KOLEJ UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Assignment

BACS1113 Computer Organisation and Architecture

2020/2021

Student’s name/ ID Number : LEE SHU ERN

Student’s name/ ID Number : HEE SZE WEI

Student’s name/ ID Number : CHEW HWA ERN

Programme : RMM2

Tutorial Group : G1

Date of Submission to Tutor : 12 SEPTEMBER 2020

Content

System Title : Wee Cafe - Food Ordering Kiosk

User’s Guide :

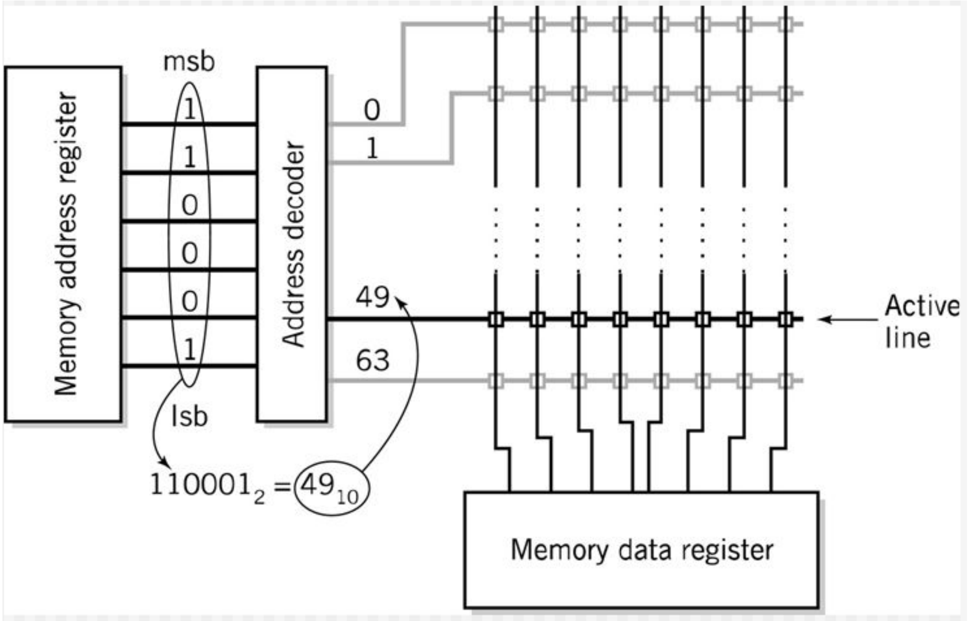
This food ordering kiosk is created to allow both staff and customers to place orders easily and with efficiency. This includes presenting the users with the cafe menu within the kiosk and also allowing users to place orders, view receipts and perform payments. To begin with the kiosk system, Logo is displayed followed by a list of options of actions that the user wishes to proceed with. The user is then required to enter an option input. If the user wishes to place an order, the menu is displayed and the user will be prompted with selections. After so, the kiosk system generates an invoice of the order details and asks the user to choose their preferred payment methods. The user will have to enter the amount paid and Receipt will then be generated. Only users who are out shop customers can make ratings based on their experience at the Wee Cafe which will then be displayed as a percentage rating of the cafe in the daily report. Last but not least, if there are no longer new customers or a staff wishes to login, the kiosk will prompt a question to make sure the user is a staff and then allow the user to enter staff ID and password. The system encrypts the password to ensure security and matches it with the encrypted password in the program. If staff ID and password are valid and matches, the staff is successfully logged in and allowed access to the report. If the staff fails to login for more than 3 attempts, the system freezes and delays for 1minute before the next prompt.

Business Assumptions:

1. The Wee cafe’s selling items mainly consist of cakes and drinks.
2. This kiosk can be used by both customers and staff.
3. Purchases are limited to 5 order items per customer.
4. Assume only a maximum of 100 purchases can be made per day.
5. Report is generated on a daily basis.
6. Staff have 3 login attempts. 3 failed attempts prompts a 1 minute delay.
7. Reports can only be accessed by staff with a successful login.
8. Only our shop customers can rate make ratings based on their experience at the WEE cafe.
9. Ratings can only be done if there are at least 1 or more previous order records.

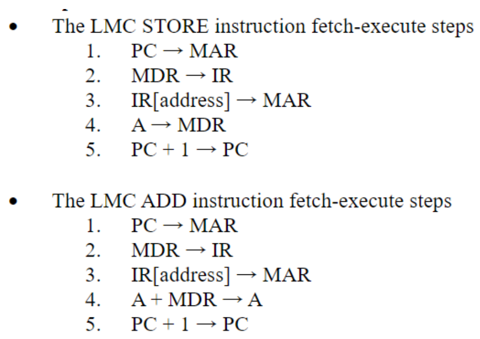
Chew Hwa Ern - Login Module

B1) The inter-relationship between MAR and MDR of computer architecture



Components: Memory Address Register (MAR) and Memory Data Register (MDR)

The MAR and MDR registers act as an interface between the CPU and memory. The MAR is a register that stores the memory address of the destination of data transferred to and from the CPU, whereas MDR is a register that holds data in the process of data being processed from the memory to the central processor, or vice versa. The MAR is connected to an address decoder that helps the MDR to retrieve the data stored at the address. When reading from memory, the CPU copies the address of memory to be accessed into MAR then sets Activation line on to start the data transfer that takes place by retrieving data from the specific memory location and stores it into the MDR register. When writing to memory, the CPU writes data from MDR and sets the activation line on to start the data transfer to the memory location whose address is stored in MAR.



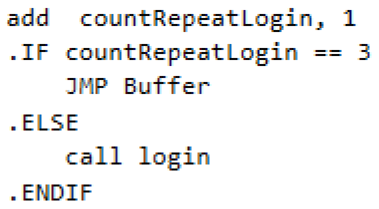
- The LMC STORE instruction fetch-execute steps
  1. PC → MAR
  2. MDR → IR
  3. IR[address] → MAR
  4. A → MDR
  5. PC + 1 → PC

- The LMC ADD instruction fetch-execute steps
  1. PC → MAR
  2. MDR → IR
  3. IR[address] → MAR
  4. A + MDR → A
  5. PC + 1 → PC

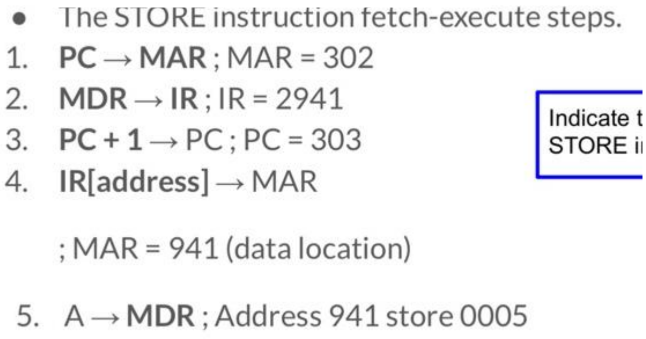
B2) Mathematical Operations used in Login Module

This module utilizes the add operation. To carry out the operation, the Fetch-Execute Instruction Cycle was used.. The fetch-execute instruction cycle is separated into two stages which are the fetch phase and the execute phase separately. In the fetch phase, the PC retrieves an instruction from the memory then translates the instruction into a set of commands. Then in the execution phase, the LOAD instruction execution steps consists of fetching the instruction pointed by the PC to the MAR, then copy the content in MDR to IR. PC then increases by 1 to hold the address of the next instruction to fetch. The ALU retrieves the address of instruction from the IR that is stored in the MAR. The ALU executes the instruction by adding the data held in MDR with the accumulator. The result is stored in the accumulator. Then the STORE instruction is executed.

