STUDENTS ATTENDANCE MANAGEMENT SYSTEM

MINI PROJECT REPORT

Submitted by

KALAISANKARAN B
Roll No: 11MCA020

in partial fulfillment of the requirements
For the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

Dr. MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY
POLLACHI - 642 003
(Approved by AICTE, Affiliated to Anna University and Accredited by NBA & NAAC with 'A' Grade)

MAY 2013

Dr. MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY
POLLACHI - 642 003

Department of Computer Applications

MINI PROJECT WORK

MAY 2013

This is to certify that the project entitled

STUDENTS ATTENDANCE MANAGEMENT SYSTEM

is the bonafide record of mini project work done by

KALAISANKARAN B
Roll No: 11MCA020

of Master of Computer Applications during the year 2012-2013.

----------	----------
Project Guide	Head of the Department
Mr.R.MUTHUSAMI MCA.,M.Phil.	**DR.V.S.MEENAKSHI MCA.,M.Phil.,Ph.D**
Asst.Prof(SS)/CA	Assoc.Prof & Head/CA

Submitted for the Mini Project Viva-Voce examination held on_____

----------	----------
Internal Examiner	External Examiner

DECLARATION

I affirm that the mini project work titled **"STUDENT ATTENDANCE MANAGEMENT SYSTEM"** being submitted in partial fulfillment for the award of **Master of Computer Applications** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

(Signature of the Candidate)

KALAISANAKARAN B,

Roll No: **11MCA020**

ACKNOWLDGEMENT

I express my gratitude to **Prof. C. RAMASWAMY, M.E., F.I.V., Secretary,** NIA Educational Institutions, Pollachi, for having provided me the facilities to do the project successfully.

My heartful thanks to **Dr. S. VIJAYARANGAN, M.E, F.I.E., Ph.D., Director(Academic),** Dr. Mahalingam College of Engineering and Technology, Pollachi, for his patronage and giving us an opportunity to undertake this Project.

My sincere thanks to **Dr. S. CHENTHUR PANDIAN, B.Sc., LLB., ME., P.hd., MISTE., FIE., MIEEE., (USA), Principal** , who has allowed me to do this project and encouragement given to me.

I owe deep sense of gratitude to **Dr.V.S.MEENAKSHI MCA.,M.Phil.,Ph.D,** Head of Department of Computer Applications for appreciating my goal. I express my sincere thanks to her for her constant encouragement.

I would also like to thank my Project Co-ordinator **Mr. R. MUTHUSAMI, M.C.A., M.Phil.,** Assistant Professor(Senior Scale),Department of Computer Applications for his valuable guidance and support to meet the successful completion of my project.

I express my sincere thanks to my mini project guide **Mr. R. MUTHUSAMI M.C.A., M.Phil.,** Assistant Professor(Senior Scale), and staff members of Department of Computer Applications for their encouragement and valuable guidance throughout this project.

Last but not the least; I extend my sincere thanks to my family members and my friends for their constant support throughout this project.

TABLE OF CONTENTS

ABSTRACT

Student attendance management system deals with the maintenance of the student's attendance details. It is generates the attendance of the student on basis of presence in class. It is maintained on the daily basis of their attendance. the staffs will be provided with the separate username & password to make the student's status.

The staffs handling the particular subjects responsible to make the attendance for all students. Only if the student present on that particular period, the attendance will be calculated. The students attendance reports based on weekly and consolidate will be generated.

LIST OF FIGURES

5.4.5 LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE:

"Attendance Management System" is software developed for maintaining the attendance of the student on the daily basis in the collage. Here the staffs, who are handling the subjects, will be responsible to mark the attendance of the students. Each staff will be given with a separate username and password based on the subject they handle. An accurate report based on the student attendance is generated here. This system will also help in evaluating attendance eligibility criteria of a student. Report of the student's attendance on weekly and monthly basis is generated.

CHAPTER 2

SYSTEM ANALYSIS

2.1 INTRODUCTION

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skillful wall. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation.

The various tasks in the system analysis include the following.

- Understanding application.
- Planning.
- Scheduling.
- Developing candidate solution.
- Performing trade studies.
- Performing cost benefit analysis.
- Recommending alternative solutions.
- Selling of the system.
- Supervising, installing and maintaining the system.

This system manages to the analysis of the report creation and develops manual entry of the student attendance. First design the students entry form , staff allocation and time table allocation forms. This project will helps the attendance system for the department calculate percentage and reports for eligibility criteria of examination .The application attendance entry system will provide flexible report for all students.

2.2 EXISTING SYSTEM

The Existing system is a manual entry for the students. Here the attendance will be carried out in the hand written registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers.

This application requires correct feed on input into the respective field. Suppose the wrong inputs are entered, the application resist to work. so the user find it difficult to use.

2.3 PROPOSED SYSTEM:

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and saving time to generate accurate results from the student's attendance. The system provides with the best user interface. The efficient reports can be generated by using this proposed system.

2.3.1 Advantages of Proposed System

- It is trouble-free to use.
- It is a relatively fast approach to enter attendance
- Is highly reliable, approximate result from user
- Best user Interface
- Efficient reports

3. FEASIBILITY STUDY:

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look lime. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal.

Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

3.1 Economically Feasibility:

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision.

It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

3.2 Technical feasibility:

The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it.

Install all upgrades framework into the .Net package supported widows based application. this application depends on Microsoft office and intranet service ,database. Enter their attendance and generate report to excel sheet.

3.3 Operational Feasibility:

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENTS (Minimum Requirement)

- **Minimum RAM:**-1GB
- **Hard Disk:-**128 GB
- **Processor:-**Intel Pentium 4(1.50 GHZ) or above

3.2SOFTWARE REQUIREMENTS (minimum Requirement)

- **Operating system :**Windows XP
- **Front_Design:**VB.Net version 10.0 ,.NET framework 4.0
- **Front-End Language :**Visual basic
- **Back-End :** Oracle 10g
- **Back-End Connectivity:**ADO.net

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 PACKAGE - VISUAL STUDIO 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft.It is used to develop console and graphical user interface applications along with Windows Forms or WPF applications, web sites, web applications, and web services in both native codetogether with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silver light.

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists.

Visual Studio also includes a web-site editor and designer that allows web pages to be authored by dragging and dropping widgets. It is used for developing VB.NET application efficiently to get input and output design easiest one. It will be run at windows application based services provide the user.

4.2DEVELOPMENT TOOLS AND TECHNOLOGIES

VB.NET Version 10.0:

The latest version of Visual Basic .NET, which runs on .NET framework 4.5. Async Feature, Iterators, Call Hierarchy, Caller Information and Global Keyword in Namespace Statements are some of the major features introduced in this version of VB. **Visual Basic .NET** (**VB.NET**) is an object-oriented computer programming language that can be viewed as an evolution of the classic Visual Basic (VB), implemented on the .NET Framework. Microsoft currently supplies two main editions of IDEs for developing in Visual Basic: Microsoft Visual Studio 2012, which is commercial software and Visual Basic Express Edition 2012,which is free of charge. The command-line compiler, VBC.EXE, is installed as part of the freeware .NET Framework SDK. Mono also includes a command-line VB.NET compiler. The most recent version is VB 2012, which was released on August 15, 2012.

My goal in this article is to provide you with an introductory,yet intensive, look at Visual Basic .NET and the new Microsoft®.NET platform. In order to learn what Visual Basic .NET is all about, you must first understand a few core aspects of the .NET platform. This article will build your knowledge of Visual Basic .NET from the ground up, so I'll begin by discussing the new programming model and the high-level architecture of the platform's execution engine called the common language runtime (CLR).

While explaining what the CLR is and how it works, I'll show a few examples using Visual Basic .NET. As you'll see, Visual Basic® has undergone a significant overhaul to accommodate the CLR and its associated programming model. Consequently, Visual Basic .NET has many new object-oriented design features and much higher levels of type safety than previous versions of Visual Basic.either language can be used to write software that takes full advantage of the CLR .NET Framework. Now, let me get started by introducing the core concepts of the .NET platform

Features of .NET

- IO management
- Windows and Web Controls
- Database access
- Multithreading
- Remoting
- Reflections

ORACLE 10G:

Oracle 10g has come with purpose of improving manageability and performance in all areas, right from the process of installation, server configuration, database upgrades to application tuning, space and storage management and so on. This Oracle version has been designed to reduce the cost of manageability and deliver high performance for all key workloads. Also various new features are provided for high-availability, including new flashback capabilities, virtualization of computing resources in Grid environment that reduce the cost of hardware and storage, security enhancement, Business intelligent solutions etc. Let have a glance at some exiting features of Oracle 10g.

- Clustering
- Grid computing
- Server manageability
- Network management
- Storage management
- Space,object transaction management
- Back up recovery management
- Reduce down time for application and database upgrades

ADO.Net:

An evolutionary, more flexible successor to ADO.A system designed for connected environments. A programming model with advanced XML support A set of classes, interfaces, structures, and enumerations that manage data access from within the .NET Framework

Data Providers

- MS SQL Server 7.0+
- Oracle
- OLE DB (old SQL & Access- Jet 4.0)
- Open Database Connectivity (ODBC)- earlier Visual Studio, Access Driver, ODBC for Oracle

CHAPTER 5

PROJECT DESCRIPTION

5.1PROBLEM DEFINITION:

This system developed will reduce the manual work and avoid redundant data. By maintaining the attendance manually, then efficient reports cannot be generated. The system can generate efficient weekly,consolidate report based on the attendance. As the attendances are maintained in registers it has been a tough task for admin and staff to maintain for long time. Instead the software can keep long and retrieve the information when needed.

5.2 PROJECT OVERVIEW

Attendance Management System basically has two main modules for proper functioning

- Admin module is has rights for creating any new entry of faculty and student details.
- User has a rights of making daily attendance, generating report. Attendance report can be taken by given details of student details, date, class.

5.3 MODULE DESCRIPTION

The system should be designed in such a way that only authorized people should be allowed to access some particular modules. The records should be modified by only administrators and no one else. The user should always be in control of the application and not the vice versa.

The user interface should be consistent so that the user can handle the application with ease and speed. The application should be visually, conceptually clear.

5.3.1 ADMINISTRATOR MODULE:

- **Student Details:**

  In this module deals with the allocation of roll no and personal details for new batch.It will generate of personal details of student and academic details of the students with the photos.

- **Staff Details:**

  - It helps to allot the subject and the subject code to the particular staffs.
  - It provides the facility to have a user name and password to the staffs .

- **Time table details:**

  - It will retrieve the subject information from the subject database and assign time table to the staffs.
  - It will help the admin, staff to make the entry of attendance based of the subject and period allotted to the respective staff.

- **Attendance details:**

  - It will be makes to the attendance database all students. Entered attendance to stored in the database subject ,period wise into the particular date.
  - It will help s to the get report of weekly and consolidate of the attendance.

Report details:

Report can be taken by daily,weekly and consolidate:

- weekly report get all hour details of attendance starting date to ending date and display the status
- Consolidate report get all student attendance details starting date to ending date status help for the eligibility criteria of the student to attend the examination.

5.3.2 STAFFS MODULE:

- **Attendance details:**
  - It assists the staff to mark attendance to the students for their subject. This will authenticate the staff before making the entry.
- **Report details:**
  1. weekly report get particular hour details of attendance from starting date to ending date and display the status .
  2. consolidate report get all student attendance details from starting date to ending date status help for the eligibility criteria of the student to attend the examination

