

**Fundamental of software development**

**Group Assignment Report**

**Intake:**

**UC1F1809IT(IOT)/IS**

**Group Members:**

**Zhang ZiTeng, TP052096**

**Leong Soong Jun, TP050994**

**Fayyaadh Beeharry, TP050731**

# TABLE OF CONTENT

[TABLE OF CONTENT 3](#_Toc533984778)

[ASSESSMENT CRITERIA 4](#_Toc533984779)

[WORK BREAK DOWN STRUCTURE 5](#_Toc533984780)

[INTRODUCTION 6](#_Toc533984781)

[PSEUDOCODE 7](#_Toc533984782)

[#PROGRAM READ DATA FROM HELLOWORLD.JSON AND DEFINE THE SIZE OF THE DATA 7](#_Toc533984783)

[#PROGRAM SAVE DATA 7](#_Toc533984784)

[#MAIN MENU OF M 8](#_Toc533984785)

[#SUBMENU OF P 8](#_Toc533984786)

[#PROGRAM TO SHOW AND SELECT SHIP SCHEDULE 9](#_Toc533984787)

[#PROGRAM TO SHOW SEAT 11](#_Toc533984788)

[#PROGRAM PRINT TICKET 12](#_Toc533984789)

[#PROGRAM PURCHASE TICKET 13](#_Toc533984790)

[INPUT AND OUTPUT 17](#_Toc533984791)

[ASSUMPTION 22](#_Toc533984792)

[CONCLUSION 23](#_Toc533984793)

# ASSESSMENT CRITERIA

**Individual Component (50%)**

Algorithm (10%)

Presentation & QA (15%)

Understanding of programming constructs (15%)

Contribution (10%)

**Group Component (50%)**

Design (20%)

Module Integration (10%)

Documentation (20%)

# WORK BREAK DOWN STRUCTURE

|  |  |  |
| --- | --- | --- |
| **NAME OF GROUP MEMBERS** | **PERCENTAGE** | **SIGNATURE** |
| **ZHANG ZITENG** | **40%** |  |
| **LEONG SOONGJUN** | **30%** |  |
| **FAYYAADH BEEHARRY** | **30%** |  |

# INTRODUCTION

This group assignment is to build a python program to automate the ticketing system of a small ferry company. The design of the new system incorporates allocating seats to passengers for each trip of the 50-seater ferry, which covers the route from Penang to Langkawi and back daily. The comprises of two decks –upper and main. The upper deck has 10 seats for business class travellers. The main deck has 40 seats for economy class passengers. The ferries of schedule are as follow:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FERRY ID ROUTE SCHEDULE TIME \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Ferry 1,2,3,4 Penang->Langkawi 0 10:00 AM\*

\* Ferry 5,6,7,8 Penang->Langkawi 1 11:00 AM\*

\* Ferry 1,2,3,4 Penang->Langkawi 2 12:00 AM\*

\* Ferry 5,6,7,8 Penang->Langkawi 3 01:00 PM\*

\* Ferry 1,2,3,4 Penang->Langkawi 4 02:00 PM\*

\* Ferry 5,6,7,8 Penang->Langkawi 5 03:00 PM\*

\* Ferry 1,2,3,4 Penang->Langkawi 6 04:00 PM\*

\* Ferry 5,6,7,8 Penang->Langkawi 7 05:00 PM\*

\* Ferry 5,6,7,8 Langkawi->Penang 8 10:00 AM\*

\* Ferry 1,2,3,4 Langkawi->Penang 9 11:00 AM\*

\* Ferry 5,6,7,8 Langkawi->Penang 10 12:00 AM\*

\* Ferry 1,2,3,4 Langkawi->Penang 11 01:00 PM\*

\* Ferry 5,6,7,8 Langkawi->Penang 12 02:00 PM\*

\* Ferry 1,2,3,4 Langkawi->Penang 13 03:00 PM\*

\* Ferry 5,6,7,8 Langkawi->Penang 14 04:00 PM\*

\* Ferry 1,2,3,4 Langkawi->Penang 15 05:00 PM\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# PSEUDOCODE

IMPORT JSON

## #PROGRAM READ DATA FROM HELLOWORLD.JSON AND DEFINE THE SIZE OF THE DATA

BEGIN

DEFINE import\_file()

TRY

WITH OPEN('Hello world.json') AS f

RETURN json.load(f)

EXCEPT FileNotFoundError

data = []

FOR i IN RANGE(16)

data.append([[[0]\*10, [0]\*40],[[0]\*10, [0]\*40],[[0]\*10, [0]\*40],[[0]\*10, [0]\*40]])

RETURN data

ENDFOR

ENDTRY

ENDDEFINE

END

## #PROGRAM SAVE DATA

BEGIN

DEFINE export\_file(data)

WITH OPEN('Hello world.json','w') AS f

json.dumpdata,f

data=import\_file

ENDEFINE

END

## #MAIN MENU OF M

BEGIN

DEFINE M

Print ‘‘‘FERRY TICKETING SYSTEM

P - to Purchase Ticket

V - to View Seating Arrangement

Q - to Quit the system’’’

READ x

RETURN x

ENDDEFINE

END

## #SUBMENU OF P

BEGIN

DEFINE P

Print ‘‘‘PURCHASING MODULE

B - to purchase ticket for Business class

E - to purchase ticket for Economy class

M - to return to Main Menu’’’

READ y

RETURN y

ENDDEFINE

END

#PROGRAM FIND EMPTY SEATS

BEGIN

DEFINE find\_empty\_seat(schedule,Class)

FOR i IN RANGE(4)

IF Class == 'B'

seats=schedule[i][0]

ELSE

seats=schedule[i][1]

ENDIF

ENDFOR

FOR j IN RANGE(len(seats))

IF seats [j]==0

RETURN i , j

ENDIF

ENDFOR

RETURN -1,-1

ENDFOR

ENDDIFINE

END

## #PROGRAM TO SHOW AND SELECT SHIP SCHEDULE

BEGIN

DEFINE V

DOWHILE TRUE

PRINT ‘‘‘

Please select your ship number.

