**Table of Contents**

[**Introduction** 4](#_Toc82094918)

[**Flowchart** 6](#_Toc82094919)

[**Design-Pseudocode** 17](#_Toc82094920)

[**Program Illustrate:** 39](#_Toc82094921)

[**Selected output** 48](#_Toc82094922)

[**Conclusion** 55](#_Toc82094923)

# **Introduction**

The project's purpose of this assessment aims is to develop a single Python program in which bank administrators and purchasers can use to manage credit. Each part possesses its own set of modules and every segment is composed of order and clear strands. It's also possible to create a different collection book which is record from a different party. The chief relates to everyone’s documents and has the right to authorize new hires, change information and other records, and issue clear credit identifications to buyers. On the other hand, the administrators will provide clients with access to a wide range of entry points and sections. Clients are also two-part division groups: "registered" and "unregistered". Each of these two types of purchasers has a smaller and different market share. Note buyers must use a unique certification method to access the purchaser's door, which requires them to click on their loading credentials. This task was developed to meet a large number of requirements.

**Presupposition:**

1. Create a loading framework for the administrators and client parts, and set up a separate loading procedure foreach different client title and password. Purchasers may be able to access their chosen entrance if their credentials are successfully provided.
2. Text documents should be created for a variety of applications, as needed. As a sample, a single item document is going to use to maintain a record of purchaser's individual documentation, as well as another used to maintain tabs on the situation purchaser's respect advice.
3. Manufacture multiple variables and each variable has a title that is distinct.
4. Each section has been archived for future reference in the course of the interaction of the encoding test records to address recurring difficulties in future development.
5. After entering the credit quantity, credit residence and financing costs, the prepaid credit calculator calculates the required monthly payments.

Greeting in the bank of Malaysia online credit and management order (MBOLMS) application form. All credit, portion, and payment information are here. You may use the website to check the payment era and follow your bank activities. Purchasers' information may also be found by administrators, and orders can be authorized or denied. Python scripts make it simple to carry out each of these tasks.

**Assumption:**

First Reader will fill out a paper form with all the information and submit it to the account. After that, the paper is more likely to be lost. It took a long time to complete the task. Finally, it takes time to report the supporter.

# **Flowchart**

Flowchart overview

Section of the leader

If answer ==?

Document consumer initialize sheet

Close

Latest client leaf

Leader load sheet

See answer

First page

Create

Do you wish to check the client's credits?

Back to the home page

The registration procedure has been completed

loading

Are you sure you want to certify?

False

False

True

Leader load sheet

d

c

b

a

Look at the purchaser's settlements

Check the purchaser's credit application

Check the purchaser's application

Option not valid

answer ==?

Items are wrong

Fill in the blanks

Please fill in the blanks

True

 Unable to Prove your credit

It is impossible to Prove you

False

True

False

Check out the settlement card

False

Answer

To access client statistics, fill in the essential fields

e

d

c

b

a

Observe all types of credit settlements

Purchasers' points of view

view results by credit type

View client statistics by category

answer==?

To display the credit type, enter the required documentation

Back to the leader page

To display every purchaser, enter the required information

True

option for reading

signet every searched client to certify from the search directory

certifying the purchaser's application

Keeping track of the due era

Separate client IDs with commas (,)

Put all credit IDs called using the put role

Document placement so that all purchasers' credit IDs can be undersTHROUGHod

Generates a unique credit ID for every purchaser

Certify all applications

Certify the items required by the purchaser

Show all uncertifyed applications

If credit hold<=5

Credit cost = 10.50

Elif credit hold<= 10

Credit tax = 11.50

Else print “The credit application stage is out of rank”

If kind of credit == personal

Whether credit hold<=5

Credit cost= 11.02

Elif credit hold<= 10

Credit fee = 12.52

Elif credit hold>10

Credit rate = 12.50

If kind of credit == home

If credit hold<=5

Credit cost =11.32

Elif credit hold<=10

Credit rate = 12.32

Else print “The credit application period isn't applicable”

If variety credit== education

Credit type EMI calculation

if class of credit == car

Whether credit hold<=5

credit cost= 11.02

Elif credit hold<= 10

Credit wage = 12.52

Else print “The credit application period is out of bounds”

EMI = Suggest Credit Quantity \* r \* ((1+r) \*\* ((1+r)/ (number of months) \*\* number of months - 1)

Use the (r+) function to add to a document to check the client's credit ID

Read and prove all documentation in the document

Show uncertifyed credits

Reading documentation in fie and applying (r) role via locate the portion

Check deadline

Reading the documentation from the document of certifyed clienttitles, create a new clienttitle

Make a new password that includes one capital letter, one number, and one unique symbol

Gate

New client function card

Do you want to go back to the previous card?

d

c

b

a

True

Back to the first card

Create a new account

Check Credit Statistics

Calculator for credits

Answer ==?

See option

Latest client card

If your age >= 18

True

You can open a new account

You are unable to create a new account

Wrong choice

False

False

Loading page as submit clients

You have logged in successfully.

Supplied id plus password are wrong

Within c documentation, certify client title also password

Enter your loading, sign, and more details

True

False

If you keep the login and password in the c documentation

Card as clients who have record

d

c

e

b

a

Back to the home page

Credit status

Check out the settlement

Make a credit payment

Wrong option

answer ==?

Study answer

Displays all payments made by the client, the credit executed, the quantity of the payment, and the due era

False

True

View credit information and apply for a credit

True

False

Back to the client registration card

Make the payment

Are you willing to make the payment?

Client title, Credit ID Type of Credit, Credit Quantity Covered, Percentage of Credit Quantity Covered Credit Holding Era Portion Era, Payment Quantity and Monthly Portion Quantity Included

Types of credits available

d

Education credit

True

False

Stop

Select the desired credit option

c

b

a

Study answer

Personal credit

Home credit

Car credit

Wrong answer

If the answer ==?

# **Design-Pseudocode**

# Birendra dahal

# NP000573

BROUGHT sys

BROUGHT re

BROUGHT eratime

BROUGHT calendar

BROUGHT sys

from uuid BROUGHT uuid4

BROUGHT tabulate

EXPLAIN ROLE reading\_document\_documentation(documenttitle, mode):

with open(documenttitle, mode) as fo:

STICK document\_content THROUGH fo.read()

STICK document\_content THROUGH document\_content.split('\n')

STICK documentation\_record THROUGH []

AS documentation IN document\_content:

STICK documentation THROUGH documentation.split(',')

documentation\_record.append(documentation)

documentation\_record.pop(-1)

BACK documentation\_record

# Leader Loading

EXPLAIN ROLE leader\_loading():

WHILE True:

STICK leader\_title THROUGH PROCESS("Clienttitle: ")

STICK leader\_psd THROUGH PROCESS("Password: ")

STICK leader\_record THROUGH reading\_document\_documentation('leaders.txt', 'r')

ACHIEVE(leader\_record)

STICK flag THROUGH 0

AS c\_documentation IN leader\_record:

IF leader\_title IN c\_documentation plus leader\_psd IN c\_documentation:

ACHIEVE(c\_documentation[0], c\_documentation[1])

STICK flag THROUGH 1

IF flag:

ACHIEVE("\You have successfully logged into your account.\n")

every\_leader\_ROLEs()

ELSE:

ACHIEVE("\nEntry Denied !!!\n")

# Leader card

EXPLAIN ROLE all\_leader\_ROLEs():

ACHIEVE("""\n

----------Leader Card ------------\n

1. ProvePurchaser Applications

2. ProveCredit Applications

3. Outlook Settlements

4. Move THROUGH Head Card

\n""")

TEST:

STICK ch THROUGH int(PROCESS("Choose your options THROUGH do: "))

BUT:

ACHIEVE("\nNumber PROCESS are only acceptable !!!\n")

every\_leader\_ROLEs()

ELSE:

IF ch SIMILAR 1:

EXPLAIN ROLE certify\_purchaser\_application():

outlook\_unverified\_req()

STICK certify\_req THROUGH PROCESS("\nDo you want THROUGH Proveclient application (y/n) ? ")

WHILE certify\_req SIMILAR 'y':

STICK clienttitle THROUGH PROCESS("Enter clienttitle: ")

IF certify\_purchasers(clienttitle):

ACHIEVE(f"\n{clienttitle} has been verified...\n")

certify\_purchaser\_application()

all\_leader\_ROLEs()

certify\_purchaser\_application()

ELSEIF ch SIMILAR 2:

EXPLAIN ROLE certify\_credit\_application():

placing\_credit\_id\_in\_document()

outlook\_unverified\_credit()

STICK certify\_credit THROUGH PROCESS("\nDo you want THROUGH Proveclient credits (y/n) ?")

WHILE certify\_credit SIMILAR 'y':

STICK credit\_id THROUGH PROCESS("Enter Credit ID: ")

IF certify\_client\_credit\_id(

credit\_id):

IF placing\_portion\_era(credit\_id):

IF calculates\_credit\_emi(

credit\_id): # calling ROLE that writes credit calculation items IN a document

ACHIEVE(f"\n Credit ID '{credit\_id}' has been verified...\n")

certify\_credit\_application()

all\_leader\_ROLEs()

certify\_credit\_application()

ELSEIF ch SIMILAR 3:

outlook\_settlements\_card()

ELSE:

main\_card()

EXPLAIN ROLE outlook\_settlements\_card():

ACHIEVE("""

------ Settlements Outlook Card -------\n

1. Outlook by Specific Purchasers

2. Outlook by Specific Credit Type

3. Outlook of all purchaser

4. Outlook settlement of all types Credit

5. Go THROUGH Leader Card

""")

STICK ch THROUGH int(PROCESS("\nSelect what you are looking for THROUGH perform (1-4):"))

IF ch SIMILAR 1:

outlook\_of\_certain\_purchasers()

ELSEIF ch SIMILAR 2:

outlook\_of\_certain\_credit\_kind()

ELSEIF ch SIMILAR 3:

outlook\_of\_all\_purchasers()

ELSEIF ch SIMILAR 4:

outlook\_settlements\_of\_all\_set\_credit()

ELSE:

all\_leader\_ROLEs()

EXPLAIN ROLE goTHROUGH\_outlook\_settlements\_card():

STICK inp\_client THROUGH PROCESS("\n Do you wish to THROUGH return THROUGH previous card (y/n)?")