5.4 SYSTEM FLOW DIAGRAM:

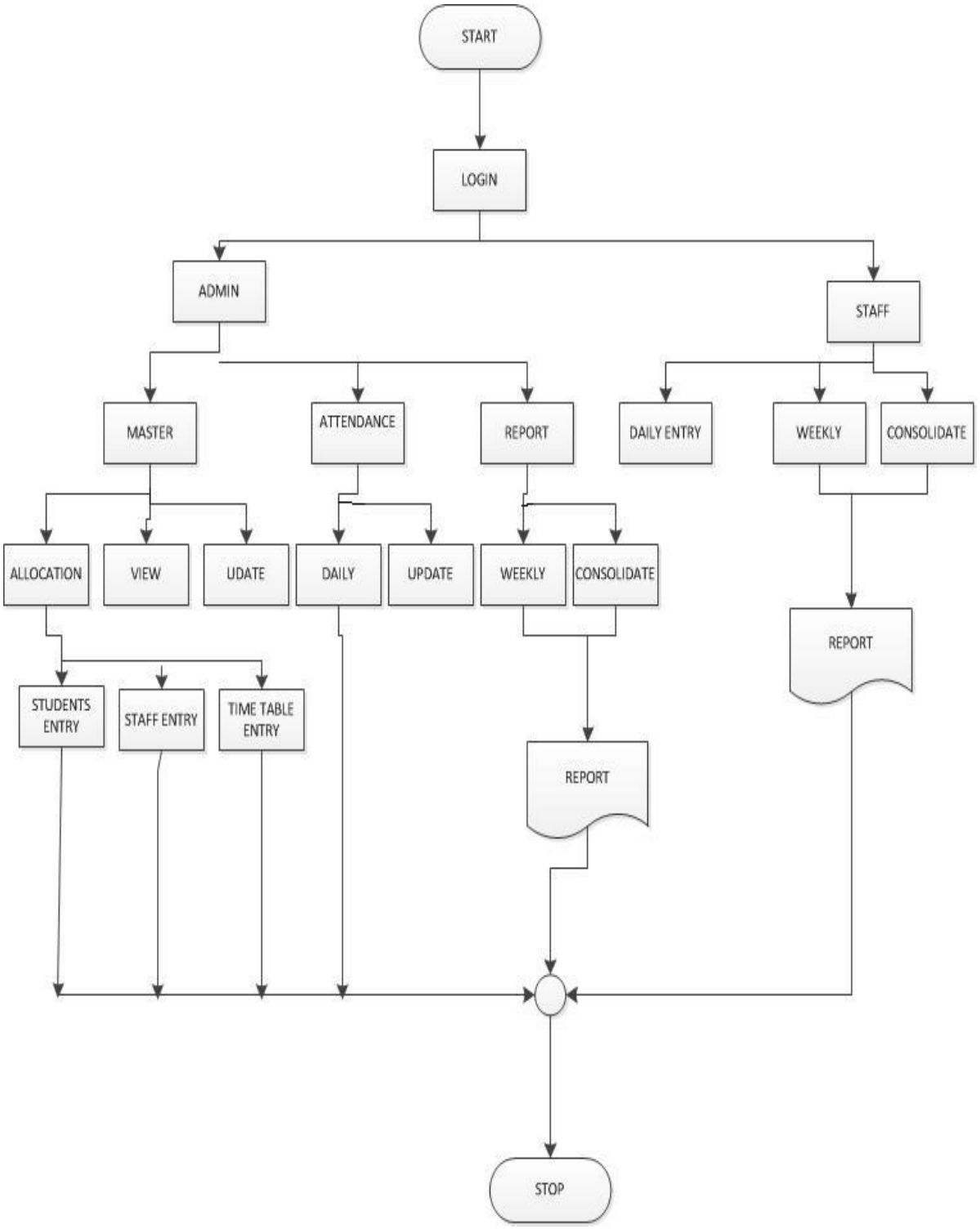


Figure 5.4-System Flow Diagram

5.5 Data Flow Diagram

5.5.1 DFD level 0:

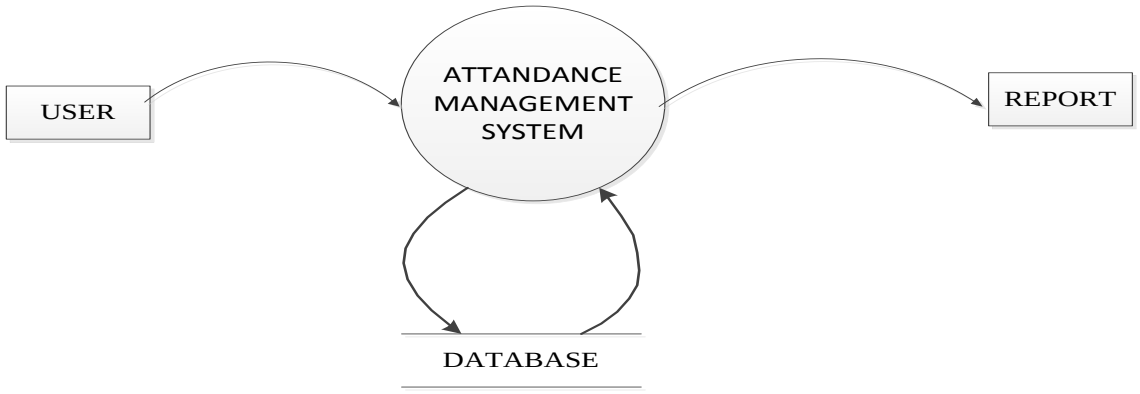


Figure 5.5.1-DataFlowDiagram Level1

5.5.2 DFD level 1:

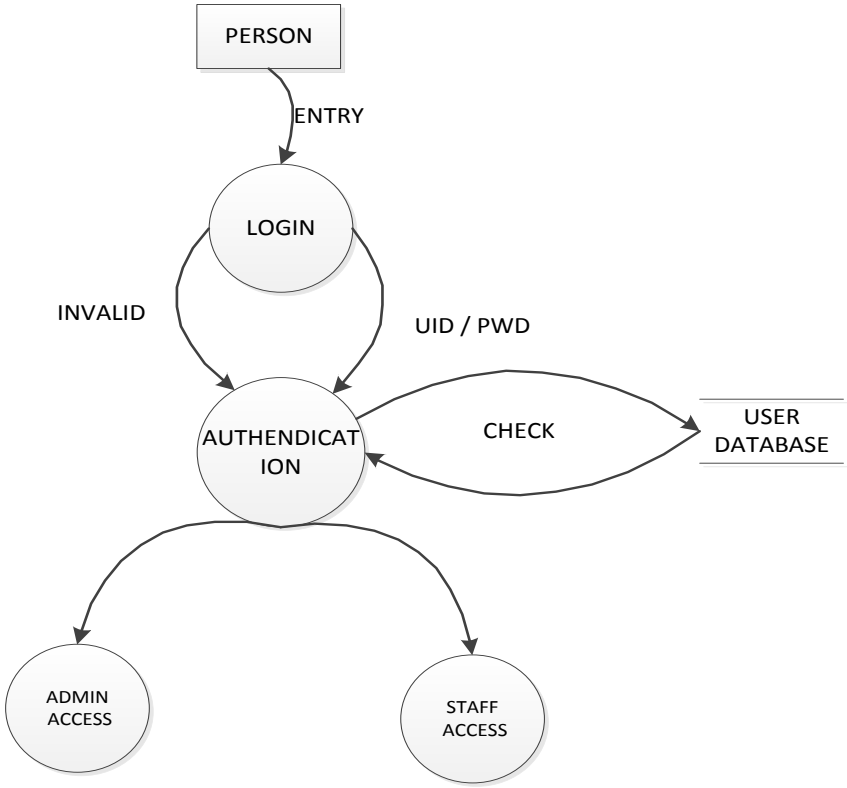


Figure 5.5.2-DataFlowDiagram Level1

5.5.3 DFD level 2:

5.5.3.1 Admin:

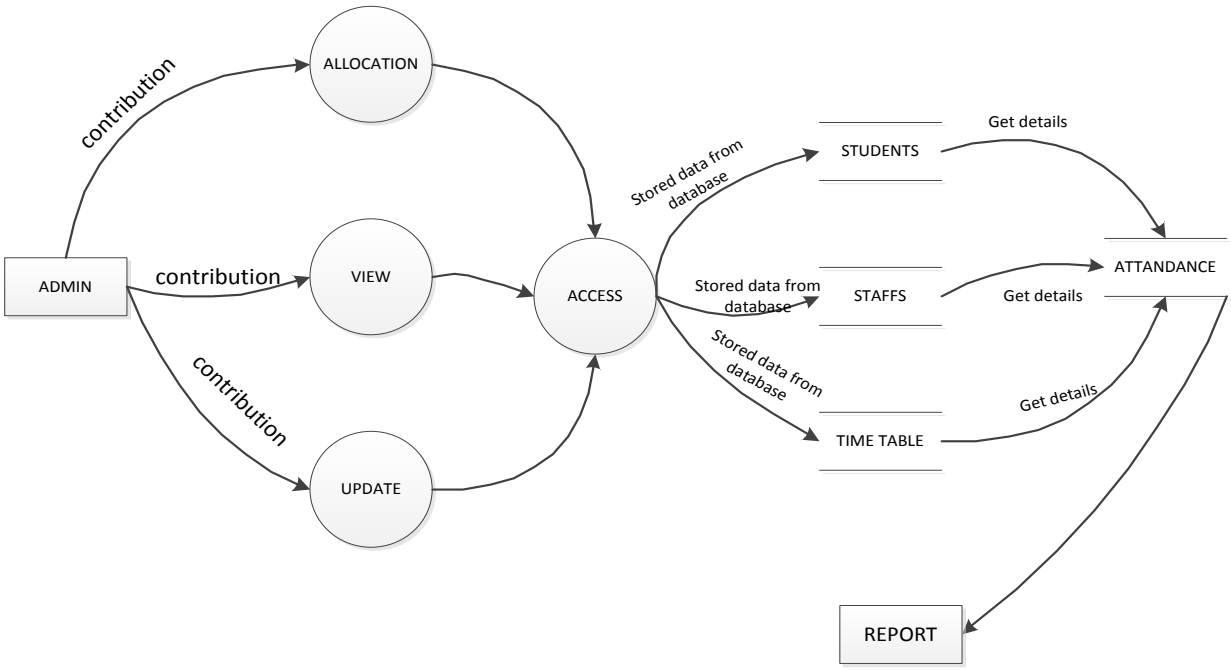


Figure 5.5.3.1-DataFlowDiagram Level2

5.5.3.2 staffs:

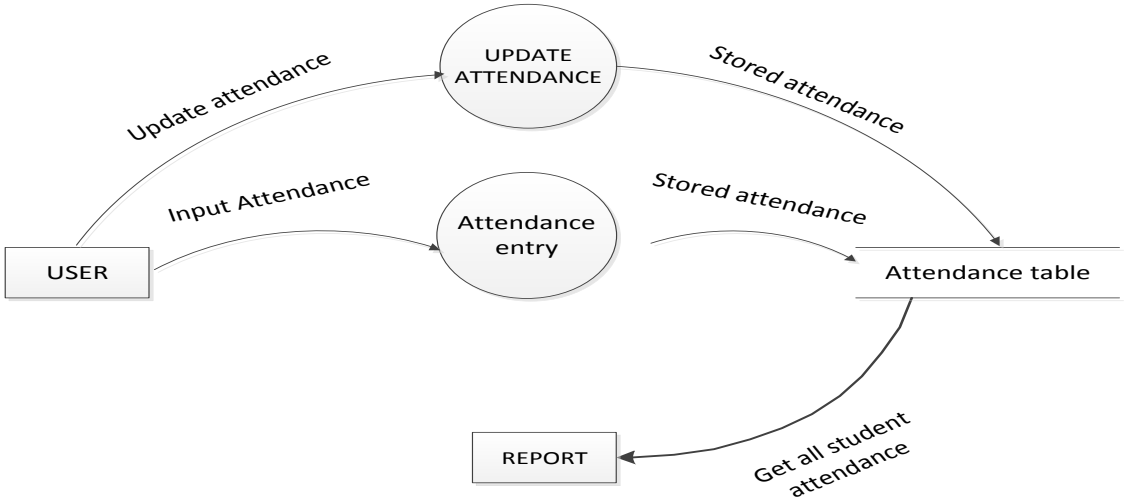


Figure 5.5.3.2-DataFlowDiagram Level2

5.6 SYSTEM DESIGN:

5.6.1 Entity Relationship Diagram:

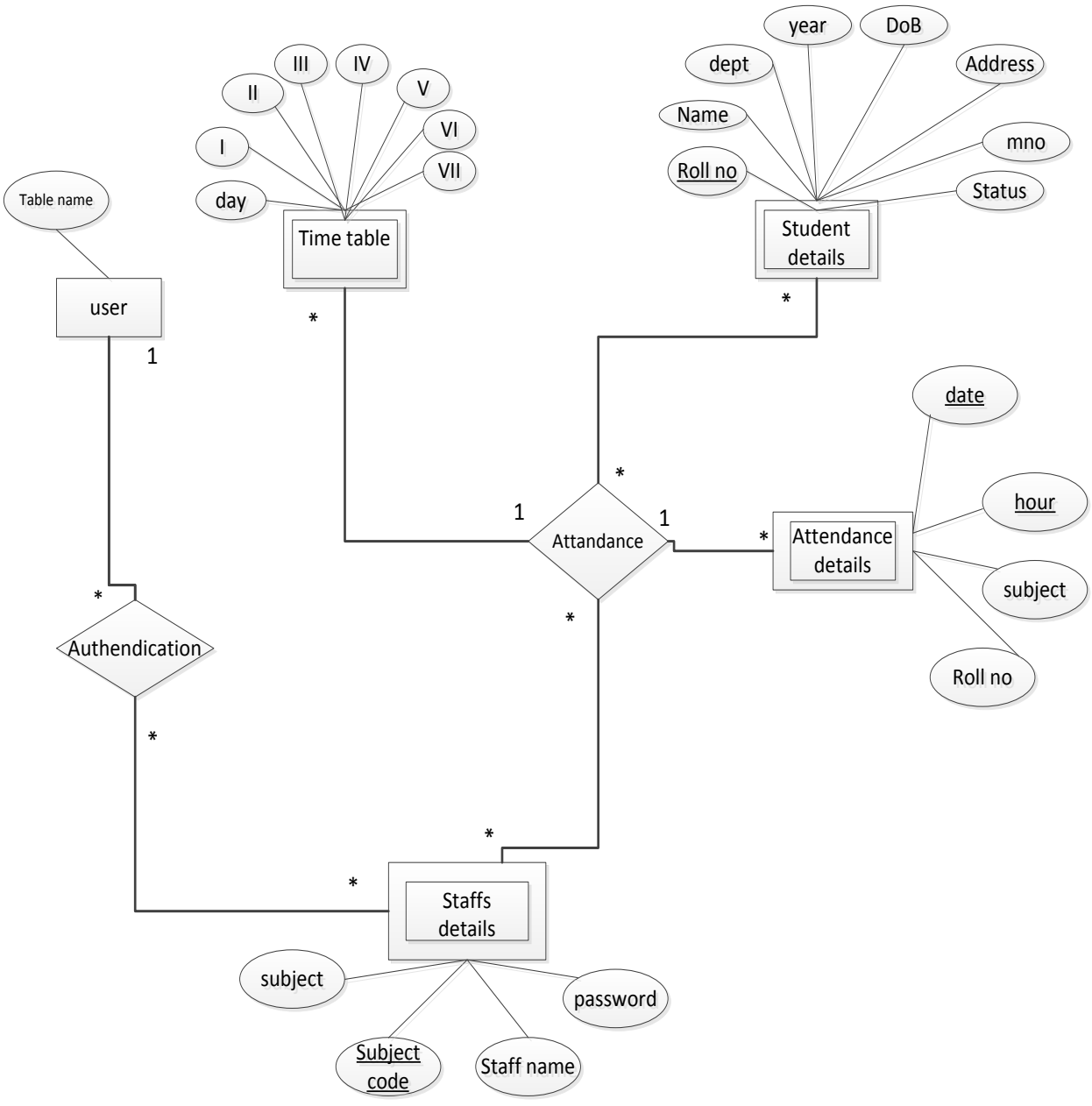


Figure 5.6.1-Entity Relationship Diagram

5.6.2 Use case Diagram:

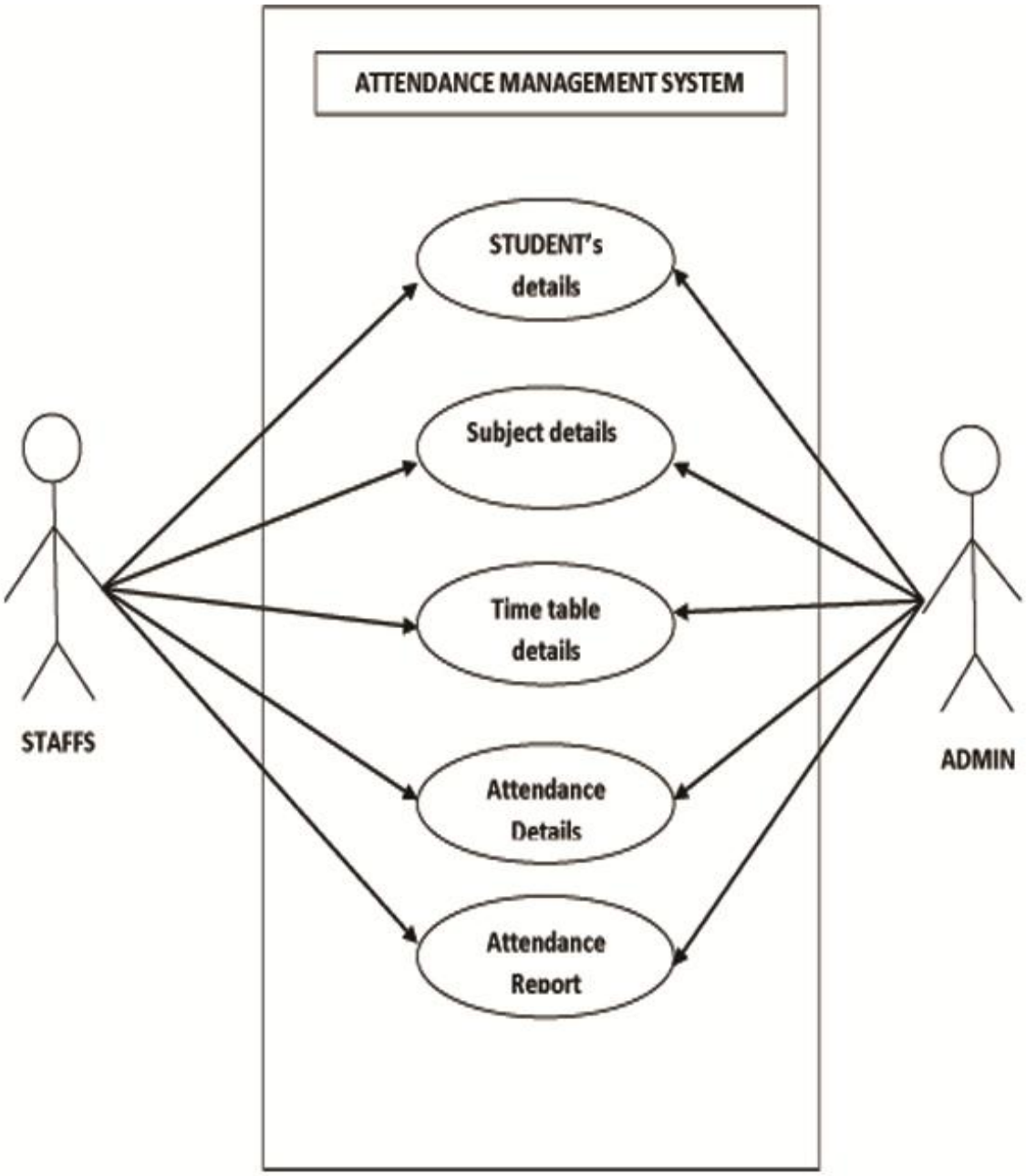


Figure:5.6.2 -Use case Diagram

5.6.2 Database Design:

5.6.2.1 LOGIN TABLE:

- To create a login details for the table.

FIELDS	DATATYPE	CONTRAINTS	DESCRIPTION
Tablename	varchar(20)	primarykey	Stored number of tables from login

Table:5.7.2.1 -Login Table

5.6.2.2 Staffs Table:

- To create username and password for the staff details.

FIELDS	DATATYPE	CONSTRAINTS	DESCRIPTION
Scode	varchar(20)	primarykey	Define separate subject code id
ssname	Varchar(15)	NotNull	Short subject name (ex:cpp)
sname	Varchar(20)	NotNull	Staffs name
Password	Varchar(20)	NotNull	Staff login password

Table:5.7.2.2 –Student details Table

5.6.2.3 Student table:

- To create table for Student personal details for our department.

FIELDS	DATATYPE	CONSTRAINTS	DESCRIPTION
Rollno	Varchar(15)	Primarykey	Student rollnumber
Name	Varchar(20)	NotNull	Student name
Dept	Varchar(30)	NotNull	Department name
Year	Number	NotNull	Batch year
DOB	Varchar(20)	NotNull	Student date of birth
ADDRESS	Varchar(20)	NotNull	Student permanent address
MNO	Varchar(20)	NotNull	Student mobile number
EID	Varchar(30)	NotNull	Student E-mail id
CSTATUS	Varchar(20)	NotNull	Student status for dayscholler/Hosteller

Table:5.7.2.3 –Staff Details Table

5.6.2.4 Time table:

- To create the subject time table for a particular class.

FIELDS	DATATYPE	CONSTRAINTS	DESCRIPTION
Day	Varchar(20)	Primarykey	Days insert (ex:Monday)
I	Varchar(20)	NotNull	set the period for 1 particular subject
II	Varchar(20)	NotNull	set the period for 2 particular subject
III	Varchar(20)	NotNull	set the period for 3 particular subject
IV	Varchar(20)	NotNull	set the period for 4 particular subject
V	Varchar(20)	NotNull	set the period for 5 particular subject
VI	Varchar(20)	NotNull	set the period for 6 particular subject
VII	Varchar(20)	NotNull	set the period for 7 particular subject

Table:5.7.2.4 -Time Table

5.6.2.5 Attendance table:

- To create attendance details for particular class .

FIELDS	DATATYPE	CONSTRAINTS	DESCRIPTION
Dates	Date	Primarykey	Enter day by day attendance
Hour	Number	primarykey	Set particular hour only
Subject	Varchar(15)	NotNull	Particular Subject
Rollno (1 to 60)	Varcahar(20)	NotNull	Enter Present absent details in particular student(ex:M11MCA001)

Table:5.7.2.5 -Attendance Table

5.6.3 INPUT DESIGN

Input design is part of overall system design that requires special attention designing input data is to make the data entered easy and free from **errors**. The input forms are designed using the controls available in .NET framework. Validation is made for each and every data that is entered. Help information is provided for the users during when the customer feels difficult.

Input design is the process of converting the user originated inputs to a computer based format. A system user interacting through a workstation must be able to tell the system whether to accept the input to produce reports. The collection of input data is considered to be most expensive part of the system design. Since the input has to be planned in such a manner so as to get relevant information, extreme care is taken to obtain pertinent information

This project first will entered to the input of allocation forms it will be created on student details form and subject entry form, time table form .it will helps to calculate subject wise attendance system. next one if u want any verification on your data's also available in details show forms. Attendance to entered single subject wise or all subject wise attendance system available in this project.

5.6.4 OUTPUT DESIGN

Output design this application "**Student Attendance management system"** generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

The output is designed in such a way that it is attractive, convenient and informative. Forms are designed with various features, which make the console output more pleasing.

As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision making. Form design elaborates the way output is presented and the layout available for capturing information.

One of the most important factors of the system is the output it produces. This system refers to the results and information generated. Basically the output from a computer system is used to communicate the result of processing to the user.

Attendance management system to show the report subject wise attendance maintaining by staffs. Taken as a whole report obtain on a administrator privileges only. this forms will show weekly report and consolidate report generated date, batch, and class wise to our end user. we want to change our report to convert Excel format .if you want change any modification.

CHAPTER 6
SYSTEM TESTING

6.1 Introduction

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. To uncover the errors software techniques are used. These techniques provide systematic guidance for designing test that

(1) Exercise the internal logic of software components, and

(2) Exercise the input and output domains of the program to uncover errors In program function, behavior and performance.

6.1.1 Steps: Software is tested from two different perspectives:

(1) Internal program logic is exercised using ―White box‖ test case design Techniques.

(2) Software requirements are exercised using ―block box‖ test case Design techniques.

In both cases, the intent is to find the maximum number of errors with the Minimum amount of effort and time.

6.2 Testing Methodologies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when deadline pressure begins to rise, progress must be measurable and problems must surface as early as possible. Following testing techniques are well known and the same strategy is adopted during this project testing.

6.2.1 Unit testing:

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System. by giving correct manual input to the system ,the datas are stored in database and retrieved. If you want required module to access input or get the output from the End user. any error will accrued the time will provide handler to show what type of error will accrued .

6.2.2 System testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Below we have described the two types of testing which have been taken for this project. it is to check all modules worked on input basis .if you want change any values or inputs will change all information. so specified input is must.

6.2.4 Performance Testing

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted.

This project reduce attendance table, codes. it will generate report fast.no have extra time or waiting of results .entered correct data will show result few millisecond. just used only low memory of our system. Automatically do not getting access at another software. Get user permission and access to other applications.

6.3 Test cases

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:-*manual* and *automated.* A manual test case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger errors.

TEST CASE:

6.3.1 Agent and admin login form

Sno	Test case id	Test case name	Test case desc	Step	Expected result	Actual Result	Test case status pass/fail
1	Login admin	Validate login	To verify that login name on login page	Enter the login name and password and click submit button	Login successful or an error message “In valid login or password” must be displayed	Login successful	Pass
2	Login Staff	Validate login	To verify that login name on login page	Enter the login name and password and click submit button	Login successful or an error message “In valid login or password” must be displayed	Login successful	Pass
3	Password	Validate password	To verify that password on login page	Enter password and login name click submit button	An error message “password invalid” must be displayed	An error message “password invalid” must be displayed	fail

6.3.2 MASTER form

Sno	Test case id	Test case name	Test case desc	Step	Expected result	Actual Result	Test case status pass/fail
1	Create suden t details	Validate allocatio n form	To allocate separate roll no for the students	Nothing entered and click submit button	An error message student name not equal to null must be displayed	Inserted succesfu l	Pass
2	Creat e staff detail s	Validate allocatio n form	To allocate separate subject usernam e passwor d for the staffs	Nothing entered and click submit button	An error message staff details password,usernam e not equal to null must be displayed	Inserted succesfu l	Pass
3	Creat e time table	Validate allocate period form	To verify that data stored on database	Nothing entered and click submit button	An error message not click not allocation subject table not equal to null must be displayed	Inserted succesfu l	Pass
4	View	Check details of all data	To verify that data stored on database	generate d	An error message return null will be displayed	An error message return null will be displaye d	fail

6.3.3 Report form

Sno	Test case id	Test case name	Test case desc	Step	Expected result	Actual Result	Test case status pass/fail
1	Weekly report	Validate class attendance form	To select that source and destination	Nothing entered and click submit button	An error message on not selected	Retrived data successful	Pass
2	Consolidate report	Validate class attendance form	To select that depart on and time	Nothing entered and click submit button	An error message on not selected	Retrived data successful	Pass

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 Purpose

System implementation is the important stage of project when the theoretical design is tuned into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover Planning

Planning is the first task in the system implementation. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments.

The line managers controlled through an implementation coordinating committee. The committee considers ideas, problems and complaints of user department, it must also consider:

- The implication of system environment
- Self selection and allocation for implementation tasks
- Consultation with unions and resources available
- Standby facilities and channels of communication

Student Attendance management system will implement student details ,staff handle subjects details, separate login details ,time table details. It will used to entered subject wise attendance .This application elaborate attendance table generate weekly, consolidate report provide to the End user. Mostly this application will calculate date wise attendance .To select starting date to end date generate reports at the time of activities.

7.2 SYSTEM MAINTENANCE

Software maintenance is far more than finding mistakes. Provision must be made for environment changes, which may affect either the computer, or other parts of the computer based systems. Such activity is normally called maintenance. It includes both the improvement of the system functions and the corrections of faults, which arise during the operation of a new system.

It may involve the continuing involvement of a large proportion of computer department recourses. The main task may be to adapt existing systems in a changing environment.

Back up for the entire database files are taken and stored in storage devices like flash drives, pen drives and disks so that it is possible to restore the system at the earliest. If there is a breakdown or collapse, then the system gives provision to restore database files. Storing data in a separate secondary device leads to an effective and efficient maintains of the system. The nominated person has sufficient knowledge of the organization's computer passed based system to be able to judge the relevance of each proposed change.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

To conclude, Project Data Grid works like a component which can access all the databases and picks up different functions. It overcomes the many limitations incorporated in the attendance.

- Easy implementation Environment
- Generate report Flexibly

8.2 Scope for future development

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

- Discontinue of particular student eliminate potential attendance.
- Bar code Reader based attendance system.
- Individual Attendance system With photo using Student login.

CHAPTER 9

APPENDICES

9.1 Source code:

LOGIN:

```
Imports System.Data

Public Class login
    Dim con As New ADODB.Connection
    Dim rs, rs1 As New ADODB.Recordset
    Public str, temp1, temp2, temp3, temp4 As String
    Dim i As Integer
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        rs = New ADODB.Recordset
        rs1 = New ADODB.Recordset
        If String.Equals(TextBox1.Text, "Admin") Or String.Equals(TextBox1.Text,
"admin") Or String.Equals(TextBox1.Text, "ADMIN") And
String.Equals(TextBox2.Text, "Admin") Or String.Equals(TextBox2.Text, "admin") Or
String.Equals(TextBox2.Text, "ADMIN") Then
            temp4 = "MDIParent2"
            TextBox1.Text = ""
            TextBox2.Text = ""
            MDIParent2.Show()
            Me.Hide()
            i = 1
        Else
            Try
                str = "select * from logintable"
                rs.Open(str, con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
                rs.MoveFirst()
                While (rs.EOF <> True)
                    str = "select * from " & rs.Fields("tablename").Value & ""
                    rs1.Open(str, con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
                    While (rs1.EOF <> True)
                        If String.Equals(rs1.Fields("sname").Value, TextBox1.Text) And
String.Equals(rs1.Fields("pass").Value, TextBox2.Text) Then
                            temp1 = rs1.Fields("sname").Value
                            temp2 = rs1.Fields("scode").Value
                            temp3 = rs1.Fields("ssname").Value
                            temp4 = "MDIParent1"
                            TextBox1.Text = ""
```

```
                    TextBox2.Text = ""

                    MDIParent1.Show()
                    Me.Hide()
                    i = 1
                    Exit While
                End If
                rs1.MoveNext()
            End While
            rs1.Close()
            rs.MoveNext()

        End While


        If i = 0 Then
            MsgBox("LOGIN NOT VAILD")
        End If
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End If



End Sub

Private Sub Form6_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    con = New ADODB.Connection

    If (con.State = ConnectionState.Open) Then
        con.Close()
    End If

    con.Open("driver={microsoft ODBC for
Oracle};server=test;uid=M11MCA20;pwd=M11MCA20;")
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    End

End Sub
End Class
```

