Jaypee Institute of Information Technology, Noida

COMPUTER ORGANIZATION AND ARCHITECTURE LAB

TOPIC : Railway Ticket Reservation System

Faculty :

- Dr. Pawan Kumar Upadhyay
- Dr. Kapil Madan

Group members:

- Riya Kansal (20103251, B9)
- Kritarth Bansal (20103256, B9)
- Saksham Gupta (20103268, B9)

INDEX :

INTRODUCTION :

The project Railway Ticket Reservation System has been developed on an 8086 16-bit microprocessor ,which was designed by **INTEL** in 1979.It runs on emulator **emu8086**. The reservation system provides an ease to its customers by letting them choose their train and ticket according to their choices.

The idea behind Developing Railway Ticket Reservation System was to allow people to know about different trains schedule ,

Book their tickets and then let them enquire about the status of their booked ticket(s) at the comfort of their home.

Railway Ticket Reservation System not only displays the train details but it also display the features as mentioned below:

OBJECTIVES:

1. To diminish complexity of existing systems.
2. Successful administration of time.
3. To make work simple, straightforward and blunder free.
4. Compelling use of accessible assets.
5. To improve the productivity and enhancement of administration exercises.
6. Easy to use.

Functionality provided by Railway Reservation System are as follows:

1. Provides the searching facilities based on various factors such as Train, booking, Confirmation, allotting seats.
2. It tracks all the information of Ticket, seat, customer etc.
3. Manage the info of the ticket.
4. To increase efficiency of managing the train booking.

CODE :

```
data segment
;strings
newstring db 0ah,0dh,' $'
welcomeStr db 0ah,0dh,'Welcome to Railway Ticket
Reservation $ '
thankyouStr db 0ah,0dh,'Thank you for using our system $'


;train menu
trainMenuStr db 0ah,0dh,'Please Select a Train to Book
Tickets: $'
trainSel db 0ah,0dh,'Enter Train number: $'
trainA db 0ah,0dh,'1: Train A $'
trainB db 0ah,0dh,'2: Train B $'
trainC db 0ah,0dh,'3: Train C $'


;class menu
classMenuStr db 0ah,0dh,'Please select a class $'
classSel db 0ah,0dh,'Select a class: $'
classA db 0ah,0dh,'1: CLASS AC $'
classB db 0ah,0dh,'2: CLASS SLEEPER $'
classC db 0ah,0dh,'3: CLASS CHAIR $'


;error message
errorStr db 0ah,0dh,'INVALID OPTION CHOOSEN.PLEASE
TRY AGAIN!!!$'


;status messages
currentStat db 0ah,0dh,'Current Status: $'
noMoreAvail db 0ah,0dh,'Sorry!! No more seats are
available $'
returnClassMenu db 0ah,0dh,'1: Return to class selection $'
returnMainMenu db 0ah,0dh,'2: Return to main menu $'
exitProg db 0ah,0dh,'3: Exit $'
choiceRead db 0ah,0dh,'Enter choice: $'
availSeats db 0ah,0dh,'No. of seats available: $'
numofSeatsToBook db 0ah,0dh,'Enter no .of seats to be
booked,[max 5]:$'
overflow db 0ah,0dh,'Maximum limit exceeded,give less than
5: $'
successful db 0ah,0dh,'Booking successfulful! $'
bookedSeat db 0ah,0dh,'Your seats are: $'
anotherTick db 0ah,0dh,'1: Book another ticket $'
exitProgam db 0ah,0dh,'2: Exit $'
```

```
;variables to store available seating information
trainASeatsNumber db 45 dup(15)
trainBSeatsNumber db 45 dup(15)
trainCSeatsNumber db 45 dup(15)


;variables to display booked/unbooked seats
trainASeats db 45 dup(0)
trainBSeats db 45 dup(0)
trainCSeats db 45 dup(0)


;currently chosen items
currentlyChosenTrain db ?
currentlyChosenClass db ?
choiceCompartment dw ?
choiceavailSeats dw ?
chosenTrainID db ?
chosenClassID db ?
printVal db ?
data ends


;macro to print strings
printString macro arg
lea dx,arg
mov ah,09h
int 21h
endm


;macro to print char
printChar macro arg
mov dl,arg
mov ah,02h
int 21h
endm


;code segment
code segment
assume cs:code,ds:data
start: mov ax,data
mov ds,ax


;to clear screen
menu: mov ah,00h
mov al,02h
int 10h


;clear screen
printString welcomeStr
printString newstring


;train menu
printString trainMenuStr
printString newstring
printString trainA
printString newstring
printString trainB
printString newstring
```

