Case Study Submission Part-1

**Title: Railway Zone Management**

|  |  |  |
| --- | --- | --- |
| **ROLL Number** | **Name** | **Contribution in Case Study** |
| CB.EN.U4CSE19153 | T. Krishna Chaitanya | Tourism And Catering Department |
| CB.EN.U4CSE19155 | U. Yeshwanth | Railway Hospital Department |
| CB.EN.U4CSE19156 | V. S. S. K. Chaitanya | Railway Reservation Department |
| CB.EN.U4CSE19165 | P. Prem Sai | Accounts Department |

Name: V.S.S.K.Chaitanya

Roll Number : CB.EN.U4CSE19156

**Description of the case study:**

Railway zones play a very crucial part in maintenance of Indian Railway system

various data that need to be managed between the stations coming under a particular zone such as maintaining the record of passenger data, health system, amount sanctioned to the station for development etc. All this data must be collected from various stations falling under a particular zone must be managed by the zone headquarters

So In order to operate effectively exchange of data among railway stations located in various place is crucial and needs a proper architecture. Our case study is aimed to demonstrate how different departments within a railway zone headquarter mange this effectively.

**Railway Zone Network**

**NETWORKS**

Tourism And Catering Department – Wide Area Network

Railway Reservation – Wide Area Network

Railway Hospital – Local Area Network

|  |  |
| --- | --- |
| **Client configuration** | **Server Configuration** |
| Intel(R) Core 4005U Series Processor | 2 Intel(R) Itanium 2 9100 Series Processors |
| 8 GB RAM | 23.98 GB RAM |
| 8 MB Cache | 12 MB Cache |
| 1 TB | 4.5 TB Storage (EMC Box) |
| Dual External Storage Connectivity | Dual External Storage Connectivity |
| DVD ROM Drive | DVD ROM Drive |

Accounts Department – Virtual Private Network

**SERVERS**

Database Server

Proxy Server

Mail Server

Application Server

Client Server

**CABLES**

Coaxial Fibre

Twisted Fibre

Optical Fibre

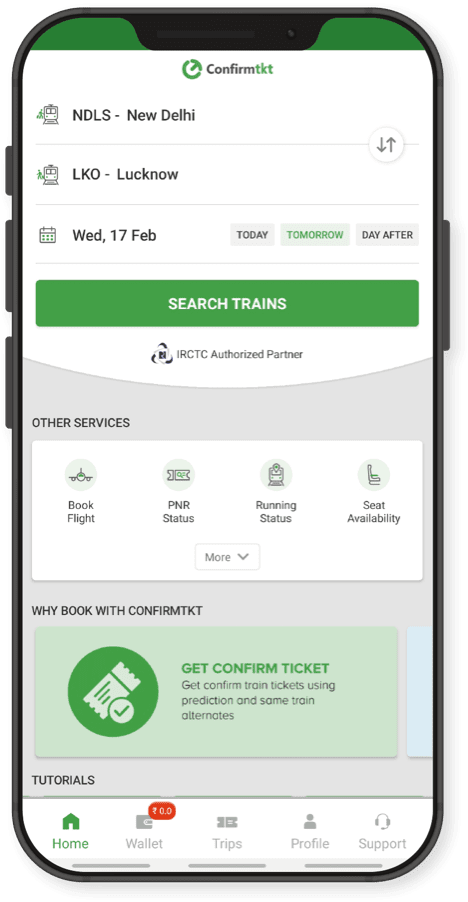
**TOPOLOGY**

Reservation, Accounts - Star Topology

Medical - Bus Topology

Tourism And Catering Department -



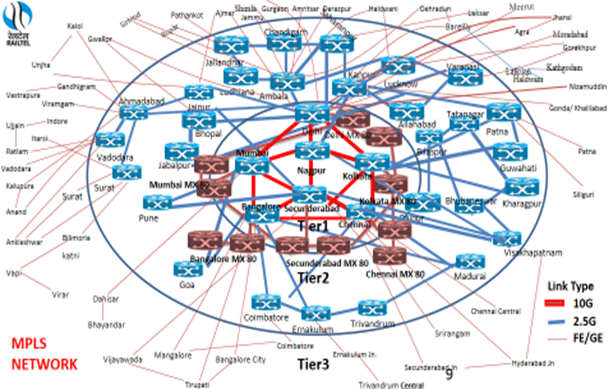
****

****

****

**Why is Networking required?**

As we saw before railway zone headquarters needs to collect and send info to various railway stations and should be able to manage the passenger data. So computer network plays a vital role to establish a proper connection between various departments. A proper architecture is needed because various devices like printer, ip phone, pcs are to be connected. So that a zone headquarter can manage all the railway stations effectively.