IF inp\_client SIMILAR 'y':

outlook\_settlements\_card()

ELSE:

sys.exit()

EXPLAIN ROLE outlook\_of\_certain\_purchasers():

STICK customer\_title THROUGH PROCESS("Enter clienttitle: ")

STICK credit\_items THROUGH reading\_document\_documentation('applied\_credit.txt', 'r')

STICK table\_documentation THROUGH []

AS credit\_documentation IN credit\_items:

IF credit\_documentation[2] SIMILAR client\_title:

STICK titled\_documentation THROUGH credit\_documentation[1], credit\_documentation[2], credit\_documentation[4], credit\_documentation[5], credit\_documentation[6] + ' years', credit\_documentation[

7] + ' %', credit\_documentation[8], credit\_documentation[9]

table\_documentation.append(titled\_documentation)

STICK title THROUGH ['Credit Id', 'Clienttitle', 'Credit Types', 'Applied Credit Quantity', 'Credit Hold', 'Credit Rate', 'Portion Quantity',

'Due Quantity']

ACHIEVE(tabulate.tabulate(table\_documentation, headers=title, tablefmt="github"))

goTHROUGH\_outlook\_settlements\_card()

EXPLAIN ROLE outlook\_of\_specific\_credit\_type():

STICK credit\_type THROUGH PROCESS("\nEnter types of credit: ")

STICK credit\_items THROUGH reading\_document\_documentation('applied\_credit.txt', 'r')

STICK table\_documentation THROUGH []

AS credit\_documentation IN credit\_items:

IF credit\_type IN credit\_documentation:

STICK titled\_documentation THROUGH credit\_documentation[1], credit\_documentation[2], credit\_documentation[4], credit\_documentation[5], credit\_documentation[6] + ' years', credit\_documentation[

7] + ' %', credit\_documentation[8], credit\_documentation[9]

table\_documentation.append(titled\_documentation)

STICK title THROUGH ['Credit Id', 'Clienttitle', 'Credit Types', 'Applied Credit Quantity', 'Credit Hold', 'Credit Rate', 'Portion Quantity',

'Due Quantity']

ACHIEVE(order.tabulate([titled\_documentation], headers=title, tablefmt="github"))

goTHROUGH\_outlook\_settlements\_card()

EXPLAIN ROLE outlook\_of\_all\_purchasers():

STICK credit\_items THROUGH reading\_document\_documentation('applied\_credit.txt', 'r')

STICK table\_documentation THROUGH []

AS credit\_documentation IN credit\_items:

IF credit\_documentation[-1] SIMILAR 'Verified':

STICK titled\_documentation THROUGH credit\_documentation[1], credit\_documentation[2], credit\_documentation[4], credit\_documentation[5], credit\_documentation[6] + ' years', credit\_documentation[

7] + ' %', credit\_documentation[8], credit\_documentation[9]

table\_documentation.append(titled\_documentation)

STICK title THROUGH ['Credit Id', 'Clienttitle', 'Credit Types', 'Applied Credit Quantity', 'Credit Hold', 'Credit Rate', 'Portion Quantity',

'Due Quantity']

ACHIEVE(order.tabulate([titled\_documentation], headers=title, tablefmt="github"))

goTHROUGH\_outlook\_settlements\_card()

EXPLAIN ROLE outlook\_settlements\_of\_all\_types\_credit():

outlook\_of\_all\_purchasers()

goTHROUGH\_outlook\_settlements\_card()

EXPLAIN ROLE certify\_purchasers(clienttitle):

with open('items.txt', 'r+') as document:

STICK purchaser\_items THROUGH document.readlines()

AS elem IN range(len(purchaser\_items)):

IF clienttitle IN purchaser\_items[elem]: # checking IF clienttitle

STICK client\_documentation THROUGH purchaser\_items[elem].split(',')

STICK client\_documentation[-1] THROUGH client\_documentation[-1].replace(client\_documentation[-1][:5],

'Verified')

STICK str\_ud THROUGH ','.join([str(e) AS e IN client\_documentation])

STICK purchaser\_items[elem] THROUGH purchaser\_items[elem].replace(str(purchaser\_items[elem]), str\_ud)

document.seek(0)

document.writelines(purchaser\_items)

BACK True

EXPLAIN ROLE outlook\_unverified\_req():

STICK c\_record THROUGH reading\_document\_documentation('items.txt', 'r')

STICK coming\_clients THROUGH []

AS ele IN c\_record:

IF ele[-1] SIMILAR "False":

coming\_clients.append(ele)

STICK title THROUGH ["Title", "Address", "E-mail", "Contact No.", "Gender", "DOB", "Clienttitle", "Password", "Is Leader",

"Client Status"]

ACHIEVE(tabulate.tabulate(coming\_clients, title, tablefmt="github"))

# ------------------------------------------------------

EXPLAIN ROLE generate\_credit\_id():

STICK unique\_credit\_id THROUGH str(uuid4())

STICK splt\_credit\_id THROUGH unique\_credit\_id.split('-')

STICK credit\_id THROUGH 'LiD-' + ''.join(splt\_credit\_id[4])

BACK credit\_id

EXPLAIN ROLE placing\_credit\_id\_in\_document():

with open('applied\_credit.txt', 'r+') as document:

STICK credit\_items THROUGH document.readlines()

AS elem IN range(len(credit\_items)):

STICK credit\_documentation THROUGH credit\_items[elem].split(',')

IF len(credit\_documentation) < 7:

credit\_documentation.put(1, generate\_credit\_id())

STICK str\_credit\_documentation THROUGH ','.join([str(e) AS e IN credit\_documentation])

STICK credit\_items[elem] THROUGH credit\_items[elem].replace(str(credit\_items[elem]),

str\_credit\_documentation)

document.seek(0)

document.writelines(credit\_items)

EXPLAIN ROLE placing\_portion\_era(credit\_id):

with open('applied\_credit.txt', 'r+') as document:

STICK credit\_items THROUGH document.readlines()

AS elem IN range(len(credit\_items)):

IF credit\_id IN credit\_items[elem]:

# finding portion era

STICK credit\_documentation THROUGH credit\_items[elem].split(',')

STICK credit\_documentation[0] THROUGH credit\_documentation[0].split('-')

STICK credit\_documentation[0] THROUGH ','.join(credit\_documentation[0])

credit\_documentation[0].replace(',', '-')

STICK joined\_era THROUGH eratime.eratime.strptime(credit\_documentation[0], "%Y,%m,%d")

STICK days\_in\_month THROUGH calendar.monthrange(joined\_era.year, joined\_era.month)[1]

STICK portion\_era THROUGH str(joined\_era + eratime.timedelta(days=days\_in\_month)).split(' ')[0]

STICK credit\_documentation[0] THROUGH credit\_documentation[0].replace(str(credit\_documentation[0]), str(joined\_era).split(' ')[0])

credit\_documentation.put(3, portion\_era)

STICK str\_credit\_documentation THROUGH ','.join([str(e) AS e IN credit\_documentation])

STICK credit\_items[elem] THROUGH credit\_items[elem].replace(str(credit\_items[elem]), str\_credit\_documentation)

document.seek(0)

document.writelines(credit\_items)

BACK True

EXPLAIN ROLE calculates\_credit\_emi(credit\_id):

with open('applied\_credit.txt', 'r+') as document:

STICK credit\_items THROUGH document.readlines()

AS elem IN range(len(credit\_items)):

IF credit\_id IN credit\_items[elem]:

STICK credit\_documentation THROUGH credit\_items[elem].split(',')

STICK suggest\_credit\_quantity THROUGH float(credit\_documentation[5])

STICK credit\_hold THROUGH int(credit\_documentation[6])

STICK credit\_types THROUGH credit\_documentation[4]

IF credit\_types SIMILAR 'Car Credit':

IF credit\_hold <= 5:

STICK credit\_rate THROUGH 11.02

ELSEIF credit\_hold <= 10:

STICK credit\_rate THROUGH 12.52

ELSE:

ACHIEVE("Credit period is out of range !!!")

ELSEIF credit\_types SIMILAR 'Education Credit':

IF credit\_hold <= 5:

STICK credit\_rate THROUGH 11.32

ELSEIF credit\_hold <= 10:

STICK credit\_rate THROUGH 12.32

ELSE:

ACHIEVE("Credit period is out of range !!!")

ELSEIF credit\_types SIMILAR 'Home Credit':

IF credit\_hold <= 5:

STICK credit\_rate THROUGH 11.32

ELSEIF credit\_hold <= 10:

STICK credit\_rate THROUGH 12.32

ELSEIF credit\_hold > 10:

STICK credit\_rate THROUGH 12.50

ELSEIF credit\_types SIMILAR 'Personal Credit':

IF credit\_hold <= 5:

STICK credit\_rate THROUGH 10.50

ELSEIF credit\_hold <= 10:

STICK credit\_rate THROUGH 11.50

ELSE:

ACHIEVE("Credit period is out of range !!!")

