Automatic Time Table Generation

Objective

A system to extract relevant information like personal time table or laboratory time table of an individual faculty from the Master time table, Master time table serves as input source. Also make the master time table with load /venue allotment for subjects as per the input.

System Processing Details

This algorithm is designed to solve and generate college time tables.
Assumptions:
• The algorithm produces optimum outputs for a five-day week.
• The number of subjects (s1, s2, …, sn) need to be finalized before the algorithm begins execution.
• Number of teachers (t1, t2, …, tn) entered before execution of the algorithm are assumed to be constant and cannot be changed during or after the algorithm has been executed.
• Any change in the above two assumptions will require a new generation of Timetable for the changed data.
• In each time table, all time-slot are filled with a unique combination of subjects without any repetition of subjects.
• Any teacher is allowed at most ‘k’ number of lectures in a week. The value of k is accepted before execution of the algorithm.
• It is assumed that a teacher cannot take more than one lecture for the same class in a day.
• Timeslots ts1, ts2, … ,tsn once entered at the beginning cannot be changed throughout the execution.
• Every day in the week is assumed to have equal number of time slots.
• Classrooms for any batch are fixed throughout the day.

Variables

Used Time slots of the time tables:- ts1, ts2, ts3…., tsn
List of Subjects:- s1,s2,s3, …., sn
Teachers:- t1,t2,t3, …., tn
Batches of students:- c1,c2,c3, ….., cn
Flags indicating finalized timeslots :- tsf1, tsf2, tsf3, ….., tsfn
Data structure to hold Final Timetable:-final_tt
Count for day of week: Weekday
Number of days of the week:- n
Data structure to hold Subject-clash within the day:- clash
Each element of Clash data structure:- clash_ele
Data structure for Subject-clash across days:- Dayclash
Each element of Dayclash data structure:- day_clash_element
Subject contained in dayclash:- sdc
Teacher associated with subject in dayclash:- tdc
Max number of lectures of subject si in the week:-k
Counter for the number of subjects:- counter_sub
Random number indicating random slot allotment for subject:- rand_sub_allot
Data structure to hold randomly allotted subject:- rand_sub_seq
Data structure to hold all subjects:- init_sub

Input from the Master TImeTable
Time slots of the time tables:- ts1, ts2, ts3…. Tsn
List of Subjects:- s1,s2,s3, …., sn
Teachers:- t1,t2,t3, …., tn
Max number of lectures of subject si in the week:- k
Batches of students:- c1,c2,c3, ….., cn
Count for day of week: Weekday

ALGORITHM

```
If(Weekday>n)
    end
generate()
Lbl2:For  day_clash_element 1 today clash element n
        Retrieve sdc from dayclash
    Retrieve tdc of sdc
rehabilitate(sdc)
For(ts1 to tsn)
       If(si exists in final_tt)
          Next iteration
Else
       Lbl3:Retrieve si in tsi
       Retrieve ti of si
       If(availability =0 for tsi)
           rehabilitate(si)
        Else
        If(ki>0)
           Set si to tsi in final_tt
           tsfi=1
           ki - -
Else
Lbl3
   If (tsn has been reached)
      If(clash NOT empty)
         For clash_ele1 to clash_elen
         Retrieve each si inclash_elen
         Retrieve si with ti
         rehabilitate(si)
         Weekday++
  Else
         Weekday++
  Else
  Continue
generate()

For (each subject si)
  Place si in sub_arr
For(each dow)
      rand_seq=rand(sub_arr)
      if(dow=1)
      init_tt (dow)=rand_seq
   else
      curr_pos=length(rand_seq)-1
       temp_ele=rand_seq(last)
for (each element in rand_seq)
         if(curr_pos==1)
         rand_seq(curr_pos)=temp_ele
   else
         rand_seq(curr_pos)=rand_seq(curr_pos-1)
         init_tt(dow)=rand_seq
End

generate
rehabilitate(si)

Lbl4:
Retrieve tsj such that ti availability =1
 If(tsfj=0)
      If(ki>0)
         Move sj to Clash
         Set ti,si in tsj in final_tt
         tsfi=1
         kj--
      else
 Lbl4
 Else if(tsfj=1)
Lbl4
 Else If(all tsfj=1)
 Mark sj
Move sj to Dayclash
End rehabilitate
```

System Specifications

Operating System: Android 5.0 or above
Ram:512MB or above

Outcomes

	Mon	**Tue**	**Wed**	**Thurs**	**Fri**
TS1	**Mr.A/ Maths**	**Mr.B/ Physics**	**Ms.C/ Chemistry**	**Ms.D/ EM**	**Mr.E/ SPA**
TS2	**Mr.B/ Physics**	**Mr.A/ Maths**	**Ms.D/ EM**	**Ms.C/ Chemistry**	**Mr.E/ SPA**
TS3	**Ms.C/ Chemistry**	**Ms.D/ EM**	**Mr.E/ SPA**	**Mr.B/ Physics**	**Mr.A/ Maths**