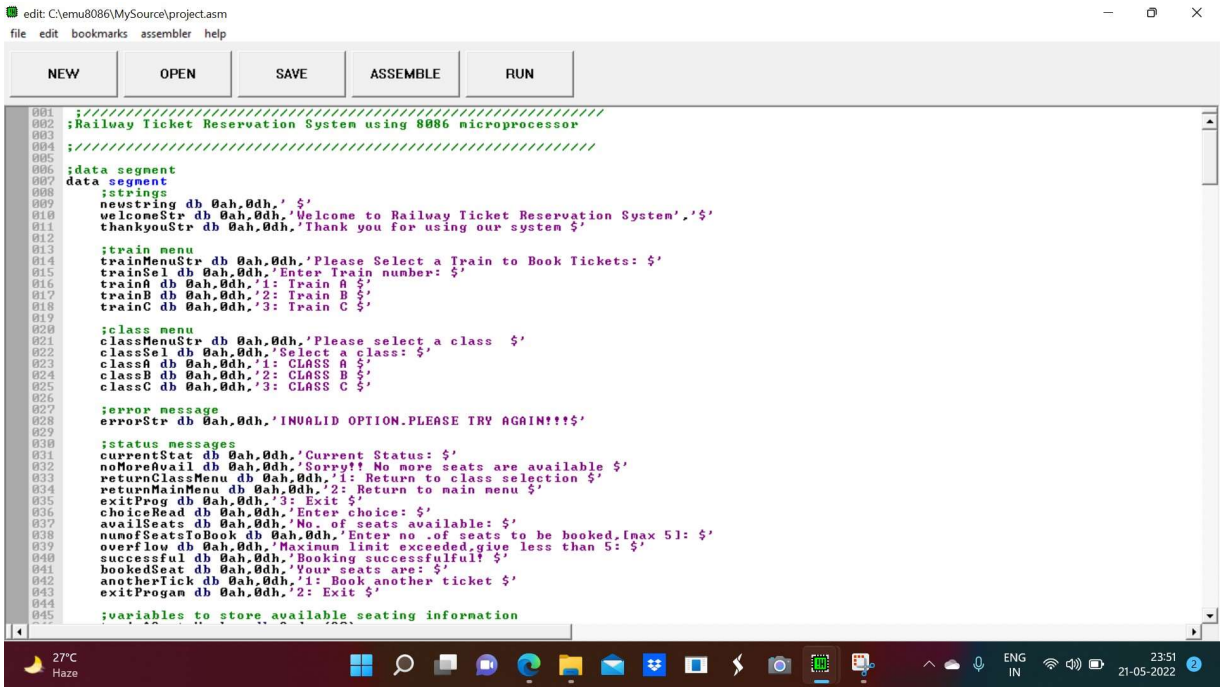
```
printString trainC
chooseTrain: printString newstring
printString trainSel
call readInt


;check for errors
cmp al,04h
jc noErrorTrain
printString newstring
printString errorStr
jmp chooseTrain


;choose class
noErrorTrain: mov currentlyChosenTrain,al
printString newstring
classMenu: printString classMenuStr
printString newstring
printString classA
printString newstring
printString classB
printString newstring
printString classC
chooseClass: printString newstring
printString classSel
call readInt


;check for errors
cmp al,04h
jc noErrorClass
printString newstring
printString errorStr
jmp chooseClass
noErrorClass: mov currentlyChosenClass,al


;set appropriate pointers
call displaySeats
mov si,choiceCompartment
mov di,choiceavailSeats
mov bh,[di]
mov ah,0fh
sub ah,bh
addLoop: cmp ah,00h
jz h
inc si
dec ah
jmp addLoop


;check if seats are available
h: cmp bh,00h
jnz seatAvailable
printString noMoreAvail
printString newstring
printString returnClassMenu
printString returnMainMenu
printString exitProg
redoChoice: printString newstring
printString choiceRead
call readInt
cmp al,04h
```