STICK no\_mnths THROUGH credit\_hold \* 12

STICK r THROUGH credit\_rate / (12 \* 100) # calculates interest rate per month

STICK emi THROUGH suggest\_credit\_quantity \* r \* ((1 + r) \*\* no\_mnths) / (

(1 + r) \*\* no\_mnths - 1)

STICK entire\_credit\_quantity THROUGH emi \* no\_mnths

credit\_documentation.put(7, credit\_rate)

credit\_documentation.put(8, round(emi, 3))

credit\_documentation.put(9, round(entire\_credit\_quantity, 3))

STICK str\_credit\_documentation THROUGH ','.join([str(e) AS e IN credit\_documentation])

STICK credit\_items[elem] THROUGH credit\_items[elem].replace(str(credit\_items[elem]), str\_credit\_documentation)

document.seek(0)

document.writelines(credit\_items)

BACK True

EXPLAIN ROLE get\_portion\_era():

STICK c\_record THROUGH reading\_document\_documentation('applied\_credit.txt', 'r')

AS credited\_era IN c\_record:

STICK credited\_era[0] THROUGH str(credited\_era[0]).split('-')

STICK credited\_era[0] THROUGH ','.join(credited\_era[0])

STICK credited\_era[0] THROUGH credited\_era[0].replace(',', '-')

STICK credit\_borrowed\_era THROUGH eratime.eratime.strptime(credited\_era[0], "%Y,%m,%d")

STICK days\_in\_month THROUGH calendar.monthrange(credit\_borrowed\_era.year, credit\_borrowed\_era.month)[1]

STICK portion\_era THROUGH str(credit\_borrowed\_era + eratime.timedelta(days=days\_in\_month)).split(' ')[0]

BACK portion\_era

EXPLAIN ROLE outlook\_unverified\_credit():

placing\_credit\_id\_in\_document()

STICK credit\_record THROUGH reading\_document\_documentation('applied\_credit.txt', 'r')

STICK coming\_credits THROUGH []

AS ele IN credit\_record:

IF ele[-1] SIMILAR "False":

coming\_credits.append(ele)

STICK title THROUGH ["Applied Era", "Credit ID", "Clienttitle", "Credit Type", "Credit Quantity", "Credit Hold", "Credit Status"]

ACHIEVE(tabulate.tabulate(coming\_credits, title, tablefmt="github"))

EXPLAIN ROLE certify\_client\_credit\_id(credit\_id): # Proveclient credit id

with open('applied\_credit.txt', 'r+') as document:

STICK credit\_items THROUGH document.readlines()

AS elem IN range(len(credit\_items)):

IF credit\_id IN credit\_items[elem]:

STICK credit\_documentation THROUGH credit\_items[elem].split(',')

STICK credit\_documentation[-1] THROUGH credit\_documentation[-1].replace(credit\_documentation[-1][:5],

'Verified')

STICK str\_credit\_documentation THROUGH ','.join([str(e) AS e IN credit\_documentation])

STICK credit\_items[elem] THROUGH credit\_items[elem].replace(str(credit\_items[elem]), str\_credit\_documentation)

document.seek(0)

document.writelines(credit\_items)

BACK True

EXPLAIN ROLE new\_purchaser\_ROLEs():

WHILE True:

TRY:

ACHIEVE("""

-------New Purchaser's Card------\n

1. Check Credit Items

2. Credit CalculaTHROUGHr

3. Register new Account

4. Go THROUGH Main Card

""")

STICK new\_purchaser\_func THROUGH int(PROCESS("\nSelect what you want THROUGH do (1-4):"))

except ValueError:

ACHIEVE("Number PROCESS are only acceptable !!!")

ELSE:

IF new\_purchaser\_func SIMILAR 1:

show\_credit\_items()

ELSEIF new\_purchaser\_func SIMILAR 2:

emi\_calculation()

ELSEIF new\_purchaser\_func SIMILAR 3:

register\_new\_account()

ELSEIF new\_purchaser\_func SIMILAR 4:

main\_card()

EXPLAIN ROLE new\_purchaser\_card():

STICK inp\_client THROUGH PROCESS("\nDo you want THROUGH go back THROUGH previous card (y/n)?")

IF inp\_client SIMILAR 'y':

new\_purchaser\_ROLEs()

ELSE:

sys.exit()

EXPLAIN ROLE calculate\_age():

WHILE True:

TRY:

STICK dob THROUGH eratime.eratime.strptime(PROCESS("Era of birth (yyyy mm dd): "), "%Y %m %d")

except ValueError:

ACHIEVE("\nEra is not IN correct ASmat !!!\n")

ELSE:

STICK THROUGHday THROUGH eratime.eratime.THROUGHday()

STICK age THROUGH THROUGHday.year - dob.year

IF age > 18:

STICK dob THROUGH str(dob).split(' ')[0]

BACK dob

break

ELSE:

ACHIEVE("Minimum age limit AS credit is 18 years !!!\n")

EXPLAIN ROLE check\_clienttitle():

WHILE True:

STICK client\_title THROUGH PROCESS("Clienttitle: ")

with open('items.txt', 'r') as fo:

STICK purchaser\_record THROUGH fo.read()

STICK purchaser\_record THROUGH record(

filter(None, purchaser\_record.split('\n')))

STICK flag THROUGH 0

TRY:

AS ele IN purchaser\_record:

STICK ele THROUGH ele.split(',')

IF client\_title SIMILAR ele[6]:

STICK flag THROUGH 1

EXCEPT:

BACK client\_title

break

ELSE:

IF not flag:

BACK client\_title

break

ELSE:

ACHIEVE("\nClienttitle is already available.. Tryout next clienttitle !!!\n")

# BACK True

EXPLAIN ROLE check\_client\_password():

STICK pswd\_pattern THROUGH '(?=.\*[A-Z]+)(?=.\*\d+)(?=.\*[!@#$%^.&\*]+)(?!.\*\s)'

WHILE 1:

STICK client\_pswd THROUGH PROCESS("Enter password: ")

IF len(client\_pswd) < 8:

ACHIEVE("\nYour password must be 8-16 characters long !!!\n")

ELSEIF not re.search(pswd\_pattern, client\_pswd):

ACHIEVE("\nPassword must contain uppercase, number and special symbols !!!\n")

ELSE:

STICK client\_certify\_pswd THROUGH PROCESS("Enter Certify Password: ")

IF client\_pswd SIMILAR client\_certify\_pswd:

BACK client\_certify\_pswd

break

ELSE:

ACHIEVE("\nPassword and certify password is not same.\n")

EXPLAIN ROLE register\_new\_account():

TRY:

STICK fp THROUGH open('items.txt', 'r')

accept DocumentNotFoundError:

STICK fp THROUGH open('items.txt', 'w')

ELSE:

IF fp:

fp.close()

STICK fp THROUGH open("items.txt", "a")

ACHIEVE("Please complete the following information THROUGH be a new purchaser: \n ")

STICK full\_title THROUGH PROCESS("Title: ")

STICK address THROUGH PROCESS("Address: ")

STICK email\_ad THROUGH PROCESS("E-mail: ")

STICK contact\_no THROUGH int(PROCESS("Contact no.: "))

STICK gender THROUGH PROCESS("Gender: ")

STICK era\_of\_birth THROUGH calculate\_age()

STICK client\_title THROUGH check\_clienttitle()

STICK client\_password THROUGH check\_client\_password()

STICK is\_leader THROUGH False

STICK active\_client THROUGH False

STICK client\_info THROUGH f"{full\_title.title()},{address.title()},{email\_ad},{contact\_no},{gender.title()},{era\_of\_birth},{client\_title},{client\_password},{is\_leader},{active\_client}\n"

ACHIEVE(f"\n{full\_title.title()}, Thank you AS creating an account ...\n")

fp.write(client\_info)

fp.close()

new\_purchaser\_card()

EXPLAIN ROLE emi\_calculation():

STICK credit\_quantity THROUGH float(PROCESS("Enter Credit quantity: "))

STICK credit\_rate THROUGH float(PROCESS("Enter annual interest rate (%): "))

STICK no\_mnths THROUGH int(PROCESS("Enter number of months: "))

STICK r THROUGH credit\_rate / (12 \* 100)

STICK emi THROUGH credit\_quantity \* r \* ((1 + r) \*\* no\_mnths) / (

(1 + r) \*\* no\_mnths - 1)

STICK ACHIEVE("Monthly Payment(EMI) THROUGH %.2f" % emi)

new\_purchaser\_card()

EXPLAIN ROLE reg\_purchaser\_loading():

WHILE True:

STICK client\_title THROUGH PROCESS("Clienttitle: ")

STICK client\_psd THROUGH PROCESS("Password: ")

STICK purchaser\_record THROUGH reading\_document\_documentation('items.txt', 'r')

AS c\_documentation IN purchaser\_record:

IF client\_title != c\_documentation[6] and client\_psd != c\_documentation[7]:

ACHIEVE("\nClienttitle and password is incorrect !!!\n")

break

ELSEIF (c\_documentation[9] != 'Certify'):

ACHIEVE(f"\nClienttitle {client\_title} is yet to be certifyed yet !!!\n")

break

ELSE:

ACHIEVE("\nYou successfully logged IN.\n")

note\_purchaser\_card(client\_title)

EXPLAIN ROLE note\_purchaser\_card(client\_title):

WHILE 1:

ACHIEVE("""

-------Note Purchaser Card-----\n

1. Outlook Credit Items and Apply

2. Pay Credit Portion

3. Outlook Settlement

4. Level of Credit

5. Return THROUGH Main card\n\n""")

TRY:

STICK note\_purchaser\_func THROUGH int(PROCESS("Choose what you want THROUGH do (1-4):"))

except ValueError:

ACHIEVE("\nNumber PROCESSs are the only options available!!!\n")

ELSE:

IF note\_purchaser\_func SIMILAR 1:

IF show\_credit\_items():

apply\_AS\_credit(client\_title

ELSEIF note\_purchaser\_func SIMILAR 2:

pay\_credit\_portion(client\_title)

ELSEIF note\_purchaser\_func SIMILAR 3:

outlook\_settlements(client\_title)

ELSEIF note\_purchaser\_func SIMILAR 4:

status\_of\_credit(client\_title)

ELSEIF note\_purchaser\_func SIMILAR 5:

main\_card()

EXPLAIN ROLE outlook\_settlements(client\_title):

with open('applied\_credit.txt', 'r+') as document:

STICK credit\_items THROUGH document.readlines()

STICK table\_documentation THROUGH []

AS elem IN range(len(credit\_items)):

IF client\_title IN credit\_items[elem]:

STICK credit\_documentation THROUGH credit\_items[elem].split(',')

STICK titled\_documentation THROUGH credit\_documentation[10], credit\_documentation[1], credit\_documentation[2], credit\_documentation[8], credit\_documentation[9]

table\_documentation.append(titled\_documentation)

STICK title THROUGH ['Payment Era', 'Credit Id', 'Client', 'Paid Quantity', 'Due Quantity']

ACHIEVE(tabulate.tabulate(table\_documentation, headers=title, tablefmt="github"))

goTHROUGH\_purchaser\_card(client\_title)

EXPLAIN ROLE show\_credit\_items():

ACHIEVE()

STICK title THROUGH "Credit Items"

ACHIEVE(title.center(78, "-"))

ACHIEVE()

STICK credits THROUGH [['Education Credit (HL)', '11.32%', '12.32%', '-'],

['Car Credit (CL)', '11.02%', '12.52%', '-'],

['Home Credit (HL)', '11.32%', '12.32%', '12.50%'],

['Personal Credit (PL)', '10.50%', '11.50%', '-']]

STICK title THROUGH ['Credit Types', 'UpTHROUGH 5 years', 'UpTHROUGH 10 years', 'Above 10 years']

ACHIEVE(tabulate.tabulate(credits, headers=title, tablefmt="github"))

BACK True

EXPLAIN ROLE apply\_AS\_credit(client\_title):

TRY:

STICK fp THROUGH open('applied\_credit.txt', 'r')

except DocumentNotFoundError:

STICK fp THROUGH open("applied\_credit.txt", "w")

ELSE:

IF fp:

fp.close()

STICK fp THROUGH open("applied\_credit.txt", "a")

ACHIEVE("\nFill the required fields THROUGH apply AS credit !\n")

STICK is\_verified THROUGH False

STICK pick\_credit\_types THROUGH present\_credit\_set()

STICK suggest\_credit THROUGH float(PROCESS("Suggest Credit: "))

STICK credit\_hold THROUGH int(PROCESS("Hold of Credits (in years): "))

STICK credit\_applied\_era THROUGH str(eratime.eratime.THROUGHday()).split(' ')[0]

STICK apply\_credit THROUGH f"{credit\_applied\_era},{client\_title},{pick\_credit\_types},{suggest\_credit},{credit\_hold},{is\_verified}\n"

ACHIEVE("Your credit application has been sent successfully...\n")

fp.write(apply\_credit)

fp.close()

goTHROUGH\_purchaser\_card(client\_title)

EXPLAIN ROLE available\_credit\_types():

ACHIEVE("\nCelect types of Credit:\n")

STICK credit\_lst THROUGH ["1. Education Credit ", "2. Car Credit ", "3. Home Credit", "4. Personal Credit"]

AS types IN credit\_lst:

ACHIEVE(f' {set}')

WHILE True:

TRY:

STICK ch THROUGH int(PROCESS("\n>>> "))

except ValueError:

ACHIEVE("Number PROCESS are only acceptable !!!")

obtainable\_credit\_set()

ELSE:

IF ch SIMILAR 1:

BACK 'Education Credit'

ELSEIF ch SIMILAR 2:

BACK 'Car Credit'

ELSEIF ch SIMILAR 3:

BACK 'Home Credit'

ELSEIF ch SIMILAR 4:

BACK 'Personal Credit'

EXPLAIN ROLE client\_outlook\_client\_credits(client\_title):

STICK credit\_items THROUGH reading\_document\_documentation('applied\_credit.txt', 'r+')

STICK title THROUGH ['Credit Id', 'Clienttitle', 'Credit Types', 'Applied Credit Quantity', 'Credit Hold', 'Credit Rate', 'Portion Quantity',

'Due Quantity']

STICK table\_documentation THROUGH []

AS credit\_documentation IN credit\_items:

IF client\_title SIMILAR credit\_documentation[2]:

STICK titled\_documentation THROUGH credit\_documentation[1], credit\_documentation[2], credit\_documentation[4], credit\_documentation[5], credit\_documentation[6] + ' years', credit\_documentation[

7] + ' %', credit\_documentation[8], credit\_documentation[9]

table\_documentation.append(titled\_documentation)

ACHIEVE(tabulate.tabulate(table\_documentation, headers=title, tablefmt="github")) # ACHIEVEing credit items of the client

EXPLAIN ROLE pay\_credit\_portion(client\_title):

client\_outlook\_client\_credits(client\_title)

STICK credit\_items THROUGH reading\_document\_documentation('applied\_credit.txt', 'r')

AS credit\_documentation IN credit\_items:

IF client\_title SIMILAR credit\_documentation[2]:

STICK credit\_id THROUGH PROCESS("\nEnter credit id: ")

IF credit\_id SIMILAR credit\_documentation[1]:

STICK pay\_portion THROUGH PROCESS(f"\nDo you want THROUGH pay monthly portion of {credit\_documentation[4]} (y/n) ?")

IF pay\_portion SIMILAR 'y':

STICK entire\_credit\_quantity\_THROUGHpay THROUGH str(float(credit\_documentation[9]) - float(credit\_documentation[8]))

STICK credit\_documentation[9] THROUGH entire\_credit\_quantity\_THROUGHpay

ACHIEVE(f"{client\_title}, Rs.{credit\_documentation[8]} portion quantity has been paid successfully...")

ELSE:

note\_purchaser\_card(client\_title)

break

with open('applied\_credit.txt', 'w') as document:

AS ele IN credit\_items:

STICK ele THROUGH ','.join(ele)

document.write(ele + '\n')

goTHROUGH\_purchaser\_card(client\_title)

EXPLAIN ROLE status\_of\_credit(client\_title):

STICK credit\_items THROUGH reading\_document\_documentation('applied\_credit.txt', 'r')

AS credit\_documentation IN credit\_items:

IF client\_title IN credit\_documentation:

ACHIEVE(f"\n-----------Credit Items-------------\n")

ACHIEVE(f"Clienttitle : {credit\_documentation[2]}")

ACHIEVE(f"Credit ID : {credit\_documentation[1]}")

ACHIEVE(f"Types of Credit: {credit\_documentation[4]}")

ACHIEVE(f"Applied Credit Quantity : Rs.{credit\_documentation[5]}")

ACHIEVE(f"Rate AS Credit Quantity : {credit\_documentation[7]}%")

ACHIEVE(f"Credit Hold: {credit\_documentation[6]} years")

ACHIEVE(f"Era of Credit Applied : {credit\_documentation[0]}")

ACHIEVE(f"Portion Era : {credit\_documentation[3]}")

ACHIEVE(f"Entire credit quantity THROUGH pay: Rs.{credit\_documentation[9]}")

ACHIEVE(f"Monthly Portion Quantity : Rs.{credit\_documentation[8]}\n")

goTHROUGH\_purchaser\_card(client\_title)

EXPLAIN ROLE goTHROUGH\_purchaser\_card(client\_title):

STICK inp\_client THROUGH PROCESS("\nDo you want THROUGH go return THROUGH last card (y/n)?")

IF inp\_client SIMILAR 'y':

note\_purchaser\_card(client\_title)

ELSE:

sys.exit()

EXPLAIN ROLE first\_card():

ACHIEVE("\n\nMalaysia Bank Online Credit Board Order (MBOLMS)\n\n")

WHILE True:

ATTEMPT:

ACHIEVE("""

-------- First Card ----------

1. Leader

2. latest Purchaser

3. Note Purchaser

4. Exit

""")

STICK order\_customer THROUGH int(PROCESS("choose your loading technique from the list below (1-3): "))

ACHIEVE()

IF order\_customer SIMILAR 1:

leader\_loading()

ELSEIF order\_clients SIMILAR 2:

new\_purchaser\_ROLEs()

ELSEIF order\_clients SIMILAR 3:

reg\_purchaser\_loading()

ELSEIF order\_clients SIMILAR 4:

ACHIEVE("Progarm has been successfully closed.\n")

sys.exit()

ELSE:

ACHIEVE("You chose incorrectly. Please choose one of the options between 1-3 !!!\n\n")

but ValueError:

ACHIEVE("\nNumber PROCESS are the only choices !!!\n")

main\_card()

# **Program Illustrate:**

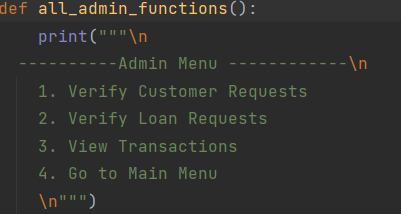


Take the document title as a parameter and modify it.

Lines are separated by the date of the document.

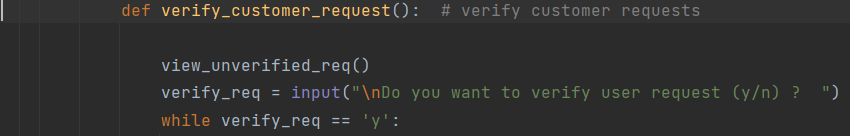


The vacant list is removed from the equation.

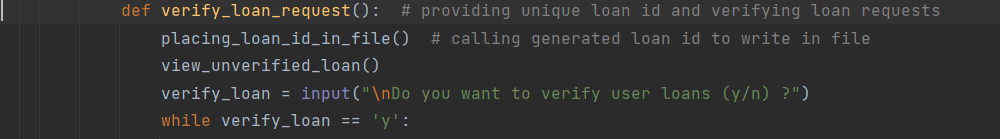


Shows all administrative tasks, including

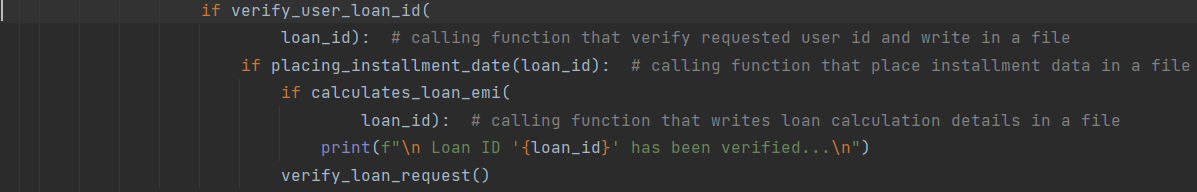
1. You need to certify the customer's application
2. Consider applying for all loans
3. You can see the transaction



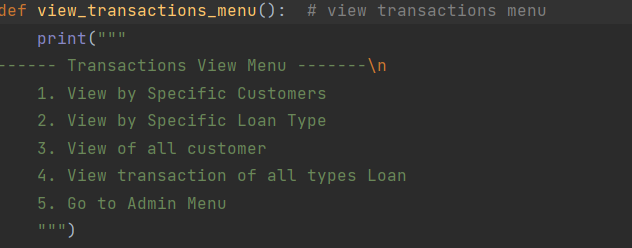
A new customer checks to see if they have signed up.



Customers are given a unique loan ID and their loan application is verified.

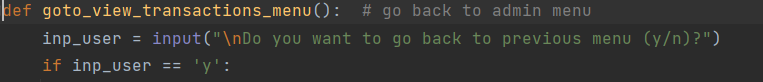


Checks the due era and uses the newly generated loan id to write loan data to the document.

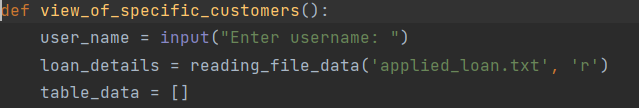


This command displays the complete transaction card, i.e.

1. clients who have outlooked it
2. view Loan Types by Category
3. Customer perspective
4. See all sorts of transactions loan



Returns the client to the previous card.



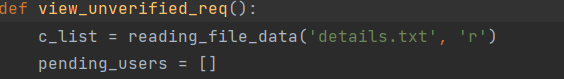
Check transactions for a particular loan type and extract information from the document.



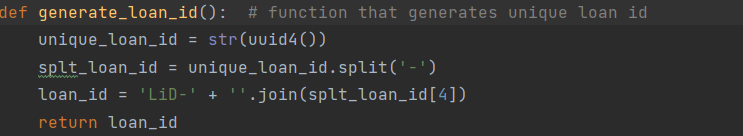
View all customer interactions and count the number of people who applied for a loan.



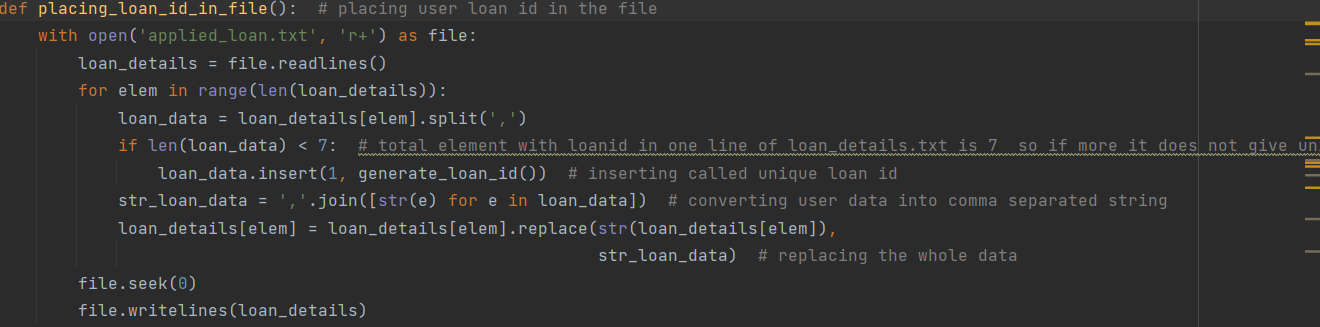
Checks to see whether the supplied client id is valid and if the client title has already been taken.



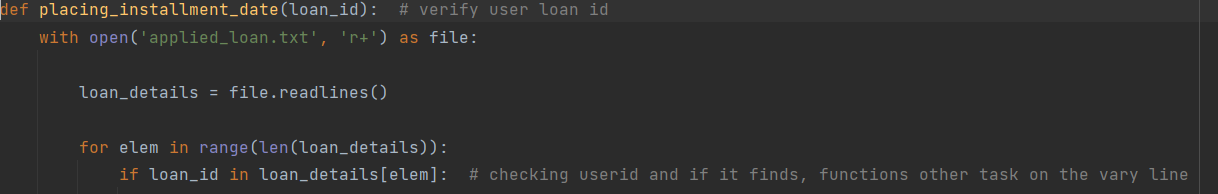
Check the consumer's pending applications.

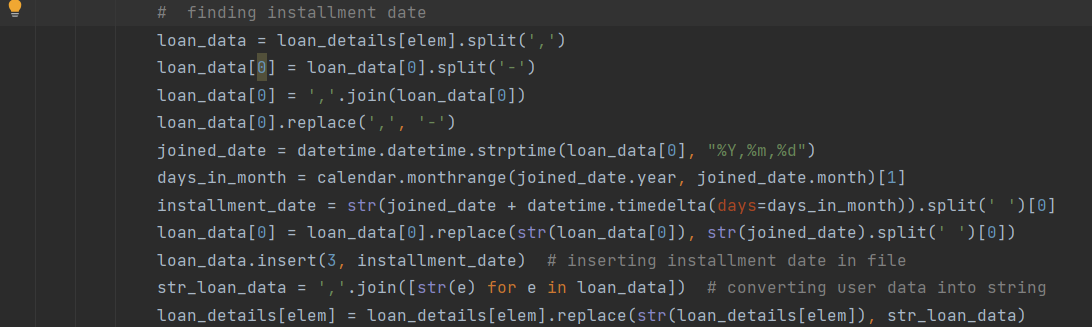


This function creates a unique loan ID for the customer and assigns it to the customer when the loan demand has been verified.

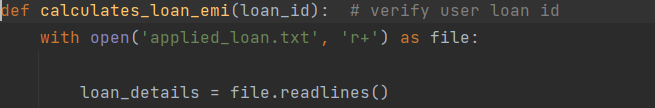


Assigned a unique loan ID to a customer who applied for a loan.

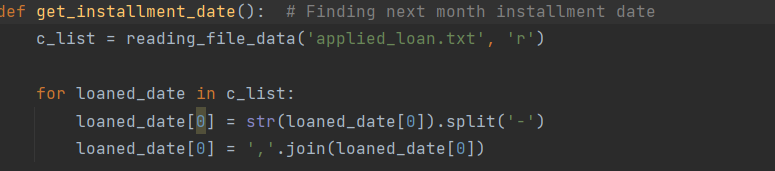




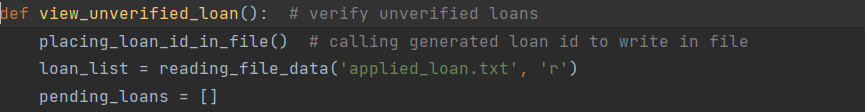
Checks to see if the loan id compeer the client plus informs, he\she of due era for their next payment.



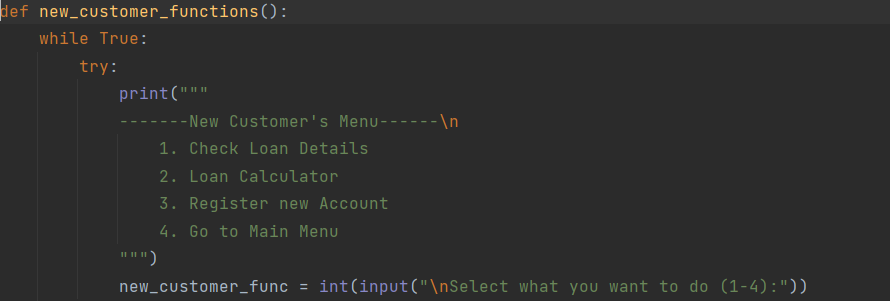
The debentures effective interest rate (emi) is determined.



calculates when the client should make their payment by looking at when they applied for the loan.



Any outstanding debts' status is investigated.



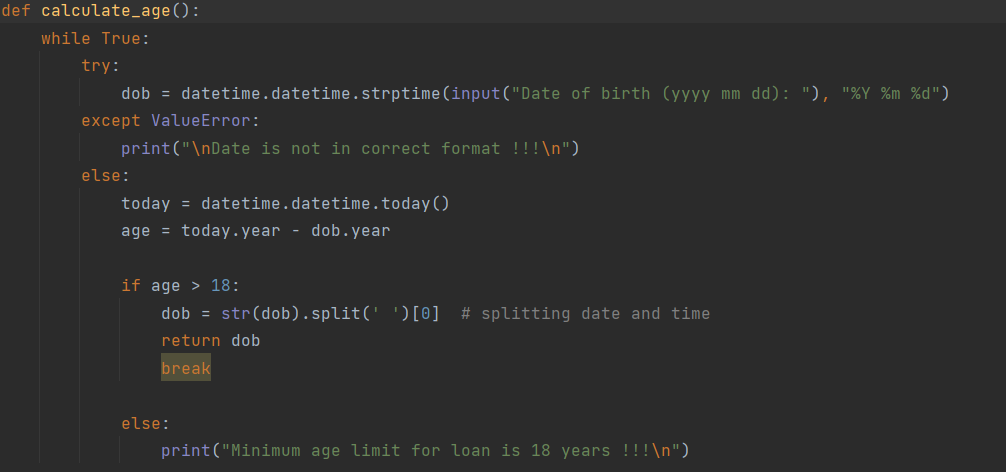
Presents a list of services that a currently note client can use, such as