List of Network performance parameters:

**Bandwidth:**

**bandwidth** is defined as the maximum number of bits that can flow through a network connection in a given period of time. The fundamental unit of network bandwidth is bits per second.

Formula:

Expressed as bits per second ( bps ), modern network links have greater capacity, which is typically measured in millions of bits per second

( megabits per second , or Mbps) or billions of bits per second ( gigabits per second ,or Gbps).

**Possible Value In General Scenario.**

normally speeds ranging from 10–1,000 Mbps

**Throughput:**

The purpose of throughput is measuring the number of messages successfully transmitted per unit time.

**Formula:**

F = Frames per minute

A = Average of bits each frame carries

Through Put = ((F x A)/Unit Time)

**Possible Value In General Scenario:**

16.2 mbps

**Latency (Delay):**

Latency meaning in networking is best thought of as the amount of time it takes for a packet of data to be captured, transmitted, processed through multiple devices, then received at its destination and decoded.

**Formula:**

Propagation time = Distance/Propagation speed

Transmission time=Message size / Bandwidth

Queuing Time = Directly Proportional to the congestion in the network

Processing Delay = Directly proportional to processing speed of the

routers.

Latency = Propagation Time + Transmission Time + Queuing Time + Processing Delay

**Possible Value In General Scenario:**

1ms to 20ms

**Bandwidth-Delay Product:**

Bandwidth and delay are two performance measurements of a link. However, what is significant in data communications is the product of the two, the bandwidth-delay product.

**Formula:**

Bandwidth x Delay

**Possible Value In General Scenario:**

500kb

Example is Residential Cable internet

**Jitter:**

JItter is used to describe the amount of inconsistency in latency across the network, while latency measures the time it takes for data to reach its destination and ultimately make a round trip

**Formula:**

To measure Jitter, we take the difference between samples,then divide by the number of samples (minus-1)

**Possible Value In General Scenario:**

below 30ms

Railway reservation department manages the passenger reservation count of trains based on different classes like sleeper class, 3A class, 2A class.

Description of CSV file:

It consists of five columns

Train\_id: Train Id is an unique id which is given to each train to avoid confusion with trains having same names.

Train name: Name of the Train.

Sleeper/3a/2a : Passenger count of respective classes on a particular day.

1. List of operations completed with the File

1.Adding new train details.

2.updating existing train details.

3.Retrieving the particular train details.

4.Viewing all train details on a particular Day.

**CODE:**

**SERVER.PY:**

**CODE:**

**SERVER.P**

import socket

import threading

import pandas as pd

df=pd.read\_csv('train\_list.csv')

df=df.set\_index('train\_id')

print(df)

s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

port=6000

host=socket.gethostname()

s.bind((host,port))

s.listen()

print('Server is listening ...Passenger\_Reservation\_\_Department')

def add\_info(p,df):

df2=pd.DataFrame.from\_records([{'train':p[0],'sleeper':int(p[1]),'3a':int(p[2]),'2a':int(p[3])}],index='train\_id')

print(df2)

df=df.append(df2)

print(df)

df.to\_csv('train\_list.csv')

def handle\_client(con,df,adr):

c=True

while c:

f=f[1:]

f=f[:len(f)-1]

f=f.split('\_')

print(f)

if f[0]=="1":

p=f[1:]

df2=pd.DataFrame.from\_records([{'train\_id':p[0],'train':p[1],'sleeper':int(p[2]),'3a':int(p[3]),'2a':int(p[4])}],index='train\_id')

print(df2)

df=df.append(df2)

print(df)

df.to\_csv('train\_list.csv')

if f[0]=="2":

p=f[1:]

df.at[int(p[0]),p[1]]=p[2]

if f[0]=="3":

d=get\_info(f[1:],df)

print(d)

con.send(str(d).encode())

if f[0]=="4":

c=False

con.close()

def get\_info(p,df):

id=df.index

if (p[1] not in df):

return 'Not exist'

return df.at[int(p[0]),p[1]]

while True:

con,addr=s.accept()

print('Connection established from ',addr)

thread=threading.Thread(target=handle\_client,args=(con,df,addr))

thread.start()

data=con.recv(1024)

f=repr(data.decode())

print(f)

f=f[1:]

f=f[:len(f)-1]

f=f.split('\_')

print(f)

if f[0]=="1":

p=f[1:]

df2=pd.DataFrame.from\_records([{'train\_id':p[0],'train':p[1],'sleeper':int(p[2]),'3a':int(p[3]),'2a':int(p[4])}],index='train\_id')

