PROJECT REPORT

R.G.M.S

Routine Generator & Management System

Submitted by:

AMRITAM SARCAR
ISHAN CHAKRABORTY
SOURISH SENGUPTA

Pursuing

"BACHELOR OF TECHNOLOGY"
IN
"COMPUTER SCIENCE & ENGINEERING"

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ST. THOMAS' COLLEGE OF ENGG. & TECHNOLOGY

(Affiliated to WBUT)

KOLKATA-700023

CONTENTS AT A GLANCE

SYSTEM ANALYSIS

1.1. General Description:

The coordinator[1] of an institution[2] is required to prepare a Routine of the regular classes that are held in the institute. This requires designing some preliminary charts, knowing institute specific conditions (Ex. are no. of periods on a day, no. of working days on a week etc.) as well as consideration given to external issues such as subjects taught by the diff. instructors[3] and their individual status vis-à-vis their designation. The project must therefore consider both the strategic and external issues centered around building and managing the Routine. The Routine thus created by the coordinator must meet a number of objectives or constraints that are forced on it by internal and external issues. The most important of these is to maintain a Routine that caters to the need of individual instructors. Its goal is to deliver a Routine that is both optimized and feasible. To achieve this goal, it is mandatory to maintain adequate decision systems at each step and to properly schedule the Routine.

1.2. Defining the problem:

The overriding goal here is to ensure that ***the Routine is both feasible and optimized such that the various constraints are met***. The user requirement may be stated explicitly, given a number of subjects, set of institution specific information, teacher constraints etc. the system must be able to generate a Routine.

A total of **S** subjects must be distributed amongst **N** periods in a time-frame satisfying the following condition:

Total of **N** periods<= **P** periods/day * **D** days/week.

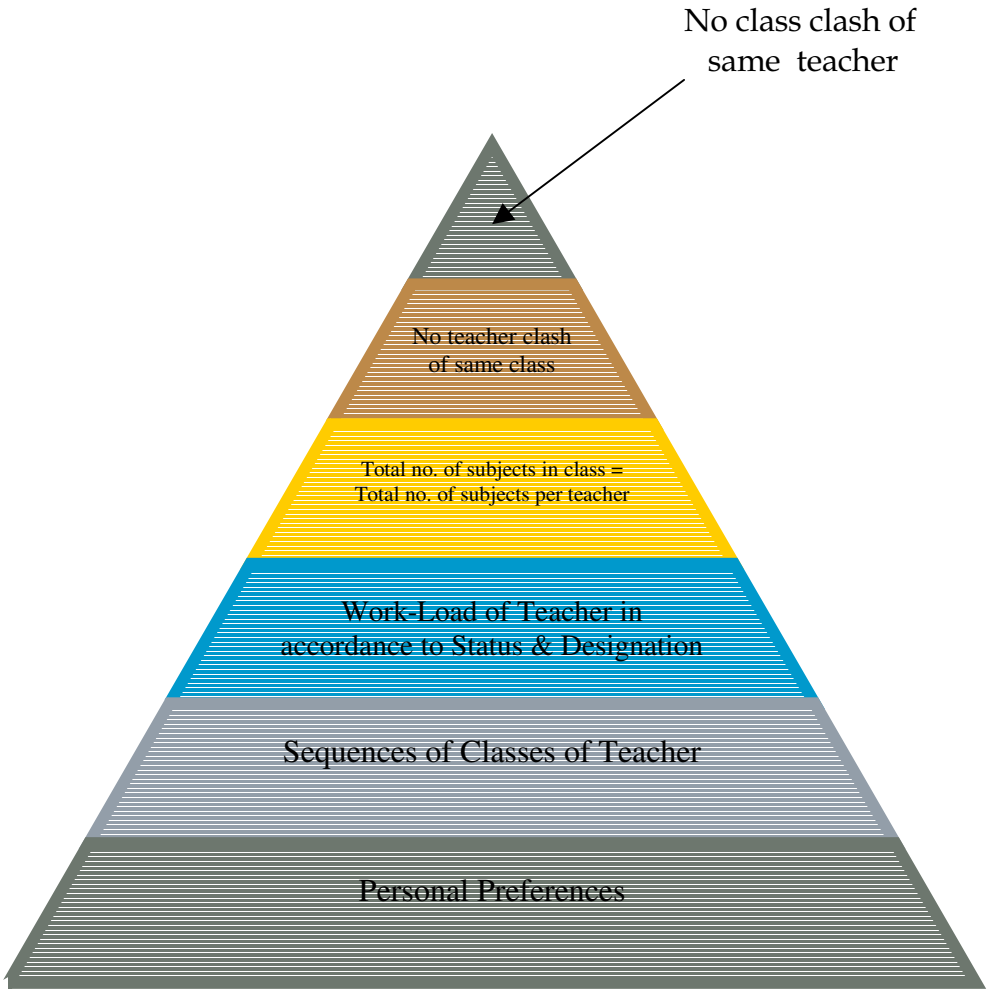
From the above inequality the following conditions may be derived:

- **P<S** i.e., number of periods on a day is less than number of subjects
  Allocate one subject /period /day and some other subjects.
- **P=S** i.e., number of periods on a day is equal to number of subjects
  A simple solution where one subject /day is allocated.
- **P>S** i.e., number of periods on a day is greater than number of subjects
  A generally impossible case. However it may be adjustable as per management decision.

---

[1] an official of the institution given the task for creation and managing a Routine
[2] college, school or university where regular classes are held to instruct students.
[3] teacher, professor or any person who takes regular classes in the institute.

1.3. Points To Ponder(PTP):



Priority-based Triangle depicting the diff. Priorities of the system constraints

Distributing the subjects amongst different periods must be done in accordance to all/some of the below mentioned points:

1) No class clash of same teacher.

A particular teacher cannot be allocated at the same time (same period, same day) to take classes to two or more different classes. This is not a feasible condition.

2) No teacher clash of same class.

In a particular class, two or more teachers cannot take the same class.

The above two conditions are of the Pigeon Hole Principle(PHP) for CBS(computer based suggestion).

3) Total number of subjects for all years(classes) and departments = total number of subjects allocated to different teachers.

Mathematically,

$$\sum_{i}^{N}\sum_{j}^{M} Y_i * D_j = \sum_{p}^{X}\sum_{q}^{R} S_p * T_q \qquad ;1<=i<=N(\text{number of years/classes})$$

$1<=j<=M$(number of departments/sections)
$1<=p<=X$(number of subjects)
$1<=q<=R$(number of instructors)

4) The work load of every teacher must be in accordance to their respective status and designation.

For every teacher there is a status and designation allocation. Accordingly, the teacher has to take **A** number of maximum classes in a week. The following table would exemplify it:

Status[4]	Designation[5]	Maximum no. of classes[6]
Visiting	Professor	18
Permanent	Academic Coordinator	21
Part-Time	Assistant Teacher	33
Visiting	Lecturer	27
Permanent	Head of Department	22

However, in doing so if condition (4) is altered then for that particular teacher (or group of teachers) this priority has to be changed.

5) Sequence of classes of a Teacher

There may be a situation where a teacher who comes 5 days / week is allocated in such a manner that all his allocation falls on the first two days of the week itself.
The following table shows a routine of a teacher whose subject allocation is feasible but not practical.

[4] The different status depends on the Management
[5] The designation are to be inserted by the Management
[6] This is just an example(User-defined)

Day	Period1	P2	P3	P4	P5	P6	P7	P8	P9
Mon	A	A	B	B	A	C	-	D	E
Tue	A	A	D	C	C	E	-	A	C
Wed	-	-	-	-	-	-	-	-	-
Thus	-	-	-	-	-	-	-	-	-
Fri	-	-	-	-	-	-	-	-	-
Sat	-	-	-	-	-	-	-	-	-

a. Maximum how many classes on a day by a teacher teaching the same subject

In the above table subject A is being taught 3 times on the same day. Hence this is not the best solution. A better solution is to be found out.

b. Maximum how many runs of a sequence per teacher

A subject that is being taught continuously for more than 1 period is termed as a Run. Hence the management must decide on how many runs to keep for a teacher or does not keep at all.

c. How many classes in a Run?

The management must decide on how many classes would comprise a run (ex. 2,3, and so on)

6) Personal Preferences:

a. Subject Preference.

b. Reservation Preference.

1.4. A Detailed Look:

When one looks at the whole system one can see a number of activities that can be related to the goal , as well as those that are not directly related. Those that are directly related include:

- Allocation of subjects that are to instructed by the diff. instructors; and
- Arrangement of classes vis-à-vis teachers as per Routine specifications.

Activities not directly related to the goal may include:

- Insertion and modification of Instructor Information;
- Insertion and modification of Routine specification; and
- Insertion and modification of different subjects that are taught.

In addition to the diff. activities that would be directly related to the goal, the information or data that is also directly involved are:

- Specification of institution timings. This may include number of working days in week, number of periods, duration of each period etc.;
- Availability constraints of different instructors;
- The diff. subjects taught by the instructors; and
- Designation and status of the instructors.

1.5. Feasibility Analysis:

There are a number of possibilities for meeting the goal determined in System Analysis. Not all of them use computers. For example, we may suggest the following:

- Employ an (external) agency to structure and hence schedule the Routine;
- Single-person (coordinator) developing the routine; and
- Appending the Routine on an existing routine.

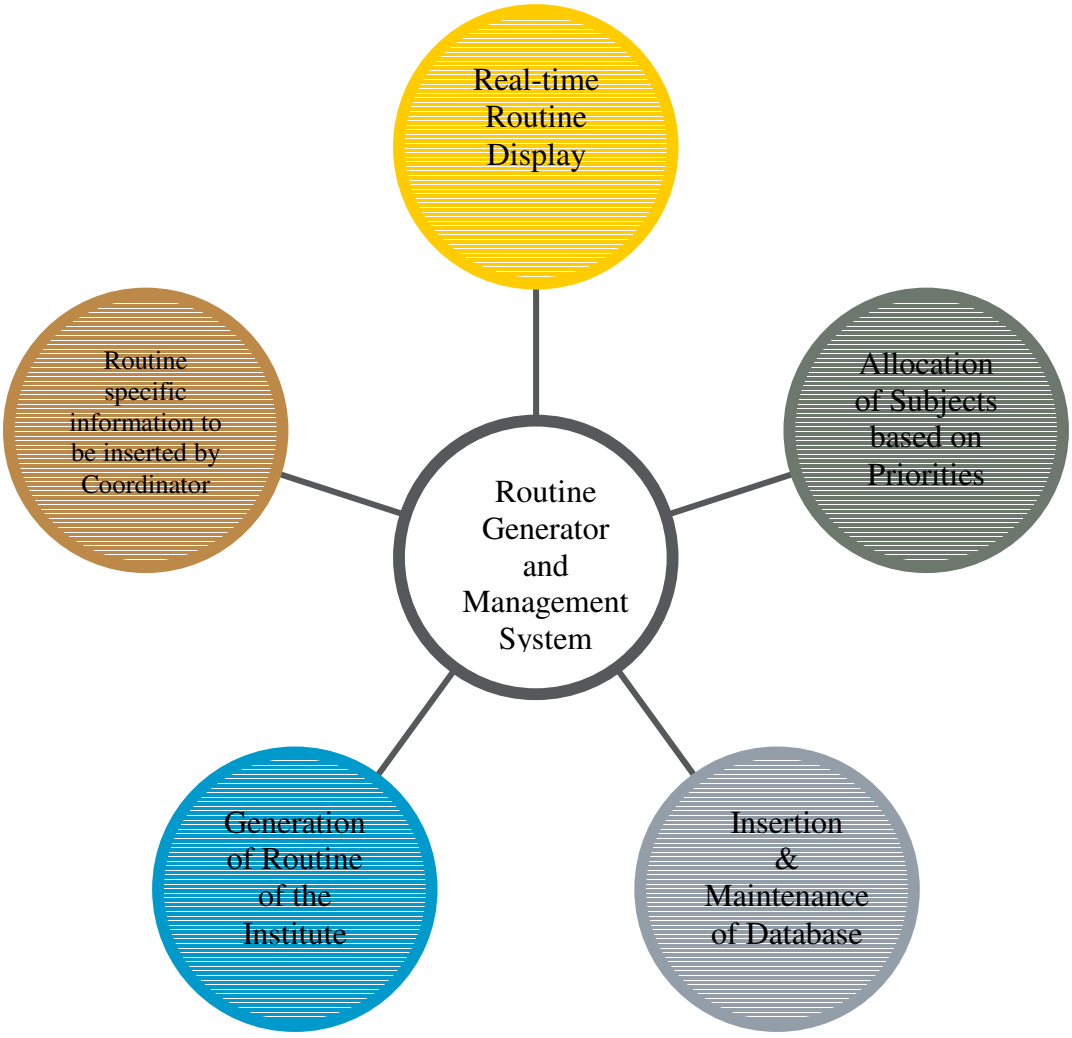
There is also, of course, a computer solution, especially that of using Software system. Here we might have a computer with a database that contains details of diff. constraints and information. Coordinator can enter information about the different specifications of the Routine. Users can browse through this database at any time to enquire about their Routine details through the computer. A more sophisticated approach would be to allow users to place different constraints into the system stating their respective requirements.