Enter M to go back!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* FERRY ID ROUTE SCHEDULE TIME \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Ferry 1,2,3,4 Penang->Langkawi 0 10:00 AM\*

\* Ferry 5,6,7,8 Penang->Langkawi 1 11:00 AM\*

\* Ferry 1,2,3,4 Penang->Langkawi 2 12:00 AM\*

\* Ferry 5,6,7,8 Penang->Langkawi 3 01:00 PM\*

\* Ferry 1,2,3,4 Penang->Langkawi 4 02:00 PM\*

\* Ferry 5,6,7,8 Penang->Langkawi 5 03:00 PM\*

\* Ferry 1,2,3,4 Penang->Langkawi 6 04:00 PM\*

\* Ferry 5,6,7,8 Penang->Langkawi 7 05:00 PM\*

\* Ferry 5,6,7,8 Langkawi->Penang 8 10:00 AM\*

\* Ferry 1,2,3,4 Langkawi->Penang 9 11:00 AM\*

\* Ferry 5,6,7,8 Langkawi->Penang 10 12:00 AM\*

\* Ferry 1,2,3,4 Langkawi->Penang 11 01:00 PM\*

\* Ferry 5,6,7,8 Langkawi->Penang 12 02:00 PM\*

\* Ferry 1,2,3,4 Langkawi->Penang 13 03:00 PM\*

\* Ferry 5,6,7,8 Langkawi->Penang 14 04:00 PM\*

\* Ferry 1,2,3,4 Langkawi->Penang 15 05:00 PM\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

’’’

ENDDO

ENDDEFINE

READ t

IF t == 'M'

RETURN t

ENDIF

t = int(t)

IF (t < 0) OR (t > 15)

CONTINUE

RETURN t

ENDIF

END

## #PROGRAM TO SHOW SEAT

BEGIN

FROM datetime IMPORT date

stars='\*'\*41

DEFINE print\_seats(data,schedule\_id,ferry\_id)

b,e=data[schedule\_id][ferry\_id]

s=' '

s=s+stars+'\n'

s=s+'\*Ferry ID: ' + str(ferry\_id) + 'Date:' + date.today().strftime('%d %b %Y') + '\*\n'

s=s+stars+'\n'

s=s+'\* Business class \*\n'

s=s+stars+'\n'

FOR j IN RANGE(2)

FOR i IN RANGE(5)

s=s+'\* '+str(b[j\*5+i])+' '

ENDFOR

s=s+'\* \n'

s=s+stars+'\n'

s=s+'\* Economy class \*\n'

s=s+stars+'\n'

ENDDO

FOR J IN RANGE(8)

FOR I IN RANGE(5)

s=s+'\* '+str(e[J\*5+I])+' '

ENDFOR

s=s+'\*\n'

s=s+stars+'\n'

PRINT(s)

ENDFOR

ENDDEFINE

END

## #PROGRAM PRINT TICKET

BEGIN

DEFINE print\_ticket(c, i,schedule\_id,ferry\_id)

READ name (‘Please enter your name’)

READ passport\_no (‘Please enter your passport NO.’)

IF (schedule\_id <= 7) AND (schedule\_id % 2) OR (schedule\_id > 7) AND (schedule\_id % 2 == 0: ferry\_id += 4)

ENDIF

PRINT(You have purchased a business class)

PRINT(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

PRINT(\* WELCOME TO THE TRIP! \*)

PRINT(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

PRINT(\*Schedule NO: {: <15}\*.format(schedule\_id))

PRINT(\*Ferry ID: {: <15}\*.format(ferry\_id+1))

PRINT(\*Name: {: <15}\*.format(name))

PRINT(\*Passport NO: {: <15}\*.format(passport\_no))

PRINT(\*Seat Number: {: <15}\*.format(c + str(i+1)))

PRINT(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

ENDDEFINE

END

## #PROGRAM PURCHASE TICKET

BEGIN

WHILE True

userinput =M()

IF userinput=='P'

WHILE True

schedule\_id = V()

IF schedule\_id =='M'

BREAK

ENDIF

PRINT ‘Please select which kinds of class you want to purchase!’

userinput=P()

IF userinput=='B'

ferry\_id, n=find\_empty\_seat(data[schedule\_id], 'B')

IF n == -1

PRINT ‘‘‘Business Class is full!

Press E to select an Economy seat!

Press M to return to the previous menu!’’’

userinput = READ()

IF userinput=='E'

ferry\_id,m=find\_empty\_seat(data[schedule\_id], 'E')

IF m==-1

PRINT '''Ticket sold out for this hour!

Next trip leaves in 1hour!

Please select another schedule!''')

ENDIF

ELSE

print\_ticket(userinput, m,schedule\_id,ferry\_id)

data[schedule\_id][ferry\_id][1][m]=1

export\_file(data)

ELSE

PRINT ‘Sorry!Next trip leaves in 1 hour’

ENDIF

ELSE

print\_ticket(userinput, n,schedule\_id,ferry\_id)

data[schedule\_id][ferry\_id][0][n]=1

export\_file(data)

ENDIF

ENDIF

ELSE

IF userinput=='E'

ferry\_id,m=find\_empty\_seat(data[schedule\_id], 'E')

IF m == -1

PRINT ‘‘‘Economy Class is full!

Press B to select a Business seat!

Press M to return to the previous menu!’’’

userinput=READ()

IF userinput=='B'

ferry\_id,n=find\_empty\_seat(data[schedule\_id], 'B')

IF n==-1

PRINT'''Ticket sold out for this hour!

Next trip leaves in 1hour!

Please select another schedule!''')

ENDIF

ELSE

print\_ticket(userinput, n,schedule\_id,ferry\_id)

data[schedule\_id][ferry\_id][0][n]=1

export\_file(data)

ELSE

PRINT ‘Sorry! Next trip leaves in 1 hour’

ENDIF

ELSE

print\_ticket(userinput, m,schedule\_id,ferry\_id)

data[schedule\_id][ferry\_id][1][m]=1

export\_file(data)

ENDIF

ELSE

IF userinput=='M'

BREAK

ENDIF

ENDIF

ENDIF

ENDWHILE

ELSE

IF userinput=='V'

PRINT ‘Please select your Ferry ID’

schedule\_id=V()

IF schedule\_id=='M'

CONTINUE

ENDIF

ferry\_id = int(input('Enter ferry ID: ')) - 1

IF (ferry\_id >3)

ferry\_id-=4

ENDIF

print\_seats(data,schedule\_id,ferry\_id)

ENDIF

ELSE

IF userinput=='Q':

PRINT'Thank you for visiting!'