Attendance Entry:

```
Public Class attentry
    Dim con As New ADODB.Connection
    Dim rs, rs1 As New ADODB.Recordset
    Dim str, dat As String
    Dim att As String
    Dim i As Integer = 1
    Dim flag As Integer = 1
    Dim chk1 As New DataGridViewCheckBoxColumn()
     Dim chk As New DataGridViewCheckBoxColumn()
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        rs = New ADODB.Recordset
        Try
            str = "select * from " & ComboBox1.SelectedItem & "_" &
ComboBox5.SelectedItem & ""
            rs.Open(str, con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
            rs.MoveFirst()
            DataGridView1.Rows.Clear()
            i = 1
            While (rs.EOF <> True)
                Dim row As String() = New String() {i, rs.Fields("rollno").Value,
rs.Fields("name").Value}
                DataGridView1.Rows.Add(row)
                i = i + 1
                rs.MoveNext()
            End While

            rs.Close()


            DataGridView1.Columns.Add(chk)
            chk.HeaderText = "PRESENT/ABSENT"
            chk.Name = "chk"
            chk.Selected = True


            DataGridView1.Columns.Add(chk1)
            chk1.HeaderText = "ONDUTY"
            chk1.Name = "chk1"
            timetb()


        Catch ex As Exception
```

```
            'rs.Close()
            MsgBox(ex.ToString)
        End Try


    End Sub

    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        con = New ADODB.Connection

        'If (con.State = ConnectionState.Open) Then
        '    con.Close()
        'End If

        con.Open("driver={microsoft ODBC for
Oracle};server=test;uid=M11MCA20;pwd=M11MCA20;")
        Label15.Text = login.temp1
        Label16.Text = login.temp2
        Label7.Text = login.temp3


    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        str = String.Empty
        att = ""
        flag = 1
        dat = DateTimePicker1.Value.Date.ToString("dd-MMM-yyyy")

            For Me.i = 0 To DataGridView1.RowCount - 1
                If DataGridView1.Rows(i).Cells(3).Value = True Then
                    If (flag < 2) Then
                    att = "'P'"
                        flag = 3
                Else
                    att = att + ",'P'"
                    End If

            ElseIf DataGridView1.Rows(i).Cells(4).Value = True Then
                If (flag < 2) Then
                    att = "'O'"
                    flag = 3

                Else
```

```
                att = att + ",'O'"
            End If
        Else
            If (flag < 2) Then
                att = "'A'"
                flag = 3

            Else
                att = att + ",'A'"
            End If
            End If

    Next
    Try
        str = "insert into " & ComboBox1.SelectedItem & "_" &
ComboBox5.SelectedItem & "_" & ComboBox2.SelectedItem & "_" &
ComboBox3.SelectedItem & "_att values('" & dat & "'," & ComboBox4.Text & ",'" &
Label7.Text & "'," & att & ")"
        con.Execute(str)
        MsgBox("insert")
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try


End Sub

Private Sub CREATEToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CREATEToolStripMenuItem.Click

    rs1 = New ADODB.Recordset
    str = "select * from " & ComboBox1.Text & "_" & ComboBox5.Text & ""
    rs1.Open(str, con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
    rs1.MoveFirst()
    str = "create table " & ComboBox1.Text & "_" & ComboBox5.Text & "_" &
ComboBox2.Text & "_" & ComboBox3.Text & "_att(days Date,hour number,subject
varchar(15),primary key(days,hour))"
    con.Execute(str)
    While (rs1.EOF <> True)
        str = "alter table " & ComboBox1.Text & "_" & ComboBox5.Text & "_" &
ComboBox2.Text & "_" & ComboBox3.Text & "_att add(M" &
rs1.Fields("rollno").Value & " varchar(20))"
        con.Execute(str)
        rs1.MoveNext()
    End While
```

```
    End Sub

    Private Sub timetb()
        Dim temp As String
        rs1 = New ADODB.Recordset
        ComboBox4.Text = "Select One"
        Try
                    temp = "select * from " & ComboBox1.Text & "_" & ComboBox5.Text &
"_" & ComboBox2.Text & "_" & ComboBox3.Text & "_time where(day='" &
DateTimePicker1.Value.ToString("dddd") & "')"
            rs1.Open(temp, con, ADODB.CursorTypeEnum.adOpenUnspecified,
ADODB.LockTypeEnum.adLockPessimistic)
            ComboBox4.Items.Clear()

            For Me.i = 1 To 7
                If String.Equals(rs1.Fields(i).Value, Label7.Text) Then
                    ComboBox4.Items.Add(i)
                End If

            Next
            rs1.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try

    End Sub

    Private Sub DateTimePicker1_ValueChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles DateTimePicker1.ValueChanged
        timetb()

    End Sub

    Private Sub DELETEToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DELETEToolStripMenuItem.Click
        str = "drop table " & ComboBox1.Text & "_" & ComboBox5.Text & "_" &
ComboBox2.Text & "_" & ComboBox3.Text & "_" & Label7.Text & " "
        con.Execute(str)
        MsgBox("TABLE DELETED SUCCESSFULLY")

    End Sub

    Private Sub HOMEToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles HOMEToolStripMenuItem.Click
        MDIParent1.Show()
        Me.Close()
```

```
    End Sub


    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub

    Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles CheckBox1.CheckedChanged

        If CheckBox1.Checked = True Then
            i = 0
            While (i < DataGridView1.Rows.Count)
                DataGridView1.Rows(i).Cells(3).Value = True
                i = i + 1
            End While
        Else
            i = 0
            While (i < DataGridView1.Rows.Count)
                DataGridView1.Rows(i).Cells(3).Value = False
                i = i + 1
            End While
        End If

    End Sub

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
DataGridView1.CellContentClick
        i = 0
        While (i < DataGridView1.Rows.Count)
            If DataGridView1.Rows(i).Cells(3).Value <> True Then
                DataGridView1.Rows(i).Cells(3).Style.BackColor = Color.Red
            Else
                DataGridView1.Rows(i).Cells(3).Style.BackColor = Color.White
            End If
            i = i + 1
        End While
    End Sub
End Class
```

WEEKLY REPORT:

```
Imports Microsoft.Office.Interop

Public Class awreport

    Dim conn As New ADODB.Connection
    Dim rs, rs1 As New ADODB.Recordset
    Dim str, dat As String
    Dim i, j, flag, diff, count1 As New Integer
    Dim ro, temp, tot_day, pre_day, ab_day As Integer
    Dim holiday As String
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click

        holiday = String.Empty

        Try

            rs = New ADODB.Recordset
            rs1 = New ADODB.Recordset


            DataGridView1.Rows.Clear()
            DataGridView1.Columns.Clear()
            DataGridView2.Rows.Clear()
            DataGridView2.Columns.Clear()

            Dim clm1 As New DataGridViewTextBoxColumn()
            DataGridView2.Columns.Add(clm1)
            clm1.HeaderText = ComboBox1.Text + "-" + ComboBox5.Text
            clm1.Name = "clm1"

            Dim clm2 As New DataGridViewTextBoxColumn()
            DataGridView2.Columns.Add(clm2)
            clm2.HeaderText = "SEMESTER" + "-" + ComboBox3.Text
            clm2.Name = "clm3"
            DataGridView2.Columns(1).Width = 130


            Dim clm3 As New DataGridViewTextBoxColumn()
            DataGridView1.Columns.Add(clm3)
            clm3.HeaderText = "ROLLNO"
            clm3.Name = "clm3"

            Dim clm4 As New DataGridViewTextBoxColumn()
            DataGridView1.Columns.Add(clm4)
```

```
        clm4.HeaderText = "STUDENT NAME"
        clm4.Name = "clm4"
        DataGridView1.Columns(1).Width = 130


        str = "select * from " & ComboBox1.SelectedItem & "_" &
ComboBox5.SelectedItem & ""
        rs.Open(str, conn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
        rs.MoveFirst()

        While (rs.EOF <> True)
            Dim row As String() = New String() {rs.Fields("rollno").Value,
rs.Fields("name").Value}
            DataGridView1.Rows.Add(row)
            rs.MoveNext()
        End While
        rs.Close()

        Dim d As Date
        d = DateTimePicker1.Value.Date
        Dim d1 As Date
        d1 = DateTimePicker2.Value.Date

        diff = DateDiff(DateInterval.Day, d, d1)
        j = 2
        While diff >= 0
            Try
                str = "Select * from " & ComboBox1.SelectedItem & "_" &
ComboBox5.Text & "_" & ComboBox2.SelectedItem & "_" &
ComboBox3.SelectedItem & "_att where(days='" & d.Date.ToString("dd-MMM-yyyy")
& "')order by hour asc "
                rs1.Open(str, conn, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
                rs1.MoveFirst()
                count1 = 1
                Dim dtxt As New DataGridViewTextBoxColumn()
                DataGridView2.Columns.Add(dtxt)
                dtxt.HeaderText = d.Date.ToString("dd-MMM-yyyy")
                dtxt.Width = 140


                While (rs1.EOF <> True)
                    Dim dtxt1 As New DataGridViewTextBoxColumn()
                    DataGridView1.Columns.Add(dtxt1)
                    dtxt1.HeaderText = rs1.Fields("hour").Value.ToString
```

```
                dtxt1.Width = 20

                Dim rcount As Integer = 0
                Dim count As Integer = 3
                While (rs1.Fields.Count > count)
                    DataGridView1.Rows(rcount).Cells(j).Value = rs1.Fields(count).Value
                    DataGridView1.Rows(rcount).HeaderCell.Value = (rcount +
1).ToString

                    If String.Equals(rs1.Fields(count).Value, "A") Then
                        DataGridView1.Rows(rcount).Cells(j).Style.BackColor = Color.Red
                    End If
                    rcount = rcount + 1
                    count = count + 1
                End While
                count1 = count1 + 1
                j = j + 1
                rs1.MoveNext()
            End While
            rs1.Close()
            d = DateAdd(DateInterval.Day, 1, d)
            diff = diff - 1
        Catch ex As Exception

            holiday += "(" + d.Date.ToString("dd-MMM-yyyy") + "-HOLIDAY) "

            d = DateAdd(DateInterval.Day, 1, d)

            diff = diff - 1
            rs1.Close()
        End Try
    End While

    ' MsgBox(holiday)
    DataGridView1.Rows.Add(holiday)
Catch ex As Exception
    MsgBox(ex.ToString)
End Try

End Sub

Private Sub creport_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    conn = New ADODB.Connection
    rs = New ADODB.Recordset
```

```
        conn.Open("driver={microsoft ODBC for
Oracle};server=test;uid=M11MCA20;pwd=M11MCA20;")
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        Panel1.Visible = True
        ProgressBar1.Minimum = 0
        ProgressBar1.Maximum = 100

        Dim xlApp As Excel.Application
        Dim xlWorkBook As Excel.Workbook
        Dim xlWorkSheet As Excel.Worksheet
        Dim misValue As Object = System.Reflection.Missing.Value
        Dim i As Integer
        Dim j As Integer

        xlApp = New Excel.Application

        xlWorkBook = xlApp.Workbooks.Add(misValue)

        xlWorkSheet = xlWorkBook.Sheets("sheet1")

        flag = 0
        j = 1



        xlWorkSheet.Cells(1, 1) = "Dr.Mahalingam College of Engineering & Technology
".ToString
        xlWorkSheet.Cells(2, 1) = "NPT -MCET Campus, Udumalai Road -
Makkinaickenpatti - Pollachi".ToString
        xlWorkSheet.Cells(3, 1) = "Phone : 04259-236030 Fax : 04259-236070".ToString
        xlWorkSheet.Cells(4, 1) = "E-Mail : principal@drmcet.ac.in Web Site :
www.mcet.in".ToString
        xlWorkSheet.Range("A5").Value = "BATCH:" + ComboBox1.Text + "-" +
ComboBox5.Text + "       ATTENDANCE DETAILS FROM  " +
DateTimePicker1.Value.ToString("dd-MMM-yyyy") + "  TO  " +
DateTimePicker2.Value.ToString("dd-MMM-yyyy") + "       SEMESTER:" + "-" +
ComboBox3.Text


        For Each col1 As DataGridViewColumn In DataGridView2.Columns
            If flag < 2 Then
```

```
            xlWorkSheet.Cells(6, col1.Index + 1) = col1.HeaderText.ToString
            flag = flag + 1
            j = j + 1
        Else
            j = j + 1
            xlWorkSheet.Cells(6, j) = col1.HeaderText.ToString
            For i = 1 To 6
                j = j + 1
                xlWorkSheet.Cells(6, j + i - 1) = "".ToString
            Next
        End If
    Next


    xlWorkSheet.Cells(6, 1) = "SNO".ToString
    flag = 0
    For Each col As DataGridViewColumn In DataGridView1.Columns
        If flag < 2 Then
            xlWorkSheet.Cells(6, col.Index + 2) = col.HeaderText.ToString
            flag = flag + 1
        Else
            xlWorkSheet.Cells(7, col.Index + 2) = col.HeaderText.ToString
        End If



    Next

    For i = 1 To DataGridView1.Rows.Count - 1
        xlWorkSheet.Cells(i + 7, 1) = i.ToString
        flag = 0
        For j = 0 To DataGridView1.ColumnCount - 1
            Dim vv As String
            If DataGridView1(j, i - 1).Value Is Nothing Then
                vv = "Niet ingevuld"
            Else
                vv = DataGridView1(j, i - 1).Value.ToString
                xlWorkSheet.Cells(i + 7, j + 2) = vv
                If flag < 2 Then
                    xlWorkSheet.Columns(j + 2).ColumnWidth = 15
                    'xlWorkSheet.Columns.Merge(2)
                    flag = flag + 1
                Else
                    xlWorkSheet.Columns(j + 2).ColumnWidth = 1
                End If
            End If
```

```
            ProgressBar1.Value = (i / DataGridView1.Rows.Count) * 100

        Next

    Next

    xlWorkSheet.Range("A1:AS1").Merge()
    xlWorkSheet.Range("A2:AS2").Merge()
    xlWorkSheet.Range("A3:AS3").Merge()
    xlWorkSheet.Range("A4:AS4").Merge()
    xlWorkSheet.Range("A5:AS5").Merge()

    xlWorkSheet.Range("D6:J6").Merge()
    xlWorkSheet.Range("K6:Q6").Merge()
    xlWorkSheet.Range("R6:X6").Merge()
    xlWorkSheet.Range("Y6:AE6").Merge()
    xlWorkSheet.Range("AF6:AL6").Merge()
    xlWorkSheet.Range("AM6:AS6").Merge()

    xlWorkBook.Activate()
    xlWorkBook.SaveAs("D:\export.xls")

    xlWorkBook.Close()
    xlApp.Quit()
    Panel1.Visible = False


    MsgBox("You can find your report at " & "D:\export.xls")

End Sub

Private Sub HOMEToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles HOMEToolStripMenuItem.Click
    MDIParent2.Show()
    Me.Close()

End Sub
End Class
```