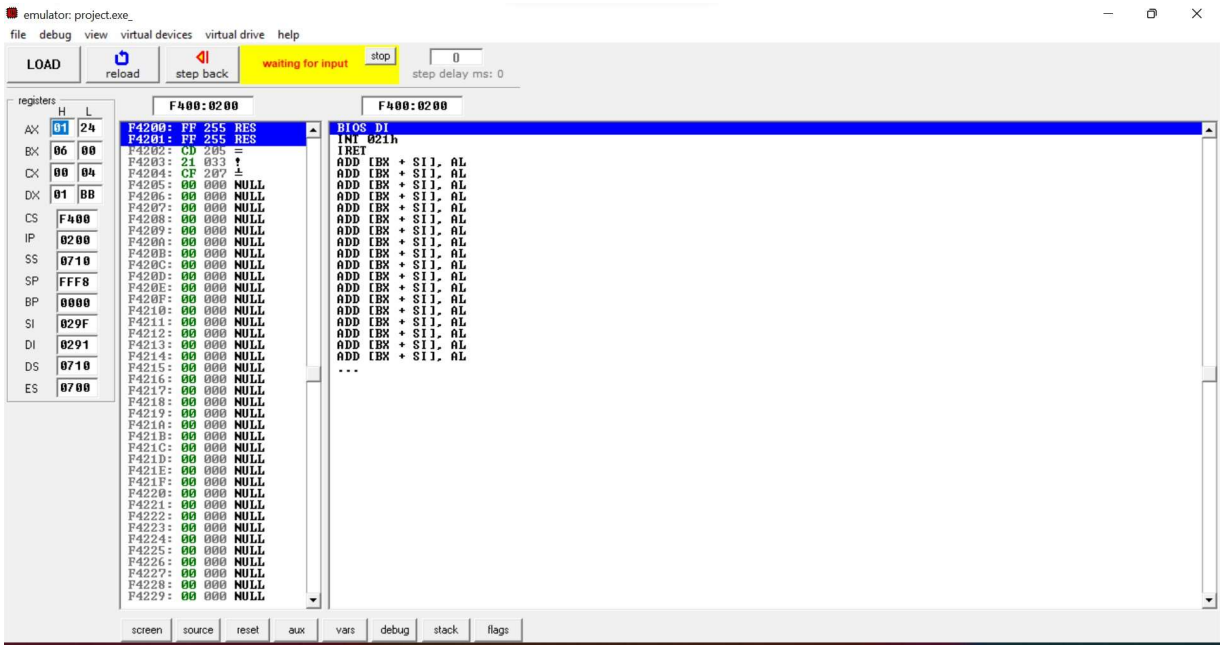
```
jc process
printString errorStr
jmp redoChoice
process: cmp al,01h
jz classMenu
cmp al,02h
jz menu
jmp exit


seatAvailable: printString availSeats
mov printVal,bh
call printInt
repeatBook: printString newstring
printString numofSeatsToBook
call readInt
cmp al,06h
jc proceedBooking
printString overflow
jmp repeatBook
proceedBooking: mov bl,al
mov bh,0fh
mov ah,[di]
sub bh,ah
inc bh
printString newstring
printString successful
printString bookedSeat


printSeats: printChar chosenTrainID
printChar chosenClassID
mov printVal,bh
call printInt
inc bh
printChar ' '
mov [si],01h
inc si
dec [di]
dec bl
jnz printSeats
;what to do after booking
whatNext: printString newstring
printString anotherTick
printString exitProg
printString newstring
printString choiceRead
call readInt


;check for errors
cmp al,04h
jc noProbs
printString newstring
printString errorStr
jmp whatNext
noProbs: cmp al,01h
jz menu
exit: printString newstring
printString thankyouStr
mov ah,4ch
int 21h
```