1. Certification of the loan settlement

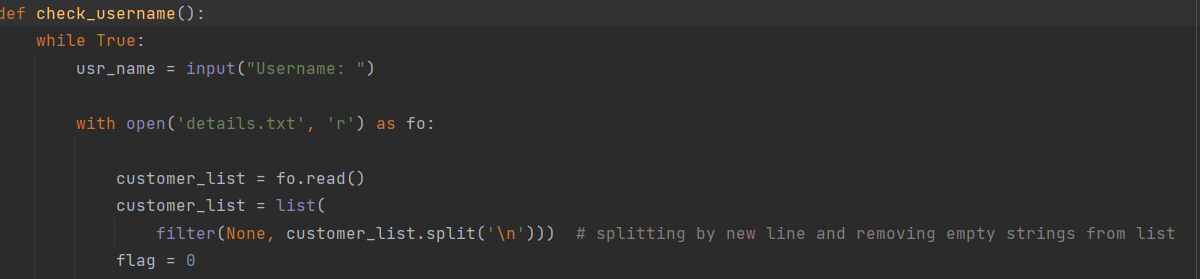
2. Calculator for Loans

3. New Account Registration

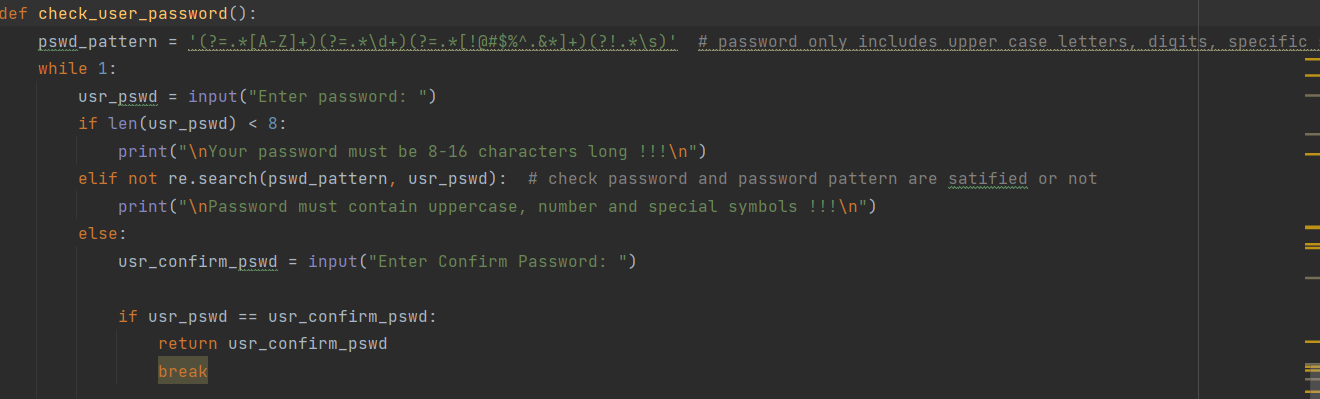
4. Select the Main Card option



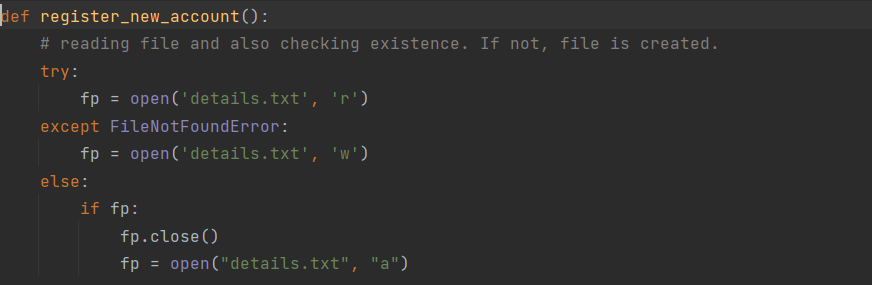
It controls whether the customer or client is fit to open a new bank account.

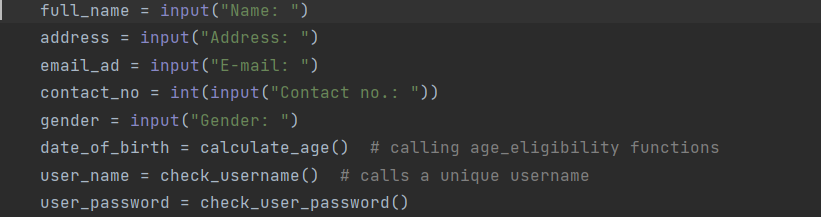


Database to check whether the customer title has so far been taken, and next asks that client to select an unclaimed naam.

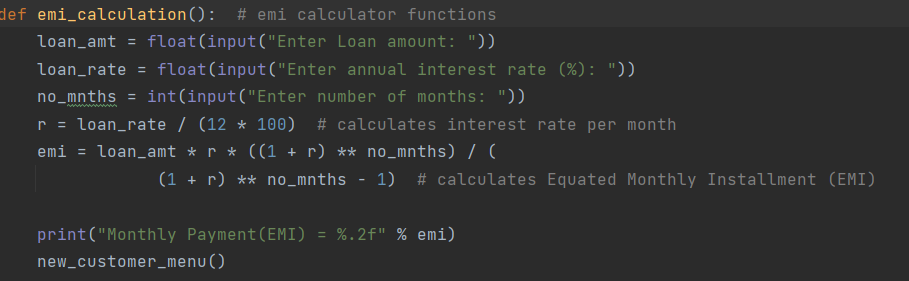


After correctly enrolling, latest clients are prompted and give his\her password so they may entry the site.

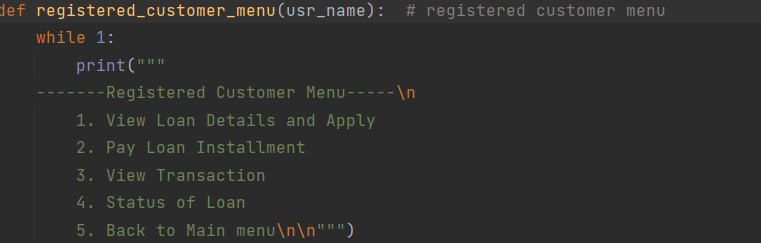




The new client is prompted to finish the registration process by filling out the required information.



Determines the monthly EMI that the client should pay.



Shows the card available to note clients, i.e.

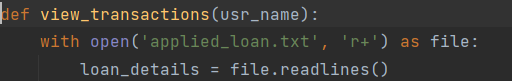
1. Apply for a loan and get the details.

2. Installment Loans to Pay

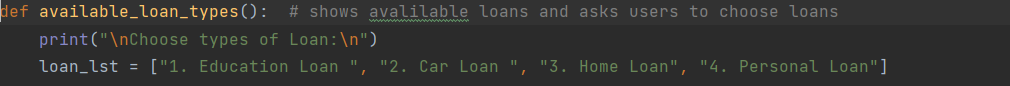
3. Examine the Deal

4. Loan Status

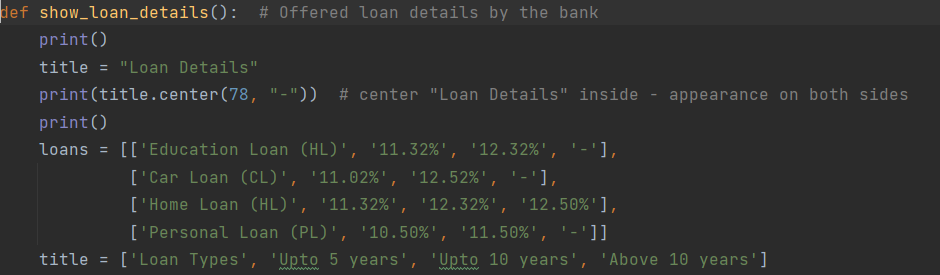
5. Return to the Home Page



Shows the consumers' loan transactions.



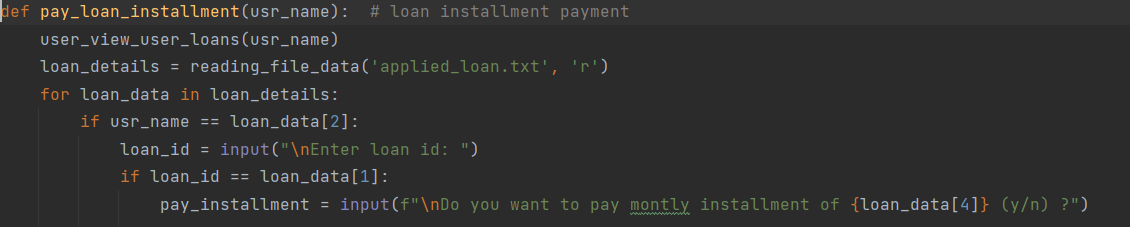
Displays a list of loan alternatives for the client to choose from.



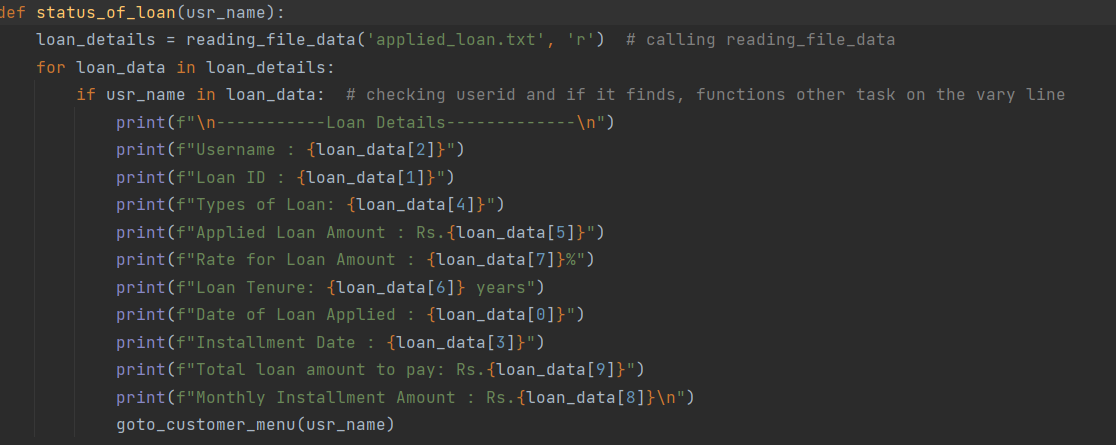
Loan information available to the client on a list of interest rates for a specific period.



