**1. INTRODUCTION**

Time Table Management system is an automated system which generates time table according to the data given by the user. The main requirement of the application is to provide the details about the branch, subjects, number of labs, total number of period and details about the lab assistance. Then the application generates the time table as per the requirement.

**1.1 Problem Statement**

Development of **Timetable Management System** to generate and manage timetable for an institution.

**1.2 Problem Definition**

**Timetable Management System** is an automated system developed for any institution. Timetabling is the allocation, subject to constraints, of given resources to objects in space-time domain to satisfy a set of desirable objectives as nearly as possible. Particularly, the university timetabling problem for courses can be viewed as fixing in time and space a sequence of meetings between instructors and students, while simultaneously satisfying a number of various essential conditions or constraints. Planning timetables is one of the most complex and error-prone applications. There are still serious problems occurring and these problems are repeating frequently. Therefore there is a great requirement for an application distributing the courses evenly and without collisions. Graph coloring algorithm is one of the most used algorithms. [1]

Manually creating a timetable and managing it is a very tedious job and maintain all the databases required to design the timetable is very time consuming. So we intend to design an automated system which will generate the timetable considering faculty details, teacher details, semester details etc.

* 1. **Objectives**

To create a **Timetable Management System** to be used by any Institution and will reduce time consumption and the labor of manually designing the timetable and managing it. To create a leave management system to manage the leaves of faculties and to post a notice for substitute class or extra class.

* 1. **Tools & Platform**
  2. **Discussion On Problem**

In a typical semester, the courses are required to be scheduled at different times in order to avoid conflict. The formal definition for university timetable is given in detailed by Carter et al (1995).In event that the following resources and time slot exist as follows:

• A set of lecturers {t1, t2,.............,tn}

• A set of subjects {c1, c2,.......,cm}

• A set of classes {r1, r2,...............,rq}

• A set of time {p1, p2..................ps}

As a preliminary step, construct a three verse set to represent the requirement which are (i=1, 2....n), (j=1, 2..........m) and (k=1, 2....q). Each set indicates lecturer ti lecturing the subject cj in class rk. Further, determine each three verse set to time p1 which is (i =1, 2.....s). This will form a four verse set < ti, cj, rk, p1> which represents the stipulation that lecturer t, lectures the subject c, class r and at p time. For every two four verse set, for example < te, cf, rg, ph> < tw, cx, ry, pz> and the constriction below must be fulfilled. :

If te = tw hence ph ≠ pz (1)

If cf = cx hence ph ≠ pz (2)

If rg = ry hence ph ≠ pz (3)

Constriction (1) ensures that no lecturer will scheduled for more than once in each period, constriction (2) ensures that no subject will be scheduled for the same time and constriction (3) ensures that no class is scheduled for more than once for each period

The problem is to design an algorithm to create a semester course timetable by assigning time slots and rooms to given set of courses to be run that semester under given constraints. The constraints include avoiding clashes of time slots and rooms, assigning appropriate rooms and appropriate number of slots and contact hours to the courses etc. Although most of the college administrative work has been computerized. The lecture timetable scheduling is still mostly done manually due to its technical difficulties. The manual scheduling of lecture-timetable required considerable time and efforts. The lecture-timetable scheduling is constraints satisfaction problem in which we find an optimal solution that satisfies the given set of constraints.

**1.5.1. Constraints:**

Constraints are the fundamental elements of a timetable that the software use to create the timetable and must be abided by.

**1.5.1.1 Hard constraints**: All hard constraints must not followed, no hard constraints can be violated in any case.

1. A classroom is not assigned to more than one lecture at same time.
2. An instructor cannot teach more than one subject at the same time.
3. Two Courses for the same year-session students of a department cannot take place at same time except for elective subject in theoretical class and two different labs for which department are divided into sub groups may be scheduled together.
4. The lectures are not allotted to time slots which come under the lecturer’s prohibited time zones.
5. Practical subjects can start from 1st, 2nd and 5th period only and are held for 3 hours continuously so they end on 3rd, 4th and 7th period respectively.

**1.5.1.2 Soft Constraints**: It is desirable to satisfy all soft constraints but may be sacrificed to find a feasible timetable.

1. Practical classes of 4th year should get first preference and must be scheduled in the first half i.e. first four periods.
2. All non-laboratory classes should be scheduled in the second half i.e. last 3 periods for any year.
3. A subject should not have more than one class on a particular day.
4. All teachers should have a free day i.e. no classes are assigned on that day.
5. No teacher should have class on first period more than three times in a week.