Consolidate Report:

```
Imports Microsoft.Office.Interop
Public Class consli

    Dim con As New ADODB.Connection
```

```
    Dim rs, rs1 As New ADODB.Recordset
    Dim str, dat As String
    Dim i, j, k, diff, count1 As New Integer
    Dim pre_hours(100), tot_hours(100), ab_hours(100) As Integer
    Dim tot_day(100), pre_day(100), ab_day(100) As Double

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        DataGridView1.Rows.Clear()
        rollno()
        daycalc()

    End Sub

    Private Sub rollno()
        DataGridView1.Rows.Clear()
        Try
            str = "select * from " & ComboBox1.SelectedItem & "_" &
ComboBox5.SelectedItem & ""
            rs.Open(str, con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockOptimistic)
            rs.MoveFirst()
            i = 0
            While (rs.EOF <> True)
                Dim row As String() = New String() {rs.Fields("rollno").Value,
rs.Fields("name").Value}
                DataGridView1.Rows.Add(row)
                DataGridView1.Rows(i).HeaderCell.Value = (i + 1).ToString
                rs.MoveNext()
                i = i + 1
            End While
            rs.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try

    End Sub



    Private Sub adconsoli_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        con = New ADODB.Connection
        rs = New ADODB.Recordset
```

```
        con.Open("driver={microsoft ODBC for
Oracle};server=test;uid=M11MCA20;pwd=M11MCA20;")


        Label8.Text = login.temp1
        Label11.Text = login.temp2
        Label10.Text = login.temp3

    End Sub

    Private Sub daycalc()

        Dim pre_hours(100), tot_hours(100), ab_hours(100) As Integer
        Try
            Dim d As Date
            d = DateTimePicker1.Value.Date
            Dim d1 As Date
            d1 = DateTimePicker2.Value.Date
            diff = DateDiff(DateInterval.Day, d, d1)

            j = 2

            While diff >= 0

                Try
                    str = "Select * from " & ComboBox1.SelectedItem & "_" &
ComboBox5.Text & "_" & ComboBox2.SelectedItem & "_" &
ComboBox3.SelectedItem & "_att where(days='" & d.Date.ToString("dd-MMM-yyyy")
& "' and subject='" & Label10.Text & "')order by hour asc "
                    rs.Open(str, con, ADODB.CursorTypeEnum.adOpenDynamic,
ADODB.LockTypeEnum.adLockPessimistic)
                    rs.MoveFirst()

                    Dim temp(100), temp1(100) As Integer

                    Dim flag1(100) As Integer

                    While (rs.EOF <> True)
                        Dim rcount As Integer = 0
                        Dim count As Integer = 3
                        k = 0
                        While (rs.Fields.Count > count)
                            If String.Equals(rs.Fields(count).Value, "P") Or
String.Equals(rs.Fields(count).Value, "O") Then
                                pre_hours(k) = pre_hours(k) + 1
                            ElseIf String.Equals(rs.Fields(count).Value, "A") Then
```

```
                    ab_hours(k) = ab_hours(k) + 1
                End If
                tot_hours(k) = tot_hours(k) + 1


                rcount = rcount + 1
                count = count + 1
                k = k + 1
            End While
            j = j + 1

            rs.MoveNext()
        End While

        For Me.i = 0 To DataGridView1.Rows.Count - 1
            tot_day(i) = tot_day(i) + 1
        Next
        d = DateAdd(DateInterval.Day, 1, d)
        diff = diff - 1
        rs.Close()
    Catch ex As Exception
        d = DateAdd(DateInterval.Day, 1, d)
        diff = diff - 1
        rs.Close()
    End Try

  End While


Catch ex As Exception
  MsgBox(ex.ToString)
End Try



i = 0
While (i < DataGridView1.Rows.Count - 1)
  DataGridView1.Rows(i).Cells(2).Value = pre_hours(i)
  DataGridView1.Rows(i).Cells(3).Value = ab_hours(i)
  DataGridView1.Rows(i).Cells(4).Value = tot_hours(i)

  DataGridView1.Rows(i).Cells(5).Value = Math.Round((pre_hours(i) /
tot_hours(i) * 100), 2)

  i = i + 1
```

```
        End While

    End Sub

    Private Sub HOMEToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles HOMEToolStripMenuItem.Click
        MDIParent1.Show()
        Me.Close()


    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Panel1.Visible = True
        ProgressBar1.Minimum = 0
        ProgressBar1.Maximum = 100


        Dim xlApp As Excel.Application
        Dim xlWorkBook As Excel.Workbook
        Dim xlWorkSheet As Excel.Worksheet
        Dim misValue As Object = System.Reflection.Missing.Value
        Dim i As Integer
        Dim j As Integer


        xlApp = New Excel.Application

        xlWorkBook = xlApp.Workbooks.Add(misValue)

        xlWorkSheet = xlWorkBook.Sheets("sheet1")

        xlWorkSheet.Cells(1, 1) = "Dr.Mahalingam College of Engineering & Technology
".ToString
        xlWorkSheet.Cells(2, 1) = "NPT -MCET Campus, Udumalai Road -
Makkinaickenpatti - Pollachi".ToString
        xlWorkSheet.Cells(3, 1) = "Phone : 04259-236030 Fax : 04259-236070".ToString
        xlWorkSheet.Cells(4, 1) = "E-Mail : principal@drmcet.ac.in Web Site :
www.mcet.in".ToString

        xlWorkSheet.Range("A5").Value = "BATCH:" + ComboBox1.Text + "-" +
ComboBox5.Text + "       ATTENDANCE DETAILS FROM  " +
DateTimePicker1.Value.ToString("dd-MMM-yyyy") + "  TO  " +
DateTimePicker2.Value.ToString("dd-MMM-yyyy") + "       SEMESTER:" + "-" +
ComboBox3.Text
```

```
        For Each col As DataGridViewColumn In DataGridView1.Columns
            xlWorkSheet.Cells(6, col.Index + 1) = col.HeaderText.ToString
        Next
        For i = 1 To DataGridView1.Rows.Count - 1
            For j = 0 To DataGridView1.ColumnCount - 1
                Dim vv As String
                If DataGridView1(j, i - 1).Value Is Nothing Then
                    vv = "Niet ingevuld"
                Else
                    vv = DataGridView1(j, i - 1).Value.ToString

                    xlWorkSheet.Cells(i + 6, j + 1) = vv


                End If


            Next
            ProgressBar1.Value = (i / DataGridView1.Rows.Count) * 100
        Next

        xlWorkBook.Activate()
        xlWorkBook.SaveAs("D:\Consolidate.xls")
        xlWorkBook.Close()
        xlApp.Quit()
        Panel1.Visible = False
        MsgBox("You can find your report at " & "D:\Consolidate.xls")
    End Sub
End Class
```

9.2 Screen Shots:

9.2.1LOGIN:

Figure: 9.2.1-login

9.2.2ADMIN HOME PAGE:

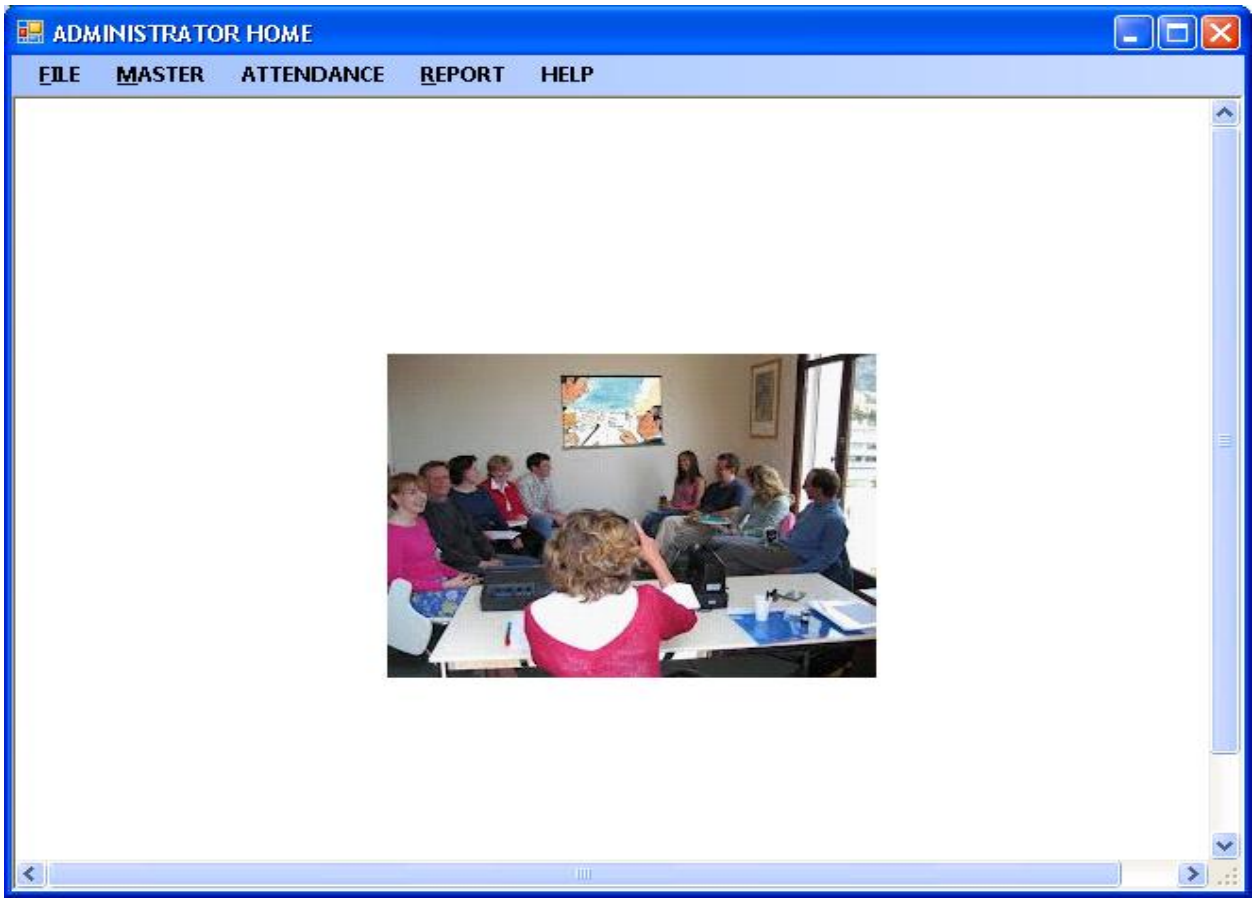


Figure :9.2.2 admin home page

9.2.2.1 STUDENT DETAILS:

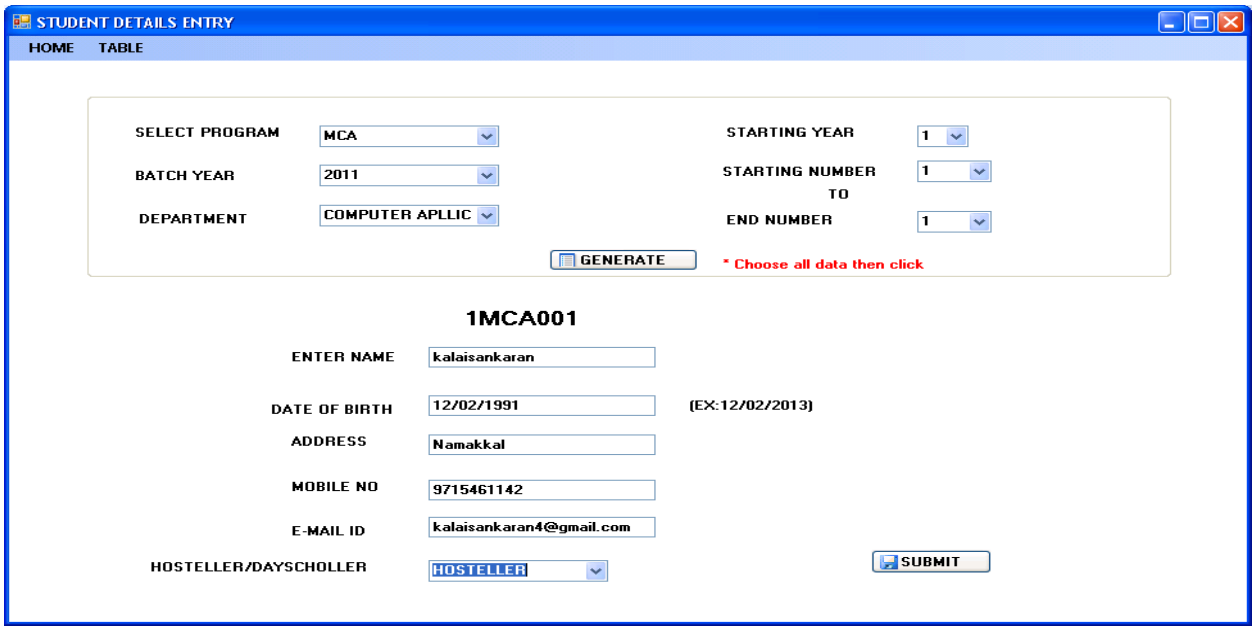


Figure:9.2.2.1 student details Entry

9.2.2.2 Staffs:

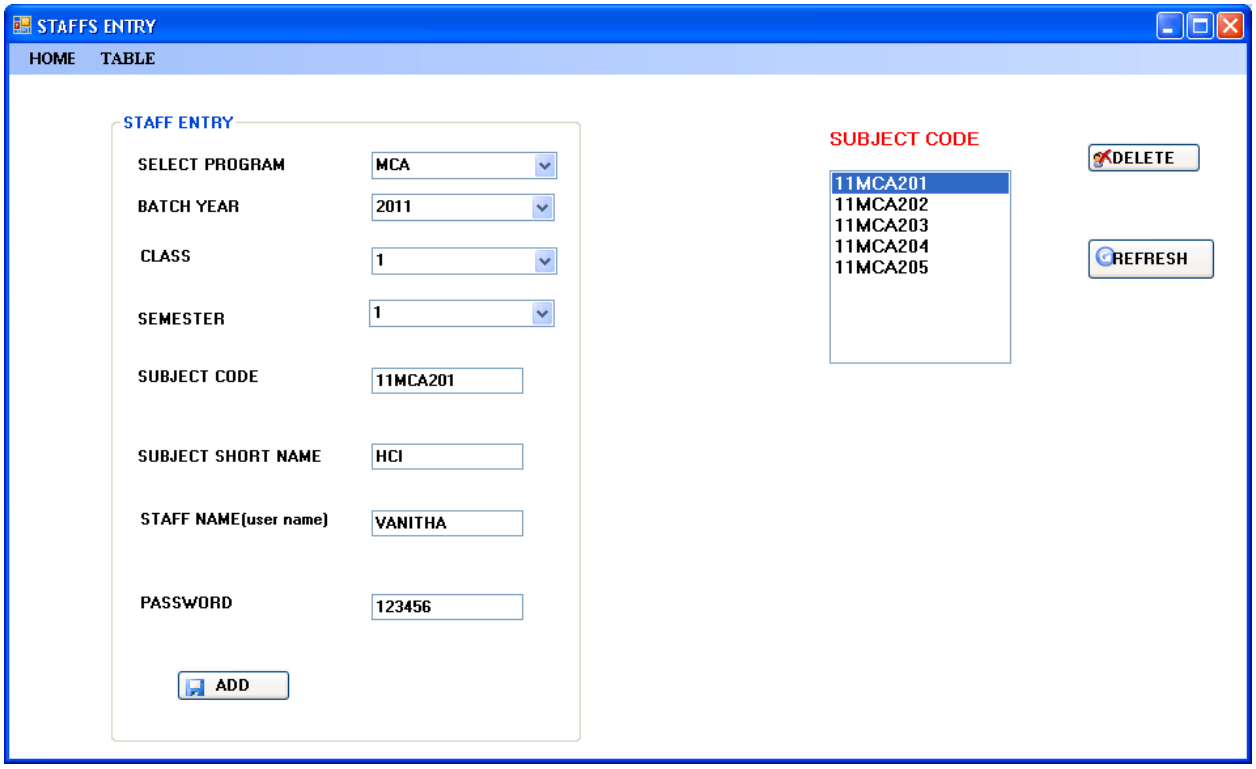


Figure:9.2.2.2 Staffs insertion

9.2.2.3TIME TABLE:

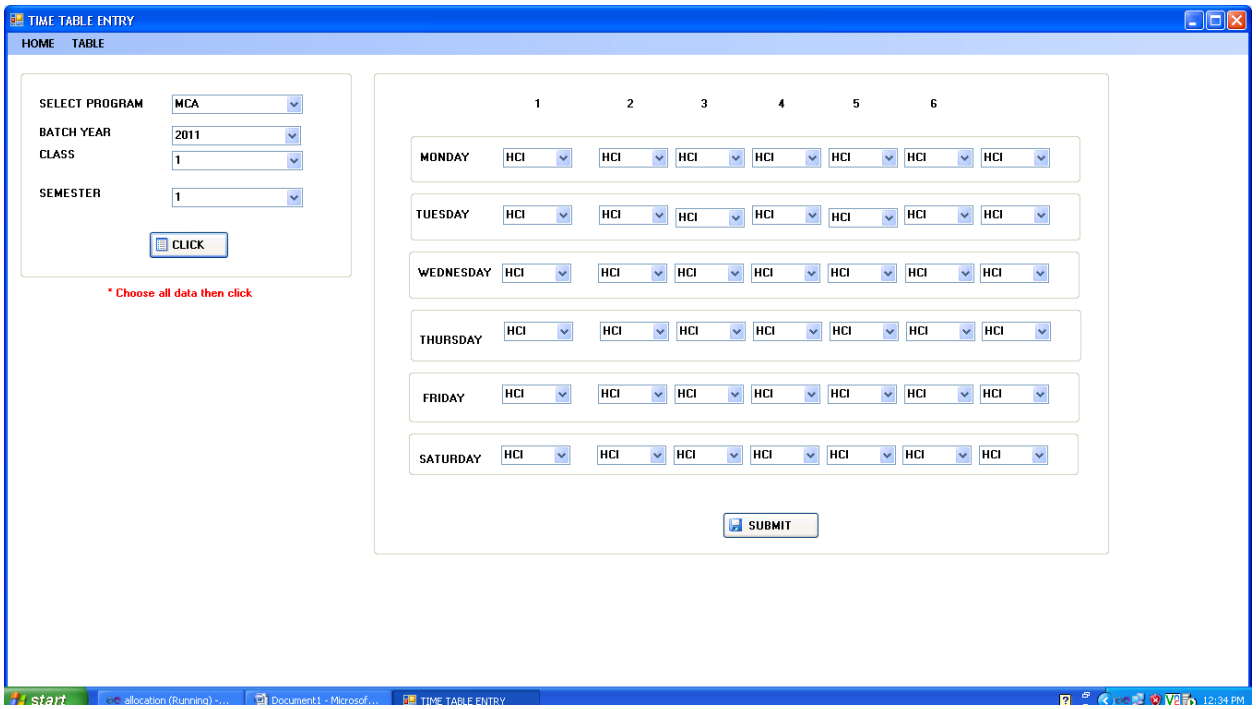
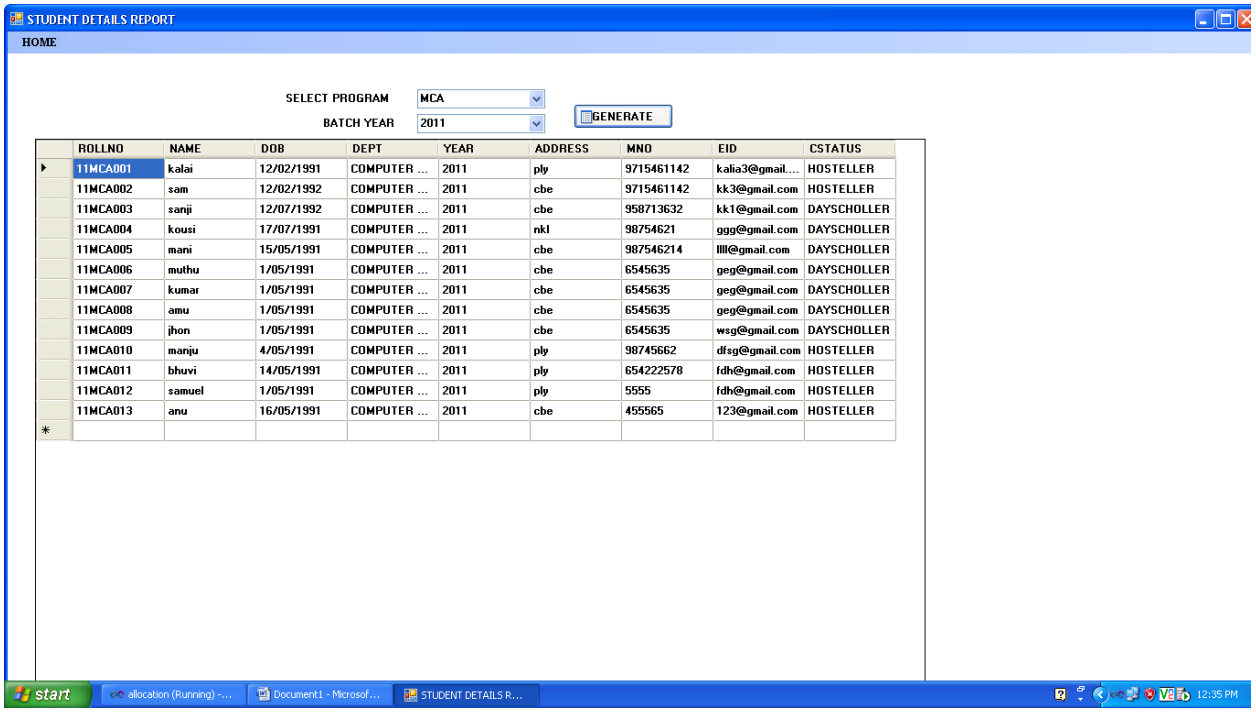


Figure :9.2.2.3 time table

9.2.2.4 VIEW STUDENTS DETAILS:



ROLLNO	NAME	DOB	DEPT	YEAR	ADDRESS	MNO	EID	CSTATUS
11MCA001	kalai	12/02/1991	COMPUTER ...	2011	ply	9715461142	kalia3@gmail....	HOSTELLER
11MCA002	sam	12/02/1992	COMPUTER ...	2011	cbe	9715461142	kk3@gmail.com	HOSTELLER
11MCA003	sanji	12/07/1992	COMPUTER ...	2011	cbe	958713632	kk1@gmail.com	DAYSCHOLLER
11MCA004	kousi	17/07/1991	COMPUTER ...	2011	nkl	98754621	ggg@gmail.com	DAYSCHOLLER
11MCA005	mani	15/05/1991	COMPUTER ...	2011	cbe	987546214	llll@gmail.com	DAYSCHOLLER
11MCA006	muthu	1/05/1991	COMPUTER ...	2011	cbe	6545635	geg@gmail.com	DAYSCHOLLER
11MCA007	kumar	1/05/1991	COMPUTER ...	2011	cbe	6545635	geg@gmail.com	DAYSCHOLLER
11MCA008	amu	1/05/1991	COMPUTER ...	2011	cbe	6545635	geg@gmail.com	DAYSCHOLLER
11MCA009	jhon	1/05/1991	COMPUTER ...	2011	cbe	6545635	wsg@gmail.com	DAYSCHOLLER
11MCA010	manju	4/05/1991	COMPUTER ...	2011	ply	98745662	dfsg@gmail.com	HOSTELLER
11MCA011	bhuvi	14/05/1991	COMPUTER ...	2011	ply	654222578	fdh@gmail.com	HOSTELLER
11MCA012	samuel	1/05/1991	COMPUTER ...	2011	ply	5555	fdh@gmail.com	HOSTELLER
11MCA013	anu	16/05/1991	COMPUTER ...	2011	cbe	455565	123@gmail.com	HOSTELLER

Figure:9.2.2.4 view students details

9.2.2.5Staffs details:

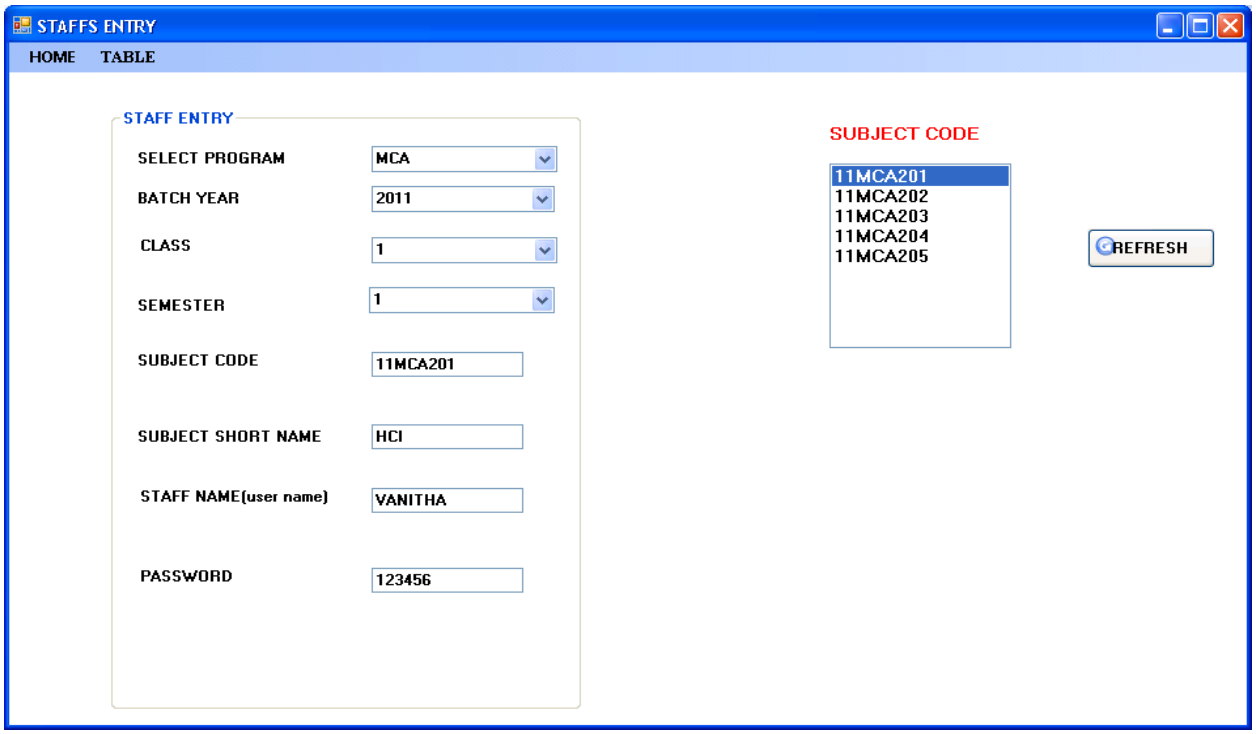
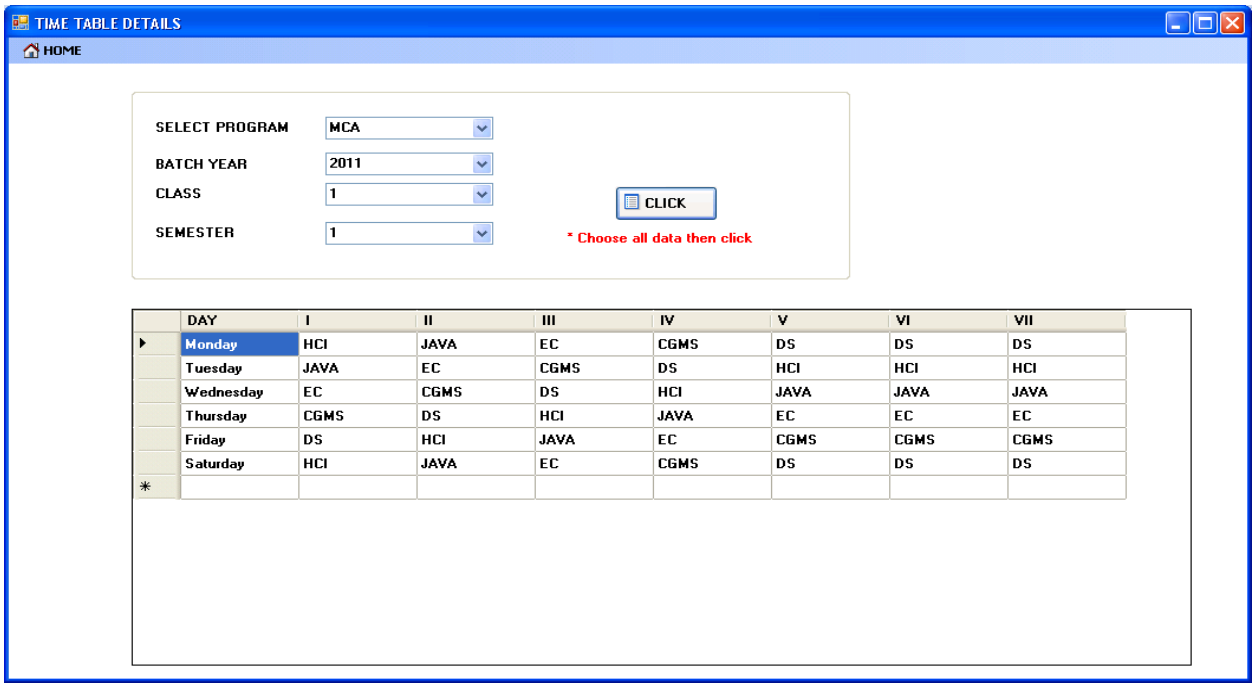