BREAK

ENDIF

ELSE

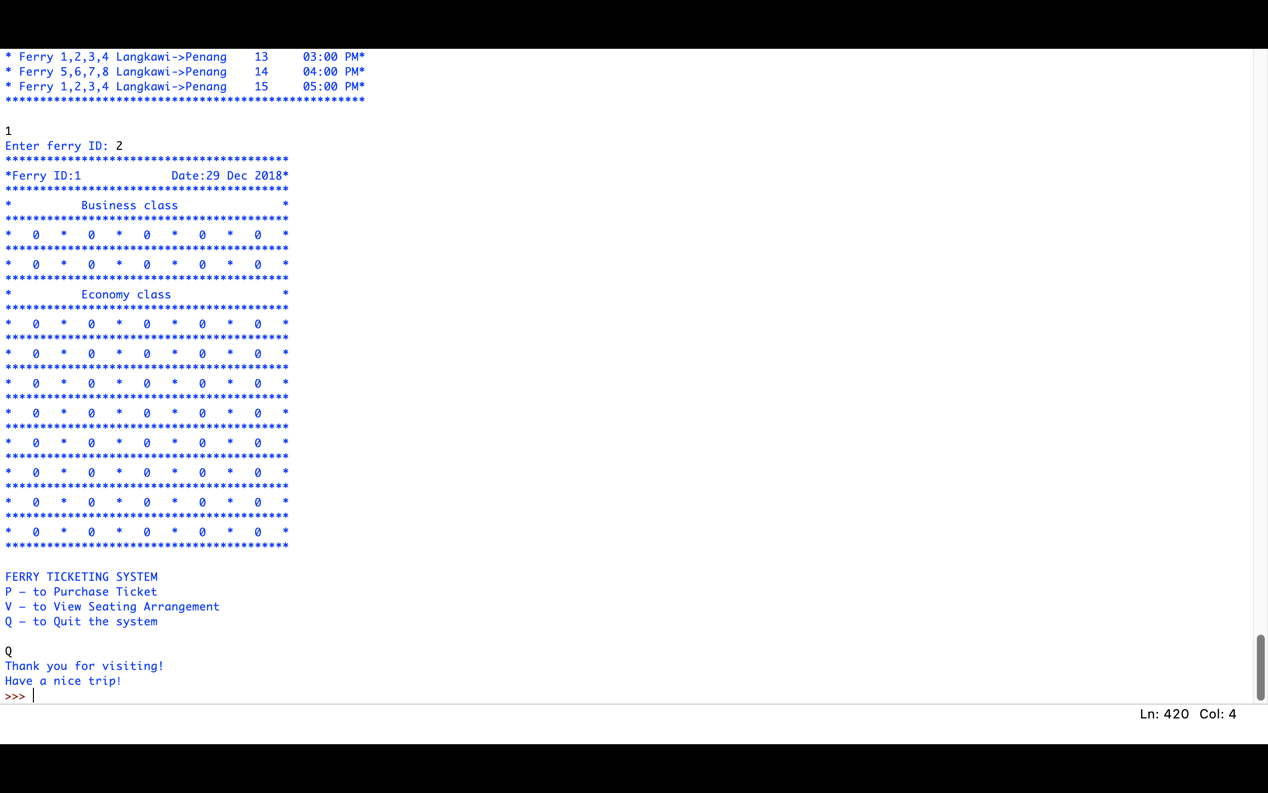
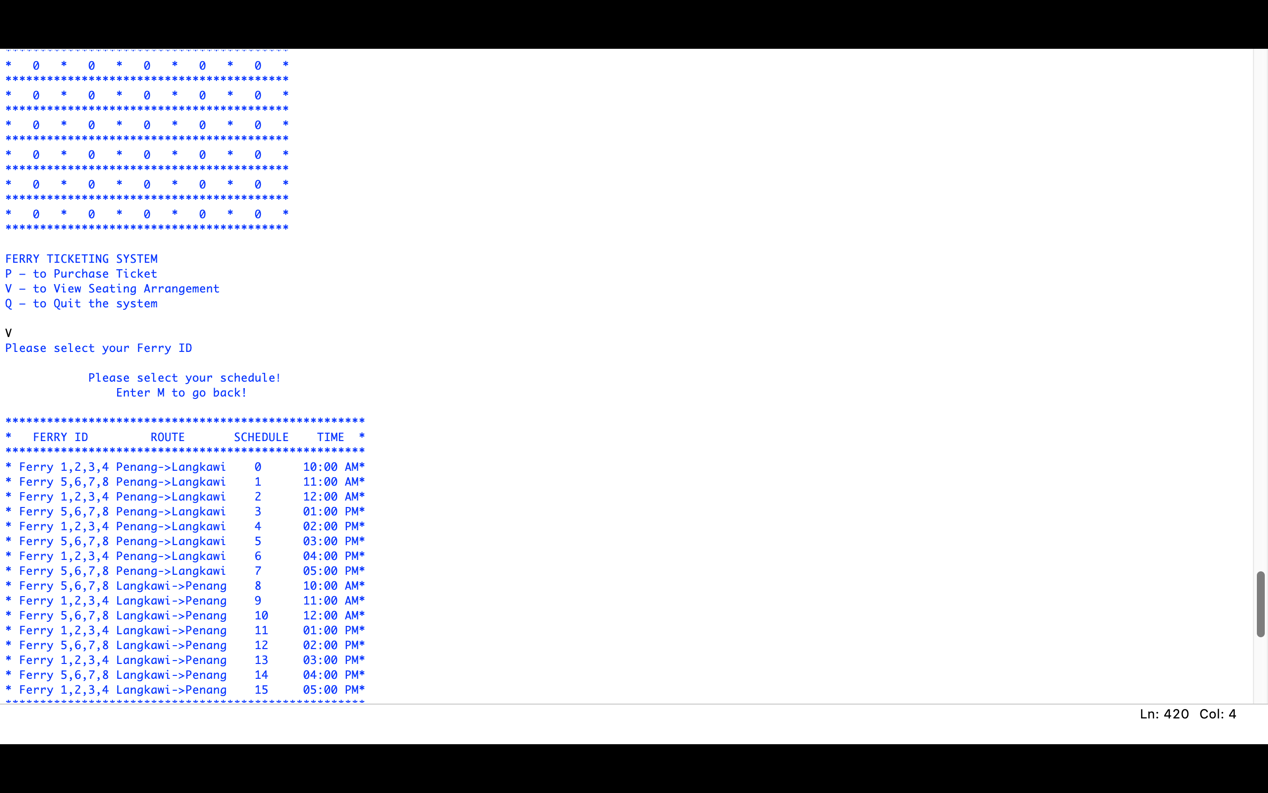