Let us consider each of the possibilities in turn: The first option is obviously not operational. Understanding the complicacy of the different constraints probably cannot be overcome. Again, it incurs an extra cost on a yearly basis. Only a single-person developing the Routine is obviously impractical and not feasible for a person to optimize the Routine. Modifying the Routine using an existing routine is not a very practical solution. This is because the routine(designed previously) will obviously fail to cater to the present need of the teachers. All of the above ways suffer from a basic problem: It is impractical for a human to work with hundreds of teachers and their different constraints and then come up with a *complete* solution.

The decision is made, in this case, to develop a software scheme for the above problem. The concept is then elaborated into a broad statement of user requirements, which is to support on-demand individual specifications. In detail, it requires:

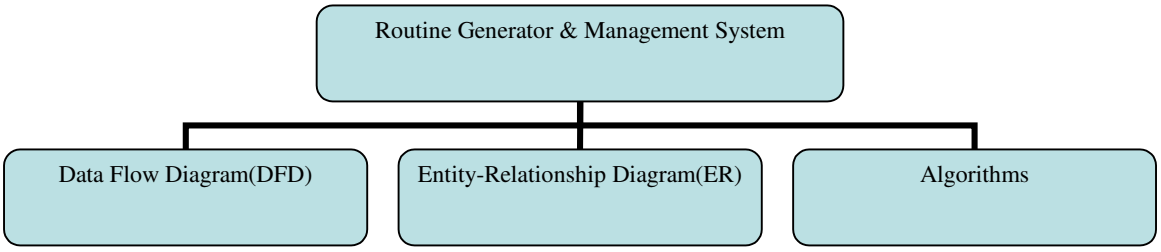
- direct entry of Personal Information of instructors into the database by the coordinator;
- users to be able to browse the database using pre-specified schemes;
- direct display of instructor-subject-class relationship; and
- users to be able to view the Routine.

1.6. What the system will offer?



DESIGN ISSUES
FOR
ROUTINE GENERATOR AND MANAGEMENT SYSTEM

The design techniques that are used for the depicting the working of the different parts of the software can be broadly classified as per the following diagram:



DATA FLOW DIAGRAM

The different processes of the System are hereby depicted using Data Flow Diagram(DFD):

LEVEL 0:

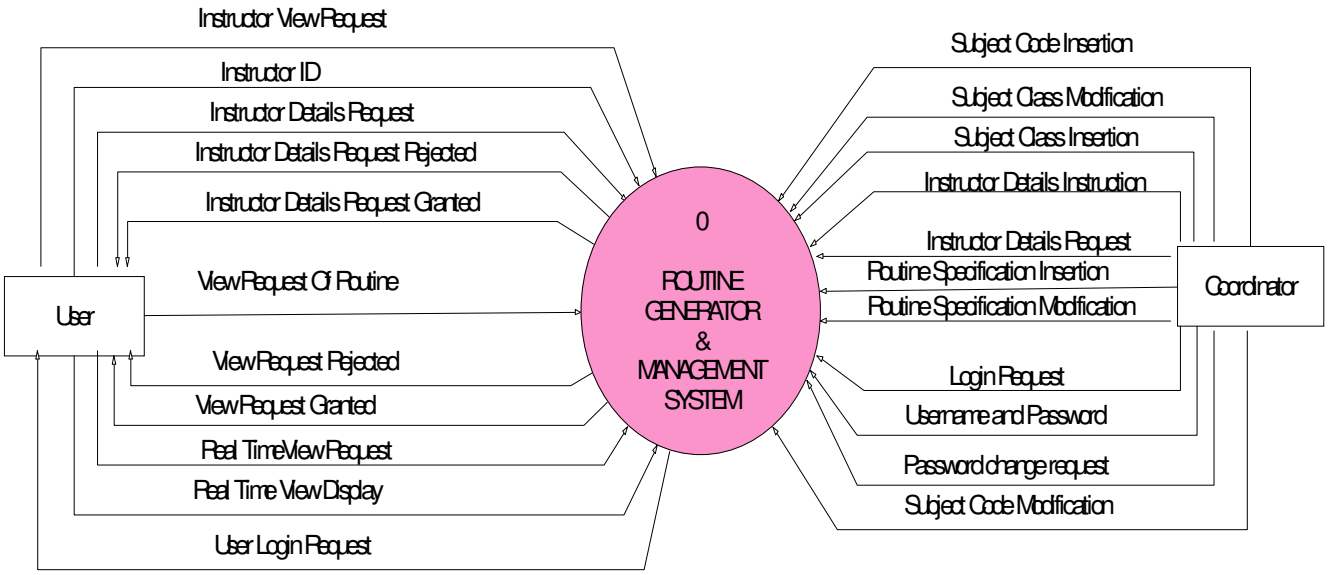


Fig1.0: The Context Diagram of RGM System

LEVEL 1:

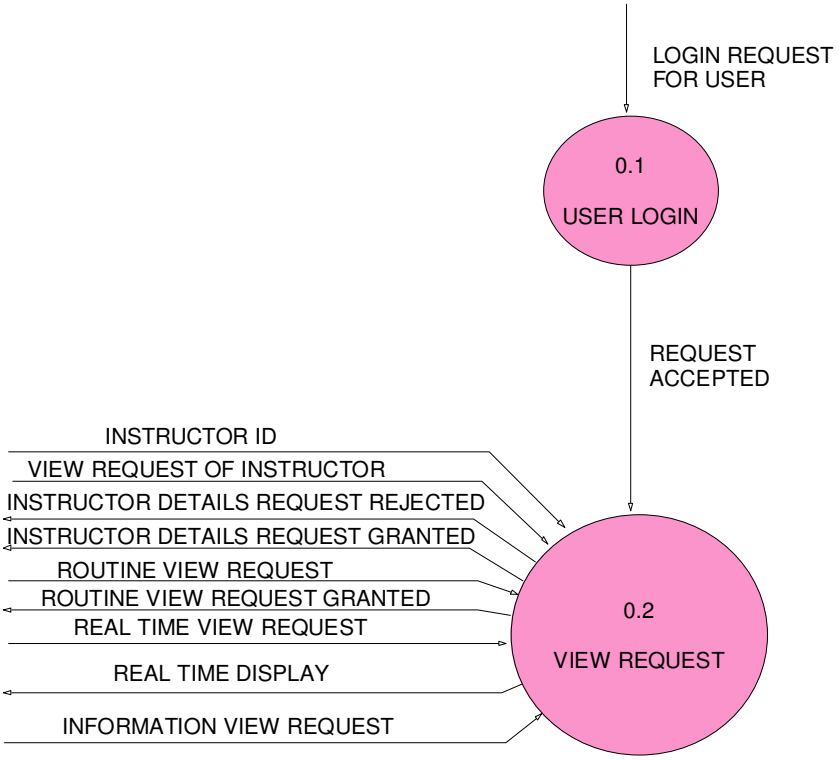


Fig.2.0: Level 1 diagram of the User Login and Request

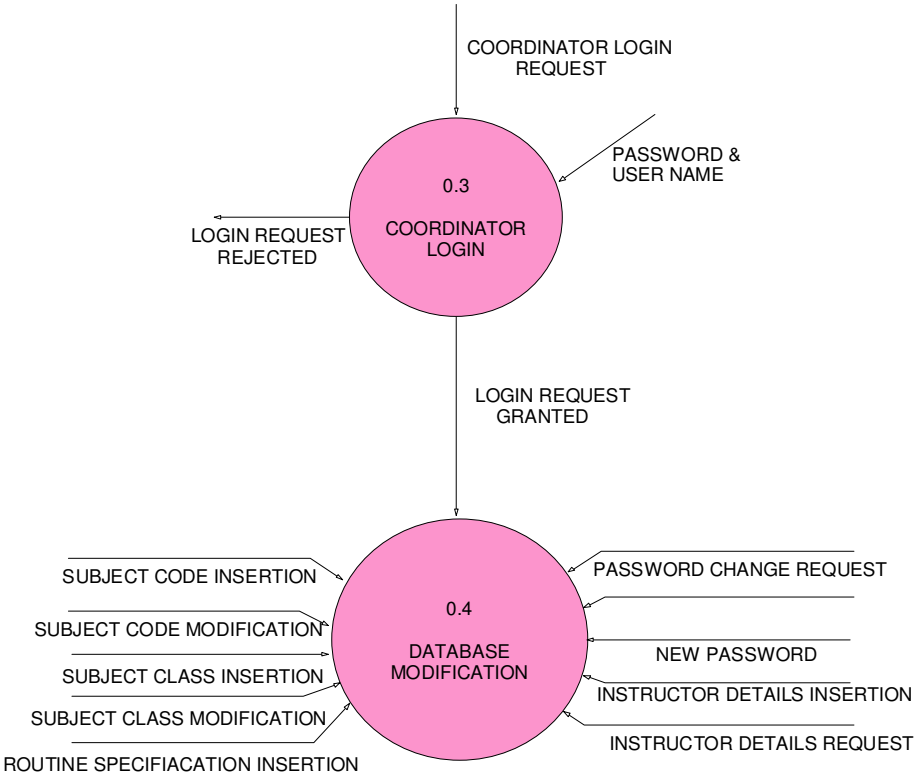


Fig.2.1: Level 1 diagram of the Coordinator Login and Database Modification

LEVEL 2:

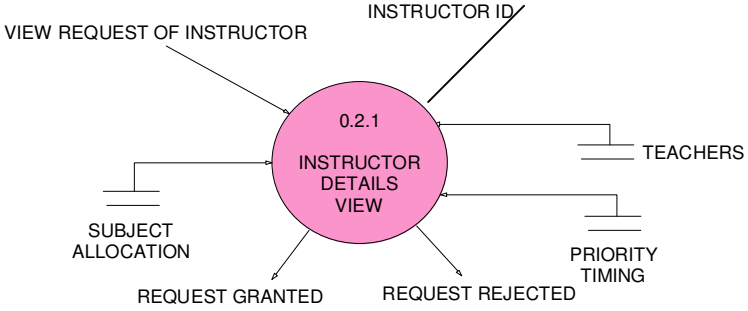


Fig.3.0: Level 2 Instr. Details View

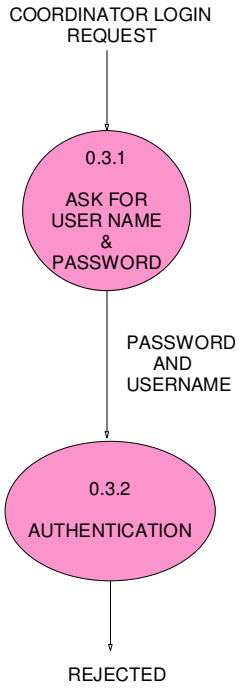


Fig3.1: Level 2 Expansion of Coordinator Login

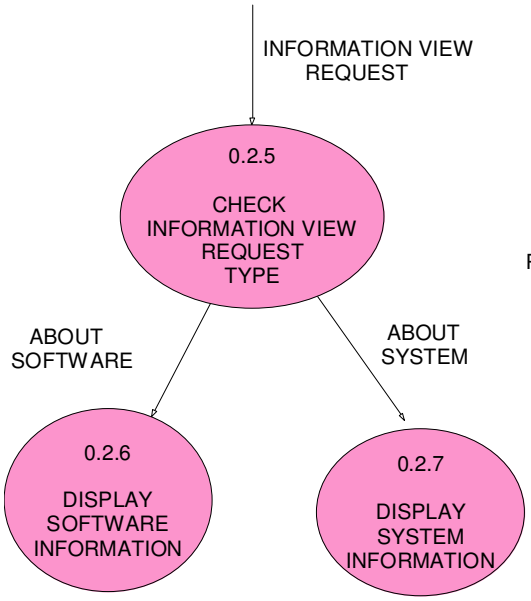


Fig.3.2: Level 2 Expansion of Software Details View

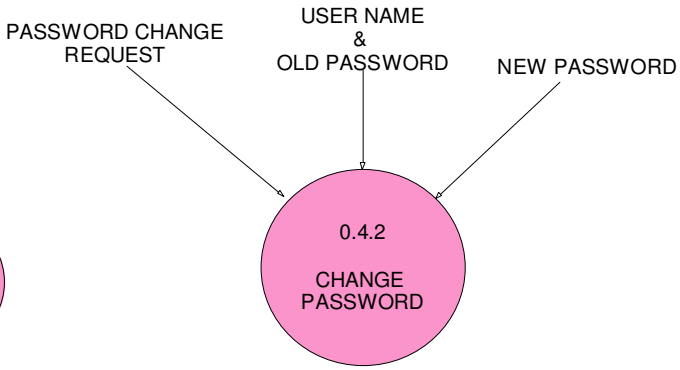


Fig3.3: Level 3 Passowrd Change

LEVEL 2(contd.):

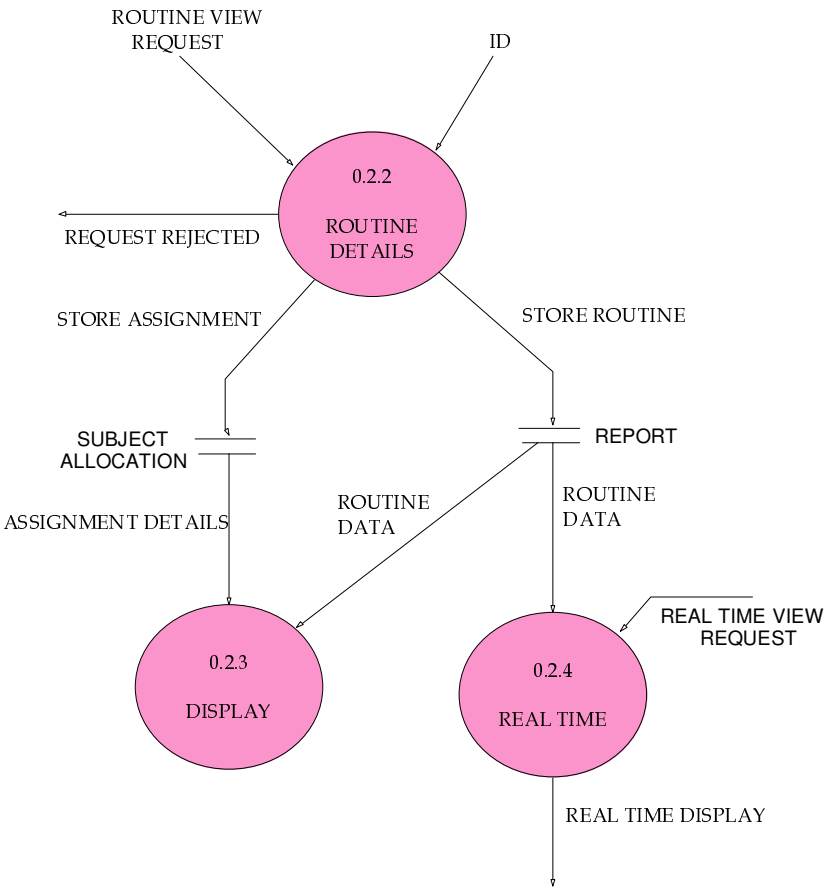


Fig3.4: Level 2 Display Of Routine Schedule

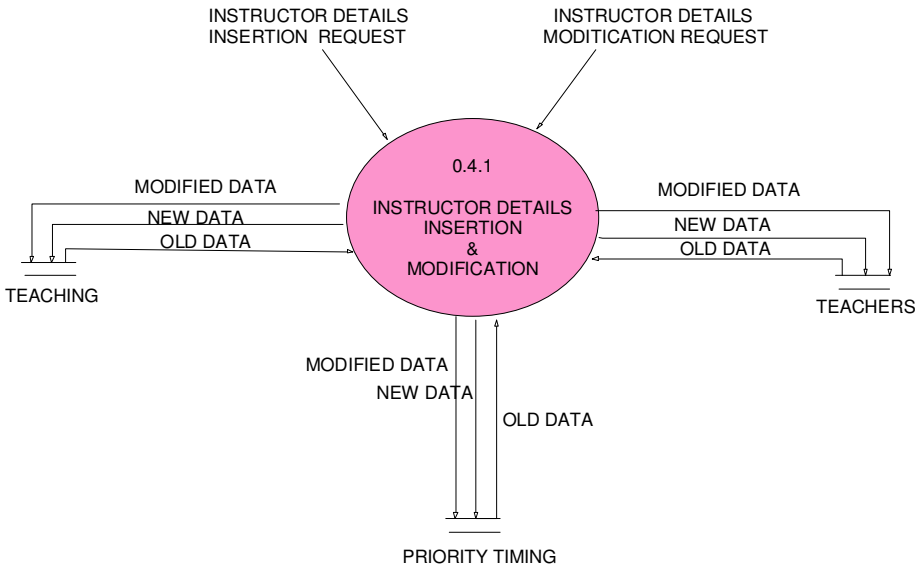


Fig3.5:Expansion of Instructor Details Insertion and Modification

LEVEL 3:

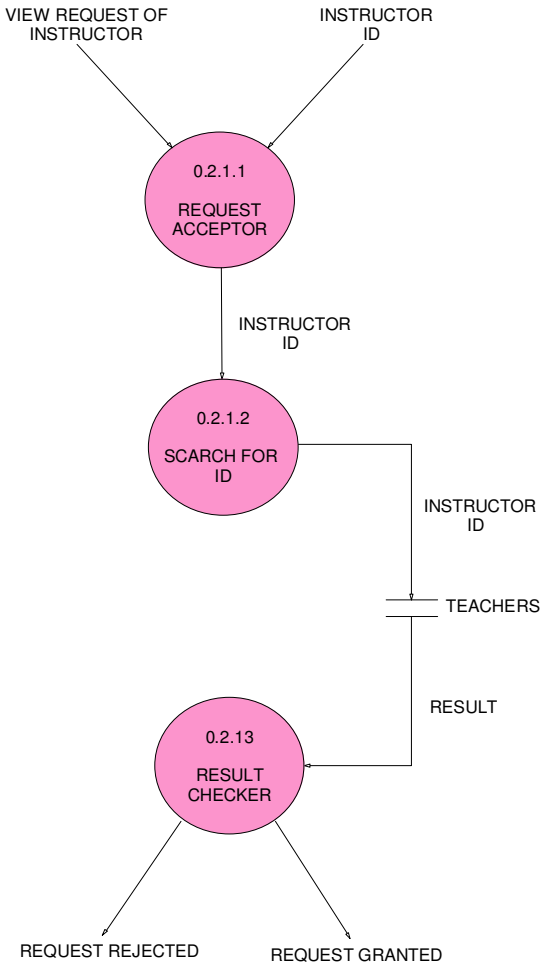


Fig4.0: Level 3 Expansion of View Of Instructtor Details

LEVEL 3(contd):