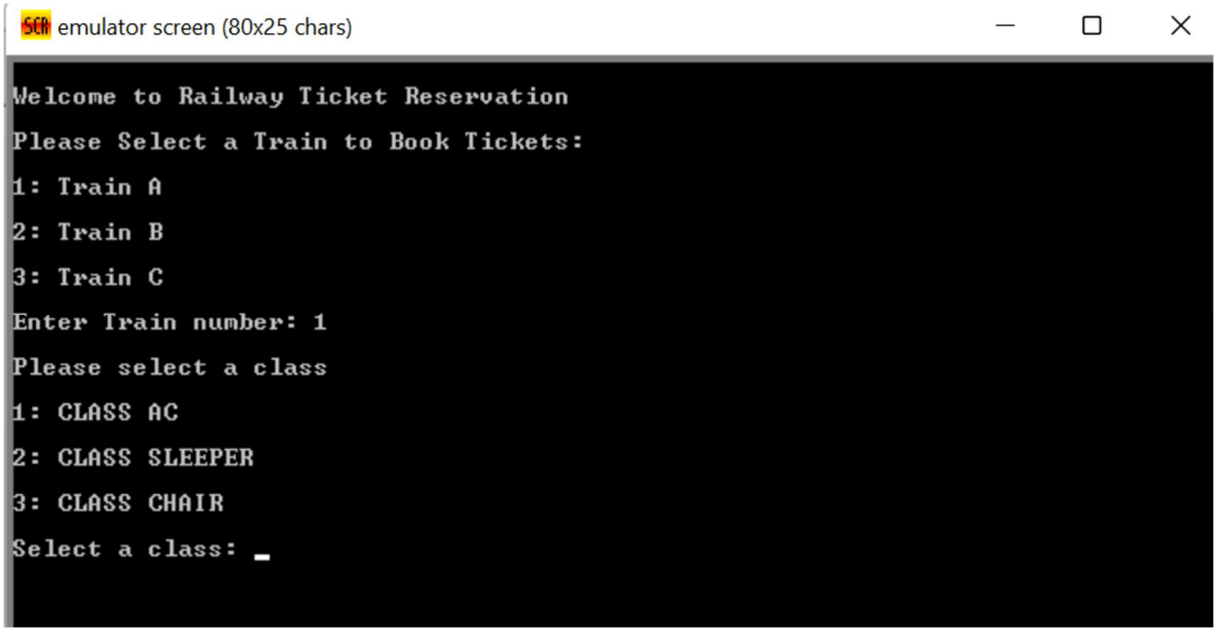
```
;procedure to read int
readInt proc
mov ah,01h
int 21h
sub al,30h
cmp al,09h
jc rn
jz rn
sub al,07h
rn:ret
endp


;procedure to print integer
printInt proc
mov dl,printVal
and dl,0F0h
mov cl,04h
shr dl,cl
add dl,30h
cmp dl,39h
jc rnu
jz rnu
add dl,07h
rnu: mov ah,02h
int 21h
mov dl,printVal
and dl,0Fh
add dl,30h
cmp dl,39h
jc rnp
jz rnp
add dl,07h
rnp: mov ah,02h
int 21h
ret
endp
;procedure to display seating
displaySeats proc
;load the appropriate location into si
trainAChosen: cmp currentlyChosenTrain,01h
jnz trainBChosen
lea si,trainASeats
lea di,trainASeatsNumber
mov chosenTrainID,'A'
jmp classAChosen
trainBChosen: cmp currentlyChosenTrain,02h
jnz trainCChosen
lea si,trainBSeats
lea di,trainBSeatsNumber
mov chosenTrainID,'B'
jmp classAChosen
trainCChosen: lea si,trainCSeats
lea di,trainCSeatsNumber
mov chosenTrainID,'C'
classAChosen: cmp currentlyChosenClass,01h
jnz classBChosen
mov chosenClassID,'A'
jmp chooseOver
classBChosen: cmp currentlyChosenClass,02h
jnz classCChosen
add si,0fh
add di,01h
mov chosenClassID,'B'
```

