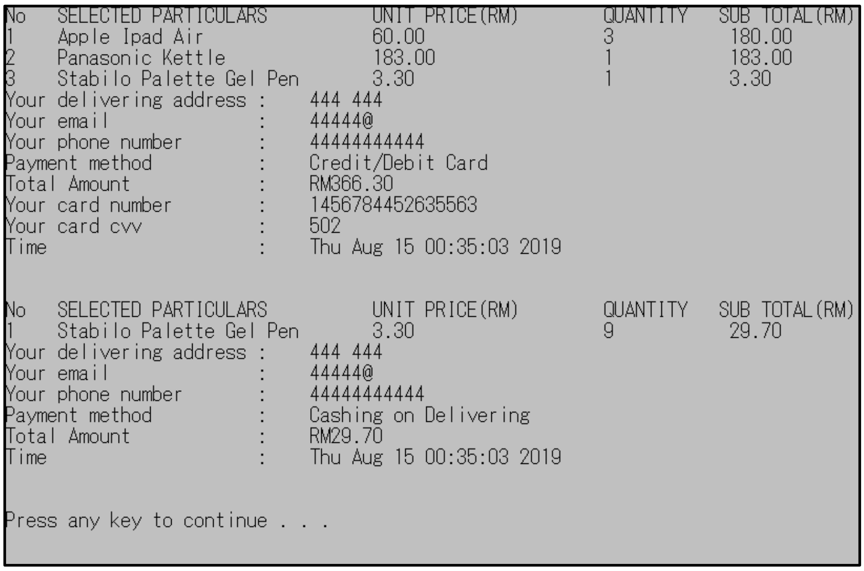
```
MY CART
No   SELECTED PARTICULARS          UNIT PRICE(RM)          QUALITY    SUB TOTAL(RM)
1    Apple Ipad Air                60.00                   3          180.00
2    Panasonic Kettle              183.00                  1          183.00
3    Stabilo Palette Gel Pen       3.30                    1          3.30
Press any key to continue . . .
```

Print invoice after payment and save into users’ payment history

‘

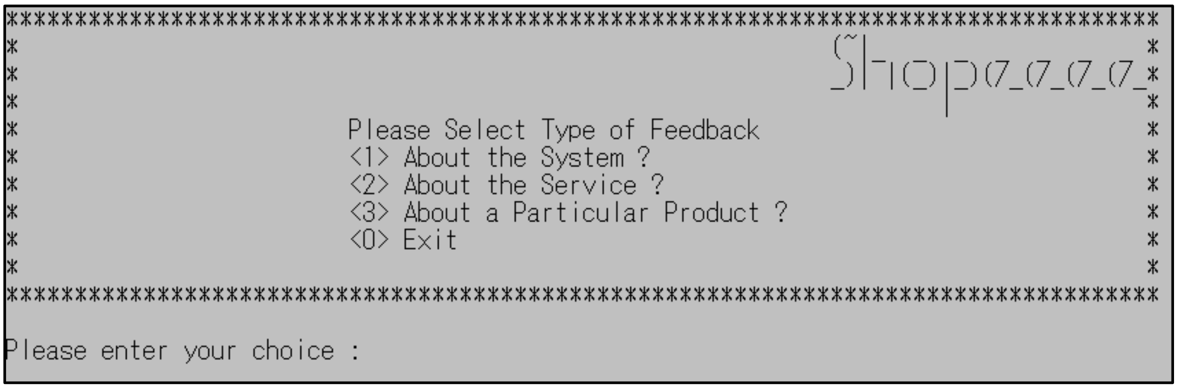


```
No   SELECTED PARTICULARS          UNIT PRICE(RM)       QUANTITY   SUB TOTAL(RM)
1    Apple Ipad Air                60.00                3           180.00
2    Panasonic Kettle              183.00               1           183.00
3    Stabilo Palette Gel Pen       3.30                 1           3.30
Your delivering address :    444 444
Your email              :    44444@
Your phone number       :    44444444444
Payment method          :    Credit/Debit Card
Total Amount            :    RM366.30
Your card number        :    1456784452635563
Your card cvv           :    502
Time                    :    Thu Aug 15 00:35:03 2019


Your payment has been accepted.We'll deliver your parcel to you as soon as possible!!
Thank you!!!
Press any key to continue . . .
```

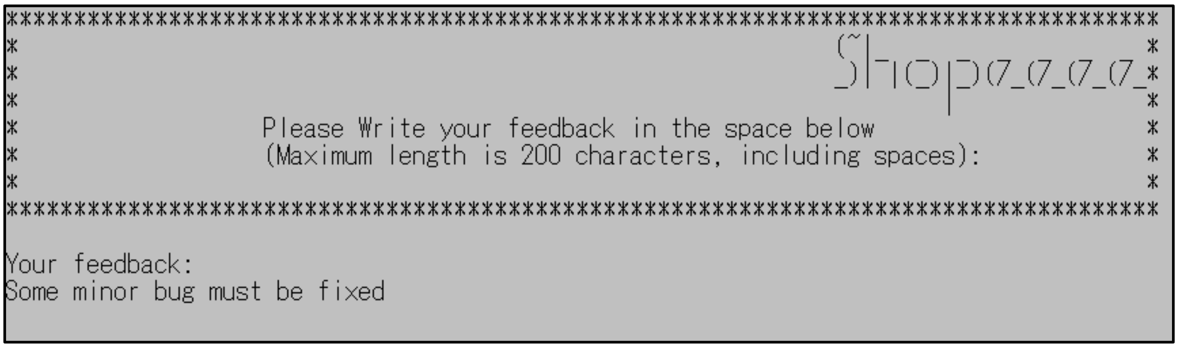


```
No   SELECTED PARTICULARS          UNIT PRICE(RM)       QUANTITY   SUB TOTAL(RM)
1    Apple Ipad Air                60.00                3           180.00
2    Panasonic Kettle              183.00               1           183.00
3    Stabilo Palette Gel Pen       3.30                 1           3.30
Your delivering address :    444 444
Your email              :    44444@
Your phone number       :    44444444444
Payment method          :    Credit/Debit Card
Total Amount            :    RM366.30
Your card number        :    1456784452635563
Your card cvv           :    502
Time                    :    Thu Aug 15 00:35:03 2019


No   SELECTED PARTICULARS          UNIT PRICE(RM)       QUANTITY   SUB TOTAL(RM)
1    Stabilo Palette Gel Pen       3.30                 9           29.70
Your delivering address :    444 444
Your email              :    44444@
Your phone number       :    44444444444
Payment method          :    Cashing on Delivering
Total Amount            :    RM29.70
Time                    :    Thu Aug 15 00:35:03 2019


Press any key to continue . . .
```

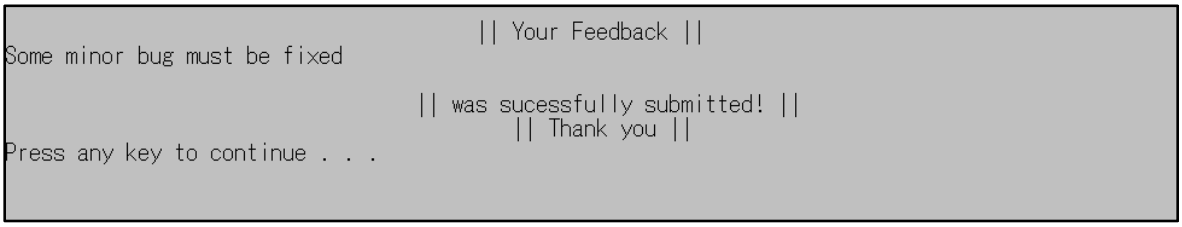
Users can choose to provide different type of feedback an feedback will be saved



```
*************************************************************************************
*                                                            (~|                    *
*                                                            _)|¯|(_)|¯)(7_(7_(7_(7_*
*                                                                    |              *
*                         Please Select Type of Feedback                            *
*                         <1> About the System ?                                    *
*                         <2> About the Service ?                                   *
*                         <3> About a Particular Product ?                          *
*                         <0> Exit                                                  *
*                                                                                   *
*************************************************************************************

Please enter your choice :
```

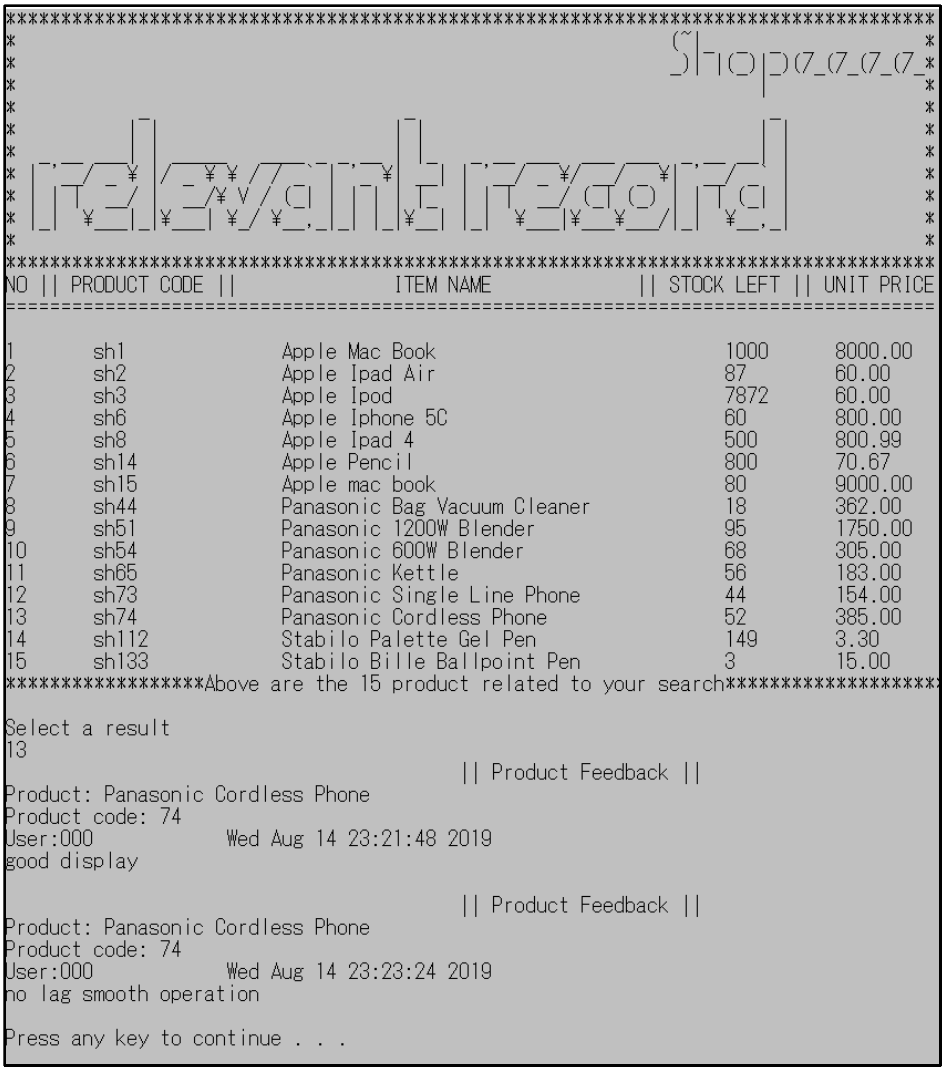


```
*************************************************************************************
*                                                            (~|                    *
*                                                            _)|¯|(_)|¯)(7_(7_(7_(7_*
*                                                                    |              *
*                   Please Write your feedback in the space below                   *
*                   (Maximum length is 200 characters, including spaces):           *
*                                                                                   *
*************************************************************************************

Your feedback:
Some minor bug must be fixed
```