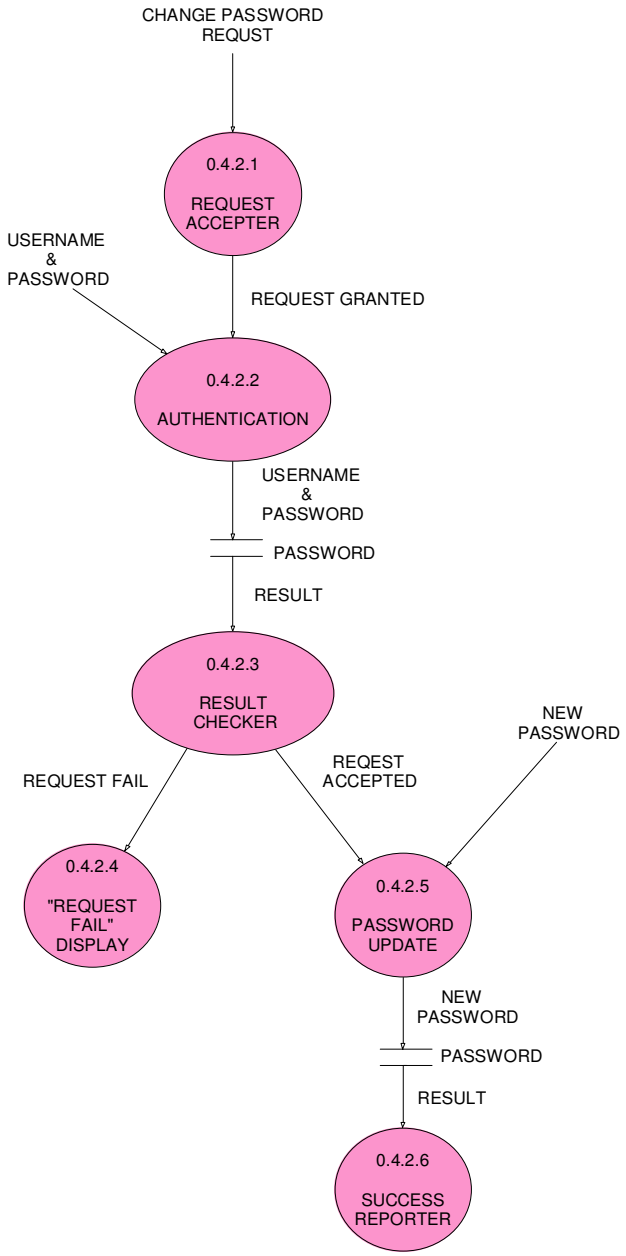
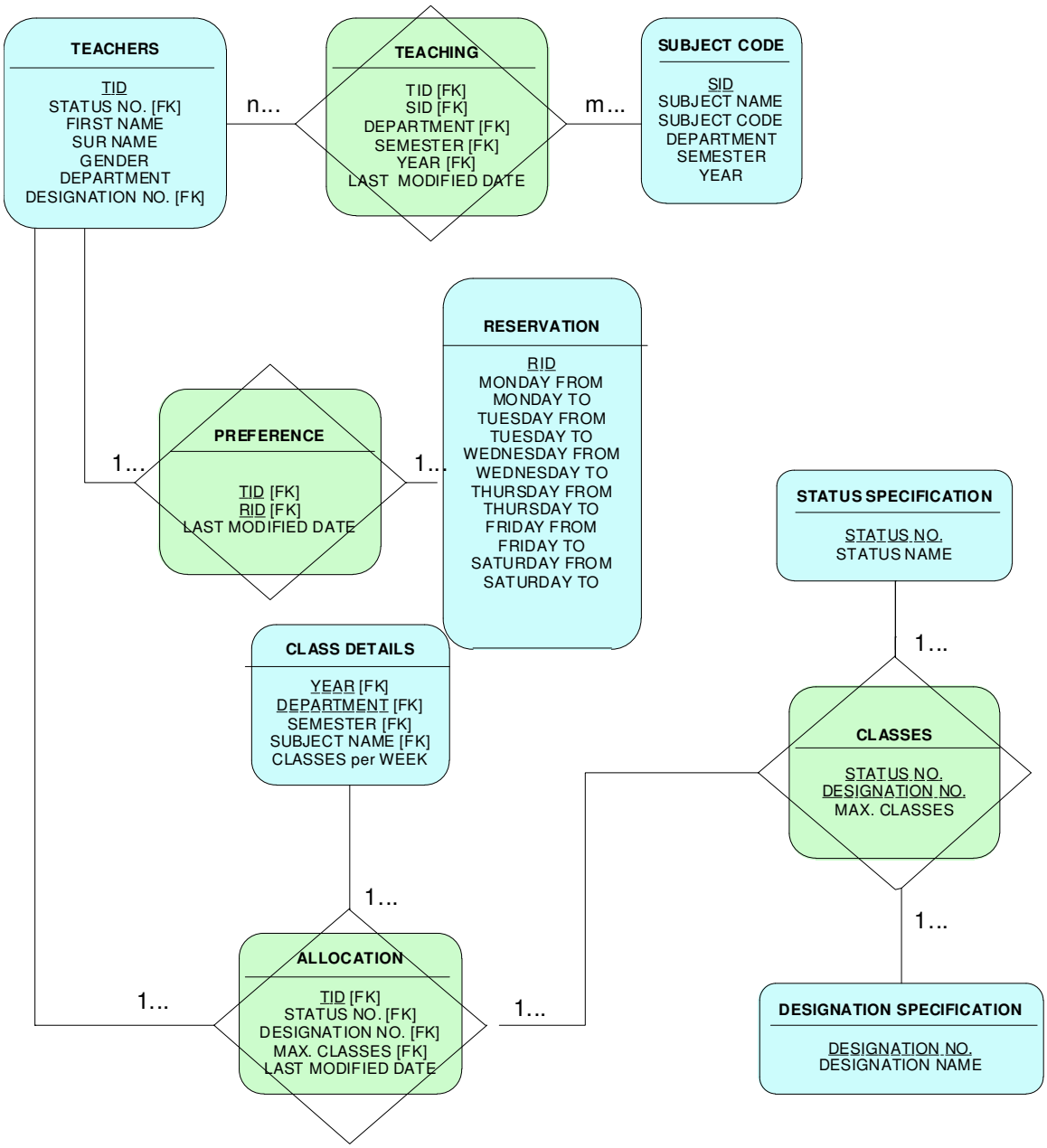


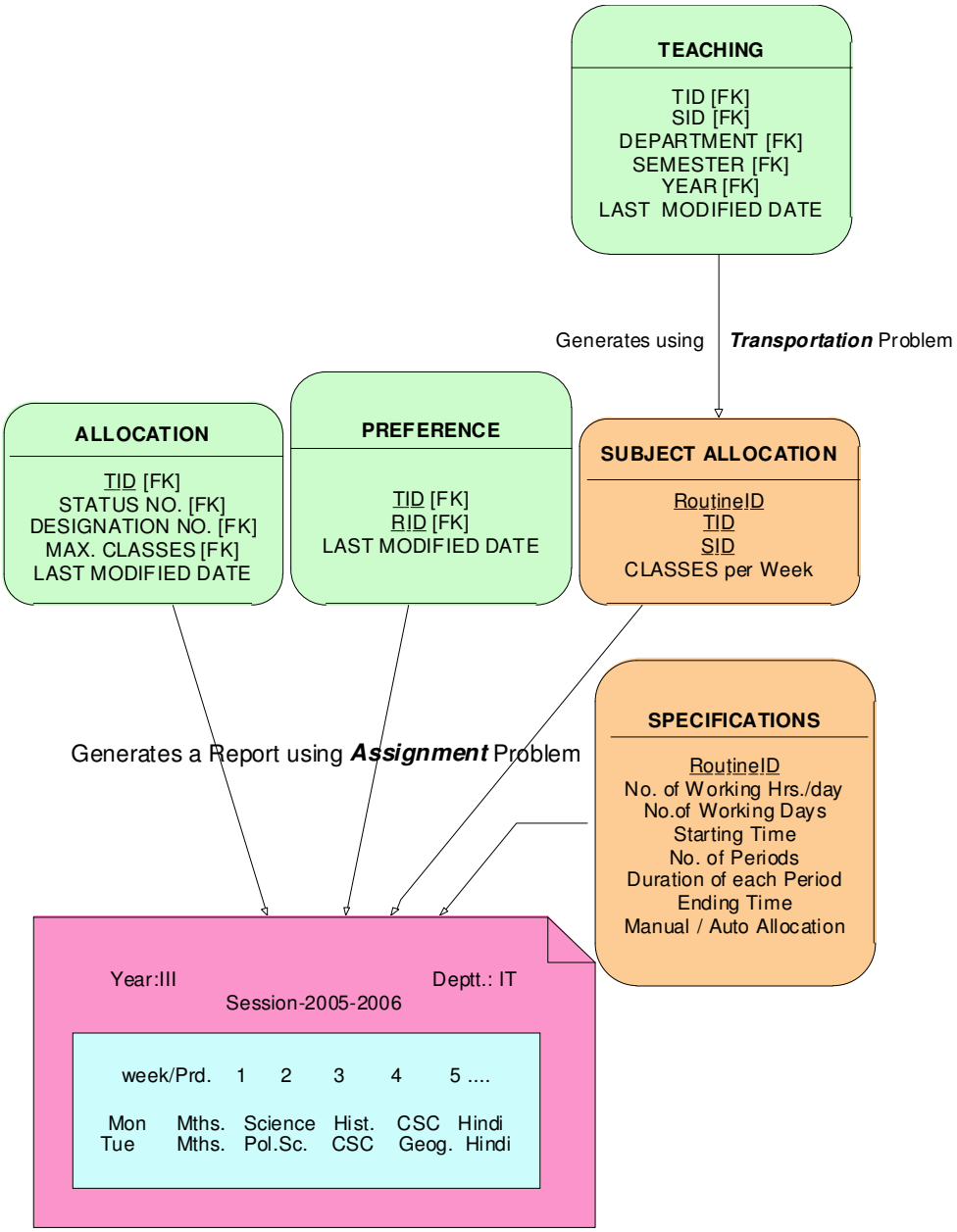
Fig.4.1: Level 3 Expansion of Password Change

ENTITY-RELATIONSHIP DIAGRAM

The different TABLES of the System are hereby depicted using ER Diagram:



In addition to the above tables there are some other tables or Reports which are generated by the help of the abve tables using certain algorithm. They are depicted as follows:



We also have a password file containing the following items for authentication purposes:

THE DATABASE DESIGN DETAILS

TEACHERS

Field name	Data type	Length	Key	Remarks
TID	Number	3	Primary key	Teacher ID of a specific teacher/instructor.
STATUS NUMBER	Number	2		An unique number that resembles a particular status of a teacher.
FIRST NAME	Text	20		First name of a teacher/instructor.
SUR NAME	Text	15		Sur name of a teacher/instructor.
GENDER	Text	10		Gender of a teacher/instructor.
DEPARTMENT	Text	30		Department in which a instructor belongs.
DESIGNATION NUMBER	Number	2		An unique number that resembles a particular designation of a teacher.

TEACHING

Field name	Data type	Length	Key	Remarks
TID	Number	3	Primary key1	Teacher ID of a specific teacher/instructor.
SID	Number	3	Primary key2, Foreign key referencing to table SUBJECT CODE	Subject ID of a specific teacher/instructor.
DEPARTMENT	Text	30		Department(s) in which the instructor teaches.
SEMESTER	Text	10		Current semester of the course.
DATE LAST MODIFIED	Text	10		Date of last modification of the table.

SUBJECT CODE

Field name	Data type	Length	Key	Remarks
SID	Number	3	Primary key	Subject ID of a specific teacher/instructor.
SUBJECT NAME	Text	40		Name of the subject taught.
SUBJECT CODE	Text	3		An unique number that resembles a particular subject.
DEPARTMENT	Text	30		Dapartment(s) in which the subject is taught.
SEMESTER	Text	10		Current semester of the course.

PRIORITY TIMING

Field name	Data type	Length	Key	Remarks
TID	Number	3	Primary key1, Foreign key1 referencing to table TEACHERS	Teacher ID of a specific teacher/instructor.
RID	Number	3	Primary key2, Foreign key2 referencing to table RESERVATION	Reservation ID of a specific teacher/instructor.
DATE LAST MODIFIED	Text	10		Date of last modification of the table.

CLASS DETAILS

Field name	Data type	Length	Key	Remarks
YEAR	Number	1	Primary key1	Year(s) in which the subject is taught.
DEPARTMENT	Text	30	Primary key2	Dapartment(s) in which the subject is taught.
SUBJECT	Text	40		Name of the subject taught.
CLASSES PER WEEK	Number	2		Number of classes of that subject per week.

RESERVATION

Field name	Data type	Length	Key	Remarks
RID	Number	3	Primary key	Reservation ID of a specific teacher/instructor.
MONDAY FROM	Text	5		Preffered starting time on Monday.
MONDAY TO	Text	5		Preffered ending time on Monday.
TUESDAY FROM	Text	5		Preffered starting time on Tuesday.
TUESDAY TO	Text	5		Preffered ending time on Tuesday.
WEDNESDAY FROM	Text	5		Preffered starting time on Wednesday.
WEDNESDAY TO	Text	5		Preffered ending time on Wednesday.
THURSDAY FROM	Text	5		Preffered starting time on Thursday.
THURSDAY TO	Text	5		Preffered ending time on Thursday.
FRIDAY FROM	Text	5		Preffered starting time on Friday.
FRIDAY TO	Text	5		Preffered ending time on Friday.
SATURDAY FROM	Text	5		Preffered starting time on Saturday.
SATURDAY TO	Text	5		Preffered ending time on Saturday.