Figure: 9.2.2.5-Staffs details

9.2.2.6TIME TABLE:



DAY	I	II	III	IV	V	VI	VII
Monday	HCI	JAVA	EC	CGMS	DS	DS	DS
Tuesday	JAVA	EC	CGMS	DS	HCI	HCI	HCI
Wednesday	EC	CGMS	DS	HCI	JAVA	JAVA	JAVA
Thursday	CGMS	DS	HCI	JAVA	EC	EC	EC
Friday	DS	HCI	JAVA	EC	CGMS	CGMS	CGMS
Saturday	HCI	JAVA	EC	CGMS	DS	DS	DS

Figure:9.2.2.6 time table details

9.2.2.7UPDATE:

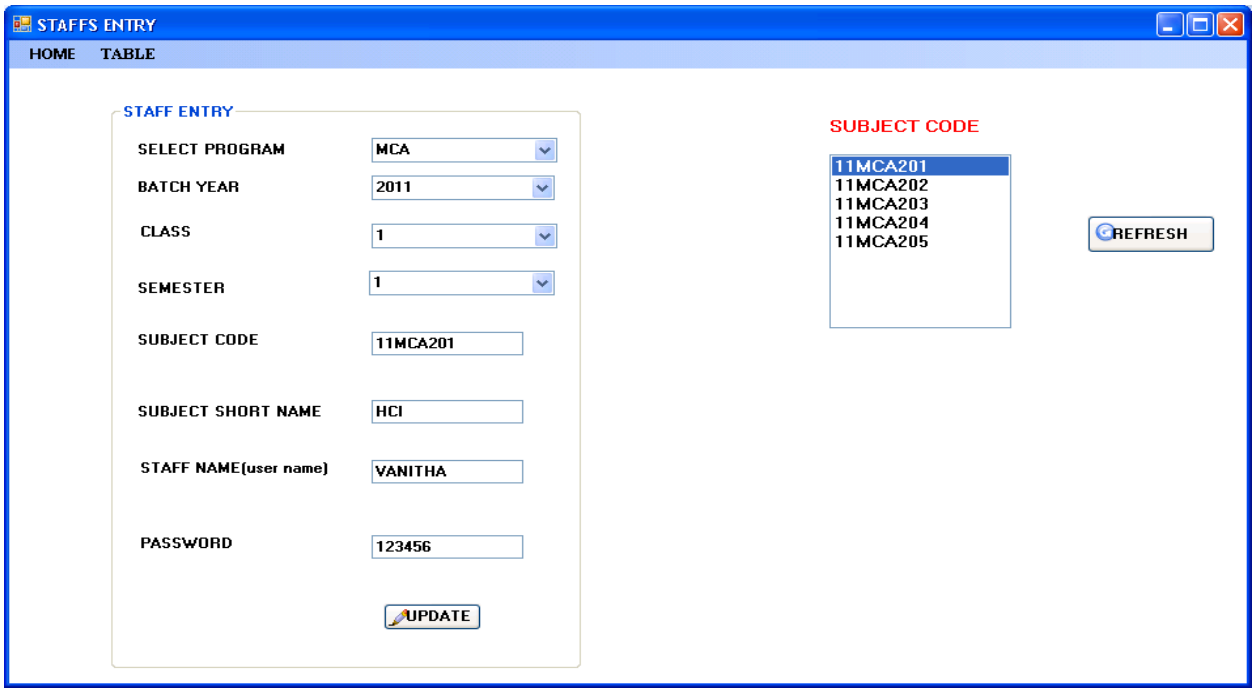


Figure :9.2.2.7 update details

9.2.2.8 STUDENT:

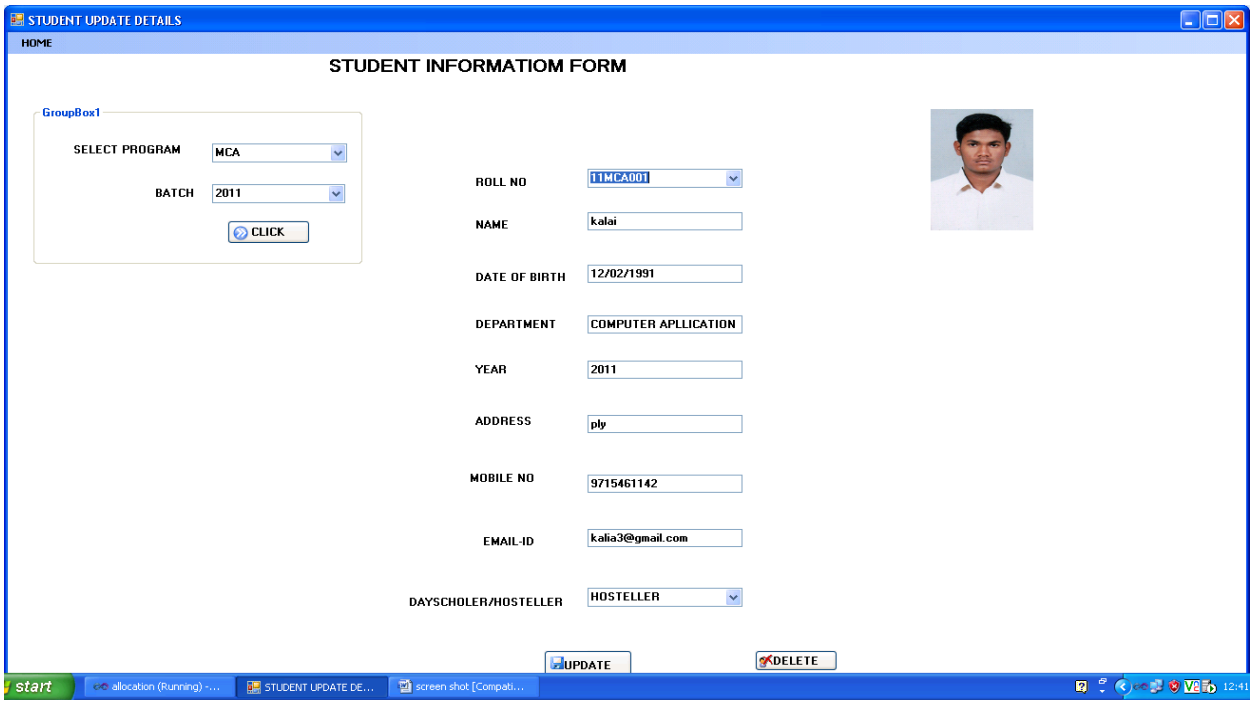


Figure: 9.2.2.8 student details

9.2.2.9 ATTENDANCE ENTRY:

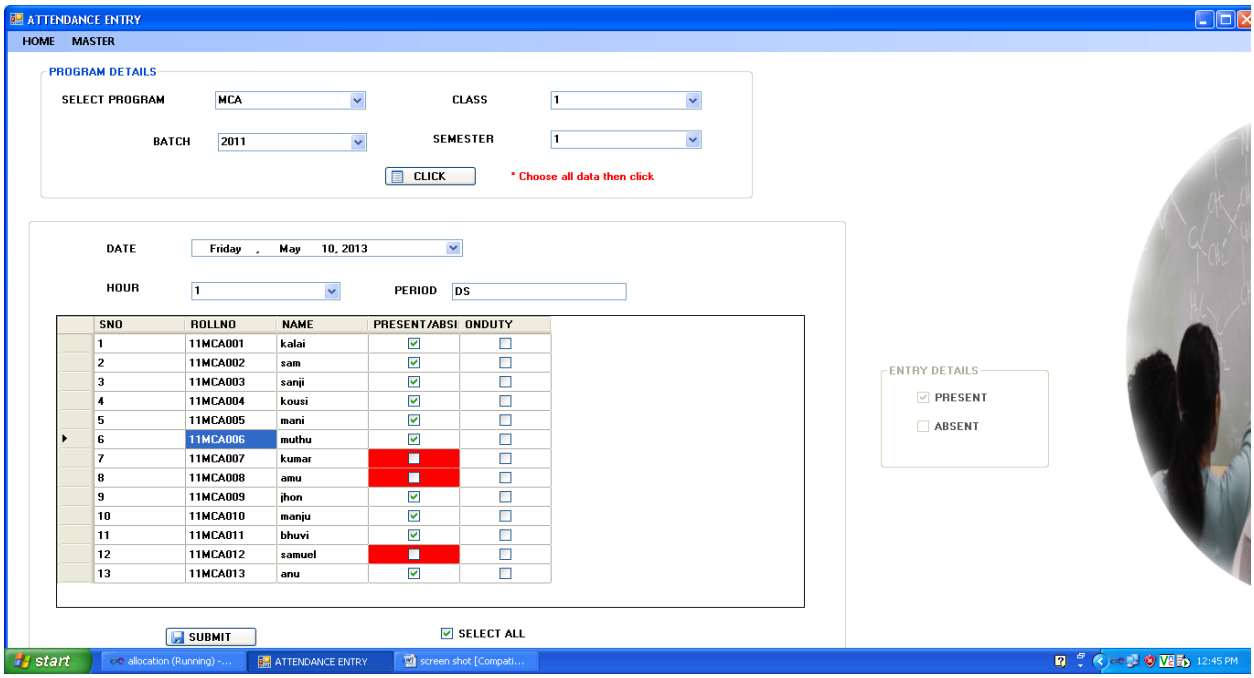


Figure: 9.2.2.9 attendance entry

9.2.3 WEEKLY REPORT:

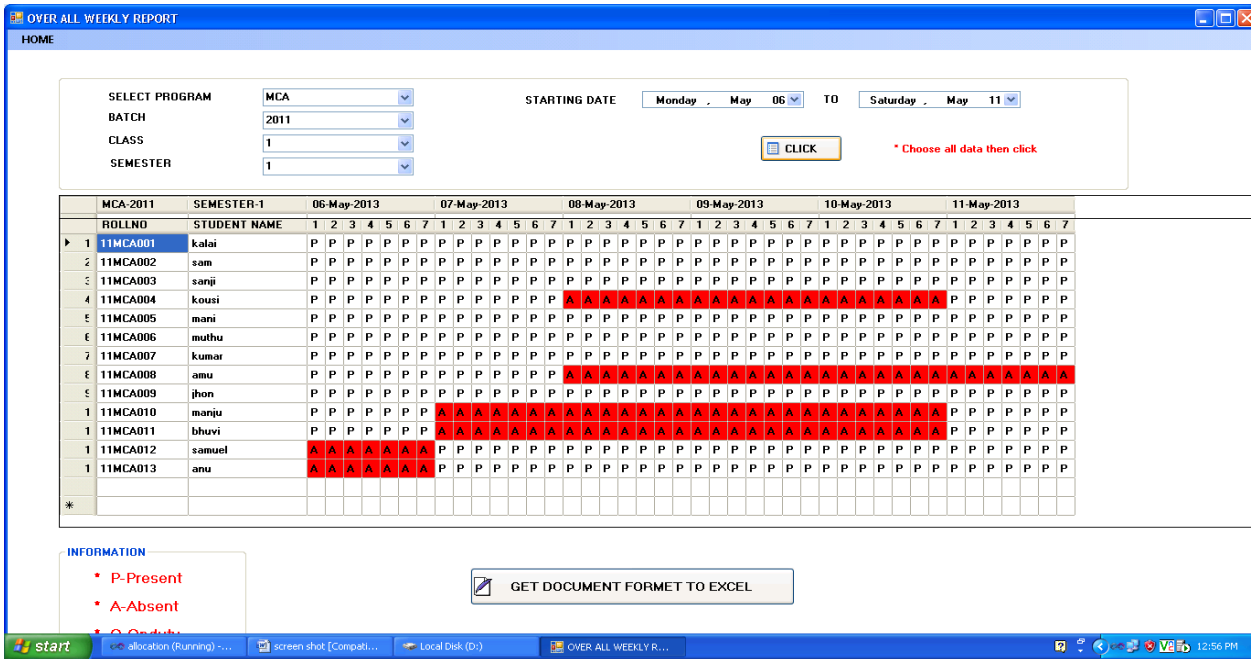


Figure :9.2.3 weekly report

9.2.4 CONSOLIDATE REPORT:

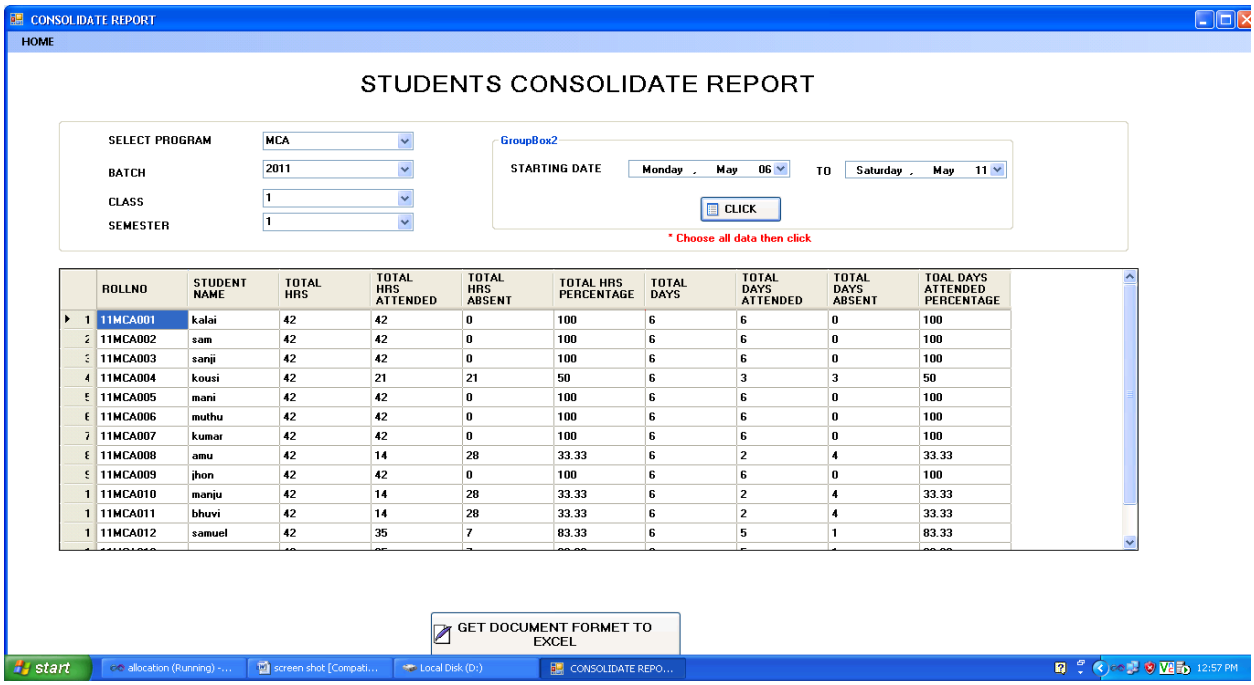


Figure:9.2.4 consolidate report details

9.3 STAFFS LOGIN:

9.3.1DAILY ATTENDANCE:

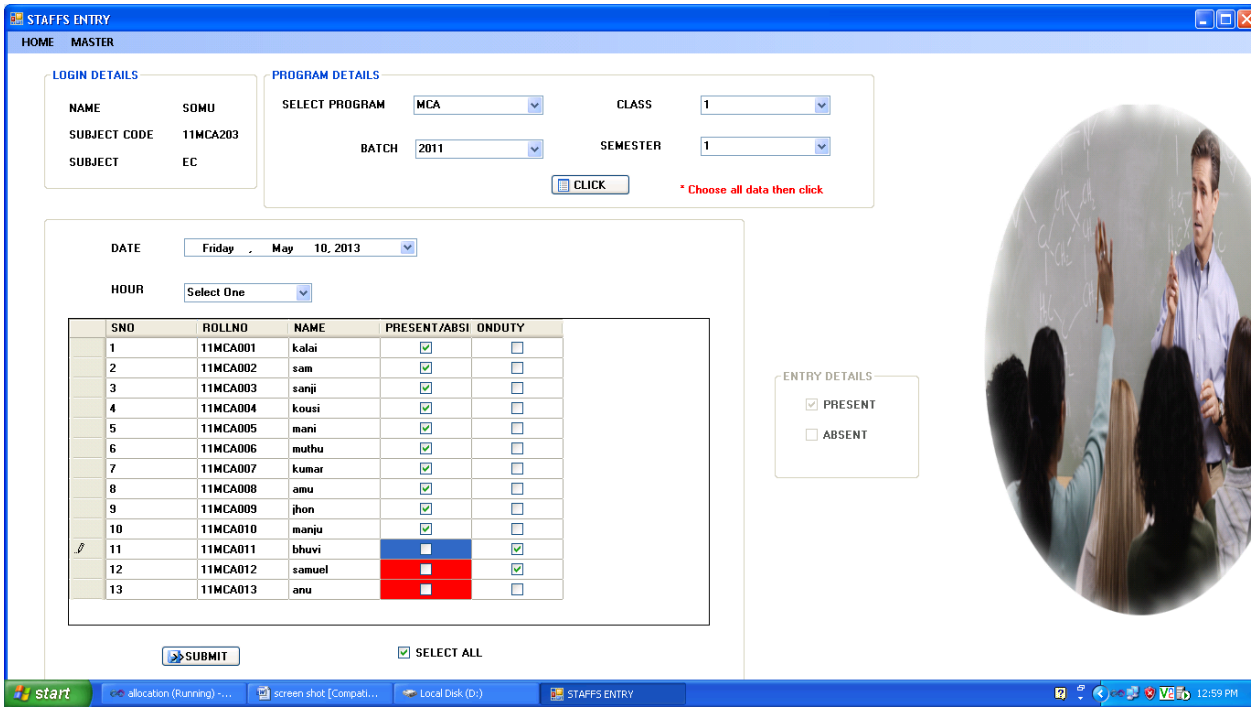


Figure:9.3.1daily attendance

9.3.2 UPDATE ATTENDANCE:

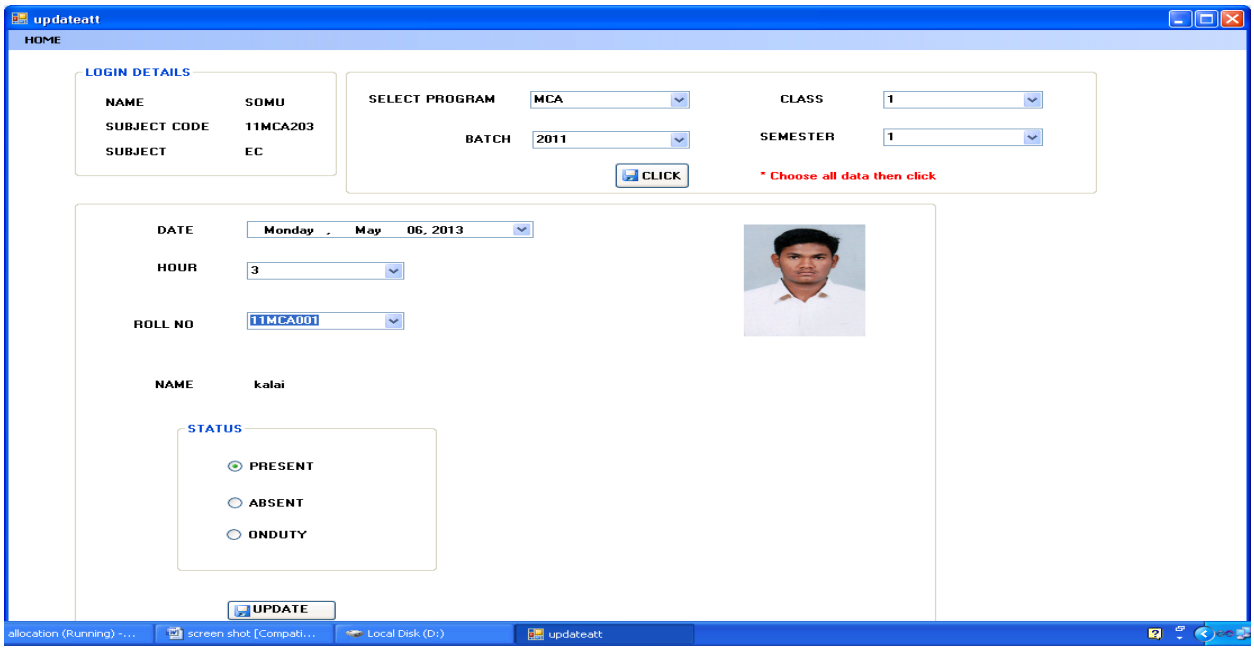


Figure:9.3.1 Update attendance

9.3.3 WEEKLY REPORT:

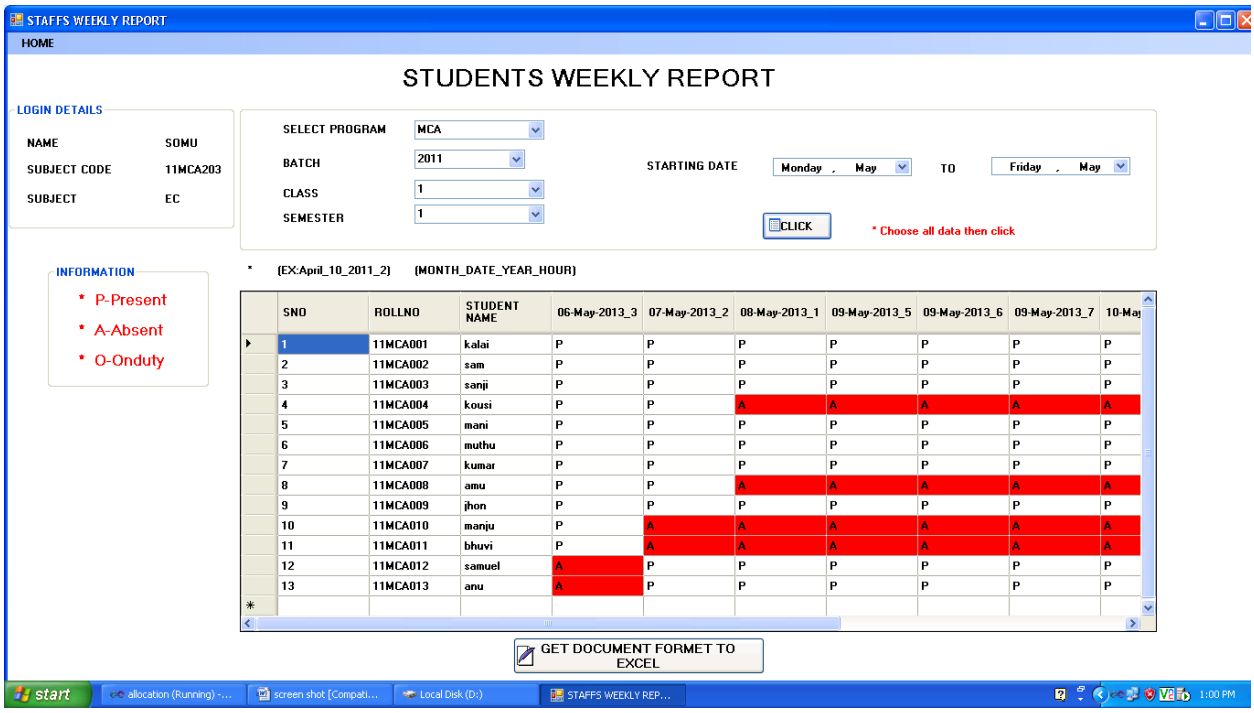


figure :9.3.3 weekly report details

9.3.4 Consolidate:

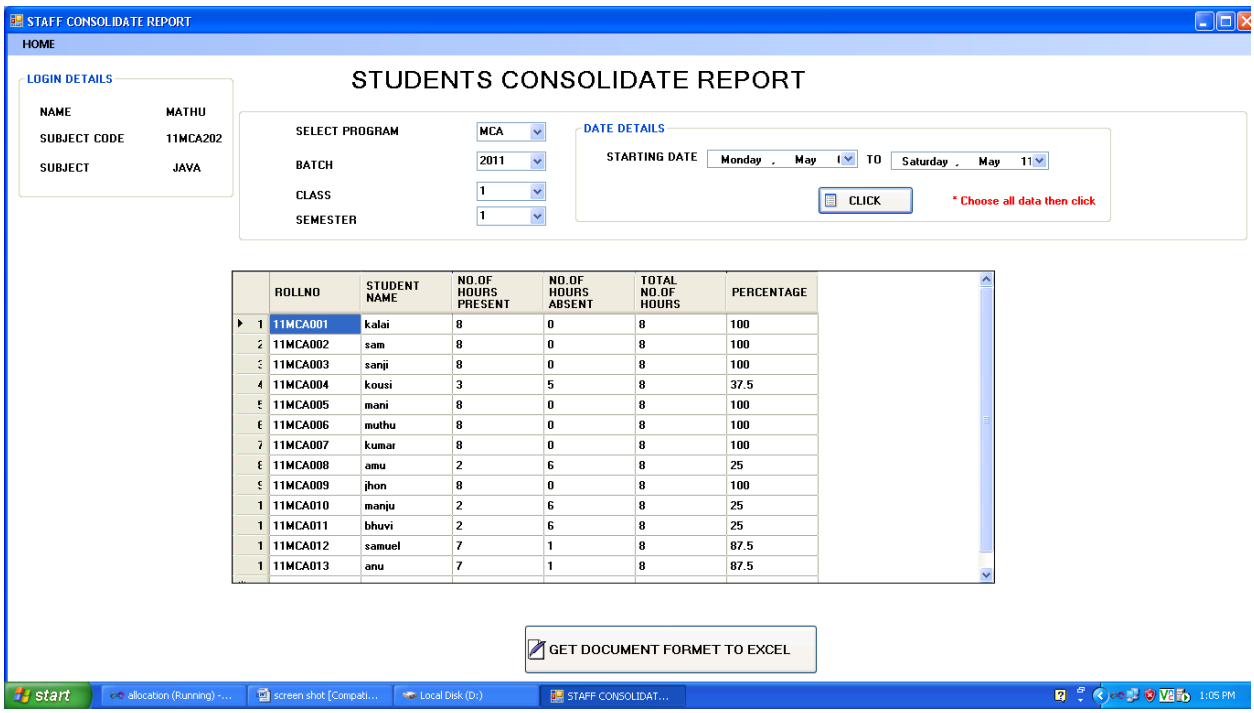


Figure:9.3.4 Consolidate details

CHAPTER 10

REFERENCES

Books References

1. Introducing Microsoft .NET, Second Edition author David S. Platt.
2. Joe Mayo, "**Microsoft Visual Studio 2010: A Beginner's Guide"**, Tata McGraw Hill, 2010.
3. Alex Mackey, **"Introducing .NET 4.0: With Visual Studio 2010"**, Press, USA, 2010.

WEBSITES

1. http://www.msdn.net/
2. http://msdn.microsoft.com/en-us/library/orm-9780596518455-02.aspx
3. http://www.w3schools.com/asp.net/
4. http://www.cramerz.com/aspdotnet
5. http://www.dotnetspider.net/
6. http://www.stackoverflow.com
7. http://www.codeproject.com