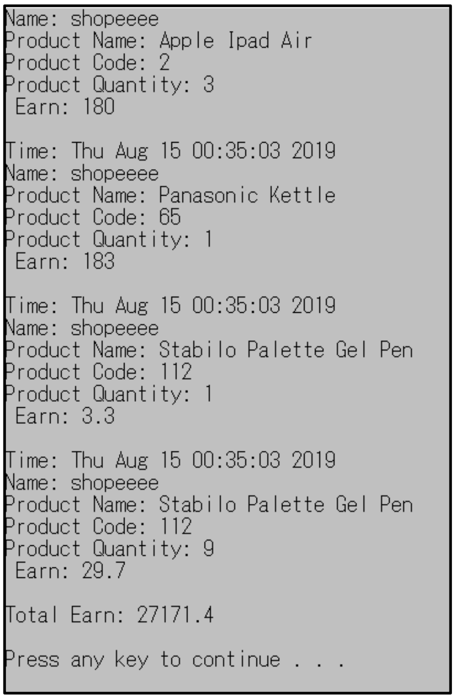


```
                                       || Your Feedback ||
Some minor bug must be fixed

                                  || was sucessfully submitted! ||
                                          || Thank you ||
Press any key to continue . . .
```

Admin can view any feedback received from users and other admins and the selling record of their own product



```
*************************************************************************************
*************************************************************************************
NO || PRODUCT CODE ||              ITEM NAME              || STOCK LEFT || UNIT PRICE
=====================================================================================

1       sh1              Apple Mac Book                          1000       8000.00
2       sh2              Apple Ipad Air                          87         60.00
3       sh3              Apple Ipod                              7872       60.00
4       sh6              Apple Iphone 5C                         60         800.00
5       sh8              Apple Ipad 4                            500        800.99
6       sh14             Apple Pencil                            800        70.67
7       sh15             Apple mac book                          80         9000.00
8       sh44             Panasonic Bag Vacuum Cleaner            18         362.00
9       sh51             Panasonic 1200W Blender                 95         1750.00
10      sh54             Panasonic 600W Blender                  68         305.00
11      sh65             Panasonic Kettle                        56         183.00
12      sh73             Panasonic Single Line Phone             44         154.00
13      sh74             Panasonic Cordless Phone                52         385.00
14      sh112            Stabilo Palette Gel Pen                 149        3.30
15      sh133            Stabilo Bille Ballpoint Pen             3          15.00
******************Above are the 15 product related to your search*******************

Select a result
13
                                         || Product Feedback ||
Product: Panasonic Cordless Phone
Product code: 74
User:000           Wed Aug 14 23:21:48 2019
good display

                                         || Product Feedback ||
Product: Panasonic Cordless Phone
Product code: 74
User:000           Wed Aug 14 23:23:24 2019
no lag smooth operation

Press any key to continue . . .
```



```
Name: shopeeee
Product Name: Apple Ipad Air
Product Code: 2
Product Quantity: 3
 Earn: 180

Time: Thu Aug 15 00:35:03 2019
Name: shopeeee
Product Name: Panasonic Kettle
Product Code: 65
Product Quantity: 1
 Earn: 183

Time: Thu Aug 15 00:35:03 2019
Name: shopeeee
Product Name: Stabilo Palette Gel Pen
Product Code: 112
Product Quantity: 1
 Earn: 3.3

Time: Thu Aug 15 00:35:03 2019
Name: shopeeee
Product Name: Stabilo Palette Gel Pen
Product Code: 112
Product Quantity: 9
 Earn: 29.7

Total Earn: 27171.4

Press any key to continue . . .
```

Source Code

Main menu.cpp

```
#include<iostream>
#include<iomanip>
#include<string>
#include<cctype>
#include<fstream>
#include<Windows.h>
using namespace std;
#include "Feature.h"
#include "Account.h"
#include "Record.h"
#include "Payment.h"
#include "Feedback.h"

void member()
{
	char member_choose, profile_change, update_select, payment_choices = 'y';
	int payment_choice = 0, item,total;
	ifstream file;
	string line;

	do{
		system("cls");
		input_char("Layout/memberpage1-1.txt", obtain.userid, "Layout/memberpage1-2.txt", 
"Selection:", member_choose, 5, ' ', ' ');

		switch (member_choose)
		{
		case '1':
			input_char("Layout/Profile.txt", " ", " ", "Selection:", update_select, 6, '
', ' ');
			update_menu(update_select);
			break;
		case '2':
			payment_choice = select_item();
			if (payment_choice != 1)
				break;
		case '3':
			system("cls");
			if (payment_choice != 1) {
				total = printcart("Cart/" + list.userid + ".txt", item);
				if (total == 0) {
					cout << "Your cart is empty!!\n";
					Sleep(500);
				}
				else
					system("pause");
				break;
			}
		case '4':
			payment_choice = payment();
			if (payment_choice != 3)
				break;
		case '5':
			if (payment_choice != 3){
				system("cls");
```

```
                //print content
                file.open("Invoice/" + obtain.userid + ".txt");
                if (file.is_open())
                    while (!file.eof()) {
                        getline(file, line);
                        cout << line;
                        if (!file.eof())
                            cout << endl;
                    }
                else cout << "No payment history found.Please view our product catalogue 
to start purchase" << endl;
                file.close();
                system("pause");
                break;
            }
        case '0':
            member_choose = '0';
            provide_feedback();
            system("cls");
            cout << "Logging out ..." << endl;
            Sleep(500);
        }
    } while (member_choose != '0');
}

void admin()
{
    char admin_choose, profile_change,update_select;
    do
    {
        input_char("Layout/adminpage1-1.txt", obtain.userid, "Layout/adminpage1-2.txt", 
"Selection:", admin_choose, 4, ' ', ' ');
        switch (admin_choose)
        {
        case '1':
            input_char("Layout/Profile.txt", " ", " ", "Selection:", update_select, 6, '
', ' ');
            update_menu(update_select);
            break;
        case '2':
            record_menu();
            break;
        case '3':
            view_feedback();
            break;
        case '4':
            earning();
            break;
        case '0':
            provide_feedback();
            system("cls");
            cout << "Logging out ..." << endl;
            Sleep(1000);
            return;
        }
    } while (admin_choose != '0');
}

int main()
{
    system("color E0");
    cout << "\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\tWelcome\n\t\t\t\t\t\t   Just a moment..";
    Beep(2000, 100); Beep(1500, 120); Beep(2000, 80); Beep(1500, 100);
    Sleep(2000);
    system("cls");
    system("color 70");
    char menu_input;
    do {
        system("cls");
        input_char("Layout/Menu.txt", " ", " ", "Selection:", menu_input, 3, ' ', ' ');
        switch (menu_input){
        case '1':
            switch (login())
            {
            case 1:
                member();
                break;
            case 2:
                admin();
                break;
            case 0:
                continue;
            }
            break;
        case '2':
            system("cls");
            create();
            break;
        case '0':
            system("cls");
            break;
        }
    } while (menu_input!='0');
    cout << endl<<endl<<endl<<endl<<endl<<endl<< endl << endl << endl << endl<<"
Thank you for using our product" << endl;
    cout << "                                     If you want to feedback please login first"
<< endl << "                                          Contact Group 1 if have any enquiry" <<
endl;
    system("color E0"); Beep(2000, 100); Beep(1500, 120); Beep(2000, 80); Beep(1500, 100);
    cout<<"                                           ";
    system("pause");
    return 0;
}
```