ALLOCATION

Field name	Data type	Length	Key	Remarks
TID	Number	3	Primary key, Foreign key referencing to table TEACHERS	Teacher ID of a specific teacher/instructor.
STATUS NUMBER	Number	2		An unique number that resembles a particular status of a teacher.
DESIGNATION NUMBER	Number	2		An unique number that resembles a particular designation of a teacher.
MAXIMUM CLASSES	Number	2		Maximum number of classes of that subject per week.
DATE LAST MODIFIED	Text	10		Date of last modification of the table.

STATUS SPECIFICATION

Field name	Data type	Length	Key	Remarks
STATUS NUMBER	Number	2	Primary key	An unique number that resembles a particular status of a teacher.
STATUS	Text	30		Status of the particular teacher.

DESIGNATION SPECIFICATION

Field name	Data type	Length	Key	Remarks
DESIGNATION NUMBER	Number	2	Primary key	An unique number that resembles a particular designation of a teacher.
DESIGNATION	Text	30		Designation of the particular teacher.

MAX CLASSES

Field name	Data type	Length	Key	Remarks
STATUS NUMBER	Number	2	Primary key1, Foreign key1 referencing to table STATUS SPECIFICATION	An unique number that resembles a particular status of a teacher.
DESIGNATION NUMBER	Number	2	Primary key2, Foreign key2 referencing to table STATUS SPECIFICATION	An unique number that resembles a particular designation of a teacher.
MAXIMUM CLASSES	Number	2		Maximum number of classes of that subject per week.

SPECIFICATIONS

Field name	Data type	Length	Key	Remarks
NUMBER OF WORKING HOURS PER DAY	Number	2		Number of working hours per day in the institution.
NUMBER OF WORKING DAYS	Number	1		Number of working days in the institution.
STARTING TIME	Text	10		Starting time of the institution.
NUMBER OF PERIODS	Number	2		Number of periods taken in the institution.
EACH PERIOD INTERVAL	Number	2		Period interval of the institution.
ENDING TIME	Text	10		Ending time of the institution.
RESERVATION INTERVAL	Text	20		Reservation interval of each teacher/instructor.

DESIGN TECHNIQUES
FOR
ROUTINE GENERATOR AND MANAGEMENT SYSTEM

6.1. TECHNIQUE USED FOR SUBJECT ALLOCATION[7]:

6.1.1. Definition:

Corresponding to every teacher there is a subject allocation. This can be done using auto or manual allocation. In manual allocation the coordinator is required to manually feed in data for each teacher. In auto allocation the coordinator is supposed to enter a list(choice) of subjects for each teacher. The system then using a technique (which would be discussed subsequently) feasibly allocates different subjects to teachers.

6.1.2. The Technique:

For a particular subject there may be more than one teacher who would like to teach the subject. The system must be able to distinguish between the different teachers and allocate these teachers to different subjects.

The question is how would the system be able to distinguish and differentiate amongst the teachers and accordingly allocate subjects to them.

From this real-life model if this is changed to a mathematical model implementing the ***Transportation Problem*** the above question can be answered.

Mathematically, the problem can be redesigned as ,

$$\text{Maximize} \qquad Z=\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} X_{ij}$$

Subject to the constraints

$$\sum_{j=1}^{n} X_{ij} = A_i, \quad i=1,2,\ldots..m$$

$$\sum_{i=1}^{n} X_{ij} = B_j, \quad j=1,2,\ldots..n$$

and

$$\sum_{i=1}^{m} A_i = \sum_{j=1}^{n} B_j$$

Here A_i denotes the maximum classes corresponding to every teacher.

---

[7] This technique is used only for Auto allocation. For manual allocation it is a straight forward method.

and, B_j denotes the maximum classes corresponding to every subject.

From the above diagram, the different constraints can be written easily. The sum of the variables of the $i_{th.}$ Row is equal to a_i and the sum of the variable of the jth. Column is equal to bj.

6.2. TECHNIQUE USED FOR ROUTINE GENERATION:

6.2.1. Definition:

The primary aim of the software is to develop a routine for the institute. Corresponding to every teacher/class there is a Routine. A simple mathematical technique is used in order to develop this Routine. The Routine thus developed must be feasible and applicable.

6.2.2. The Technique:

For a particular teacher/class there may be more than one solution i.e. a routine. However the system must be able to distinguish between these solutions and come up with the most appropriate one.

The question is how would the system be able to distinguish and differentiate amongst the different solutions and how at all come up with the initial basic feasible solution.

This is done implementing a mathematical model, i.e. the ***Assignment Problem*** Algorithm.

Assignment Problem is a particular type of Transportation Problem where *n* origins are to be assigned to equal number of destinations on a *one to one basis*[8].

Mathematically, the problem can be redesigned as ,

Let Xij be a variable defined by,
Xij = 1 if ith. (teacher, subject) pair is assigned to jth. (yr, deptt., day, prd.)
Xij = 0 if ith. (teacher, subject) pair is not assigned to jth. (yr, deptt., day, prd.)

Hence,

$$\text{Optimize} \qquad Z=\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} X_{ij}$$

Subject to the constraints

[8] For every pair of TID,SID (i.e. Teacher teaching a particular subject) at a particular time can be assigned to one place only. For this reason and assignment problem being a one-to-one mapping problem, this technique is implemented.

$$\sum_{j=1}^{n} X_{ij} = A_i = 1, \quad i=1,2,\ldots..n$$

$$\sum_{i=1}^{n} X_{ij} = B_j = 1, \quad j=1,2,\ldots..n$$

and $$\sum_{i=1}^{m} A_i = \sum_{j=1}^{n} B_j = n$$

Here A_i denotes the teacher, subject pair allocation
and, B_j denotes the allocation for a particular quadrate (yr, deptt., day, prd.)

ALGORITHM
FOR
ROUTINE GENERATOR AND MANAGEMENT SYSTEM

7.1. Procedure Subject Allocation():

	S1	*S2*	*S3*	*S4*	*S5*	
T1	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	a_1
T2	....	....	....	....	....	a_2
T3						a_3
T4			C_{43}			a_4
T5				C_{54}	C_{55}	a_5
	b_1	b_2	b_3	b_4	b_5	

T1	0	10	10	10	*12*
T2	0	0	20	0	*10*
T3	0	30	0	30	*14*
T4	40	0	40	0	*4*
	6	8	15	11	

This table shows an example of how T1,T2... have different status & designation .And they teach different subjects. The subjects that they don not teach are indicated by 0.

Table 1.1

Step 1. Retrieve information from Teaching , Class Details and Allocation tables of the database.

Step 2. Form the Transportation Table with rows indicating the teachers and columns indicating the subjects.

Step 3. Insert the Teacher requirements, a_i i.e. Maximum classes from Allocation table and Subject requirements, b_j i.e. Classes per Week from Class Details table.

Step 4. Determine the cost-matrices C_{ij} as per gradation of Status and Designation no. of individual teachers.

Step 5. Construct this transportation problem as per table 1.1.

Step 6. Determine an initial basic feasible solution by *Voggel's Approximation Method*(VAM).

Step 7. For all the basic variable x_{ij}, solve the system of equations

$u_i+v_j=c_{ij}$ for all I,j for which (I,j) is in the basis

starting initially with some $u_1=0$.

Step 8. Compute the net evaluations $z_{ij}-c_{ij}=u_i+v_j-c_{ij}$, for all non-basic cells.

Step 9. Examine the sign of each z_{ij}-c_{ij},If all z_{ij}-$c_{ij} \leq 0$, then the current basic feasible solution is an optimum one. If atleast one z_{ij}-$c_{ij} > 0$, select the variable x_r, having the largest positive net evaluation to enter the basis.

Step 10. Let the variable x_r, enter the basis. Allocate an unknown quantity say θ, to the cell (r,s). Identify a loop[9] that starts and ends at the cell(r,s) and connects some of the basic cells. Add and subtract interchangeably, θ to and from the transition cells of the loop in such a way that the rim requirements remain satisfied.

Step 11. Assign a maximum value to θ in such a way that the value of one basic variable becomes zero and the other basic variables remain non-negative. The basic cell whose allocation has been reduced to zero, leaves the basis.

Step 12. Return to *step 7* and repeat the process until an optimum basic feasible solution has been obtained.

Step 13. Hence we get the actual allocation for the teachers along with the number of classes for a particular teacher. Store the information form the transportation table to the database table named *Subject Allocation*.

[9] Identification of loop is done by using Floyd's Algorithm.

7.2. Procedure Routine Generation():

TID	SID	Yr1 Dept1 Day1 Prd1	Yr1 Dept1 Day1 Prd2	Yr1 Dept1 .. Day1 Prd3..	Yr1 Dept1….Day2 Prd1…	Yr1 Dept2…. Day1 Prd1….	Yr2 Dept1….Day1 Prd1….	Yr4 Dept1 Day1 Prd1	
1	12	0	0	100	0	100	100	0	
1	12	0	100	100	0	100	100	0	
1	12	100	0	0	100	0	0	100	
2	1	0	0	0	0	100	0	0	
…	…	…	…	…	…	…	…	…	

Corresponding to TID 1 and SID 12 there are 3 classes.
Hence there are 3 rows of TID 1 and SID 12

SUBJECT ALLOCATION

RoutineID	TID	SID	Classes
1	1	12	3
1	2	1	2
1	2	2	5
1	3	2	4

Table 1.2

An illustration:

The colmns. are to be arranged in the following order starting from colmn. 1

Yr.	1	1	1	1	1 ...	1 ...	2	4
Dept.	CSE	CSE	CSE ...	CSE	CSE ...	IT ...	CSE	EE
Day	MON	MON	MON...	MON	TUE ...	MON...	MON	SAT
Perd.	1	2	3	7	1 ...	1 ...	1	7
Colmn.	2	3	4 …..	8	9	.....		

Now calculating the starting colm. no for each dept. as per year as per day as per prd.
Say Year=yr,Dept=dpt,day=dy,Period=pd then the formula is
1<=yr<=4 1<=dpt<=4 1<=dy<=6 1<=pd<=7
(yr,dpt,dy,pd)=7*6*4*(yr-1)+7*6*(dpt-1)+7*(dy-1)+pd+1

Step 1. Retrieve information from Class Details, Specifications and Subject Allocation tables of the database.

Step 2. Form the Assignment Table with rows indicating the teachers alongwith subjects that are to be taught and columns indicating year, department, days and periods as per Table 1.2 indicates.

Step 3. Determine the classes(year,deptt.) where each teacher would teach. Also determine on which days,periods are they available and accordingly construct the Assignment Table.(*Please refer illustration*)

Step 4. Modify the cost matrix by subtracting the smallest element in each ow from all the elements in that row. Modify the resulting matrix by repeating this procedure on its columns. The reduced matrix will then have non-negative elements with atleast one zero in each row and each column.

Step 5. In the modified matrix, search for an optimal solution as follows:

(i) Count the number of zeroes of each row and store this value in the last column.

(ii) Repeat this step for columns also.

(iii) For each corresponding row check with column whether the no. of zeroes of the row = no. of zeroes of columns. If yes, then indicate that (row,column) with a parity bit set to 1.

(iv) Repeat step (iii) for all columns also till no. of parity bits=no. of Rows.(whichever earlier)

Step 6. If no. of parity bits=no. of rows then optimal solution is reached so go to *Step 9*.