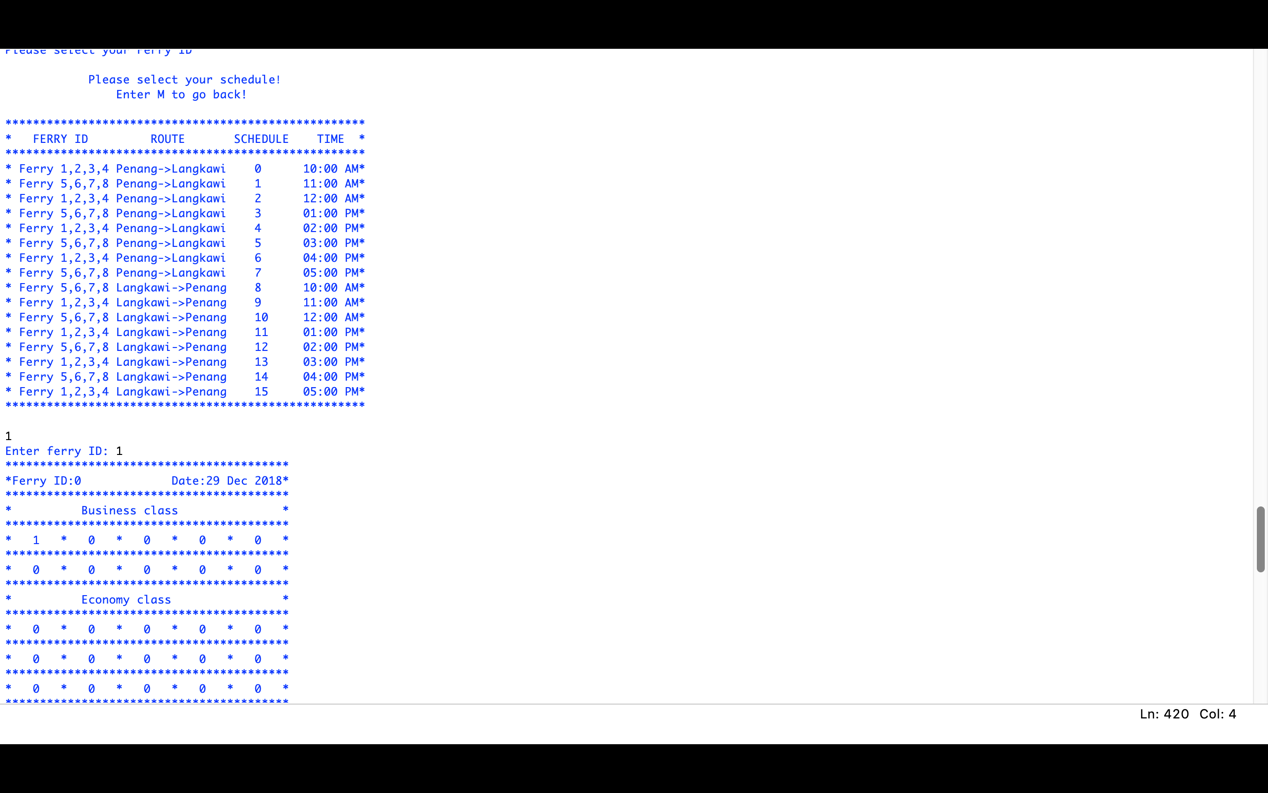
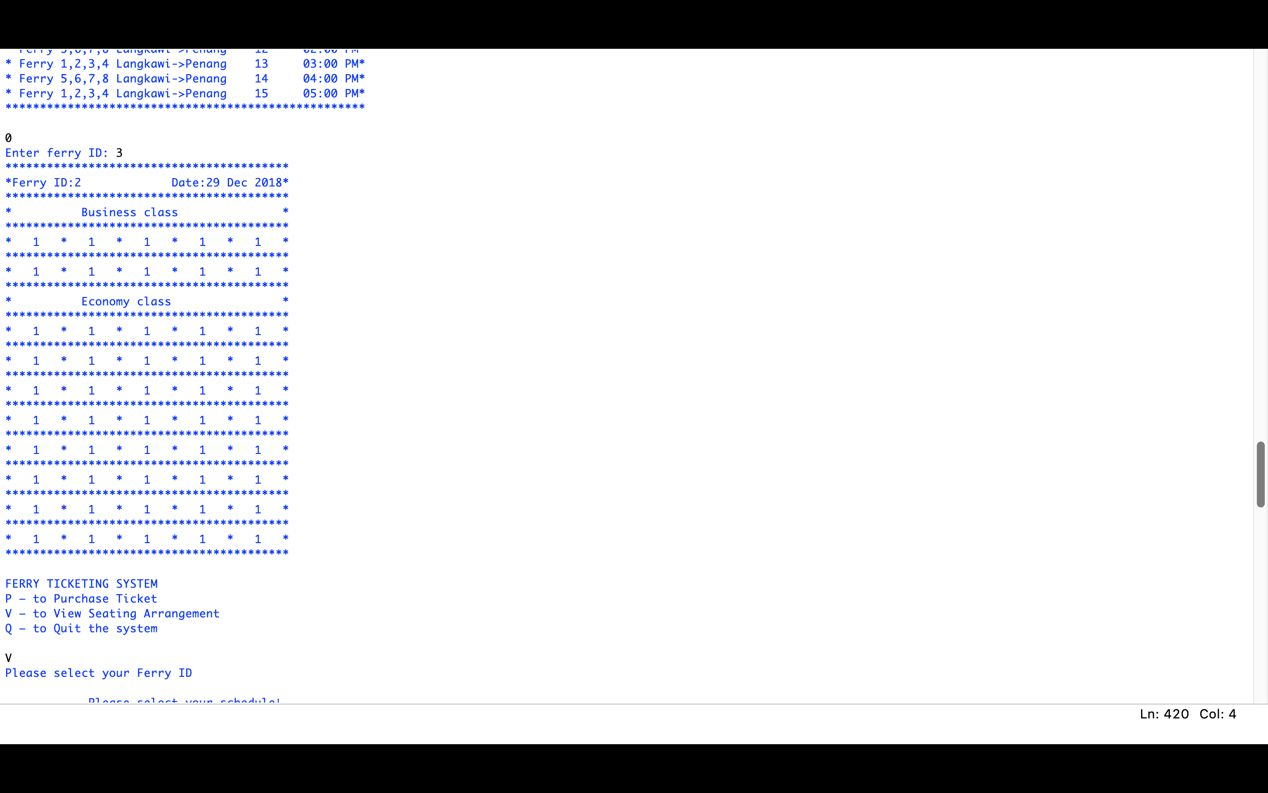