Feature.h

```
void readfile(string filename)
{
    //print content
    ifstream file(filename);
    string line;
    if (file.is_open())
        while (!file.eof()) {
            getline(file, line);
            cout << line;
            if (!file.eof())
                cout << endl;
        }
    else
        cout << "Please include Layout folder for layout printing. Please contact group 1 if 
this message is printed" << endl;
    file.close();
}

int encrypt_system(string enter)
{
    int encrypt = 271828314159;//just make it complex
    char temp;
    for (int i = 0; i < enter.length(); i++) {//loop for encryption of each character by using 
acssi code
        temp = enter.at(i); //encrypt by using the concept of acsii
        encrypt = encrypt * 3 + int(temp);
    }
    return encrypt * 100 + int(temp);
}

void input_char(string file1, string addi, string file2, string prompt, char& var, int indicator, 
char accept1, char accept2)
{
    //indicator=0 mean char accept1, char accept2 is use and if indicator >0 mean accept the 
number. For example indicator =5 can accept the range of char from 0-5
    string object = "0";
    do {
        system("cls");
        if (file1 != " ")
            readfile(file1);
        if (addi != " ")
        {
            cout << "\n*                       User ID: " << left << setw(50) << addi << 
"*" << endl;//if got
            readfile(file2); //if got
        }

        if (object != "0")
            cout << "*                                Error, Invalid input.
*" << "\a" << endl;
        else
            cout << endl;

        cout << prompt << endl;
        getline(cin, object);
        //primary check
        if (object == " ")
            object = "00";
        if (object.length()!= 1)//compare the length is it  is only have 1 char
            continue;
        else
        {
            var = object.at(0);
            if (indicator == 0)
                if (var == accept1 || var == accept2 || int(var) == int(accept1) - 32 ||
int(var) == int(accept2) - 32)
                    return;
                else
                    continue;
            else
                if (int(var) <= indicator + 48)
                    return;
                else
                    continue;
        }
    } while (1);
}

void input_string(string file1, string error, string prompt, string& var, int indicator, char
target, int min, int max)
{
    //indicator 1 means want, 2 mean dont want the target
    char temp;
    do {
        system("cls");
        readfile(file1);

        if (var == "1")
            cout << endl<< error << "\a" << endl;
        else
            cout << endl;
        cout << prompt << endl;
        getline(cin, var);
        //primary check
        if (var == "0")
            return;

        if (var.length() > max || var.length() < min)
            var = "1";
        else
        {
            for (int i = 0; i < var.length(); i++)//loop  each character by using acssi
            {
                temp = var.at(i); //using the concept of acsii and use it to compare

                if (temp == target)
                {
                    if (indicator == 1)
                        return;
                    else if (indicator == 2)
                    {
                        var = "1";
                        break;
                    }
                }
                else if (indicator == 2 && i + 1 == var.length())
                    return;
            }
        }
```

```
	} while (1);
}


int input_integer(string show, int min, int max)
{
	string input;
	char temp;
	int var = 0;
	cout << "\n" << show << "\n";
	getline(cin, input);

	//primary check
	if (input.empty())
		return -2;
	if (input == "0")
		return 0;
	else if (input == "-1")
		return -1;

	for (int i = 0; i < input.length(); i++)//loop  each character by using acssi
	{
		temp = input.at(i); //using the concept of acsii and use it to compare
		if (int(temp) >= 48 && int(temp) <= 57)
			var = var * 10 + int(temp) - 48;
		else
			return -2;
	}
	if (var <= max && var >= min)
		return var;
	else
		return -2;
}

void input_float(string show, float& var, float min, float max)
{
	string input;
	char temp;
	int count = 0;
	var = 0;
	cout << "\n" << show << "\n";
	getline(cin, input);

	if (input == "0")
		return;
	else if (input == "-1")
	{
		var = -1;
		return;
	}


	if ((input.length() == 1 && input==".")||input.empty())
	{
		var = -2;
		return;
	}

	for (int i = 0; i < input.length(); i++)//loop  each character by using acssi
	{
		temp = input.at(i); //using the concept of acsii and use it to compare
```

```
        if (int(temp) < 48 || int(temp) > 57 || temp == '.')
            if (temp == '.')
                count++;
            else
            {
                var = -2;
                return;
            }

        if (count == 2)
        {
            var = -2;
            return;
        }
    }
    var = stod(input);
    if (var > max || var < min)
        var = -2;
    return;
}


void grading_system(string target, int code, string source)
{
    ofstream search("Data/Search_temp.txt", ios::app);
    int value = 0, mark = 0;
    //we use grading system in our searching of the product name
    for (int t = 0; t < target.length(); t++)
    {
        for (int s = 0; s < source.length(); s++)
        {
            value = 0;
            if ((target.at(t) == source.at(s)) || ((int(target.at(t)) - 32 == source.at(s)
|| int(target.at(t)) + 32 == source.at(s)) && !isdigit(target.at(t))))
            {
                if (target.at(t) == 1 || mark > 3) //if the target found is at the
beginning add mark to it
                    value = 2;
                for (int t1 = t, s1 = s; t1 < target.length() && s1 < source.length();
t1++, s1++)
                {
                    if (target.at(t1) == source.at(s1))
                        value = value + 3; //every similar(continuous) just add 3
mark
                    else if ((int(target.at(t1)) - 32 == source.at(s1) ||
int(target.at(t1)) + 32 == source.at(s1)) && !isdigit(target.at(t1)))
                        value = value + 2; //if differ from capital/lowwer level
add 2 mark
                    else break;
                    if (value > mark)//renew
                        mark = value + 1;
                    else if (value >= mark / 2)//got same at after part but not more
than previous
                        mark = mark + 2;
                    if (t==0 && t1 == target.length() - 1)//same as all, add100
                        mark = mark + 100;
                }
            }
        }
    }
    search << mark << " " << code << endl;
```

```
}

void arrange_search()
{
        //arrange data
        ifstream search_t("Data/Search_temp.txt");
        int mark = 0, code = 0, max = 0;
        while (!search_t.eof())
        {
                if (!search_t.eof())
                {
                        search_t >> mark >> code;
                        if (mark > max)
                                max = mark;
                }
        }
        search_t.close();

        ifstream search_tp;
        ofstream search("Data/Search.txt", ios::ate);
        if (max == 0)
                return;
        do {
                search_tp.open("Data/Search_temp.txt");

                while (!search_tp.eof())
                {
                        search_tp >> mark >> code;
                        if (!search_tp.eof())
                        {
                                if (mark == max)
                                {
                                        search << code << endl;
                                }

                        }
                        search_tp.ignore();
                }
                search_tp.close();
                max--;
        } while (max != 0);
        search.close();
}

void updatefile(string file1, string file2)
{
        //replace all data to file 2 from file1
        ifstream f1(file1);
        ofstream f2(file2, ios::ate);
        string line;
        while (!f1.eof())
        {
                getline(f1, line);
                if (!f1.eof())
                        f2 << line << endl;
        }
        f1.close();
        f2.close();
}
```