TS4	**Ms.C/ Chemistry**	**Ms.D/ EM**	**Mr.A/ Maths**	**Mr.E/ SPA**	**Mr.B/ Physics**
TS5	**Ms.D/ EM**	**Mr.A/ Maths**	**Ms.C/ Chemistry**	**Mr.B/ Physics**	**Mr.E/ SPA**

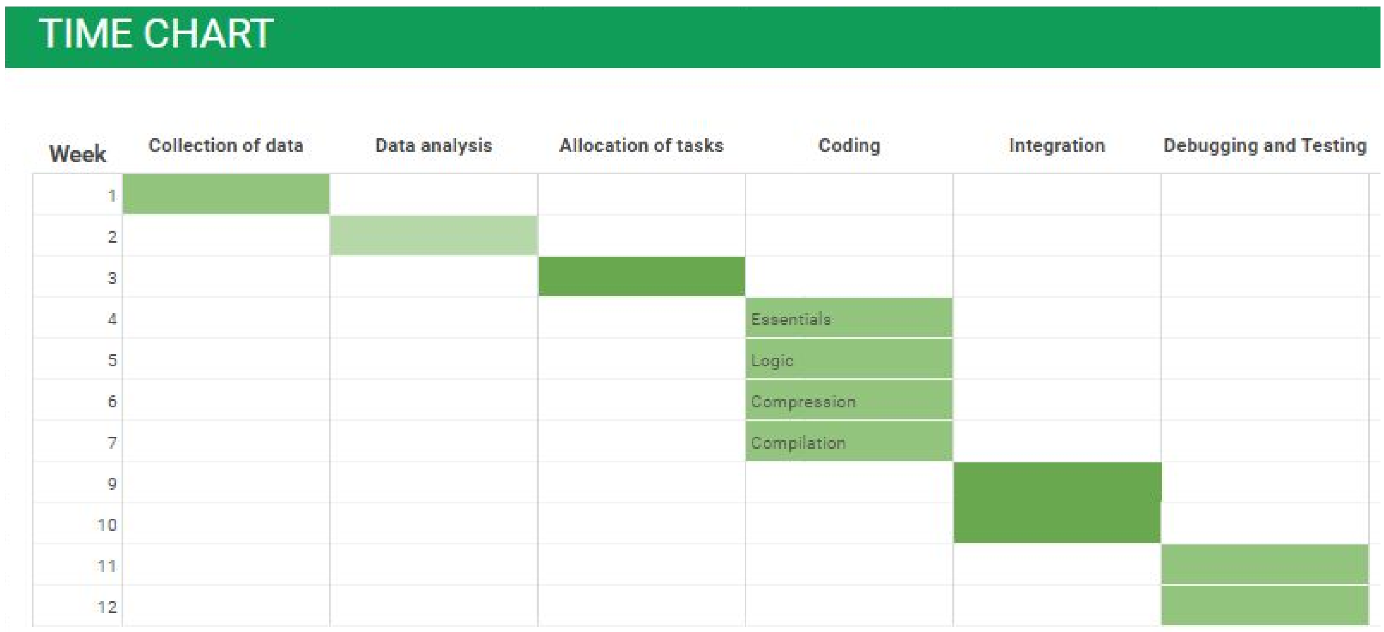
The algorithm after implementation, results in the creation of a time table of batch/class of students displaying a grid of time slots.

• Each time slot is filled by a teacher and the subject that is being conducted. The output of the algorithm's implementation will be as per the above Table

• The allotments of teachers to the slots will change the composition of the generated time table. Hence, all clashes of availability of teachers will be analyzed and the algorithm will be applied again to improve by reducing the clashes.

• Hence, all clashes of availability of teachers will be analyzed and the algorithm will be applied again to improve the generated schedule by reducing clashes.

Time chart



TIME CHART

Week	Collection of data	Data analysis	Allocation of tasks	Coding	Integration	Debugging and Testing
1	🟩					
2		🟩				
3			🟩			
4				Essentials		
5				Logic		
6				Compression		
7				Compilation		
9					🟩	
10					🟩	
11						🟩
12						🟩

COST

Minimal cost will be incurred except for the system and internet costs.

Group Members

1.Mayur Singal
2.Esha Vijayvargiya
3.Kanishk Sonee
4.Harsh Vyas
5.Sneha Yadav
6.Samdharsi Kumar