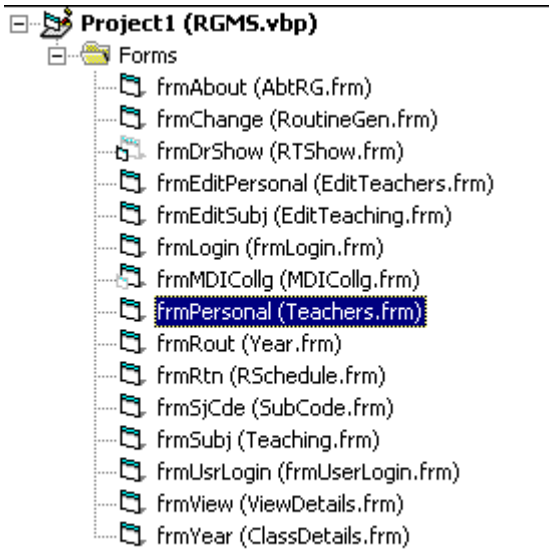
Step 7. Find a row and column whose difference between no. of zeroes is minimum and such that the tuple(row,column) ≠ 0.

Step 8. Go to *Step 7* and repeat the procedure till an optimal solution is reached.

Step 9. Store the information of the optimal solution from the table to the database table.

THE SAMPLE CODE

The entire system is subdivided into several modules. They are:



8.1.The Real Time Display Function:

```
Private Sub Timer1_Timer()
Dim bn As String
Dim Pf As Variant
Dim tFile As String
Dim sFile As String
Dim nC As String
Dim FSys As New FileSystemObject
Dim ASys As New FileSystemObject
Dim OnStrm As TextStream
Dim InStrm As TextStream
Dim cnum As Integer, dt As Date
Dim number As Integer
Pf = Array("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
number = 0
'Text1.Text = Time()
'Text2.Text = Date

'CALCULATING THE DAY OF THE WEEK
dt = Date
cnum = Abs(#1/1/2006# - dt)
j = cnum / 365
If (cnum / 365 > 0.5 And cnum / 365 < 5) Then
cnum = (cnum + cnum / 365 - 1) Mod 365
Else
cnum = (cnum + cnum / 365) Mod 365
End If
cnum = cnum Mod 7
```

```
If (dt < #1/1/2006#) Then
   cnum = 6 - cnum + 1
   cnum = cnum Mod 7
End If

Dy = cnum

' CALCULATING THE PERIOD
Prd = 0
adoTime.Recordset.MoveFirst
While Not adoTime.Recordset.EOF
If Time() >= adoTime.Recordset(1) And Time() <= adoTime.Recordset(2) Then
   Label8.Caption = adoTime.Recordset(1)
   Label10.Caption = adoTime.Recordset(2)
   Prd = adoTime.Recordset(0)
   GoTo frd
End If
adoTime.Recordset.MoveNext
Wend

frd:

Label4.Caption = Pf(Dy - 1)
If (Prd <> 0) Then
Label6.Caption = Prd
Else
Label6.Caption = " "
End If

'Text2.Text = Dy
'Text1.Text = Prd


tFile = "ARout.txt"
sFile = "Ru.txt"
Set InStrm = FSys.OpenTextFile(tFile, 1, False, False)
Set OnStrm = ASys.OpenTextFile(sFile, 1, False, False)

Do While OnStrm.AtEndOfStream = False
OnStrm.Skip (1)
nC = OnStrm.Read(3)
cnum = nC
Tchr = cnum

'If Tchr = 1 Then
OnStrm.Skip (11)
nC = OnStrm.Read(3)
cnum = nC
Subj = cnum

OnStrm.Skip (11)
nC = OnStrm.Read(3)
cnum = nC
j = cnum
```

```
OnStrm.Skip (10)
nC = OnStrm.Read(3)
cnum = nC
Clr = cnum

Set InStrm = FSys.OpenTextFile(tFile, 1, False, False)
Do While InStrm.AtEndOfStream = False

    InStrm.Skip (1)
    nC = InStrm.Read(3)
    cnum = nC
    tt = cnum

    InStrm.Skip (11)
    nC = InStrm.Read(1)
    cnum = nC
    zp = cnum

    InStrm.Skip (13)
    nC = InStrm.Read(1)
    cnum = nC
    Temp = cnum

    InStrm.Skip (13)
    nC = InStrm.Read(1)
    cnum = nC
    vR = cnum

    InStrm.Skip (13)
    nC = InStrm.Read(1)
    cnum = nC
    bn = cnum
    vC = cnum
   If tt = j And zp = Prd And Temp = Dy Then
   number = number + 1
   Select Case vR
   Case 1: nC = "CSE"
   Case 2: nC = "IT"
   Case 3: nC = "ECE"
   Case 4: nC = "EE"
   End Select

   nC = nC + "    " + bn

   Select Case vC
   Case 1: nC = nC + "st.Yr"
   Case 2: nC = nC + "nd.Yr"
   Case 3: nC = nC + "rd.Yr."
   Case 4: nC = nC + "th.Yr"
   End Select

   adoSubjCode.Recordset.MoveFirst
   YrdeptOrSbj(number, 1) = nC ' which yr and dept.
```

```
    While Not adoSubjCode.Recordset.EOF
        If Tchr = adoSubjCode.Recordset(0) Then
            nC = adoSubjCode.Recordset(3) + " " + adoSubjCode.Recordset(4)
           'adoTTable.Recordset(zp) = nC
            SbjOrTeachr(number, 2) = nC
            adoSubjCode.Recordset.MoveLast
        End If
    adoSubjCode.Recordset.MoveNext
    Wend
    End If
    'End If
    InStrm.SkipLine
Loop
OnStrm.SkipLine
Loop

Text4.Text = YrdeptOrSbj(1, 1)
Text3.Text = SbjOrTeachr(1, 2)
Text6.Text = YrdeptOrSbj(2, 1)
Text5.Text = SbjOrTeachr(2, 2)
Text8.Text = YrdeptOrSbj(3, 1)
Text7.Text = SbjOrTeachr(3, 2)
Text10.Text = YrdeptOrSbj(4, 1)
Text9.Text = SbjOrTeachr(4, 2)
Text12.Text = YrdeptOrSbj(5, 1)
Text11.Text = SbjOrTeachr(5, 2)

End Sub
```

8.2.The Routine Generation Procedure:

```
For i = 1 To maxR
   Temp = 0
   vR = 1
   vC = 1
      adoClass.Recordset.MoveFirst
      For tt = 1 To 6
      adoClass.Recordset.MoveFirst
      Strt(tt) = -1
      Stp(tt) = -1
         While Not adoClass.Recordset.EOF
            If (Routne(i, 0) = adoClass.Recordset(0)) Then
               If (Routne(i, 0) = 7) Then
               ii = 1
               End If
               If (adoClass.Recordset((tt - 1) * 2 + 1) = " ") Then
               GoTo outLoop
               End If
               For e = 0 To 15 Step 2
                  If (adoClass.Recordset((tt - 1) * 2 + 1) = StCs(e)) Then
                     For optimal = 0 To 13 Step 2
                        If (StCs(e + 1) = Cls(optimal)) Then
                           For f = 0 To 15 Step 2
                              If (adoClass.Recordset(2 * tt) = EdCs(f)) Then
                                 For g = 1 To 14 Step 2
                                    If (EdCs(f + 1) = Cls(g)) Then
                                    Strt(tt) = (optimal + 2) \ 2
                                    Stp(tt) = (g + 1) \ 2
                                    GoTo outLoop
                                    End If
                                 Next
                              End If
                           Next
                        End If
                     Next
                  End If
               Next
               End If
         adoClass.Recordset.MoveNext
         Wend
outLoop:
   Next

      adoResv.Recordset.MoveFirst ' Class Details
      adoSj.Recordset.MoveFirst 'SubjectAllocation
      adoAbc.Recordset.MoveFirst 'SubjectCode

   While Not adoSj.Recordset.EOF
      If Routne(i, 0) = adoSj.Recordset(1) And adoSj.Recordset(0) = RtnId Then
            adoAbc.Recordset.MoveFirst
               While Not adoAbc.Recordset.EOF
               If Routne(i, 1) = adoAbc.Recordset(0) Then
                  adoResv.Recordset.MoveFirst
```

```
                While Not adoResv.Recordset.EOF
                   clYr = -1
                   clDt = -1
                         If adoAbc.Recordset(1) = adoResv.Recordset(3) Then
                            clYr = adoResv.Recordset(0)
                               If (adoResv.Recordset(1) = "CSE") Then
                                  clDt = 1
                               End If
                               If (adoResv.Recordset(1) = "IT") Then
                                  clDt = 2
                               End If
                               If (adoResv.Recordset(1) = "ECE") Then
                                  clDt = 3
                               End If
                               If (adoResv.Recordset(1) = "EE") Then
                                  clDt = 4
                               End If
                            GoTo Begin
                         End If
                adoResv.Recordset.MoveNext
                Wend
             End If
          adoAbc.Recordset.MoveNext
          Wend
      End If
   adoSj.Recordset.MoveNext
   Wend
Begin:
   For j = 2 To maxC
   ' Here zp is the period
      zp = (j - 1) Mod (7)
      If (zp = 0) Then
      zp = 7
      End If
      If zp = 1 Then
      Temp = Temp + 1
      If (Temp = 7) Then
         Temp = 1
         vR = vR + 1
         If (vR = 5) Then
         vR = 1
         vC = vC + 1
         End If
      End If
      End If
      If (i = 1) Then
      Print #1, j, zp, Temp, vR, vC
      Day(j) = Temp
      End If
                'Here the actual insertion will be done to the aray Routne[]
                     ' Here zp is the period
                     ' Here Temp is the day
                     ' vR is the deptt.
                     ' vC is the year
```

```
                    If Strt(Temp) <> -1 Or Stp(Temp) <> -1 Then
                       If (Strt(Temp) <= zp And Stp(Temp) >= zp) Then
                          If (vR = clDt And vC = clYr) Then
                             Routne(i, j) = 20
                             GoTo Last
                          End If
                       End If
                    End If
                    If (vR = clDt And vC = clYr) Then
                      Routne(i, j) = -10
                    End If
Last: Next
Close #1
Next
```

HARDWARE & SOFTWARE SPECIFICATIONS

9.1: System Hardware:

- Pentium II or above processor
- 64 MB RAM or above
- 20 GB Hard Disk or above

9.2: Software Support:

- Windows 98 or above Operating System
- Microsoft Visual Studio 6.0 or above
- Microsoft Office 98 or above

TEST RESULTS

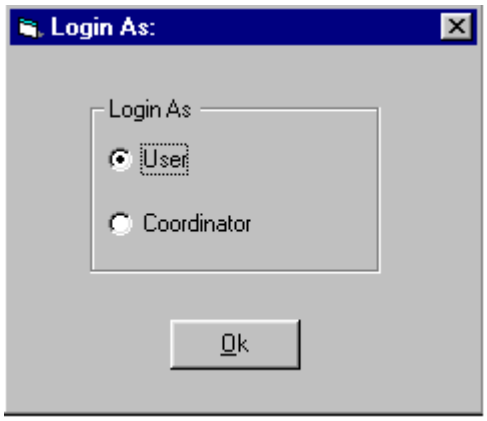
Routine Generator was tested with more than 16 different people across 20 different su bjects. RGMS checked for erroneous inputs and showed appropriate warning messages.