Account.h

```
struct user {
	string  userid="0", favn="000000000000", email="@@@@@@@@", phone="00000000000", 
address="programming c++";
	int pass, role,favonum;
};
int view_feedback(), provide_feedback(), view_record(), search_record(string name_search), 
select_item(), payment();
void delete_record(), update_record(), add_record();
double printcart(string, int&);
user obtain, list;

void earning()
{
	string time_record, code, name, seller;
	int quantity;
	double unit_price, totalprice = 0, price = 0;
	system("cls");
	ifstream file;
	file.open("Data/history_paid.txt");
	while (!file.eof())
	{
		getline(file, time_record);
		getline(file, seller);
		getline(file, code);
		getline(file, name);
		file >> quantity >> unit_price;
		file.ignore();

		if (seller == obtain.userid && !file.eof())
		{
			price = (quantity * unit_price);
			totalprice = totalprice + price;
			cout << "Time: " << time_record << endl;
			cout << "Name: " << seller << endl;
			cout << "Product Name: " << name << endl;
			cout << "Product Code: " << code << endl;
			cout << "Product Quantity: " << quantity << endl;
			cout << " Earn: " << price << endl << endl;
		}
	}
	file.close();
	cout << "Total Earn: " << totalprice << endl << endl;
	system("pause");
}

void create() {
	user newacc, check;
	//get user id
	do {
		input_string("Layout/createaccountuserid.txt", "\n* 
Error, Invalid username                              *", "User ID:", newacc.userid, 2, ' ', 2, 
30);
		if (newacc.userid == "0") //break this loop when 1 been selected(back to menu)
			return;
		else
		{
			ifstream file("Data/user.txt");
			while (!file.eof())
```

```
            {
                string line;
                getline(file, check.userid);
                getline(file, line); getline(file, line); getline(file, line);
getline(file, line); getline(file, line); getline(file, line);
                if (check.userid == newacc.userid)
                {
                    file.close();
                    cout << "\n*                          Error, User name been
used.                *\n";
                    system("pause");
                    break;
                }
            }
            if (check.userid == newacc.userid)
                continue;
            else
                break;
        }
    } while (1);
    string pass1, pass2;

    //get password
    do {
        system("cls");
        readfile("Layout/createaccountpassword1.txt");
        if (pass1 != pass2)
            cout << "*                           Error, Password Not match
*" << endl;

        cout << "\nPassword:";
        getline(cin, pass1);
        if (pass1 == "0") //break this loop when 0 is selected(back to menu)
            return;

        system("cls");
        readfile("Layout/createaccountpassword2.txt");
        cout << "\nPassword:";
        getline(cin, pass2);
        if (pass2 == "0")//break this loop when 1 is selected(back to menu)
            return;
    } while (pass1 != pass2);//repeat this loop when password keyin not same
    newacc.pass = encrypt_system(pass1);//encrypt the password

    //get acc type
    char rolechoose;
    //let user to create type of account, role==3 mean do not save and return to menu
    input_char("Layout/createaccountrole.txt", " ", " ", "User input", rolechoose, 3, ' ', '
');
    if (rolechoose == '0')
        return;

    //get favorite num
    do {
        system("cls");
        readfile("Layout/createaccountfn.txt");//select role of the account
        if (newacc.favn.length() != 12)
            cout << "*                           Error, Invalid Favorite Number
*" << endl;
        else
            cout << endl;
        cout << "\nFavorite Number :" << endl;
        getline(cin, newacc.favn);
        if (newacc.favn == "0")
            return;
        for (int i = 0; i < newacc.favn.length(); i++)
            if (!isdigit(newacc.favn.at(i)))
                newacc.favn = "0";
    } while (newacc.favn.length() != 12);
    newacc.favonum = encrypt_system(newacc.favn);

    //get email
    input_string("Layout/createaccountemail.txt", "*                           Error, Invalid
email adress                        *", "Email address:", newacc.email, 1, '@', 5, 29);
    if (newacc.email == "0")
        return;

    //get phone
    do {
        system("cls");
        readfile("Layout/createaccountephone.txt");//select role of the account
        if (newacc.phone.length() < 10 || newacc.phone.length() > 13)
            cout << "*                           Error, Invalid phone number
*" << endl;
        else
            cout << endl;
        cout << "\nPhone number:" << endl;
        getline(cin, newacc.phone);
        if (newacc.phone == "0")
            return;
        for (int i = 0; i < newacc.phone.length(); i++)
            if (!isdigit(newacc.phone.at(i)))
                newacc.phone = "0";
    } while (newacc.phone.length() < 10 || newacc.phone.length() > 13);

    //get address
    input_string("Layout/createaccountaddress.txt", "*                           Error, Invalid
address                        *", "Delivery Address:", newacc.address, 1, ' ', 6, 100);
    if (newacc.address == "0")
        return;

    ofstream file("Data/user.txt", ios::app);
    file << newacc.userid << endl << newacc.pass << endl << rolechoose << endl <<
newacc.favonum << endl << newacc.email << endl << newacc.phone << endl << newacc.address<<endl;
    file.close();
    cout << "Creating account" << endl;
    Sleep(1000);
    cout << "Account Successfully Created" << endl;
    cout << "Returning";
    Sleep(1000);
}

int update_menu(char selection)
{
    remove("Data/user_new.txt");
    string pass1, pass2;

    user update = list;
    switch (selection)
    {
    case '1':
        do {
```

```
                  system("cls");
                  readfile("Layout/upaccountpassword1.txt");
                  if (pass1 != pass2)
                        cout << "*                           Error, Password Not match
*" << endl;

                  cout << "\nPassword:";
                  getline(cin, pass1);
                  if (pass1 == "0") //break this loop when 1 is selected(back to menu)
                        return 0;

                  system("cls");
                  readfile("Layout/upaccountpassword2.txt");
                  cout << "\nPassword:";
                  getline(cin, pass2);
                  if (pass2 == "0")//break this loop when 1 is selected(back to menu)
                        return 0;
            } while (pass1 != pass2);//repeat this loop when password keyin not same
            update.pass = encrypt_system(pass1);
            break;

      case '2':
            do {
                  system("cls");
                  readfile("Layout/upaccountfn.txt");//select role of the account
                  if (update.favn.length() != 12)
                        cout << "*                           Error, Invalid Favorite Number
*" << endl;
                  else
                        cout << endl;
                  cout << "\nFavorite number:" << endl;
                  getline(cin, update.favn);
                  if (update.favn == "0")
                        return 0;
                  for (int i = 0; i < update.favn.length(); i++)
                        if (!isdigit(update.favn.at(i)))
                              update.favn = "0";
            } while (update.favn.length() != 12);
            update.favonum = encrypt_system(update.favn);
            break;
      case '3':
            input_string("Layout/upaccountemail.txt", "*                           Error, Invalid
email adress                       *", "Original Email: "+list.email+"\nEmail address:",
update.email, 1, '@', 5, 29);
            if (update.email == "0")
                  return 0;
            break;
      case '4':
            do {
                  system("cls");
                  readfile("Layout/upaccountephone.txt");//select role of the account
                  if (update.phone.length() < 10 || update.phone.length() > 13)
                        cout << "*                           Error, Invalid phone number
*" << endl;
                  else
                        cout << endl;
                  cout << "Original Phone number: " << list.phone << endl;
                  cout << "Phone number:" << endl;
                  getline(cin, update.phone);
                  if (update.phone == "0")
                        return 0;
```

```
                for (int i = 0; i < update.phone.length(); i++)
                    if (!isdigit(update.phone.at(i)))
                        update.phone = "0";
            } while (update.phone.length() < 10 || update.phone.length() > 13);
            break;
        case '5':
            input_string("Layout/upaccountaddress.txt", "*                           Error,
Invalid address                        *", "Original Delivery address: " +list.address+"\nDelivery
Address:", update.address, 1, ' ', 6, 100);
            if (update.address == "0")
                return 0;
            break;
        case '0':
            return 0;
            break;
        }

        ifstream oldacc("Data/user.txt");
        ofstream newacc("Data/user_new.txt");

        while (!oldacc.eof())
        {
            getline(oldacc, list.userid);
            oldacc >> list.pass >> list.role >> list.favonum; oldacc.ignore();
            getline(oldacc, list.email);
            getline(oldacc, list.phone);
            getline(oldacc, list.address);
            if (!oldacc.eof())
                if (list.userid != obtain.userid)
                    newacc << list.userid << endl << list.pass << endl << list.role << endl
<< list.favonum << endl << list.email << endl << list.phone << endl << list.address << endl;
                else
                    newacc << list.userid << endl << update.pass << endl << list.role <<
endl << update.favonum << endl << update.email << endl << update.phone << endl << update.address
<< endl;
        }
        oldacc.close();
        newacc.close();
        list.email = update.email;
        list.phone = update.phone;
        list.address = update.address;
        updatefile("Data/user_new.txt", "Data/user.txt");
        return 1;
}

int login() {
        obtain.userid = "0";
        obtain.favn = "000000000000";
        obtain.phone = "00000000000";

        do {
            system("cls");
            readfile("Layout/Login-username.txt");
            if (obtain.userid == "1")//if looping the second time
                cout << "*                             Error, Username not exist
*";
            cout << "\nUser ID:\n";
            getline(cin ,obtain.userid);
            if (obtain.userid == "0")
                return 0;
            else if (obtain.userid == "1")
```

```
             {
                    create();//create account
                    return 0;
             }
             else if (obtain.userid.length() == 0)
             {
                    obtain.userid = "1";
                    continue;
             }
             //get the detail including encrypted password
             ifstream file("Data/user.txt");
             while (!file.eof())//for vertication of password
                    if(!file.eof())
                    {
                    getline(file, list.userid);
                    file >> list.pass >> obtain.role >> list.favonum; file.ignore();
                    getline(file, list.email);
                    getline(file, list.phone);
                    getline(file, list.address);
                    if (list.userid == obtain.userid)
                    {
                          string  obtain_pass = "0";//string for getting input and it will be
saved after encryption
                          do
                          {
                                 obtain.favn == "0";
                                 system("cls");
                                 readfile("Layout/Login-password1.txt");
                                 cout << "\n*                        Welcome " << left << setw(50)
<< obtain.userid << "*" << endl;
                                 readfile("Layout/Login-password2.txt");

                                 if (obtain_pass == "1")//if second loop
                                       cout << "*                            Error, Invalid
Password                        *" << "\a" << endl;
                                 cout << "\nPassword:\n";
                                 getline(cin, obtain_pass);

                                 if (obtain_pass == "0")
                                       return 0;
                                 else if (obtain_pass == "1")
                                 {
                                       obtain_pass = "0";
                                       obtain.favn == "0";
                                       //recover accout by using favourite num
                                       do {
                                              system("cls");
                                              readfile("Layout/recoverpass.txt");
                                              if (obtain.favn == "1")
                                                    cout << "*                Favorite number
given is not match. Please try again              *" << endl;
                                              else cout << endl;
                                              cout << "Favorite number:\n";
                                              getline(cin, obtain.favn);
                                              if (obtain.favn == "0")
                                                    return 0;
                                              else if (obtain.favn == "1")
                                                    break;
                                              else if (encrypt_system(obtain.favn) ==
list.favonum)//make encryption to compare with encrypted num in the list
                                                    if (update_menu('1') == 0)
```

```
                                                                        break;//return to passlogin
                                                                else
                                                                        return obtain.role;
                                                        else obtain.favn = "1";
                                                } while (1);
                                        }
                                        else if (encrypt_system(obtain_pass) == list.pass)//make
encryption to compare with encrypted pass in the list
                                                return obtain.role;
                                        else
                                                obtain_pass = "1";
                                } while (1);
                                break;
                        }
                }
                file.close();
                //if until here nothing is return it will sure be the username error,so convert
obtain.userid to "1" for printing error message.
                obtain.userid = "1";
        } while (1);
}
```