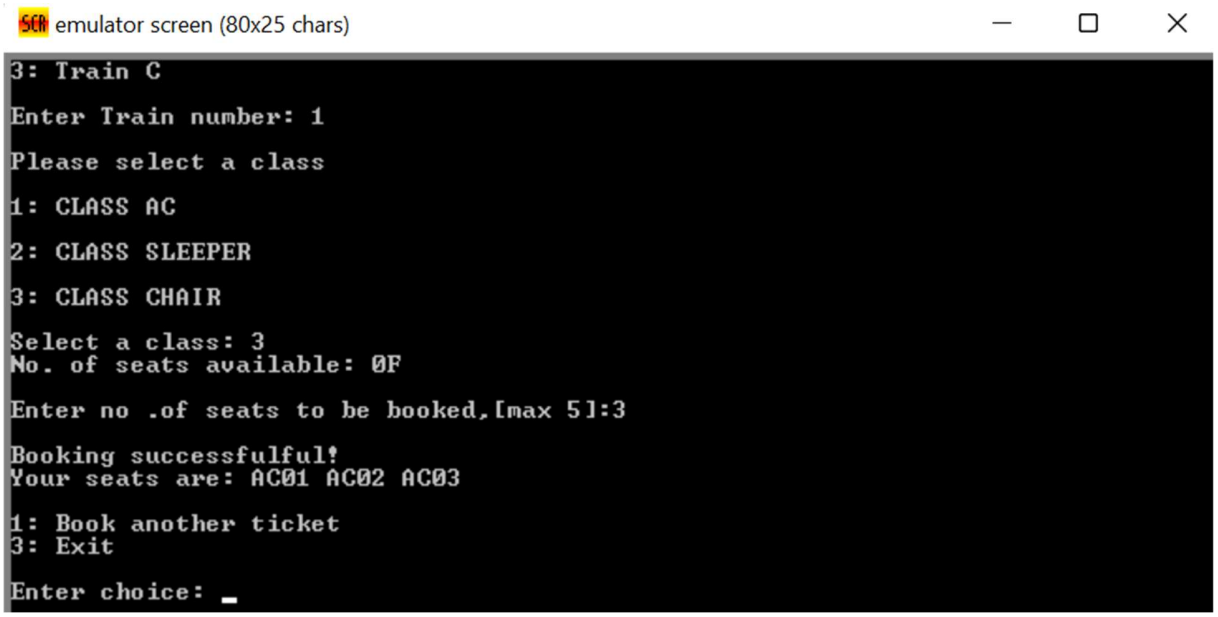
```
jmp chooseOver
classCChosen: add si,28h
add di,02h
mov chosenClassID,'C'


chooseOver: mov choiceavailSeats,di
mov di,si
mov choiceCompartment,si
inc di
ret
endp
code ends
end start
```

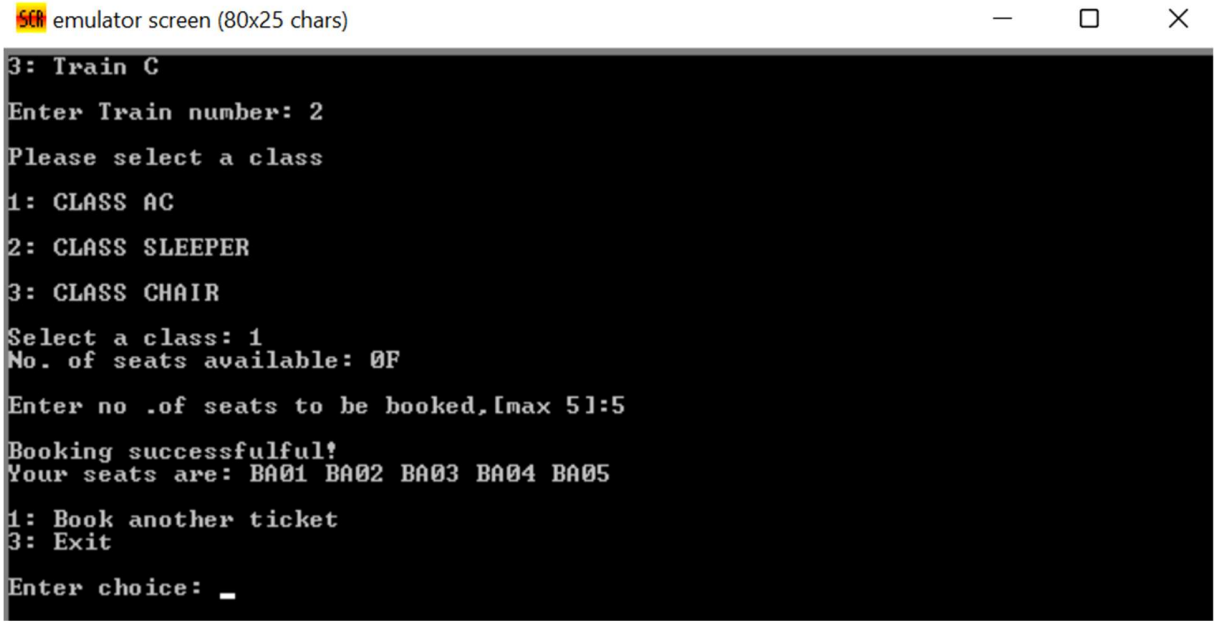
Screenshots:

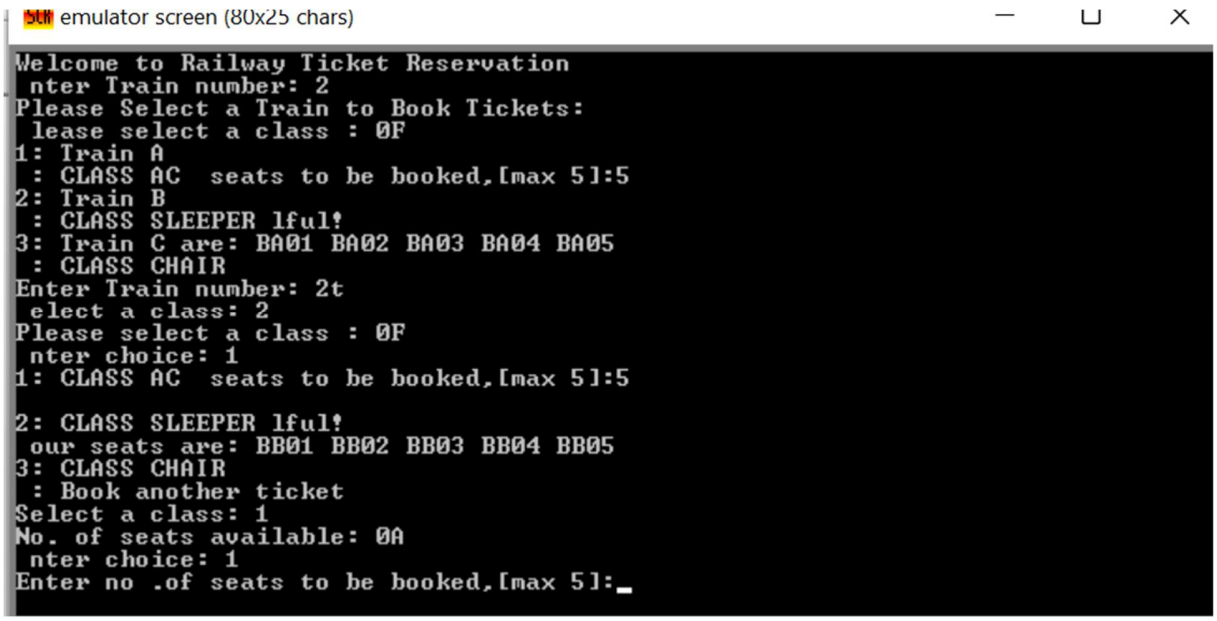



emulator: project.exe_
file debug view virtual devices virtual drive help
LOAD
reload
step back
waiting for input
stop
step delay ms: 0
registers
AX 01 24
BX 06 00
CX 00 04
DX 01 BB
CS F400
IP 0200
SS 0710
SP FFF8
BP 0000
SI 029F
DI 0291
DS 0710
ES 0700
F400:0200
F400:0200
BIOS DI
INT 021h
IRET
screen
source
reset
aux
vars
debug
stack
flags