We want to develop an algorithm for developing an effective and practical timetabling algorithm which will satisfy constraints. We primarily focused on developing algorithm which is easy to implement without compromising on it’s effectiveness and performance. As represented by the constraints designing a timetable is a very complex job. For this reason, creating a reliable automatic server which requires no manual intervention is a very difficult problem and most organizations do not have such a solution. Instead, most timetables are created manually by expert administrator who have deep knowledge of all requirements of all parties involved. Our system will provide a feature which will automatically manage the leaves.

**1.5. Assumptions**

The application is designed for St. Thomas’ College of Engineering and Technology on that basis following assumptions are made while designing the system:

1. Classroom are assigned to each year and department manually for their theoretical lecture.
2. Classes are held from 9:30am -5:30pm on weekdays and have 7 periods for lectures/labs to be assigned excluding recess time.
3. Classes are held from 9:30am -1:45pm on Saturday and have 4 periods for lecture/labs to be assigned excluding recess time.
4. On some days classes may start late or finish early so starting or ending periods can be empty.
5. Faculty are assigned the subjects they will teach by Head of the Department manually.
6. Users have basic knowledge of computer.

**2. Literature Review**

There are few research working papers on university timetable that are published. Most of the timetable system produced can only be used at the university involved in the research, as each university has its own needs and requirements that differs specifically.

**2.1. Graph Coloring:**

Most timetable software is based on graph delegation. Graph can be defined as vertex connected by lines where dots will represent events and line will correlate with those events. A vertex in Graph represents a subject, an edge represents a pair of courses that conflict, and a color represents the period in which that particular course is to be scheduled. The objective is to find minimum color and make a suitable timetable using it. The minimum coloring problem and the timetabling problem have been classified as NP-hard problems in the general case. This means that it is unlikely that it will be possible to find fast (i.e., polynomial-time) algorithms to solve these problems. In order to find optimal solutions to such NP-hard problems, it is usually necessary to consider all possible solutions to choose the best one.[2]

1.6.1.1 Strength: No conflict occurred because different colors were used for every subject

1.6.1.2 Weakness: Could not represent certain elements such as connecting two different subjects with different time.

**2.2. Network flow technique:** Dyer and Mulvey (1976), Mulvey (1982), Chahal and de Werra (1989), Dinkel et al (1989) propose to use a network model as the core of the timetabling algorithm. The network employed by Dinkel et al. contains three levels, plus a source and a sink vertex. The first level is the Department Level which includes a vertex for each department, such that all of these vertices are connected to the source. The second level is the Faculty/Staff Level which includes a vertex for each possible combination of teacher and course taught by the teacher; these vertices are connected to the vertices representing the departments to which the teachers belong. The third level is the Room Size/Time Level, which contains a vertex for each combination of room and time. Each vertex of this level is connected to a vertex of the second level only if the size of the room represented by the vertex in compatible with the number of students of the course represented by the other vertex. An edge between levels 2 and 3 represents a possible lecture. The capacities and the lower bounds of edges representing the lectures are 0 and 1 respectively and due to uni-modularity this ensures that the optimal solution to the problem will possess all integer values. The coefficients of the objective function are assigned based on availabilities of teachers and rooms and preferences of the teachers. [3]

1.6.2.1 Strength: Allocate time for the combination of lecturer-classroom whereby no conflicts generated in the schedule.

1.6.2.2 Weakness: The network model does not prevent the solution from assigning a single teacher to multiple lectures at the same time.

**3. Concept and problem analysis**

We want to develop a system which will generate the timetable as well as mange it. Admin and user both will have to login to access the software. The system has three end user Admin, Head of the Department and Faculty. Users are provided with following features

* Admin:
* Generate timetable
* Add/Remove user
* Add/Remove lab
* Head of the Department:
* View timetable
* View leave request
* Approve/Reject leave
* Faculty:
* View Timetable
* Apply for leave
* View substitute request
* Register request
* Give subject preference