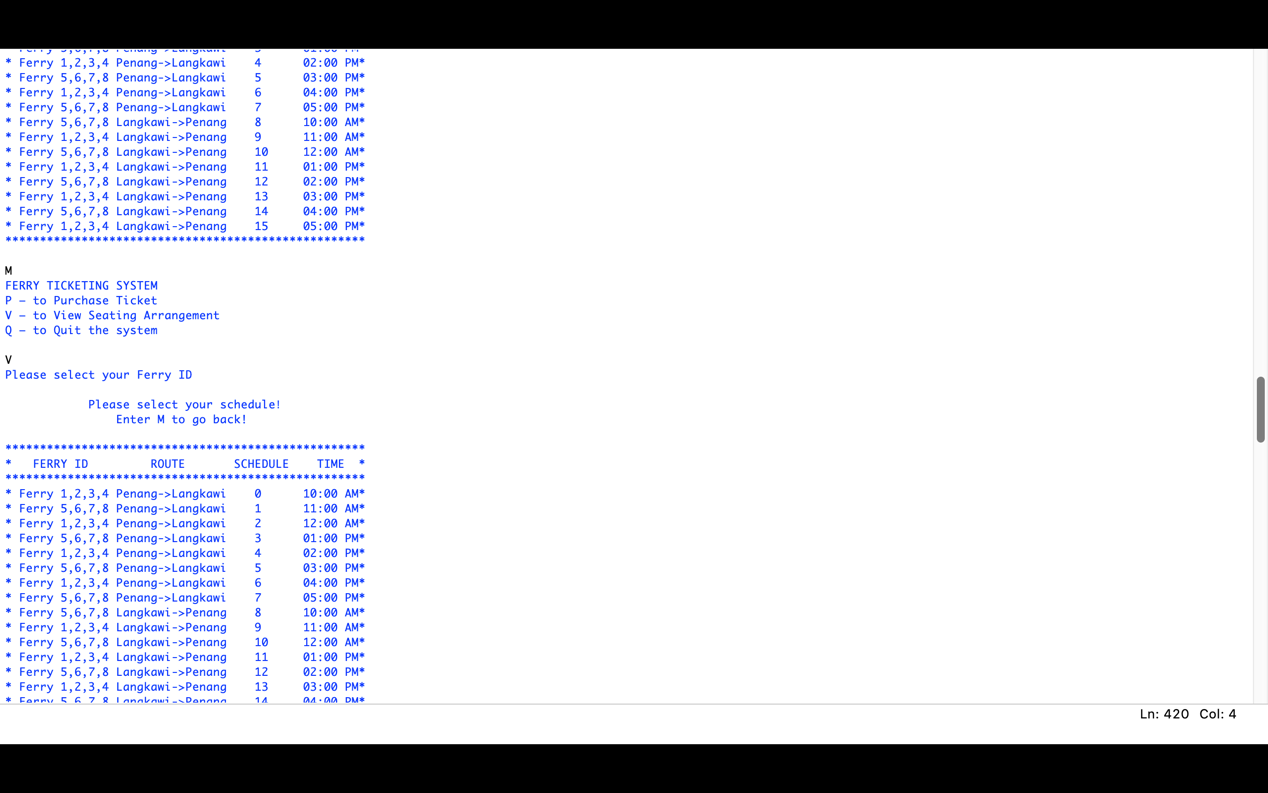
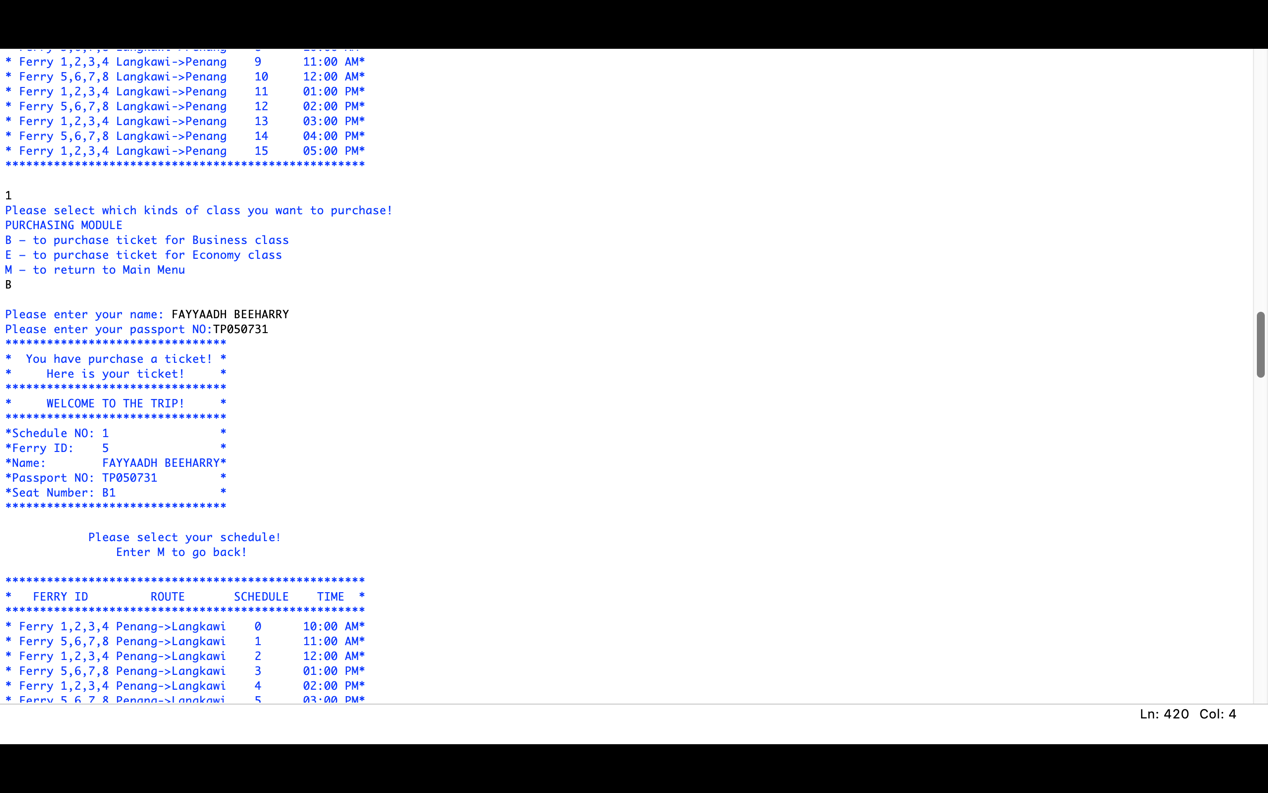