Record.h

```
struct record
{
	int no = 0, code = 0, stocknum = 0;
	string productname, seller;
	float unitprice = 0;
};

int search_record(string name_search){
	record get, r_list,searching;
	remove("Data/Search_temp.txt");

	//nothing print error
	if (name_search.length() == 0){
		cout<<"******************There are no products matching your
search*******************\n\n";
		system("pause");
		return 0;
	}

	//match the target on a record list's name
	int check=0;
	ifstream file("Data/record_list.txt", ios::in);
	while (!file.eof()){
		check++;
		getline(file, r_list.productname);
		if (r_list.productname.empty()&&check==1){
			cout << "No any item is in this system. Please add record";
			return -1;
		}
		getline(file, r_list.seller);
		file >> r_list.code>> r_list.stocknum>> r_list.unitprice;
		file.ignore();
		if (!file.eof())
			grading_system(name_search, r_list.code, r_list.productname); //give grade to
a record
	}
	file.close();

	arrange_search();//arrange mark from highest to lowest

	//print the result from highest mark to lowest and limit to 15 result only
	ofstream searchr("Data/search_result.txt",ios::ate);
	ifstream search("Data/Search.txt");
	while (!search.eof() && searching.no + 1 <= 15){
		search >> searching.code;
		if (!search.eof()){
			ifstream record("Data/record_list.txt");
			while (!record.eof()){
				getline(record, r_list.productname);
				getline(record, r_list.seller);
				record >> r_list.code >> r_list.stocknum >> r_list.unitprice;
				if (r_list.code == searching.code && r_list.no < 15 && !record.eof()){
					searching.no++;
					cout << setw(8) << searching.no << r_list.seller[0] <<
r_list.seller[1] <<setw(15)<< r_list.code << setw(40) << r_list.productname << setw(10) <<
r_list.stocknum << setw(10) <<fixed<<setprecision(2)<< r_list.unitprice << endl;
					searchr << setw(30) << searching.no << setw(30) <<r_list.code<<
endl;
```

```
                }record.ignore();
            }record.close();
        }search.ignore();
    }searchr.close();

    if (searching.no >= 15) cout << "******************Above are the 15 product related to your
search********************\n";
    else cout << "******************Above are the " << searching.no << " product(s) related to
your search********************\n";
    return searching.no;
}

int view_record() {
    record all;
    ifstream file("Data/record_list.txt");
    //print record
    while (!file.eof())
        if (!file.eof()){
            getline(file, all.productname);
            if (all.no == 0 && all.productname.empty()){
                cout << "No any item is in this system. Please add record" << endl;
                return -1;
            }
            if (!all.productname.empty()) {
                getline(file, all.seller);
                file >> all.code>> all.stocknum >> all.unitprice; file.ignore();
                all.no++;
                cout << setw(8) << all.no << all.seller[0] << all.seller[1] << setw(15)
<< all.code << setw(40) << all.productname << setw(10) << all.stocknum << setw(10) << fixed <<
setprecision(2) << all.unitprice << endl;
            }
        }file.close();
    return all.no;
}

void update_record() {
    record update,searching, r_list;
    string tempname;
    int selectresult, max = 1, tempstock;
    char option;
    double tempprice;
    //determine the way to find out the target to update
    input_char("Layout/Updaterecord.txt", " ", " ", "Press <1> to view entire database\nPress
<2> to search for a particular product.\nPress <0> to exit\n\nSelection:", option, 2, ' ', '
');//selection of a record database

    do{
        if (option == '1'){
            r_list.no = 0;
            do {
                system("cls");
                readfile("Layout/Updaterecord.txt");
                max = view_record();
                if (max == -1)
                    return;
                selectresult=input_integer("Key in the NO. you wish to update: (Press
<0> to exit) ",  0, max);//select the target from result
                if (selectresult == 0)
                    return;
                else if (selectresult < 0)
                {
                    cout << "Invalid input, Please select again." << endl;
                    system("pause");
                }
            } while (selectresult < 0);

            ifstream searchre("Data/record_list.txt");
            while (!searchre.eof()){
                r_list.no++;
                getline(searchre, r_list.productname);
                getline(searchre, r_list.seller);
                searchre >> r_list.code>>r_list.stocknum >> r_list.unitprice;
                searchre.ignore();
                if (selectresult == r_list.no)//get the row of the product list and
break(all the info ald save into the variable)
                    break;
            }searchre.close();
        }else if (option == '2'){
            do{
                system("cls");
                readfile("Layout/Updaterecord.txt");
                cout << "\n\n\nSearch for product: (Press <0> to exit)";
                getline(cin, update.productname);
                if (update.productname == "0")
                    return;
                system("cls");
                readfile("Layout/Updaterecord.txt");
                max = search_record(update.productname);
                if (max == 0){
                    cout << "No results found. Please try again.";
                    Sleep(500);continue;
                }
                else if (max == -1) return;
                else break;
            }while (1);
            do{
                update.no=input_integer("Select the NO. of the item you wish to update:
(Press <0> to exit) ", 0, max);
                if (update.no == 0)
                    return;
                else if (update.no > 0)
                    break;
                else{
                    cout << "Invalid input. Please try again.\n";
                    system("pause");
                    system("cls");
                    readfile("Layout/Updaterecord.txt");
                    search_record(update.productname);
                }
            } while (1);

            //get the product code of the product
            system("cls");
            ifstream searchr("Data/search_result.txt");
            while (!searchr.eof()){
                searchr >> searching.no >> searching.code;
                if (searching.no == update.no)
                    break;
            }
            update.code = searching.code;

            //save the infomation onto the variabe matching the product code
```

```
                        ifstream searchre("Data/record_list.txt");
                        while (!searchre.eof()){
                              getline(searchre, r_list.productname);
                              getline(searchre, r_list.seller);
                              searchre >> r_list.code >> r_list.stocknum >> r_list.unitprice;
                              searchre.ignore();
                              if (update.code == r_list.code)
                                    break;
                        }searchre.close();
                  }else if (option == '0') return;

                  update = r_list;

                  if (update.seller != obtain.userid){
                        system("cls");
                        readfile("Layout/Updaterecord.txt");
                        cout << "You are not authorised to edit other seller content." << endl;
                        system("pause");
                  }
            } while (update.seller != obtain.userid);

            //update product name
            tempname = update.productname;
            do {
                  system("cls");
                  readfile("Layout/Updaterecord.txt");
                  cout << setw(8) << 1 << update.seller[0] << update.seller[1] << setw(15) <<
update.code << setw(40) << tempname << setw(10) << update.stocknum << setw(10) << fixed <<
setprecision(2) << update.unitprice << endl;
                  cout << "\nEnter new product name:" << endl;
                  getline(cin, update.productname);
                  if (update.productname == "-1")
                        return;
                  else if (update.productname == "0")
                  {
                        update.productname = tempname;
                        break;
                  }
                  else if (update.productname.empty()||update.productname.length() > 30)
                  {
                        cout << "Invalid input, length should not empty or greater than 30
character"<<endl;
                        system("pause");
                  }
                  else
                        break;
            } while (1);

            //update stocknum
            tempstock = update.stocknum;
            do {
                  system("cls");
                  readfile("Layout/Updaterecord.txt");
                  cout << setw(8) << 1 << update.seller[0] << update.seller[1] << setw(15) <<
update.code << setw(40) << update.productname << setw(10) << tempstock << setw(10) << fixed <<
setprecision(2) << update.unitprice << endl;
                  update.stocknum = input_integer("\nEnter number of stock available (minimum stock is
1, maximum stock is 10000):", 0, 10000);
                  if (update.stocknum == -1)
                        return;
                  else if (update.stocknum == 0)
```

```
                        update.stocknum = tempstock;
                else if (update.stocknum < 0){
                        cout << "Invalid input, minimum stock is 1, maximum stock is 10000"<<endl;
                        system("pause");
                }
        } while (update.stocknum < 0);

        //update unitprice
        tempprice = update.unitprice;
        do {
                system("cls");
                readfile("Layout/Updaterecord.txt");
                cout << setw(8) << 1 << update.seller[0] << update.seller[1] << setw(15) <<
update.code << setw(40) << update.productname << setw(10) << update.stocknum << setw(10) << fixed
<< setprecision(2) << tempprice << endl;
                 input_float("\nEnter the item's price (minimum price is larger than RM0.00, maximum
price is RM10000: ", update.unitprice , 0, 10000);
                if (update.unitprice == -1)
                        return;
                else if (update.unitprice == 0)
                        update.unitprice = tempprice;
                else if (update.unitprice ==-2){
                        cout << "Invalid input, minimum price is larger than RM0.00, maximum price is
RM10000:"<<endl;
                        system("pause");
                }
        } while (update.unitprice < 0);

        if (update.unitprice != 0){
                system("cls");
                readfile("Layout/Updaterecord.txt");
                cout << setw(8) << 1 << update.seller[0] << update.seller[1] << setw(15) <<
update.code << setw(40) << update.productname << setw(10) << update.stocknum << setw(10) << fixed
<< setprecision(2) << update.unitprice << endl;
        }

        //update account here
        record oldf;
        ofstream newlist("Data/record_n.txt", ios::ate);
        ifstream old("Data/record_list.txt");
        while (!old.eof()){
                getline(old, oldf.productname);
                getline(old, oldf.seller);
                old >> oldf.code >> oldf.stocknum >> oldf.unitprice;
                old.ignore();
                if (!old.eof())
                        if (update.code != oldf.code)
                                newlist << oldf.productname << endl << oldf.seller << endl << oldf.code
<< endl << oldf.stocknum << endl << fixed << setprecision(2) << oldf.unitprice << endl;
                        else
                                newlist << update.productname << endl << update.seller << endl <<
update.code << endl << update.stocknum << endl << fixed << setprecision(2) << update.unitprice <<
endl;
        }old.close();
        newlist.close();

        updatefile("Data/record_n.txt", "Data/record_list.txt");
        cout << "Record updated successfully..." << endl;
        Sleep(100);
        return;
}
```

```
//been call after payment to update the stock which user bought
void update_buy(int code, int quantity) {
      record update, oldf;
      ofstream newlist("Data/record_n.txt");
      ifstream old("Data/record_list.txt");
      while (!old.eof())
      {
            getline(old, oldf.productname);