This module uses the add operation by adding the login counts to count the number of attempts made. The accumulator in this case is 1, which is added to the data in MDR which is countRepeatLogin.



```
add  countRepeatLogin, 1
.IF countRepeatLogin == 3
    JMP Buffer
.ELSE
    call login
.ENDIF
```



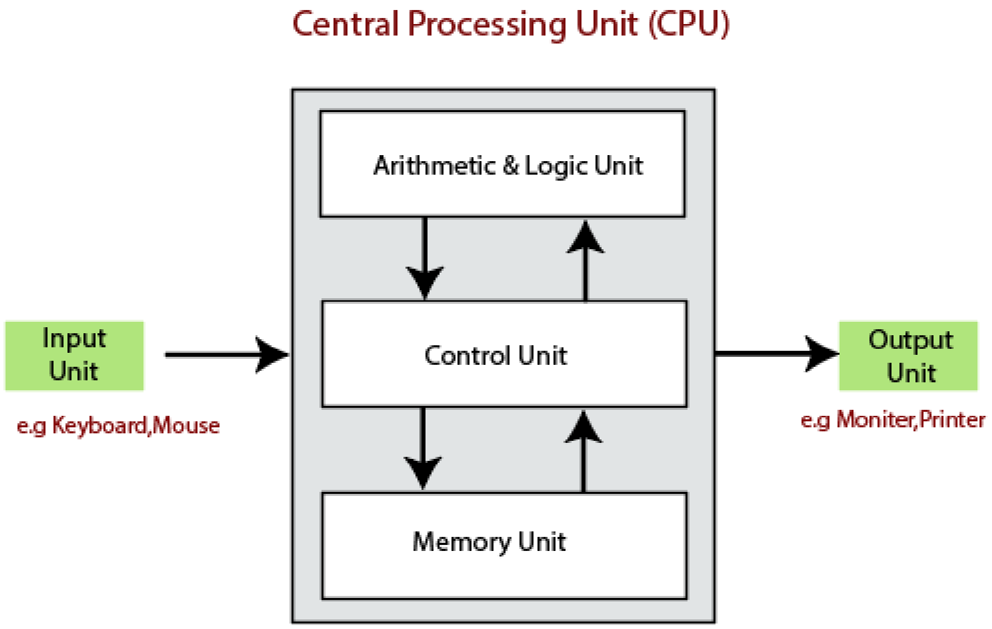
- The STORE instruction fetch-execute steps.

1. **PC → MAR** ; MAR = 302
2. **MDR → IR** ; IR = 2941
3. **PC + 1** → PC ; PC = 303
4. **IR[address]** → MAR

   ; MAR = 941 (data location)

5. A → **MDR** ; Address 941 store 0005

Lee Shu Ern - Order Module



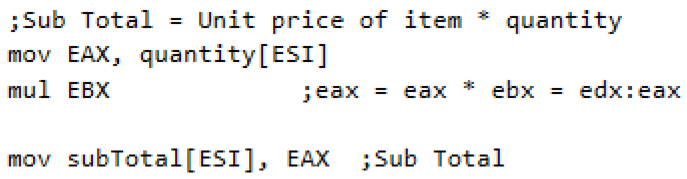
B1)The interrelationship between ALU and CU of computer architecture

Components : Arithmetic Logic Unit (ALU) and Control Unit (CU)

The ALU and CU are two CPU components where the CU controls all the operations in the CPU and is responsible for managing and controlling the instruction execution of the CPU. Registers are accessed directly by the Control Unit and not addressed as a memory location hence the CU is able to receive instructions or information directly from the main memory of the computer. When the CU receives an instruction or information, it converts the instruction to control signals which will then be sent to the central processor for further processing. The CU acts as a communicator between the memory and the ALU that is responsible for logical and arithmetic calculations by instructions provided by the CU and also depends entirely on the CU. The ALU presents the arithmetic and logical operation. The outputs of ALU will change asynchronously in response to the input. The basic arithmetic and bitwise logic functions are supported by ALU.

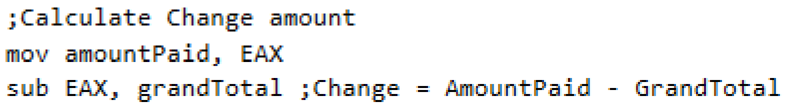
B2) Mathematical Operations used in Login Module

In this module, subtraction and multiplication operations are used. In both subtraction and multiplication operations, MAR will have to fetch the instruction code from the memory address from the PC(program counter). Then, the instruction located in the specific address will be retrieved by the MDR register. Then, instruction is loaded into IR from the MDR of which the IR will then decode and perform load action and add the PC by 1. This is to get to the next instructions in the PC. Then, ALU will perform the subsequent instruction that is to subtract or multiply the number in the PC with the data in MDR. The output/result of the ALU is then written back to update the CPU register file. The result is saved from the CPU to memory location product.



```
;Sub Total = Unit price of item * quantity
mov EAX, quantity[ESI]
mul EBX              ;eax = eax * ebx = edx:eax

mov subTotal[ESI], EAX  ;Sub Total
```

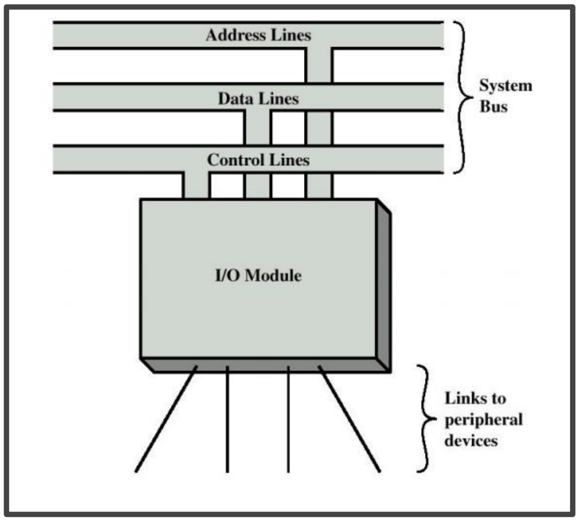
Subtotal is calculated by multiplying the unit price (declared constant) with the quantity ordered. In this case, the address of quantity is captured by the MAR to be moved into the MDR register. The ALU performs the subtraction with the data in MDR and stores the address of the result, subtotal, into the MAR.

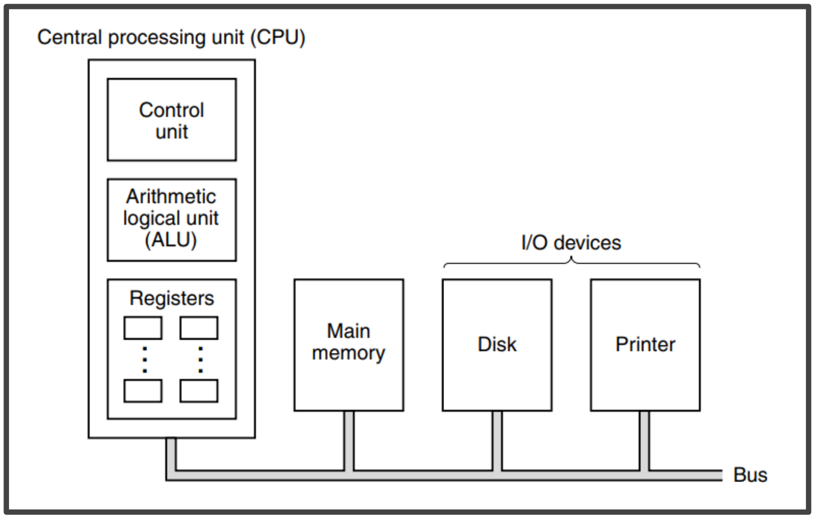


```
;Calculate Change amount
mov amountPaid, EAX
sub EAX, grandTotal ;Change = AmountPaid - GrandTotal
```

This module uses subtraction when calculating the amount paid and change due by the fetch and execute cycle.