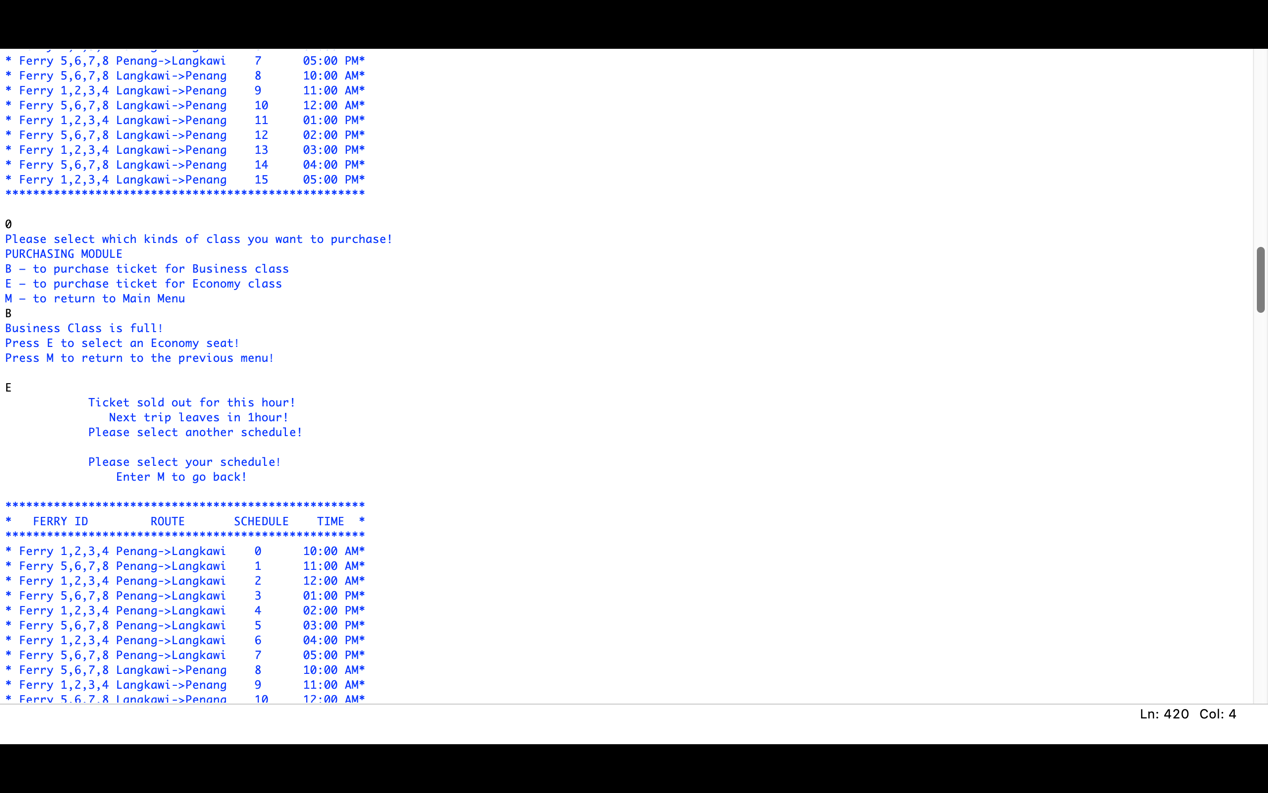
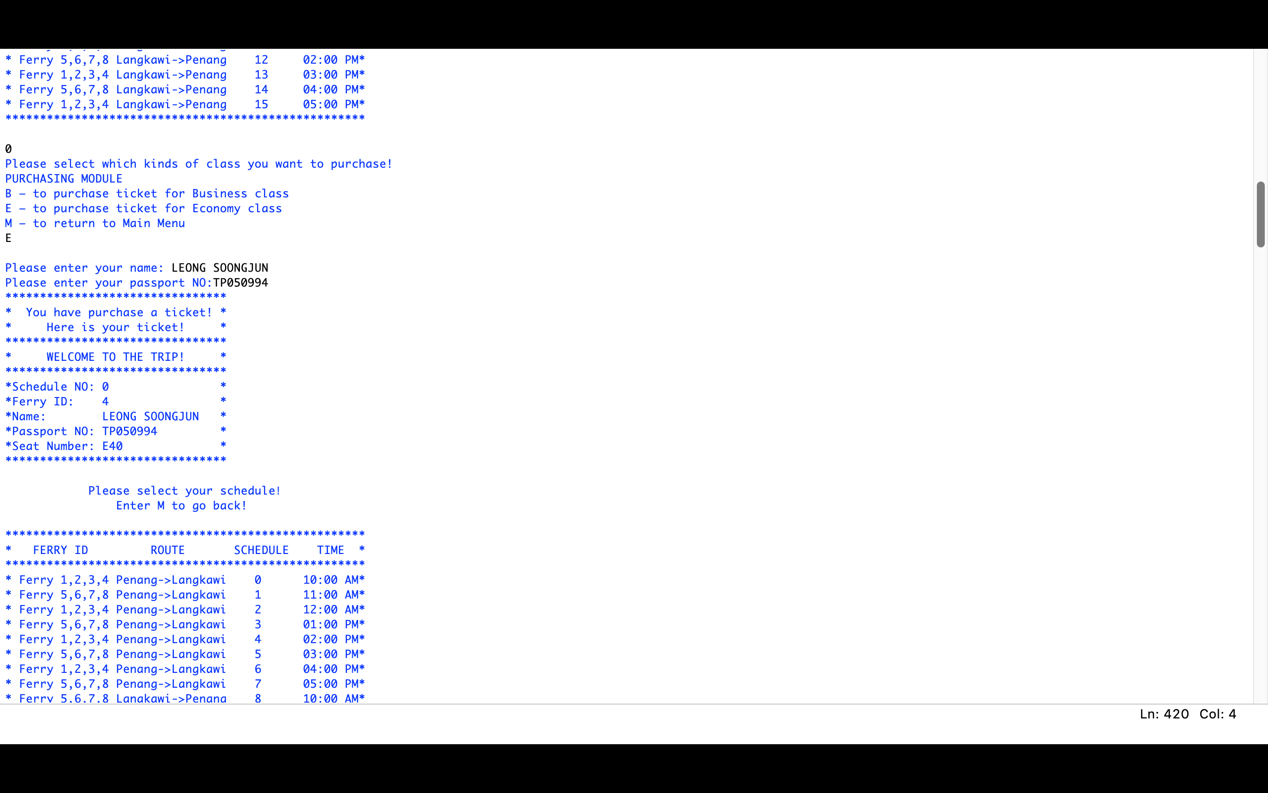