User

**4.1 Data Flow Diagram**

**4.1.1 Data Flow Diagram Level 0**

A D M I N

View timetable

Generate timetable

View user details

Add user

H

O

D

F

A

C

U

L

T

Y

Assign subject

Post notice

Response to leave request

View leave box

Read Leave Request

View the response

Give Subject Preference

Figure 1: Data Flow Diagram Level 0

Figure 2: Data Flow Diagram Level 1

**4.1.2. Data Flow Diagram Level 1**

Leave Response

Leave Request

HOD

Faculty

Leave

Preference

Subject assigned

**Admin**

**Faculty Details**

Timetable

Subject preference

Preference details

Faculty’s preference

Subject assignment details

Generate Request

Subject assignment details

Timetable

Timetable

View request

View request

View request

Login credentials

Login credentials

Login credentials

Login credential

Login credential

Login credentials

Login access Logcredentials

Login access Logcredentials

**4.2 Entity Relationship Diagram**

1

n

Faculty

Class

takes

1

1

1

n

Lab

preferred

n

Leave

requests

n

assigned

n

Subject

Figure 3: Entity Relationship Diagram

**4.2.1. Attribute**

* Faculty( FacultyId, Name, Password, Post, Teaching\_hours, Admin\_rights)
* Class( Year, Dept, Semester, Period)
* Subject(Code, Sub\_name, Semester, Theory, Practical)
* Leave(FacultyId, Start\_date, Duration)
* Lab( Name, Code)

**Sample Code:**

**Code of class definition**

public class Subject {

String subjectID;

int sem;

String dept;

int deptcode;

int year;

String subjectName;

int theortical;

int practical;

int tutorial;

}

class Sclass{

int sem;

int year;

String dept;

Subject[][] timetable=new Subject[6][7];

Sclass()

{

for(int i=0;i<6;i++){

for(int j=0;j<7;j++){

timetable[i][j]=new Subject();

}

}

for(int i=4;i<7;i++)

{

timetable[5][i].subjectID="X";//Represents classes cannot be alloted.

}

}

}

class Teacher

{

String name;

String tID;

Subject[][] timetable=new Subject[6][7];

int teaching\_hours;

LinkedList<Subject> subQ =new LinkedList<Subject>();

Teacher()

{

for(int i=0;i<6;i++)

{

for(int j=0;j<7;j++)

{

timetable[i][j]=new Subject();

}

}

for(int i=4;i<7;i++)

timetable[5][i].subjectID="X";

}

public void notavailable(int day,int start\_period,int end\_period)

{

for(int i=start\_period;i<=end\_period;i++)

{

timetable[day][i].subjectID="X";

}

private boolean non\_teachingday(int day){

for(int i=0;i<6;i++)

{

for(int j=0;j<7;j++){

if(timetable[i][j].subjectID!=null&&timetable[i][j].subjectID!="X")

break;

if(j==7)

return false;

}

}

return true;

}

public int first\_period(){

int day=0;

for(int i=0;i<6;i++)

{

if(timetable[i][0].subjectID!=null && timetable[i][0].subjectID!="X")

day++;

}

return day;

}

private float day\_ratio(int day)

{

int unassigned=0,assigned=0;

for(int j=0;j<7;j++)

{

if(timetable[day][j].subjectID==null)

unassigned++;

else

assigned++;

}

return ((float)unassigned/(assigned+unassigned));

}

public int max\_day\_ratio()

{

float max=-1,x;

int day=6,free\_day=-1;

for(int i=0;i<6;i++)

{

x=day\_ratio(i);

if(x>max)

{

if(x==1 && free\_day==-1)

{

free\_day=i;

}

else

{

max=x;

day=i;

}

}

}

if(max==0)

return free\_day;

else

return day;

}

}

public class Lab

{

String labName;

Subject[][] timetable=new Subject[6][7];

LinkedList<Subject> labQ =new LinkedList<Subject>();

int labmax=10;

Lab()

{

for(int i=0;i<6;i++)

{

for(int j=0;j<7;j++)

{

timetable[i][j]=new Subject();

}

}

for(int i=4;i<7;i++)

timetable[5][i].subjectID="X";

}

}

1. **Conclusion:**

The primary objective in preparing timetable is obtaining conflict free for each activity sharing the same resources. Most of the method faces difficulties in generating a feasible timetable which faces long processing period without complying with several conditional constraints. The aim of the project is to create timetable in an efficient way which will be conflict free and abiding by all the constraints.

Future Work

* Leave Management module to assign substitute teacher when one teacher is on leave or absent.
* Room Management for tutorial/elective subjects.
* Assigning a teacher to more than one department for a combined class of more than one department.

1. **Reference:**

Books:

* Andrea Schaerf. A survey of automated timetabling.Technical Report CS-R9567, CWI - Centrum voor Wiskunde en Informatica, 1995
* Student Time Table By using Graph Colouring Algorithm: Baki Koyuncu ,Mahmut Seçir.
* Dinkel,J,J.,Mote,J., & Venkataraman, M.A.(1989). An efficient decision support system for academic course scheduling.
* Database System Concepts by Avi Silberschatz,Henry F.Korth & S.Sudarshan.
* Fundamentals of Database System by Elmasri Navathe.

Website:

* http://www.cs.qub.ac.uk/itc2007/