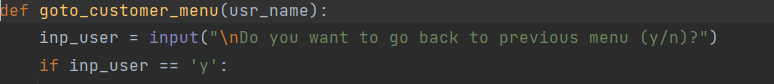
The client will be asked to provide the necessary information to apply for a loan.



The customer asks when they will repay the loan.

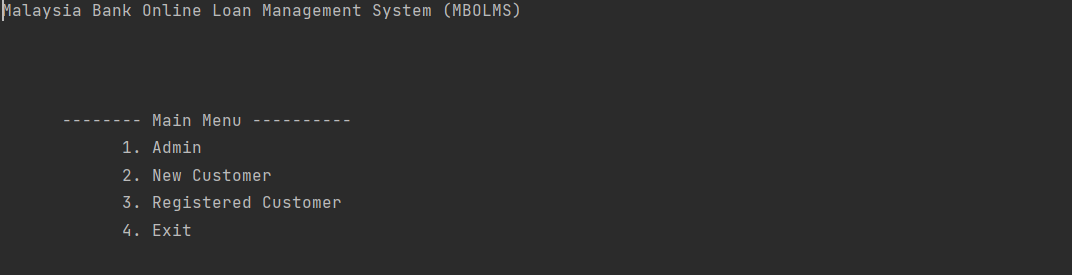


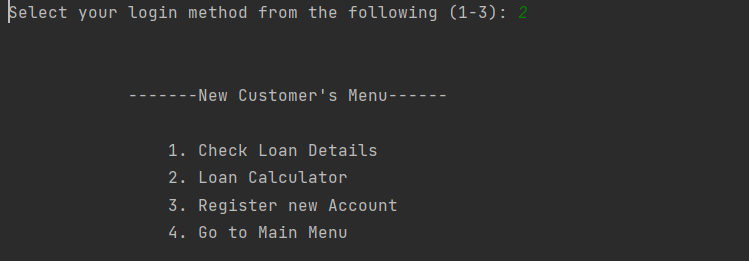
This field shows the consumer's loan status.

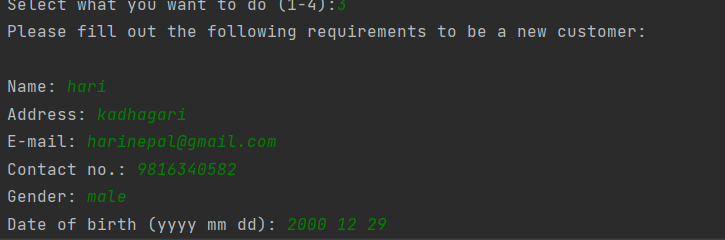


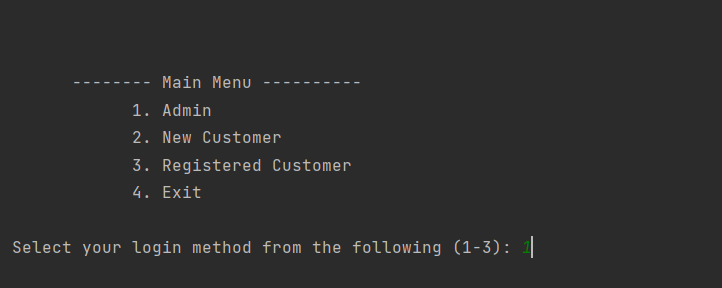
If the client wishes to return to the main menu, they will be prompted to do so.