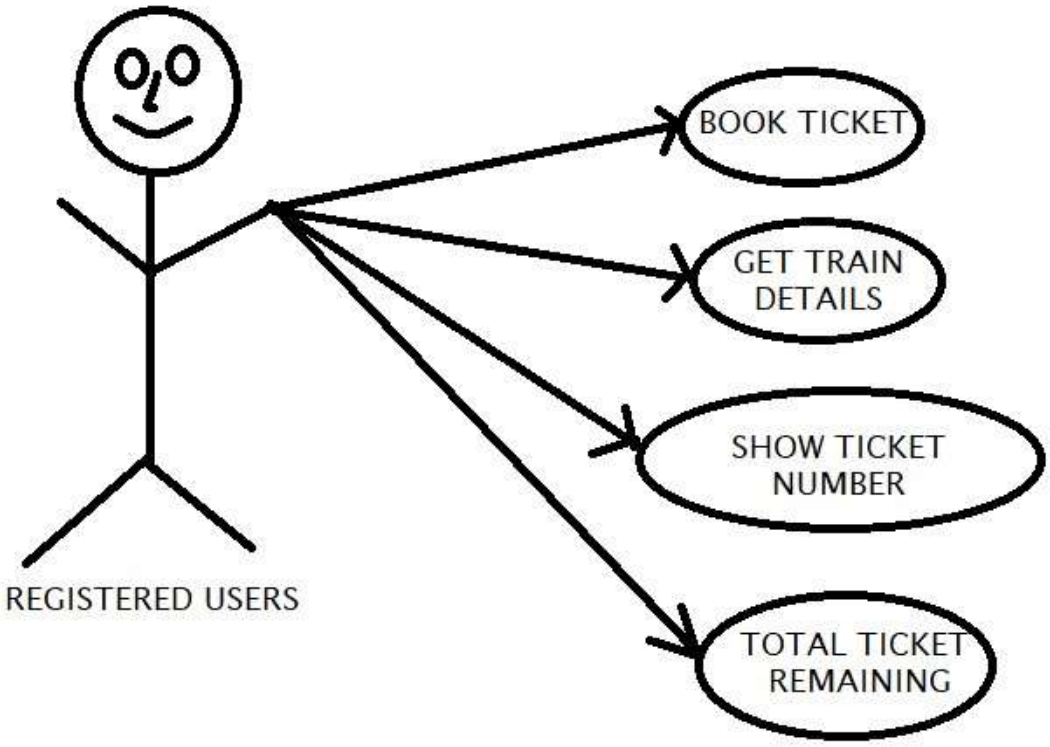

emulator screen (80x25 chars)
Welcome to Railway Ticket Reservation
Please Select a Train to Book Tickets:
1: Train A
2: Train B
3: Train C
Enter Train number: 1
Please select a class
1: CLASS AC
2: CLASS SLEEPER
3: CLASS CHAIR
Select a class:


emulator screen (80x25 chars)
3: Train C
Enter Train number: 1
Please select a class
1: CLASS AC
2: CLASS SLEEPER
3: CLASS CHAIR
Select a class: 3
No. of seats available: 0F
Enter no .of seats to be booked,[max 5]:3
Booking successfulful!
Your seats are: AC01 AC02 AC03
1: Book another ticket
3: Exit
Enter choice:


emulator screen (80x25 chars)
3: Train C
Enter Train number: 2
Please select a class
1: CLASS AC
2: CLASS SLEEPER
3: CLASS CHAIR
Select a class: 1
No. of seats available: 0F
Enter no .of seats to be booked,[max 5]:5
Booking successfulful!
Your seats are: BA01 BA02 BA03 BA04 BA05
1: Book another ticket
3: Exit
Enter choice:


8086 emulator screen (80x25 chars)
Welcome to Railway Ticket Reservation
nter Train number: 2
Please Select a Train to Book Tickets:
lease select a class : 0F
1: Train A
: CLASS AC seats to be booked,[max 5]:5
2: Train B
: CLASS SLEEPER lful!
3: Train C are: BA01 BA02 BA03 BA04 BA05
: CLASS CHAIR
Enter Train number: 2t
elect a class: 2
Please select a class : 0F
nter choice: 1
1: CLASS AC seats to be booked,[max 5]:5
2: CLASS SLEEPER lful!
our seats are: BB01 BB02 BB03 BB04 BB05
3: CLASS CHAIR
: Book another ticket
Select a class: 1
No. of seats available: 0A
nter choice: 1
Enter no .of seats to be booked,[max 5]:

WORKFLOW - DIAGRAM:



Conclusion:

Through this project we learnt that there are several applications of emulator emu8086.
Its applications include string print ,char print ,char read and various other features which lead to new learnings and we basically got introduced to new concept.We learnt how to use emu8086 and importance of registers in 8086.
Thereby ,with the completion of the project under the guidance of our professor Dr.Kapil Madan and Dr.Pawan Kumar Upadhyay we were successfully able to complete our project .

References:

- Computer Organization and Architecture Tutorial | COA Tutorial javatpoint
- https://www.geeksforgeeks.org/
- Book :Barry b bre