RGMS has been completely tested under Microsoft Windows 98 Operating System. The system configuration used for testing the software is given below:

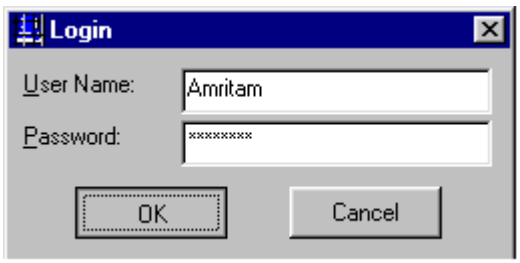
- Processor: AMD Sempro n™ 2200+
- Clock Speed: 1.50 GHz
- RAM: 192 MB
- Operating System Windows 98.

Under these test conditions, the RGMS was able to deliver desired results to the user with rich graphics display.

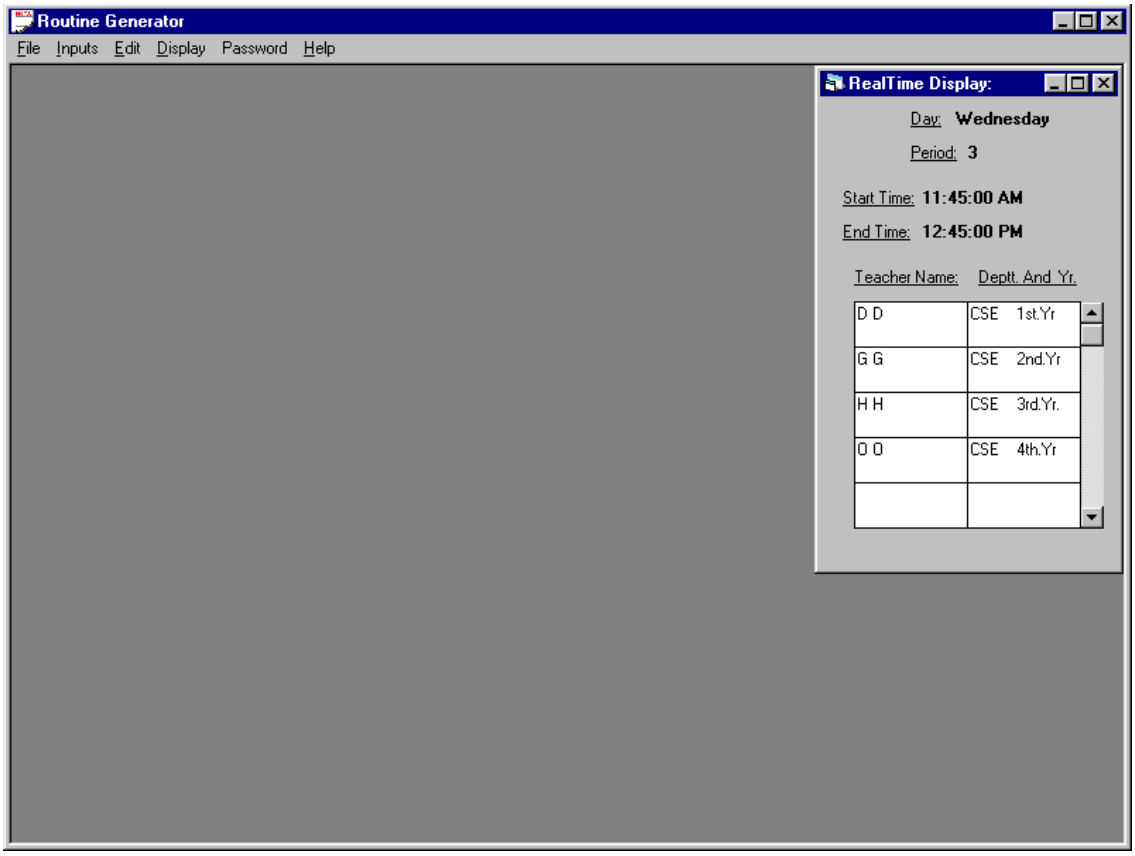
Some of the Screen Shots of RGMS are shown below:



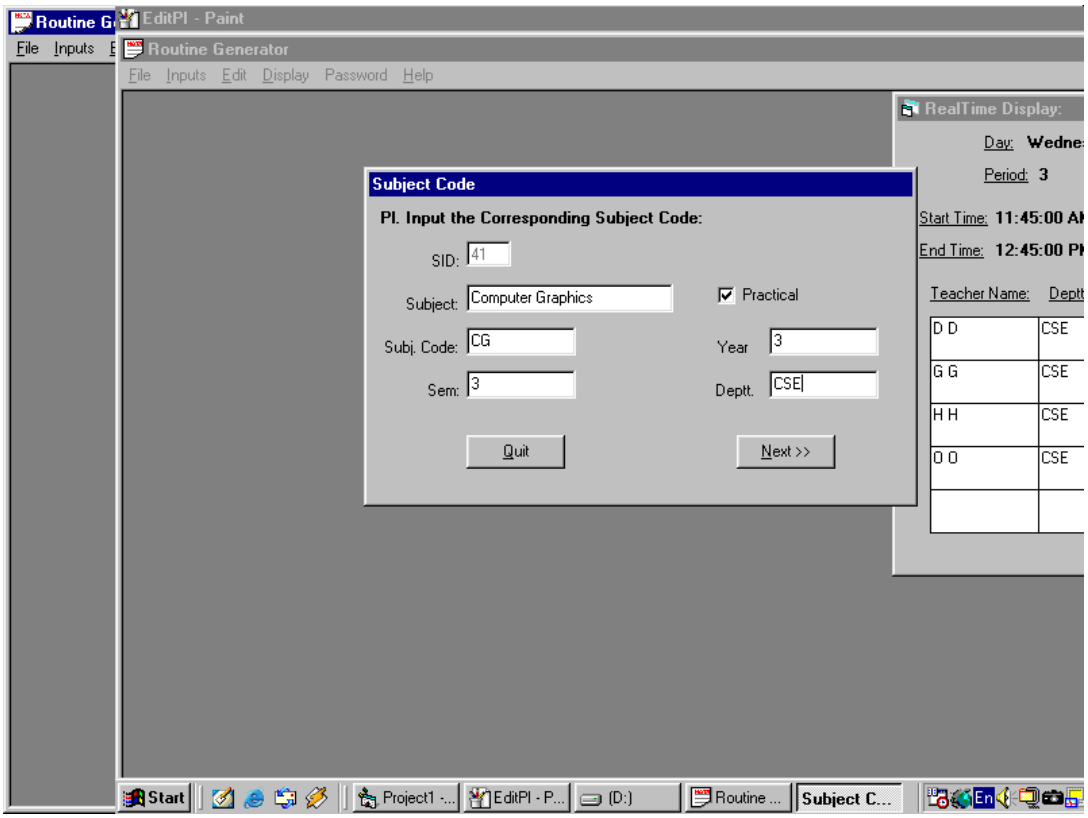
Login As Screen



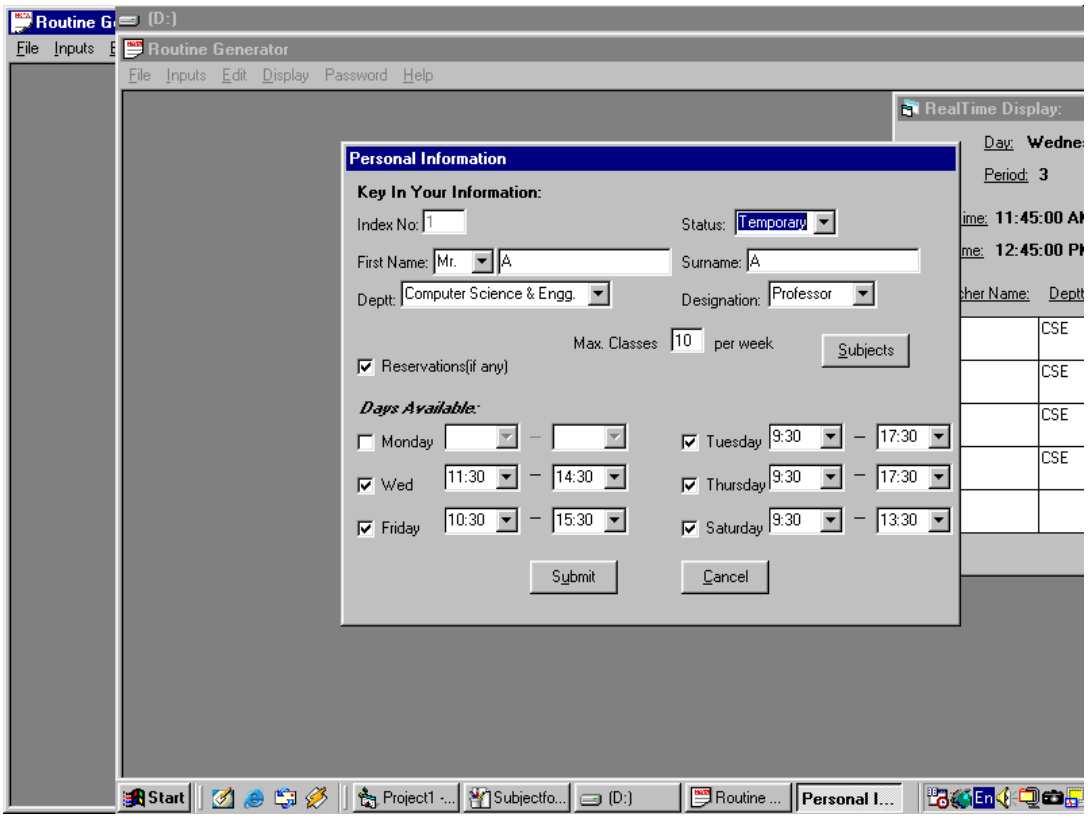
Login As a Coordinator



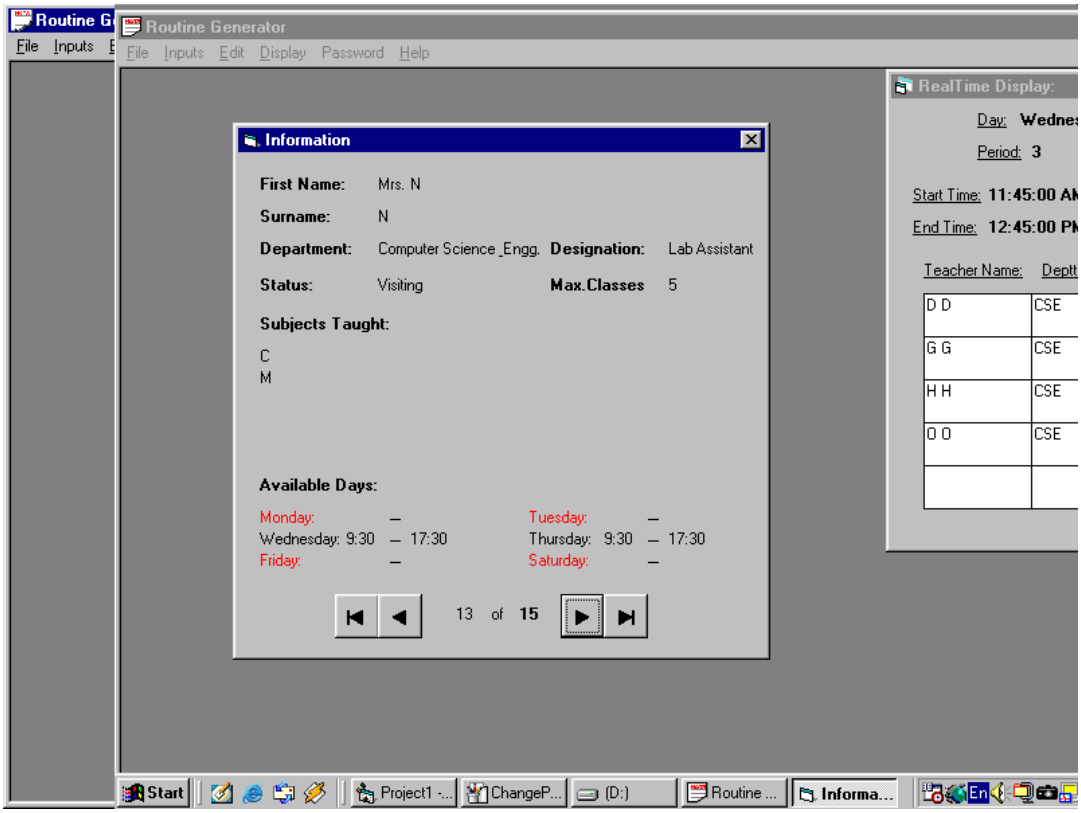
The Main Menu of the RGMS ….. The smaller window shows the Real time Display i.e. Allotment of Teacher at 'that' particular time.