            getline(old, oldf.seller);
            old >> oldf.code >> oldf.stocknum >> oldf.unitprice;
            old.ignore();

            if (!oldf.productname.empty()) {
                  if (code != oldf.code)
                        newlist << oldf.productname << endl << oldf.seller << endl << oldf.code
<< endl << oldf.stocknum << endl << fixed << setprecision(2) << oldf.unitprice << endl;
                  else
                  {
                        update.stocknum = oldf.stocknum - quantity;
                        newlist << oldf.productname << endl << oldf.seller << endl << oldf.code
<< endl << update.stocknum << endl << fixed << setprecision(2) << oldf.unitprice << endl;
                  }
            }
      }
      old.close();
      newlist.close();
      updatefile("Data/record_n.txt", "Data/record_list.txt");
}

void add_record() {
      record r_list;
      record checking, add[17];
      for (int i = 0; i < 15; i++)
            add[i].seller = obtain.userid;

      system("cls");
      ifstream check("Data/record_list.txt");

      //get the new product code base on the last product code to make sure that no product code
is repeat to easy for other function to locate the target
      while (!check.eof())
            if (!check.eof()){
                  getline(check, checking.productname);
                  getline(check, checking.seller);
                  check >> checking.code >> checking.stocknum >> checking.unitprice;
check.ignore();
            }

      add[0].code = checking.code + 1;
      //to prevent update spam so using for loop to control
      for (int i = 0; i < 15; i++){
            //get product name
            do {
                  system("cls");
                  readfile("Layout/add_record.txt");
                  for (int i = 0; i < 15; i++)
                        if (!add[i].productname.empty() && add[i].productname!="0")
                              cout << setw(8) << i + 1 << add[i].seller[0] << add[i].seller[1]
<< setw(15) << add[i].code << setw(40) << add[i].productname << setw(10) << add[i].stocknum <<
setw(8) << fixed << setprecision(2) << add[i].unitprice << endl;
```

```
                else{
                    cout << setw(8) << i + 1 << add[i].seller[0] << add[i].seller[1]
<< setw(15) << add[i].code << setw(40) << "-------------------------------" << setw(10) << "----"
<< setw(8) << "----" << endl;
                    break;
                }
            cout << "\nEnter product name:";
            getline(cin, add[i].productname);
            if (add[i].productname == "-1")
                return;
            else if (add[i].productname == "0")
                break;
            else if (add[i].productname.empty()||add[i].productname.length() > 30){
                add[i].productname = "0";
                cout << "Invalid input, length should not empty or greater than 30
character"<<endl;
                system("pause");
            }
            else
                break;
        } while (1);
        if (add[i].productname == "0")break;

        //get stock num
        do {
            system("cls");
            readfile("Layout/add_record.txt");
            for (int i = 0; i < 15; i++)
                if (!add[i + 1].productname.empty())
                    cout << setw(8) << i + 1 << add[i].seller[0] << add[i].seller[1]
<< setw(15) << add[i].code << setw(40) << add[i].productname << setw(10) << add[i].stocknum <<
setw(8) << fixed << setprecision(2) << add[i].unitprice << endl;
                else{
                    cout << setw(8) << i + 1 << add[i].seller[0] << add[i].seller[1]
<< setw(15) << add[i].code << setw(40) << add[i].productname << setw(10) << "----" << setw(8) <<
"----" << endl;
                    break;
                }
            add[i].stocknum = input_integer("\nEnter number of stock available (minimum
stock is 1, maximum stock is 10000):", -1, 10000);
            if (add[i].stocknum == -1)
                return;
            else if (add[i].stocknum == 0)
                break;
            else if (add[i].stocknum ==-2){
                cout << "Invalid input, minimum stock is 1, maximum stock is
10000"<<endl;
                system("pause");
            }
        } while (add[i].stocknum == -2);
        if (add[i].stocknum == 0) break;

        //get unit price
        do {
            system("cls");
            readfile("Layout/add_record.txt");
            for (int i = 0; i < 15; i++)
                if (!add[i + 1].productname.empty())
                    cout << setw(8) << i + 1 << add[i].seller[0] << add[i].seller[1]
<< setw(15) << add[i].code << setw(40) << add[i].productname << setw(10) << add[i].stocknum <<
setw(8) << fixed << setprecision(2) << add[i].unitprice << endl;
```

```
                else{
                    cout << setw(8) << i + 1 << add[i].seller[0] << add[i].seller[1]
<< setw(15) << add[i].code << setw(40) << add[i].productname << setw(10) << add[i].stocknum <<
setw(8) << "----" << endl;
                    break;
                }
             input_float("\nEnter the item's price (minimum price is larger than RM0.00,
maximum price is RM10000: ", add[i].unitprice , -1, 10000);
            if (add[i].unitprice == -1)
                return;
            else if (add[i].unitprice == 0)
                break;
            else if (add[i].unitprice ==-2){
                cout << "Invalid input, minimum price is larger than RM0.00, maximum
price is RM10000"<<endl;
                system("pause");
            }
        } while (add[i].unitprice == -2);
        if (add[i].unitprice == 0)
            break;

        add[i + 1].code = add[i].code + 1;
        //when 15product was key in
        if (i == 14 && add[i].unitprice != 0 && add[i].stocknum != 0){
            system("cls");
            readfile("Layout/add_record.txt");
            for (int i = 0; i < 15; i++)
                cout << setw(8) << i + 1 << add[i].seller[0] << add[i].seller[1] <<
setw(15) << add[i].code << setw(40) << add[i].productname << setw(10) << add[i].stocknum <<
setw(8) << fixed << setprecision(2) << add[i].unitprice << endl;
            cout << "Maximum add record is reached(15),it will be automatically save and
return to menu" << endl;
            Sleep(500);
        }
    }

    //print all the message into file
    ofstream file("Data/record_list.txt", ios::app);
    for (int i = 0; i < 15; i++)
        if (!add[i].unitprice == 0)
            file << add[i].productname << endl << add[i].seller << endl << add[i].code <<
endl << add[i].stocknum << endl << fixed << setprecision(2) << add[i].unitprice << endl;
    if (add[0].unitprice != 0)
        cout << "Record added successfully..." << endl;
    cout << "Returning to main menu..." << endl << endl;
    return;
}

void delete_record() {
    record oldr,searchdel;
    char again,option;
    string line;
    int selectresult, max = 0;

    system("cls");
    input_char("Layout/Deleterecord.txt", " ", " ", "Press <1> to view entire database\nPress
<2> to search for a particular product.\nPress <0> to exit\n\nSelection:", option, 2, ' ', ' ');
    //select way to delete

    if (option == '1') {
        do {
            oldr.no = 0;
            system("cls");
            readfile("Layout/Deleterecord.txt");
            max = view_record();
            if (max == -1)
                return;
            selectresult=input_integer("Key in the NO. you wish to delete:", 0, max);
            if (selectresult == 0)
                return;

            //delete data by no copy in into tempfile
            ofstream newlist("Data/record_n.txt", ios::ate);
            ifstream old("Data/record_list.txt");
            while (!old.eof()){
                oldr.no++;
                getline(old, oldr.productname);
                getline(old, oldr.seller);
                old >> oldr.code >> oldr.stocknum >> oldr.unitprice; old.ignore();
                if (!old.eof())
                    if (selectresult != oldr.no)
                        newlist << oldr.productname << endl << oldr.seller << endl
<< oldr.code << endl << oldr.stocknum << endl << fixed << setprecision(2) << oldr.unitprice <<
endl;
                    else if (selectresult == oldr.no && oldr.seller !=
obtain.userid){
                        newlist << oldr.productname << endl << oldr.seller << endl
<< oldr.code << endl << oldr.stocknum << endl << fixed << setprecision(2) << oldr.unitprice <<
endl;
                        cout << "You are not authorised to edit other seller
content." << endl;
                        system("pause");
                    }
            }old.close();newlist.close();
            updatefile("Data/record_n.txt", "Data/record_list.txt");
        } while (1);
    }else if (option == '2'){
        oldr.no = 0;
        do {
            system("cls");
            readfile("Layout/Deleterecord.txt");
            cout << "\n\n\nSearch for record:";
            getline(cin, searchdel.productname);
            system("cls");
            readfile("Layout/Deleterecord.txt");
            if (searchdel.productname == "0")
                return;

            max = search_record(searchdel.productname);
            if (max == 0){
                cout << "No results found!\n";
                system("pause");
                continue;
            }
            else if (max == -1)
                return;
            selectresult=input_integer("Key in the NO. of the product you wish to delete:
", 0, max);
            if (selectresult > 0)
                break;
        } while (1);
        //targeting target
        ifstream searchr("Data/search_result.txt");
        while (!searchr.eof()){
            searchr >> searchdel.no >> searchdel.code >> searchdel.stocknum >>
searchdel.unitprice;
            searchr.ignore();
            if (selectresult == searchdel.no) break;
        }searchr.close();

        //delete data by not writing into temp file
        ofstream newlist("Data/record_n.txt", ios::ate);
        ifstream old("Data/record_list.txt");
        while (!old.eof()){
            getline(old, oldr.productname);
            getline(old, oldr.seller);
            old >> oldr.code >> oldr.stocknum >> oldr.unitprice;
            old.ignore();
            if (!old.eof())
                if (searchdel.code != oldr.code)
                    newlist << oldr.productname << endl << oldr.seller << endl <<
oldr.code << endl << oldr.stocknum << endl << fixed << setprecision(2) << oldr.unitprice << endl;
                else if (searchdel.code == oldr.code && oldr.seller != obtain.userid){
                    newlist << oldr.productname << endl << oldr.seller << endl <<
oldr.code << endl << oldr.stocknum << endl << fixed << setprecision(2) << oldr.unitprice << endl;
                    cout << "You are not authorised to edit other seller content." <<
endl;
                    system("pause");
                }
            }old.close(); newlist.close();
            updatefile("Data/record_n.txt", "Data/record_list.txt");
        }else return;
    return;
}

void record_menu() {
    char recordmenu_input;
    string namesearch;
    do {
        input_char("Layout/Recordmanagement.txt", " ", " ", "Selection:", recordmenu_input,
5, ' ', ' ');
        system("cls");
        switch (recordmenu_input) {
        case '1':
            readfile("Layout/Searchrecord.txt");
            cout << "\n\n\nEnter name of the product. Press <0> to exit:";
            getline(cin, namesearch);
            if (namesearch == "0") break;
            system("cls");
            readfile("Layout/Searchrecord.txt");
            search_record(namesearch);
            system("pause");
            break;
        case '2':
            add_record();
            system("pause");
            break;
        case '3':
            delete_record();
            system("cls");
            readfile("Layout/Deleterecord.txt");
            cout << "Returning to main menu..." << endl << endl;
            system("PAUSE");
            break;
        case '4':
            update_record();
            cout << "Returning to main menu..." << endl << endl;
            system("PAUSE");
            break;
        case '5':
            readfile("Layout/viewrecord.txt");
            view_record();
            system("pause");
            break;
        case '0':
            system("cls");
            return;
            break;
        }
    } while (recordmenu_input != 0);
}
```