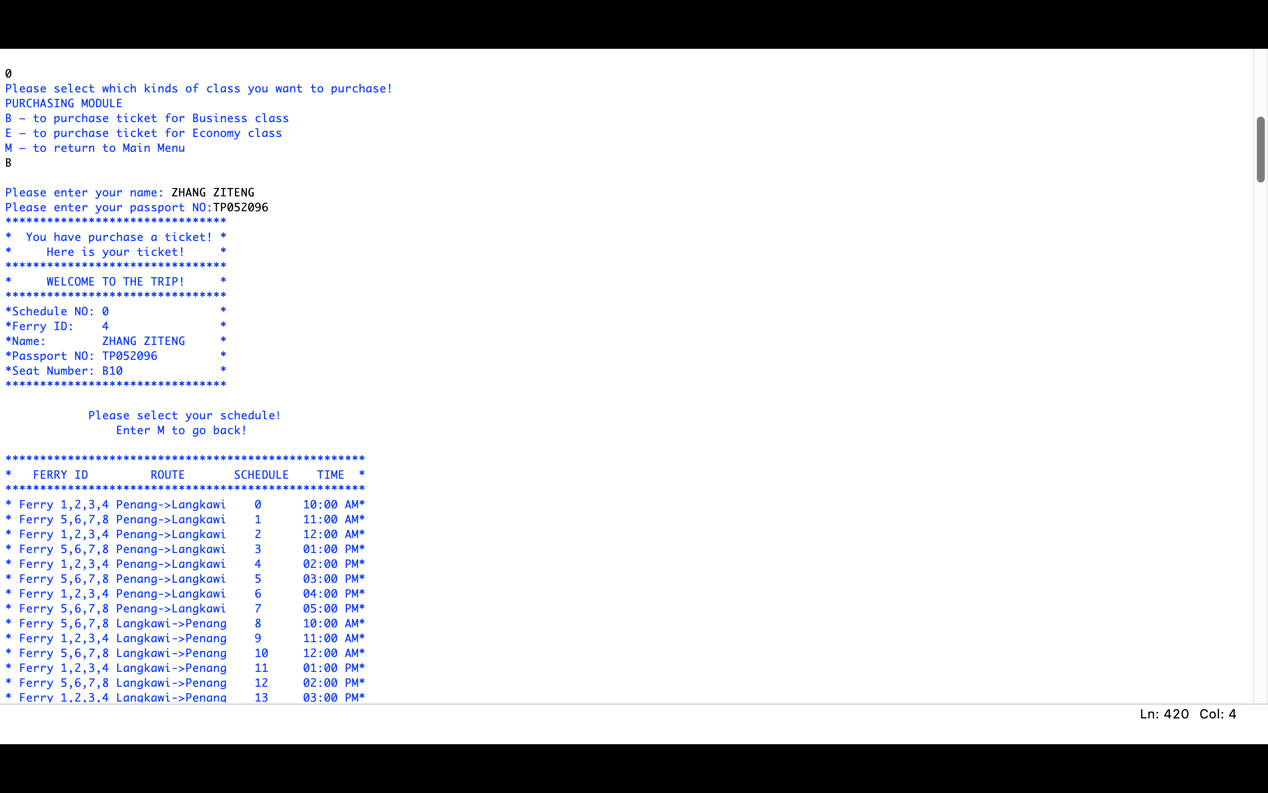
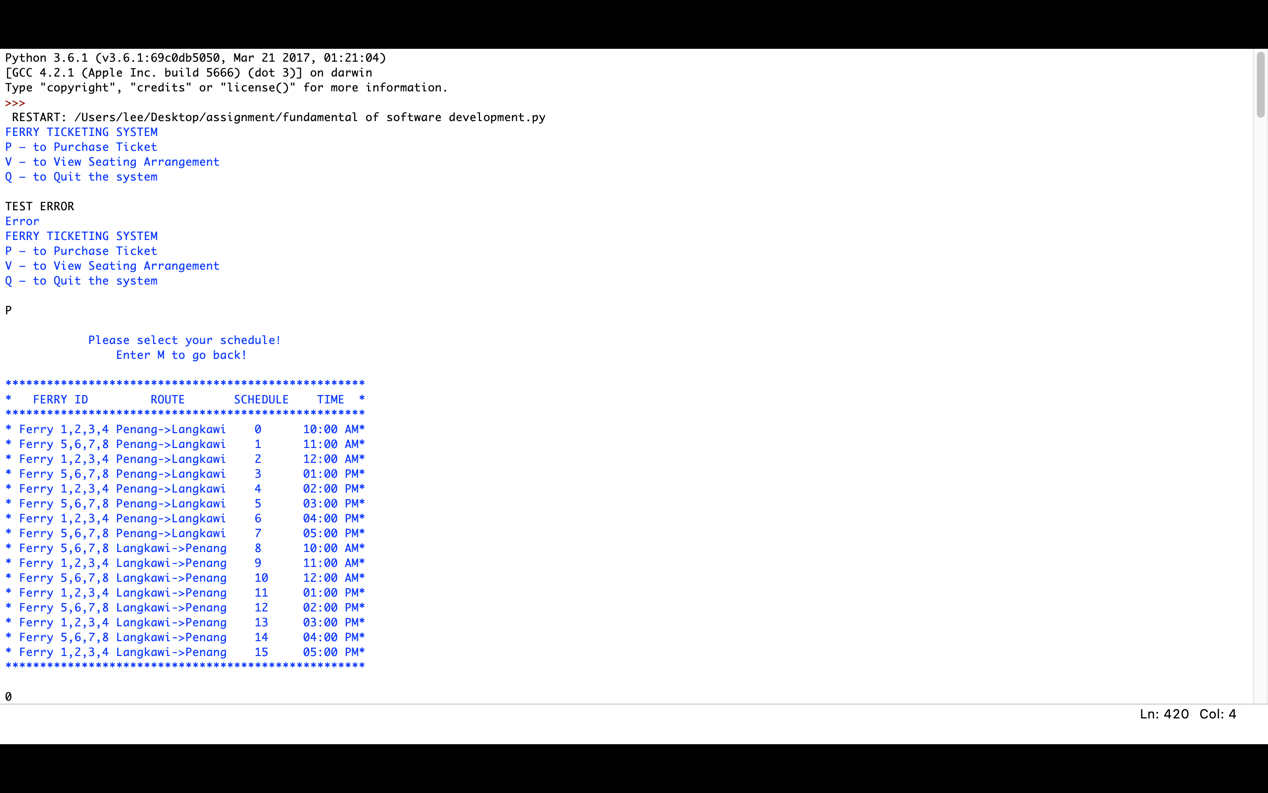
PRINT ‘Error’

ENDIF

PRINT 'Have a nice trip'

END

# INPUT AND OUTPUT



# ASSUMPTION

The system firstly will find and open the file called ''Hello world.json'' if it has, or it will create a new file with the full empty seats. The main menu of the system will have three options for the user, which are input 'P' to purchase the ticket, input 'V' to view the seating arrangement, input 'Q' to quit the system.

Type 'P' will go to the submenu to choose the kinds of seats that user wants to purchase, business or economy. the total number of seats for business and economy classes are 40 and 160 respectively. That is, each ferry has 10 business and 40 economic seats. User Name, Passport Number, Schedule Number will be asked if there is an empty seat available. After that, the system will print a ticket including username, passport number, schedule number, ferry ID, and the seat number that will be allocated. If the business class is full, it will prompt the user if he or she wants to change to the economy class. Seats will be arranged in economy class if the user agrees. In economy class, it will cover 160 seats and 40 for each ship. If the economy class is full, the system will ask if the user wants to change to the business class. Seats will be arranged in business class if the user agrees. The data will be saved after the ticket is printed. If user does not agree or both of the business and economy class are full, it will print the message ''Next trip leaves in 1 hour!''.

Type 'V' will go to the submenu to select the schedule number and ferry ID to view the seats that were sold or not directly. If user type 'M', it will go back to the main menu. Type 'Q' will quit the system and save data into the file ''Hello world.json''.

# CONCLUSION

We created our group during the tutorial class which is received the assignment from our lecturer. We decided to distribute our work that come out the result of 2 persons work together with the python coding meanwhile 1 person helps in pseudocode. However, we exchange our opinion during the meetings to improve our assignment. We try to write the program based on the knowledge that learned from the lecture and tutorial class, but in fact, we look for a more supportive website in the coding process. For instance, StackOverflow is one of the websites that we used to have a better understanding of python programming. We tried to discuss our idea with the lecturer. We had agreed to add an extra 8 ferries with 8 timelines respectively. Therefore, we have no doubt in writing the code on the part of main menu and submenu. Unfortunately, we faced obstacles while assigning the ferry tickets, seat arrangements, purchased date, library to store data and etc. We managed this problem by surf and analyze through the websites yet seek help from our beloved peeps and lecturer. Thus, pseudocode is written based on python programming. It is showed in simple English to allow readers to understand the program easily. In the endpoint of the assignment, we had our meetings to share the experiences and knowledge. We had decided to approve 40% of contributions to Ziteng and 30% to the rest of the members respectively. In the nutshell, we had completed our assignment.

REFERENCE

Detome, K. (2018). *5.7. (Part 1) Stadium Seating - Python*. [online] YouTube. Available at: https://www.youtube.com/watch?v=SYgQ5iFnBXc&feature=youtu.be [Accessed 6 Dec. 2018].

Detome, K. (2018). *5.7. (Part 2) Stadium Seating - Python*. [online] YouTube. Available at: https://www.youtube.com/watch?v=x15MrO8sqh0&feature=youtu.be [Accessed 11 Dec. 2018].

SEYYAD, M. (2018). *fundamentals of software development*. [online] APU Webspace. Available at: https://lms2.apiit.edu.my/pluginfile.php/83722/mod\_resource/content/1/MenuNavigation.py [Accessed 29 Dec. 2018].

Hellmann, D. (2018). *datetime — Date and Time Value Manipulation — PyMOTW 3*. [online] Pymotw.com. Available at: https://pymotw.com/3/datetime/ [Accessed 18 Dec. 2018].

Stack Overflow. (2018). *Stack Overflow - Where Developers Learn, Share, & Build Careers*. [online] Available at: https://stackoverflow.com/ [Accessed 13 Dec. 2018].