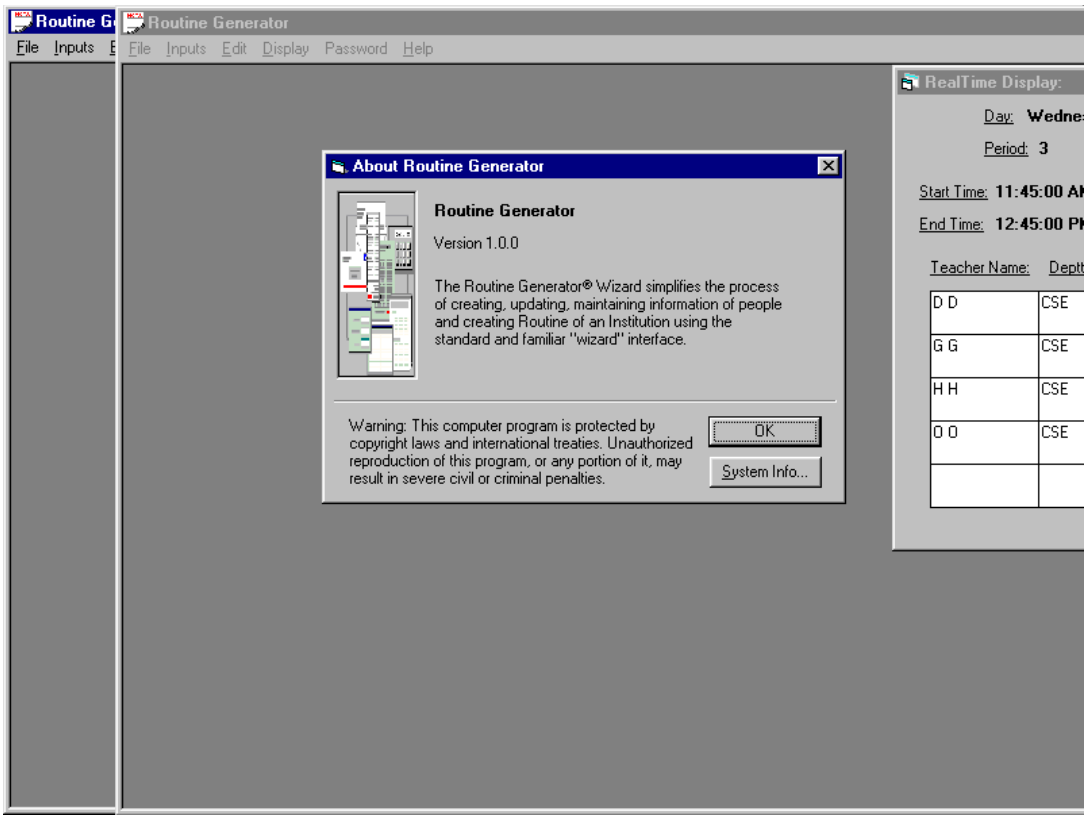
Entry Form for Subject Information



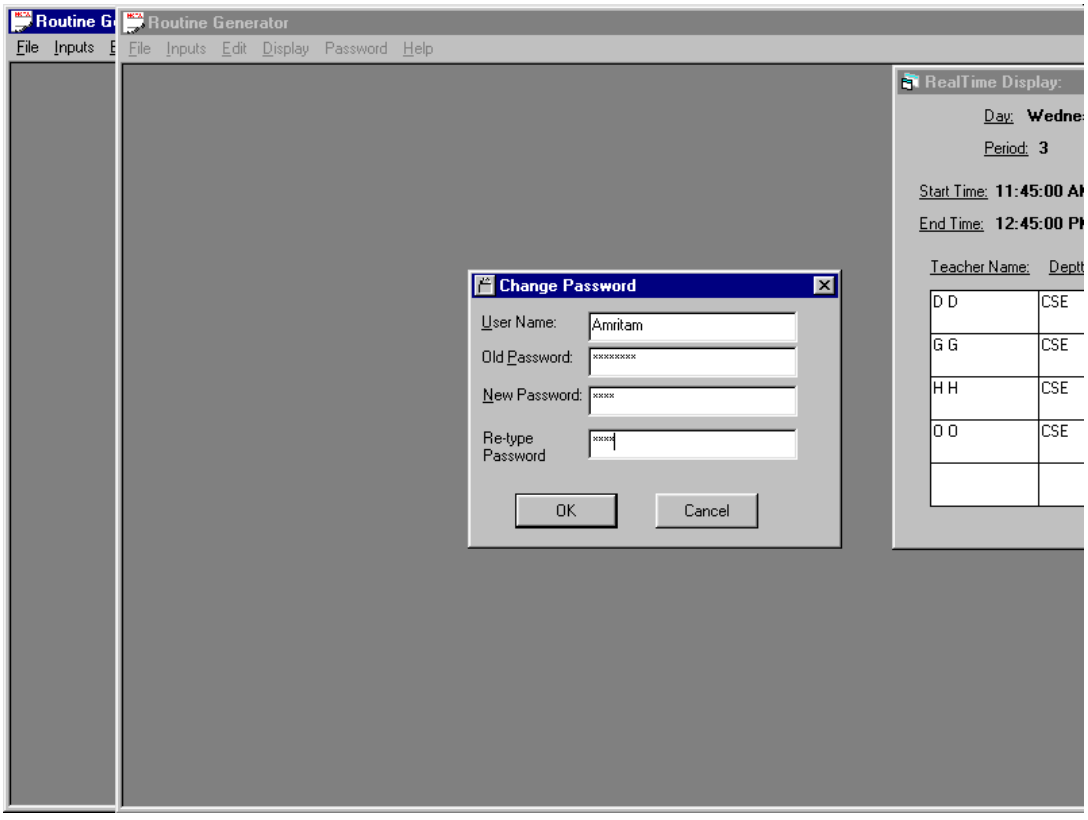
Form for entering Personal Information of Teachers.



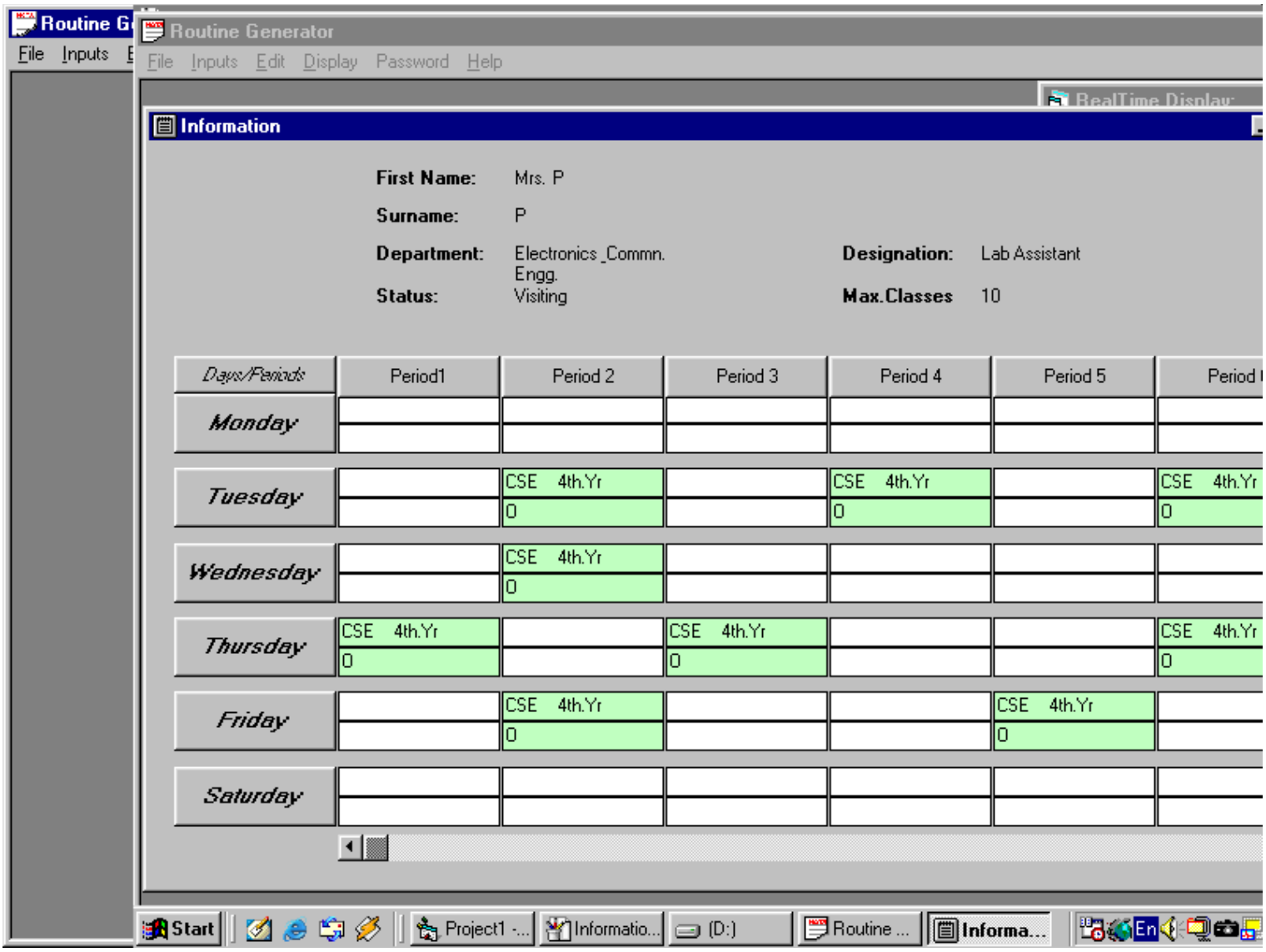
Displaying the Records of the Teachers.



About the Software…



Changing the Password…



The actual Routine Generated Form .

N.B.
Green- Signifies STRICT allocation
Yellow- Signifies PRACTICAL
Orange- Signifies WEAK allocation

CONCLUSION

The project RGMS has been developed usi ng VB 6.0 and MSAccess. Vb was used for front end , while MSAccess was used in the backend for database operations.

Though the project is technically sound, on a creative fron there are lots to improve. Obviously, the searching algorithm and allocation a lgorithm may need to be improved. But, given the time frame, we are satisfied with the outcome of the project.