Payment.h

```
double printcart(string, int&);
int payment()
{
    double balance = 0, price;
    char paymenttype, confirm, final_decision = 'Y', end = 'Y';
    string  cvv, cardnums, address, input, line, change = "y";
    int final_result, code, quantity, total_item, selected_product = 0, item, selection = 0,
another_add = 0;
    ifstream cart, searchre, receipt;
    fstream temp;
    ofstream finalproduct;
    record product;
    system("cls");
    balance = printcart("Cart/" + list.userid + ".txt", total_item);
    if (balance == 0) {
        system("cls");
        cout << "Your cart is empty!!\n";
        Sleep(500);
        return 0;
    }
    if (balance > 0) {
        system("pause");
        input_char("Layout/payment_confirm.txt", " ", " ", "Input: ", confirm, 0, 'y', 'n');
        if (confirm == 'n' || confirm == 'N') {
            do {
                balance = 0;
                do {
                    system("cls");
                    printcart("Cart/" + list.userid + ".txt", total_item);
                    if (selection < 0 || (selection < 0 && balance <= 0))
                        cout << "Error!!Please input again\n";
                    if (selected_product > 0) {
                        cout << "\nSelected from\n";
                        printcart("Cart/temp.txt", item);
                    }
                    selection = input_integer("Enter the no that you want make a
payment\n(Enter <0> to buy all or finish selecting): ", 0, total_item);
                    if (selection == 0)
                        break;
                    if (selection != -1) {
                        cart.open("Cart/" + list.userid + ".txt");
                        item = 1;
                        while (!cart.eof()) {
                            cart >> code >> quantity;
                            cart.ignore();
                            if (code > 0 && quantity > 0) {
                                searchre.open("Data/record_list.txt");
                                while (!searchre.eof()) {
                                    getline(searchre, product.productname);
                                    getline(searchre, product.seller);
                                    searchre >> product.code >>
product.stocknum >> product.unitprice;

                                    searchre.ignore();
                                    if (product.code == code)
                                        break;
                                }
                                searchre.close();
                                price = quantity * product.unitprice;
                                if (item == selection) {
                                    temp.open("Cart/temp.txt", ios::in |
ios::out | ios::app);

                                    temp << endl << code << endl <<
quantity;

                                    temp.close();
                                    balance += price;
                                    selected_product++;
                                }
                                else {
                                    temp.open("Cart/temp_" + list.userid +
".txt", ios::in | ios::out | ios::app);

                                    temp << endl << code << endl <<
quantity;

                                    temp.close();
                                }
                                item++;
                            }
                        }
                        cart.close();
                    }
                } while (selection != 0);
            } while (selection != 0);
            if (balance > 0)
                input = "Cart/temp.txt";
            else {
                input = "Cart/" + list.userid + ".txt";
                temp.open("Cart/temp_" + list.userid + ".txt", ios::in | ios::out);
                temp << endl;
                temp.close();
            }

        }
        else {
            input = "Cart/" + list.userid + ".txt";
            temp.open("Cart/temp_" + list.userid + ".txt", ios::in | ios::out);
            temp << endl;
            temp.close();
        }

        receipt.open(input);
        finalproduct.open("Cart/finalproduct.txt");
        temp.open("Data/temp_receipt.txt", ios::in | ios::out | ios::app);
        temp << "No   SELECTED PARTICULARS          UNIT PRICE(RM)        QUANTITY   SUB
TOTAL(RM)" << endl;
        item = 1;
        while (!receipt.eof()) {
            receipt >> code >> quantity;
            receipt.ignore();
            if (code > 0 && quantity > 0) {
                searchre.open("Data/record_list.txt");
                while (!searchre.eof()) {
                    getline(searchre, product.productname);
                    getline(searchre, product.seller);
                    searchre >> product.code >> product.stocknum >>
product.unitprice;

                    searchre.ignore();
                    if (product.code == code)
                        break;
                }
                searchre.close();
                price = quantity * product.unitprice;
                temp << left << fixed << setprecision(2);
                temp << setw(5) << fixed << item << setw(30) << product.productname;
                temp << fixed << setw(22) << product.unitprice << fixed << setw(12) <<
quantity;
                temp << fixed << setw(12) << price << endl;
                finalproduct << endl << code << endl << quantity;
                item++;
            }
        }
        temp.close();
        receipt.close();
        finalproduct.close();

        do {
            system("cls");
            readfile("Data/temp_receipt.txt");
            cout << "Your delivering address :    " << list.address << endl<<endl;
            if (another_add <0)
                readfile("Layout/Error.txt");
            another_add = input_integer("Use another address?(1-yes \ 2-no):", 1, 2);
            if (another_add == 1) {
                cout << "Please enter your delivering address:" << endl;
                getline(cin, address);
            }
            else
                address = list.address;
        } while (another_add <0);

        temp.open("Data/temp_receipt.txt", ios::in | ios::out | ios::app);
        temp << "Your delivering address :    " << address << endl;
        temp << "Your email              :    " << list.email << endl;
        temp << "Your phone number       :    " << list.phone << endl;
        temp.close();

        system("cls");
        input_char("Layout/payment_menu.txt", " ", " ", "Input: ", paymenttype, 3, ' ', ' ');

        if (paymenttype == '1'){
            temp.open("Data/temp_receipt.txt", ios::in | ios::out | ios::app);
            temp << "Payment method          :    Credit/Debit Card" << endl;
            temp << "Total Amount            :    RM" << balance << endl;
            temp.close();
            do{
                cardnums = "0000000000000000";
                cvv = "000";
                do {
                    system("cls");
                    readfile("Data/temp_receipt.txt");
                    if (cardnums != "0000000000000000")
                        cout << "Error!!Please input again" << endl;
                    cout << "Please enter your card number\nNo spacing in between:"
<< endl;
                    getline(cin, cardnums);
                    if (cardnums == "0")
                        break;
                    for (int i = 0; i < cardnums.length(); i++)
                        if (!isdigit(cardnums.at(i)))
                            cardnums = "0";
                } while (cardnums.length() != 16);
                if (cardnums == "0")
```

```
                    break;
                do {
                    system("cls");
                    readfile("Data/temp_receipt.txt");
                    if (cvv != "000")
                        cout << "Error!!Please input again" << endl;
                    cout << "Please enter your card CVV:" << endl;
                    getline(cin, cvv);
                    if (cvv == "0")
                        break;
                    for (int i = 0; i < cvv.length(); i++)
                        if (!isdigit(cvv.at(i)))
                            cvv = "0";
                } while (cvv.length() != 3);
                if (cvv == "0")
                    break;
                do {
                    system("cls");
                    readfile("Data/temp_receipt.txt");
                    cout << "Your card number        :    " << cardnums << endl;
                    cout << "Your card cvv           :    " << cvv << endl;
                    if (change != "y" && change != "Y" && change != "n" && change !=
"N")
                        cout << "\nError, Invalid input." << "\a" << endl;
                    cout << "Please make sure that there is no error in your personal
infomation.\nDo you want to change any info you have entered?(Y-yes/N-no)" << endl;
                    getline(cin, change);
                } while (change != "y" && change != "Y" && change != "n" && change !=
"N");
            } while (change == "y" || change == "Y'");
            if (cardnums != "0" && cvv != "0") {
                paymenttype = '1';
                temp.open("Data/temp_receipt.txt", ios::in | ios::out | ios::app);
                temp << "Your card number        :    " << cardnums << endl;
                temp << "Your card cvv           :    " << cvv << endl;
                temp << "Time                    :    " << __TIMESTAMP__ << endl <<
endl;
                temp.close();
            }
            else
                paymenttype = '3';
        }
        else if (paymenttype == '2') {
            temp.open("Data/temp_receipt.txt", ios::in | ios::out | ios::app);
            temp << "Payment method          :    Cashing on Delivering" << endl;
            temp << "Total Amount            :    RM" << balance << endl;
            temp << "Time                    :    " << __TIMESTAMP__ << endl << endl;
            temp.close();
        }

        if (paymenttype != '3') {
            input_char("Data/temp_receipt.txt", " ", " ", "Are you confirm to make this
payment?(Y-yes \ N-no):", final_decision, 0, 'y', 'n');

            if (final_decision == 'n' || final_decision == 'N') {
                temp.open("Data/temp_receipt.txt");
                temp.close();
                final_result = 0;
            }
            else {
                system("cls");
                readfile("Data/temp_receipt.txt");
                if (paymenttype == '1')
                    cout << "\nYour payment has been accepted.We'll deliver your
parcel to you as soon as possible!!" << endl;
                else
                    cout << "\nYour payment will be made when you receive the
parcel.We'll deliver your parcel to you as soon as possible!!" << endl;
                cout << "Thank you!!!\n";
                system("pause");
                fstream file("Invoice/" + list.userid + ".txt", ios::in | ios::out |
ios::app);
                fstream temp("Data/temp_receipt.txt", ios::in | ios::out | ios::app);
                while (temp.good()) {
                    getline(temp, line);
                    file << line << endl;
                }
                temp.close();
                file.close();
                code = 0;
                ofstream file1("Cart/" + list.userid + ".txt");
                ifstream temp1("Cart/temp_" + list.userid + ".txt");
                while (!temp1.eof()) {
                    temp1 >> code;
                    temp1.ignore();
                    if (code == 0)
                        break;
                    file1 << endl << code;
                }
                temp1.close();
                file1.close();
                receipt.open("Cart/finalproduct.txt");
                temp.open("Data/history_paid.txt", ios::in | ios::out | ios::app);
                while (!receipt.eof()) {
                    receipt >> code >> quantity;
                    if (code > 0 && quantity > 0) {
                        receipt.ignore();
                        update_buy(code, quantity);
                        searchre.open("Data/record_list.txt");
                        while (!searchre.eof()) {
                            getline(searchre, product.productname);
                            getline(searchre, product.seller);
                            searchre >> product.code >> product.stocknum >>
product.unitprice;

                            searchre.ignore();
                            if (product.code == code)
                                break;
                        }
                        searchre.close();
                        temp << __TIMESTAMP__ << endl << product.seller << endl <<
code << endl << product.productname << endl << quantity << endl << product.unitprice << endl;
                    }
                }
                temp.close();
                receipt.close();

            }
        }
    }
    input_char("Layout/payment_end.txt", " ", " ", "Input: ", end, 0, 'y', 'n');
    ofstream clear;
    clear.open("Cart/temp_" + list.userid + ".txt");
    clear.close();
    clear.open("Cart/temp.txt");
    clear.close();
    clear.open("Data/temp_receipt.txt");
    clear.close();
    if ((end == 'y' || end == 'Y') && (final_decision == 'y' || final_decision == 'Y'))
        final_result = 2;
    else
        if ((end == 'y' || end == 'Y') && (final_decision == 'n' || final_decision == 'N'))
            final_result = 0;
        else
            final_result = 3;
    return final_result;
}

int select_item()
{
    cout << left;
    double price,total_price=0;
    char returning, savecart;
    ifstream searchr, searchre;
    fstream temp;
    record product;
    product.productname = "0";
    int selectresult, max = 1,payment_choice = 0, total_item = 1, quantity=1;
    do {
        system("cls");
        do {
            system("cls");
            readfile("Layout/Select_item.txt");
            if (total_item > 1)
                total_price = printcart("Data/temp.txt",total_item);
            cout << setw(67) << "TOTAL PRICE:" << setw(12) << total_price << endl;
            cout << "\n\n\nEnter name of the product. Press <0> to exit:";
            getline(cin, product.productname);
            if (product.productname == "0")
                break;
            system("cls");
            readfile("Layout/Select_item.txt");
            max = search_record(product.productname);
            if (total_item > 1)
                total_price = printcart("Data/temp.txt",total_item);
            cout << endl;
            cout << setw(67) << "TOTAL PRICE:" << setw(12) << total_price << endl;
            if (max == 0) {
                cout << "\nNo result found"<<endl;
                Sleep(500);
                continue;
            }
            else if (max == -1)
            {
                cout << "\nNo record found"<<endl;
                Sleep(500);
                continue;
            }
            selectresult=input_integer("Select a result", 0, max);
            if (selectresult > 0)
                break;
            else {
                cout << "\nInvalid result is entered!!\nPlease try again!!";
                Sleep(500);
            }
        } while (1);

        if (product.productname == "0")
            break;
        searchr.open("Data/search_result.txt");
        while (!searchr.eof()) {
            searchr >> product.no >> product.code;
            if (selectresult == product.no)
                break;
        }
        selectresult = product.code;

        searchre.open("Data/record_list.txt");
        while (!searchre.eof()) {
            getline(searchre, product.productname);
            getline(searchre, product.seller);
            searchre >> product.code >> product.stocknum >> product.unitprice;
            searchre.ignore();
            if (selectresult == product.code)
                break;
        }
        searchr.close();
        searchre.close();

        string input = "Enter the quantity of item that want buy\n(not exceeding the 
available stock or enter <0> to exit):\n";
        do {
            if(quantity<0)
                cout << "\nError!!Please input again";
            cout << "Stock: " << product.stocknum;
            quantity=input_integer(input, 0, product.stocknum);
            if (quantity == 0)
                break;
        } while (quantity <0);
        if (quantity > 0) {
            price = quantity * product.unitprice;
            total_price += price;
            temp.open("Data/temp.txt", ios::in | ios::out | ios::app);
            temp << left << fixed << setprecision(2);
            temp << endl << product.code << endl << quantity;
            temp.close();
            total_item++;
        }
    } while (1);

    if (total_price > 0) {
        input_char("Layout/savingcart.txt", " ", " ", "Your decision: ", savecart, 0, 'y', 
'n');
    }
    else
        savecart = 'n';
    input_char("Layout/select_end.txt", " ", " ", "Selection: ", returning, 0, 'm', 'p');
    if (savecart == 'y' || savecart == 'Y') {
        fstream savecart;
        savecart.open("Cart/" + list.userid + ".txt", ios::in | ios::out | ios::app);
        ifstream cart;
        cart.open("Data/temp.txt");
        while (!cart.eof()) {
            cart >> product.code >> quantity;
            if (product.code > 0 && quantity > 0)
```

```
                savecart << endl << product.code << endl << quantity;
        }
        cart.close();
        savecart.close();
    }
    ofstream cart;
    cart.open("Data/temp.txt");
    cart.close();
    if (returning == 'p' || returning == 'P')
        payment_choice = 1;
    else
        payment_choice = 0;
    return payment_choice;
}

double printcart(string file,int &total_item)//printcart("Cart/" + list.userid + ".txt", 
total_item);
{
    double subtotal=0;
    ifstream cart, searchre;
    ofstream newcart;
    int code[1000] = {-1}, quantity[1000],i=0,k;
    double price;
    record product;
    total_item = 1;
    cart.open(file);
    newcart.open("Cart/" + list.userid + "_temp.txt");
    if (cart.is_open()) {
        while (!cart.eof()) {
            cart >> code[i] >> quantity[i];
                cart.ignore();
            if (code[i] > 0 && quantity[i] > 0) {
                i++;
            }
        }
        for (int j = 0; j < i; j++) {
            for (k = 1; k < i-j; k++) {
                if (code[j] == code[j + k]) {
                    quantity[j] += quantity[j + k];
                    code[j + k] = 0;
                }
            }
        }
        for (int j = 0; j < i; j++) {
            if (code[j] != 0) {
                newcart << endl << code[j] << endl << quantity[j];
            }
        }
    }
    cart.close();
    newcart.close();
    cart.open("Cart/" + list.userid + "_temp.txt");
    newcart.open(file);
    while(!cart.eof()){
        cart >> code[0] >> quantity[0];
        cart.ignore();
        if (code[0] > 0 && quantity[0] > 0) {
            searchre.open("Data/record_list.txt");
            while (!searchre.eof()) {
                getline(searchre, product.productname);
                getline(searchre, product.seller);
                searchre >> product.code >> product.stocknum >> product.unitprice;
                searchre.ignore();
                if (product.code == code[0])
                    break;
            }
            searchre.close();
            if (quantity[0] <= product.stocknum)
                newcart << endl << code[0] << endl << quantity[0];
            else
                newcart << endl << code[0] << endl << product.stocknum;
        }
    }
    cart.close();
    newcart.close();
    cart.open(file);
    readfile("Layout/cart.txt");
    if (cart.is_open()) {
        while (!cart.eof()) {
            cart >> code[0] >> quantity[0];
            cart.ignore();
            if (code[0] > 0 && quantity[0] > 0) {
                searchre.open("Data/record_list.txt");
                while (!searchre.eof()) {
                    getline(searchre, product.productname);
                    getline(searchre, product.seller);
                    searchre >> product.code >> product.stocknum >>
product.unitprice;

                    searchre.ignore();
                    if (product.code == code[0])
                        break;
                }
                searchre.close();
                price = quantity[0] * product.unitprice;
                cout << left << fixed << setprecision(2);
                cout << setw(5) << fixed << total_item << setw(30) <<
product.productname;
                cout << fixed << setw(22) << product.unitprice << fixed << setw(10) <<
quantity[0];
                cout << fixed << setw(12) << price << endl;
                subtotal += price;
                total_item++;
            }
        }
    }
    cart.close();
    return subtotal;
}
```