print(df2)

df=df.append(df2)

print(df)

df.to\_csv('train\_list.csv')

if f[0]=="2":

p=f[1:]

df.at[int(p[0]),p[1]]=p[2]

if f[0]=="3":

d=get\_info(f[1:],df)

print(d)

con.send(str(d).encode())

if f[0]=="4":

c=False

if f[0]=="5":

dfi=str(df)

con.send(dfi.encode())

con.close()

def get\_info(p,df):

id=df.index

if (p[1] not in df):

return 'Not exist'

return df.at[int(p[0]),p[1]]

while True:

con,addr=s.accept()

print('Connection established from ',addr)

thread=threading.Thread(target=handle\_client,args=(con,df,addr))

thread.start()

dfi=str(df)

con.send(dfi.encode())

def get\_info(p,df):

id=df.index

if (p[1] not in df):

return 'Not exist'

return df.at[int(p[0]),p[1]]

while True:

con,addr=s.accept()

print('Connection established from ',addr)

thread=threading.Thread(target=handle\_client,args=(con,df,addr))

thread.start()

**CLIENT.PY**

**import socket**

**s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)**

**host=socket.gethostname()**

**port=6000**

**s.connect((host,port))**

**print("WELCOME TO RAILWAY PASSNGER RESERVATION DEPARTMENT")**

**s1=0**

**while s1!=4:**

**print("Select the actions below")**

**print('1-add\_info')**

**print('2-update\_info')**

**print('3-get\_info')**

**print('4-stop')**

**s1=int(input())**

**if s1==1:**

**train\_id=input('enter train id:')**

**train=input('enter train name:')**

**sleeper=input('enter number of passengers in sleeper:')**

**ac3=input('enter no of passengers in 3A:')**

**ac2=input('enter no of passengers in 2A:')**

**p=str(s1)+'\_'+str(train\_id)+'\_'+str(train)+'\_'+str(sleeper)+'\_'+str(ac3)+'\_'+str(ac2)**

**s.send(p.encode())**

**print('sent')**

**if s1==2:**

**train\_id=input('enter train id:')**

**sleeper=input('enter class(sleeper/3a/2a:')**

**num=input('enter no of passengers to update:')**

**p=str(s1)+'\_'+str(train\_id)+'\_'+str(sleeper)+'\_'+str(num)**

**p=str(p)**

**s.send(p.encode())**

**print('updated')**

**if s1==3:**

**train=input('enter train id:')**

**sleeper=input('enter class(sleeper/3a/2a:')**

**p=str(s1)+'\_'+str(train)+'\_'+str(sleeper)**

**p=str(p)**

**s.send(p.encode())**

**print('sent')**

**data=s.recv(1024)**

**print('no of pasengers are',data.decode())**

**if s1==4:**

**break**

**train\_id=input('enter train id:')**

**sleeper=input('enter class(sleeper/3a/2a:')**

**num=input('enter no of passengers to update:')**

**p=str(s1)+'\_'+str(train\_id)+'\_'+str(sleeper)+'\_'+str(num)**

**p=str(p)**

**s.send(p.encode())**

**print('updated')**

**if s1==3:**

**train=input('enter train id:')**

**sleeper=input('enter class(sleeper/3a/2a:')**

**p=str(s1)+'\_'+str(train)+'\_'+str(sleeper)**

**p=str(p)**

**s.send(p.encode())**

**print('sent')**

**data=s.recv(1024)**

**print('no of pasengers are',data.decode())**

**if s1==4:**

**break**

Client 3.py:

**import socket**

**import pandas as pd**

**s=socket.socket()**

**host=socket.gethostname()**

**port=6000**

**s.connect((host,port))**

**print("WELCOME TO RAILWAY PASSNGER RESERVATION DEPARTMENT")**

**s1=0**

**while s1!=4:**

**print("Select the actions below")**

**print('5-view data')**

**print('4-stop')**

**s1=int(input())**

**if s1==5:**

**p=str(s1)**

**s.send(p.encode())**

**print('sent')**

**data=s.recv(1024)**

**l=data.decode()**

**d=""**

**for i in l:**

**if i!="\n" and i!="\r":**

**if i==",":**

**d+=" "**

**else:**

**d+=i**

**elif i=="\n":**

**print(d)**

**d=""**

**if s1==4:**

**break**

**else:**

**d+=i**

**elif i=="\n":**

**print(d)**

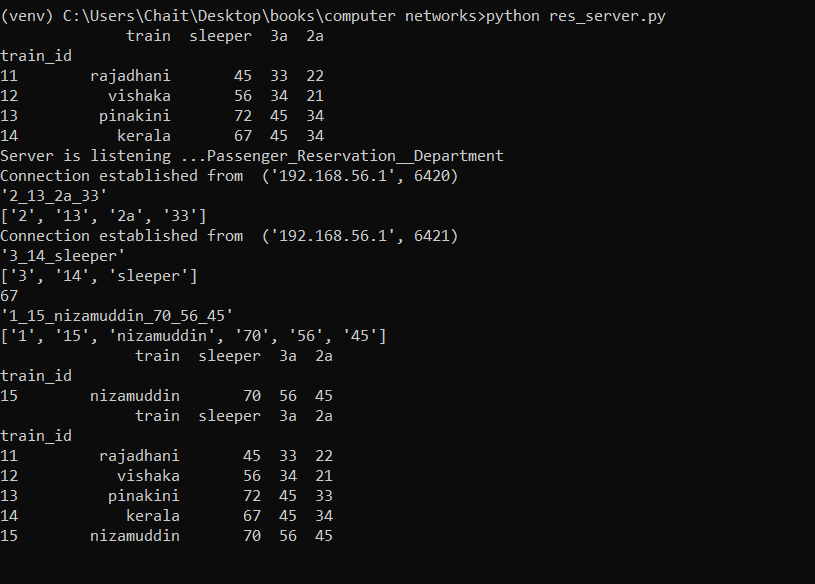
**d=""**

**if s1==4:**

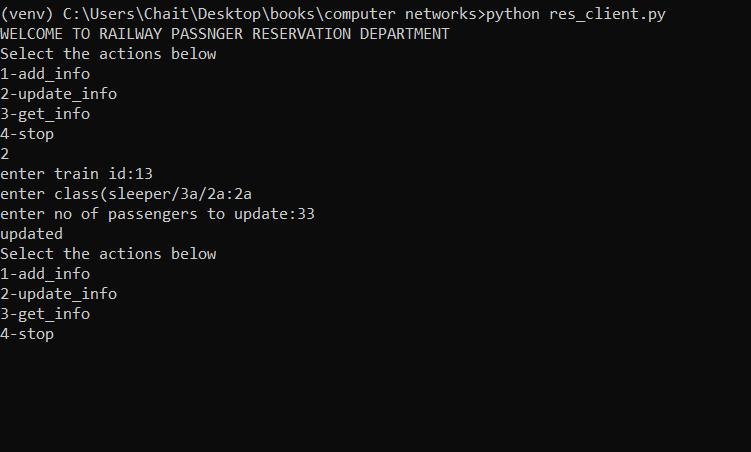
**break**

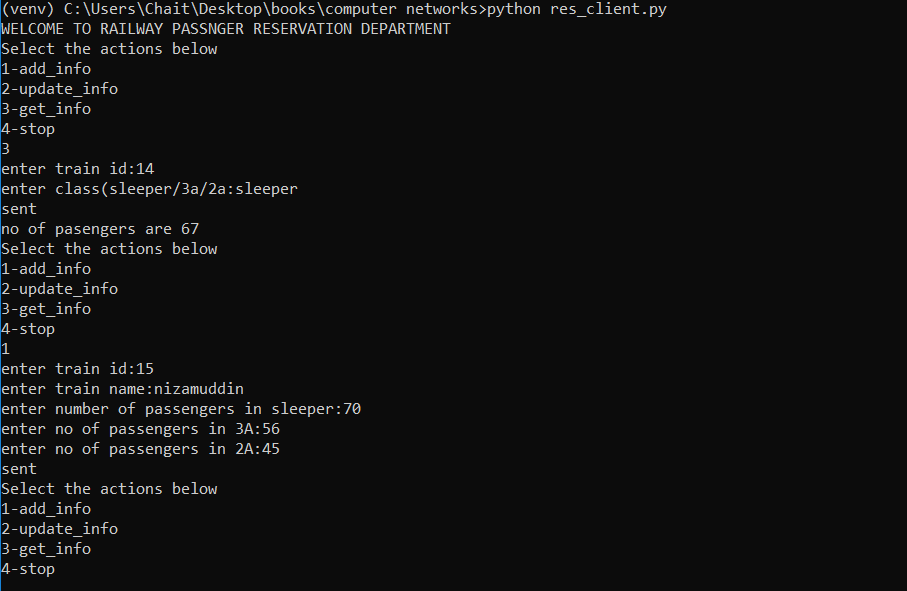
Output screen shorts:

SERVER:



CLIENT 1:



CLIENT 2: 

Client 3:

