



### TITLE OF THE PROJECT

This project dedicated for “**College Management System**” throw Imarat Institute of Computer & Electronics (IICE) which initiated the educational aspects of Students especially for the poor students. This college made possible for the poor students to study easily. This project provide the detailed information about IICE, KRC<sub>N</sub> of ‘Maulana Mazharul Haque Arabic & Persian University’.

College Management Provides the affiliation details of college, details of the staff as well as the students, details of students fees or their billing details and also the payments made in case of the faculty or staff in the form of payroll. It also focuses on the final reports and the result of the students of various courses. So the suggested name of this project “College Management System”.



## **INTRODUCTION OF THE PROJECT**

“College Management System” has been created using two mastermind Language JAVA & ORACLE. The former is being in the foreground and the latter in the background respectively.

The background language is favorite of many programmers because it offers the much wanted structure query facilities. Oracle provides high securities at the database are stored in the logical area called “db Engine”. As db engine is created in the virtual memory and end user cannot access that directly it. Now a day’s Java is very usable programming language because it is purely object Oriented Language and java program’s runs on any machine easily. With these thought in mind “College Management System” has been develop after brain storming the needs and requirements of the College.



## **OBJECTIVE OF THE PROJECT**

### **(1) . CAPABILITY:-**

As the system process large buck of graphical as well as large sum of textual data, the daily activity influenced by increasing the ability to process query, transaction quickly and efficiently. The system adds capability in three ways.

#### **(a) Improved processing speed:-**

The “College Management System” used virtual memory which is supported by java JVM which keep the data intact in memory and in case of virtual memory it provide better management and provide fast processing also DLL to increase inherent speed to process data, graphic files.

#### **(b) Increased volume:-**

The “College Management System” provides many activity and events to avail the increased volume of processing, as data available in main memory and in the virtual memory to keep data intact, and depending on increased volume within processing the data kept in the virtual memory. Thus system avail the increased volume.

#### **(c) Faster retrieval of information:-**

The system provides faster retrieval of information as this is a complex task and the system sported the RDBMS engine of oracle. The front end Java support in this process by bringing the more and more data in main memory and in virtual memory, and swiftly interacting with the Oracle.

### **(2). CONTROL:-**

#### **(A). Greater accuracy and consistency: -**

It showing details, information, graphic require greater volume accuracy and consistency these best supported as the java has math coprocessor and multithreading capability ,also the consistency improved exponentially by using Oracle as a back-end.



### **(B). Better security: -**

The security is important attribute of any software system. This security measures are by authorizing the people which accomplished by three level of authorization. First at user level second at data level and third at administrative level.

### **(3). GRAPHICAL USER INTERFACE:-**

#### **(A). Enhanced communication: -**

Java as a front end provides powerful graphical tools like applets, Servlets, swing, AWT to design an application and Provides facilities of dynamic linking and embedding between web pages. It has many graphical function and classes which facilitates us to design a good informatics application window. The Java is a process oriented and object oriented based paradigm which organizes a program around it data and thus the system achieves several organizational benefit.

#### **(B). Integration: -**

The project “College Management System” built in java platform, all information regarding of processing bring to a single area of virtual memory or in buffer for fast processing in remote site. The system provide common interface through applet which is a standardized tool for integrating.

### **(4). COST: -**

#### **(A). Monitor cost: -**

“College Management System” monitors cost by reducing cost of transaction through intranet, cost of processing of graphic, reducing the paper works and retrieving data from database.

#### **(B). Reduce Cost: -**

“College Management System” S/w developed in platform independent environment thus reducing the cost of adding new architecture , the system also reduces the cost of fetching and processing of file by partially loading the file, thus it is capable of processing data at lower cost.



### **(C). Competitiveness: -**

The “College Management System” designed such a way that it must be flexible, have accuracy, reliable and robust and better documented so further requirement and specifications easily added. Thus it easily faces the newer requirements.



### **INTRODUCTION TO THE ORGANIZATION**

This project dedicated for “College Management System” throw Imarat Institute of Computer & Electronics (IICE) which initiated the educational aspects of Students especially for the poor students. This college made possible for the poor students to study easily. This project provide the detailed information about IICE, KRC<sub>N</sub> of ‘Maulana Mazharul Haque Arabic & Persian University’.

Imarat Institute of Computer & Electronics is among those colleges which provide every facility to students especially for the students which can’t afford the college fees. This project is about all the details related to College Management.



## **PROJECT CATEOGRY**

This project has been developed in “JAVA”. Hence the category of the project comes into GUI (Graphical User Interface) and it is an Object Oriented System. So relational database management system (RDBMS) used in JAVA is a power full programming language to develop sophisticated window very quickly.

Our project is delicately falls in Relational Database Management System (RDBMS) environment; RDBMS is one of the important environments for commercial data. Processing in the world. The oracle enterprise version takes care of the following:

Updating the database

- 1) Retrieving information from the data base
- 2) Accepting query language statement
- 3) Enforcing security specification
- 4) Enforcing data integrity specification
- 5) Enforcing transaction consistency
- 6) Managing data sharing
- 7) Optimizing query
- 8) Managing system catalog



### TOOLS AND PLATFORM

#### FRONT-END

The need of now a day's software development is competence in a web based application front-end to which can connect to relation database engine. This gives the programmer the opportunity to develop client-server based commercial application. This application gave user the power and ease of user of web based application with the multi-user capabilities of auto based RDBMS engine such as oracle.

From the own point of view, **JAVA** is one of Today's most popular software development language. Java is a fully Object-Oriented language with strong support for proper software engineering techniques. Java is used to create web pages with dynamic and interactive content, develop large scale enterprises application, enhance the functionality of web servers and provide applications for consumer devices (e.g. cell phones, pagers and personal digital assistance) and for many other purposes. Java provides many features which are described below:-

- **Simple :-**

Java was designed to be very simple and easy to learn. The syntax of Java has kept nearer to C++ so that the usage of Java does not require extensive training programmes to be undertaken. A programmer aware of the various Object-Oriented concepts can easily develop application in Java.

- **Object-Oriented :-**

Object oriented design is a technique that focuses design on the data and on the interfaces rather the modularization of the functionalities or the tools used to develop them.

- **Robust :-**

The multi-platform environment of the web places high demand on the reliability of the program to execute on a variety of systems. Thus, high priority has been given to create robust and highly reliable programs in the design of Java. It provides extensive compile-time checking followed by a second level of run-time checking. The most common problems in programming language are related to memory management and exception handling.

- **Secure :-**

Java is designed to be used in network and distributed environment where security is of paramount importance. Java supports the creation of applications that cannot be invaded from outside. On the other side, Java programs are executed in their own environment and do not go outside these boundaries unless they are authorized to do so.





- **Interpreted :-**

Java byte code is not directly executed by the system, because Java is interpreted. However, the speed is more than adequate for most interactive application.

- **Multithreaded :-**

Java's multithreading capability provides the mean to build applications with many concurrent threads of activity. The multithreading feature of Java has various sophisticated synchronization primitives. Moreover, Java's high level system libraries have been written to be thread safe, i.e., the functionality provided by the available without conflict to multiple concurrent threads of execution.

- **Dynamic :-**

While the Java compiler is strict in its static checking during compile time, the language and run time systems are dynamic during linking and loading stages. Classes are linked only as needed. New code modules can be linked in on demand from a variety of sources, even from sources across the network in a large number

### **BACK-END**

As we know that the worlds longest selling are the DBMS product. Oracle work groups or enterprise server account major percentage of RDBMS engine sold worldwide. The natural programming language of this product is SQL/PLSQL the creation of user defined function procedure database trigger and other specific database processing PLSQL for commercial application will always related to oracle related to oracle engine.

Oracle can run across different platforms. They can also transfer data from various hardware platforms without any conversions of programs. They thus satisfy the feature of having open architecture and distributed data management capability. This database have industry standard SQL & Report writer which are 4GL tools. This not only facilitates faster development, but also enhances professional productivity. Oracle has a query optimizer. The main function of a query optimizer is to determine automatically the fastest method in

Which a database request can be handled. As a result of this programmers and end users do no need any additional training to obtain good RDBMS performance. These databases have servers or data mangers, which minimize both memory and CPU resource utilization. This ensures high performance during transaction processing.

Oracle has introduced a large number of new features such as analysis queries, scalar sub-queries, joins, pivot insert statement and merger statement. Object oriented feature such as inheritance, Polymorphism, undo table space and various PLSQL packages. It continues to provide the most secure application development and deployment platform in the industry. Exception handling Oracle locating and user defined locking are fully supported. It also provides the key concept such as primary key, unique key and foreign key.



## **HARDWARE / SOFTWARE REQUIREMENTS**

### **MINIMUM HARDWARE REQUIRED**

- Intel (or equivalent) dual core or more
- 256 MB RAM
- 10 GB HARD DISK.
- One printer

### **MINIMUM SOFTWARE REQUIREMENT:**

- O/S – WINDOWS 7 / 8 / 10.
- FRONT END TOOLS-- JAVA
- BACK END TOOLS—ORACLE 10g



## DATA STRUCTURE

### 1. Affiliation

Field	Type	Width	Constraint
Afflno	Varchar	15	Primary key
Afflto	Varchar	30	
Courses	Varchar	30	
Affldt	Date		
Reg	Varchar2	15	

### 2. StaffDetail

Field	Type	Width	Constraint
Eno	Number	20	Primary key
Ename	Varchar2	30	
Address	Varchar2	40	
City	Varchar	30	
State	Varchar	30	
Pin	Number	10	
Phone	Number	15	
Dob	Date		
Dept	Varchar2	40	
Bp	Number	10	
Gender	Varchar	8	
Email	Varchar	40	
Doj	Date		
Mstatus	Varchar	20	
Designation	Varchar	30	



## COLLEGE MANAGEMENT SYSTEM



### 3. StudentDetail

Field	Type	Width	Constraint
Reg	Varchar	15	Primary key
Roll number	Number	15	
Sname	Varchar	30	
Fname	Varchar	30	
Mname	Varchar	30	
Dob	Date		
Gender	Varchar	8	
Ecourse	Varchar	10	
Edate	Date		
Esession	Varchar	20	
Temp	Varchar2	40	
Parm	Varchar2	40	
Contact	Varchar	15	
Email	Varchar	40	
Remarks	Varchar	30	

### 4. StudentFeeDetail

Field	Type	Width	Constraint
Reg	Varchar	15	
Sname	Varchar	30	
Course	Varchar	10	
Cfee	Number	10,2	
Mfee	Number	10,2	
Other	Number	10,2	
Tdue	Number	10,2	
Deposit	Number	10,2	
Ddate	Date		
Remarks	Varchar	30	



## COLLEGE MANAGEMENT SYSTEM



### 5. PayRoll

Field	Type	Width	Constraint
Eno	Number	15	
Ename	Varchar	30	
Month	Varchar	10	
Year	Varchar	10	
Bp	Number	10,2	
Hra	Number	10,2	
Da	Number	10,2	
Ma	Number	10,2	
Ta	Number	10,2	
Pf	Number	10,2	
Esi	Number	10,2	
Advance	Number	10,2	
Lic	Number	10,2	
Other	Number	10,2	
Nsalary	Number	10,2	
Remarks	Varchar	30	

### 6. BillingMaster

Field	Type	Width	Constraint
Stdreg	Varchar	20	
Stdroll	Number	20	
Stdname	Varchar	30	
Session	Varchar	20	
Coursefee	Number	10,2	
Examfee	Number	10,2	
Adminfee	Number	10,2	
Labfee	Number	10,2	
Libfee	Number	10,2	
Miscfee	Number	10,2	
Others	Number	10,2	
Total	Number	10,2	



## COLLEGE MANAGEMENT SYSTEM



### 7. BbaResult

Field	Type	Width	Constraint
Reg	Varchar	15	
Name	Varchar	30	
Course	Varchar	15	
Semester	Varchar	8	
Paper1	Number	8	
Paper2	Number	8	
Paper3	Number	8	
Paper4	Number	8	
Paper5	Number	8	
Paper6	Number	8	
Pract1	Number	8	
Pract2	Number	8	
Percentage	Number	8	
Grade	Varchar	8	

### 8. BcaResult

Field	Type	Width	Constraint
Reg	Varchar	15	
Name	Varchar	30	
Course	Varchar	15	
Semester	Varchar	8	
Paper1	Number	8	
Paper2	Number	8	
Paper3	Number	8	
Paper4	Number	8	
Paper5	Number	8	
Paper6	Number	8	
Pract1	Number	8	
Pract2	Number	8	
Percentage	Number	8	
Grade	Varchar	8	



# INPUT TO THE PROJECT

## 1. AFFILIATION:-

Affiliation is an important module which stores the information about the affiliation date, the university it is affiliated to, the courses college offers, registration, is maintained. In this module we use the primary key as affiliation number.

## 2. STAFF DETAIL:-

Staff Detail is also an important module which stores the information about the staff of the colleges whether it be the employees, faculty or other staff, employee Number (or ID number), city, state, contact number, email (if any), DOJ, DOB etc is maintained. In this module we use the primary key as employee number.

## 3. STUDENT DETAIL:-

Student Detail is also an important module which stores the information about the students studying in the college like their names, registration number, roll number, DOB, Gender etc. In this module we use primary key as Registration number.

## 4. STUDENT FEE DETAIL:-

Student Fee Detail is also a module which keeps the information about the fees of the students whether it be main course fees or miscellaneous charges or any other fees. In this module the Students registration number is used to view the fee details which was used as a primary key in Student Detail.

## 5. PAYROLL:-

Payroll is an important module as it stores the salaries of the employees, faculty and other staff members. Salary can be viewed using the employee number which was used as a primary key in Staff Detail. Here employer's salary is mentioned.

## 6. BILLING MASTER:-

Billing Master is a module where we see student's final fees. It can be called as a final fee slip with all kinds of fees provided to students.



## COLLEGE MANAGEMENT SYSTEM



### 7. BBA RESULT:-

BBA Result is an important module. In this module the marks of all the subjects of the course of the students are mentioned which they scored in the Examination.

### 8. BCA RESULT:-

BBA Result is an important module. In this module the marks of all the subjects of the course of the students are mentioned which they scored in the Examination





# OUTPUT TO THE PROJECT

The output to the project means that what is required to “College Management System” simply the desired data by organization. The desired data may be any type .It may be different from one one to another, but some important report may be required to all. These are following.

- Affiliation details
- Staff information
- Students information
- Students fee details
- Payroll information.
- Billing details of the students
- BBA results information
- BCA results information

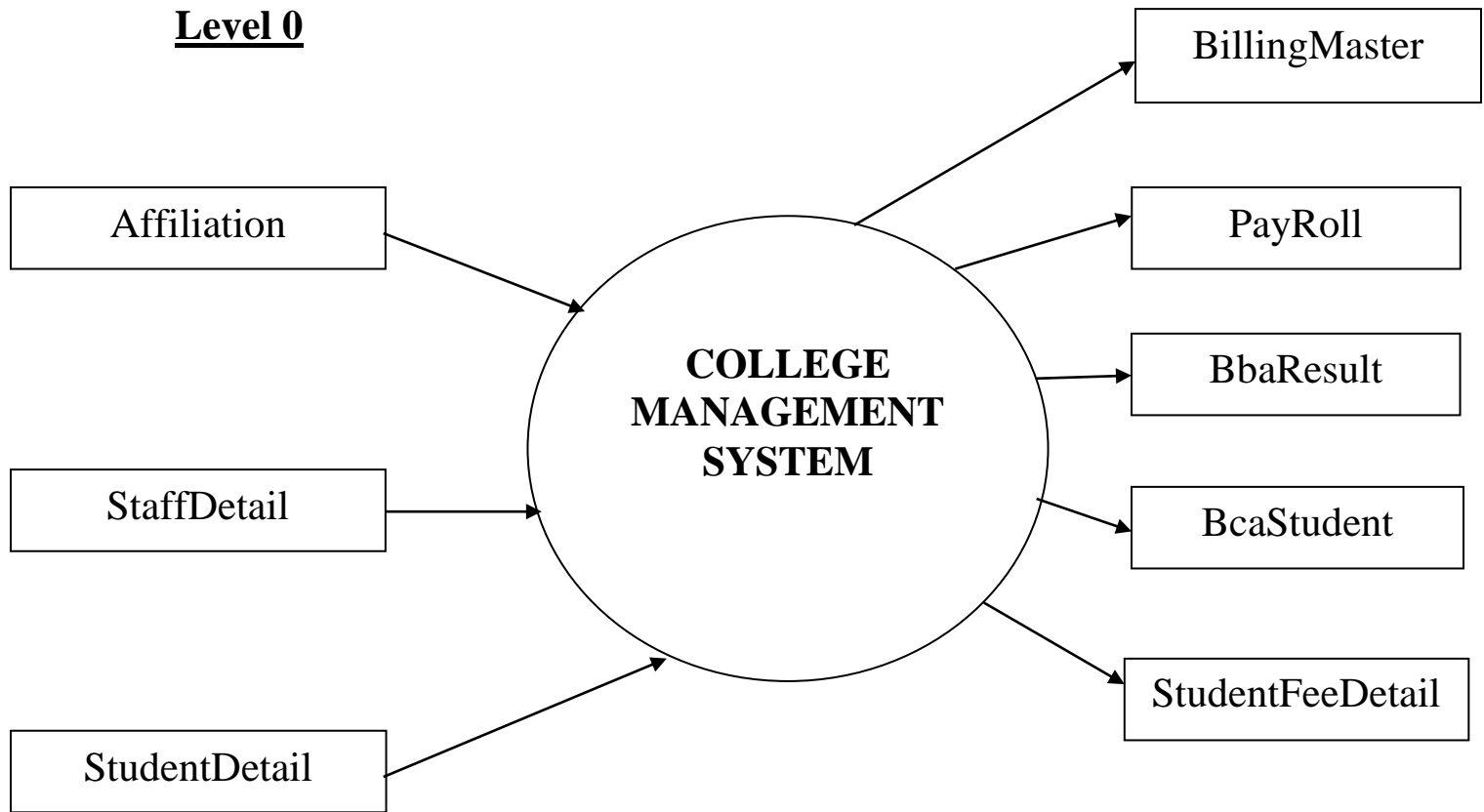
**So we can say finally they consist in the following modules.**

1. AFFILIATION
2. STAFF DETAIL
3. STUDENT DETAIL
4. STUDENT FEE DETAIL
5. PAYROLL
6. BILLING MASTER
7. BBA RESULT
8. BCA RESULT



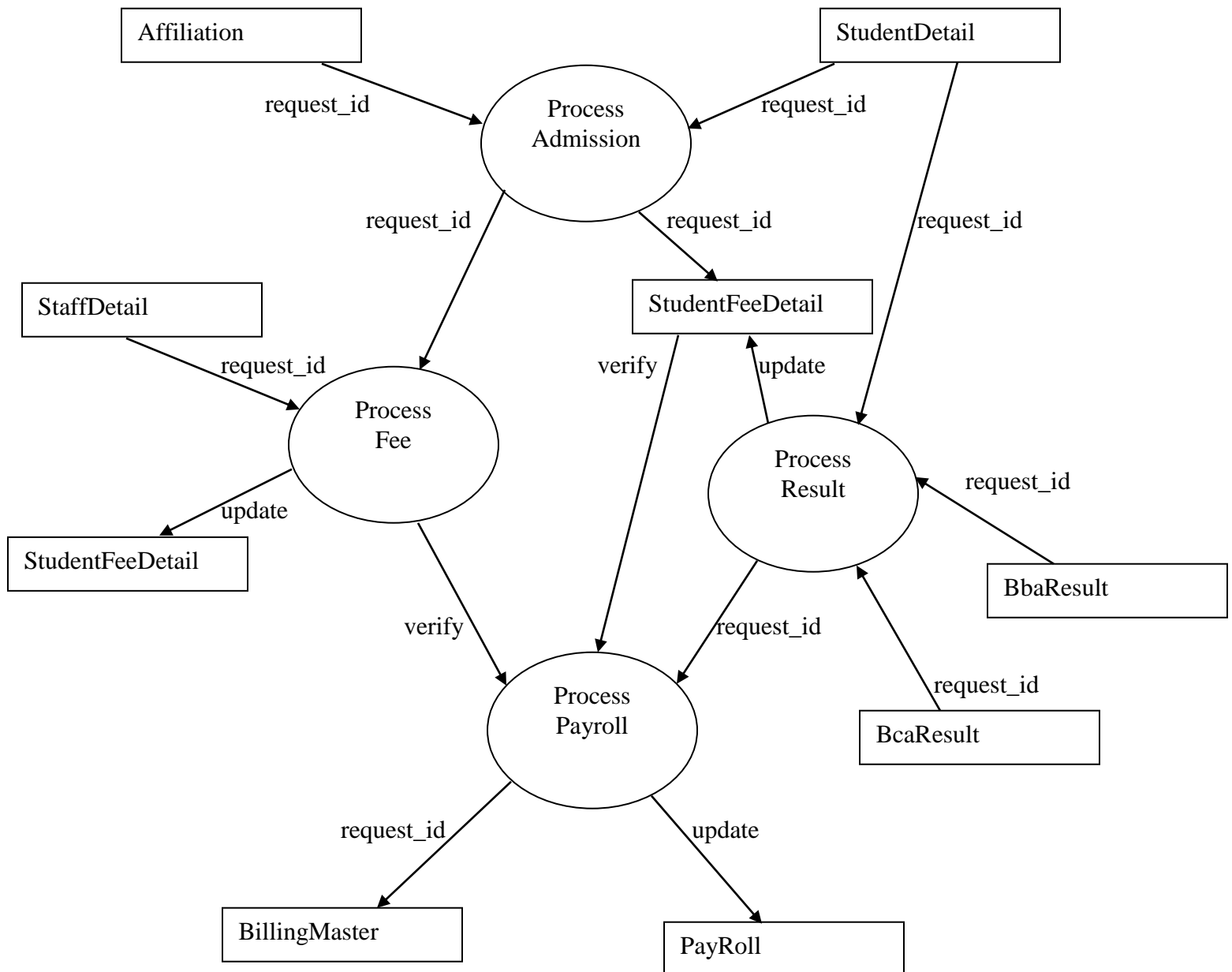
## DATA- FLOW DIAGRAM

### Level 0



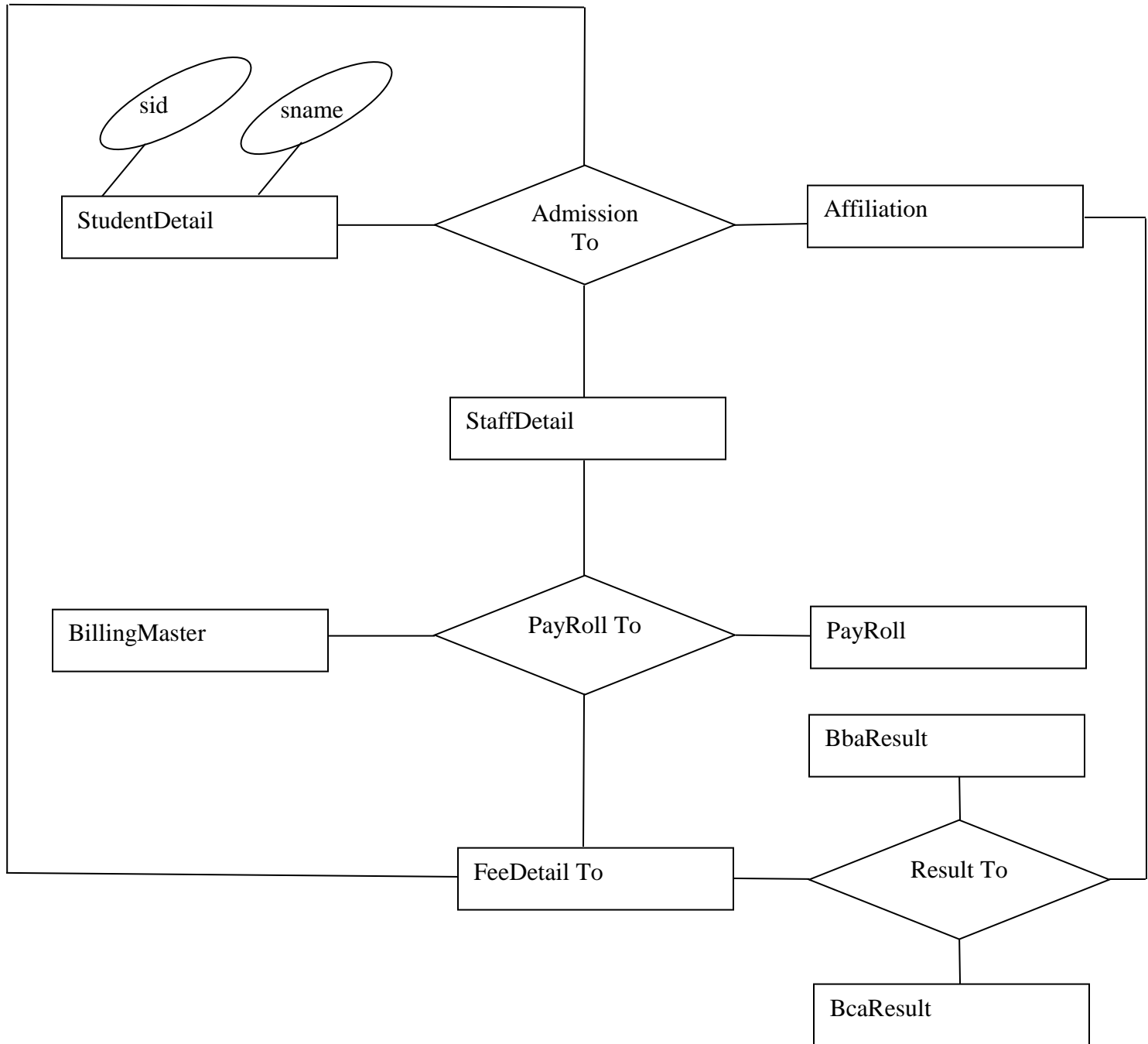


## Level 1





## ER – DIAGRAM





### **REPORT GENERATION**

Report system is used to retrieve records according to the requirements of the organization. Report system contains several types of reports for current and future requirements in order to keep query purpose in this system. There are many type of report generated such as students fee report, students result report etc. All these report are helpful for college in order to proceed. There is a provision to list the individual status of each person with association of its given period of time. It also consists of the following modules, they are:

- Affiliation details
- Staff information
- Students information
- Students fee details
- Payroll information.
- Billing details of the students
- BBA results information
- BCA results information



## **SECURITY MECHANISM**

In the project we are using 'oracle RDBMS' as our back-end and "JAVA" as our front-end.

The realm of oracle security is very broad and

It is needless to say that we can't deal each and every Security features of oracle here!

Nevertheless we can roughly make a distinction of the security features which is provided by oracle co. And can be applied in the project:

- ✓ Internal database security
- ✓ Application security
- ✓ External database security

### **Internal security security:**

There are several steps to implement internal security.

Some of them to implement on database level are:

- A. Create user account and distribute appropriate privilege as needed to approve people by authority of approved people.
- B. Establish and protect the dba privileged account and password.
- C. Ensure that the user accounts are all password protected.
- D. When appropriate, use oracle's auditing facilities to monitor table/user activities.
- E. Write down own set of auditing procedures if oracle's are not granular enough.
- F. Develop, document and maintain change control procedures to prevent unauthorized

### **Application security:**

One of the most common security problem areas is the application code that the developers and third party vendors write. Poor programming invariably lead to inadvertent security Holes. In these one of the most common is:

Buffer overflow – which let the hackers to damage the system and easily creep into the system. to over come this problem among several technique virtual private database (vpd)has got the preference .to overcome such type of problem this mechanism(vpd) or another may be applied in this project.

In addition, to prevent user from accessing unauthorized data virtual private database (vpd) mechanism may be applied

Data in invalid format – it is a common area where developers always make a mistake .in the project major consideration will be paid on this section.



## COLLEGE MANAGEMENT SYSTEM



### **External database security:**

Since the project is not network based project, the Features of oracle's external database security are not much relevant. So this section is not applicable to our Project i.e. We will not work on these features.

So we can say that by using oracle's various security features we can ensure The security of our application to a greater extent. By doing so, not only we can keep secure our database from unauthorized access but also ensure the reliability Of the Software against frequent failure.



### **VALIDATION CHECKS**

The purpose of validation check is to check the validity of data entered into the system, this is for check that proper and correct data with proper format is entered thus minimum error is provided by the system. Since security is of at most important for any project, so this project is dedicated to solve problem of security. This project is dedicated to solve the internal restriction of the databases for the security reasons. The validation check is accomplished through the use of primary key, foreign key and not null constraint.

The key concepts of primary key, foreign key, null values are frequently used in this project. The check constraint is frequently used to accept proper formatted data like data without blank, formatted date; Capitalize data etc, tools like sequence is used to proper sequencing of the data in the table. The use of primary key in the database helps to avoid the redundancy of data. This is one of the unique facilities provided by DBMS. The use of foreign is used to establish relationship. Thus proper and valid data is accepted by the system.





### **FUTURE SCOPE AND ENHANCEMENT OF THE PROJECT**

This system is designed by looking near future of organization to manage the system of college, thus further this system is implemented to

This project is developed by keeping in mind the proper rules and regulation with standards of the structure of data to manage the tour and travel in Bihar, thus system is implemented in states and for country also without any big change in the system either in structure, numbers of table or increasing attributes in the tables.

Thus system fulfills most future requirements with lesser amount of change, either to manage increased volume of data, load time, manual power to maintain the increase processing etc. This system also runs in any environment or any newer architecture or software (o/s).

The system developed with limited domain but the newer criteria are added without any big changes. The system cooperates fully to increase the domain. In future this application may be expanded for the web enabled information or work on web enabled network, code with little modification worked on the designer according the requirements.



## **LIMITATION OF THE PROJECT**

In the program a lot of thing could be included but due to the storage of time and several other reasons, these are not included in the program some of these possibilities are.

- This software does not maintain the backup copy.
- There no recovery facility; means if any data loses then one can not recover that lost data.
- There is no any audit trail facility; there for one can not check found data is right or wrong.
- ODBC drivers must have installed or oracle 10g must be installed to run the project properly.
- It is not dedicated to any organization.
- It is purely intranet based.
- It support on line back up system but don't
- Support off line back up system.
- The maximum number of record in a table is 50,000
  
- Errors generated in project are limited. We used this project for named exception, not for number exception.



## **TESTING PROCESS**

There are a lot of possibilities, which could be included in the program, but due to the shortage of time and several other reasons, these are not included in the program.

Some of these possibilities are:

- This software doesn't maintain the backup copy parallel.
- There is no recovery facility; means, if any data loses then one can't recover that lost data any way.
- There is no audit-trail facility; therefore one can't check fended data is right or wrong.
- ODBC drivers must have installed or Oracle 10g

Must be installed to run the project properly.

- It is not dedicated to any organization.
- It is purely INTRANET based.
- It supports on-line backup system but don't support Off-line backup system.
- The maximum number of record in a table is 50,000.

Errors generated in project are limited



# INPUT FORM



## LOGIN FORM

COLLEGE MANAGEMENT SYSTEM

**COLLEGE MANAGEMENT SYSTEM**

**Login Form**

Refresh

USER NAME :

PASSWORD :

Login Cancel

Please Fill The Password Correct  
You Have Maximum Three (3) Chances To Try

## MENU PAGES







## AFFILIATION FORM

WELCOME IN COLLEGE MANAGEMENT SYSTEM

EntryForm Report Tools Help Exit

## WELCOME IN COLLEGE MANAGEMENT SYSTEM

### AFFILIATION FORM

### WELCOME SCREEN FOR AFFILIATION

Enter information for AFFILIATION

**AFFILIATION NO.**  **NEW**

**AFFILIATED TO**  **INSERT**

**COURSE OFFERED** ☐ BCA ☐ BBA ☐ BSc ☐ BLMC **UPDATE**

**AFFILIATION DATE**    **VIEW**

**REGISTERED NO.**  **DELETE**

**LOAD**

Affiliation no.	Affiliated to	Course	Affiliation date	Registration no.

Certificates and Approvals

Accreditation & Affiliation

## STAFF DETAIL FORM

The screenshot displays a web application titled 'WELCOME IN COLLEGE MANAGEMENT SYSTEM'. The main content area is a 'WELCOME SCREEN FOR STAFF DETAIL' form for a staff member named Wasim Haider. The form includes fields for EMP NUMBER (126), EMP NAME (Wasim Haider), ADDRESS (phulwari Sharif), CITY (Patna), STATE (Bihar), PIN CODE (801505), PHONE NO. (7631897927), DATE OF BIRTH (09 Feb 1992), DEPARTMENT (Office assi), BEST PAY (100000), GENDER (Male), EMAIL (wasim19@gmail.com), DATE OF JOINING (01 Jan 2011), MARITAL STATUS (Unmarried), and DESIGNATION (BCA Faculty). Action buttons include NEW, INSERT, UPDATE, VIEW, DELETE, and LOAD. A table below the form lists staff details with columns: slno, surname, name, city, state, pin, phone, dob, mar, up, gender, email, sta, marital, and designat. The table contains 5 rows of data.

slno	surname	name	city	state	pin	phone	dob	mar	up	gender	email	sta	marital	designat
123	Azama...	Phulw...	Patna	Bihar	801505	99319...	15-Jan...	Other	100000	Male	akr15...	12-ma...	Unmar...	BCA...
124	Altam...	shubh...	Patna	Bihar	801505	88777...	09-Feb...	Office...	90000	Male	altama...	01-Jan...	Unmar...	Office...
125	Mid E...	shubh...	Patna	Bihar	801505	88775...	09-Feb...	Teacher	70000	Male	ektam...	01-Jan...	Unmar...	Lab In...
126	Wasim...	shubh...	Patna	Bihar	801505	76318...	09-Feb...	Office...	100000	Male	wasim...	01-Jan...	Unmar...	BCA...
127	Quazi...	shubh...	Patna	Bihar	801505	99736...	09-Feb...	Teacher	45000	Male	quazi...	01-Jan...	Unmar...	Lab In...
129	Iran...	shubh...	Patna	Bihar	801505	78777...	11-Jun...	Office...	90000	Male	altama...	02-Apr...	Unmar...	Office...

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj



## COLLEGE MANAGEMENT SYSTEM



### STUDENT DETAIL FORM

WELCOME IN COLLEGE MANAGEMENT SYSTEM

Entry Form Report Tools Help Exit

### WELCOME IN COLLEGE MANAGEMENT SYSTEM

STUDENT DETAIL FORM

### WELCOME SCREEN FOR STUDENT DETAIL

REG NUMBER:  TEMP ADDRESS:

ROLL NUMBER:

STUDENT NAME:

FATHER NAME:

MOTHER NAME:

DATE OF BIRTH: DD MM YYYY

GENDER: ☐ Male ☐ Female

ENROLL COURSE:  CONTACT:

ENROLL DATE: DD MM YYYY

ENROLL SESSION:  EMAIL:

REMARKS:

NEW INSERT UPDATE VIEW DELETE LOAD

Reg no.	Roll	Student	Father's	Mother's	Date of b.	Gender	Enroll co.	Enroll date	Enroll se.	Tempora	Permiana	Contact	Email	Remarks
16201	12122	Quazi	meha...	Quazi	10-ma	Male	BCA	09-jul	2011-2	Sham	Daud	99736...	quazi1	Good
16201	12122	Ampia	Shashi	Kiran	15-jan	Male	BCA	14-jul	2011-2	harun	sikroll	99319...	akr26	good s
16201	12122	Altam	Md Az	NDefa	09-feb	Male	BCA	06-jul	2011-2	phub	Schaw	88777...	altam	Good
16201	12122	Wasi	Md Y	Anjua	04-jan	Male	BCA	10-jul	2011-2	Fedral	Ara	76318...	wasim	Good
16201	12122	Ibran	kamra	Noor j	02-jan	Male	BCA	23-aug	2011-2	patna	phub	94723...	ibran	good s

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj

### STUDENT FEE DETAIL FORM

WELCOME IN COLLEGE MANAGEMENT SYSTEM

Entry Form Report Tools Help Exit

STUDENT FEE DETAIL FORM

### WELCOME SCREEN FOR STUDENT FEE DETAIL

Enter Information for Student

REG NUMBER: 16201100635 OTHER: 500 NEW

STUDENT NAME: Wasim kaidar TOTAL DUES: 1000 INSERT

COURSE: BCA DEPOSITE FEE: 4350 UPDATE

COURSE FEE:  DATE: 01 Jan 2013 VIEW

MISC FEE:  PAID: paid DELETE

Message: Record Reported OK

Reg no.	Student name	Course	Deposite fee	Misc fee	Other	Total dues	Deposite fee	Deposite date	Remarks
162011001	Qazi md ar	BCA	4500	1000	500	10000	4250	03-feb-2013	paid
162011001	Qazi md ar	BCA	4500	1000	500	10000	4250	03-feb-2013	paid
162011001	Qazi md ar	BCA	4500	1000	500	10000	4250	03-feb-2013	paid
162011001	Qazi md ar	BCA	4500	1000	500	10000	4250	03-feb-2013	paid
162011006	Altamash	BCA	4500	1000	500	10000	4250	03-feb-2013	paid
162011006	Altamash	BCA	4500	1000	500	10000	4250	03-feb-2013	paid

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj





## PAYROLL FORM

WELCOME IN COLLEGE MANAGEMENT SYSTEM  
Entry Form Report Tools Help Exit

**WELCOME IN COLLEGE**

**WELCOME SCREEN FOR PAYROLL DETAIL**

Enter information for Employee Payroll

Employee number: 126 TA: 10000 NEW  
Employee name: Wasim Haidar PF: 1050 INSERT  
Month: May ESI: 500 UPDATE  
Year: 2014 Message: 20000 VIEW  
Basic pay: 9000 Records Updated! 3000 DELETE  
HRA: 500  
DA: 600  
MA: 3000 Remarks: paid

LOAD

EMP NO	EMP NAME	MONTH	YEAR	BASIC PAY	TEN	PF	ESI	TA	PF	ESI	ADVANCE	RS	OT	TOTAL	REMARKS
126	Wasim	Feb	2014	9000	5000	6000	3000	10000	1050	500	20000	3000	12000	148450	paid

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj

## BILLING MASTER FORM

WELCOME IN COLLEGE MANAGEMENT SYSTEM  
Entry Form Report Tools Help Exit

**WELCOME IN COLLEGE MANAGEMENT**

**WELCOME SCREEN FOR STUDENT BILL DETAIL**

Enter information for Student bill

REG NUMBER: 16201100635 ADMIN FEE: NEW  
ROLL NUMBER: LAB FEE: INSERT  
STUDENT NAME: Input: 16201100635 UPDATE  
SESSION: ? Input Registration number to be view VIEW  
COURSE FEE: OK Cancel DELETE  
EXAM FEE: TOTAL: LOAD

Reg no	Roll	Student no	Session	Course fee	Exam fee	Admin fee	Lab fee	Lib fee	Mat fee	Other	Total
1620110...	1212202	Asamam	2011-2014	8500	1050	1300	200	300	1000	6000	18350
1620110...	1212202	Altamash	2011-2014	8500	1050	1300	200	300	1000	6000	18350
1620110...	1212202	Wasim h.	2011-2014	8500	1050	1300	200	300	1000	6000	18350
1620110...	1212202	Asamam	2011-2014	8500	1050	1300	200	300	1000	6000	18350
1620110...	1212202	Altamash	2011-2014	8500	1050	1300	200	300	1000	6000	18350
1620110...	1212202	Wasim h.	2011-2014	8500	1050	1300	200	300	1000	6000	18350

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj





## COLLEGE MANAGEMENT SYSTEM



### BCA RESULT FORM

WELCOME IN COLLEGE MANAGEMENT SYSTEM

EntryForm Report Tools Help Exit

**WELCOME SCREEN FOR BCA RESULT**

REG NUMBER: 16201100124

STUDENT NAME:

COURSE: SELECT

SEMESTER: SELECT SEMESTER

PAPER 1:

PAPER 2:

PAPER 3:

PAPER 4:

PAPER 5:

PAPER 6:

PRACTICAL 1:

PRACTICAL 2:

PERCENTAGE:

GRADE: SELECT GRADE

NEW

INSERT

UPDATE

VIEW

DELETE

LOAD

Reg.no.	Student.n.	Course	Semester	Paper1	Paper2	Paper3	Paper4	Paper5	Paper6	Practical1	Practical2	Percentage	Grade
162011...	Amra...	BCA	VI	98	96	89	88	85	90	95	96	92	A1
162011...	Almas...	BCA	VI	90	95	85	87	88	92	95	94	91	A1
162011...	Waxim...	BCA	VI	98	94	89	88	85	92	95	96	92	A1
162011...	Ekram...	BCA	VI	98	94	89	88	85	92	95	96	92	A1

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj

### BBA RESULT FORM

WELCOME IN COLLEGE MANAGEMENT SYSTEM

EntryForm Report Tools Help Exit

**WELCOME SCREEN FOR BBA RESULT**

REG NUMBER: 16201100635

STUDENT NAME: Waxim kaider

COURSE: BBA

SEMESTER: IV

PAPER 1: 97

PAPER 2: 93

PAPER 3: 97

PAPER 4: 94

PAPER 5: 98

PAPER 6: 89

PRACTICAL 1: 82

PRACTICAL 2: 98

PERCENTAGE: 94

GRADE: A1

NEW

INSERT

UPDATE

VIEW

DELETE

LOAD

Reg.no.	Student.n.	Course	Semester	Paper1	Paper2	Paper3	Paper4	Paper5	Paper6	Practical1	Practical2	Percentage	Grade
---------	------------	--------	----------	--------	--------	--------	--------	--------	--------	------------	------------	------------	-------

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj



# REPORT FORMS



## COLLEGE MANAGEMENT SYSTEM



### REPORT OF AFFILIATION

WELCOME IN COLLEGE MANAGEMENT SYSTEM

EntryForm Report Tools Help Exit

REPORT

Report of Affiliation Print Close

Affiliation no	Assigned to	Course	Affiliation date	Registered no
125	MMHAPU	BCA	21-jul-1972	16201100124
124	MMHAPU	BCA	19-jun-1972	16201100635
122	MMHAPU	BCA	03-sep-1972	16201100090
123	MMHAPU	BCA	26-sep-2000	16201100633
127	MMHAPU	BCA	09-mar-1993	16201100119
126	MMHAPU	BCA BBA BSc	26-may-1990	16201100118
125	MMHAPU	BCA	21-jul-1972	16201100124
124	MMHAPU	BCA	19-jun-1972	16201100635
122	MMHAPU	BCA	03-sep-1972	16201100090
123	MMHAPU	BCA	26-sep-2000	16201100633
127	MMHAPU	BCA	09-mar-1993	16201100119
126	MMHAPU	BCA BBA BSc	26-may-1990	16201100118

There are 12 Record Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj

### REPORT OF STAFF DETAIL

WELCOME IN COLLEGE MANAGEMENT SYSTEM

EntryForm Report Tools Help Exit

REPORT

Report of Staff Detail Print Close

id	aname	add	city	state	pin	phone	dob	dept	bp	gender	email	daj	status	signature
126	Arupam K.	phulwari	Patna	Bihar	801505	99219094	15-jan-19	Office asst	100000	Male	arupamk	01-jan-20	Unmarried	Office Asst.
125	Wasim H.	phulwari	Patna	Bihar	801505	78219879	09-feb-19	Office asst	100000	Male	wasimh	01-jan-20	Unmarried	BCA Facu.
124	Atamash	phulwari	Patna	Bihar	801505	98777972	09-feb-19	Office asst	90000	Male	atamash	01-jan-20	Unmarried	Office Asst.
123	Md Shyam	phulwari	Patna	Bihar	801505	98775155	09-feb-19	Teacher	75000	Male	shyamult	01-jan-20	Unmarried	Lab Instru.
127	Quaid Md	phulwari	Patna	Bihar	801505	987798193	09-feb-19	Teacher	45000	Male	quaidmd	01-jan-20	Unmarried	Lab Instru.

There are 5 Record Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj





## COLLEGE MANAGEMENT SYSTEM



### REPORT OF STUDENT DETAIL

WELCOME IN COLLEGE MANAGEMENT SYSTEM

Entry Form Report Tools Help Exit

**WELCOME IN COLLEGE MANAGEMENT SYSTEM**

**Report of Student Detail** Print Close

Reg.no.	Roll	Student name	Father's name	Mother's name	Date of birth	Gender	Enroll course	Enroll date	Enroll sem.	Temporary	Permanent	Contact no.	Email	Remarks
16201100	12122028	Altamash	Md Azad A.	Noorul Ha.	04-Feb-19	Male	BCA	06-Jul-2013	2013-2014	Shuhwan	Suhwan	80777972	altamash	Good stu.
16201100	12122028	Wasim H.	Md Tanveer	Anqum Ara	04-Jan-19	Male	BCA	10-Jul-2013	2013-2014	Fatma Ca.	Ara	76319879	wasim19	Good stu.
16201100	12122028	Qasid Md.	Mohamm.	Qudusia	10-Mar-19	Male	BCA	09-Jul-2013	2013-2014	Shameer	Qasid Hq.	99726193	qasid100	Good stu.
16201100	12122028	Anqum k.	Shahidk.	Kiran Dwi	15-Jan-19	Male	BCA	15-Jul-2013	2013-2014	hassan co.	shroff lak.	99319894	shr20900	good stud
16201100	12122028	Ibran ahm.	hassan a.	Noorjahan	02-Jan-19	Male	BCA	23-Aug-2	2013-2014	patna	shuhwan s.	94723540	ibran@col	good stud

There are 5 Record Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj

### REPORT OF STUDENT FEE DETAIL

WELCOME IN COLLEGE MANAGEMENT SYSTEM

Entry Form Report Tools Help Exit

**WELCOME IN COLLEGE MANAGEMENT SYSTEM**

**Report of Student Fee Detail** Print Close

Reg.no.	Student name	Course	Course fee	Misc fee	Other	Total dues	Deposit fee	Deposit date	Remarks
16201100131	Anqum kumar	BCA	8500	1000	500	10000	4250	01-Jan-2013	paid
16201100124	Qasid md arham	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100633	Altamash Azad	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100635	Wasim haidar	BCA	8500	1000	500	10000	4250	01-Jan-2014	paid
16201100131	Anqum kumar	BCA	8500	1000	500	10000	4250	01-Jan-2013	paid
16201100124	Qasid md arham	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100633	Altamash Azad	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100635	Wasim haidar	BCA	8500	1000	500	10000	4250	01-Jan-2014	paid

There are 8 Record Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj



## COLLEGE MANAGEMENT SYSTEM



### REPORT OF PAYROLL

WELCOME IN COLLEGE MANAGEMENT SYSTEM

EntryForm Report Tools Help Exit

WELCOME IN

REPORT

**Report of Student Fee Detail** Print Close

Reg no	Student name	Course	Course fee	Misc fee	Other	Total dues	Deposited fee	Deposited date	Remarks
16201100131	Anupam kumar	BCA	8500	1000	500	10000	4350	01-Jan-2013	paid
16201100124	Geetika aham	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100633	Akramash Azad	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100635	Wasim khalid	BCA	8500	1500	500	10500	4000	01-Jan-2014	paid
16201100131	Anupam kumar	BCA	8500	1000	500	10000	4350	01-Jan-2013	paid
16201100124	Geetika aham	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100633	Akramash Azad	BCA	8500	1000	500	10000	4250	03-Feb-2013	paid
16201100635	Wasim khalid	BCA	8500	1500	500	10500	4000	01-Jan-2014	paid

There are 8 Records Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj

### REPORT OF BILLING MASTER

WELCOME IN COLLEGE MANAGEMENT SYSTEM

EntryForm Report Tools Help Exit

WELCOME IN COLLEGE MANAGEMENT SYSTEM

REPORT

**Report of Billing Master** Print Close

Reg no	Roll	Student name	Session	Course fee	Exam fee	Admission fee	Lab fee	Lib fee	Misc fee	Others	Total
16201100131	1212202001	Anupam kumar	2011-2014	8500	1050	1300	250	300	1000	0000	18350
16201100633	1212202001	Akramash azad	2011-2014	8500	1050	1300	250	300	1000	0000	18350
16201100635	1212202001	Wasim khalid	2011-2014	8500	1050	1300	250	300	1000	0000	18350
16201100131	1212202001	Anupam kumar	2011-2014	8500	1050	1300	250	300	1000	0000	18350
16201100633	1212202001	Akramash azad	2011-2014	8500	1050	1300	250	300	1000	0000	18350
16201100635	1212202001	Wasim khalid	2011-2014	8500	1050	1300	250	300	1000	0000	18350

There are 6 Records Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj



## COLLEGE MANAGEMENT SYSTEM



# REPORT OF BCA RESULT

WELCOME IN COLLEGE MANAGEMENT SYSTEM

Entry Form Report Tools Help Exit

**WELCOME IN COLLEGE**

**Report of BCA Result** Print Close

Reg no	Student no	Course	Semester	Paper1	Paper2	Paper3	Paper4	Paper5	Paper6	Practical1	Practical2	Percentage	Grade
16201101	Anupam ku.	BCA	V	88	95	88	88	85	90	95	88	92	A1
16201102	Wasim hai.	BCA	V	88	94	88	88	85	92	95	98	92	A1
16201103	Ekratul H.	BCA	W	88	94	88	88	95	92	95	98	92	A1
16201104	Atmash az.	BCA	V	88	95	85	87	88	92	95	94	91	A1

There are 4 Record Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj

# REPORT OF BBA RESULT

WELCOME IN COLLEGE MANAGEMENT SYSTEM

Entry Form Report Tools Help Exit

**WELCOME IN COLLEGE MANAGEMENT SYSTEM**

**Report of BBA Result** Print Close

Reg no	Student no	Course	Semester	Paper1	Paper2	Paper3	Paper4	Paper5	Paper6	Practical1	Practical2	Percentage	Grade
16201101	Gaz mid ar.	BBA	V	96	93	88	95	97	88	80	90	92	A1
16201102	Atmash az.	BBA	V	97	93	97	84	98	88	82	88	94	A1
16201103	Wasim hai.	BBA	IV	97	93	97	84	98	88	82	88	94	A1
16201104	Anupam ku.	BBA	IV	97	93	97	84	98	88	82	88	94	A1
16201105	Gaz mid ar.	BBA	V	96	93	88	95	97	88	80	90	92	A1
16201106	Atmash az.	BBA	V	97	93	97	84	98	88	82	88	94	A1
16201107	Wasim hai.	BBA	IV	97	93	97	84	98	88	82	88	94	A1
16201108	Anupam ku.	BBA	IV	97	93	97	84	98	88	82	88	94	A1

There are 8 Record Found

Developed by : Amairul , Arif , Mahtab , Tauqueer , Motahhir , Sanoj





# CODING



## COLLEGE MANAGEMENT SYSTEM



### **/\* Welcome Screen \*/**

```
import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;
import java.awt.event.*;
public class WelcomeScreen extends JWindow implements ActionListener
{
    private JLabel title,message,pic,lb1;
    private JProgressBar pbar;
    private JTextArea area,developer;
    private JButton b1;
    Timer tm;
    int i=0,x=0,c=0;

    public WelcomeScreen()
    {
        Container cn = getContentPane();
        cn.setLayout(null);
        cn.setBackground(new Color(126,3,6));

        tm = new Timer(450,this);
        tm.start();
        i = 0; x = 0; c = 0;
        JPanel pn = new JPanel();
        title = new JLabel("WELCOME TO COLLEGE MANAGEMENT
SYSTEM");
        title.setFont(new Font("Monotype corsiva",Font.BOLD,25));
        title.setBounds(10,2,900,20);
        pn.setBounds(0,2,900,245);
        pn.setLayout(null);

        b1 = new JButton("Quick Start");
        b1.setBounds(800,0,100,25);
        b1.addActionListener(new quickListener());
        pn.add(b1);

        pn.setBackground(new Color(168,79,255));
        pn.add(title);
        JLabel tmp = new JLabel(new ImageIcon("welcome.jpg"));
        tmp.setBounds(0,0,900,245);
        pn.add(tmp);

        developer = new JTextArea(4,4);
```





## COLLEGE MANAGEMENT SYSTEM



developer.setText("DEVELOPED BY:- Md. Amairul , Md. Arif , Md  
Mahtab , Md. Tauqueer , Motahhir , Sanoj \nPLEASE SEND FEEDBACK  
ABOUT THE PROJECT TO : \n EMAIL :-

collegemanagementsystem@gmail.com");

developer.setBounds(550,410,400,100);

developer.setBackground(new Color(126,3,6));

developer.setForeground(Color.green);

pic = new JLabel();

pic.setIcon(new ImageIcon("lm.jpg"));

pic.setBounds(550,260,500,150);

message = new JLabel();

message.setForeground( Color.white);

message.setBounds(10,495,250,25);

lb1 = new JLabel();

lb1.setForeground(Color.white);

lb1.setBounds(10,380,250,105);

lb1.setFont(new Font("Bell Mt",Font.BOLD,25));

pbar = new JProgressBar();

pbar.setBounds(10,470,870,25);

pbar.setBackground(Color.white);

pbar.setForeground( Color.green);

area=new JTextArea();

area.setText("College Management System has been created \n with  
the help of two languages known as java and oracle.\n College Managment System  
is used for \n capturing storing, analyzing and managing data and associated  
attributes. \n It is a system based of computer system. \n It is Basically use to  
analyze the college, staff application form status ,\n student information,payroll  
managment, student bill status, etc.. ");

area.setBounds(10,260,520,150);

area.setBackground(new Color(126,3,6));

area.setForeground(Color.green);

area.setFont(new Font("Times of New Roman",Font.BOLD,14));

cn.add(pn);

cn.add(area);

cn.add(message);

cn.add(pic);

cn.add(pbar);

cn.add(devloper);

cn.add(lb1);



## COLLEGE MANAGEMENT SYSTEM



```
pack();
setBounds(250,120,900,530);
setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
    x = x+50;
    if(x>=90)
    {
        //i+=1;
        i=i+1;
        pbar.setValue(c+1);
        //lb1.setText(String.valueOf(pbar.getValue()+"%"));
        lb1.setText(String.valueOf(c+"%"));
        x=0;
        c++;
    }
    switch (c)
    {
        case 1: message.setText("Please Wait While
installing Component.....");break;
        case 9: message.setText("Mounting
Database.....");break;
        case 18: message.setText("Checking
Connection.....");break;
        case 27: message.setText("Extracting Neccessory
File.....");break;
        case 36: message.setText("Checking For Space
Nedded.....");break;
        case 45: message.setText("Creating Backup of
File.....");break;
        case 54: message.setText("Installing
Modules.....");break;
        case 63: message.setText("Verifying
Data.....");break;
        case 72: message.setText("Loading
Component.....");break;
        case 81: message.setText("Loading
Module.....");break;
        case 85: message.setText("Finishing
Installation.....");break;
    }
    if(i>=100)
    {
        tm.stop();
        dispose();
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        new Login();
    }
}
public class quickListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            tm.stop();
            dispose();
            new Login();
        }
        catch(Exception e1)
        {
        }
    }
}
public static void main(String args[])
{
    new WelcomeScreen();
}
}
```



## COLLEGE MANAGEMENT SYSTEM



### **/\* Login \*/**

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;

public class Login extends JFrame implements ActionListener
{
    private JLabel lblusrnm, lblpasswd, header, title;
    private JTextArea footer;
    private TextField txtpasswd;
    private JButton login, cancel, refresh;
    private Choice usnm;
    private int c, x;
    private Connection conn;
    private Statement stat;
    private ResultSet rs;
    private String str1, str2;
    private Container cn;

    public Login()
    {
        setTitle("COLLEGE MANAGEMENT SYSTEM");
        cn=getContentPane();
        cn.setLayout(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        cn.setBackground(new Color(125, 1, 5));

        Timer tm=new Timer(100,new tmListener());
        tm.start();
        title=new JLabel("COLLEGE MANAGEMENT SYSTEM");
        title.setBounds(0,0,300,30);
        title.setFont(new Font("Monotype
cursiva",Font.ITALIC+Font.BOLD,15));
        title.setForeground(Color.black);

        x=0;
        JLabel img=new JLabel(new ImageIcon("login.jpg"));
```



## COLLEGE MANAGEMENT SYSTEM



```
img.setBounds(0,0,400,350);
```

```
header=new JLabel("Login Form");  
header.setForeground( Color.red);  
header.setFont(new Font("luxury",Font.BOLD,20));  
header.setBounds(125,25,200,25);  
c=1;
```

```
lblusrnm=new JLabel("USER NAME :");  
lblusrnm.setBounds(50,120,100,25);  
lblusrnm.setForeground(Color.red);
```

```
lblpasswd=new JLabel("PASSWORD :");  
lblpasswd.setBounds(50,170,100,25);  
lblpasswd.setForeground(Color.red);
```

```
txtpasswd=new TextField(10);  
txtpasswd.setBounds(150,170,150,25);  
txtpasswd.setBackground( Color.yellow);  
txtpasswd.addActionListener(new enterListener());  
txtpasswd.setEchoChar('*');
```

```
login=new JButton(" Login ");  
login.setBounds(60,220,100,25);  
login.addActionListener(this);  
login.setToolTipText("press to login");  
login.setMnemonic('l');
```

```
cancel=new JButton(" Cancel ");  
cancel.setBounds(190,220,100,25);  
cancel.addActionListener(this);  
cancel.setToolTipText("press to exit");  
cancel.setMnemonic('c');
```

```
refresh = new JButton("Refresh");  
refresh.setBounds(170,55,80,25);  
refresh.addActionListener(this);  
refresh.setToolTipText("Press to Refresh");
```



## COLLEGE MANAGEMENT SYSTEM



```
refresh.setMnemonic('r');

usnm = new Choice();
usnm.setBounds(150,120,150,25);

footer=new JTextArea(3,1);
footer.setText("    Please Fill The Password Correct \n
You Have Maximum Three (3) Chances To Try");
footer.setWrapStyleWord(true);
footer.setBounds(0,270,400,50);
footer.setForeground(Color.white);
footer.setBackground(new Color(126,1,5));
footer.setEditable(false);
footer.setFont(new Font("Times of new
Roman",Font.BOLD,12));

cn.add(header);
cn.add(title);
cn.add(lblusnm);
cn.add(lblpasswd);
cn.add(txtpasswd);
cn.add(usnm);
cn.add(login);
cn.add(cancel);
cn.add(refresh);
cn.add(footer);
cn.add(img);
Dimension
wind=Toolkit.getDefaultToolkit().getScreenSize();
pack();
setBounds((wind.width-400)/2,(wind.height-
350)/2,400,350);

setVisible(true);
connect();
try
{
    rs=stat.executeQuery("select username from login
order by username");
```



## COLLEGE MANAGEMENT SYSTEM



```
        while(rs.next())
            usnm.addItem(rs.getString(1));
    }
    catch(SQLException sqle)
    {
        try
        {
            stat.executeUpdate("create table login
(username varchar2(40) primary key, password varchar2(30));");
        }
        catch(Exception ec)
        {
            JOptionPane.showMessageDialog(null,"error to load
Form"+sqle,"Form loadin Error",JOptionPane.ERROR_MESSAGE);
        }
    }
}
private void log()
{
    str1=usnm.getSelectedItem().toString();
    str2=txtpasswd.getText();
    if(c==3)
    {
        JOptionPane.showMessageDialog(null,"Sorry Youre
Chances Will No Longer \n Please Try Again Later");
        System.exit(0);
    }
    try
    {
        rs=stat.executeQuery("select password from login where
username='"+str1+"'");
        rs.next();
        String tmp=rs.getString(1);
        if(str2.compareTo(tmp.trim())==0)
        {
            new Menu(str1);
            rs.close();
            dispose();
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
    }
    else
    {
        JOptionPane.showMessageDialog(null," Youre
"+c+" Chances Will go \n Try Again");
        txtpasswd.setText("");
        txtpasswd.requestFocus();
        c++;
    }
}
catch(SQLException sqle)
{
    JOptionPane.showMessageDialog(null,"error to load
Form"+sqle,"Form loadin Error",JOptionPane.ERROR_MESSAGE);
}
}
public void actionPerformed(ActionEvent e)
{
    String tmp=e.getActionCommand();
    tmp=tmp.trim();
    if(tmp.equalsIgnoreCase("login"))
    {
        log();
    }
    if(tmp.equalsIgnoreCase("refresh"))
    {
        try
        {
            String nm;
            int n = JOptionPane.showConfirmDialog(null,
"ARE U SURE TO RESTORE","Confirmation",JOptionPane.YES_NO_OPTION);
            if (n == JOptionPane.YES_OPTION)
            {
                nm=JOptionPane.showInputDialog(null,"please
enter the bakup File name to be restored \n OTHERWISE DEFAULT NAME bkp FILE
WILL BE USED","bkp");

                if(nm==null || nm.compareTo("")==0)
                nm="bkp";
            }
        }
        catch (Exception ex)
        {
            JOptionPane.showMessageDialog(null,"Error in Refreshing Data");
        }
    }
}
```





## COLLEGE MANAGEMENT SYSTEM



Process

```
p=Runtime.getRuntime().exec("imp USERID=system/123 BUFFER=2000
FILE="+nm+" IGNORE=yes ROWS=yes
TABLES=login,affiliation,staffdetail,studentdetail,studentfeedetail,payroll,billingm
aster,bcaresult,bbaresult CONSTRAINTS=YES");
        JOptionPane.showMessageDialog(null,"RESTORING
DATABASE FROM BACKUP FILE ","INFORMATION
MESSAGE",JOptionPane.INFORMATION_MESSAGE);
    }
    }
    catch(Exception io)
    {
        JOptionPane.showMessageDialog(null,"Sorry we
Can't RESTORE"+io,"ERROR",JOptionPane.ERROR_MESSAGE);
    }
}
else if(tmp.equalsIgnoreCase("cancel"))
{
    System.exit(0);
}
}
private class enterListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        log();
    }
}
private class tmListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if(x<0)
            x=130;
        title.setLocation(x,0);
        x=x-1;
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            conn=DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stat=conn.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,"error in
Creating Connection","Database Error",JOptionPane.ERROR_MESSAGE);
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not connected to
Database"+e,"DataBase Error",JOptionPane.ERROR_MESSAGE);
    }
}

public static void main(String args[])
{
    new Login();
}
}
```



## /\* Menu Pages \*/

```
import java.lang.*;
import module.*;
import report.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;
import java.io.*;
import java.applet.*;
public class Menu extends JFrame
{
    private JMenuBar mb;
    private JMenu im,om,rm,tm,am,hm,em;
    private JMenuItem
miaffiliation,mistaffdetail,mistudentdetail,mistudentfeedetail,mipayroll,mibilling
master,mibcaresult,mibbaresult;
    private JMenuItem
raffiliation,rstaffdetail,rstudentdetail,rstudentfeedetail,rpayroll,rbillingmaster,rbc
aresult,rbbaresult;

    private JMenuItem
miDeveloper,miSoftware,miYes,miNo,miDocu,mibakup,mirestore,michngpasswd,
micreateusr;
    private ImageIcon image;
    private JLabel l1,l2,l3,l4;
    private JPanel p1,p2;
    private Container cn;
    private String usrtype,str1;
    int r;

    File wavFile = new File("c:\\\\123.wav");
    AudioClip sound;
    public Menu(String ss)
    {

        setTitle("WELCOME IN COLLEGE MANAGEMENT SYSTEM");
```



## COLLEGE MANAGEMENT SYSTEM



```
getContentPane().setLayout(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
str1 = ss.trim();

try{sound = Applet.newAudioClip(wavFile.toURL());}
catch(Exception e){e.printStackTrace();}
sound.play();

p1 = new JPanel();
p2 = new JPanel();

p1.setLayout(null);
p2.setLayout(null);

p2.setBounds(0,0,800,40);

p1.setBackground( new Color(126,2,5));
p2.setBackground(new Color(255,200,255));
r = 0;
l2 = new JLabel("WELCOME IN COLLEGE MANAGEMENT
SYSTEM");
l2.setForeground(Color.white);
l2.setFont(new Font("monotype cursiva",Font.BOLD,35));
l2.setFont(new Font("impact",Font.BOLD,35));
l2.setBounds(0,10,1365,30);

Timer tim = new Timer(5,new timerListener());
tim.start();

image = new ImageIcon("menupic.jpg");

l1=new JLabel(image);
l1.setBounds(0,45,1365,580);
l4=new JLabel(new ImageIcon("123.jpg"));
l4.setBounds(0,0,1365,600);
l3=new JLabel("    DEVELOPED BY :- MD. AMAIRUL , MD. ARIF ,
MD MAHTAB , MD. TAUQUEER , MOTAHHIR , SANOJ          email id:
collegemanagementsystem@gmail.com");
```



## COLLEGE MANAGEMENT SYSTEM



```
l3.setFont(new Font("Arial",Font.BOLD,20));  
l3.setForeground(new Color(255,0,255));  
l3.setForeground(Color.cyan);  
l3.setBounds(0,640,1365,20);
```

```
cn = getContentPane();  
cn.setLayout(new GridLayout(1,1));  
p1.add(l2);  
p1.add(l1);  
p1.add(l3);  
p1.add(l4);  
cn.add(p1);  
mb = new JMenuBar();
```

```
im = new JMenu("Entry Form");  
rm = new JMenu("Report");  
tm = new JMenu("Tools");  
am = new JMenu("About");  
hm = new JMenu("Help");  
em = new JMenu("Exit");
```

```
im.setMnemonic('i');  
rm.setMnemonic('r');  
tm.setMnemonic('t');  
am.setMnemonic('a');  
hm.setMnemonic('h');  
em.setMnemonic('e');
```

```
miaffiliation = new JMenuItem("Affiliation");  
mistaffdetail = new JMenuItem("Staff Detail");  
mistudentdetail = new JMenuItem("Student Detail");
```

```
mistudentfeedetail = new JMenuItem("Student Fee Detail");  
mipayroll = new JMenuItem("PayRoll");  
mibillingmaster = new JMenuItem("Billing Master");  
mibcaresult = new JMenuItem("BCA result");  
mibbaresult = new JMenuItem("BBA result");
```



## COLLEGE MANAGEMENT SYSTEM



```
miDeveloper = new JMenuItem("Developer");
miSoftware = new JMenuItem("Software");
miYes = new JMenuItem("Yes");
miNo = new JMenuItem("No");
miDocu = new JMenuItem("Documentation");
mibakup = new JMenuItem("BackUp");
mirestore = new JMenuItem("Restore");
michngpasswd = new JMenuItem("Change Password");
micreateusr = new JMenuItem("Manage User");

raffiliation = new JMenuItem("Affiliation Report");
rstaffdetail = new JMenuItem("StaffDetail Report");
rstudentdetail = new JMenuItem("StudentDetail Report");
rstudentfeedetail = new JMenuItem("StudentFeeDetail Report");
rpayroll = new JMenuItem("PayRoll Report");
rbillingmaster = new JMenuItem("BillingMaster Report");
rbcaresult = new JMenuItem("BcaResult Report ");
rbbaresult = new JMenuItem("BbaResult Report ");

im.add(miaffiliation);
im.add(mistaffdetail);
im.add(mistudentdetail);

im.add(mistudentfeedetail);
im.add(mipayroll);
im.add(mibillingmaster);
im.add(mibcaresult);
im.add(mibbaresult);

rm.add(raffiliation);

rm.add(rstaffdetail);
rm.add(rstudentdetail);
rm.add(rstudentfeedetail);
rm.add(rpayroll);
rm.add(rbillingmaster);
rm.add(rbcaresult);
rm.add(rbbaresult);
```



## COLLEGE MANAGEMENT SYSTEM



```
tm.add(mibakup);  
tm.add(mirestore);  
tm.add(micreateusr);  
tm.add(michngpasswd);
```

```
hm.add(am);  
am.add(miSoftware);  
am.add(miDeveloper);  
hm.add(miDocu);
```

```
em.add(miYes);  
em.add(miNo);
```

```
setJMenuBar(mb);  
mb.add(im);  
mb.add(rm);  
mb.add(tm);  
mb.add(hm);  
mb.add(em);
```

```
miaffiliation.addActionListener(new menuListener());  
mistaffdetail.addActionListener(new menuListener());  
mistudentdetail.addActionListener(new menuListener());  
mistudentfeeddetail.addActionListener(new menuListener());  
mipayroll.addActionListener(new menuListener());  
mibillingmaster.addActionListener(new menuListener());  
mibcaresult.addActionListener(new menuListener());  
mibbresult.addActionListener(new menuListener());
```

```
raffiliation.addActionListener(new reportListener());  
rstaffdetail.addActionListener(new reportListener());  
rstudentdetail.addActionListener(new reportListener());  
rstudentfeeddetail.addActionListener(new reportListener());  
rpayroll.addActionListener(new reportListener());  
rbillingmaster.addActionListener(new reportListener());  
rbcaresult.addActionListener(new reportListener());  
rbbresult.addActionListener(new reportListener());
```



## COLLEGE MANAGEMENT SYSTEM



```
miSoftware.addActionListener(new menuListener());
miDocu.addActionListener(new menuListener());
miDeveloper.addActionListener(new menuListener());
miYes.addActionListener(new menuListener());
mibakup.addActionListener(new menuListener());
mirestore.addActionListener(new menuListener());
michngpasswd.addActionListener(new menuListener());
micreateusr.addActionListener(new menuListener());

em.addActionListener(new menuListener());
try
{
    try
    {
        String tmp = "LOCAL";
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
        Statement stm = con.createStatement();
        ResultSet rs = stm.executeQuery("select usertype
from login where username = '"+str1.trim()+"'");
        rs.next();
        String str = rs.getString(1);

        if(tmp.equalsIgnoreCase(str.trim()))
        {
            im.setEnabled(false);
        }
    }
    catch(SQLException sqle)
    {
        JOptionPane.showMessageDialog(null,"error in
Creating Connection","Database Error",JOptionPane.ERROR_MESSAGE);
    }
}
catch(Exception e)
{
```





## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"Not connected to  
Database","DataBase Error",JOptionPane.ERROR_MESSAGE);  
}
```

```
mb.setBackground(new Color(140,251,248));
```

```
miaffiliation.setBackground(new Color(204,255,0));
```

```
mistaffdetail.setBackground(new Color(204,255,0));
```

```
mistudentdetail.setBackground(new Color(204,255,0));
```

```
mistudentfeedetail.setBackground(new Color(204,255,0));
```

```
mipayroll.setBackground(new Color(204,255,0));
```

```
mibillingmaster.setBackground(new Color(204,255,0));
```

```
mibcaresult.setBackground(new Color(204,255,0));
```

```
mibbaresult.setBackground(new Color(204,255,0));
```

```
raffiliation.setBackground(new Color(204,255,0));
```

```
rstaffdetail.setBackground(new Color(204,255,0));
```

```
rstudentdetail.setBackground(new Color(204,255,0));
```

```
rstudentfeedetail.setBackground(new Color(204,255,0));
```

```
rpayroll.setBackground(new Color(204,255,0));
```

```
rbillingmaster.setBackground(new Color(204,255,0));
```

```
rbcaresult.setBackground(new Color(204,255,0));
```

```
rbbaresult.setBackground(new Color(204,255,0));
```

```
miDeveloper.setBackground(new Color(204,255,0));
```

```
miYes.setBackground(new Color(204,255,0));
```

```
miNo.setBackground(new Color(204,255,0));
```

```
mibakup.setBackground(new Color(204,255,0));
```

```
mirestore.setBackground(new Color(204,255,0));
```

```
michngpasswd.setBackground(new Color(204,255,0));
```

```
micreateusr.setBackground(new Color(204,255,0));
```

```
miDocu.setBackground(new Color(204,255,0));
```

```
miSoftware.setBackground(new Color(204,255,0));
```

```
am.setBackground(new Color(204,255,0));
```

```
hm.setBackground(new Color(204,255,0));
```

```
setSize(1360,730);
```

```
setLocation(0,0);
```



## COLLEGE MANAGEMENT SYSTEM



```
setResizable(false);
setVisible(true);
}
private class menuListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == miaffiliation)
        {
            Affiliation affl = new Affiliation();
        }
        if(e.getSource() == mistaffdetail)
        {
            StaffDetail staff = new StaffDetail ();
        }
        if(e.getSource() == mistudentdetail)
        {
            StudentDetail std = new StudentDetail();
        }
        if(e.getSource() == mistudentfeedetail)
        {
            StudentFeeDetail stf = new StudentFeeDetail();
        }
        if(e.getSource() == mipayroll)
        {
            PayRoll payr = new PayRoll();
        }
        if(e.getSource() == mibillingmaster)
        {
            BillingMaster bilm = new BillingMaster();
        }
        if(e.getSource() == mibcaresult)
        {
            BcaResult bcar = new BcaResult();
        }
        if(e.getSource() == mibbaresult)
        {
            BbaResult bbar = new BbaResult();
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
    }
    if(e.getSource()==miDocu)
    {
        try
        {
            Process Process
p=Runtime.getRuntime().exec("C:/Program Files/Microsoft
Office/Office12/winword.exe documentation.doc");
        }
        catch(IOException io)
        {
            JOptionPane.showMessageDialog(null,"Some Intel2al
Problem"+io,"ERROR",JOptionPane.ERROR_MESSAGE);
        }
    }
    if(e.getSource() == miSoftware)
    {
        aboutSoftware abs = new aboutSoftware();
    }
    if(e.getSource() == miDeveloper)
    {
        aboutDeveloper abd = new aboutDeveloper();
    }
    if(e.getSource()==miYes)
    {
        int n = JOptionPane.showConfirmDialog(null, "Would you
really want to EXIT","Confirmation",JOptionPane.YES_NO_OPTION);
        if (n == JOptionPane.YES_OPTION)
            System.exit(0);
    }
    if(e.getSource() == mibakup)
    {
        try
        {
            String nm;
            int n = JOptionPane.showConfirmDialog(null,
"ARE YOU SURE TO TAKE BAKUP OF THE
FILE","Confirmation",JOptionPane.YES_NO_OPTION);
```



## COLLEGE MANAGEMENT SYSTEM



```
        if (n == JOptionPane.YES_OPTION)
        {
            nm=JOptionPane.showInputDialog(null,"PLEASE ENTER A BACKUP FILE NAME
            TO BE CREATE \n OTHERWISE DEFAULT NAME bkp WILL BE USED","bkp");
            if(nm==null || nm.compareTo("")==0)
                nm = "bkp";
            Process
            p=Runtime.getRuntime().exec("exp USERID=system/123 BUFFER=2000
            FILE="+nm+" ROWS=yes
            TABLES=login,affiliation,staffdetail,studentdetail,studentfeeddetail,payroll,billingm
            aster,bcaresult,bbaresult RECORD=yes TRIGGERS=yes CONSTRAINTS=YES");

            JOptionPane.showMessageDialog(null,"TAKING BACKUP OF THE
            DATABASE ","INFORMATION MESSAGE",JOptionPane.INFORMATION_MESSAGE);
        }
    }
    catch(IOException io)
    {
        JOptionPane.showMessageDialog(null,"Sorry we
        Can't take Backup"+io,"ERROR",JOptionPane.ERROR_MESSAGE);
    }
}
if(e.getSource()==mirestore)
{
    try
    {
        String nm;
        int n = JOptionPane.showConfirmDialog(null,
        "ARE U SURE TO RESTORE","Confirmation",JOptionPane.YES_NO_OPTION);
        if (n == JOptionPane.YES_OPTION)
        {
            nm=JOptionPane.showInputDialog(null,"please enter the bakup
            File name to be restored \n OTHERWISE DEFAULT NAME bkp FILE WILL BE
            USED","bkp");
            if(nm==null || nm.compareTo("")==0)
                nm="bkp";
            Process p=Runtime.getRuntime().exec("imp USERID=system/123 BUFFER=2000
            FILE="+nm+" IGNORE=yes ROWS=yes
```



## COLLEGE MANAGEMENT SYSTEM



TABLES=login,affiliation,staffdetail,studentdetail,studentfeedetail,payroll,billingmaster,bcaresult,bbaresult CONSTRAINTS=YES");

```
JOptionPane.showMessageDialog(null,"RESTORING DATABASE FROM  
BACKUP FILE ", "INFORMATION  
MESSAGE",JOptionPane.INFORMATION_MESSAGE);
```

```
}
```

```
}
```

```
catch(IOException io)
```

```
{
```

```
JOptionPane.showMessageDialog(null,"Sorry we Can't  
RESTORE"+io,"ERROR",JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
}
```

```
if(e.getSource() == michngpasswd)
```

```
{
```

```
new ChangePassword(str1);
```

```
}
```

```
if(e.getSource() == micreateusr)
```

```
{
```

```
new CreateUser(str1);
```

```
}
```

```
}
```

```
}
```

```
private class reportListener implements ActionListener
```

```
{
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
if(e.getSource() == raffiliation)
```

```
{
```

```
new Affiliation_Report();
```

```
}
```

```
if(e.getSource() == rstaffdetail)
```

```
{
```

```
new StaffDetail_Report();
```

```
}
```

```
if(e.getSource() == rstudentdetail)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    new StudentDetail_Report();
}
if(e.getSource() == rstudentfeedetail)
{
    new StudentFeeDetail_Report();
}
if(e.getSource() == rpayroll)
{
    new PayRoll_Report();
}
if(e.getSource() == rbillingmaster)
{
    new BillingMaster_Report();
}
if(e.getSource() == rbcaresult)
{
    new BcaResult_Report();
}
if(e.getSource() == rbbareresult)
{
    new BbaResult_Report();
}
}
}
private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l2.setLocation(r,10);
        r=r-1;
        if(r<0)
            r=1365;
    }
}
public static void main(String arg[])
{
    new Menu(arg[0]); } }
```





## COLLEGE MANAGEMENT SYSTEM



**/\* Create User \*/**

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;

public class CreateUser extends JFrame implements ActionListener
{
    private JLabel lblusnm, lblpasswd, lblconfirmpasswd, header, img;
    private JTextArea footer;
    private TextField usnm, txtpasswd, txtconfirmpasswd;
    private JButton create, cancel, del;
    private JRadioButton rb1, rb2;
    private ButtonGroup gp;
    private int c, x;
    private Connection conn;
    private Statement stat;
    private ResultSet rs;
    private String str1, str2, str3, tp;
    private Container cn;
    public String uss;
    public CreateUser(String usr)
    {
        setTitle("Create User");
        cn = getContentPane();
        cn.setLayout(null);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        cn.setBackground(new Color(86, 12, 14));
        uss = usr;

        img = new JLabel(new ImageIcon("cuser.jpg"));
        img.setBounds(0, 0, 400, 410);
        JPanel pn = new JPanel();
        pn.setBounds(100, 10, 200, 45);
        pn.setBackground(new Color(86, 12, 14));
```



## COLLEGE MANAGEMENT SYSTEM



```
header=new JLabel(" New User Form ");
header.setForeground(Color.white);
header.setFont(new Font("luxury",Font.BOLD,20));

lblusnm=new JLabel("USER NAME :");
lblusnm.setBounds(50,70,100,25);
lblusnm.setForeground(Color.white);

lblpasswd=new JLabel("PASSWORD :");
lblpasswd.setBounds(50,110,100,25);
lblpasswd.setForeground(Color.white);

lblconfirmpasswd=new JLabel("Confirm PASSWORD :");
lblconfirmpasswd.setBounds(50,145,150,25);
lblconfirmpasswd.setForeground(Color.white);

gp=new ButtonGroup();
rb1=new JRadioButton("ADMIN");
rb2=new JRadioButton("LOCAL");
rb1.setBackground(new Color(86,12,14));
rb2.setBackground(new Color(86,12,14));
rb1.setForeground(new Color(255,255,255));
rb2.setForeground(new Color(255,255,255));
rb1.setBounds(50,180,100,25);
rb2.setBounds(230,180,100,25);
rb1.addActionListener(new rbListener());
rb2.addActionListener(new rbListener());
gp.add(rb1);
gp.add(rb2);
tp="ADMIN";

txtpasswd=new TextField(10);
txtpasswd.setBounds(200,110,150,25);
txtpasswd.setEchoChar('*');

txtconfirmpasswd=new TextField(10);

txtconfirmpasswd.setBounds(200,145,150,25);
```



## COLLEGE MANAGEMENT SYSTEM



```
txtconfirmpasswd.setEchoChar('*');
```

```
create=new JButton("Create User");  
create.setBounds(50,290,120,25);  
create.addActionListener(this);  
create.setToolTipText("press to Create User");  
create.setMnemonic('U');
```

```
cancel=new JButton(" Cancel ");  
cancel.setBounds(220,290,120,25);  
cancel.addActionListener(this);  
cancel.setToolTipText("press to exit");  
cancel.setMnemonic('c');
```

```
del=new JButton(" Delete User Account ");  
del.setBounds(70,330,250,25);  
del.setForeground(Color.blue);  
del.addActionListener(this);  
del.setToolTipText("press to Delete User Account");  
del.setMnemonic('D');
```

```
usrnm=new TextField();  
usrnm.setBounds(200,70,150,25);
```

```
footer=new JTextArea(3,1);  
footer.setText("IF YOU WILL CREATE ADMIN USER \nTHEN YOU  
WILL SET ALL THE PROPERTY OF ENTRY FORM \nBE CAREFULL IT WILL HARMS  
THE SOFTWARE IN WRONG EMPLOYEE");  
footer.setBounds(0,220,400,50);  
footer.setEditable(false);  
cn.add(footer);
```

```
pn.add(header);  
cn.add(lblusrnm);  
cn.add(lblpasswd);
```

```
cn.add(txtpasswd);  
cn.add(usrnm);
```



## COLLEGE MANAGEMENT SYSTEM



```
cn.add(lblconfirmpasswd);
cn.add(txtconfirmpasswd);
cn.add(create);
cn.add(cancel);
cn.add(del);
cn.add(rb1);
cn.add(rb2);
cn.add(pn);
cn.add(img);
```

```
Dimension wind=Toolkit.getDefaultToolkit().getScreenSize();
```

```
pack();
setBounds((wind.width-400)/2,(wind.height-410)/2,400,410);
setVisible(true);
connect();
```

```
try
{
    rs=stat.executeQuery("select usertype from login where
username='"+usr.trim()+"'");
    rs.next();
    str1=rs.getString(1);
}
catch(SQLException sqle)
{
    JOptionPane.showMessageDialog(null,"could not update
User"+sqle,"USER ACCOUNT",JOptionPane.ERROR_MESSAGE);
}
str1=str1.trim();
JOptionPane.showMessageDialog(null,str1);
if(str1.equalsIgnoreCase("ADMIN"))
{
    del.setEnabled(true);

    rb1.setSelected(true);
}
else
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    del.setEnabled(false);
    rb1.setEnabled(false);
    rb2.setSelected(true);
    tp="LOCAL";
    footer.setText("IF YOU WILL CREATE Local USER \n THEN
YOU WILL SET MOST OF THE PROPERTY OF ENTRY FORM \n RECOMMENDED FOR
MOST EMPLOYEE");
    footer.setBounds(0,220,400,50);
    cn.add(footer);
}
}
private void createuserid()
{
    str1=usnm.getText();
    str2=txtpasswd.getText();
    str3=txtconfirmpasswd.getText();
    if(str1.compareTo("")!=0 || str2.compareTo("")!=0 ||
str3.compareTo("")!=0)
    {
        if(str2.compareTo(str3.trim())==0)
        {
            try
            {
                stat.executeUpdate("insert into login
values('"+str1.trim()+"','"+str2.trim()+"','"+tp.trim()+"')");
                stat.executeUpdate("commit");
                JOptionPane.showMessageDialog(null,"User
Created Succesfully","USER ACCOUNT",JOptionPane.INFORMATION_MESSAGE);
                dispose();
            }
            catch(SQLException sqle)
            {
                JOptionPane.showMessageDialog(null,"could not Create User"+sqle,"USER
ACCOUNT",JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        else
        {
            JOptionPane.showMessageDialog(null,"Password
doesn't match","USER ACCOUNT",JOptionPane.WARNING_MESSAGE);
        }
    }
    else
        JOptionPane.showMessageDialog(null,"Fill All The Field in
New User Form","USER ACCOUNT",JOptionPane.WARNING_MESSAGE);
    }
    public void actionPerformed(ActionEvent e)
    {
        String tmp=e.getActionCommand();
        tmp=tmp.trim();
        if(e.getSource()==create)
        {
            createuserid();
        }
        else if(e.getSource()==cancel)
        {
            dispose();
        }
        else if(e.getSource()==del)
            new delDialog(this,"Delete user account");
    }
    private class rbListener implements ActionListener
    {
        public void actionPerformed(ActionEvent e)
        {
            tp=e.getActionCommand();
            if(tp.compareTo("ADMINISTRATOR")==0)
            {
                footer.setText("IF YOU WILL CREATE ADMINSTATOR
USER \n THEN YOU WILL SET ALL THE PROPERTY OF ENTRY FORM \n BE
CAREFULL IT WILL HARMS THE SOFTWARE IN WRONG EMPLOYEE");
                footer.setBounds(0,220,400,50);
                cn.add(footer);
            }
        }
    }
```



## COLLEGE MANAGEMENT SYSTEM



```
else if(tp.compareTo("LOCAL")==0)
{
    footer.setText("IF YOU WILL CREATE Local USER \n
THEN YOU WILL SET MOST OF THE PROPERTY OF ENTRY FORM \n
RECOMMENDED FOR MOST EMPLOYEE");
    footer.setBounds(0,220,400,50);
    cn.add(footer);
}
}
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            conn=DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stat=conn.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,"error in
Creating Connection","Database Error",JOptionPane.ERROR_MESSAGE);
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not connected to
Database","DataBase Error",JOptionPane.ERROR_MESSAGE);
    }
}
private class delDialog extends JDialog implements ActionListener
{
    private JLabel lblusrnm,lblpasswd,header;
    private TextField txtpasswd;
    private JButton del,cancel;
    private Choice usrn;
```





## COLLEGE MANAGEMENT SYSTEM



```
private String str1,str2;
private Container cn;

public delDialog(JFrame parent,String title)
{
    super(parent,title,false);
    cn=getContentPane();
    cn.setLayout(null);
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    cn.setBackground(new Color(250, 216, 217));

    JPanel pn=new JPanel();
    pn.setBounds(100,10,200,45);
    pn.setBackground(new Color(255,240,255));

    JLabel img=new JLabel(new ImageIcon("Copy of
1658221146.jpg"));
    img.setBounds(0,0,400,200);
    header=new JLabel("Delete User Account");
    header.setForeground(Color.BLUE);
    header.setFont(new Font("luxury",Font.BOLD,20));

    lblusnm=new JLabel("USER NAME :");
    lblusnm.setBounds(50,70,100,25);
    lblusnm.setForeground(Color.blue);

    lblpasswd=new JLabel("PASSWORD :");
    lblpasswd.setBounds(50,100,100,25);
    lblpasswd.setForeground(Color.blue);

    txtpasswd=new TextField(10);
    txtpasswd.setBounds(200,100,150,25);
    txtpasswd.setBackground(Color.green);
    txtpasswd.setEchoChar('*');

    del=new JButton(" Delete User ");
    del.setBounds(50,135,150,25);
    del.addActionListener(this);
```



## COLLEGE MANAGEMENT SYSTEM



```
del.setToolTipText("press to Delete User");
del.setMnemonic('D');

cancel=new JButton(" Cancel ");
cancel.setBounds(230,135,150,25);
cancel.addActionListener(this);
cancel.setToolTipText("press to exit");
cancel.setMnemonic('c');

usrnm=new Choice();
usrnm.setBounds(200,70,150,25);

pn.add(header);
cn.add(lblusrnm);
cn.add(lblpasswd);
cn.add(txtpasswd);
cn.add(usrnm);
cn.add(del);
cn.add(cancel);
cn.add(pn);

cn.add(img);
Dimension
wind=Toolkit.getDefaultToolkit().getScreenSize();
pack();
setBounds((wind.width-400)/2,(wind.height-
200)/2,400,200);

setVisible(true);
connect();
try
{
    String tmp;
    rs=stat.executeQuery("select username from login
order by username");

    while(rs.next())
    {
        tmp=rs.getString(1);
        if(uss.compareTo(tmp.trim())==0);
```



## COLLEGE MANAGEMENT SYSTEM



```
else
    usnm.addItem(tmp);
}
}
catch(SQLException sqle)
{
    JOptionPane.showMessageDialog(null,"error to load
Form"+sqle,"Form loadin Error",JOptionPane.ERROR_MESSAGE);
}
}
private void Delete()
{
    str1=usnm.getSelectedItem().toString();
    str2=txtpasswd.getText();
    if(str2.compareTo("")!=0)
    try
    {
        rs=stat.executeQuery("select password from login
where username='"+str1+"'");
        rs.next();
        String tmp=rs.getString(1);
        if(str2.compareTo(tmp.trim())==0)
        {
            stat.executeUpdate("delete from login
where username='"+str1+"'");
            stat.executeUpdate("commit");

            JOptionPane.showMessageDialog(null,"User Account Deleted Successfully","USER
ACCOUNT",JOptionPane.INFORMATION_MESSAGE);
            dispose();
        }
        else
            JOptionPane.showMessageDialog(null,"Incorrect
Password","USER ACCOUNT",JOptionPane.WARNING_MESSAGE);
    }
    catch(SQLException sqle)
    {

```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"error to load
Form"+sqle,"User Account",JOptionPane.ERROR_MESSAGE);
    }
    else
        JOptionPane.showMessageDialog(null,"Please Fill
The Password","USER ACCOUNT",JOptionPane.WARNING_MESSAGE);
    }
    public void actionPerformed(ActionEvent e)
    {
        String tmp=e.getActionCommand();
        tmp=tmp.trim();
        if(tmp.equalsIgnoreCase("Delete User"))
        {
            Delete();
        }
        else if(tmp.equalsIgnoreCase("cancel"))
        {
            dispose();
        }
    }
}
public static void main(String args[])
{
    new CreateUser(args[0]);
}
}
```



## COLLEGE MANAGEMENT SYSTEM



### **/\* Change Password \*/**

```
import java.awt.*;
import javax.swing.border.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;

public class ChangePassword extends JFrame implements ActionListener
{
    private JLabel
lbluser_name, lbluser_type, lblpassword, lblconfirmpassword, lblcurrent, header;
    private JTextArea footer;
    private TextField txtcurrent, txtpassword, txtconfirmpassword;
    private JTextField user_name, user_type;
    private JButton upd, cancel;
    private ButtonGroup gp;
    private int c, x;
    private Connection conn;
    private Statement stat;
    private ResultSet rs;
    private String str1, str2, str3, str4, tmp, tp;
    private Container cn;

    public ChangePassword(String usr)
    {
        setTitle("college management system");
        cn=getContentPane();
        cn.setLayout(null);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        JPanel pn=new JPanel();
        pn.setBounds(100,0,200,45);
        pn.setBackground(new Color(255,240,255));

        str4=usr;

        JLabel img=new JLabel(new ImageIcon("Copy3.jpg"));
        img.setBounds(0,0,400,300);

        header=new JLabel(" Change User Setting ");
```



## COLLEGE MANAGEMENT SYSTEM



```
header.setForeground(Color.BLUE);  
header.setFont(new Font("luxury",Font.BOLD,20));
```

```
lbluser_name=new JLabel("USER NAME :");  
lbluser_name.setBounds(50,60,100,25);  
lbluser_name.setForeground(Color.blue);
```

```
lbluser_type=new JLabel("USER TYPE :");  
lbluser_type.setBounds(50,90,100,25);  
lbluser_type.setForeground(Color.blue);
```

```
lblcurrent=new JLabel(" Current Password ");  
lblcurrent.setBounds(30,130,150,25);  
lblcurrent.setForeground(Color.blue);
```

```
lblpassword=new JLabel(" New PASSWORD :");  
lblpassword.setBounds(30,165,150,25);  
lblpassword.setForeground(Color.blue);
```

```
lblconfirmpassword=new JLabel(" Confirm PASSWORD :");  
lblconfirmpassword.setBounds(30,200,150,25);  
lblconfirmpassword.setForeground(Color.blue);
```

```
tp="ADMIN";
```

```
user_name=new JTextField();  
user_name.setText(usr.toUpperCase());  
user_name.setBorder(new  
BevelBorder(BevelBorder.RAISED));  
user_name.setForeground(Color.blue);  
user_name.setEditable(false);  
user_name.setBackground(new Color(255,240,255));  
user_name.setBounds(200,60,150,25);
```

```
user_type=new JTextField();  
user_type.setBorder(new  
BevelBorder(BevelBorder.RAISED));  
user_type.setForeground(Color.blue);
```



## COLLEGE MANAGEMENT SYSTEM



```
user_type.setEditable(false);
user_type.setBackground(new Color(255,240,255));
user_type.setBounds(200,90,150,25);

txtcurrent=new TextField(10);
txtcurrent.setBounds(200,130,150,25);
txtcurrent.setEchoChar('*');

txtpassword=new TextField(10);
txtpassword.setBounds(200,165,150,25);
txtpassword.setEchoChar('*');

txtconfirmpassword=new TextField(10);
txtconfirmpassword.setBounds(200,200,150,25);
txtconfirmpassword.setEchoChar('*');

upd=new JButton(" Update ");
upd.setBounds(50,235,150,25);
upd.addActionListener(this);
upd.setToolTipText("press to Change The Password");
upd.setMnemonic('U');

cancel=new JButton(" Cancel ");
cancel.setBounds(220,235,150,25);
cancel.addActionListener(this);
cancel.setToolTipText("press to exit");
cancel.setMnemonic('c');

cn.setBackground(new Color(250, 216, 217));
pn.add(header);
cn.add(lbluser_name);
cn.add(lblpassword);

cn.add(txtpassword);
cn.add(lblcurrent);
cn.add(lbluser_type);
cn.add(user_name);
cn.add(user_type);
```





## COLLEGE MANAGEMENT SYSTEM



```
cn.add(txtcurrent);
cn.add(lblconfirmpassword);
cn.add(txtconfirmpassword);
cn.add(upd);
cn.add(cancel);
cn.add(pn);
cn.add(img);
Dimension
wind=Toolkit.getDefaultToolkit().getScreenSize();
pack();
setBounds((wind.width-400)/2,(wind.height-
300)/2,400,300);

setVisible(true);
connect();
txtcurrent.requestFocus();
try
{
    rs=stat.executeQuery("select
usertype from login where username='"+usr.trim()+"'");
    rs.next();
    user_type.setText(rs.getString(1));
}
catch(SQLException sqle)
{
    JOptionPane.showMessageDialog(null,"could not Change User
Password"+sqle,"USER ACCOUNT",JOptionPane.ERROR_MESSAGE);
}
}

private void upduserid()
{
    str1=txtcurrent.getText();
    str2=txtpassword.getText();
    str3=txtconfirmpassword.getText();
    try
    {
        rs=stat.executeQuery("select password from login
where username='"+str4.trim()+"'");
        rs.next();
```



## COLLEGE MANAGEMENT SYSTEM



```
        tmp=rs.getString(1);
    }
    catch(SQLException sqle)
    {
        JOptionPane.showMessageDialog(null,"could not
update User"+sqle,"USER ACCOUNT",JOptionPane.ERROR_MESSAGE);
    }
    if(str1.compareTo(tmp.trim())==0)
    {
        try
        {
            stat.executeUpdate("update login set
password='"+str2.trim()+"' where username='"+str4.trim()+"'");
            stat.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"User Password Changes
Succesfully","USER ACCOUNT",JOptionPane.INFORMATION_MESSAGE);
            dispose();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,"could not
Change User Password"+sqle,"USER ACCOUNT",JOptionPane.ERROR_MESSAGE);
        }
    }
    else
        JOptionPane.showMessageDialog(null,"new
Password and Confirm Password does not match \n        Try Again","USER
ACCOUNT",JOptionPane.WARNING_MESSAGE);
}
public void actionPerformed(ActionEvent e)
{
    String tmp=e.getActionCommand();
    tmp=tmp.trim();
    if(e.getSource()==upd)
    {
        upuserid();
    }
    else if(e.getSource()==cancel)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    dispose();
}
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            conn=DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stat=conn.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,"error in
Creating Connection","Database Error",JOptionPane.ERROR_MESSAGE);
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not connected to
Database","DataBase Error",JOptionPane.ERROR_MESSAGE);
    }
}
public static void main(String args[])
{
    new ChangePassword(args[0]);
}
}
```



## **/\* About Developer \*/**

```
import java.lang.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class aboutDeveloper extends JFrame
{
    private JButton btnClose;
    private JLabel lblHeader;
    private JTextArea txtArea;
    Dimension screenSize;
    public aboutDeveloper()
    {
        lblHeader = new JLabel();
        lblHeader.setForeground(Color.white);
        txtArea = new JTextArea();
        btnClose = new JButton();

        getContentPane().setLayout(null);
        getContentPane().setBackground(Color.yellow);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        lblHeader.setFont(new Font("Copperplate Gothic Bold", 1, 36));
        lblHeader.setText("Developer Details");
        getContentPane().add(lblHeader);
        lblHeader.setBounds(10, 10, 510, 40);

        txtArea.setBackground( Color.red);
        txtArea.setForeground( Color.white);
        txtArea.setText("  Name\t   :\t MD. AMAIRUL , MD. ARIF , MD MAHTAB ,
MD. TAUQUEER , MOTAHHIR , SANJOJ \n  Mobile No.\t   :\t Address \t
:\tPatna (Bihar)\n  Email :- collegemanagementsystem@gmail.com ");
        getContentPane().add(txtArea);

        txtArea.setBounds(40, 250, 350, 100);
        txtArea.setEditable(false);
        btnClose.setText("Close");
        getContentPane().add(btnClose);
```



## COLLEGE MANAGEMENT SYSTEM



```
btnClose.setBounds(350, 360, 80, 23);
    btnClose.addActionListener(new btnCloseListener());

    setResizable(false);
    setUndecorated(true);
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    setBounds((screenSize.width-500)/2, (screenSize.height-400)/3, 500, 400);
    setVisible(true);
}
private class btnCloseListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        dispose();
    }
}
}
```



## **/\* About Software \*/**

```
import java.lang.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class aboutSoftware extends JFrame
{
    private JButton btnClose;
    private JLabel lblPlatform;
    private JLabel lblHeader;
    private JLabel lblWindows;
    private JLabel lblPowered;
    private JLabel lblAbhi;
    Dimension screenSize;

    public aboutSoftware()
    {
        btnClose = new JButton();
        lblPlatform = new JLabel();
        lblHeader = new JLabel();
        lblWindows = new JLabel();
        lblPowered = new JLabel();
        lblAbhi = new JLabel();

        getContentPane().setLayout(null);
        getContentPane().setBackground(new Color(250, 216, 217));
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setResizable(false);
        setUndecorated(true);

        JLabel img=new JLabel(new ImageIcon("Copy1.jpg"));
        img.setBounds(0,0,600,220);

        getAccessibleContext().setAccessibleName("About");
        btnClose.setText("Close");

        getContentPane().add(btnClose);
        btnClose.setBounds(340, 190, 80, 23);
        btnClose.addActionListener(new btnCloseListener());
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
lblPlatform.setFont(new Font("Microsoft Sans Serif", 0, 18));  
lblPlatform.setForeground(new java.awt.Color(153, 153, 255));  
lblPlatform.setText("Platform :");  
getContentPane().add(lblPlatform);  
lblPlatform.setBounds(180, 80, 120, 20);
```

```
lblHeader.setFont(new Font("Century Schoolbook", 1, 24));  
lblHeader.setForeground(new java.awt.Color(255, 51, 51));  
lblHeader.setText("COLLEGE MANAGEMENT SYSTEM");  
getContentPane().add(lblHeader);  
lblHeader.setBounds(0, 10, 590, 30);
```

```
lblWindows.setForeground(new Color(0, 0, 153));  
lblWindows.setText("Windows 8/Windows 7/Windows 10");  
getContentPane().add(lblWindows);  
lblWindows.setBounds(220, 100, 250, 20);
```

```
lblPowered.setFont(new Font("Microsoft Sans Serif", 0, 18));  
lblPowered.setForeground(new java.awt.Color(153, 153, 255));  
lblPowered.setText("Powered by :");  
getContentPane().add(lblPowered);  
lblPowered.setBounds(180, 134, 130, 20);
```

```
lblAbhi.setForeground(new Color(0, 102, 51));  
lblAbhi.setText("MD. AMAIRUL , MD. ARIF , MD MAHTAB , MD. TAUQUEER ,  
MOTAHHIR , SANOJ");  
getContentPane().add(lblAbhi);  
lblAbhi.setBounds(250, 160, 190, 20);
```

```
getContentPane().add(img);  
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();  
setBounds((screenSize.width-600)/2, (screenSize.height-220)/2, 600, 220);
```

```
setVisible(true);  
}
```

```
private class btnCloseListener implements ActionListener
```



## COLLEGE MANAGEMENT SYSTEM



```
{  
    public void actionPerformed(ActionEvent e)  
    {  
        dispose();  
    }  
}
```





**/\* Affiliation \*/**

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class Affiliation extends JFrame
{
    private JLabel title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12;
    private JTextField t1,t2;
    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;
    private JComboBox cb1,cb2,cb3,cb4;
    private JCheckBox ch1,ch2,ch3,ch4;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;

    ResultSet rs;
    String str1,str2,str3,str4,str5,str,st;
    Connection con;
    Statement stm;
    Dimension screensize;
    String[] colhead = {"Affiliation no.,"Affiliated to","Course","Affiliation
date","Registration no."};
    Object data[][] = new Object[20][5];
    JTable table = new JTable(data,colhead);
    public Affiliation()
    {
        super("AFFILIATION FORM");
        r=1; l=1;
        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);
        cn.setBackground(new Color(128,128,25));
        EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);
        BevelBorder edge = new BevelBorder(BevelBorder.RAISED);

        p1 = new JPanel();
        p2 = new JPanel();
        p3 = new JPanel();
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
p4 = new JPanel();

p1.setBounds(5,5,1225,45);
p2.setBounds(5,55,300,475);
p3.setBounds(310,55,620,475);
p4.setBounds(935,55,295,475);

p1.setBackground(new Color(255,0,0));
p2.setBackground(new Color(112,0,0));
p3.setBackground(new Color(0,250,0));
p4.setBackground(new Color(254,200,0));

title = new JLabel("WELCOME SCREEN FOR AFFILIATION");
title.setFont(new Font("Comic Sans MS",1,28));
title.setForeground(Color.yellow);
p1.add(title);

Timer tm = new Timer(50,new timerListener());
tm.start();

Timer tm1 = new Timer(30,new timerListener1());
tm1.start();

pic = new JLabel();
Icon img = new ImageIcon("affiliation.jpg");
pic.setIcon(img);
p2.add(pic);

p3.setLayout(null);
p4.setLayout(null);

p3.setBorder(BorderFactory.createTitledBorder("Enter Information
for Affiliation"));

l1 = new JLabel("AFFILIATION NO.");
l2 = new JLabel("AFFILIATED TO");
l3 = new JLabel("COURSE OFFERED");
l4 = new JLabel("AFFILIATION DATE");
l5 = new JLabel("REGISTERED NO.");
l6 = new JLabel("DEVELOPED BY");

l6.setFont(new Font("STENCIL",3,20));
l6.setForeground(Color.blue);
l6.setBounds(0,850,140,25);
p4.add(l6);
```



## COLLEGE MANAGEMENT SYSTEM



```
17 = new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.
Tauqueer , Motahhir , Sanoj");
17.setFont(new Font("Monotype Corsiva",1,17));
17.setForeground(Color.blue);
17.setBounds(50,850,220,25);
p4.add(17);

18 = new JLabel("email: collegemanagementsystem@gmail.com");
18.setFont(new Font("Monotype Corsiva",1,17));
18.setForeground(Color.blue);
18.setBounds(50,850,220,25);
p4.add(18);

19 = new JLabel("Amairul Hassan");
19.setFont(new Font("Monotype Corsiva",1,17));
19.setForeground(Color.blue);
19.setBounds(50,850,220,25);
p4.add(19);

110 = new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.
Tauqueer , Motahhir , Sanoj");
110.setFont(new Font("Monotype Corsiva",1,17));
110.setForeground(Color.blue);
110.setBounds(50,850,220,25);
p4.add(110);

111 = new JLabel("Arif Hussain");
111.setFont(new Font("Monotype Corsiva",1,17));
111.setForeground(Color.blue);
111.setBounds(50,850,220,25);
p4.add(111);

112 = new JLabel("Mahtab");
112.setFont(new Font("Monotype Corsiva",1,17));
112.setForeground(Color.blue);
112.setBounds(50,850,220,25);
p4.add(112);

t1 = new JTextField(15);
t2 = new JTextField(15);

cb1 = new JComboBox();
cb1.addItem("-----SELECT-----");
cb1.addItem("MMHAPU");
cb1.addItem("MGM");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb1.addItem("PU");  
cb1.addItem("IGNU");
```

```
cb2 = new JComboBox();  
cb2.addItem("DD");  
cb2.addItem("01");  
cb2.addItem("02");  
cb2.addItem("03");  
cb2.addItem("04");  
cb2.addItem("05");  
cb2.addItem("06");  
cb2.addItem("07");  
cb2.addItem("08");  
cb2.addItem("09");  
cb2.addItem("10");  
cb2.addItem("11");  
cb2.addItem("12");  
cb2.addItem("13");  
cb2.addItem("14");  
cb2.addItem("15");  
cb2.addItem("16");  
cb2.addItem("17");  
cb2.addItem("18");  
cb2.addItem("19");  
cb2.addItem("20");  
cb2.addItem("21");  
cb2.addItem("22");  
cb2.addItem("23");  
cb2.addItem("24");  
cb2.addItem("25");  
cb2.addItem("26");  
cb2.addItem("27");  
cb2.addItem("28");  
cb2.addItem("29");  
cb2.addItem("30");  
cb2.addItem("31");
```

```
cb3 = new JComboBox();  
cb3.addItem("MM");  
cb3.addItem("Jan");
```

```
cb3.addItem("Feb");  
cb3.addItem("Mar");  
cb3.addItem("Apr");  
cb3.addItem("May");  
cb3.addItem("Jun");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb3.addItem("Jul");
cb3.addItem("Aug");
cb3.addItem("Sep");
cb3.addItem("Oct");
cb3.addItem("Nov");
cb3.addItem("Dec");

cb4 = new JComboBox();
cb4.addItem("YYYY");
cb4.addItem("1970");
cb4.addItem("1971");
cb4.addItem("1972");
cb4.addItem("1973");
cb4.addItem("1974");
cb4.addItem("1975");
cb4.addItem("1976");
cb4.addItem("1977");
cb4.addItem("1978");
cb4.addItem("1979");
cb4.addItem("1980");
cb4.addItem("1981");
cb4.addItem("1982");
cb4.addItem("1983");
cb4.addItem("1984");
cb4.addItem("1985");
cb4.addItem("1986");
cb4.addItem("1987");
cb4.addItem("1988");
cb4.addItem("1989");
cb4.addItem("1990");
cb4.addItem("1991");
cb4.addItem("1992");
cb4.addItem("1993");
cb4.addItem("1994");
cb4.addItem("1995");
cb4.addItem("1996");
cb4.addItem("1997");
cb4.addItem("1998");
cb4.addItem("1999");

cb4.addItem("2000");

ch1 = new JCheckBox("BCA");
ch2 = new JCheckBox("BBA");
ch3 = new JCheckBox("BSc");
ch4 = new JCheckBox("BJMC");
```



## COLLEGE MANAGEMENT SYSTEM



```
b1 = new JButton("NEW");
b2 = new JButton("INSERT");
b3 = new JButton("UPDATE");
b4 = new JButton("VIEW");
b5 = new JButton("DELETE");
b6 = new JButton("LOAD");
```

```
JTable table = new JTable(data,colhead);
table.setForeground(Color.blue);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
```

```
JScrollPane jsp=new JScrollPane(table);
```

```
jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
```

```
p3.add(jsp);
```

```
b1.setToolTipText("Refresh the page");
b2.setToolTipText("Insert the records");
b3.setToolTipText("Update the records");
b4.setToolTipText("View the records");
b5.setToolTipText("Delete the records");
b6.setToolTipText("Load the records");
```

```
b1.setCursor(cor);
b2.setCursor(cor);
b3.setCursor(cor);
b4.setCursor(cor);
b5.setCursor(cor);
b6.setCursor(cor);
```

```
b1.setMnemonic('n');
b2.setMnemonic('i');
b3.setMnemonic('u');
b4.setMnemonic('v');
b5.setMnemonic('d');
b6.setMnemonic('l');
```

```
b1.setFont(new Font("Times New Roman",1,16));
b2.setFont(new Font("Times New Roman",1,16));
b3.setFont(new Font("Times New Roman",1,16));
b4.setFont(new Font("Times New Roman",1,16));
b5.setFont(new Font("Times New Roman",1,16));
b6.setFont(new Font("Times New Roman",1,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
l1.setFont(new Font("Times New Roman",1,16));
l2.setFont(new Font("Times New Roman",1,16));
l3.setFont(new Font("Times New Roman",1,16));
l4.setFont(new Font("Times New Roman",1,16));
l5.setFont(new Font("Times New Roman",1,16));
l6.setFont(new Font("Times New Roman",1,16));
l7.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));
t2.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));
cb2.setFont(new Font("Times New Roman",1,16));
cb3.setFont(new Font("Times New Roman",1,16));
cb4.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(Color.blue);
l2.setForeground(Color.blue);
l3.setForeground(Color.blue);
l4.setForeground(Color.blue);
l5.setForeground(Color.blue);
l6.setForeground(Color.blue);
l7.setForeground(Color.blue);
```

```
t1.setForeground(Color.blue);
t2.setForeground(Color.blue);
```

```
cb1.setForeground(Color.blue);
cb2.setForeground(Color.blue);
cb3.setForeground(Color.blue);
cb4.setForeground(Color.blue);
```

```
t1.addActionListener(new enterListener());
t2.addActionListener(new enterListener());
```

```
b1.addActionListener(new newListener());
```

```
b2.addActionListener(new insertListener());
b3.addActionListener(new updateListener());
b4.addActionListener(new viewListener());
b5.addActionListener(new deleteListener());
b6.addActionListener(new loadListener());
```

```
l1.setBounds(30,30,160,25);
l2.setBounds(30,90,160,25);
```



## COLLEGE MANAGEMENT SYSTEM



```
l3.setBounds(30,150,160,25);  
l4.setBounds(30,210,160,25);  
l5.setBounds(30,270,160,25);
```

```
t1.setBounds(250,30,185,25);  
cb1.setBounds(250,90,185,25);  
ch1.setBounds(250,150,60,25);  
ch2.setBounds(350,150,60,25);  
ch3.setBounds(250,180,60,25);  
ch4.setBounds(350,180,60,25);  
cb2.setBounds(250,210,55,25);  
cb3.setBounds(305,210,60,25);  
cb4.setBounds(365,210,70,25);  
t2.setBounds(250,270,185,25);
```

```
b1.setBounds(500,30,100,30);  
b2.setBounds(500,76,100,30);  
b3.setBounds(500,122,100,30);  
b4.setBounds(500,168,100,30);  
b5.setBounds(500,214,100,30);  
b6.setBounds(500,260,100,30);
```

```
jsp.setBounds(10,300,600,165);
```

```
p3.add(b1);  
p3.add(b2);  
p3.add(b3);  
p3.add(b4);  
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);  
p3.add(l2);  
p3.add(l3);  
p3.add(l4);  
p3.add(l5);
```

```
p3.add(t1);  
p3.add(t2);
```

```
p3.add(cb1);  
p3.add(cb2);  
p3.add(cb3);  
p3.add(cb4);  
p3.add(ch1);  
p3.add(ch2);
```





## COLLEGE MANAGEMENT SYSTEM



```
p3.add(ch3);
p3.add(ch4);

cn.add(p1);
cn.add(p2);
cn.add(p3);
cn.add(p4);

l1.setBorder(edge);
l2.setBorder(edge);
l3.setBorder(edge);
l4.setBorder(edge);
l5.setBorder(edge);

t1.setBorder(edge1);
t2.setBorder(edge1);

setVisible(true);
setSize(1250,573);
setLocation(5,95);

connect();
}

public class newListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            t1.setText(" ");
            t2.setText(" ");

            ch1.setSelected(false);
            ch2.setSelected(false);

            ch3.setSelected(false);
            ch4.setSelected(false);

            cb1.setSelectedIndex(0);
            cb2.setSelectedIndex(0);
            cb3.setSelectedIndex(0);
            cb4.setSelectedIndex(0);
        }
        catch(Exception e1)
        {
        }
```



## COLLEGE MANAGEMENT SYSTEM



```
    }  
    }  
}  
private class insertListener implements ActionListener  
{  
    public void actionPerformed(ActionEvent e)  
    {  
        try  
        {  
            String chh1,chh2,chh3,chh4,dd,mm,yy;  
            chh1 = " ";  
            chh2 = " ";  
            chh3 = " ";  
            chh4 = " ";  
            str1 = t1.getText();  
            str2 = (String) cb1.getSelectedItem();  
            if(ch1.isSelected())  
            {  
                chh1 = ch1.getLabel();  
            }  
            if(ch2.isSelected())  
            {  
                chh2 = ch2.getLabel();  
            }  
  
            if(ch3.isSelected())  
            {  
                chh3 = ch3.getLabel();  
            }  
            if(ch4.isSelected())  
            {  
                chh4 = ch4.getLabel();  
            }  
  
            str3 = chh1+" "+chh2+" "+chh3+" "+chh4;  
  
            dd = (String) cb2.getSelectedItem();  
            mm = (String) cb3.getSelectedItem();  
            yy = (String) cb4.getSelectedItem();  
  
            str4 = dd+"-"+mm+"-"+yy;  
            str5 = t2.getText();
```



## COLLEGE MANAGEMENT SYSTEM



```
stm.executeUpdate("insert into affiliation
values('"+str1+"','"+str2+"','"+str3+"','"+str4+"','"+str5+"')");
stm.executeUpdate("commit");
JOptionPane.showMessageDialog(null,"Record
Inserted");

load();

} //end of try
catch(Exception sql)
{
JOptionPane.showMessageDialog(null,sql);
}
}

}
public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            String chh1,chh2,chh3,chh4,dd,mm,yy;
            chh1 = " ";
            chh2 = " ";
            chh3 = " ";
            chh4 = " ";
            str1 = t1.getText();
            str2 = (String) cb1.getSelectedItem();
            if(ch1.isSelected())
            {
                chh1 = ch1.getLabel();
            }
            if(ch2.isSelected())
            {
                chh2 = ch2.getLabel();
            }
            if(ch3.isSelected())
            {
                chh3 = ch3.getLabel();
            }
            if(ch4.isSelected())
            {
                chh4 = ch4.getLabel();
            }

            str3 = chh1+" "+chh2+" "+chh3+" "+chh4;
```



## COLLEGE MANAGEMENT SYSTEM



```
dd = (String) cb2.getSelectedItemAt();
mm = (String) cb3.getSelectedItemAt();
yy = (String) cb4.getSelectedItemAt();

str4 = dd+"-"+mm+"-"+yy;
str5 = t2.getText();

stm.executeUpdate("update affiliation set afflno = '"+str1+"',afflto =
 '"+str2+"',courses = '"+str3+"',affldt = '"+str4+"',reg = '"+str5+"' where afflno =
 '"+str1+"'");

stm.executeUpdate("commit");

JOptionPane.showMessageDialog(null,"Records Updated ");
load();
}
catch(SQLException sql)
{
    JOptionPane.showMessageDialog(null,"Could not
Updated"+sql);
}
}

public class viewListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("Enter
Affiliation number to be view");
            rs = stm.executeQuery("select * from affiliation
where afflno = '"+st+"'");
            while(rs.next())
            {
                String date,dd,mm,yy,chw;

                t1.setText(rs.getString(1));
                cb1.setSelectedItem(rs.getString(2));
                str1 = rs.getString(3);
                str2 = str1.substring(0,3);
                str3 = str1.substring(4,7);
                str4 = str1.substring(8,11);
                str5 = str1.substring(12,16);
                if(str2.equals("BCA"))
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    ch1.setSelected(true);
}
if(str3.equals("BBA"))
{
    ch2.setSelected(true);
}
if(str4.equals("BSc"))
{
    ch3.setSelected(true);
}
if(str5.equals("BJMC"))
{
    ch4.setSelected(true);
}

date = rs.getString(4);
yy = date.substring(0,4);
mm = date.substring(5,7);

dd = date.substring(8,10);

cb4.setSelectedItem(String.valueOf(yy));
cb3.setSelectedIndex(Integer.parseInt(mm));
cb2.setSelectedItem(dd);

t2.setText(rs.getString(5));
}
}
catch(SQLException sql)
{
    JOptionPane.showMessageDialog(null,"Not found");
}
}

public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("enter Affiliation
number to be deleted");
            stm.executeUpdate("delete from affiliation where
afflno = '"+st+"'");
            stm.executeUpdate("commit");
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
deleted");
        JOptionPane.showMessageDialog(null,"Record is
load();
    }
    catch(SQLException sql)
    {
        JOptionPane.showMessageDialog(null,"Not
deleted");
    }
}
protected void processWindowEvent(WindowEvent we)
{
    if(we.getID()==we.WINDOW_CLOSING)
    {
        r = JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
        if(r == JOptionPane.YES_OPTION)

            dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Exception"+sql);
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not Connected");
    }
}
public class loadListener implements ActionListener
{
```



## COLLEGE MANAGEMENT SYSTEM



```
public void actionPerformed(ActionEvent e)
{
    load();
}

public void load()
{
    int k;
    k=0;
    try
    {
        stm.executeUpdate("commit");
        rs = stm.executeQuery("select
afflno,afflto,courses,to_char(affldt,'dd-mon-yyyy'),reg from affiliation order by
afflno");

        while(rs.next())
        {
            str1 = rs.getString(1);
            str2 = rs.getString(2);
            str3 = rs.getString(3);
            str4 = rs.getString(4);
            str5 = rs.getString(5);

            data[k][0]=str1;
            data[k][1]=str2;
            data[k][2]=str3;
            data[k][3]=str4;
            data[k][4]=str5;

            k++;
        }
        data[k][0] = " ";
        data[k][1] = " ";
        data[k][2] = " ";
        data[k][3] = " ";
        data[k][4] = " ";

        JTable table = new JTable(data,colhead);
        table.setFont(new Font("Times New Roman",Font.BOLD,16));
        table.setForeground(Color.blue);
        JScrollPane jsp = new JScrollPane(table);

        jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
        p3.add(jsp);
```



## COLLEGE MANAGEMENT SYSTEM



```
jsp.setBounds(10,300,600,165);
p3.add(jsp);
}
catch(SQLException sqle)
{
JOptionPane.showMessageDialog(null,"Record not transfer");
}
}
private class enterListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource() == t1)
            t2.requestFocus();
    }
}
private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l6.setLocation(80,r);
        l7.setLocation(90,r+50);
        l8.setLocation(90,r+100);
        l9.setLocation(90,r+150);
        l10.setLocation(90,r+200);
        l11.setLocation(80,r+250);
        l12.setLocation(80,r+300);
        r = r-1;
        if(r<0)
            r = 475;
    }
}
private class timerListener1 implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(1,10);

        l = l - 1;
        if( l < 0)
            l = 1225;
    }
}
public class exitListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
```





## COLLEGE MANAGEMENT SYSTEM



```
    {  
        dispose();  
    }  
}
```



## **/\* Staff Detail \*/**

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class StaffDetail extends JFrame
{
    private JLabel
title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17,l18;
    private JTextField t1,t2,t3,t4,t5,t6,t7;
    private JTextArea ta;
    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;
    private JScrollPane jp;
    private JComboBox cb1,cb2,cb3,cb4,cb5,cb6,cb7,cb8,cb9,cb10;
    private JRadioButton rb1,rb2;
    private ButtonGroup bg;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;

    ResultSet rs;
    String
str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11,str12,str13,str14,str15,str,st;
    Connection connection;
    Statement statement;
    Dimension screensize;
    String[] colhead =
{"eno","ename","addr","city","state","pin","phone","dob","dept","bp","gender","e
mail","doj","mstatus","designation"};
    Object data[][] = new Object[50][15];
    JTable table = new JTable(data,colhead);
    public StaffDetail()
    {
        super("STAFF DETAIL FORM");
        l=1;r=1;

        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);
```



## COLLEGE MANAGEMENT SYSTEM



```
cn.setBackground(new Color(128,128,25));
```

```
EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);  
BevelBorder edge=new BevelBorder(BevelBorder.RAISED);
```

```
p1 = new JPanel();  
p2 = new JPanel();  
p3 = new JPanel();  
p4 = new JPanel();
```

```
p1.setBounds(5,5,1240,50);  
p2.setBounds(5,60,300,400);  
p3.setBounds(310,60,935,470);  
p4.setBounds(5,465,300,65);
```

```
p1.setBackground(new Color(160,220,120));  
p2.setBackground(new Color(112,0,0));  
p3.setBackground(new Color(253,234,170));  
p4.setBackground(new Color(16,220,120));
```

```
title = new JLabel("WELCOME SCREEN FOR STAFF  
DETAIL");
```

```
title.setFont(new Font("Monotype Corsiva",1,40));  
title.setForeground(Color.red);  
p1.add(title);
```

```
Timer tm = new Timer(50,new timerListener());  
tm.start();
```

```
Timer tm1 = new Timer(5,new timerListener1());  
tm1.start();
```

```
pic = new JLabel();  
Icon img = new ImageIcon("123.jpg");  
pic.setIcon(img);  
p2.add(pic);
```

```
p3.setLayout(null);  
p4.setLayout(null);
```

```
p3.setBorder(BorderFactory.createTitledBorder("Enter Information  
for Staff"));
```

```
l1 = new JLabel("EMP NUMBER");  
l2 = new JLabel("EMP NAME");  
l3 = new JLabel("ADDRESS");  
l4 = new JLabel("CITY");
```



## COLLEGE MANAGEMENT SYSTEM



```
l5 = new JLabel("STATE");
l6 = new JLabel("PIN CODE");
l7 = new JLabel("PHONE NO.");
l8 = new JLabel("DATE OF BIRTH");
l9 = new JLabel("DEPARTMENT");
l10 = new JLabel("BEST PAY");
l11 = new JLabel("GENDER");
l12 = new JLabel("EMAIL");
l13 = new JLabel("DATE OF JOINING");
l14 = new JLabel("MARITAL STATUS");
l15 = new JLabel("DESIGNATION");
```

```
l16 = new JLabel("Developed by:-");
l16.setFont(new Font("Times new Roman",1,16));
l16.setForeground(Color.red);
p4.add(l16);
l16.setBounds(0,850,140,25);
```

```
l17=new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.
Tauqueer , Motahhir , Sanoj");
l17.setFont(new Font("Monotype Corsiva",1,15));
l17.setForeground(Color.white);
l17.setBounds(50,850,220,25);
p4.add(l17);
```

```
l18=new JLabel("collegemanagementsystem@gmail.com");
l18.setFont(new Font("Monotype Corsiva",1,15));
l18.setForeground(Color.white);
l18.setBounds(50,850,220,25);
p4.add(l18);
```

```
t1 = new JTextField(15);
t2 = new JTextField(15);
t3 = new JTextField(15);
t4 = new JTextField(15);
t5 = new JTextField(15);
t6 = new JTextField(15);
t7 = new JTextField(15);
```

```
cb1 = new JComboBox();
cb1.addItem("-----SELECT-----");
cb1.addItem("Bihar");
cb1.addItem("Jharkhand");
cb1.addItem("UttarPradesh");
cb1.addItem("Kolkata");
cb1.addItem("Gujarat");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb1.addItem("Haryana");
```

```
cb2 = new JComboBox();
cb2.addItem("DD");
cb2.addItem("01");
cb2.addItem("02");
cb2.addItem("03");
cb2.addItem("04");
cb2.addItem("05");
cb2.addItem("06");
cb2.addItem("07");
cb2.addItem("08");
cb2.addItem("09");
cb2.addItem("10");
cb2.addItem("11");
cb2.addItem("12");
cb2.addItem("13");
cb2.addItem("14");
cb2.addItem("15");
cb2.addItem("16");
cb2.addItem("17");
cb2.addItem("18");
cb2.addItem("19");
cb2.addItem("20");
cb2.addItem("21");
cb2.addItem("22");
cb2.addItem("23");
cb2.addItem("24");
cb2.addItem("25");
cb2.addItem("26");
cb2.addItem("27");
cb2.addItem("28");
cb2.addItem("29");
cb2.addItem("30");
cb2.addItem("31");
```

```
cb3 = new JComboBox();
```

```
cb3.addItem("MM");
cb3.addItem("Jan");
cb3.addItem("Feb");
cb3.addItem("Mar");
cb3.addItem("Apr");
cb3.addItem("May");
cb3.addItem("Jun");
cb3.addItem("Jul");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb3.addItem("Aug");
cb3.addItem("Sep");
cb3.addItem("Oct");
cb3.addItem("Nov");
cb3.addItem("Dec");
```

```
cb4 = new JComboBox();
cb4.addItem("YYYY");
cb4.addItem("1970");
cb4.addItem("1971");
cb4.addItem("1972");
cb4.addItem("1973");
cb4.addItem("1974");
cb4.addItem("1975");
cb4.addItem("1976");
cb4.addItem("1977");
cb4.addItem("1978");
cb4.addItem("1979");
cb4.addItem("1980");
cb4.addItem("1981");
cb4.addItem("1982");
cb4.addItem("1983");
cb4.addItem("1984");
cb4.addItem("1985");
cb4.addItem("1986");
cb4.addItem("1987");
cb4.addItem("1988");
cb4.addItem("1989");
cb4.addItem("1990");
cb4.addItem("1991");
cb4.addItem("1992");
cb4.addItem("1993");
cb4.addItem("1994");
cb4.addItem("1995");
cb4.addItem("1996");
cb4.addItem("1997");
```

```
cb4.addItem("1998");
cb4.addItem("1999");
cb4.addItem("2000");
```

```
cb5 = new JComboBox();
cb5.addItem("-----SELECT-----");
cb5.addItem("Teacher");
cb5.addItem("Office assi");
cb5.addItem("Other");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb6 = new JComboBox();
cb6.addItem("DD");
cb6.addItem("01");
cb6.addItem("02");
cb6.addItem("03");
cb6.addItem("04");
cb6.addItem("05");
cb6.addItem("06");
cb6.addItem("07");
cb6.addItem("08");
cb6.addItem("09");
cb6.addItem("10");
cb6.addItem("11");
cb6.addItem("12");
cb6.addItem("13");
cb6.addItem("14");
cb6.addItem("15");
cb6.addItem("16");
cb6.addItem("17");
cb6.addItem("18");
cb6.addItem("19");
cb6.addItem("20");
cb6.addItem("21");
cb6.addItem("22");
cb6.addItem("23");
cb6.addItem("24");
cb6.addItem("25");
cb6.addItem("26");
cb6.addItem("27");
cb6.addItem("28");
cb6.addItem("29");
cb6.addItem("30");
cb6.addItem("31");
```

```
cb7 = new JComboBox();
cb7.addItem("MM");
cb7.addItem("Jan");
cb7.addItem("Feb");
cb7.addItem("Mar");
cb7.addItem("Apr");
cb7.addItem("May");
cb7.addItem("Jun");
cb7.addItem("Jul");
cb7.addItem("Aug");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb7.addItem("Sep");  
cb7.addItem("Oct");  
cb7.addItem("Nov");  
cb7.addItem("Dec");
```

```
cb8 = new JComboBox();  
cb8.addItem("YYYY");  
cb8.addItem("1970");  
cb8.addItem("1971");  
cb8.addItem("1972");  
cb8.addItem("1973");  
cb8.addItem("1974");  
cb8.addItem("1975");  
cb8.addItem("1976");  
cb8.addItem("1977");  
cb8.addItem("1978");  
cb8.addItem("1979");  
cb8.addItem("1980");  
cb8.addItem("1981");  
cb8.addItem("1982");  
cb8.addItem("1983");  
cb8.addItem("1984");  
cb8.addItem("1985");  
cb8.addItem("1986");  
cb8.addItem("1987");  
cb8.addItem("1988");  
cb8.addItem("1989");  
cb8.addItem("1990");  
cb8.addItem("1991");  
cb8.addItem("1992");  
cb8.addItem("1993");  
cb8.addItem("1994");  
cb8.addItem("1995");  
cb8.addItem("1996");
```

```
cb8.addItem("1997");  
cb8.addItem("1998");  
cb8.addItem("1999");  
cb8.addItem("2000");  
cb8.addItem("2001");  
cb8.addItem("2002");  
cb8.addItem("2003");  
cb8.addItem("2004");  
cb8.addItem("2005");  
cb8.addItem("2006");  
cb8.addItem("2007");
```





## COLLEGE MANAGEMENT SYSTEM



```
cb8.addItem("2008");
cb8.addItem("2009");
cb8.addItem("2010");
cb8.addItem("2011");
cb8.addItem("2012");
cb8.addItem("2013");
cb8.addItem("2014");
cb8.addItem("2015");
cb8.addItem("2016");
cb8.addItem("2017");
```

```
cb9 = new JComboBox();
cb9.addItem( "-----SELECT-----" );
cb9.addItem("Married");
cb9.addItem("Unmarried");
```

```
cb10 = new JComboBox();
cb10.addItem( "-----SELECT-----" );
cb10.addItem("BCA Faculty");
cb10.addItem("Lab Instructor");
cb10.addItem("Office Asst.");
```

```
rb1 = new JRadioButton("Male");
rb2 = new JRadioButton("Female");
bg = new ButtonGroup();
bg.add(rb1);
bg.add(rb2);
```

```
ta = new JTextArea();
ta.setLineWrap(true);
```

```
JScrollPane jp = new
JScrollPane(ta,JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,JScrollPane.H
ORIZONTAL_SCROLLBAR_ALWAYS);
ta.setEditable(true);
```

```
b1 = new JButton("NEW");
b2 = new JButton("INSERT");
b3 = new JButton("UPDATE");
b4 = new JButton("VIEW");
b5 = new JButton("DELETE");
b6 = new JButton("LOAD");
```

```
JTable table = new JTable(data,colhead);
table.setForeground(Color.blue);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
JScrollPane jsp=new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);

b1.setToolTipText("Refresh the page");
b2.setToolTipText("Insert the records");
b3.setToolTipText("Update the records");
b4.setToolTipText("View the records");
b5.setToolTipText("Delete the records");
b6.setToolTipText("Load the records");

b1.setCursor(cor);
b2.setCursor(cor);
b3.setCursor(cor);
b4.setCursor(cor);
b5.setCursor(cor);
b6.setCursor(cor);

b1.setMnemonic('n');
b2.setMnemonic('i');
b3.setMnemonic('u');
b4.setMnemonic('v');
b5.setMnemonic('d');
b6.setMnemonic('l');

b1.setFont(new Font("Times New Roman",1,16));

b2.setFont(new Font("Times New Roman",1,16));
b3.setFont(new Font("Times New Roman",1,16));
b4.setFont(new Font("Times New Roman",1,16));
b5.setFont(new Font("Times New Roman",1,16));
b6.setFont(new Font("Times New Roman",1,16));

i1.setFont(new Font("Times New Roman",1,16));
i2.setFont(new Font("Times New Roman",1,16));
i3.setFont(new Font("Times New Roman",1,16));
i4.setFont(new Font("Times New Roman",1,16));
i5.setFont(new Font("Times New Roman",1,16));
i6.setFont(new Font("Times New Roman",1,16));
i7.setFont(new Font("Times New Roman",1,16));
i8.setFont(new Font("Times New Roman",1,16));
i9.setFont(new Font("Times New Roman",1,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
l10.setFont(new Font("Times New Roman",1,16));  
l11.setFont(new Font("Times New Roman",1,16));  
l12.setFont(new Font("Times New Roman",1,16));  
l13.setFont(new Font("Times New Roman",1,16));  
l14.setFont(new Font("Times New Roman",1,16));  
l15.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));  
t2.setFont(new Font("Times New Roman",1,16));  
t3.setFont(new Font("Times New Roman",1,16));  
t4.setFont(new Font("Times New Roman",1,16));  
t5.setFont(new Font("Times New Roman",1,16));  
t6.setFont(new Font("Times New Roman",1,16));  
t7.setFont(new Font("Times New Roman",1,16));
```

```
ta.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));  
cb2.setFont(new Font("Times New Roman",1,16));  
cb3.setFont(new Font("Times New Roman",1,16));  
cb4.setFont(new Font("Times New Roman",1,16));  
cb5.setFont(new Font("Times New Roman",1,16));  
cb6.setFont(new Font("Times New Roman",1,16));  
cb7.setFont(new Font("Times New Roman",1,16));  
cb8.setFont(new Font("Times New Roman",1,16));  
cb9.setFont(new Font("Times New Roman",1,16));  
cb10.setFont(new Font("Times New Roman",1,16));
```

```
rb1.setFont(new Font("Times New Roman",1,16));  
rb2.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(Color.blue);  
l2.setForeground(Color.blue);  
l3.setForeground(Color.blue);  
l4.setForeground(Color.blue);  
l5.setForeground(Color.blue);  
l6.setForeground(Color.blue);  
l7.setForeground(Color.blue);  
l8.setForeground(Color.blue);  
l9.setForeground(Color.blue);  
l10.setForeground(Color.blue);  
l11.setForeground(Color.blue);  
l12.setForeground(Color.blue);  
l13.setForeground(Color.blue);  
l14.setForeground(Color.blue);  
l15.setForeground(Color.blue);
```



## COLLEGE MANAGEMENT SYSTEM



```
rb1.setForeground(Color.blue);  
rb2.setForeground(Color.blue);
```

```
t1.setForeground(Color.blue);  
t2.setForeground(Color.blue);  
t3.setForeground(Color.blue);  
t4.setForeground(Color.blue);  
t5.setForeground(Color.blue);  
t6.setForeground(Color.blue);  
t7.setForeground(Color.blue);
```

```
ta.setForeground(Color.blue);
```

```
cb1.setForeground(Color.blue);  
cb2.setForeground(Color.blue);  
cb3.setForeground(Color.blue);  
cb4.setForeground(Color.blue);  
cb5.setForeground(Color.blue);  
cb6.setForeground(Color.blue);  
cb7.setForeground(Color.blue);  
cb8.setForeground(Color.blue);  
cb9.setForeground(Color.blue);  
cb10.setForeground(Color.blue);
```

```
t1.addActionListener(new enterListener());
```

```
t2.addActionListener(new enterListener());  
t3.addActionListener(new enterListener());  
t4.addActionListener(new enterListener());  
t5.addActionListener(new enterListener());  
t6.addActionListener(new enterListener());  
t7.addActionListener(new enterListener());  
b1.addActionListener(new newListener());  
b2.addActionListener(new insertListener());  
b3.addActionListener(new updateListener());  
b4.addActionListener(new viewListener());  
b5.addActionListener(new deleteListener());  
b6.addActionListener(new loadListener());
```

```
l1.setBounds(30,30,120,25);  
l2.setBounds(30,60,120,25);  
l3.setBounds(30,90,120,25);  
l4.setBounds(30,120,120,25);  
l5.setBounds(30,150,120,25);  
l6.setBounds(30,180,120,25);
```



## COLLEGE MANAGEMENT SYSTEM



```
17.setBounds(400,30,145,25);
18.setBounds(400,60,145,25);
19.setBounds(400,90,145,25);
110.setBounds(400,120,145,25);
111.setBounds(400,150,145,25);
112.setBounds(400,180,145,25);
113.setBounds(400,210,145,25);
114.setBounds(400,240,145,25);
115.setBounds(400,270,145,25);
```

```
t1.setBounds(160,30,185,25);
t2.setBounds(160,60,185,25);
jp.setBounds(160,90,185,115);
t3.setBounds(160,210,185,25);
t4.setBounds(160,270,185,25);
t5.setBounds(560,30,185,25);
t6.setBounds(560,120,185,25);
t7.setBounds(560,180,185,25);
```

```
cb1.setBounds(160,240,185,25);
cb2.setBounds(560,60,55,25);
cb3.setBounds(615,60,60,25);
cb4.setBounds(675,60,70,25);
```

```
cb5.setBounds(560,90,185,25);
cb6.setBounds(560,210,55,25);
cb7.setBounds(615,210,60,25);
cb8.setBounds(675,210,70,25);
cb9.setBounds(560,240,185,25);
cb10.setBounds(560,270,185,25);
```

```
rb1.setBounds(560,150,92,25);
rb2.setBounds(652,150,92,25);
```

```
b1.setBounds(780,30,100,30);
b2.setBounds(780,76,100,30);
b3.setBounds(780,122,100,30);
b4.setBounds(780,168,100,30);
b5.setBounds(780,214,100,30);
b6.setBounds(780,260,100,30);
```

```
jsp.setBounds(12,305,910,150);
```

```
p3.add(b1);
p3.add(b2);
p3.add(b3);
```



## COLLEGE MANAGEMENT SYSTEM



```
p3.add(b4);  
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);  
p3.add(l2);  
p3.add(l3);  
p3.add(l4);  
p3.add(l5);  
p3.add(l6);  
p3.add(l7);  
p3.add(l8);  
p3.add(l9);  
p3.add(l10);  
p3.add(l11);  
p3.add(l12);  
p3.add(l13);  
p3.add(l14);  
p3.add(l15);
```

```
p3.add(t1);
```

```
p3.add(t2);  
p3.add(t3);  
p3.add(t4);  
p3.add(t5);  
p3.add(t6);  
p3.add(t7);
```

```
p3.add(jp);
```

```
p3.add(cb1);  
p3.add(cb2);  
p3.add(cb3);  
p3.add(cb4);  
p3.add(cb5);  
p3.add(cb6);  
p3.add(cb7);  
p3.add(cb8);  
p3.add(cb9);  
p3.add(cb10);
```

```
p3.add(rb1);  
p3.add(rb2);
```



## COLLEGE MANAGEMENT SYSTEM



```
cn.add(p1);  
cn.add(p2);  
cn.add(p3);  
cn.add(p4);
```

```
l1.setBorder(edge);  
l2.setBorder(edge);  
l3.setBorder(edge);  
l4.setBorder(edge);  
l5.setBorder(edge);  
l6.setBorder(edge);  
l7.setBorder(edge);  
l8.setBorder(edge);  
l9.setBorder(edge);  
l10.setBorder(edge);  
l11.setBorder(edge);  
l12.setBorder(edge);  
l13.setBorder(edge);  
l14.setBorder(edge);  
l15.setBorder(edge);
```

```
t1.setBorder(edge1);  
t2.setBorder(edge1);  
t3.setBorder(edge1);  
t4.setBorder(edge1);  
t5.setBorder(edge1);  
t6.setBorder(edge1);  
t7.setBorder(edge1);
```

```
setVisible(true);  
setSize(1265,570);  
setLocation(5,95);  
connect();
```

```
}  
public class newListener implements ActionListener  
{  
    public void actionPerformed(ActionEvent e)  
    {  
        try  
        {  
            t1.setText(" ");  
            t2.setText(" ");  
            t3.setText(" ");  
            t4.setText(" ");  
            t5.setText(" ");
```



## COLLEGE MANAGEMENT SYSTEM



```
t6.setText(" ");
t7.setText(" ");

ta.setText(" ");

cb1.setSelectedIndex(0);
cb2.setSelectedIndex(0);
cb3.setSelectedIndex(0);
cb4.setSelectedIndex(0);
cb5.setSelectedIndex(0);
cb6.setSelectedIndex(0);
cb7.setSelectedIndex(0);
cb8.setSelectedIndex(0);
cb9.setSelectedIndex(0);
cb10.setSelectedIndex(0);

rb1.setSelected(true);
    }
    catch(Exception e1)
    {
    }
}
}
private class insertListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        //k=0;
        try
        {
            str1 = t1.getText();
            if (str1.compareTo("") == 0)

OptionPane.showMessageDialog(null,"Employee no. can't be null");
            else
            {
                String dd,mm,yyyy;
                str1 = t1.getText();
                str2 = t2.getText();
                str3 = ta.getText();
                str4 = t3.getText();
                str5 = (String) cb1.getSelectedItem();
                str6 = t4.getText();
                str7 = t5.getText();

                dd = (String) cb2.getSelectedItem();
```





## COLLEGE MANAGEMENT SYSTEM



```
mm = (String) cb3.getSelectedItem();
yyyy = (String) cb4.getSelectedItem();
str8 = dd+"-"+mm+"-"+yyyy;
str9 = (String) cb5.getSelectedItem();
str10 = t6.getText();
if(rb1.isSelected())
{
    str11 = "Male";
}
if(rb2.isSelected())
{
    str11 = "Female";
}
str12 = t7.getText();
dd = (String) cb6.getSelectedItem();
mm = (String) cb7.getSelectedItem();

yyyy = (String) cb8.getSelectedItem();
str13 = dd+"-"+mm+"-"+yyyy;
str14 = (String) cb9.getSelectedItem();
str15 = (String) cb10.getSelectedItem();

statement.executeUpdate("insert into staffdetail
values("+str1+", '"+str2+"', '"+str3+"', '"+str4+"', '"+str5+"', '"+str6+"', '"+str7+"', '"+str8
+'', '"+str9+"', '"+str10+"', '"+str11+"', '"+str12+"', '"+str13+"', '"+str14+"', '"+str15+"')");
;

statement.executeUpdate("commit");
JOptionPane.showMessageDialog(null, "Record
Inserted");

load();
} //end of else
} //end of try
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null, sql);
}
}

public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            if(t1.getText().compareTo("")==0)
```



## COLLEGE MANAGEMENT SYSTEM



```
Fields");
JOptionPane.showMessageDialog(null,"Fill all
else
{
    String dd,mm,yyyy;
    str1 = t1.getText();
    str2 = t2.getText();
    str3 = ta.getText();
    str4 = t3.getText();
    str5 = (String) cb1.getSelectedItem();
    str6 = t4.getText();
    str7 = t5.getText();

    dd = (String) cb2.getSelectedItem();
    mm = (String) cb3.getSelectedItem();
    yyyy = (String) cb4.getSelectedItem();

    str8 = dd+"-"+mm+"-"+yyyy;
    str9 = (String) cb5.getSelectedItem();
    str10 = t6.getText();
    if(rb1.isSelected())
    {
        str11 = "Male";
    }
    if(rb2.isSelected())
    {
        str11 = "Female";
    }
    str12 = t7.getText();
    dd = (String) cb6.getSelectedItem();
    mm = (String) cb7.getSelectedItem();
    yyyy = (String) cb8.getSelectedItem();
    str13 = dd+"-"+mm+"-"+yyyy;
    str14 = (String) cb9.getSelectedItem();
    str15 = (String) cb10.getSelectedItem();

    statement.executeUpdate("update staffdetail
set eno = "+str1+",ename = '"+str2+"',address = '"+str3+"',city = '"+str4+"',state =
 '"+str5+"',pin = '"+str6+"',phone = '"+str7+"',dob = '"+str8+"',dept = '"+str9+"',bp =
 '"+str10+"',gender = '"+str11+"',email = '"+str12+"',doj = '"+str13+"',mstatus =
 '"+str14+"',designation = '"+str15+" where eno = "+str1);
    statement.executeUpdate("commit");

JOptionPane.showMessageDialog(null,"Records Updated ");
load();
}
```



## COLLEGE MANAGEMENT SYSTEM



```
    }
    catch(SQLException sql)
    {
        JOptionPane.showMessageDialog(null,"Could not
Updated"+sql);
    }
}
}
public class viewListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("Enter Employee
number to be view");
            rs = statement.executeQuery("select * from
staffdetail where eno = "+st);
            while(rs.next())
            {
                String dd,mm,yy,dob,gender;

                t1.setText(rs.getString(1));
                t2.setText(rs.getString(2));
                ta.setText(rs.getString(3));
                t3.setText(rs.getString(4));
                cb1.setSelectedItem(rs.getString(5));
                t4.setText(rs.getString(6));
                t5.setText(rs.getString(7));

                dob = rs.getString(8);
                yy = dob.substring(0,4);
                mm = dob.substring(5,7);

                dd = dob.substring(8,10);
                cb4.setSelectedItem(String.valueOf(yy));
                cb3.setSelectedIndex(Integer.parseInt(mm));
                cb2.setSelectedItem(dd);

                cb5.setSelectedItem(rs.getString(9));
                t6.setText(rs.getString(10));
                gender = rs.getString(11).trim();
                if(gender.equals("Male"))
                {
                    rb1.setSelected(true);
                }
            }
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        if(gender.equals("Female"))
        {
            rb2.setSelected(true);
        }
        t7.setText(rs.getString(12));
        dob = rs.getString(13);
        yy = dob.substring(0,4);
        mm = dob.substring(5,7);

        dd = dob.substring(8,10);
        cb8.setSelectedItem(String.valueOf(yy));
        cb7.setSelectedIndex(Integer.parseInt(mm));

        cb6.setSelectedItem(dd);

        cb9.setSelectedItem(rs.getString(14));
        cb10.setSelectedItem(rs.getString(15));

    }
}
catch(SQLException sql)
{
    JOptionPane.showMessageDialog(null,"Not found");
}
}

}
public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("enter Employee
number to be deleted");
            statement.executeUpdate("delete from staffdetail
where eno = '"+st+"'");
            statement.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"Record is
deleted");
            load();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not
deleted");
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
    }
}
protected void processWindowEvent(WindowEvent we)
{
    if(we.getID()==we.WINDOW_CLOSING)
    {
        r=JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
        if(r == JOptionPane.YES_OPTION)
            dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            connection =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            statement = connection.createStatement();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Exception"+sql);
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not Connected");
    }
}
public class loadListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            load();
        }
        catch(Exception e1)
        {
            JOptionPane.showMessageDialog(null,"Exception"+e1);
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"Exception"+e1);
    }
}
public void load()
{
    int k;
    k=0;
    try
    {
        statement.executeUpdate("commit");
        rs = statement.executeQuery("select
eno,ename,address,city,state,pin,phone,to_char(dob,'dd-mon-
yyyy'),dept,bp,gender,email,to_char(doj,'dd-mon-yyyy'),mstatus,designation from
staffdetail order by eno");

        while(rs.next())
        {
            str1 = rs.getString(1);
            str2 = rs.getString(2);
            str3 = rs.getString(3);
            str4 = rs.getString(4);
            str5 = rs.getString(5);
            str6 = rs.getString(6);
            str7 = rs.getString(7);
            str8 = rs.getString(8);
            str9 = rs.getString(9);
            str10 = rs.getString(10);
            str11 = rs.getString(11);
            str12 = rs.getString(12);
            str13 = rs.getString(13);
            str14 = rs.getString(14);
            str15 = rs.getString(15);

            data[k][0]=str1;
            data[k][1]=str2;
            data[k][2]=str3;
            data[k][3]=str4;
            data[k][4]=str5;
            data[k][5]=str6;
            data[k][6]=str7;
            data[k][7]=str8;
            data[k][8]=str9;
            data[k][9]=str10;
```



## COLLEGE MANAGEMENT SYSTEM



```
data[k][10]=str11;
data[k][11]=str12;
data[k][12]=str13;
data[k][13]=str14;
data[k][14]=str15;

k++;
}
data[k][0] = " ";

data[k][1] = " ";
data[k][2] = " ";
data[k][3] = " ";
data[k][4] = " ";
data[k][5] = " ";
data[k][6] = " ";
data[k][7] = " ";
data[k][8] = " ";
data[k][9] = " ";
data[k][10] = " ";
data[k][11] = " ";
data[k][12] = " ";
data[k][13] = " ";
data[k][14] = " ";

JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
table.setForeground(Color.blue);
JScrollPane jsp = new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);
jsp.setBounds(12,305,910,150);
p3.add(jsp);
}
catch(SQLException sqle)
{
JOptionPane.showMessageDialog(null,"Record not transfer");
}
}
private class enterListener implements ActionListener
{
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==t1)
```



## COLLEGE MANAGEMENT SYSTEM



```
        t2.requestFocus();
    if(e.getSource()==t2)
        t3.requestFocus();
    if(e.getSource()==t3)
        t4.requestFocus();
    if(e.getSource()==t4)
        t5.requestFocus();
    if(e.getSource()==t5)
        t6.requestFocus();
    if(e.getSource()==t6)
        t7.requestFocus();
    }
}

private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l18.setLocation(r,40);
        l17.setLocation(r,20);
        l16.setLocation(r,0);
        r = r+1;
        if(r>290)
            r = 0;
    }
}

private class timerListener1 implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(l,10);
        l = l - 1;
        if( l < 0)
            l = 1240;
    }
}

public class exitListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        dispose();
    }
}
}
```





## **/\* Student Detail \*/**

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class StudentDetail extends JFrame
{
    private JLabel
title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17,l18;
    private JTextField t1,t2,t3,t4,t5,t6,t7,t8,t9;
    private JTextArea ta1,ta2;
    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;
    private JScrollPane jp1,jp2;
    private JComboBox cb1,cb2,cb3,cb4,cb5,cb6,cb7;
    private JRadioButton rb1,rb2;
    private ButtonGroup bg;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;
    ResultSet rs;
    String
str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11,str12,str13,str14,str15,str,st;
    Connection con;
    Statement stm;
    Dimension screenSize;
    String[] colhead = {"Reg no.,"Roll","Student name","Father's
name","Mother's name","Date of birth","Gender","Enroll course","Enroll
date","Enroll session","Temporary address ","Permanent address","Contact
no.,"Email","Remarks"};
    Object data[][] = new Object[50][15];
    JTable table = new JTable(data,colhead);
    public StudentDetail()
    {
        super("STUDENT DETAIL FORM");
        l=1;r=1;

        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);
```



## COLLEGE MANAGEMENT SYSTEM



```
cn.setBackground(new Color(128,128,25));
```

```
EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);  
BevelBorder edge=new BevelBorder(BevelBorder.RAISED);
```

```
p1 = new JPanel();  
p2 = new JPanel();  
p3 = new JPanel();  
p4 = new JPanel();
```

```
p1.setBounds(5,5,1240,50);  
p2.setBounds(5,60,305,400);  
p3.setBounds(315,60,930,470);  
p4.setBounds(5,465,305,60);
```

```
p1.setBackground(new Color(61,0,0));  
p2.setBackground(new Color(38,24,21));  
p3.setBackground(new Color(66,45,219));  
p4.setBackground(new Color(31,58,4));
```

```
title = new JLabel("WELCOME SCREEN FOR STUDENT  
DETAIL");
```

```
title.setFont(new Font("Monotype Corsiva",1,40));  
title.setForeground(Color.red);  
p1.add(title);
```

```
Timer tm = new Timer(50,new timerListener());  
tm.start();
```

```
Timer tm1 = new Timer(5,new timerListener1());  
tm1.start();
```

```
pic = new JLabel();  
Icon img=new ImageIcon("student.jpg");  
pic.setIcon(img);  
p2.add(pic);
```

```
p3.setLayout(null);  
p4.setLayout(null);
```

```
p3.setBorder(BorderFactory.createTitledBorder("Enter Information  
for Student"));
```

```
l1 = new JLabel("REG NUMBER");  
l2 = new JLabel("ROLL NUMBER");
```



## COLLEGE MANAGEMENT SYSTEM



```
l3 = new JLabel("STUDENT NAME");
l4 = new JLabel("FATHER NAME");
l5 = new JLabel("MOTHER NAME");
l6 = new JLabel("DATE OF BIRTH");
l7 = new JLabel("GENDER");
l8 = new JLabel("ENROLL COURSE");
l9 = new JLabel("ENROLL DATE");
l10 = new JLabel("ENROLL SESSION");
l11 = new JLabel("TEMP ADDRESS");
l12 = new JLabel("PARM ADDRESS");
l13 = new JLabel("CONTACT");
l14 = new JLabel("EMAIL");
l15 = new JLabel("REMARKS");

l16 = new JLabel("Developed by:-");
l16.setFont(new Font("Times new Roman",1,16));
l16.setForeground(Color.red);
p4.add(l16);
l16.setBounds(0,850,140,25);
```

```
l17=new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.
Tauqueer , Motahhir , Sanoj");
l17.setFont(new Font("Monotype Corsiva",1,15));
l17.setForeground(Color.white);
l17.setBounds(50,850,220,25);
p4.add(l17);
```

```
l18=new JLabel("collegemanagementsystem@gmail.com");
l18.setFont(new Font("Monotype Corsiva",1,15));
l18.setForeground(Color.white);
l18.setBounds(50,850,220,25);
p4.add(l18);
```

```
t1 = new JTextField(15);
t2 = new JTextField(15);
t3 = new JTextField(15);
t4 = new JTextField(15);
t5 = new JTextField(15);
t6 = new JTextField(15);
t7 = new JTextField(15);
```

```
t8 = new JTextField(15);
t9 = new JTextField(15);
```

```
cb1 = new JComboBox();
cb1.addItem("DD");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb1.addItem("01");
cb1.addItem("02");
cb1.addItem("03");
cb1.addItem("04");
cb1.addItem("05");
cb1.addItem("06");
cb1.addItem("07");
cb1.addItem("08");
cb1.addItem("09");
cb1.addItem("10");
cb1.addItem("11");
cb1.addItem("12");
cb1.addItem("13");
cb1.addItem("14");
cb1.addItem("15");
cb1.addItem("16");
cb1.addItem("17");
cb1.addItem("18");
cb1.addItem("19");
cb1.addItem("20");
cb1.addItem("21");
cb1.addItem("22");
cb1.addItem("23");
cb1.addItem("24");
cb1.addItem("25");
cb1.addItem("26");
cb1.addItem("27");
cb1.addItem("28");
cb1.addItem("29");
cb1.addItem("30");
cb1.addItem("31");
```

```
cb2 = new JComboBox();
cb2.addItem("MM");
cb2.addItem("Jan");
cb2.addItem("Feb");
cb2.addItem("Mar");
cb2.addItem("Apr");
cb2.addItem("May");
```

```
cb2.addItem("Jun");
cb2.addItem("Jul");
cb2.addItem("Aug");
cb2.addItem("Sep");
cb2.addItem("Oct");
cb2.addItem("Nov");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb2.addItem("Dec");

cb3 = new JComboBox();
cb3.addItem("YYYY");
cb3.addItem("1970");
cb3.addItem("1971");
cb3.addItem("1972");
cb3.addItem("1973");
cb3.addItem("1974");
cb3.addItem("1975");
cb3.addItem("1976");
cb3.addItem("1977");
cb3.addItem("1978");
cb3.addItem("1979");
cb3.addItem("1980");
cb3.addItem("1981");
cb3.addItem("1982");
cb3.addItem("1983");
cb3.addItem("1984");
cb3.addItem("1985");
cb3.addItem("1986");
cb3.addItem("1987");
cb3.addItem("1988");
cb3.addItem("1989");
cb3.addItem("1990");
cb3.addItem("1991");
cb3.addItem("1992");
cb3.addItem("1993");
cb3.addItem("1994");
cb3.addItem("1995");
cb3.addItem("1996");
cb3.addItem("1997");
cb3.addItem("1998");
cb3.addItem("1999");
cb3.addItem("2000");
cb4 = new JComboBox();
cb4.addItem("-----SELECT-----");
cb4.addItem("BCA");
cb4.addItem("BBA");
cb4.addItem("BBM");
cb4.addItem("MCA");
cb4.addItem("MBA");

cb5 = new JComboBox();
cb5.addItem("DD");
cb5.addItem("01");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb5.addItem("02");
cb5.addItem("03");
cb5.addItem("04");
cb5.addItem("05");
cb5.addItem("06");
cb5.addItem("07");
cb5.addItem("08");
cb5.addItem("09");
cb5.addItem("10");
cb5.addItem("11");
cb5.addItem("12");
cb5.addItem("13");
cb5.addItem("14");
cb5.addItem("15");
cb5.addItem("16");
cb5.addItem("17");
cb5.addItem("18");
cb5.addItem("19");
cb5.addItem("20");
cb5.addItem("21");
cb5.addItem("22");
cb5.addItem("23");
cb5.addItem("24");
cb5.addItem("25");
cb5.addItem("26");
cb5.addItem("27");
cb5.addItem("28");
cb5.addItem("29");
cb5.addItem("30");
cb5.addItem("31");
```

```
cb6 = new JComboBox();
cb6.addItem("MM");
cb6.addItem("Jan");
```

```
cb6.addItem("Feb");
cb6.addItem("Mar");
cb6.addItem("Apr");
cb6.addItem("May");
cb6.addItem("Jun");
cb6.addItem("Jul");
cb6.addItem("Aug");
cb6.addItem("Sep");
cb6.addItem("Oct");
cb6.addItem("Nov");
cb6.addItem("Dec");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb7 = new JComboBox();
cb7.addItem("YYYY");
cb7.addItem("1970");
cb7.addItem("1971");
cb7.addItem("1972");
cb7.addItem("1973");
cb7.addItem("1974");
cb7.addItem("1975");
cb7.addItem("1976");
cb7.addItem("1977");
cb7.addItem("1978");
cb7.addItem("1979");
cb7.addItem("1980");
cb7.addItem("1981");
cb7.addItem("1982");
cb7.addItem("1983");
cb7.addItem("1984");
cb7.addItem("1985");
cb7.addItem("1986");
cb7.addItem("1987");
cb7.addItem("1988");
cb7.addItem("1989");
cb7.addItem("1990");
cb7.addItem("1991");
cb7.addItem("1992");
cb7.addItem("1993");
cb7.addItem("1994");
cb7.addItem("1995");
cb7.addItem("1996");
cb7.addItem("1997");
cb7.addItem("1998");
cb7.addItem("1999");

cb7.addItem("2000");
cb7.addItem("2001");
cb7.addItem("2002");
cb7.addItem("2003");
cb7.addItem("2004");
cb7.addItem("2005");
cb7.addItem("2006");
cb7.addItem("2007");
cb7.addItem("2008");
cb7.addItem("2009");
cb7.addItem("2010");
cb7.addItem("2011");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb7.addItem("2012");
cb7.addItem("2013");
cb7.addItem("2014");
cb7.addItem("2015");
cb7.addItem("2016");
cb7.addItem("2017");

rb1 = new JRadioButton("Male");
rb2 = new JRadioButton("Female");
bg = new ButtonGroup();
bg.add(rb1);
bg.add(rb2);

ta1 = new JTextArea();
ta1.setLineWrap(true);
JScrollPane jp1 = new
JScrollPane(ta1,JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,JScrollPane.
HORIZONTAL_SCROLLBAR_ALWAYS);
ta1.setEditable(true);

ta2 = new JTextArea();
ta2.setLineWrap(true);
JScrollPane jp2 = new
JScrollPane(ta2,JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,JScrollPane.
HORIZONTAL_SCROLLBAR_ALWAYS);
ta2.setEditable(true);

b1 = new JButton("NEW");
b2 = new JButton("INSERT");
b3 = new JButton("UPDATE");
b4 = new JButton("VIEW");
b5 = new JButton("DELETE");
b6 = new JButton("LOAD");

JTable table = new JTable(data,colhead);
table.setForeground(Color.blue);
table.setFont(new Font("Times New Roman",Font.BOLD,16));

JScrollPane jsp=new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);

b1.setToolTipText("Refresh the page");
b2.setToolTipText("Insert the records");
```





## COLLEGE MANAGEMENT SYSTEM



```
b3.setToolTipText("Update the records");  
b4.setToolTipText("View the records");  
b5.setToolTipText("Delete the records");  
b6.setToolTipText("Load the records");
```

```
b1.setCursor(cor);  
b2.setCursor(cor);  
b3.setCursor(cor);  
b4.setCursor(cor);  
b5.setCursor(cor);  
b6.setCursor(cor);
```

```
b1.setMnemonic('n');  
b2.setMnemonic('i');  
b3.setMnemonic('u');  
b4.setMnemonic('v');  
b5.setMnemonic('d');  
b6.setMnemonic('l');
```

```
b1.setFont(new Font("Times New Roman",1,16));  
b2.setFont(new Font("Times New Roman",1,16));  
b3.setFont(new Font("Times New Roman",1,16));  
b4.setFont(new Font("Times New Roman",1,16));  
b5.setFont(new Font("Times New Roman",1,16));  
b6.setFont(new Font("Times New Roman",1,16));
```

```
l1.setFont(new Font("Times New Roman",1,16));  
l2.setFont(new Font("Times New Roman",1,16));  
l3.setFont(new Font("Times New Roman",1,16));
```

```
l4.setFont(new Font("Times New Roman",1,16));  
l5.setFont(new Font("Times New Roman",1,16));  
l6.setFont(new Font("Times New Roman",1,16));  
l7.setFont(new Font("Times New Roman",1,16));  
l8.setFont(new Font("Times New Roman",1,16));  
l9.setFont(new Font("Times New Roman",1,16));  
l10.setFont(new Font("Times New Roman",1,16));  
l11.setFont(new Font("Times New Roman",1,16));  
l12.setFont(new Font("Times New Roman",1,16));  
l13.setFont(new Font("Times New Roman",1,16));  
l14.setFont(new Font("Times New Roman",1,16));  
l15.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));  
t2.setFont(new Font("Times New Roman",1,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
t3.setFont(new Font("Times New Roman",1,16));
t4.setFont(new Font("Times New Roman",1,16));
t5.setFont(new Font("Times New Roman",1,16));
t6.setFont(new Font("Times New Roman",1,16));
t7.setFont(new Font("Times New Roman",1,16));
t8.setFont(new Font("Times New Roman",1,16));
t9.setFont(new Font("Times New Roman",1,16));
```

```
ta1.setFont(new Font("Times New Roman",1,16));
ta2.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));
cb2.setFont(new Font("Times New Roman",1,16));
cb3.setFont(new Font("Times New Roman",1,16));
cb4.setFont(new Font("Times New Roman",1,16));
cb5.setFont(new Font("Times New Roman",1,16));
cb6.setFont(new Font("Times New Roman",1,16));
cb7.setFont(new Font("Times New Roman",1,16));
```

```
rb1.setFont(new Font("Times New Roman",1,16));
rb2.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(Color.green);
l2.setForeground(Color.green);
l3.setForeground(Color.green);
l4.setForeground(Color.green);
l5.setForeground(Color.green);
l6.setForeground(Color.green);
l7.setForeground(Color.green);
```

```
l8.setForeground(Color.green);
l9.setForeground(Color.green);
l10.setForeground(Color.green);
l11.setForeground(Color.green);
l12.setForeground(Color.green);
l13.setForeground(Color.green);
l14.setForeground(Color.green);
l15.setForeground(Color.green);
```

```
rb1.setForeground(Color.blue);
rb2.setForeground(Color.blue);
```

```
t1.setForeground(Color.blue);
t2.setForeground(Color.blue);
t3.setForeground(Color.blue);
t4.setForeground(Color.blue);
```



## COLLEGE MANAGEMENT SYSTEM



```
t5.setForeground(Color.blue);
t6.setForeground(Color.blue);
t7.setForeground(Color.blue);
t8.setForeground(Color.blue);
t9.setForeground(Color.blue);
```

```
ta1.setForeground(Color.blue);
ta2.setForeground(Color.blue);
```

```
cb1.setForeground(Color.blue);
cb2.setForeground(Color.blue);
cb3.setForeground(Color.blue);
cb4.setForeground(Color.blue);
cb5.setForeground(Color.blue);
cb6.setForeground(Color.blue);
cb7.setForeground(Color.blue);
```

```
t1.addActionListener(new enterListener());
t2.addActionListener(new enterListener());
t3.addActionListener(new enterListener());
t4.addActionListener(new enterListener());
t5.addActionListener(new enterListener());
t6.addActionListener(new enterListener());
t7.addActionListener(new enterListener());
t8.addActionListener(new enterListener());
t9.addActionListener(new enterListener());
```

```
b1.addActionListener(new newListener());
b2.addActionListener(new insertListener());
b3.addActionListener(new updateListener());
b4.addActionListener(new viewListener());
b5.addActionListener(new deleteListener());
b6.addActionListener(new loadListener());
```

```
l1.setBounds(30,30,145,25);
l2.setBounds(30,60,145,25);
l3.setBounds(30,90,145,25);
l4.setBounds(30,120,145,25);
l5.setBounds(30,150,145,25);
l6.setBounds(30,180,145,25);
l7.setBounds(30,210,145,25);
l8.setBounds(30,240,145,25);
l9.setBounds(30,270,145,25);
l10.setBounds(30,300,145,25);
l11.setBounds(400,30,145,25);
l12.setBounds(400,120,145,25);
```



## COLLEGE MANAGEMENT SYSTEM



```
l13.setBounds(400,240,145,25);  
l14.setBounds(400,270,145,25);  
l15.setBounds(400,300,145,25);
```

```
t1.setBounds(190,30,185,25);  
t2.setBounds(190,60,185,25);  
t3.setBounds(190,90,185,25);  
t4.setBounds(190,120,185,25);  
t5.setBounds(190,150,185,25);  
t6.setBounds(190,300,185,25);  
t7.setBounds(560,240,185,25);  
t8.setBounds(560,270,185,25);  
t9.setBounds(560,300,185,25);
```

```
jp1.setBounds(560,30,185,80);  
jp2.setBounds(560,120,185,115);
```

```
cb1.setBounds(190,180,55,25);  
cb2.setBounds(245,180,60,25);  
cb3.setBounds(305,180,70,25);  
cb4.setBounds(190,240,185,25);  
cb5.setBounds(190,270,55,25);  
cb6.setBounds(245,270,60,25);  
cb7.setBounds(305,270,70,25);
```

```
rb1.setBounds(190,210,92,25);  
rb2.setBounds(282,210,92,25);
```

```
b1.setBounds(780,30,100,30);  
b2.setBounds(780,76,100,30);  
b3.setBounds(780,122,100,30);  
b4.setBounds(780,168,100,30);  
b5.setBounds(780,214,100,30);  
b6.setBounds(780,260,100,30);
```

```
jsp.setBounds(12,335,910,125);
```

```
p3.add(b1);  
p3.add(b2);  
p3.add(b3);  
p3.add(b4);  
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);  
p3.add(l2);
```



## COLLEGE MANAGEMENT SYSTEM



```
p3.add(l3);  
p3.add(l4);  
p3.add(l5);  
p3.add(l6);  
p3.add(l7);  
p3.add(l8);  
p3.add(l9);  
p3.add(l10);  
p3.add(l11);  
p3.add(l12);  
p3.add(l13);  
p3.add(l14);  
p3.add(l15);
```

```
p3.add(t1);  
p3.add(t2);  
p3.add(t3);  
p3.add(t4);  
p3.add(t5);  
p3.add(t6);  
p3.add(t7);  
p3.add(t8);  
p3.add(t9);
```

```
p3.add(jp1);  
p3.add(jp2);
```

```
p3.add(cb1);  
p3.add(cb2);  
p3.add(cb3);  
p3.add(cb4);  
p3.add(cb5);  
p3.add(cb6);  
p3.add(cb7);
```

```
p3.add(rb1);  
p3.add(rb2);
```

```
cn.add(p1);  
cn.add(p2);  
cn.add(p3);  
cn.add(p4);
```

```
l1.setBorder(edge);  
l2.setBorder(edge);  
l3.setBorder(edge);
```



## COLLEGE MANAGEMENT SYSTEM



```
l4.setBorder(edge);
l5.setBorder(edge);
l6.setBorder(edge);
l7.setBorder(edge);
l8.setBorder(edge);
l9.setBorder(edge);
l10.setBorder(edge);

l11.setBorder(edge);
l12.setBorder(edge);
l13.setBorder(edge);
l14.setBorder(edge);
l15.setBorder(edge);

t1.setBorder(edge1);
t2.setBorder(edge1);
t3.setBorder(edge1);
t4.setBorder(edge1);
t5.setBorder(edge1);
t6.setBorder(edge1);
t7.setBorder(edge1);
t8.setBorder(edge1);
t9.setBorder(edge1);
ta1.setBorder(edge1);
ta2.setBorder(edge1);

setVisible(true);
setSize(1265,570);
setLocation(5,95);

connect();
}
public class newListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            t1.setText(" ");
            t2.setText(" ");
            t3.setText(" ");
            t4.setText(" ");
            t5.setText(" ");
            t6.setText(" ");
            t7.setText(" ");
            t8.setText(" ");
```



## COLLEGE MANAGEMENT SYSTEM



```
t9.setText(" ");

ta1.setText(" ");
ta2.setText(" ");

cb1.setSelectedIndex(0);

cb2.setSelectedIndex(0);
cb3.setSelectedIndex(0);
cb4.setSelectedIndex(0);
cb5.setSelectedIndex(0);
cb6.setSelectedIndex(0);
cb7.setSelectedIndex(0);

rb1.setSelected(true);

    }
    catch(Exception e1)
    {
    }
}

private class insertListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = t1.getText();
            if (str1.compareTo("") == 0)

OptionPane.showMessageDialog(null,"Employee no.  can't be null");
            else
            {
                String dd,mm,yy;
                str1 = t1.getText();
                str2 = t2.getText();
                str3 = t3.getText();
                str4 = t4.getText();
                str5 = t5.getText();
                dd = (String) cb1.getSelectedItem();
                mm = (String) cb2.getSelectedItem();
                yy = (String) cb3.getSelectedItem();
                str6 = dd+"-"+mm+"-"+yy;
                if(rb1.isSelected())
                    str7 = "Male";
```



## COLLEGE MANAGEMENT SYSTEM



```
        if(rb2.isSelected())
            str7 = "Female";
        str8 = (String) cb4.getSelectedItem();

        dd = (String) cb5.getSelectedItem();

        mm = (String) cb6.getSelectedItem();
        yy = (String) cb7.getSelectedItem();
        str9 = dd+"-"+mm+"-"+yy;

        str10 = t6.getText();
        str11 = ta1.getText();
        str12 = ta2.getText();
        str13 = t7.getText();
        str14 = t8.getText();
        str15 = t9.getText();

        stm.executeUpdate("insert into studentdetail
values("+str1+", "+str2+", "+str3+", "+str4+", "+str5+", "+str6+", "+str7+", "+str
8+", "+str9+", "+str10+", "+str11+", "+str12+", "+str13+", "+str14+", "+str15+")
");

        stm.executeUpdate("commit");
        JOptionPane.showMessageDialog(null, "Record
Inserted");

        load();
    } //end of else
} //end of try
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null, sql);
}
}

}
public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            if(t1.getText().compareTo("")==0)
                JOptionPane.showMessageDialog(null, "Fill all
Fields");

            else
            {
                String dd, mm, yy;
```





## COLLEGE MANAGEMENT SYSTEM



```
str1 = t1.getText();
str2 = t2.getText();
str3 = t3.getText();
str4 = t4.getText();

str5 = t5.getText();
dd = (String) cb1.getSelectedItem();
mm = (String) cb2.getSelectedItem();
yy = (String) cb3.getSelectedItem();
str6 = dd+"-"+mm+"-"+yy;
if(rb1.isSelected())
    str7 = "Male";
if(rb2.isSelected())
    str7 = "Female";
str8 = (String) cb4.getSelectedItem();

dd = (String) cb5.getSelectedItem();
mm = (String) cb6.getSelectedItem();
yy = (String) cb7.getSelectedItem();
str9 = dd+"-"+mm+"-"+yy;

str10 = t6.getText();
str11 = ta1.getText();
str12 = ta2.getText();
str13 = t7.getText();
str14 = t8.getText();
str15 = t9.getText();

stm.executeUpdate("update studentdetail set roll = "+str2+",sname =
"+str3+",fname = "+str4+",mname = "+str5+",dob = "+str6+",gender =
"+str7+",ecourse = "+str8+",edate = "+str9+",esession = "+str10+",temp =
"+str11+",parm = "+str12+",contact = "+str13+",email = "+str14+",remarks =
"+str15+" where reg = "+str1+""");

stm.executeUpdate("commit");

JOptionPane.showMessageDialog(null,"Records Updated ");
load();
    }
    }
    catch(SQLException sql)
    {
        JOptionPane.showMessageDialog(null,"Could not
Updated"+sql);
    }
}

}

public class viewListener implements ActionListener
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("Enter Employee
number to be view");
            rs = stm.executeQuery("select * from studentdetail
where reg = '"+st+"'");
            while(rs.next())
            {
                String dd,mm,yy,dob,gender;

                t1.setText(rs.getString(1));
                t2.setText(rs.getString(2));

                t3.setText(rs.getString(3));
                t4.setText(rs.getString(4));

                t5.setText(rs.getString(5));

                dob = rs.getString(6);
                yy = dob.substring(0,4);
                mm = dob.substring(5,7);

                dd = dob.substring(8,10);
                cb3.setSelectedItem(String.valueOf(yy));
                cb2.setSelectedIndex(Integer.parseInt(mm));
                cb1.setSelectedItem(dd);

                gender = rs.getString(7).trim();
                if(gender.equals("Male"))
                {
                    rb1.setSelected(true);
                }
                if(gender.equals("Female"))
                {
                    rb2.setSelected(true);
                }
                cb4.setSelectedItem(rs.getString(8));

                dob = rs.getString(9);
                yy = dob.substring(0,4);
```



## COLLEGE MANAGEMENT SYSTEM



```
mm = dob.substring(5,7);

dd = dob.substring(8,10);
cb7.setSelectedItem(String.valueOf(yy));
cb6.setSelectedIndex(Integer.parseInt(mm));
cb5.setSelectedItem(dd);

t6.setText(rs.getString(10));

ta1.setText(rs.getString(11));
ta2.setText(rs.getString(12));
t7.setText(rs.getString(13));
t8.setText(rs.getString(14));
t9.setText(rs.getString(15));
    }
}
catch(SQLException sql)
{
    JOptionPane.showMessageDialog(null,"Not found");
}
}
}
public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("enter
Registration number to be deleted");
            stm.executeUpdate("delete from studentdetail where
reg = '"+st+"'");
            stm.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"Record is
deleted");
            load();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not
deleted");
        }
    }
}
protected void processWindowEvent(WindowEvent we)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    if(we.getID()==we.WINDOW_CLOSING)
    {
        r=JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
        if(r == JOptionPane.YES_OPTION)
            dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Exception"+sql);
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not Connected");
    }
}
public class loadListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        load();
    }
}
public void load()
{
    int k;
    k=0;
    try
    {
        stm.executeUpdate("commit");
```



## COLLEGE MANAGEMENT SYSTEM



```
rs = stm.executeQuery("select  
reg,roll,sname,fname,mname,to_char(dob,'dd-mon-  
yyyy'),gender,ecourse,to_char(edate,'dd-mon-  
yyyy'),esession,temp,parm,contact,email,remarks from studentdetail order by  
reg");
```

```
while(rs.next())  
{  
    str1 = rs.getString(1);  
    str2 = rs.getString(2);  
    str3 = rs.getString(3);  
    str4 = rs.getString(4);  
    str5 = rs.getString(5);  
    str6 = rs.getString(6);  
    str7 = rs.getString(7);  
    str8 = rs.getString(8);  
    str9 = rs.getString(9);  
    str10 = rs.getString(10);  
    str11 = rs.getString(11);  
    str12 = rs.getString(12);  
    str13 = rs.getString(13);  
    str14 = rs.getString(14);  
    str15 = rs.getString(15);
```

```
    data[k][0]=str1;  
    data[k][1]=str2;  
    data[k][2]=str3;  
    data[k][3]=str4;  
    data[k][4]=str5;  
    data[k][5]=str6;  
    data[k][6]=str7;  
    data[k][7]=str8;  
    data[k][8]=str9;  
    data[k][9]=str10;  
    data[k][10]=str11;  
    data[k][11]=str12;  
    data[k][12]=str13;  
    data[k][13]=str14;  
    data[k][14]=str15;
```

```
    k++;  
}  
data[k][0] = " ";  
data[k][1] = " ";  
data[k][2] = " ";  
data[k][3] = " ";  
data[k][4] = " ";
```



## COLLEGE MANAGEMENT SYSTEM



```
data[k][5] = " ";
data[k][6] = " ";
data[k][7] = " ";
data[k][8] = " ";
data[k][9] = " ";
data[k][10] = " ";
data[k][11] = " ";
data[k][12] = " ";
data[k][13] = " ";
data[k][14] = " ";
```

```
JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
table.setForeground(Color.blue);
JScrollPane jsp = new JScrollPane(table);
```

```
jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
```

```
p3.add(jsp);
jsp.setBounds(12,335,910,125);
p3.add(jsp);
}
catch(SQLException sqle)
{
JOptionPane.showMessageDialog(null,"Record not transfer");
}
```

```
}
private class enterListener implements ActionListener
{
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==t1)
t2.requestFocus();
if(e.getSource()==t2)
t3.requestFocus();
if(e.getSource()==t3)
t4.requestFocus();

if(e.getSource()==t4)
t5.requestFocus();
if(e.getSource()==t5)
t6.requestFocus();
if(e.getSource()==t6)
t7.requestFocus();
if(e.getSource()==t7)
t8.requestFocus();
```



## COLLEGE MANAGEMENT SYSTEM



```
        if(e.getSource()==t8)
            t9.requestFocus();
    }
}
private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {118.setLocation(r,40);
     117.setLocation(r,20);
     116.setLocation(r,0);
     r = r+1;
     if(r>290)
         r = 0;
    }
}
private class timerListener1 implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(1,10);

        l = l - 1;
        if( l < 0)
            l = 1240;
    }
}
public class exitListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        dispose();
    }
}
}
```



## /\* Student Fee Detail \*/

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class StudentFeeDetail extends JFrame
{
    private JLabel title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12;
    private JTextField t1,t2,t3,t4,t5,t6,t7;

    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;
    private JScrollPane jp1,jp2;
    private JComboBox cb1,cb2,cb3,cb4,cb5;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;
    ResultSet rs;
    String str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str,st;
    Connection con;
    Statement stm;
    Dimension screensize;
    String[] colhead = {"Reg no.,"Student name","Course","Course
fee","Misc fee","Other","Total dues","Deposit fee","Deposit date","Remarks"};
    Object data[][] = new Object[50][10];
    JTable table = new JTable(data,colhead);
    public StudentFeeDetail()
    {
        super("STUDENT FEE DETAIL FORM");
        l=1;r=1;
        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);
        cn.setBackground(new Color(128,128,25));

        EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);
        BevelBorder edge = new BevelBorder(BevelBorder.RAISED);
```





## COLLEGE MANAGEMENT SYSTEM



```
p1 = new JPanel();  
p2 = new JPanel();  
p3 = new JPanel();  
p4 = new JPanel();
```

```
p1.setBounds(5,5,1240,50);  
p2.setBounds(5,60,300,465);  
p3.setBounds(310,60,935,465);  
p4.setBounds(5,530,1240,55);
```

```
p1.setBackground(new Color(0,0,120));  
p2.setBackground(new Color(112,0,0));  
p3.setBackground(new Color(231,234,104));  
p4.setBackground(new Color(0,0,0));
```

```
title = new JLabel("WELCOME SCREEN FOR STUDENT FEE  
DETAIL");
```

```
title.setFont(new Font("Monotype Corsiva",1,40));  
title.setForeground(Color.red);  
p1.add(title);
```

```
Timer tm = new Timer(50,new timerListener());  
tm.start();
```

```
Timer tm1 = new Timer(5,new timerListener1());  
tm1.start();
```

```
pic = new JLabel();  
Icon img=new ImageIcon("fees.jpg");  
pic.setIcon(img);  
p2.add(pic);
```

```
p3.setLayout(null);  
p4.setLayout(null);
```

```
p3.setBorder(BorderFactory.createTitledBorder("Enter Information  
for Student"));
```

```
l1 = new JLabel("REG NUMBER");  
l2 = new JLabel("STUDENT NAME");
```

```
l3 = new JLabel("COURSE");  
l4 = new JLabel("COURSE FEE");  
l5 = new JLabel("MISC FEE");  
l6 = new JLabel("OTHER");  
l7 = new JLabel("TOTAL DUES");
```



## COLLEGE MANAGEMENT SYSTEM



```
l8 = new JLabel("DEPOSITE FEE");  
l9 = new JLabel("DEPOSITE DATE");  
l10 = new JLabel("REMARKS");
```

```
l11 = new JLabel("Developed by:-");  
l11.setFont(new Font("Times new Roman",1,16));  
l11.setForeground(Color.red);  
p4.add(l11);  
l11.setBounds(0,850,140,25);
```

```
l12=new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.  
Tauqueer , Motahhir , Sanoj collegemanagementsystem@gmail.com");  
l12.setFont(new Font("Monotype Corsiva",1,15));  
l12.setForeground(Color.white);  
l12.setBounds(50,850,340,25);  
p4.add(l12);
```

```
t1 = new JTextField(15);  
t2 = new JTextField(15);  
t3 = new JTextField(15);  
t4 = new JTextField(15);  
t5 = new JTextField(15);  
t6 = new JTextField(15);  
t7 = new JTextField(15);
```

```
cb1 = new JComboBox();
```

```
cb2 = new JComboBox();  
cb2.addItem("-----SELECT-----");  
cb2.addItem("BCA");  
cb2.addItem("BBA");  
cb2.addItem("BBM");  
cb2.addItem("MCA");  
cb2.addItem("MBA");
```

```
cb3 = new JComboBox();  
cb3.addItem("DD");  
cb3.addItem("01");  
cb3.addItem("02");  
cb3.addItem("03");
```

```
cb3.addItem("04");  
cb3.addItem("05");  
cb3.addItem("06");  
cb3.addItem("07");  
cb3.addItem("08");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb3.addItem("09");
cb3.addItem("10");
cb3.addItem("11");
cb3.addItem("12");
cb3.addItem("13");
cb3.addItem("14");
cb3.addItem("15");
cb3.addItem("16");
cb3.addItem("17");
cb3.addItem("18");
cb3.addItem("19");
cb3.addItem("20");
cb3.addItem("21");
cb3.addItem("22");
cb3.addItem("23");
cb3.addItem("24");
cb3.addItem("25");
cb3.addItem("26");
cb3.addItem("27");
cb3.addItem("28");
cb3.addItem("29");
cb3.addItem("30");
cb3.addItem("31");
```

```
cb4 = new JComboBox();
cb4.addItem("MM");
cb4.addItem("Jan");
cb4.addItem("Feb");
cb4.addItem("Mar");
cb4.addItem("Apr");
cb4.addItem("May");
cb4.addItem("Jun");
cb4.addItem("Jul");
cb4.addItem("Aug");
cb4.addItem("Sep");
cb4.addItem("Oct");
cb4.addItem("Nov");
cb4.addItem("Dec");
```

```
cb5 = new JComboBox();
cb5.addItem("YYYY");
cb5.addItem("1970");
cb5.addItem("1971");
cb5.addItem("1972");
cb5.addItem("1973");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb5.addItem("1974");
cb5.addItem("1975");
cb5.addItem("1976");
cb5.addItem("1977");
cb5.addItem("1978");
cb5.addItem("1979");
cb5.addItem("1980");
cb5.addItem("1981");
cb5.addItem("1982");
cb5.addItem("1983");
cb5.addItem("1984");
cb5.addItem("1985");
cb5.addItem("1986");
cb5.addItem("1987");
cb5.addItem("1988");
cb5.addItem("1989");
cb5.addItem("1990");
cb5.addItem("1991");
cb5.addItem("1992");
cb5.addItem("1993");
cb5.addItem("1994");
cb5.addItem("1995");
cb5.addItem("1996");
cb5.addItem("1997");
cb5.addItem("1998");
cb5.addItem("1999");
cb5.addItem("2000");
cb5.addItem("2001");
cb5.addItem("2002");
cb5.addItem("2003");
cb5.addItem("2004");
cb5.addItem("2005");
cb5.addItem("2006");
cb5.addItem("2007");
cb5.addItem("2008");
cb5.addItem("2009");
cb5.addItem("2010");
cb5.addItem("2011");
```

```
cb5.addItem("2012");
cb5.addItem("2013");
cb5.addItem("2014");
cb5.addItem("2015");
cb5.addItem("2016");
cb5.addItem("2017");
cb5.addItem("2018");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb5.addItem("2019");
cb5.addItem("2020");
cb5.addItem("2021");
cb5.addItem("2022");
cb5.addItem("2023");
cb5.addItem("2024");
cb5.addItem("2025");
cb5.addItem("2026");
cb5.addItem("2027");
cb5.addItem("2028");
cb5.addItem("2029");
cb5.addItem("2030");

b1 = new JButton("NEW");
b2 = new JButton("INSERT");
b3 = new JButton("UPDATE");
b4 = new JButton("VIEW");
b5 = new JButton("DELETE");
b6 = new JButton("L O A D");

JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
table.setForeground(new Color(94,128,10));

JScrollPane jsp=new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);

b1.setToolTipText("Refresh the page");
b2.setToolTipText("Insert the records");
b3.setToolTipText("Update the records");
b4.setToolTipText("View the records");
b5.setToolTipText("Delete the records");
b6.setToolTipText("Load the records");

b1.setCursor(cor);
b2.setCursor(cor);
b3.setCursor(cor);
b4.setCursor(cor);
b5.setCursor(cor);
b6.setCursor(cor);

b1.setMnemonic('n');
```



## COLLEGE MANAGEMENT SYSTEM



```
b2.setMnemonic('i');  
b3.setMnemonic('u');  
b4.setMnemonic('v');  
b5.setMnemonic('d');  
b6.setMnemonic('l');
```

```
b1.setFont(new Font("Times New Roman",1,16));  
b2.setFont(new Font("Times New Roman",1,16));  
b3.setFont(new Font("Times New Roman",1,16));  
b4.setFont(new Font("Times New Roman",1,16));  
b5.setFont(new Font("Times New Roman",1,16));  
b6.setFont(new Font("Times New Roman",1,16));
```

```
l1.setFont(new Font("Times New Roman",1,16));  
l2.setFont(new Font("Times New Roman",1,16));  
l3.setFont(new Font("Times New Roman",1,16));  
l4.setFont(new Font("Times New Roman",1,16));  
l5.setFont(new Font("Times New Roman",1,16));  
l6.setFont(new Font("Times New Roman",1,16));  
l7.setFont(new Font("Times New Roman",1,16));  
l8.setFont(new Font("Times New Roman",1,16));  
l9.setFont(new Font("Times New Roman",1,16));  
l10.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));  
t2.setFont(new Font("Times New Roman",1,16));  
t3.setFont(new Font("Times New Roman",1,16));  
t4.setFont(new Font("Times New Roman",1,16));  
t5.setFont(new Font("Times New Roman",1,16));  
t6.setFont(new Font("Times New Roman",1,16));  
t7.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));  
cb2.setFont(new Font("Times New Roman",1,16));  
cb3.setFont(new Font("Times New Roman",1,16));  
cb4.setFont(new Font("Times New Roman",1,16));  
cb5.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(new Color(204,51,0));  
l2.setForeground(new Color(204,51,0));  
l3.setForeground(new Color(204,51,0));  
l4.setForeground(new Color(204,51,0));  
l5.setForeground(new Color(204,51,0));  
l6.setForeground(new Color(204,51,0));  
l7.setForeground(new Color(204,51,0));
```



## COLLEGE MANAGEMENT SYSTEM



```
18.setForeground(new Color(204,51,0));  
19.setForeground(new Color(204,51,0));  
110.setForeground(new Color(204,51,0));
```

```
t1.setForeground(new Color(204,51,0));  
t2.setForeground(new Color(204,51,0));  
t3.setForeground(new Color(204,51,0));  
t4.setForeground(new Color(204,51,0));  
t5.setForeground(new Color(204,51,0));  
t6.setForeground(new Color(204,51,0));  
t7.setForeground(new Color(204,51,0));
```

```
cb1.setForeground(new Color(204,51,0));  
cb2.setForeground(new Color(204,51,0));  
cb3.setForeground(new Color(204,51,0));  
cb4.setForeground(new Color(204,51,0));  
cb5.setForeground(new Color(204,51,0));
```

```
b1.setForeground(new Color(158,89,14));  
b2.setForeground(new Color(158,89,14));  
b3.setForeground(new Color(158,89,14));  
b4.setForeground(new Color(158,89,14));  
b5.setForeground(new Color(158,89,14));  
b6.setBackground(new Color(158,89,14));
```

```
t1.addActionListener(new enterListener());  
t2.addActionListener(new enterListener());  
t3.addActionListener(new enterListener());  
t4.addActionListener(new enterListener());  
t5.addActionListener(new enterListener());  
t6.addActionListener(new enterListener());  
t7.addActionListener(new enterListener());
```

```
b1.addActionListener(new newListener());  
b2.addActionListener(new insertListener());  
b3.addActionListener(new updateListener());  
b4.addActionListener(new viewListener());  
b5.addActionListener(new deleteListener());  
b6.addActionListener(new loadListener());
```

```
11.setBounds(30,30,145,30);  
12.setBounds(30,76,145,30);  
13.setBounds(30,122,145,30);  
14.setBounds(30,168,145,30);  
15.setBounds(30,214,145,30);  
16.setBounds(400,30,145,30);
```



## COLLEGE MANAGEMENT SYSTEM



```
l7.setBounds(400,76,145,30);  
l8.setBounds(400,122,145,30);  
l9.setBounds(400,168,145,30);  
l10.setBounds(400,214,145,30);
```

```
cb1.setBounds(190,30,185,30);  
t1.setBounds(190,76,185,30);  
cb2.setBounds(190,122,185,30);  
t2.setBounds(190,168,185,30);  
t3.setBounds(190,214,185,30);  
t4.setBounds(560,30,185,30);  
t5.setBounds(560,76,185,30);  
t6.setBounds(560,122,185,30);  
cb3.setBounds(560,168,55,30);  
cb4.setBounds(615,168,60,30);  
cb5.setBounds(675,168,70,30);  
t7.setBounds(560,214,185,30);
```

```
b1.setBounds(780,30,100,30);  
b2.setBounds(780,76,100,30);  
b3.setBounds(780,122,100,30);  
b4.setBounds(780,168,100,30);  
b5.setBounds(780,214,100,30);  
b6.setBounds(12,260,910,30);
```

```
jsp.setBounds(12,300,910,160);
```

```
p3.add(b1);  
p3.add(b2);  
p3.add(b3);  
p3.add(b4);
```

```
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);  
p3.add(l2);  
p3.add(l3);  
p3.add(l4);  
p3.add(l5);  
p3.add(l6);  
p3.add(l7);  
p3.add(l8);  
p3.add(l9);  
p3.add(l10);
```





## COLLEGE MANAGEMENT SYSTEM



```
p3.add(t1);
p3.add(t2);
p3.add(t3);
p3.add(t4);
p3.add(t5);
p3.add(t6);
p3.add(t7);

p3.add(cb1);
p3.add(cb2);
p3.add(cb3);
p3.add(cb4);
p3.add(cb5);

cn.add(p1);
cn.add(p2);
cn.add(p3);
cn.add(p4);

l1.setBorder(edge);
l2.setBorder(edge);
l3.setBorder(edge);
l4.setBorder(edge);
l5.setBorder(edge);
l6.setBorder(edge);
l7.setBorder(edge);
l8.setBorder(edge);
l9.setBorder(edge);
l10.setBorder(edge);

t1.setBorder(edge1);
t2.setBorder(edge1);
t3.setBorder(edge1);
t4.setBorder(edge1);
t5.setBorder(edge1);
t6.setBorder(edge1);
t7.setBorder(edge1);

setVisible(true);
setSize(1265,630);
setLocation(5,95);

connect();
regLoad();
}
public void regLoad()
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    Connection con = null;
    Statement stm = null;
    ResultSet rs;
    String qry,rg;
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con = DriverManager.getConnection
("jdbc:odbc:adsn","system","123");
        stm = con.createStatement();
        qry = "select reg from studentdetail";
        rs = stm.executeQuery(qry);

        while(rs.next())
        {
            rg = rs.getString("reg");

            cb1.addItem(rg);
        }
    }
    catch(Exception e1)
    {
    }
    finally
    {
        try
        {
            stm.close();
            con.close();
        }
        catch(SQLException e1)
        {
        }
    }
}

public class newListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            t1.setText(" ");
            t2.setText(" ");
            t3.setText(" ");
            t4.setText(" ");
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
t5.setText(" ");
t6.setText(" ");
t7.setText(" ");

cb1.setSelectedIndex(0);
cb2.setSelectedIndex(0);
cb3.setSelectedIndex(0);
cb4.setSelectedIndex(0);
cb5.setSelectedIndex(0);
    }
    catch(Exception e1)
    {
    }
}
}
private class insertListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = t1.getText();
            if (str1.compareTo("") == 0)

JOptionPane.showMessageDialog(null,"Employee no.  can't be null");
            else
            {
                String dd,mm,yy;
                str1 = (String) cb1.getSelectedItem();
                str2 = t1.getText();
                str3 = (String) cb2.getSelectedItem();
                str4 = t2.getText();

                str5 = t3.getText();
                str6 = t4.getText();
                str7 = t5.getText();
                str8 = t6.getText();
                dd = (String) cb3.getSelectedItem();
                mm = (String) cb4.getSelectedItem();
                yy = (String) cb5.getSelectedItem();
                str9 = dd+"-"+mm+"-"+yy;
                str10 = t7.getText();
```



## COLLEGE MANAGEMENT SYSTEM



```
stm.executeUpdate("insert into studentfeedetail
values('"+str1+"','"+str2+"','"+str3+"','"+str4+"','"+str5+"','"+str6+"','"+str7+"','"+str8+"
','"+str9+"','"+str10+"')");

stm.executeUpdate("commit");
JOptionPane.showMessageDialog(null,"Record
Inserted");

load();
} //end of else
} //end of try
catch(Exception sql)
{
JOptionPane.showMessageDialog(null,sql);
}
}
}
public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = t1.getText();
            if (str1.compareTo("") == 0)

JOptionPane.showMessageDialog(null,"Registration number can't be
null");

            else
            {
                String dd,mm,yy;
                str1 = (String) cb1.getSelectedItem();
                str2 = t1.getText();
                str3 = (String) cb2.getSelectedItem();
                str4 = t2.getText();

                str5 = t3.getText();
                str6 = t4.getText();
                str7 = t5.getText();
                str8 = t6.getText();
                dd = (String) cb3.getSelectedItem();
                mm = (String) cb4.getSelectedItem();
                yy = (String) cb5.getSelectedItem();
                str9 = dd+"-"+mm+"-"+yy;
                str10 = t7.getText();

stm.executeUpdate("update StudentFeeDetail set sname = '"+str2+"',course =
 '"+str3+"',cfec = '"+str4+"',mfee = '"+str5+"',other = '"+str6+"',tdue =
```



## COLLEGE MANAGEMENT SYSTEM



```
" +str7+",deposit = "+str8+",ddate = '"+str9+"',remarks = '"+str10+" where reg =  
"+str1+"";
```

```
stm.executeUpdate("commit");
```

```
JOptionPane.showMessageDialog(null,"Records Updated ");  
load();
```

```
}
```

```
}
```

```
catch(SQLException sql)
```

```
{
```

```
JOptionPane.showMessageDialog(null,"Could not
```

```
Updated"+sql);
```

```
}
```

```
}
```

```
}
```

```
public class viewListener implements ActionListener
```

```
{
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
try
```

```
{
```

```
st = JOptionPane.showInputDialog("Enter
```

```
Registration number to be view");
```

```
rs = stm.executeQuery("select * from
```

```
studentfeeddetail where reg = '"+st+"";
```

```
while(rs.next())
```

```
{
```

```
String dd,mm,yy,ddate;
```

```
cb1.setSelectedItem(rs.getString(1));
```

```
t1.setText(rs.getString(2));
```

```
cb2.setSelectedItem(rs.getString(3));
```

```
t2.setText(rs.getString(4));
```

```
t3.setText(rs.getString(5));
```

```
t4.setText(rs.getString(6));
```

```
t5.setText(rs.getString(7));
```

```
t6.setText(rs.getString(8));
```

```
ddate = rs.getString(9);
```

```
yy = ddate.substring(0,4);
```

```
mm = ddate.substring(5,7);
```

```
dd = ddate.substring(8,10);
```



## COLLEGE MANAGEMENT SYSTEM



```
cb5.setSelectedItem(String.valueOf(yy));
cb4.setSelectedIndex(Integer.parseInt(mm));
cb3.setSelectedItem(dd);

t7.setText(rs.getString(10));
    }
}
catch(SQLException sql)
{
    JOptionPane.showMessageDialog(null,"Not found");
}
}
}
public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        Try
        {
            st = JOptionPane.showInputDialog("Enter
Registration number to be deleted");
            stm.executeUpdate("delete from studentfeedetail
where reg = '"+st+"'");
            stm.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"Record is
deleted");
            load();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not
deleted");
        }
    }
}
protected void processWindowEvent(WindowEvent we)
{
    if(we.getID()==we.WINDOW_CLOSING)
    {
        r=JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
        if(r == JOptionPane.YES_OPTION)
            dispose();
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sql)
        {

JOptionPane.showMessageDialog(null,"Exception"+sql);
        }
    }

    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not Connected");
    }
}
public class loadListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        load();
    }
}
public void load()
{
    int k;
    k=0;
    try
    {
        stm.executeUpdate("commit");
        rs = stm.executeQuery("select
reg,sname,course,cfee,mfee,other,tdue,deposit,to_char(ddate,'dd-mon-
yyyy'),remarks from studentfeedetail order by reg");

        while(rs.next())
        {
            str1 = rs.getString(1);
            str2 = rs.getString(2);
            str3 = rs.getString(3);
```



## COLLEGE MANAGEMENT SYSTEM



```
        str4 = rs.getString(4);
        str5 = rs.getString(5);
        str6 = rs.getString(6);
        str7 = rs.getString(7);
        str8 = rs.getString(8);
        str9 = rs.getString(9);
        str10 = rs.getString(10);

        data[k][0]=str1;
        data[k][1]=str2;
        data[k][2]=str3;
        data[k][3]=str4;
        data[k][4]=str5;
        data[k][5]=str6;
        data[k][6]=str7;
        data[k][7]=str8;

        data[k][8]=str9;
        data[k][9]=str10;

        k++;
    }
    data[k][0] = " ";
    data[k][1] = " ";
    data[k][2] = " ";
    data[k][3] = " ";
    data[k][4] = " ";
    data[k][5] = " ";
    data[k][6] = " ";
    data[k][7] = " ";
    data[k][8] = " ";
    data[k][9] = " ";

    JTable table = new JTable(data,colhead);
    table.setFont(new Font("Times New Roman",Font.BOLD,16));
    table.setForeground(new Color(94,128,10));
    JScrollPane jsp = new JScrollPane(table);

    jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
    p3.add(jsp);
    jsp.setBounds(12,300,910,160);
    p3.add(jsp);
    }
    catch(SQLException sqle)
    {
```





## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"Record not transfer");
    }
}
private class enterListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==t1)
            t2.requestFocus();
        if(e.getSource()==t2)
            t3.requestFocus();
        if(e.getSource()==t3)
            t4.requestFocus();
        if(e.getSource()==t4)
            t5.requestFocus();

        if(e.getSource()==t5)
            t6.requestFocus();
        if(e.getSource()==t6)
            t7.requestFocus();
    }
}
private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l12.setLocation(r,20);
        l11.setLocation(r,0);
        r = r+1;
        if(r>1240)
            r = 0;
    }
}
private class timerListener1 implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(l,10);

        l = l - 1;
        if( l < 0)
            l = 1240;
    }
}
public class exitListener implements ActionListener
{
```



## COLLEGE MANAGEMENT SYSTEM



```
public void actionPerformed(ActionEvent e)
{
    dispose();
}
}
```



## **/\* Billing Master \*/**

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class BillingMaster extends JFrame
{
    private JLabel
title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17;
    private JTextField t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11;

    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;
    private JScrollPane jp1,jp2;
    private JComboBox cb1;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;
    ResultSet rs;
    String
str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11,str12,str13,str14,str15,str16,str,s
t;

    Connection con;
    Statement stm;
    Dimension screensize;
    String[] colhead = {"Reg no.," "Roll","Student name","Session","Course
fee","Exam fee","Admin fee","Lab fee","Lib fee","Misc fee","Others","Total"};
    Object data[][] = new Object[50][12];
    JTable table = new JTable(data,colhead);
    public BillingMaster()
    {
        super("STUDENT BILL DETAIL FORM");
        l=1;r=1;
        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);

        cn.setBackground(new Color(128,128,25));

        EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);
```



## COLLEGE MANAGEMENT SYSTEM



```
BevelBorder edge = new BevelBorder(BevelBorder.RAISED);
```

```
p1 = new JPanel();  
p2 = new JPanel();  
p3 = new JPanel();  
p4 = new JPanel();
```

```
p1.setBounds(5,5,1240,50);  
p2.setBounds(5,60,300,400);  
p3.setBounds(310,60,935,465);  
p4.setBounds(5,465,300,60);
```

```
p1.setBackground(new Color(64,0,0));  
p2.setBackground(new Color(112,0,0));  
p3.setBackground(new Color(255,140,255));  
p4.setBackground(new Color(0,64,0));
```

```
title = new JLabel("WELCOME SCREEN FOR STUDENT BILL  
DETAIL");
```

```
title.setFont(new Font("Monotype Corsiva",1,40));  
title.setForeground(Color.red);  
p1.add(title);
```

```
Timer tm = new Timer(100,new timerListener());  
tm.start();
```

```
Timer tm1 = new Timer(5,new timerListener1());  
tm1.start();
```

```
pic = new JLabel();  
Icon img=new ImageIcon("billing.jpg");  
pic.setIcon(img);  
p2.add(pic);
```

```
p3.setLayout(null);  
p4.setLayout(null);
```

```
p3.setBorder(BorderFactory.createTitledBorder("Enter Information  
for Student bill"));
```

```
l1 = new JLabel("REG NUMBER");  
l2 = new JLabel("ROLL NUMBER");  
l3 = new JLabel("STUDENT NAME");  
l4 = new JLabel("SESSION");  
l5 = new JLabel("COURSE FEE");  
l6 = new JLabel("EXAM FEE");
```



## COLLEGE MANAGEMENT SYSTEM



```
l7 = new JLabel("ADMIN FEE");  
l8 = new JLabel("LAB FEE");  
l9 = new JLabel("LIB FEE");  
l10 = new JLabel("MISC FEE");  
l11 = new JLabel("OTHERS");  
l12 = new JLabel("TOTAL");
```

```
l15 = new JLabel("Developed by:- ");  
l15.setFont(new Font("Times new Roman",1,15));  
l15.setForeground(Color.white);  
l15.setBounds(0,850,140,25);  
p4.add(l15);
```

```
l16=new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.  
Tauqueer , Motahhir , Sanoj");  
l16.setFont(new Font("Monotype Corsiva",1,15));  
l16.setForeground(Color.white);  
l16.setBounds(50,850,220,25);  
p4.add(l16);
```

```
l17=new JLabel("collegemanagementsystem@gmail.com");  
l17.setFont(new Font("Monotype Corsiva",1,15));  
l17.setForeground(Color.white);  
l17.setBounds(50,850,220,25);  
p4.add(l17);
```

```
t1 = new JTextField(15);  
t2 = new JTextField(15);  
t3 = new JTextField(15);  
t4 = new JTextField(15);  
t5 = new JTextField(15);  
t6 = new JTextField(15);  
t7 = new JTextField(15);  
t8 = new JTextField(15);  
t9 = new JTextField(15);  
t10 = new JTextField(15);  
t11 = new JTextField(15);
```

```
cb1 = new JComboBox();
```

```
cb1.addItem("SELECT REG NO.");
```

```
b1 = new JButton("NEW");  
b2 = new JButton("INSERT");  
b3 = new JButton("UPDATE");  
b4 = new JButton("VIEW");
```



## COLLEGE MANAGEMENT SYSTEM



```
b5 = new JButton("DELETE");  
b6 = new JButton("LOAD");
```

```
JTable table = new JTable(data,colhead);  
table.setFont(new Font("Times New Roman",Font.BOLD,16));  
table.setForeground(Color.blue);
```

```
JScrollPane jsp=new JScrollPane(table);
```

```
jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB  
AR_NEVER);
```

```
p3.add(jsp);
```

```
b1.setToolTipText("Refresh the page");  
b2.setToolTipText("Insert the records");  
b3.setToolTipText("Update the records");  
b4.setToolTipText("View the records");  
b5.setToolTipText("Delete the records");  
b6.setToolTipText("Load the records");
```

```
b1.setCursor(cor);  
b2.setCursor(cor);  
b3.setCursor(cor);  
b4.setCursor(cor);  
b5.setCursor(cor);  
b6.setCursor(cor);
```

```
b1.setMnemonic('n');  
b2.setMnemonic('i');  
b3.setMnemonic('u');  
b4.setMnemonic('v');  
b5.setMnemonic('d');  
b6.setMnemonic('l');
```

```
b1.setFont(new Font("Times New Roman",1,16));  
b2.setFont(new Font("Times New Roman",1,16));  
b3.setFont(new Font("Times New Roman",1,16));  
b4.setFont(new Font("Times New Roman",1,16));
```

```
b5.setFont(new Font("Times New Roman",1,16));  
b6.setFont(new Font("Times New Roman",1,16));
```

```
l1.setFont(new Font("Times New Roman",1,16));  
l2.setFont(new Font("Times New Roman",1,16));  
l3.setFont(new Font("Times New Roman",1,16));  
l4.setFont(new Font("Times New Roman",1,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
15.setFont(new Font("Times New Roman",1,16));
16.setFont(new Font("Times New Roman",1,16));
17.setFont(new Font("Times New Roman",1,16));
18.setFont(new Font("Times New Roman",1,16));
19.setFont(new Font("Times New Roman",1,16));
110.setFont(new Font("Times New Roman",1,16));
111.setFont(new Font("Times New Roman",1,16));
112.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));
t2.setFont(new Font("Times New Roman",1,16));
t3.setFont(new Font("Times New Roman",1,16));
t4.setFont(new Font("Times New Roman",1,16));
t5.setFont(new Font("Times New Roman",1,16));
t6.setFont(new Font("Times New Roman",1,16));
t7.setFont(new Font("Times New Roman",1,16));
t8.setFont(new Font("Times New Roman",1,16));
t9.setFont(new Font("Times New Roman",1,16));
t10.setFont(new Font("Times New Roman",1,16));
t11.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(Color.blue);
l2.setForeground(Color.blue);
l3.setForeground(Color.blue);
l4.setForeground(Color.blue);
l5.setForeground(Color.blue);
l6.setForeground(Color.blue);
l7.setForeground(Color.blue);
l8.setForeground(Color.blue);
l9.setForeground(Color.blue);
l10.setForeground(Color.blue);
l11.setForeground(Color.blue);
l12.setForeground(Color.blue);
```

```
t1.setForeground(Color.blue);
t2.setForeground(Color.blue);
t3.setForeground(Color.blue);
t4.setForeground(Color.blue);
t5.setForeground(Color.blue);
t6.setForeground(Color.blue);
t7.setForeground(Color.blue);
t8.setForeground(Color.blue);
t9.setForeground(Color.blue);
t10.setForeground(Color.blue);
```



## COLLEGE MANAGEMENT SYSTEM



```
t11.setForeground(Color.blue);
```

```
cb1.setForeground(Color.blue);
```

```
l1.setBorder(edge);  
l2.setBorder(edge);  
l3.setBorder(edge);  
l4.setBorder(edge);  
l5.setBorder(edge);  
l6.setBorder(edge);  
l7.setBorder(edge);  
l8.setBorder(edge);  
l9.setBorder(edge);  
l10.setBorder(edge);  
l11.setBorder(edge);  
l12.setBorder(edge);
```

```
t1.setBorder(edge1);  
t2.setBorder(edge1);  
t3.setBorder(edge1);  
t4.setBorder(edge1);  
t5.setBorder(edge1);  
t6.setBorder(edge1);  
t7.setBorder(edge1);  
t8.setBorder(edge1);  
t9.setBorder(edge1);  
t10.setBorder(edge1);  
t11.setBorder(edge1);
```

```
t1.addActionListener(new enterListener());  
t2.addActionListener(new enterListener());  
t3.addActionListener(new enterListener());
```

```
t4.addActionListener(new enterListener());  
t5.addActionListener(new enterListener());  
t6.addActionListener(new enterListener());  
t7.addActionListener(new enterListener());  
t8.addActionListener(new enterListener());  
t9.addActionListener(new enterListener());  
t10.addActionListener(new enterListener());  
t11.addActionListener(new enterListener());
```

```
b1.addActionListener(new newListener());  
b2.addActionListener(new insertListener());  
b3.addActionListener(new updateListener());  
b4.addActionListener(new viewListener());
```





## COLLEGE MANAGEMENT SYSTEM



```
b5.addActionListener(new deleteListener());  
b6.addActionListener(new loadListener());
```

```
l1.setBounds(30,30,145,25);  
l2.setBounds(30,76,145,25);  
l3.setBounds(30,122,145,25);  
l4.setBounds(30,168,145,25);  
l5.setBounds(30,214,145,25);  
l6.setBounds(30,260,145,25);  
l7.setBounds(400,30,145,25);  
l8.setBounds(400,76,145,25);  
l9.setBounds(400,122,145,25);  
l10.setBounds(400,168,145,25);  
l11.setBounds(400,214,145,25);  
l12.setBounds(400,260,145,25);
```

```
cb1.setBounds(190,30,185,25);
```

```
t1.setBounds(190,76,185,25);  
t2.setBounds(190,122,185,25);  
t3.setBounds(190,168,185,25);  
t4.setBounds(190,214,185,25);  
t5.setBounds(190,260,185,25);  
t6.setBounds(560,30,185,25);  
t7.setBounds(560,76,185,25);  
t8.setBounds(560,122,185,25);  
t9.setBounds(560,168,185,25);  
t10.setBounds(560,214,185,25);  
t11.setBounds(560,260,185,25);
```

```
b1.setBounds(780,30,100,30);  
b2.setBounds(780,76,100,30);  
b3.setBounds(780,122,100,30);  
b4.setBounds(780,168,100,30);  
b5.setBounds(780,214,100,30);  
b6.setBounds(780,260,100,30);  
jsp.setBounds(12,300,910,160);
```

```
p3.add(b1);  
p3.add(b2);  
p3.add(b3);  
p3.add(b4);  
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);
```



## COLLEGE MANAGEMENT SYSTEM



```
p3.add(l2);
p3.add(l3);
p3.add(l4);
p3.add(l5);
p3.add(l6);
p3.add(l7);
p3.add(l8);
p3.add(l9);
p3.add(l10);
p3.add(l11);
p3.add(l12);

p3.add(t1);
p3.add(t2);
p3.add(t3);
p3.add(t4);
p3.add(t5);
p3.add(t6);
p3.add(t7);
p3.add(t8);
p3.add(t9);
p3.add(t10);
p3.add(t11);

p3.add(cb1);

cn.add(p1);
cn.add(p2);
cn.add(p3);
cn.add(p4);

setVisible(true);
setSize(1265,570);
setLocation(5,95);
connect();
regLoad();
}
public void regLoad()
{
    Connection con = null;
    Statement stm = null;
    ResultSet rs;
    String qry,rg;
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```



## COLLEGE MANAGEMENT SYSTEM



```
        con = DriverManager.getConnection
("jdbc:odbc:adsn","system","123");
        stm = con.createStatement();
        qry = "select reg from studentdetail;";
        rs = stm.executeQuery(qry);
        while(rs.next())
        {
            rg = rs.getString("reg");

            cb1.addItem(rg);
        }
    }
    catch(Exception e1)
    {
    }
    finally
    {
        try
        {
            stm.close();
            con.close();
        }
        catch(SQLException e1)
        {
        }
    }
}

public class newListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            t1.setText(" ");
            t2.setText(" ");
            t3.setText(" ");
            t4.setText(" ");
            t5.setText(" ");
            t6.setText(" ");
            t7.setText(" ");
            t8.setText(" ");
            t9.setText(" ");
            t10.setText(" ");
            t11.setText(" ");
            cb1.setSelectedIndex(0);
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        }
        catch(Exception e1)
        {
        }
    }
}

private class insertListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        //k=0;
        try
        {
            str1 = t1.getText();
            if (str1.compareTo("") == 0)

OptionPane.showMessageDialog(null,"Student roll can't be null");
            else
            {
                str1 = (String) cb1.getSelectedItem();
                str2 = t1.getText();
                str3 = t2.getText();

                str4 = t3.getText();
                str5 = t4.getText();
                str6 = t5.getText();
                str7 = t6.getText();
                str8 = t7.getText();
                str9 = t8.getText();
                str10 = t9.getText();
                str11 = t10.getText();
                str12 = t11.getText();

                stm.executeUpdate("insert into billingmaster
values("+str1+"",""+str2+"",""+str3+"",""+str4+"",""+str5+"",""+str6+"",""+str7+"",""+str8+"
",""+str9+"",""+str10+"",""+str11+"",""+str12+"");
                stm.executeUpdate("commit");
OptionPane.showMessageDialog(null,"Record
Inserted");

                load();
            }//end of else
        }//end of try
        catch(Exception sql)
        {
OptionPane.showMessageDialog(null,sql);
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
    }
}
public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            if(t1.getText().compareTo("")==0)
                JOptionPane.showMessageDialog(null,"Fill all
Fields");

            else
            {
                str1 = (String) cb1.getSelectedItemAt();
                str2 = t1.getText();
                str3 = t2.getText();

                str4 = t3.getText();
                str5 = t4.getText();
                str6 = t5.getText();
                str7 = t6.getText();
                str8 = t7.getText();
                str9 = t8.getText();
                str10 = t9.getText();
                str11 = t10.getText();
                str12 = t11.getText();

                stm.executeUpdate("update billingmaster set
stdroll = "+str2+",stdname = '"+str3+"',sesion = '"+str4+"',coursefee =
"+str5+",examfee = "+str6+",adminfee = "+str7+",labfee = "+str8+",libfee =
"+str9+",miscfee = "+str10+",others = "+str11+",total = "+str12+" where stdreg =
 '"+str1+"'");

                stm.executeUpdate("commit");
                JOptionPane.showMessageDialog(null,"Records Updated ");
                load();
            }
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Could not
Updated"+sql);
        }
    }
}
public class viewListener implements ActionListener
{
```



## COLLEGE MANAGEMENT SYSTEM



```
public void actionPerformed(ActionEvent e)
{
    try
    {
        st = JOptionPane.showInputDialog("Input
Registration number to be view");

        rs = stm.executeQuery("select * from billingmaster
where stdreg = '"+st+"'");
        while(rs.next())
        {
            t1.setText(rs.getString(2));
            t2.setText(rs.getString(3));

            t3.setText(rs.getString(4));
            t4.setText(rs.getString(5));

            t5.setText(rs.getString(6));
            t6.setText(rs.getString(7));
            t7.setText(rs.getString(8));
            t8.setText(rs.getString(9));
            t9.setText(rs.getString(10));
            t10.setText(rs.getString(11));
            t11.setText(rs.getString(12));
        }
    }
    catch(SQLException sql)
    {
        JOptionPane.showMessageDialog(null,"Not found");
    }
}

public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("Input
Registration number to be deleted");
            stm.executeUpdate("delete from billingmaster where
stdreg = '"+st+"'");

            stm.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"Record is
deleted");

            load();
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not deleted");
        }
    }
}
protected void processWindowEvent(WindowEvent we)
{
    if(we.getID()==we.WINDOW_CLOSING)
    {
        r=JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
        if(r == JOptionPane.YES_OPTION)
            dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Exception"+sql);
        }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not Connected");
    }
}
public class loadListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        load();
    }
}
public void load()
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    int k;
    k=0;
    try
    {
        stm.executeUpdate("commit");
        rs = stm.executeQuery("select * from BillingMaster");

        while(rs.next())
        {
            str1 = rs.getString(1);
            str2 = rs.getString(2);
            str3 = rs.getString(3);
            str4 = rs.getString(4);
            str5 = rs.getString(5);
            str6 = rs.getString(6);
            str7 = rs.getString(7);
            str8 = rs.getString(8);
            str9 = rs.getString(9);
            str10 = rs.getString(10);
            str11 = rs.getString(11);
            str12 = rs.getString(12);

            data[k][0]=str1;
            data[k][1]=str2;
            data[k][2]=str3;
            data[k][3]=str4;
            data[k][4]=str5;
            data[k][5]=str6;
            data[k][6]=str7;
            data[k][7]=str8;
            data[k][8]=str9;
            data[k][9]=str10;
            data[k][10]=str11;
            data[k][11]=str12;

            k++;
        }
        data[k][0] = " ";
        data[k][1] = " ";
        data[k][2] = " ";
        data[k][3] = " ";
        data[k][4] = " ";
        data[k][5] = " ";

        data[k][6] = " ";
```





## COLLEGE MANAGEMENT SYSTEM



```
data[k][7] = " ";
data[k][8] = " ";
data[k][9] = " ";
data[k][10] = " ";
data[k][11] = " ";

JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
table.setForeground(Color.blue);
JScrollPane jsp = new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);
jsp.setBounds(12,300,910,160);
p3.add(jsp);
}
catch(SQLException sqle)
{
JOptionPane.showMessageDialog(null,"Record not transfer");
}
}
private class enterListener implements ActionListener
{
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==t1)
t2.requestFocus();
if(e.getSource()==t2)
t3.requestFocus();
if(e.getSource()==t3)
t4.requestFocus();
if(e.getSource()==t4)
t5.requestFocus();
if(e.getSource()==t5)
t6.requestFocus();
if(e.getSource()==t6)
t7.requestFocus();
if(e.getSource()==t7)
t8.requestFocus();
if(e.getSource()==t8)
t9.requestFocus();
if(e.getSource()==t9)
t10.requestFocus();
if(e.getSource()==t10)
{
```



## COLLEGE MANAGEMENT SYSTEM



```
double z1,z2,z3,z4,z5,z6,z7,z8;
z1 = Double.parseDouble(t4.getText());
z2 = Double.parseDouble(t5.getText());
z3 = Double.parseDouble(t6.getText());
z4 = Double.parseDouble(t7.getText());
z5 = Double.parseDouble(t8.getText());
z6 = Double.parseDouble(t9.getText());
z7 = Double.parseDouble(t10.getText());
z8 = z1 + z2 + z3 + z4 + z5 + z6 + z7;
t11.setText(String.valueOf(z8));
t11.requestFocus();
    }
}
}
private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l17.setLocation(r,40);
        l16.setLocation(r,20);
        l15.setLocation(r,0);
        r = r+1;
        if(r>290)
            r = 0;
    }
}
private class timerListener1 implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(l,10);
        l = l - 1;
        if( l < 0)
            l = 1240;
    }
}
public class exitListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        dispose();
    }
}
}
```



## /\* PayRoll \*/

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class PayRoll extends JFrame
{
    private JLabel
title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17,l18,l19;
    private JTextField t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13;

    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;

    private JComboBox cb1,cb2,cb3;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;
    ResultSet rs;
    String
str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11,str12,str13,str14,str15,str16,str,s
t;

    Connection con;
    Statement stm;
    Dimension screensize;
    String[] colhead = { "eno","ename","month","year","basic
pay","hra","da","ma","ta","pf","esi","advance","lic","other","net
salary","remarks"};
    Object data[][] = new Object[50][16];
    JTable table = new JTable(data,colhead);
    public PayRoll()
    {
        super("PAYROLL FORM");
        l=1;r=1;
        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        setLayout(null);
        cn.setBackground(new Color(128,128,25));
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);  
BevelBorder edge = new BevelBorder(BevelBorder.RAISED);
```

```
p1 = new JPanel();  
p2 = new JPanel();  
p3 = new JPanel();  
p4 = new JPanel();
```

```
p1.setBounds(5,5,1240,50);  
p2.setBounds(5,60,300,400);  
p3.setBounds(310,60,935,465);  
p4.setBounds(5,465,300,60);
```

```
p1.setBackground(new Color(116,16,106));  
p2.setBackground(new Color(2,0,0));
```

```
p3.setBackground(new Color(255,255,128));  
p4.setBackground(new Color(16,20,120));
```

```
title = new JLabel("WELCOME SCREEN FOR PAYROLL  
DETAIL");
```

```
title.setFont(new Font("Rockwell",8,40));  
title.setForeground(Color.white);  
p1.add(title);
```

```
Timer tm = new Timer(100,new timerListener());  
tm.start();
```

```
Timer tm1 = new Timer(5,new timerListener1());  
tm1.start();
```

```
pic = new JLabel();  
Icon img=new ImageIcon("payroll.jpg");  
pic.setIcon(img);  
p2.add(pic);
```

```
p3.setLayout(null);  
p4.setLayout(null);
```

```
p3.setBorder(BorderFactory.createTitledBorder("Enter Information for  
Employee Payroll"));
```

```
l1 = new JLabel("Employee number");  
l2 = new JLabel("Employee name");  
l3 = new JLabel("Month");  
l4 = new JLabel("Year");
```



## COLLEGE MANAGEMENT SYSTEM



```
15 = new JLabel("Basic pay");
16 = new JLabel("HRA");
17 = new JLabel("DA");
18 = new JLabel("MA");
19 = new JLabel("TA");
110 = new JLabel("PF");
111 = new JLabel("ESI");
112 = new JLabel("Advance");
113 = new JLabel("LIC");
114 = new JLabel("Other");
115 = new JLabel("Net salary");
116 = new JLabel("Remarks");

117 = new JLabel("Developed by:-");
117.setFont(new Font("Times new Roman",1,14));
117.setForeground(Color.white);
117.setBounds(0,850,140,25);
p4.add(117);

118=new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.
Tauqueer , Motahhir , Sanoj");
118.setFont(new Font("Monotype Corsiva",1,14));
118.setForeground(Color.white);
118.setBounds(50,850,220,25);
p4.add(118);

119=new JLabel("collegemanagementsystem@gmail.com");
119.setFont(new Font("Monotype Corsiva",1,14));
119.setForeground(Color.white);
119.setBounds(50,850,220,25);
p4.add(119);

t1 = new JTextField(15);
t2 = new JTextField(15);
t3 = new JTextField(15);
t4 = new JTextField(15);
t5 = new JTextField(15);
t6 = new JTextField(15);

t7 = new JTextField(15);
t8 = new JTextField(15);
t9 = new JTextField(15);
t10 = new JTextField(15);
t11 = new JTextField(15);
t12 = new JTextField(15);
t13 = new JTextField(15);
```



## COLLEGE MANAGEMENT SYSTEM



```
cb1 = new JComboBox();
```

```
cb2 = new JComboBox();  
cb2.addItem("MM");  
cb2.addItem("Jan");  
cb2.addItem("Feb");  
cb2.addItem("Mar");  
cb2.addItem("Apr");  
cb2.addItem("May");  
cb2.addItem("Jun");  
cb2.addItem("Jul");  
cb2.addItem("Aug");  
cb2.addItem("Sep");  
cb2.addItem("Oct");  
cb2.addItem("Nov");  
cb2.addItem("Dec");
```

```
cb3 = new JComboBox();  
cb3.addItem("YYYY");  
cb3.addItem("1970");  
cb3.addItem("1971");  
cb3.addItem("1972");  
cb3.addItem("1973");  
cb3.addItem("1974");  
cb3.addItem("1975");  
cb3.addItem("1976");  
cb3.addItem("1977");  
cb3.addItem("1978");  
cb3.addItem("1979");  
cb3.addItem("1980");  
cb3.addItem("1981");  
cb3.addItem("1982");  
cb3.addItem("1983");  
cb3.addItem("1984");  
cb3.addItem("1985");  
cb3.addItem("1986");
```

```
cb3.addItem("1987");  
cb3.addItem("1988");  
cb3.addItem("1989");  
cb3.addItem("1990");  
cb3.addItem("1991");  
cb3.addItem("1992");  
cb3.addItem("1993");  
cb3.addItem("1994");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb3.addItem("1995");
cb3.addItem("1996");
cb3.addItem("1997");
cb3.addItem("1998");
cb3.addItem("1999");
cb3.addItem("2000");
cb3.addItem("2001");
cb3.addItem("2002");
cb3.addItem("2003");
cb3.addItem("2004");
cb3.addItem("2005");
cb3.addItem("2006");
cb3.addItem("2007");
cb3.addItem("2008");
cb3.addItem("2009");
cb3.addItem("2010");
cb3.addItem("2011");
cb3.addItem("2012");
cb3.addItem("2013");
cb3.addItem("2014");
cb3.addItem("2015");
cb3.addItem("2016");
cb3.addItem("2017");
cb3.addItem("2018");
cb3.addItem("2019");
cb3.addItem("2020");
cb3.addItem("2021");
cb3.addItem("2022");
cb3.addItem("2023");
cb3.addItem("2024");
cb3.addItem("2025");
cb3.addItem("2026");
cb3.addItem("2027");
cb3.addItem("2028");
cb3.addItem("2029");
cb3.addItem("2030");
```

```
b1 = new JButton("NEW");
b2 = new JButton("INSERT");
b3 = new JButton("UPDATE");
b4 = new JButton("VIEW");
b5 = new JButton("DELETE");
b6 = new JButton("LOAD");
```

```
JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
table.setForeground(Color.blue);

JScrollPane jsp=new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);

b1.setToolTipText("Refresh the page");
b2.setToolTipText("Insert the records");
b3.setToolTipText("Update the records");
b4.setToolTipText("View the records");
b5.setToolTipText("Delete the records");
b6.setToolTipText("Load the records");

b1.setCursor(cor);
b2.setCursor(cor);
b3.setCursor(cor);
b4.setCursor(cor);
b5.setCursor(cor);
b6.setCursor(cor);

b1.setMnemonic('n');
b2.setMnemonic('i');
b3.setMnemonic('u');
b4.setMnemonic('v');
b5.setMnemonic('d');
b6.setMnemonic('l');

b1.setFont(new Font("Times New Roman",1,16));
b2.setFont(new Font("Times New Roman",1,16));
b3.setFont(new Font("Times New Roman",1,16));
b4.setFont(new Font("Times New Roman",1,16));
b5.setFont(new Font("Times New Roman",1,16));
b6.setFont(new Font("Times New Roman",1,16));

i1.setFont(new Font("Times New Roman",1,16));
i2.setFont(new Font("Times New Roman",1,16));
i3.setFont(new Font("Times New Roman",1,16));
i4.setFont(new Font("Times New Roman",1,16));
i5.setFont(new Font("Times New Roman",1,16));
i6.setFont(new Font("Times New Roman",1,16));
i7.setFont(new Font("Times New Roman",1,16));
i8.setFont(new Font("Times New Roman",1,16));
i9.setFont(new Font("Times New Roman",1,16));
i10.setFont(new Font("Times New Roman",1,16));
```





## COLLEGE MANAGEMENT SYSTEM



```
l11.setFont(new Font("Times New Roman",1,16));  
l12.setFont(new Font("Times New Roman",1,16));  
l13.setFont(new Font("Times New Roman",1,16));  
l14.setFont(new Font("Times New Roman",1,16));  
l15.setFont(new Font("Times New Roman",1,16));  
l16.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));  
t2.setFont(new Font("Times New Roman",1,16));  
t3.setFont(new Font("Times New Roman",1,16));  
t4.setFont(new Font("Times New Roman",1,16));  
t5.setFont(new Font("Times New Roman",1,16));  
t6.setFont(new Font("Times New Roman",1,16));  
t7.setFont(new Font("Times New Roman",1,16));  
t8.setFont(new Font("Times New Roman",1,16));  
t9.setFont(new Font("Times New Roman",1,16));  
t10.setFont(new Font("Times New Roman",1,16));  
t11.setFont(new Font("Times New Roman",1,16));  
t12.setFont(new Font("Times New Roman",1,16));  
t13.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));  
cb2.setFont(new Font("Times New Roman",1,16));  
cb3.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(Color.blue);  
l2.setForeground(Color.blue);  
l3.setForeground(Color.blue);  
l4.setForeground(Color.blue);  
l5.setForeground(Color.blue);  
l6.setForeground(Color.blue);  
l7.setForeground(Color.blue);
```

```
l8.setForeground(Color.blue);  
l9.setForeground(Color.blue);  
l10.setForeground(Color.blue);  
l11.setForeground(Color.blue);  
l12.setForeground(Color.blue);  
l13.setForeground(Color.blue);  
l14.setForeground(Color.blue);  
l15.setForeground(Color.blue);  
l16.setForeground(Color.blue);
```

```
t1.setForeground(Color.blue);  
t2.setForeground(Color.blue);  
t3.setForeground(Color.blue);
```



## COLLEGE MANAGEMENT SYSTEM



```
t4.setForeground(Color.blue);
t5.setForeground(Color.blue);
t6.setForeground(Color.blue);
t7.setForeground(Color.blue);
t8.setForeground(Color.blue);
t9.setForeground(Color.blue);
t10.setForeground(Color.blue);
t11.setForeground(Color.blue);
t12.setForeground(Color.blue);
t13.setForeground(Color.blue);
```

```
cb1.setForeground(Color.blue);
cb2.setForeground(Color.blue);
cb3.setForeground(Color.blue);
```

```
t1.addActionListener(new enterListener());
t2.addActionListener(new enterListener());
t3.addActionListener(new enterListener());
t4.addActionListener(new enterListener());
t5.addActionListener(new enterListener());
t6.addActionListener(new enterListener());
t7.addActionListener(new enterListener());
t8.addActionListener(new enterListener());
t9.addActionListener(new enterListener());
t10.addActionListener(new enterListener());
t11.addActionListener(new enterListener());
t12.addActionListener(new enterListener());
t13.addActionListener(new enterListener());
```

```
b1.addActionListener(new newListener());
b2.addActionListener(new insertListener());
```

```
b3.addActionListener(new updateListener());
b4.addActionListener(new viewListener());
b5.addActionListener(new deleteListener());
b6.addActionListener(new loadListener());
```

```
l1.setBounds(30,30,145,25);
l2.setBounds(30,60,145,25);
l3.setBounds(30,90,145,25);
l4.setBounds(30,120,145,25);
l5.setBounds(30,150,145,25);
l6.setBounds(30,180,145,25);
l7.setBounds(30,210,145,25);
l8.setBounds(30,240,145,25);
l9.setBounds(400,30,145,25);
```



## COLLEGE MANAGEMENT SYSTEM



```
l10.setBounds(400,60,145,25);  
l11.setBounds(400,90,145,25);  
l12.setBounds(400,120,145,25);  
l13.setBounds(400,150,145,25);  
l14.setBounds(400,180,145,25);  
l15.setBounds(400,210,145,25);  
l16.setBounds(400,240,145,25);
```

```
cb1.setBounds(190,30,185,25);  
t1.setBounds(190,60,185,25);  
cb2.setBounds(190,90,185,25);  
cb3.setBounds(190,120,185,25);  
t2.setBounds(190,150,185,25);  
t3.setBounds(190,180,185,25);  
t4.setBounds(190,210,185,25);  
t5.setBounds(190,240,185,25);
```

```
t6.setBounds(560,30,185,25);  
t7.setBounds(560,60,185,25);  
t8.setBounds(560,90,185,25);  
t9.setBounds(560,120,185,25);  
t10.setBounds(560,150,185,25);  
t11.setBounds(560,180,185,25);  
t12.setBounds(560,210,185,25);  
t13.setBounds(560,240,185,25);
```

```
b1.setBounds(780,30,100,30);  
b2.setBounds(780,76,100,30);  
b3.setBounds(780,122,100,30);  
b4.setBounds(780,168,100,30);
```

```
b5.setBounds(780,214,100,30);  
b6.setBounds(12,270,910,30);
```

```
jsp.setBounds(12,305,910,150);
```

```
p3.add(b1);  
p3.add(b2);  
p3.add(b3);  
p3.add(b4);  
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);  
p3.add(l2);  
p3.add(l3);
```



## COLLEGE MANAGEMENT SYSTEM



```
p3.add(l4);  
p3.add(l5);  
p3.add(l6);  
p3.add(l7);  
p3.add(l8);  
p3.add(l9);  
p3.add(l10);  
p3.add(l11);  
p3.add(l12);  
p3.add(l13);  
p3.add(l14);  
p3.add(l15);  
p3.add(l16);
```

```
p3.add(t1);  
p3.add(t2);  
p3.add(t3);  
p3.add(t4);  
p3.add(t5);  
p3.add(t6);  
p3.add(t7);  
p3.add(t8);  
p3.add(t9);  
p3.add(t10);  
p3.add(t11);  
p3.add(t12);  
p3.add(t13);
```

```
p3.add(cb1);
```

```
p3.add(cb2);  
p3.add(cb3);
```

```
cn.add(p1);  
cn.add(p2);  
cn.add(p3);  
cn.add(p4);
```

```
l1.setBorder(edge);  
l2.setBorder(edge);  
l3.setBorder(edge);  
l4.setBorder(edge);  
l5.setBorder(edge);  
l6.setBorder(edge);  
l7.setBorder(edge);  
l8.setBorder(edge);
```



## COLLEGE MANAGEMENT SYSTEM



```
l9.setBorder(edge);
l10.setBorder(edge);
l11.setBorder(edge);
l12.setBorder(edge);
l13.setBorder(edge);
l14.setBorder(edge);
l15.setBorder(edge);
l16.setBorder(edge);

t1.setBorder(edge1);
t2.setBorder(edge1);
t3.setBorder(edge1);
t4.setBorder(edge1);
t5.setBorder(edge1);
t6.setBorder(edge1);
t7.setBorder(edge1);
t8.setBorder(edge1);
t9.setBorder(edge1);
t10.setBorder(edge1);
t11.setBorder(edge1);
t12.setBorder(edge1);
t13.setBorder(edge1);

setVisible(true);
setSize(1265,570);
setLocation(5,95);
connect();
enoLoad();
}
public void enoLoad()
{
    Connection con = null;
    Statement stm = null;
    ResultSet rs;
    String qry,rg;
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con = DriverManager.getConnection
("jdbc:odbc:adsn","system","123");
        stm = con.createStatement();
        qry = "select eno from StaffDetail;";
        rs = stm.executeQuery(qry);

        while(rs.next())
        {
```



## COLLEGE MANAGEMENT SYSTEM



```
        rg = rs.getString("eno");

        cb1.addItem(rg);
    }
}
catch(Exception e1)
{
}
finally
{
    try
    {
        stm.close();
        con.close();
    }
    catch(SQLException e1)
    {
    }
}
}

public class newListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            t1.setText(" ");
            t2.setText(" ");
            t3.setText(" ");
            t4.setText(" ");
            t5.setText(" ");
            t6.setText(" ");
            t7.setText(" ");
            t8.setText(" ");
            t9.setText(" ");
            t10.setText(" ");
            t11.setText(" ");
            t12.setText(" ");
            t13.setText(" ");

            cb1.setSelectedIndex(0);
            cb2.setSelectedIndex(0);
            cb3.setSelectedIndex(0);
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        catch(Exception e1)
        {
        }
    }
}
private class insertListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = t1.getText();
            if (str1.compareTo("") == 0)

JOptionPane.showMessageDialog(null,"Employee name can't be null");
            else
            {
                str1 = (String) cb1.getSelectedItem();
                str2 = t1.getText();
                str3 = (String) cb2.getSelectedItem();
                str4 = (String) cb3.getSelectedItem();
                str5 = t2.getText();

                str6 = t3.getText();

                str7 = t4.getText();
                str8 = t5.getText();
                str9 = t6.getText();
                str10 = t7.getText();
                str11 = t8.getText();
                str12 = t9.getText();
                str13 = t10.getText();
                str14 = t11.getText();
                str15 = t12.getText();
                str16 = t13.getText();

                stm.executeUpdate("insert into payroll
values("+str1+", '"+str2+"', '"+str3+"', '"+str4+"', '"+str5+"', '"+str6+"', '"+str7+"', '"+str8+"',
 '"+str9+"', '"+str10+"', '"+str11+"', '"+str12+"', '"+str13+"', '"+str14+"', '"+str15+"', '"+str16+"'
)");

                stm.executeUpdate("commit");
JOptionPane.showMessageDialog(null,"Record
Inserted");

                load();
            } //end of else
        } //end of try
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        catch(Exception sql)
        {
            JOptionPane.showMessageDialog(null,sql);
        }
    }
}

public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            if(t1.getText().compareTo("")==0)
                JOptionPane.showMessageDialog(null,"Fill all
Fields");

            else
            {
                str1 = (String) cb1.getSelectedItem();
                str2 = t1.getText();
                str3 = (String) cb2.getSelectedItem();
                str4 = (String) cb3.getSelectedItem();
                str5 = t2.getText();

                str6 = t3.getText();
                str7 = t4.getText();
                str8 = t5.getText();
                str9 = t6.getText();
                str10 = t7.getText();
                str11 = t8.getText();
                str12 = t9.getText();
                str13 = t10.getText();
                str14 = t11.getText();
                str15 = t12.getText();
                str16 = t13.getText();

                stm.executeUpdate("update payroll set ename
= '"+str2+"',month = '"+str3+"',year = '"+str4+"',bp = '"+str5+"',hra = '"+str6+"',da =
 '"+str7+"',ma = '"+str8+"',ta = '"+str9+"',pf = '"+str10+"',esi = '"+str11+"',advance =
 '"+str12+"',lic = '"+str13+"',other = '"+str14+"',nsalary = '"+str15+"',remarks =
 '"+str16+" where eno = '"+str1);

                stm.executeUpdate("commit");
                JOptionPane.showMessageDialog(null,"Records Updated ");
            }
        }
        catch(SQLException sql)
        {
```