Feedback.h

```
int provide_feedback()//get user id from previous processes
{
	char feedbackoption, feedbacktype;//to choose whether to give feedback or not
	string feedback="0";
	fstream testfeedback;
	int selectresult, max = 1;
	record product;
	ifstream searchr, searchre;
	system("cls");
	input_char("Layout/feedback_menu.txt", " ", " ", "Please enter your choice : ",
feedbackoption, 1, ' ', ' ');
	system("cls");
	if (feedbackoption == '1'){
		input_char("Layout/feedbackoption.txt", " ", " ", "Please enter your choice : ",
feedbacktype, 3, ' ', ' ');
		switch (feedbacktype){
		case '1':
			input_string("Layout/give_feedback.txt", "\nYour word count has exceed the
limit!\nPlease re-enter again.", "Your feedback:", feedback, 2, '@', 4, 200);
			if (feedback == "0")
				break;
			testfeedback.open("Data/system_feedback.txt", ios::in | ios::out | ios::app);
			testfeedback << "User:" << left << setw(12) << list.userid << "   " <<
__TIMESTAMP__ << endl;
			break;
		case '2':
			input_string("Layout/give_feedback.txt", "\nYour word count has exceed the
limit!\nPlease re-enter again.", "Your feedback:", feedback, 2, '@', 4, 200);
			if (feedback == "0")
				break;
			testfeedback.open("Data/service_feedback.txt", ios::in | ios::out | ios::app);
			testfeedback << "User:" << left << setw(12) << list.userid << "   " <<
__TIMESTAMP__ << endl;
			break;
		case '3':
			do {
				system("cls");
				readfile("Layout/feedback_search.txt");
				cout << "\n\n\nEnter name of the product. Press <0> to exit:";
				getline(cin, product.productname);
				system("cls");
				readfile("Layout/feedback_search.txt");
				max = search_record(product.productname);
				if (max == 0){
					cout << "No result found";
					Sleep(500);
					continue;
				}
				else if (max == -1) {
					cout << "\nNo record found" << endl;
					Sleep(500);
					continue;
				}
				selectresult=input_integer("Select a result", 0, max);
				if (selectresult > 0)
					break;
				else {
					cout << "\nInvalid result is entered!!\nPlease try again!!";
```

```
                    Sleep(500);
                }
            } while (1);
            searchr.open("Data/search_result.txt");
            while (!searchr.eof()){
                searchr >> product.no >> product.code;
                if (selectresult == product.no)
                    break;
            }
            selectresult = product.code;
            searchre.open("Data/record_list.txt");
            while (!searchre.eof()){
                getline(searchre, product.productname);
                getline(searchre, product.seller);
                searchre >> product.code >> product.stocknum >> product.unitprice;
                searchre.ignore();
                if (selectresult == product.code)
                    break;
            }
            searchr.close();
            input_string("Layout/give_feedback.txt", "\nYour word cout has exceed the 
limit!\nPlease re-enter again.", "Your feedback:", feedback, 2, '@', 4, 200);
            if (feedback == "0")
                break;
            testfeedback.open("Data/product_feedback.txt", ios::in | ios::out | ios::app);
            testfeedback << product.productname << endl << product.code << endl;
            testfeedback << "User:" << left << setw(12) << list.userid << "   " << 
__TIMESTAMP__ << endl;
            break;
        }
        if (feedback != "0") {
            testfeedback << left << setw(200) << feedback<<endl;
            testfeedback.close();
            system("cls");
            cout << "\n                                       || Your Feedback ||" << 
endl;
            cout << left << setw(200) << feedback << endl;
            cout << "                                  || was sucessfully submitted! ||" 
<< endl;
            cout << "                                          || Thank you ||" << endl;
            system("pause");
        }
    }
    system("cls");
    readfile("Layout/feedback_end.txt");
    Sleep(1000);
    return 0;
}

int view_feedback(){
    //page to view feedback
    //get admin id from previous processes
    ifstream searchr, searchre,view_feedback;
    record product;
    char feedbacktype;
    string feedback="0",name,detail,code_num;
    int selectresult=1, max = 1,noFeedback=0;
    input_char("Layout/feedback_header.txt", " ", " ", "Please enter your choice : ", 
feedbacktype, 3, ' ', ' ');
    switch (feedbacktype) {
    case '1':
        view_feedback.open("Data/system_feedback.txt");
        if (view_feedback.is_open()) {
            cout << "                                        || System Feedback ||" <<
endl;
            while (!view_feedback.eof()){
                getline(view_feedback, feedback);
                cout << feedback << endl;
            }
        }
        view_feedback.close();
        break;
    case '2':
        view_feedback.open("Data/service_feedback.txt");
        if (view_feedback.is_open()) {
            cout << "                                        || Service Feedback ||" <<
endl;
            while (!view_feedback.eof()){
                getline(view_feedback, feedback);
                cout << feedback << endl;
            }
        }
        view_feedback.close();
        break;
    case '3':
        do {
            do {
                feedback = "0";
                noFeedback = 0;
                system("cls");
                readfile("Layout/feedback_search.txt");
                cout << "\n\n\nEnter name of the product. Press <0> to exit:";
                getline(cin, product.productname);
                if (product.productname == "0")
                    break;
                system("cls");
                readfile("Layout/feedback_search.txt");
                max = search_record(product.productname);
                if (max == 0) {
                    cout << "\nNo result found" << endl;
                    Sleep(500);
                    continue;
                }
                else if (max == -1) {
                    cout << "\nNo record found" << endl;
                    Sleep(500);
                    continue;
                }
                selectresult = input_integer("Select a result", 0, max);
                if (selectresult >= 0)
                    break;
                else {
                    cout << "\nInvalid result is entered!!\nPlease try again!!";
                    Sleep(500);
                }
            } while (1);
            if (selectresult == 0|| product.productname == "0")
                break;
            searchr.open("Data/search_result.txt");
            while (!searchr.eof()) {
                searchr >> product.no >> product.code;
                if (selectresult == product.no)
```

```
                    break;
            }
            selectresult = product.code;
            searchre.open("Data/record_list.txt");
            while (!searchre.eof()) {
                getline(searchre, product.productname);
                getline(searchre, product.seller);
                searchre >> product.code >> product.stocknum >> product.unitprice;
                searchre.ignore();
                if (selectresult == product.code)
                    break;
            }
            searchre.close();
            searchr.close();
            view_feedback.open("Data/product_feedback.txt");
            if (view_feedback.is_open()) {
                while (!view_feedback.eof()) {
                    getline(view_feedback, name);
                    getline(view_feedback, code_num);
                    getline(view_feedback, detail);
                    getline(view_feedback, feedback);
                    if (name == product.productname) {
                        cout << "                                          ||
Product Feedback ||" << endl;
                        cout << "Product: " << product.productname << endl <<
"Product code: " << code_num << endl;
                        cout << detail << endl << feedback << endl;
                        noFeedback++;
                    }
                }
            }
            if (noFeedback == 0) {
                cout << "No result found!!\n";
                Sleep(1000);
            }
            else
                system("pause");
            view_feedback.close();
        } while (1);
        break;
    }
    if (feedback != "0") {
        cout << "                           || Above are the feedbacks provided by users ||"
<< endl << endl;
        system("pause");
    }
    return 0;
}
```