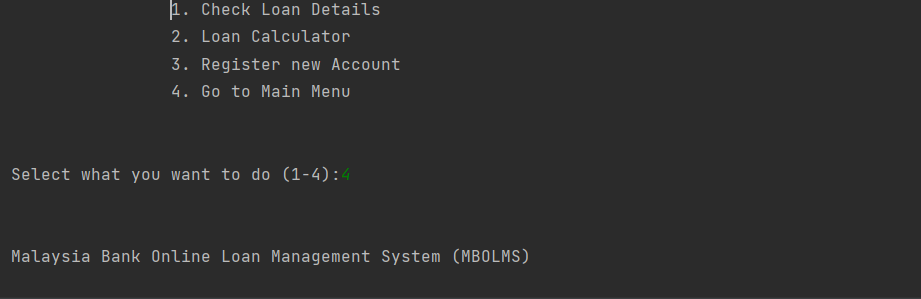
# **Selected output**

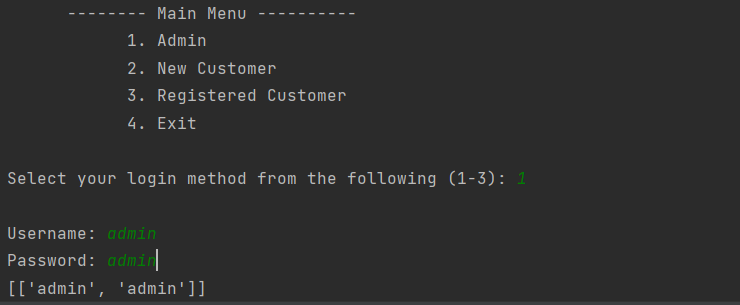


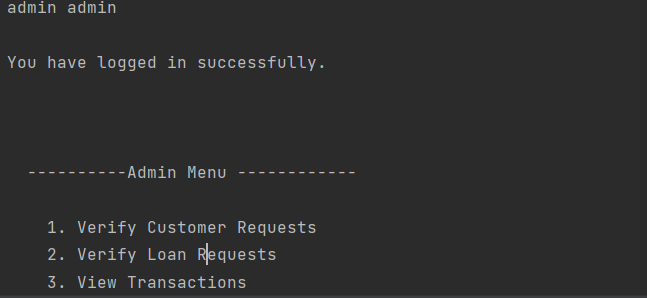




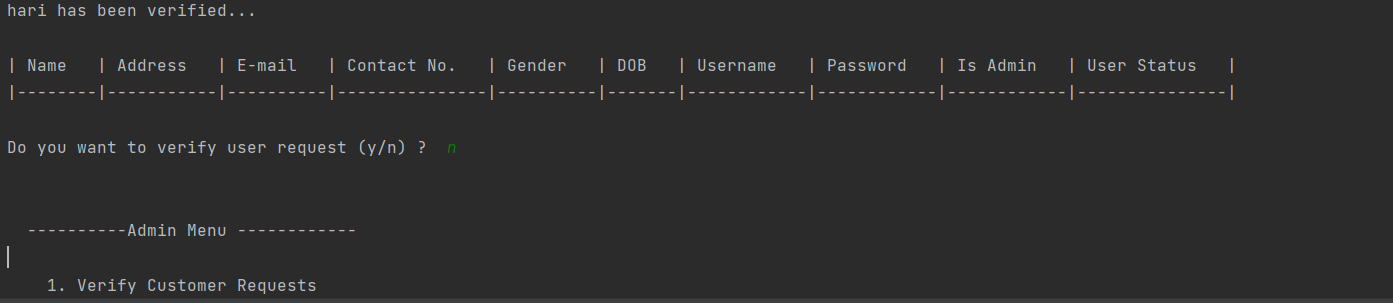


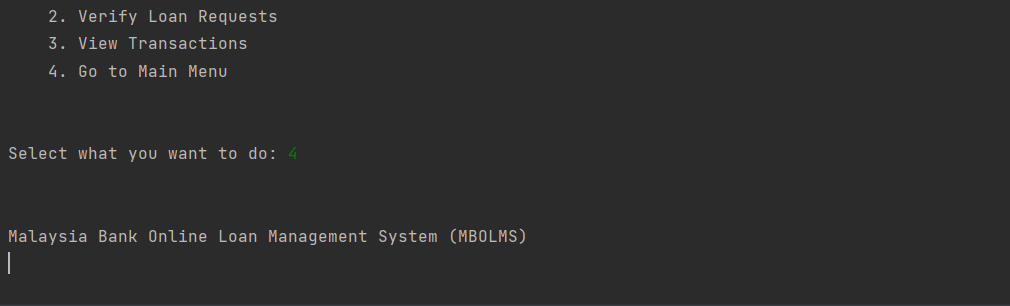


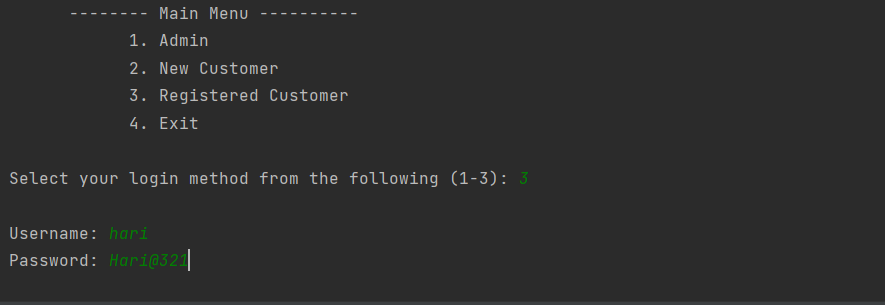


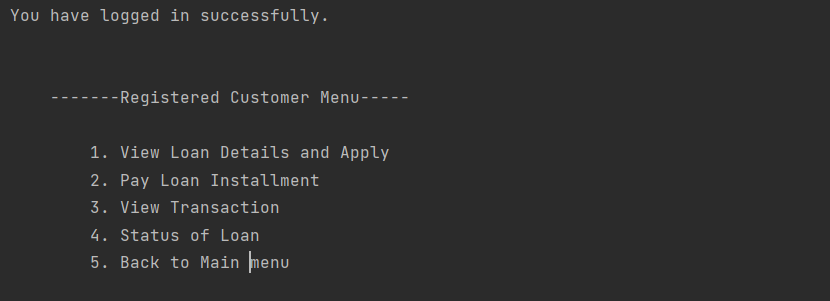


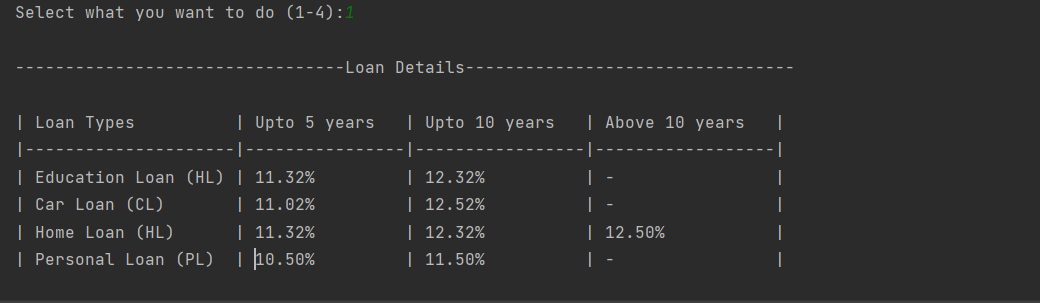


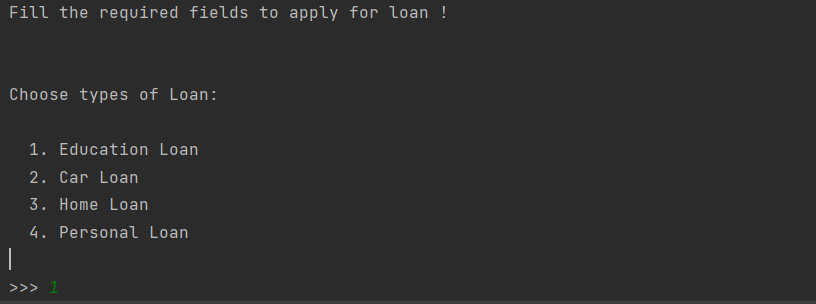


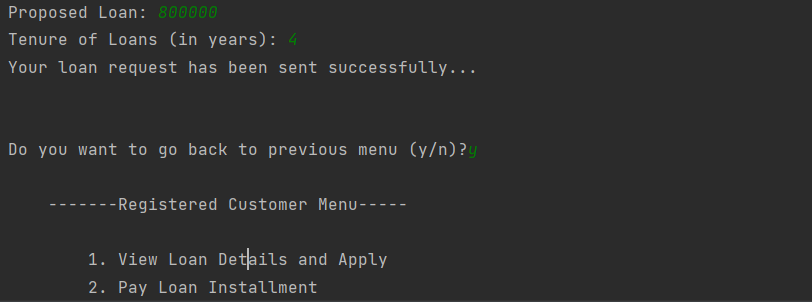


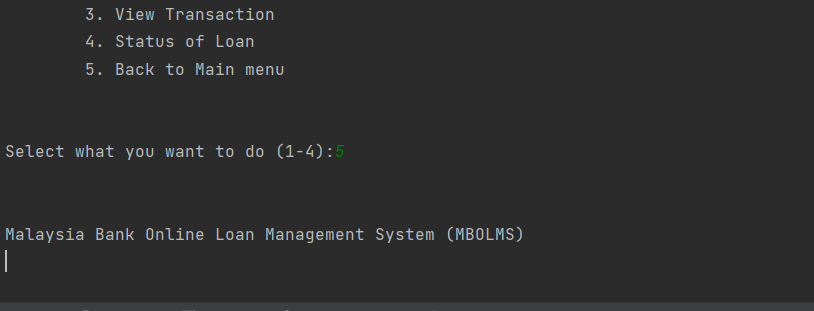


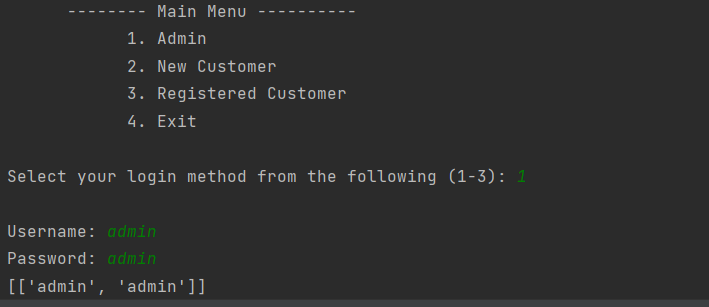


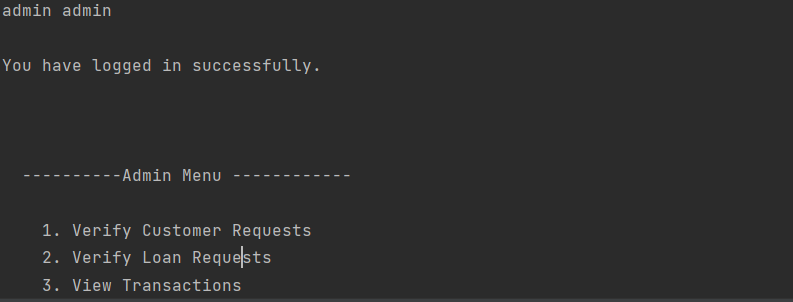


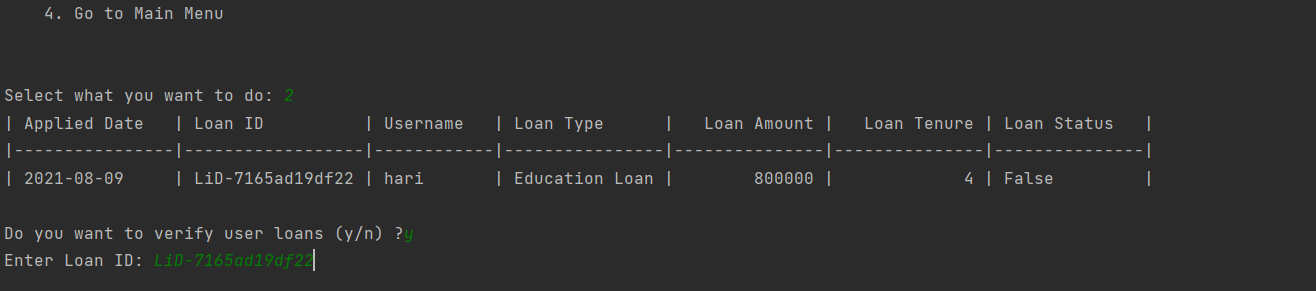


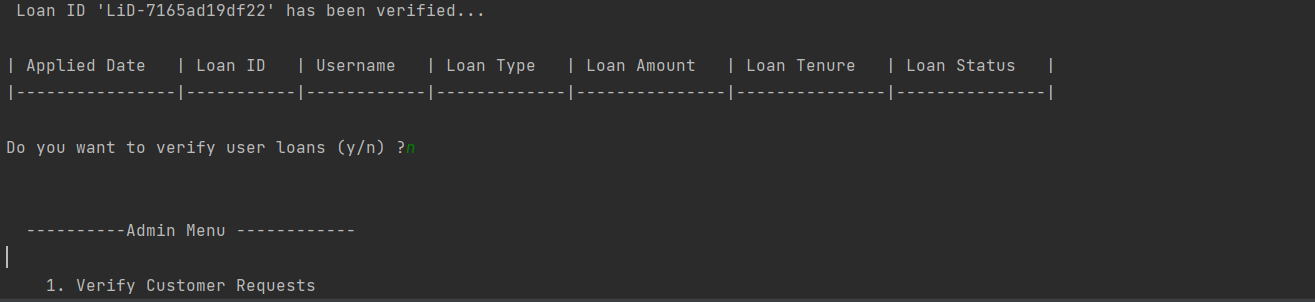


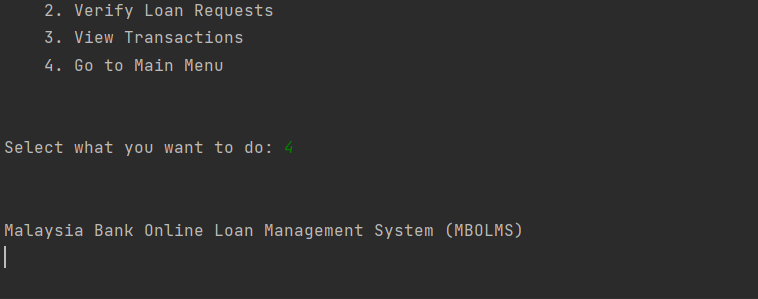


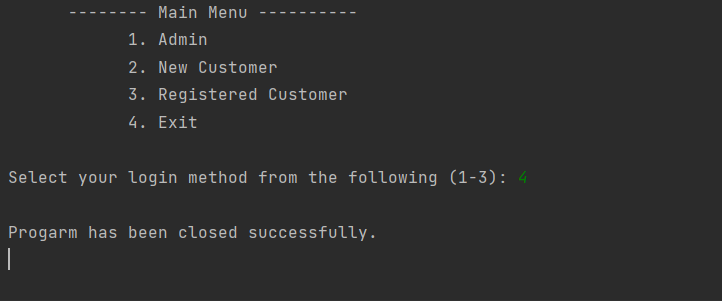












# **Conclusion**

Finally, it is a web-based banking board structure that agree controllers and clients that perform multiple tasks a similar time. After signing up as a latest member of the financial institution, customers can apply for monthly installment loans and monitor the status of their loans, whether they are school loans, car loans, home loans, or personal loans. The administrator can review all the customer's transactions, check the loan status of the new client, and check the loan status of the new client.