Design Document :**Conference Track Management**

**Below Python used for calculation conf track management**

1. **ConfTrackMnmt.py**
2. **confevent.py**
3. **confschedule.py**
4. **confslot.py**
5. **conftests.py**
6. **main.py**
7. **Input file**
8. **resource dir**

**class ConEvent:**

name: Name of event

duration: duration for event

scheduled: scheduled info boolean [true or false]

**class ConfSlot:**

max: Maximum time for slot

min: Minimum time for slot

**Method:ValidEvent:**

Check weather event is valid or not

event time is more than 180 min or combination of multiple events more than (180 or 240)

**Method:ValidSession:**

Check weather session is valid or not..

if session time is more than 0 and total session time is less than 180 or 240

**class ConfTrackManagement:**

morning=[]#list containing morning talks

evening=[]#list containing evening talks

perday=7\*60 #max session in a day

Class Method

**1.Initilazation**

talk\_list eg [Writing Fast Tests Against Enterprise Rails 60min,python class 30min]

total\_time e.g.. 815

**2.readInput file**

Read the input file and storing as event object in talk list

This method takes file as a input

talks = []

**3.schedule**

Schedule events for a list of events

This method takes list of talks as input

sort the talk\_list based on duration

find morning and evening slot

**4.combinations**

This method gives a list of list of events that can be scheduled

for given slot

It takes list of events to be scheduled, possible days and the slot

-morning or evening

e=[] event

**Logic**:

list\_size=len(event\_list)

print('event\_lenght',list\_size)

e=[]

count=0

for i in range(list\_size):

start=i

totaltime=0

comb=[]

while(start is not list\_size):

print('start\_value',start)

curr=start

start=start+1

event=event\_list[curr]

if event.scheduled or not slot.isValidEvent(event,totaltime):

continue

comb.append(event)

totaltime=totaltime+event.duration

print('totaltime',totaltime)

if totaltime>=slot.max:

break

if slot.isValidSession(totaltime):

e.append(comb)

print('e',e)

for talk in comb:

#marking events selected as scheduled

talk.scheduled=True

count=count+1

print('count==possibledays',count,possibledays)

if count==possibledays:

break

return e

**5.totaltime**

Total minutes in a list of talks

This method takes a list as a input

**6.clear**

Clear already scheduled events from list of events

This method takes list as a input

**7.print\_output**

This returns a string of output in a required format

**main.py**

from confschedule import ConfTrackManagement

CONF\_DIR='/Users/vishnoiprem/interview/thoughtworks/ConfTrackMnmt/'

"""

Please insert desire input file Project Path here:

This module create instancs of ConfTrackManagement class and print desire Conference Track Management

for sessions

"""

c = ConfTrackManagement(CONF\_DIR+’/ConfTrackManagement/resource/conf\_test.txt’)

print(c.print\_output())

**conftests.py**

**test\_isEmpty**

Test weather file is empty or not

**test\_isTime**

Test\_isTime

Test total duration of event

**test\_split**