Hee Sze Wei - Rating Module





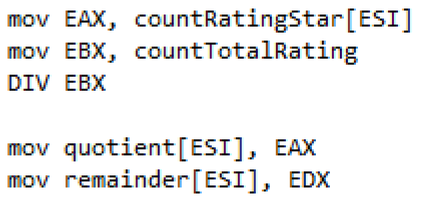
B1) The inter-relationship between I/O and Main Memory of computer architecture

Components : I/O and Memory

The I/O Module coordinates the flow of traffic between internal resources (such as the memory and CPU) and external devices. The I/O accepts instructions via input devices such as mouse and keyboard. Then converts the instructions and data to control signals and passes the signaled data to the computer system for further processing. The I/O module accepts commands from the CPU to control and perform I/O operations. It is able to transfer data directly from and to memory via system buses and provides a buffer to hold data until it is transferred. The main memory is a place that stores instructions and data as binary numbers. It consists of a set of addressed locations and is needed for I/O to communicate with the CPU. However, the Direct Memory Access (DMA) is an I/O handling method where data is transferred between the I/O device and memory without CPU intervention. This allows the DMA module itself to control the flow of data between the main memory and the I/O device.

B2) Mathematical Operations used in Rating Module

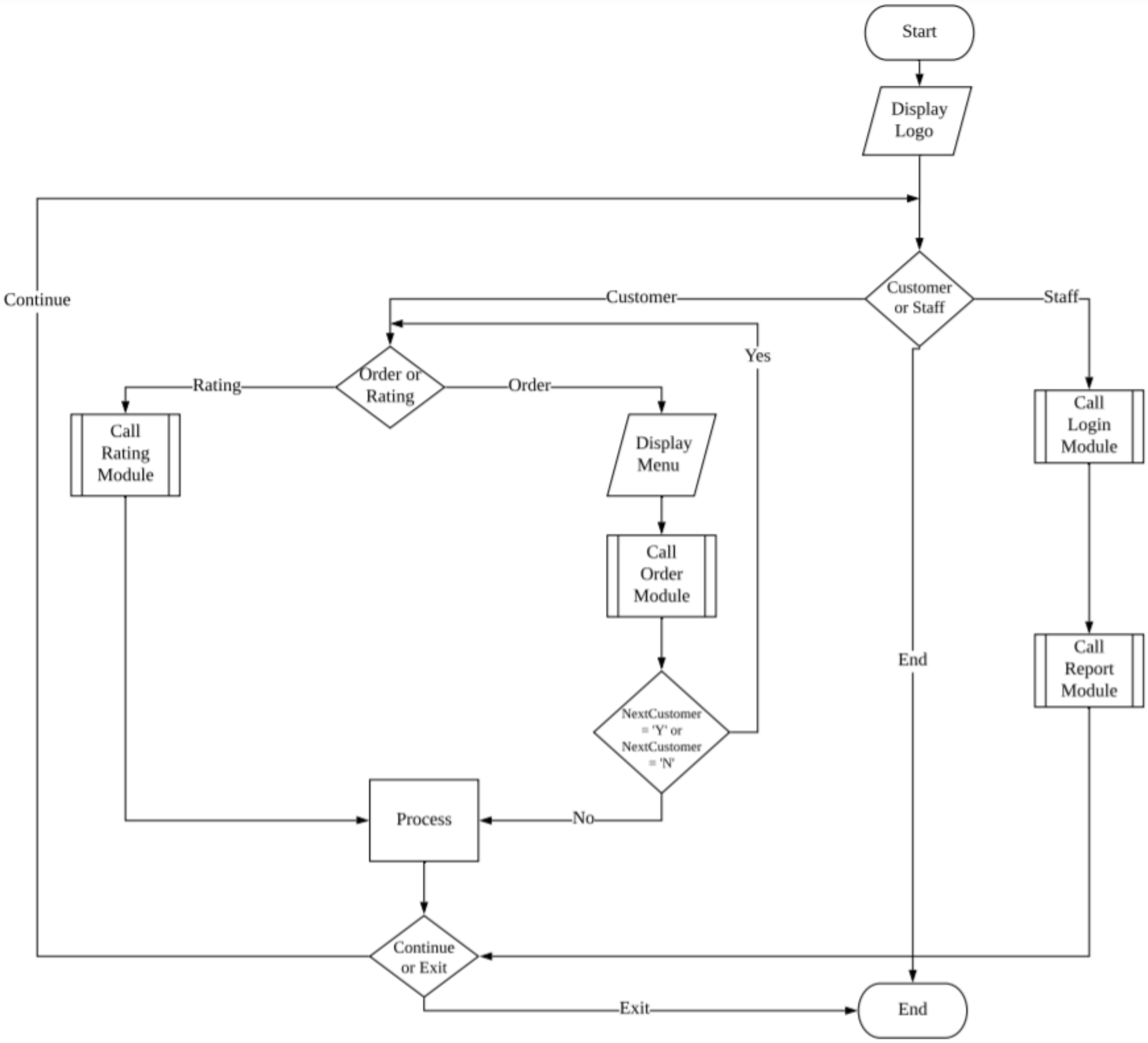
This module uses the division operation. This is done after the data is LOADED by the fetch-execution steps. Execution cycle starts with instruction at the current program counter (PC) being fetched from the main memory and stored in the IR. The PC gets the address of the instruction to the MAR fetched from the memory. Then, the address is sent from the MAR to the main memory. During the transfer of data, the address of the instruction is returned to the MDR. The encoded instruction present in the IR is then interpreted by the decoder in the CU and executes the division operation in the ALU by its decoded instructions. The program counter is incremented and the cycle repeats.

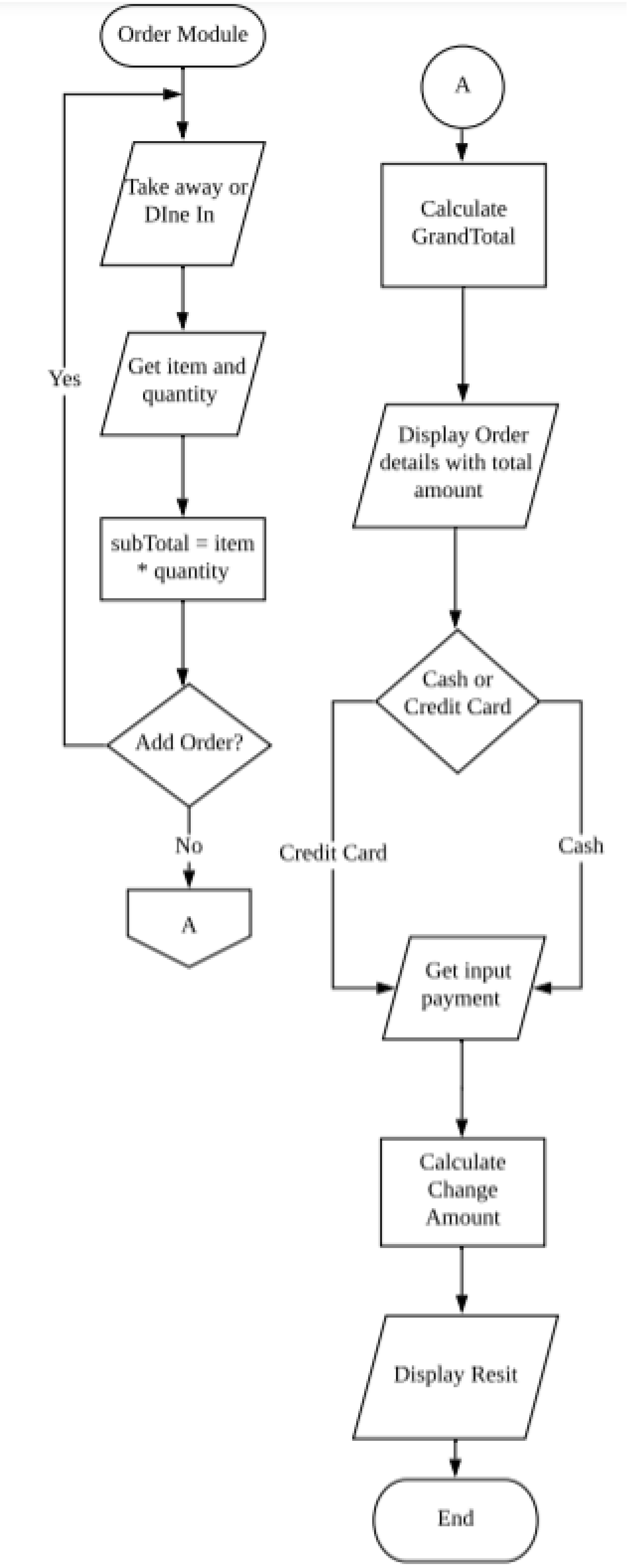


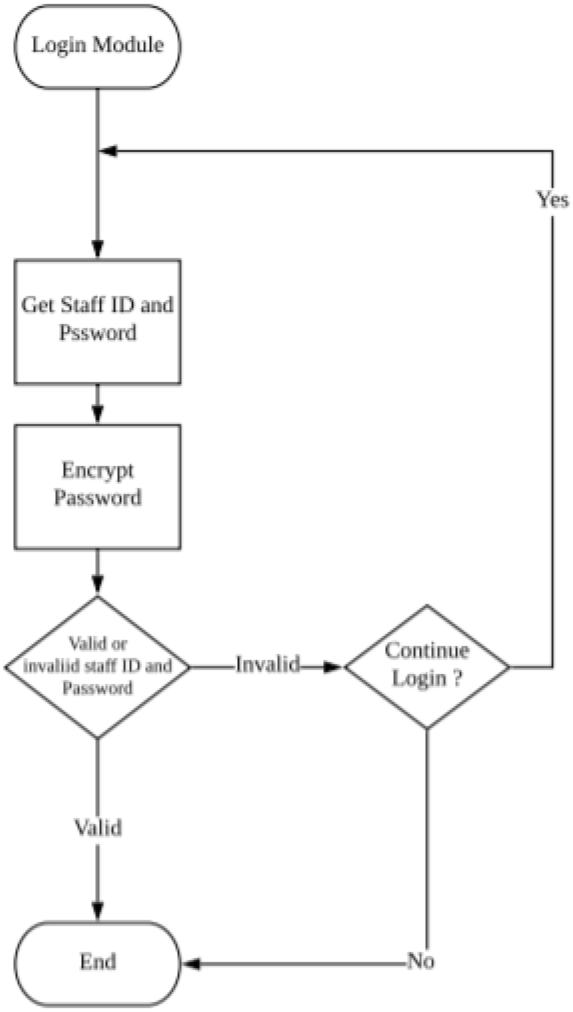
```
mov EAX, countRatingStar[ESI]
mov EBX, countTotalRating
DIV EBX

mov quotient[ESI], EAX
mov remainder[ESI], EDX
```

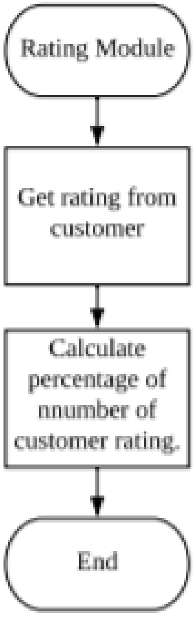
In this rating module, the division operand is fetched from the main memory to the PC to be stored in the IR. The MAR gets the address of the operand from the IR while the MDR obtains the data for the operation which are the countTotalRating and countRatingStar separately. The data is divided to obtain a remainder, which will then be stored in the accumulator from the MDR.

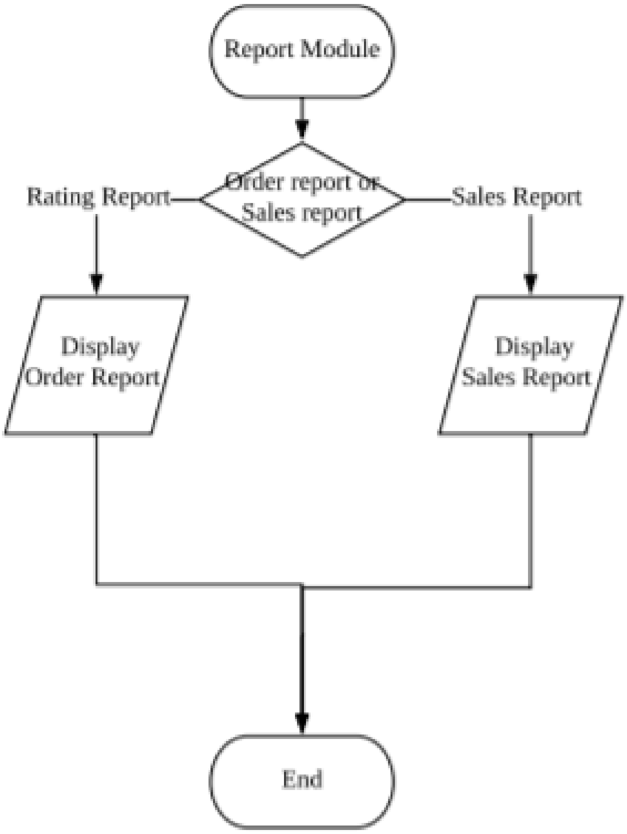
Flowchart


Start
Display Logo
Customer or Staff
Customer
Staff
Continue
Yes
Order or Rating
Rating
Order
Call Rating Module
Display Menu
Call Login Module
Call Order Module
End
Call Report Module
NextCustomer = 'Y' or NextCustomer = 'N'
Process
No
Continue or Exit
Exit
End

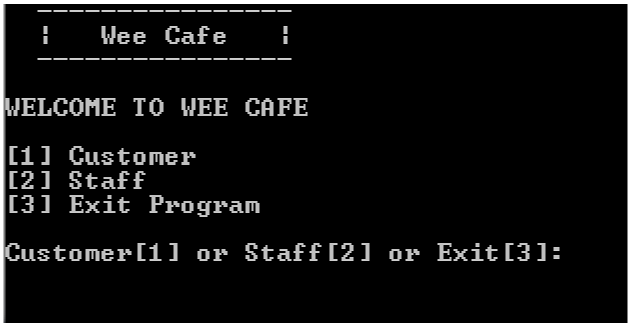

Order Module
Take away or DIne In
Get item and quantity
subTotal = item * quantity
Add Order?
Yes
No
A
A
Calculate GrandTotal
Display Order details with total amount
Cash or Credit Card
Credit Card
Cash
Get input payment
Calculate Change Amount
Display Resit
End


Login Module
Yes
Get Staff ID and Pssword
Encrypt Password
Valid or invaliid staff ID and Password
Invalid
Continue Login ?
Valid
End
No


Rating Module
Get rating from customer
Calculate percentage of nnumber of customer rating.
End

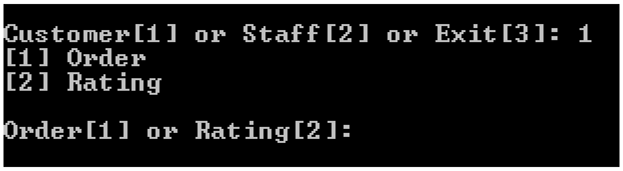

Report Module
Order report or Sales report
Rating Report
Sales Report
Display Order Report
Display Sales Report
End

Description



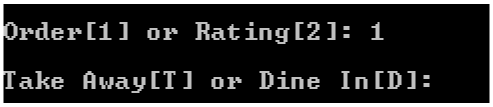
```
    ----------------
    ¦   Wee Cafe   ¦
    ----------------

WELCOME TO WEE CAFE

[1] Customer
[2] Staff
[3] Exit Program

Customer[1] or Staff[2] or Exit[3]:
```

The program will start up with our cafe logo. Then, users will be asked to input for customers or staff or exit the program. The option for customers is to let customers choose whether they want to make an order or the user wishes to rate their experience in Wee Cafe. The staff option is selected when the staff user wishes to login to generate a daily sales report. Last but not least, the exit program option is for user who wish to exit the program.



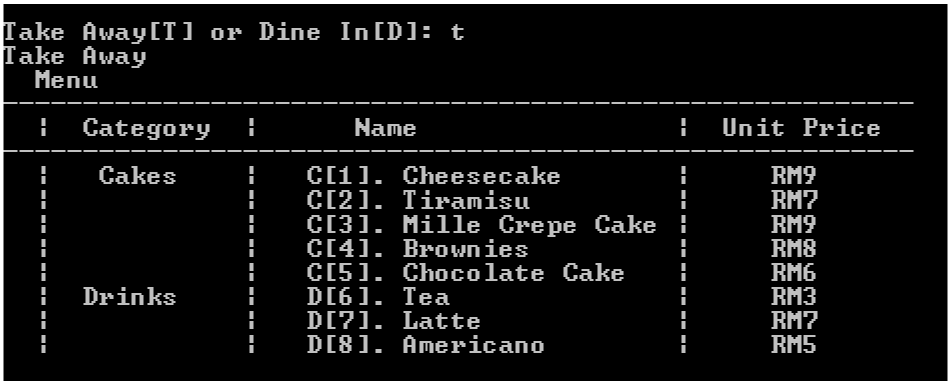
```
Customer[1] or Staff[2] or Exit[3]: 1
[1] Order
[2] Rating

Order[1] or Rating[2]:
```

If the user is a customer, the user will be asked to input for order or rating. The option for order is selected when the user wishes to place an order. Whereas rating is selected the user wishes to rate their experience in Wee Cafe.



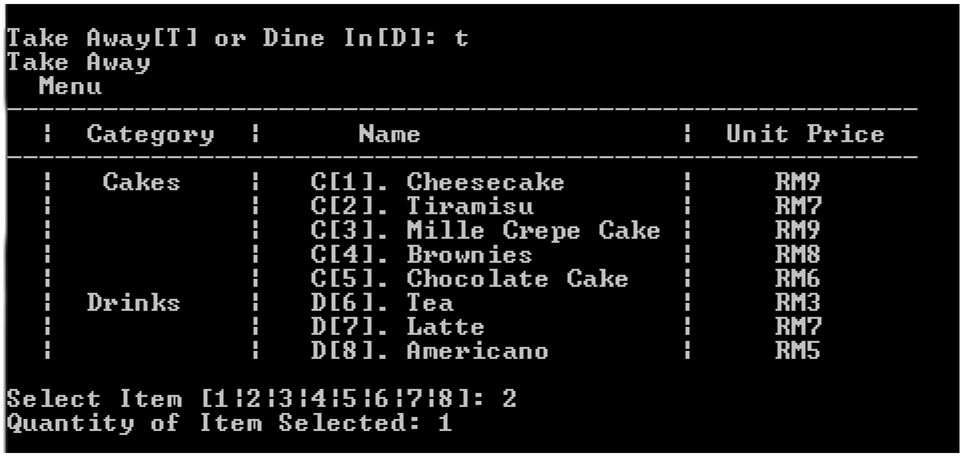
```
Order[1] or Rating[2]: 1

Take Away[T] or Dine In[D]:
```

If user input 1 for order, it will prompt the user to choose either to Take Away or Dine In.



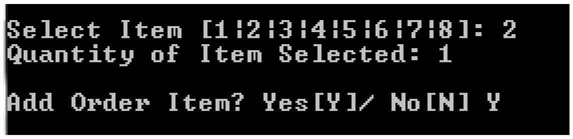
```
Take Away[T] or Dine In[D]: t
Take Away
  Menu
---------------------------------------------------------
  ¦  Category  ¦      Name                   ¦  Unit Price
---------------------------------------------------------
  ¦   Cakes    ¦   C[1]. Cheesecake          ¦     RM9
  ¦            ¦   C[2]. Tiramisu            ¦     RM7
  ¦            ¦   C[3]. Mille Crepe Cake    ¦     RM9
  ¦            ¦   C[4]. Brownies            ¦     RM8
  ¦            ¦   C[5]. Chocolate Cake      ¦     RM6
  ¦  Drinks    ¦   D[6]. Tea                 ¦     RM3
  ¦            ¦   D[7]. Latte               ¦     RM7
  ¦            ¦   D[8]. Americano           ¦     RM5
```

Then, the menu of Wee Cafe will be displayed.



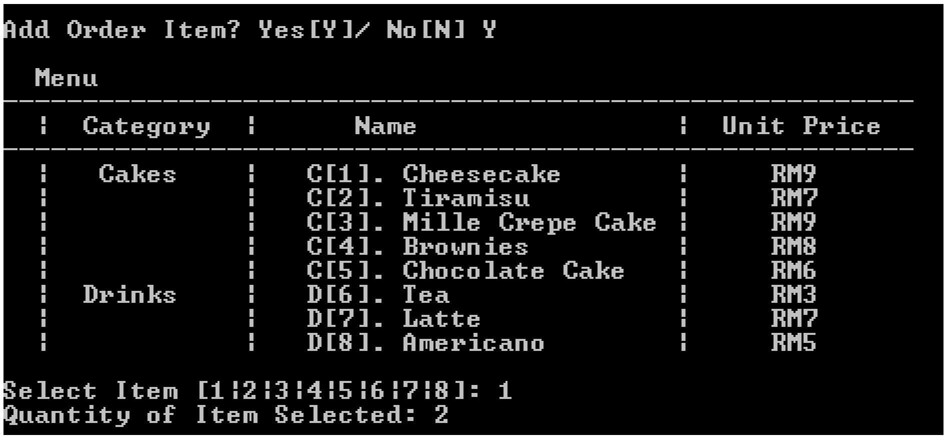
```
Take Away[T] or Dine In[D]: t
Take Away
  Menu
---------------------------------------------------------
  ¦  Category  ¦       Name               ¦  Unit Price
---------------------------------------------------------
  ¦   Cakes    ¦   C[1]. Cheesecake       ¦      RM9
  ¦            ¦   C[2]. Tiramisu         ¦      RM7
  ¦            ¦   C[3]. Mille Crepe Cake ¦      RM9
  ¦            ¦   C[4]. Brownies         ¦      RM8
  ¦            ¦   C[5]. Chocolate Cake   ¦      RM6
  ¦  Drinks    ¦   D[6]. Tea              ¦      RM3
  ¦            ¦   D[7]. Latte            ¦      RM7
  ¦            ¦   D[8]. Americano        ¦      RM5

Select Item [1¦2¦3¦4¦5¦6¦7¦8]: 2
Quantity of Item Selected: 1
```

Users are required to choose an order item and input quantity of item selected.



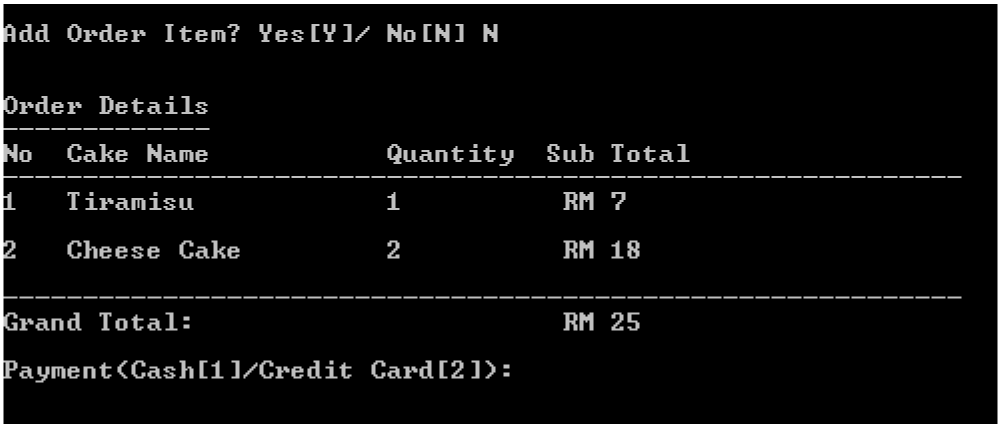
```
Select Item [1¦2¦3¦4¦5¦6¦7¦8]: 2
Quantity of Item Selected: 1

Add Order Item? Yes[Y]/ No[N] Y
```

System will then prompt the user to choose to add an order item or not to. If the user wishes to add an order item, the loop back to allow the user to select the desired order item.



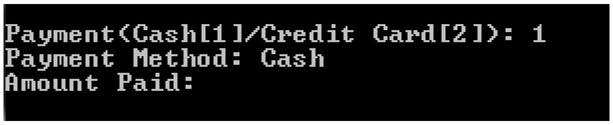
```
Add Order Item? Yes[Y]/ No[N] Y

  Menu
---------------------------------------------------------
  ¦  Category  ¦       Name               ¦  Unit Price
---------------------------------------------------------
  ¦   Cakes    ¦   C[1]. Cheesecake       ¦      RM9
  ¦            ¦   C[2]. Tiramisu         ¦      RM7
  ¦            ¦   C[3]. Mille Crepe Cake ¦      RM9
  ¦            ¦   C[4]. Brownies         ¦      RM8
  ¦            ¦   C[5]. Chocolate Cake   ¦      RM6
  ¦  Drinks    ¦   D[6]. Tea              ¦      RM3
  ¦            ¦   D[7]. Latte            ¦      RM7
  ¦            ¦   D[8]. Americano        ¦      RM5

Select Item [1¦2¦3¦4¦5¦6¦7¦8]: 1
Quantity of Item Selected: 2
```

If the user wishes to add an order item, the menu will then display again and the user will be prompted to choose the selected order item and quantity.



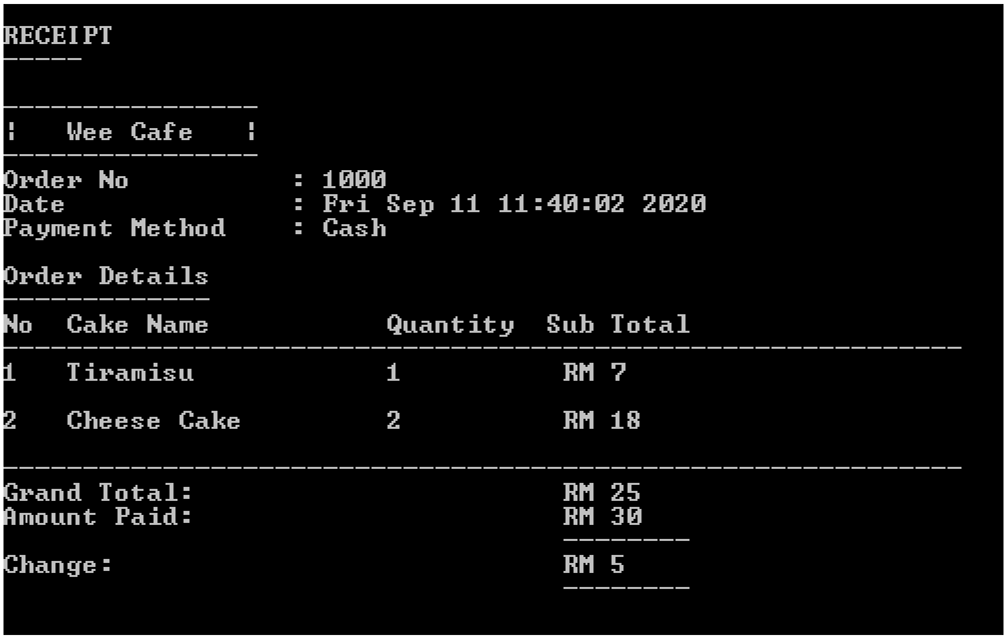
```
Add Order Item? Yes[Y]/ No[N] N

Order Details
-------------
No  Cake Name           Quantity  Sub Total
------------------------------------------------------------
1   Tiramisu            1          RM 7

2   Cheese Cake         2          RM 18

------------------------------------------------------------
Grand Total:                       RM 25

Payment(Cash[1]/Credit Card[2]):
```

If the user no longer wishes to add an order item, Display Order Details and prompt payment choice.



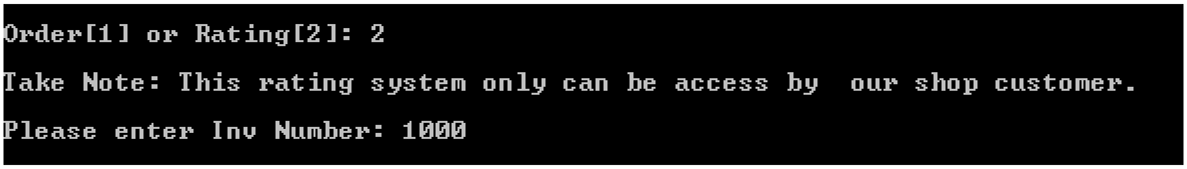
```
Payment(Cash[1]/Credit Card[2]): 1
Payment Method: Cash
Amount Paid:
```

When payment method is selected, the user is to enter the exact amount paid. This is to calculate change needed.



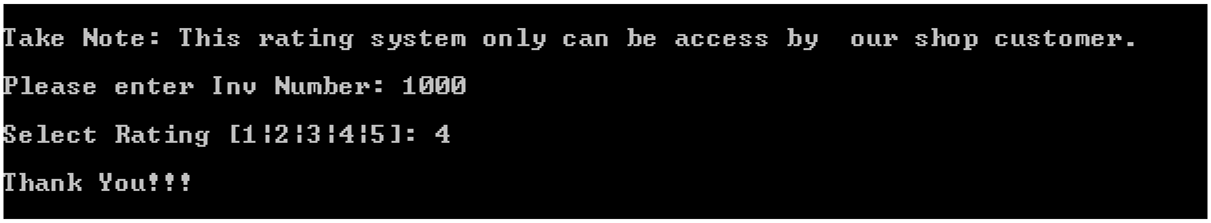
```
RECEIPT
-----

----------------
¦   Wee Cafe   ¦
----------------
Order No          : 1000
Date              : Fri Sep 11 11:40:02 2020
Payment Method    : Cash

Order Details
-------------
No  Cake Name           Quantity  Sub Total
------------------------------------------------------------
1   Tiramisu            1          RM 7

2   Cheese Cake         2          RM 18

------------------------------------------------------------
Grand Total:                       RM 25
Amount Paid:                       RM 30
                                   --------
Change:                            RM 5
                                   --------
```

After payment is made, a receipt is generated and displayed. Grand total is used to calculate change due by subtracting the amount paid with the grand total.



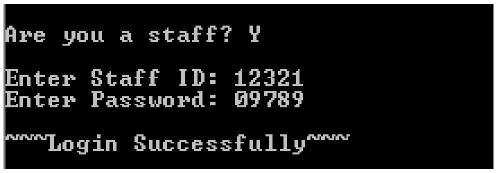
```
Order[1] or Rating[2]: 2

Take Note: This rating system only can be access by  our shop customer.

Please enter Inv Number: 1000
```

Then, if the user is a customer and wishes to rate the restaurant. The user is required to enter a valid invoice number which is printed on their receipt when they made their purchase before they can proceed to the next step.

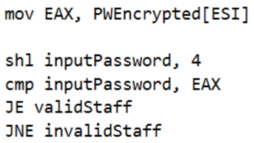


```
Take Note: This rating system only can be access by  our shop customer.

Please enter Inv Number: 1000

Select Rating [1¦2¦3¦4¦5]: 4

Thank You!!!
```

If the invoice number is valid, the user can proceed to rate their experience with the restaurant on the scale of 1 to 5 stars. This will then be use to generate a complete report.

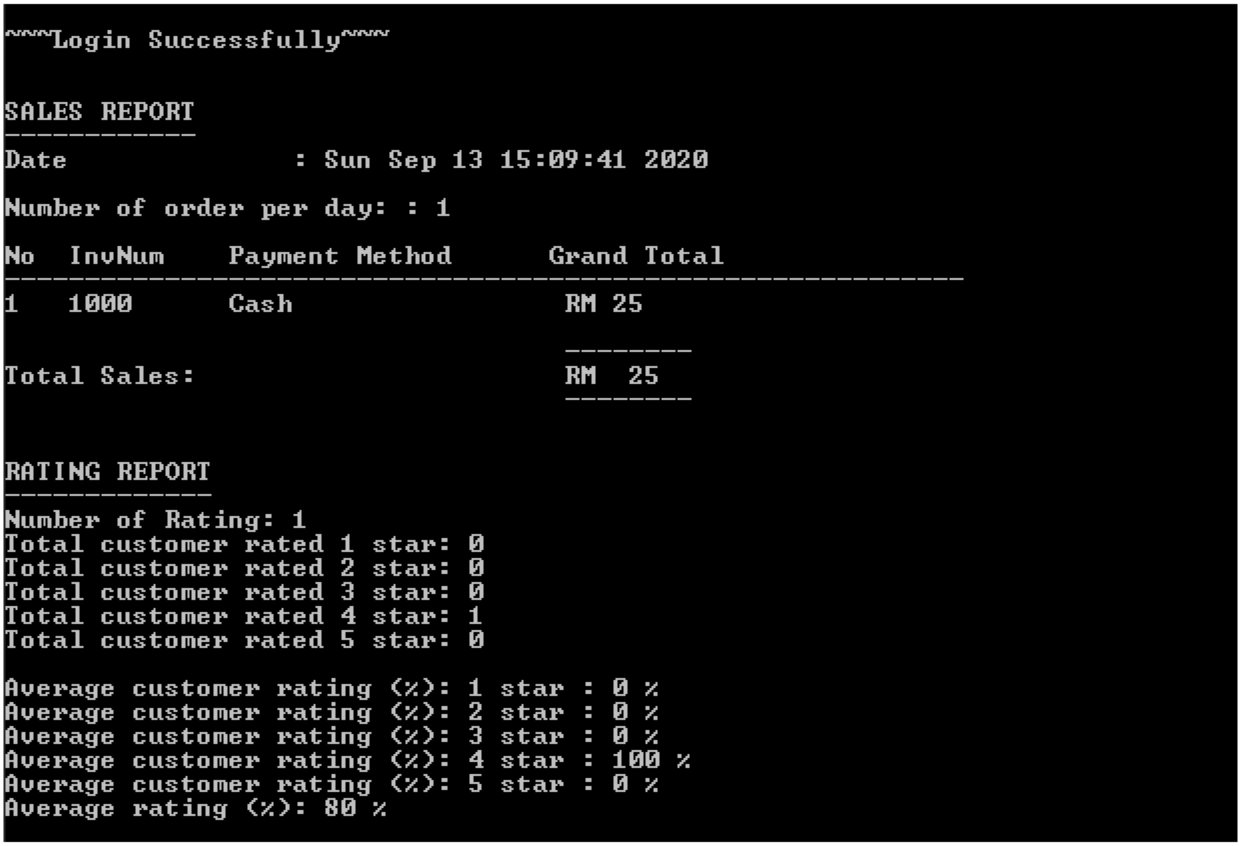


```
Are you a staff? Y

Enter Staff ID: 12321
Enter Password: 09789

~~~Login Successfully~~~
```



```
mov EAX, PWEncrypted[ESI]

shl inputPassword, 4
cmp inputPassword, EAX
JE validStaff
JNE invalidStaff
```

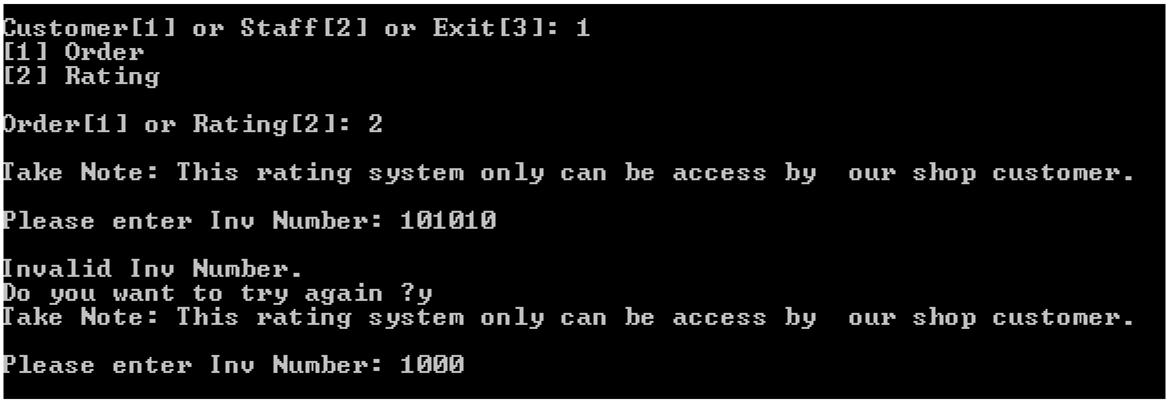
If the user is a staff and selects the Staff Login Option,the staff will be asked if he is a staff first to verify his identity. Then to further validate the staffs’ identity, the staff is prompted to enter the correct Staff ID and Password,which will be encrypted. If the password matches the encrypted password, the staff is logged in successfully.



```
~~~Login Successfully~~~

SALES REPORT
------------
Date              : Sun Sep 13 15:09:41 2020

Number of order per day: : 1

No  InvNum    Payment Method      Grand Total
------------------------------------------------------------
1   1000      Cash                 RM 25

                                   --------
Total Sales:                       RM  25
                                   --------


RATING REPORT
-------------
Number of Rating: 1
Total customer rated 1 star: 0
Total customer rated 2 star: 0
Total customer rated 3 star: 0
Total customer rated 4 star: 1
Total customer rated 5 star: 0

Average customer rating (%): 1 star : 0 %
Average customer rating (%): 2 star : 0 %
Average customer rating (%): 3 star : 0 %
Average customer rating (%): 4 star : 100 %
Average customer rating (%): 5 star : 0 %
Average rating (%): 80 %
```

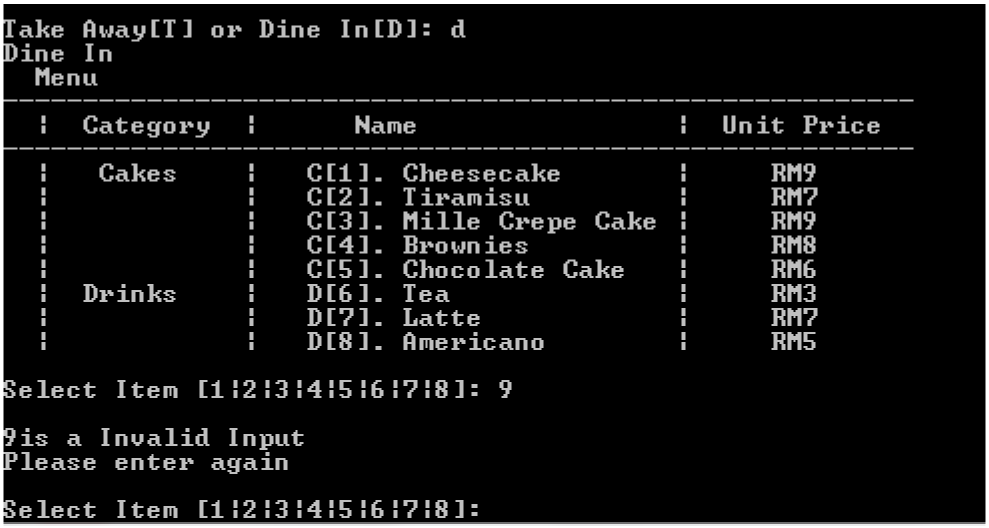
Upon successful login, a sales report and rating report will be generated and displayed for staff to read and evaluate productivity of the cafe. The average rating is calculated by dividing the ratedStar which is 4 in the screenshot above with the total ratedStar (5 - The highest possible ratedStar) and then multiplied by 100%.

Exception Handling



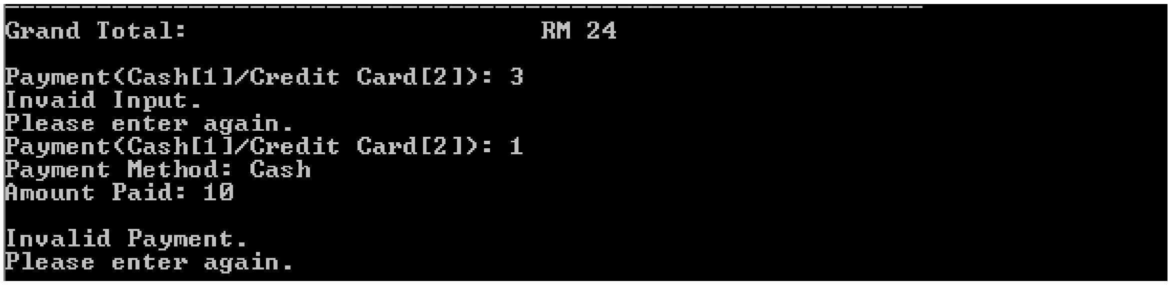
```
Customer[1] or Staff[2] or Exit[3]: 1
[1] Order
[2] Rating

Order[1] or Rating[2]: 2

Take Note: This rating system only can be access by  our shop customer.

Please enter Inv Number: 101010

Invalid Inv Number.
Do you want to try again ?y
Take Note: This rating system only can be access by  our shop customer.

Please enter Inv Number: 1000
```

If a customer enters an invalid invoice number, they are not allowed to do any rating. Then the customer needs to input their respond whether they want to re-enter a valid invoice number or exit from the rating module.



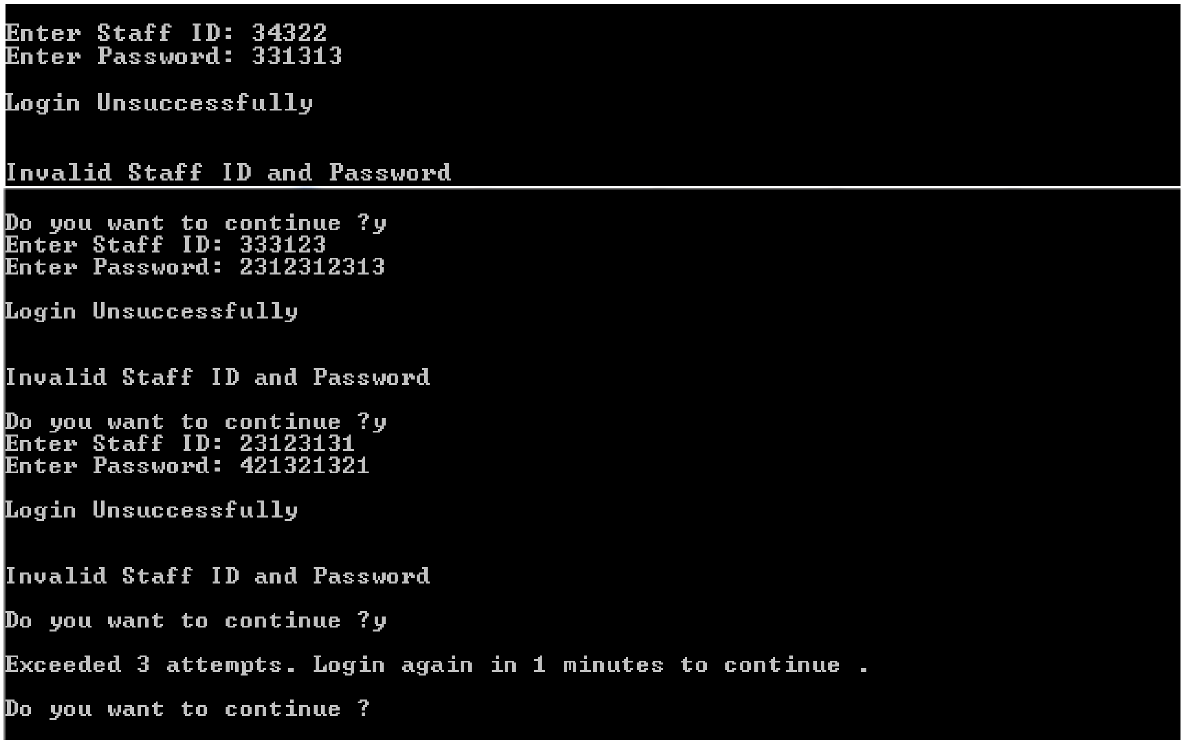
```
Take Away[T] or Dine In[D]: d
Dine In
  Menu
----------------------------------------------------------
  ¦  Category  ¦      Name               ¦  Unit Price
----------------------------------------------------------
  ¦   Cakes    ¦   C[1]. Cheesecake      ¦      RM9
  ¦            ¦   C[2]. Tiramisu        ¦      RM7
  ¦            ¦   C[3]. Mille Crepe Cake ¦     RM9
  ¦            ¦   C[4]. Brownies        ¦      RM8
  ¦            ¦   C[5]. Chocolate Cake  ¦      RM6
  ¦  Drinks    ¦   D[6]. Tea             ¦      RM3
  ¦            ¦   D[7]. Latte           ¦      RM7
  ¦            ¦   D[8]. Americano       ¦      RM5

Select Item [1¦2¦3¦4¦5¦6¦7¦8]: 9

9is a Invalid Input
Please enter again

Select Item [1¦2¦3¦4¦5¦6¦7¦8]:
```

The selected item is out of range. Loops back to select order item again.



```
----------------------------------------------------------
Grand Total:                       RM 24

Payment(Cash[1]/Credit Card[2]): 3
Invaid Input.
Please enter again.
Payment(Cash[1]/Credit Card[2]): 1
Payment Method: Cash
Amount Paid: 10

Invalid Payment.
Please enter again.
```

Payment method option selected falls out of range. Prompts to enter again. Then, If the amount paid is less than the grand total, payment is invalid.



```
Enter Staff ID: 34322
Enter Password: 331313

Login Unsuccessfully


Invalid Staff ID and Password

Do you want to continue ?y
Enter Staff ID: 333123
Enter Password: 2312312313

Login Unsuccessfully


Invalid Staff ID and Password

Do you want to continue ?y
Enter Staff ID: 23123131
Enter Password: 421321321

Login Unsuccessfully


Invalid Staff ID and Password

Do you want to continue ?y

Exceeded 3 attempts. Login again in 1 minutes to continue .

Do you want to continue ?
```

If staff ID and password is invalid, login is unsuccessful. If user exceeds 3 failed login attempts Delay for 1 minute before the next prompt.

Task Allocations

Name	Task Allocated
Chew Hwa Ern	Login Module, Report Module, Documentation, Flowchart
Hee Sze Wei	Rating Module, Report Module, Documentation, Flowchart
Lee Shu Ern	Order Module, Report Module, Documentation, Flowchart