## COLLEGE MANAGEMENT SYSTEM



```
Updated"+sql);
    }
}
}
public class viewListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = (String) cb1.getSelectedItem();
            rs = stm.executeQuery("select * from payroll where
eno = "+str1);

            while(rs.next())
            {

                t1.setText(rs.getString(2));
                cb2.setSelectedItem(rs.getString(3));

                cb3.setSelectedItem(rs.getString(4));
                t2.setText(rs.getString(5));

                t3.setText(rs.getString(6));
                t4.setText(rs.getString(7));

                t5.setText(rs.getString(8));
                t6.setText(rs.getString(9));
                t7.setText(rs.getString(10));
                t8.setText(rs.getString(11));
                t9.setText(rs.getString(12));
                t10.setText(rs.getString(13));
                t11.setText(rs.getString(14));
                t12.setText(rs.getString(15));
                t13.setText(rs.getString(16));

            }
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not found");
        }
    }
}
public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    try
    {
        st = JOptionPane.showInputDialog("Select
Employee number to be deleted");
        stm.executeUpdate("delete from payroll where eno =
st");
        stm.executeUpdate("commit");
        JOptionPane.showMessageDialog(null,"Record is
deleted");
    }
    catch(SQLException sql)
    {
        JOptionPane.showMessageDialog(null,"Not
deleted");
    }
}
protected void processWindowEvent(WindowEvent we)
{
    if(we.getID()==we.WINDOW_CLOSING)
    {
        r=JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
        if(r == JOptionPane.YES_OPTION)
        dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Exception"+sql);
        }
    }
    catch(Exception e)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    JOptionPane.showMessageDialog(null,"Not Connected");
}
}
public class loadListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        load();
    }
}
public void load()
{
    int k;
    k=0;
    try
    {
        stm.executeUpdate("commit");
        rs = stm.executeQuery("select
eno,ename,month,year,bp,hra,da,ma,ta,pf,esi,advance,lic,other,nsalary,rem
arks from payroll order by eno");

        while(rs.next())
        {
            str1 = rs.getString(1);
            str2 = rs.getString(2);
            str3 = rs.getString(3);
            str4 = rs.getString(4);
            str5 = rs.getString(5);
            str6 = rs.getString(6);
            str7 = rs.getString(7);
            str8 = rs.getString(8);
            str9 = rs.getString(9);
            str10 = rs.getString(10);
            str11 = rs.getString(11);
            str12 = rs.getString(12);
            str13 = rs.getString(13);
            str14 = rs.getString(14);
            str15 = rs.getString(15);
            str16 = rs.getString(16);

            data[k][0]=str1;
            data[k][1]=str2;
            data[k][2]=str3;
            data[k][3]=str4;
            data[k][4]=str5;
```



## COLLEGE MANAGEMENT SYSTEM



```
data[k][5]=str6;
data[k][6]=str7;
data[k][7]=str8;
data[k][8]=str9;
data[k][9]=str10;
data[k][10]=str11;
data[k][11]=str12;
data[k][12]=str13;
data[k][13]=str14;
data[k][14]=str15;
data[k][15]=str16;

k++;
}
data[k][0] = " ";
data[k][1] = " ";
data[k][2] = " ";
data[k][3] = " ";
data[k][4] = " ";
data[k][5] = " ";
data[k][6] = " ";
data[k][7] = " ";
data[k][8] = " ";
data[k][9] = " ";
data[k][10] = " ";
data[k][11] = " ";
data[k][12] = " ";
data[k][13] = " ";
data[k][14] = " ";
data[k][15] = " ";

JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
table.setForeground(Color.blue);
JScrollPane jsp = new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);
jsp.setBounds(12,305,910,150);
p3.add(jsp);
}
catch(SQLException sqle)
{
JOptionPane.showMessageDialog(null,"Record not transfer");
}
```



## COLLEGE MANAGEMENT SYSTEM



```
}
private class enterListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==t1)
            t2.requestFocus();
        if(e.getSource()==t2)
            t3.requestFocus();
        if(e.getSource()==t3)

            t4.requestFocus();
        if(e.getSource()==t4)
            t5.requestFocus();
        if(e.getSource()==t5)
            t6.requestFocus();
        if(e.getSource()==t6)
            t7.requestFocus();
        if(e.getSource()==t7)
            t8.requestFocus();
        if(e.getSource()==t8)
            t9.requestFocus();

        if(e.getSource()==t9)
            t10.requestFocus();

        if(e.getSource()==t10)
            t11.requestFocus();
        if(e.getSource()==t11)
        {
            double z1,z2,z3,z4,z5,z6,z7,z8,z9,z10,z11;

            z1 = Double.parseDouble(t2.getText());
            z2 = Double.parseDouble(t3.getText());
            z3 = Double.parseDouble(t4.getText());
            z4 = Double.parseDouble(t5.getText());
            z5 = Double.parseDouble(t6.getText());
            z6 = Double.parseDouble(t7.getText());
            z7 = Double.parseDouble(t8.getText());
            z8 = Double.parseDouble(t9.getText());
            z9 = Double.parseDouble(t10.getText());
            z10 = Double.parseDouble(t11.getText());

            z11 = z1 + z2 + z3 + z4 + z5 + z7 + z8 + z9 + z10;
            z11 = z11 - z6;
            t12.setText(String.valueOf(z11));
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        t12.requestFocus();
    }
    if(e.getSource()==t12)
        t13.requestFocus();
    }
}

private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l19.setLocation(r,40);
        l18.setLocation(r,20);
        l17.setLocation(r,0);
        r = r+1;
        if(r>290)
            r = 0;
    }
}

private class timerListener1 implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(l,10);

        l = l - 1;
        if( l < 0)
            l = 1240;
    }
}

public class exitListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        dispose();
    }
}
}
```



**/\* BCA Result \*/**

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class BcaResult extends JFrame
{
    private JLabel
title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17;
    private JTextField t1,t2,t3,t4,t5,t6,t7,t8,t9,t10;
    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;
    private JComboBox cb1,cb2,cb3,cb4;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;
    ResultSet rs;
    String
str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11,str12,str13,str14,str,st;
    Connection con;
    Statement stm;
    Dimension screensize;
    String[] colhead = {"Reg no.,"Student
name","Course","Semester","Paper1","Paper2","Paper3","Paper4","Paper5","Paper
6","Practical1","Practical2","Percentage","Grade"};
    Object data[][] = new Object[50][14];
    JTable table = new JTable(data,colhead);
    public BcaResult()
    {
        super("BCA RESULT FORM");
        l=1;r=1;
        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);
        cn.setBackground(new Color(128,128,25));

        EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);
        BevelBorder edge=new BevelBorder(BevelBorder.RAISED);
```



## COLLEGE MANAGEMENT SYSTEM



```
p1 = new JPanel();  
p2 = new JPanel();  
p3 = new JPanel();  
p4 = new JPanel();
```

```
p1.setBounds(5,5,1240,50);  
p2.setBounds(5,60,300,400);  
p3.setBounds(310,60,935,465);  
p4.setBounds(5,465,300,60);
```

```
p1.setBackground(new Color(52,13,125));  
p2.setBackground(new Color(112,0,0));  
p3.setBackground(new Color(64,0,0));  
p4.setBackground(new Color(254,200,0));
```

```
p3.setForeground(Color.white);
```

```
title = new JLabel("WELCOME SCREEN FOR BCA RESULT");  
title.setFont(new Font("STENCIL",1,40));  
title.setForeground(new Color(211,255,255));  
p1.add(title);
```

```
Timer tm = new Timer(100,new timerListener());  
tm.start();
```

```
Timer tm1 = new Timer(5,new timerListener1());  
tm1.start();
```

```
pic = new JLabel();  
Icon img=new ImageIcon("bca.jpg");  
pic.setIcon(img);  
p2.add(pic);
```

```
p3.setLayout(null);  
p4.setLayout(null);
```

```
p3.setBorder(BorderFactory.createTitledBorder("Enter Information  
for BCA result"));
```

```
l1 = new JLabel("REG NUMBER");  
l2 = new JLabel("STUDENT NAME");  
l3 = new JLabel("COURSE");  
l4 = new JLabel("SEMESTER");  
l5 = new JLabel("PAPER 1");
```





## COLLEGE MANAGEMENT SYSTEM



```
l6 = new JLabel("PAPER 2");
l7 = new JLabel("PAPER 3");
l8 = new JLabel("PAPER 4");
l9 = new JLabel("PAPER 5");
l10 = new JLabel("PAPER 6");
l11 = new JLabel("PRACTICAL 1");
l12 = new JLabel("PRACTICAL 2");
l13 = new JLabel("PERCENTAGE");
l14 = new JLabel("GRADE");

l15 = new JLabel("Developed by:- ");
l15.setFont(new Font("Times new Roman",1,15));
l15.setForeground(Color.blue);
l15.setBounds(0,850,140,25);
p4.add(l15);
```

```
l16=new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.
Tauqueer , Motahhir , Sanoj");
l16.setFont(new Font("Monotype Corsiva",1,15));
l16.setForeground(Color.blue);
l16.setBounds(50,850,220,25);
p4.add(l16);
```

```
l17=new JLabel("collegemanagementsystem@gmail.com");
l17.setFont(new Font("Monotype Corsiva",1,15));
l17.setForeground(Color.blue);
l17.setBounds(50,850,220,25);
p4.add(l17);
```

```
t1 = new JTextField(15);
t2 = new JTextField(15);
t3 = new JTextField(15);
t4 = new JTextField(15);
t5 = new JTextField(15);
t6 = new JTextField(15);
t7 = new JTextField(15);
t8 = new JTextField(15);
t9 = new JTextField(15);
t10 = new JTextField(15);
```

```
cb1 = new JComboBox();
cb2 = new JComboBox();
cb2.addItem("-----SELECT-----");
cb2.addItem("BCA");
cb3 = new JComboBox();
```



## COLLEGE MANAGEMENT SYSTEM



```
cb3.addItem("SELECT SEMESTER");
cb3.addItem("I");
cb3.addItem("II");
cb3.addItem("III");
cb3.addItem("IV");
cb3.addItem("V");
cb3.addItem("VI");
```

```
cb4 = new JComboBox();
cb4.addItem("SELECT GRADE");
cb4.addItem("A1");
cb4.addItem("A2");
cb4.addItem("B1");
cb4.addItem("B2");
cb4.addItem("c1");
cb4.addItem("C2");
cb4.addItem("D");
cb4.addItem("E1");
cb4.addItem("E2");
```

```
b1 = new JButton("NEW");
b2 = new JButton("INSERT");
b3 = new JButton("UPDATE");
b4 = new JButton("VIEW");
b5 = new JButton("DELETE");
b6 = new JButton("LOAD");
```

```
JTable table = new JTable(data,colhead);
table.setForeground(Color.blue);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
```

```
JScrollPane jsp=new JScrollPane(table);
```

```
jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
```

```
p3.add(jsp);
b1.setToolTipText("Refresh the page");
b2.setToolTipText("Insert the records");
b3.setToolTipText("Update the records");
b4.setToolTipText("View the records");
b5.setToolTipText("Delete the records");
b6.setToolTipText("Load the records");
```

```
b1.setCursor(cor);
b2.setCursor(cor);
b3.setCursor(cor);
```



## COLLEGE MANAGEMENT SYSTEM



```
b4.setCursor(cor);  
b5.setCursor(cor);  
b6.setCursor(cor);
```

```
b1.setMnemonic('n');  
b2.setMnemonic('i');  
b3.setMnemonic('u');  
b4.setMnemonic('v');  
b5.setMnemonic('d');  
b6.setMnemonic('l');
```

```
b1.setFont(new Font("Times New Roman",1,16));  
b2.setFont(new Font("Times New Roman",1,16));  
b3.setFont(new Font("Times New Roman",1,16));  
b4.setFont(new Font("Times New Roman",1,16));  
b5.setFont(new Font("Times New Roman",1,16));  
b6.setFont(new Font("Times New Roman",1,16));
```

```
l1.setFont(new Font("Times New Roman",1,16));  
l2.setFont(new Font("Times New Roman",1,16));  
l3.setFont(new Font("Times New Roman",1,16));  
l4.setFont(new Font("Times New Roman",1,16));  
l5.setFont(new Font("Times New Roman",1,16));  
l6.setFont(new Font("Times New Roman",1,16));  
l7.setFont(new Font("Times New Roman",1,16));  
l8.setFont(new Font("Times New Roman",1,16));  
l9.setFont(new Font("Times New Roman",1,16));  
l10.setFont(new Font("Times New Roman",1,16));  
l11.setFont(new Font("Times New Roman",1,16));  
l12.setFont(new Font("Times New Roman",1,16));  
l13.setFont(new Font("Times New Roman",1,16));  
l14.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));  
t2.setFont(new Font("Times New Roman",1,16));  
t3.setFont(new Font("Times New Roman",1,16));  
t4.setFont(new Font("Times New Roman",1,16));  
t5.setFont(new Font("Times New Roman",1,16));  
t6.setFont(new Font("Times New Roman",1,16));  
t7.setFont(new Font("Times New Roman",1,16));  
t8.setFont(new Font("Times New Roman",1,16));  
t9.setFont(new Font("Times New Roman",1,16));  
t10.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));  
cb2.setFont(new Font("Times New Roman",1,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
cb3.setFont(new Font("Times New Roman",1,16));  
cb4.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(Color.white);  
l2.setForeground(Color.white);  
l3.setForeground(Color.white);  
l4.setForeground(Color.white);  
l5.setForeground(Color.white);  
l6.setForeground(Color.white);  
l7.setForeground(Color.white);  
l8.setForeground(Color.white);  
l9.setForeground(Color.white);  
l10.setForeground(Color.white);  
l11.setForeground(Color.white);  
l12.setForeground(Color.white);  
l13.setForeground(Color.white);  
l14.setForeground(Color.white);
```

```
t1.setForeground(Color.blue);  
t2.setForeground(Color.blue);  
t3.setForeground(Color.blue);  
t4.setForeground(Color.blue);  
t5.setForeground(Color.blue);  
t6.setForeground(Color.blue);  
t7.setForeground(Color.blue);  
t8.setForeground(Color.blue);  
t9.setForeground(Color.blue);  
t10.setForeground(Color.blue);
```

```
cb1.setForeground(Color.blue);  
cb2.setForeground(Color.blue);  
cb3.setForeground(Color.blue);  
cb4.setForeground(Color.blue);
```

```
t1.addActionListener(new enterListener());  
t2.addActionListener(new enterListener());  
t3.addActionListener(new enterListener());  
t4.addActionListener(new enterListener());  
t5.addActionListener(new enterListener());  
t6.addActionListener(new enterListener());  
t7.addActionListener(new enterListener());  
t8.addActionListener(new enterListener());  
t9.addActionListener(new enterListener());  
t10.addActionListener(new enterListener());
```

```
b1.addActionListener(new newListener());
```



## COLLEGE MANAGEMENT SYSTEM



```
b2.addActionListener(new insertListener());  
b3.addActionListener(new updateListener());  
b4.addActionListener(new viewListener());  
b5.addActionListener(new deleteListener());  
b6.addActionListener(new loadListener());
```

```
l1.setBounds(30,30,145,25);  
l2.setBounds(30,70,145,25);  
l3.setBounds(30,110,145,25);  
l4.setBounds(30,150,145,25);  
l5.setBounds(30,190,145,25);  
l6.setBounds(30,230,145,25);  
l7.setBounds(30,270,145,25);  
l8.setBounds(400,30,145,25);  
l9.setBounds(400,70,145,25);  
l10.setBounds(400,110,145,25);  
l11.setBounds(400,150,145,25);  
l12.setBounds(400,190,145,25);  
l13.setBounds(400,230,145,25);  
l14.setBounds(400,270,145,25);
```

```
cb1.setBounds(190,30,185,25);  
t1.setBounds(190,70,185,25);  
cb2.setBounds(190,110,185,25);  
cb3.setBounds(190,150,185,25);  
t2.setBounds(190,190,185,25);  
t3.setBounds(190,230,185,25);
```

```
t4.setBounds(190,270,185,25);
```

```
t5.setBounds(560,30,185,25);  
t6.setBounds(560,70,185,25);  
t7.setBounds(560,110,185,25);  
t8.setBounds(560,150,185,25);  
t9.setBounds(560,190,185,25);  
t10.setBounds(560,230,185,25);  
cb4.setBounds(560,270,185,25);
```

```
b1.setBounds(780,30,100,30);  
b2.setBounds(780,76,100,30);  
b3.setBounds(780,122,100,30);  
b4.setBounds(780,168,100,30);  
b5.setBounds(780,214,100,30);  
b6.setBounds(780,260,100,30);
```

```
jsp.setBounds(12,300,910,160);
```



## COLLEGE MANAGEMENT SYSTEM



```
p3.add(b1);  
p3.add(b2);  
p3.add(b3);  
p3.add(b4);  
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);  
p3.add(l2);  
p3.add(l3);  
p3.add(l4);  
p3.add(l5);  
p3.add(l6);  
p3.add(l7);  
p3.add(l8);  
p3.add(l9);  
p3.add(l10);  
p3.add(l11);  
p3.add(l12);  
p3.add(l13);  
p3.add(l14);
```

```
p3.add(t1);  
p3.add(t2);  
p3.add(t3);
```

```
p3.add(t4);  
p3.add(t5);  
p3.add(t6);  
p3.add(t7);  
p3.add(t8);  
p3.add(t9);  
p3.add(t10);
```

```
p3.add(cb1);  
p3.add(cb2);  
p3.add(cb3);  
p3.add(cb4);
```

```
cn.add(p1);  
cn.add(p2);  
cn.add(p3);  
cn.add(p4);
```

```
l1.setBorder(edge);
```



## COLLEGE MANAGEMENT SYSTEM



```
l2.setBorder(edge);
l3.setBorder(edge);
l4.setBorder(edge);
l5.setBorder(edge);
l6.setBorder(edge);
l7.setBorder(edge);
l8.setBorder(edge);
l9.setBorder(edge);
l10.setBorder(edge);
l11.setBorder(edge);
l12.setBorder(edge);
l13.setBorder(edge);
l14.setBorder(edge);

t1.setBorder(edge1);
t2.setBorder(edge1);
t3.setBorder(edge1);
t4.setBorder(edge1);
t5.setBorder(edge1);
t6.setBorder(edge1);
t7.setBorder(edge1);
t8.setBorder(edge1);
t9.setBorder(edge1);
t10.setBorder(edge1);

setVisible(true);
setSize(1265,570);
setLocation(5,95);

connect();
regLoad();
}
public void regLoad()
{
    Connection con = null;
    Statement stm = null;
    ResultSet rs;
    String qry,rg;
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con = DriverManager.getConnection
("jdbc:odbc:adsn","system","123");
        stm = con.createStatement();
        qry = "select reg from studentdetail";
```



## COLLEGE MANAGEMENT SYSTEM



```
rs = stm.executeQuery(qry);

while(rs.next())
{
    rg = rs.getString("reg");

    cb1.addItem(rg);
}
}
catch(Exception e1)
{
}
finally
{
    try
    {
        stm.close();
        con.close();
    }
    catch(SQLException e1)
    {
    }
}
}

public class newListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            t1.setText(" ");
            t2.setText(" ");
            t3.setText(" ");
            t4.setText(" ");
            t5.setText(" ");
            t6.setText(" ");
            t7.setText(" ");
            t8.setText(" ");
            t9.setText(" ");
            t10.setText(" ");

            cb1.setSelectedItem(0);
            cb2.setSelectedIndex(0);
            cb3.setSelectedIndex(0);
            cb4.setSelectedIndex(0);
        }
    }
}
```





## COLLEGE MANAGEMENT SYSTEM



```
        }
        catch(Exception e1)
        {
        }
    }
}

private class insertListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {

        try
        {

            str1 = (String) cb1.getSelectedItem();
            str2 = t1.getText();
            str3 = (String) cb2.getSelectedItem();
            str4 = (String) cb3.getSelectedItem();
            str5 = t2.getText();
            str6 = t3.getText();
            str7 = t4.getText();

            str8 = t5.getText();
            str9 = t6.getText();
            str10 = t7.getText();
            str11 = t8.getText();
            str12 = t9.getText();
            str13 = t10.getText();

            str14 = (String) cb4.getSelectedItem();

            stm.executeUpdate("insert into BcaResult
values('"+str1+"','"+str2+"','"+str3+"','"+str4+"','"+str5+"','"+str6+"','"+str7+"','"+str8+
','"+str9+"','"+str10+"','"+str11+"','"+str12+"','"+str13+"','"+str14+"')");
            stm.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"Record
Inserted");

            load();

        }//end of try
        catch(Exception sql)
        {
            JOptionPane.showMessageDialog(null,sql);
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
    }
    }
}

public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = (String) cb1.getSelectedItem();
            str2 = t1.getText();
            str3 = (String) cb2.getSelectedItem();
            str4 = (String) cb3.getSelectedItem();
            str5 = t2.getText();
            str6 = t3.getText();
            str7 = t4.getText();
            str8 = t5.getText();
            str9 = t6.getText();
            str10 = t7.getText();
            str11 = t8.getText();
            str12 = t9.getText();

            str13 = t10.getText();

            str14 = (String) cb4.getSelectedItem();

            stm.executeUpdate("update BcaResult set
name = '"+str2+"',course = '"+str3+"',semester = '"+str4+"',paper1 =
 '"+str5+"',paper2 = '"+str6+"',paper3 = '"+str7+"',paper4 = '"+str8+"',paper5 =
 '"+str9+"',paper6 = '"+str10+"',pract1 = '"+str11+"',pract2 = '"+str12+"',percentage =
 '"+str13+"',grade = '"+str14 +"' where reg = '"+str1+"'");
            stm.executeUpdate("commit");

            JOptionPane.showMessageDialog(null,"Records Updated ");
            load();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Could not
Updated"+sql);
        }
    }
}

public class viewListener implements ActionListener
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("Enter
Registration number to be view");
            rs = stm.executeQuery("select * from BcaResult
where reg = '"+st+"'");
            while(rs.next())
            {

                cb1.setSelectedItem(rs.getString(1));
                t1.setText(rs.getString(2));

                cb2.setSelectedItem(rs.getString(3));
                cb3.setSelectedItem(rs.getString(4));
                t2.setText(rs.getString(5));

                t3.setText(rs.getString(6));
                t4.setText(rs.getString(7));

                t5.setText(rs.getString(8));
                t6.setText(rs.getString(9));

                t7.setText(rs.getString(10));
                t8.setText(rs.getString(11));

                t9.setText(rs.getString(12));
                t10.setText(rs.getString(13));
                cb4.setSelectedItem(rs.getString(14));

            }
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not found");
        }
    }
}

public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
```



## COLLEGE MANAGEMENT SYSTEM



```
        {
            st = JOptionPane.showInputDialog("enter
Registration number to be deleted");
            stm.executeUpdate("delete from BcaResult where
reg = '"+st+"'");
            stm.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"Record is
deleted");
            load();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not
deleted");
        }
    }

    protected void processWindowEvent(WindowEvent we)
    {
        if(we.getID()==we.WINDOW_CLOSING)
        {
            r=JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
            if(r == JOptionPane.YES_OPTION)
                dispose();
        }
    }

    public void connect()
    {
        try
        {
            try
            {
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
                stm = con.createStatement();
            }
            catch(SQLException sql)
            {
                JOptionPane.showMessageDialog(null,"Exception"+sql);
            }
        }
        catch(Exception e)
    }
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    JOptionPane.showMessageDialog(null,"Not Connected");
}
}
public class loadListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        load();
    }
}

public void load()
{
    int k;
    k=0;

    try
    {
        stm.executeUpdate("commit");
        rs = stm.executeQuery("select
reg,name,course,semester,paper1,paper2,paper3,paper4,paper5,paper6,pract1,pract
2,percentage,grade from bcaresult order by reg");

        while(rs.next())
        {
            str1 = rs.getString(1);
            str2 = rs.getString(2);
            str3 = rs.getString(3);
            str4 = rs.getString(4);
            str5 = rs.getString(5);
            str6 = rs.getString(6);
            str7 = rs.getString(7);
            str8 = rs.getString(8);
            str9 = rs.getString(9);
            str10 = rs.getString(10);
            str11 = rs.getString(11);
            str12 = rs.getString(12);
            str13 = rs.getString(13);
            str14 = rs.getString(14);

            data[k][0]=str1;
            data[k][1]=str2;
            data[k][2]=str3;
            data[k][3]=str4;
            data[k][4]=str5;
```



## COLLEGE MANAGEMENT SYSTEM



```
data[k][5]=str6;
data[k][6]=str7;
data[k][7]=str8;
data[k][8]=str9;
data[k][9]=str10;
data[k][10]=str11;
data[k][11]=str12;
data[k][12]=str13;
data[k][13]=str14;

k++;
}
data[k][0] = " ";

data[k][1] = " ";
data[k][2] = " ";
data[k][3] = " ";
data[k][4] = " ";
data[k][5] = " ";
data[k][6] = " ";
data[k][7] = " ";
data[k][8] = " ";
data[k][9] = " ";
data[k][10] = " ";
data[k][11] = " ";
data[k][12] = " ";
data[k][13] = " ";
JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
table.setForeground(Color.blue);
JScrollPane jsp = new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
p3.add(jsp);
jsp.setBounds(12,300,910,160);
p3.add(jsp);
}
catch(SQLException sqle)
{
JOptionPane.showMessageDialog(null,"Record not transfer");
}
}
private class enterListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    if(e.getSource()==t1)
        t2.requestFocus();
    if(e.getSource()==t2)
        t3.requestFocus();
    if(e.getSource()==t3)
        t4.requestFocus();
    if(e.getSource()==t4)
        t5.requestFocus();
    if(e.getSource()==t5)

        t6.requestFocus();
    if(e.getSource()==t6)
        t7.requestFocus();
    if(e.getSource()==t7)
        t8.requestFocus();
    if(e.getSource()==t8)
        t9.requestFocus();
    if(e.getSource() == t9)
    {
        double z1,z2,z3,z4,z5,z6,z7,z8,z9;

        z1 = Double.parseDouble(t2.getText());
        z2 = Double.parseDouble(t3.getText());
        z3 = Double.parseDouble(t4.getText());
        z4 = Double.parseDouble(t5.getText());
        z5 = Double.parseDouble(t6.getText());
        z6 = Double.parseDouble(t7.getText());
        z7 = Double.parseDouble(t8.getText());
        z8 = Double.parseDouble(t9.getText());

        z9 = (z1 + z2 + z3 + z4 + z5 + z6 + z7 + z8)/8;

        t10.setText(String.valueOf(z9));

        if(z9>=91 && z9 <=100)
            cb4.setSelectedIndex(1);

        if(z9>=81 && z9 <=90)
            cb4.setSelectedIndex(2);

        if(z9>=71 && z9 <=80)
            cb4.setSelectedIndex(3);

        if(z9>=61 && z9 <=70)
            cb4.setSelectedIndex(4);
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        if(z9>=51 && z9 <=60)
            cb4.setSelectedIndex(5);

        if(z9>=41 && z9 <=50)
            cb4.setSelectedIndex(6);

        if(z9>=33 && z9 <=40)

            cb4.setSelectedIndex(7);

        if(z9>=21 && z9 <=32)
            cb4.setSelectedIndex(8);

        if(z9>=00 && z9 <=20)
            cb4.setSelectedIndex(9);
    }
}

private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l17.setLocation(r,40);
        l16.setLocation(r,20);
        l15.setLocation(r,0);
        r = r+1;
        if(r>290)
            r = 0;
    }
}

private class timerListener1 implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(l,10);

        l = l - 1;
        if( l < 0)
            l = 1240;
    }
}

public class exitListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
```





## COLLEGE MANAGEMENT SYSTEM



```
        dispose();
    }
}
```



## **/\* BBA Result \*/**

```
package module;
import java.io.*;
import java.lang.*;
import java.awt.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.JTable.*;
public class BbaResult extends JFrame
{
    private JLabel
title,pic,l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13,l14,l15,l16,l17;
    private JTextField t1,t2,t3,t4,t5,t6,t7,t8,t9,t10;
    private JButton b1,b2,b3,b4,b5,b6;
    private JPanel p1,p2,p3,p4;
    private JComboBox cb1,cb2,cb3,cb4;
    private Container cn;
    Cursor cor = new Cursor(Cursor.HAND_CURSOR);
    int f,r,k,s,l;
    ResultSet rs;
    String
str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11,str12,str13,str14,str,st;
    Connection con;
    Statement stm;
    Dimension screensize;
    String[] colhead = {"Reg no.,"Student
name","Course","Semester","Paper1","Paper2","Paper3","Paper4","Paper5","Paper
6","Practical1","Practical2","Percentage","Grade"};
    Object data[][] = new Object[50][14];
    JTable table = new JTable(data,colhead);
    public BbaResult()
    {
        super("BBA RESULT FORM");
        l=1;r=1;
        cn = getContentPane();
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);
        cn.setBackground(new Color(128,128,25));
        EtchedBorder edge1 = new EtchedBorder(EtchedBorder.RAISED);

        BevelBorder edge=new BevelBorder(BevelBorder.RAISED);

        p1 = new JPanel();
```



## COLLEGE MANAGEMENT SYSTEM



```
p2 = new JPanel();  
p3 = new JPanel();  
p4 = new JPanel();
```

```
p1.setBounds(5,5,1240,50);  
p2.setBounds(5,60,300,400);  
p3.setBounds(310,60,935,465);  
p4.setBounds(5,465,300,60);
```

```
p1.setBackground(new Color(64,0,64));  
p2.setBackground(new Color(112,0,0));  
p3.setBackground(new Color(220,187,47));  
p4.setBackground(new Color(254,200,0));
```

```
title = new JLabel("WELCOME SCREEN FOR BBA RESULT");  
title.setFont(new Font("STENCIL",1,40));  
title.setForeground(Color.white);  
p1.add(title);
```

```
Timer tm = new Timer(100,new timerListener());  
tm.start();
```

```
Timer tm1 = new Timer(5,new timerListener1());  
tm1.start();
```

```
pic = new JLabel();  
Icon img=new ImageIcon("bba.jpg");  
pic.setIcon(img);  
p2.add(pic);
```

```
p3.setLayout(null);  
p4.setLayout(null);
```

```
p3.setBorder(BorderFactory.createTitledBorder("Enter Information  
for BBA result"));
```

```
l1 = new JLabel("REG NUMBER");  
l2 = new JLabel("STUDENT NAME");  
l3 = new JLabel("COURSE");  
l4 = new JLabel("SEMESTER");  
l5 = new JLabel("PAPER 1");
```

```
l6 = new JLabel("PAPER 2");  
l7 = new JLabel("PAPER 3");  
l8 = new JLabel("PAPER 4");  
l9 = new JLabel("PAPER 5");
```



## COLLEGE MANAGEMENT SYSTEM



```
110 = new JLabel("PAPER 6");
111 = new JLabel("PRACTICAL 1");
112 = new JLabel("PRACTICAL 2");
113 = new JLabel("PERCENTAGE");
114 = new JLabel("GRADE");
```

```
115 = new JLabel("Developed by:- ");
115.setFont(new Font("Times new Roman",1,15));
115.setForeground(Color.blue);
115.setBounds(0,850,140,25);
p4.add(115);
```

```
116=new JLabel("Md. Amairul , Md. Arif , Md Mahtab , Md.
Tauqueer , Motahhir , Sanoj");
```

```
116.setFont(new Font("Monotype Corsiva",1,15));
116.setForeground(Color.blue);
116.setBounds(50,850,220,25);
p4.add(116);
```

```
117=new JLabel("collegemanagementsystem@gmail.com");
117.setFont(new Font("Monotype Corsiva",1,15));
117.setForeground(Color.blue);
117.setBounds(50,850,220,25);
p4.add(117);
```

```
t1 = new JTextField(15);
t2 = new JTextField(15);
t3 = new JTextField(15);
t4 = new JTextField(15);
t5 = new JTextField(15);
t6 = new JTextField(15);
t7 = new JTextField(15);
t8 = new JTextField(15);
t9 = new JTextField(15);
t10 = new JTextField(15);
```

```
cb1 = new JComboBox();
```

```
cb2 = new JComboBox();
cb2.addItem("-----SELECT-----");
```

```
cb2.addItem("BBA");
```

```
cb3 = new JComboBox();
cb3.addItem("SELECT SEMESTER");
```



## COLLEGE MANAGEMENT SYSTEM



```
cb3.addItem("I");
cb3.addItem("II");
cb3.addItem("III");
cb3.addItem("IV");
cb3.addItem("V");
cb3.addItem("VI");
```

```
cb4 = new JComboBox();
cb4.addItem("SELECT GRADE");
cb4.addItem("A1");
cb4.addItem("A2");
cb4.addItem("B1");
cb4.addItem("B2");
cb4.addItem("C1");
cb4.addItem("C2");
cb4.addItem("D");
cb4.addItem("E1");
cb4.addItem("E2");
```

```
b1 = new JButton("NEW");
b2 = new JButton("INSERT");
b3 = new JButton("UPDATE");
b4 = new JButton("VIEW");
b5 = new JButton("DELETE");
b6 = new JButton("LOAD");
```

```
JTable table = new JTable(data,colhead);
table.setForeground(Color.blue);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
```

```
JScrollPane jsp=new JScrollPane(table);
```

```
jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
```

```
p3.add(jsp);
```

```
b1.setToolTipText("Refresh the page");
b2.setToolTipText("Insert the records");
b3.setToolTipText("Update the records");
b4.setToolTipText("View the records");
```

```
b5.setToolTipText("Delete the records");
b6.setToolTipText("Load the records");
```

```
b1.setCursor(cor);
b2.setCursor(cor);
```



## COLLEGE MANAGEMENT SYSTEM



```
b3.setCursor(cor);  
b4.setCursor(cor);  
b5.setCursor(cor);  
b6.setCursor(cor);
```

```
b1.setMnemonic('n');  
b2.setMnemonic('i');  
b3.setMnemonic('u');  
b4.setMnemonic('v');  
b5.setMnemonic('d');  
b6.setMnemonic('l');
```

```
b1.setFont(new Font("Times New Roman",1,16));  
b2.setFont(new Font("Times New Roman",1,16));  
b3.setFont(new Font("Times New Roman",1,16));  
b4.setFont(new Font("Times New Roman",1,16));  
b5.setFont(new Font("Times New Roman",1,16));  
b6.setFont(new Font("Times New Roman",1,16));
```

```
l1.setFont(new Font("Times New Roman",1,16));  
l2.setFont(new Font("Times New Roman",1,16));  
l3.setFont(new Font("Times New Roman",1,16));  
l4.setFont(new Font("Times New Roman",1,16));  
l5.setFont(new Font("Times New Roman",1,16));  
l6.setFont(new Font("Times New Roman",1,16));  
l7.setFont(new Font("Times New Roman",1,16));  
l8.setFont(new Font("Times New Roman",1,16));  
l9.setFont(new Font("Times New Roman",1,16));  
l10.setFont(new Font("Times New Roman",1,16));  
l11.setFont(new Font("Times New Roman",1,16));  
l12.setFont(new Font("Times New Roman",1,16));  
l13.setFont(new Font("Times New Roman",1,16));  
l14.setFont(new Font("Times New Roman",1,16));
```

```
t1.setFont(new Font("Times New Roman",1,16));  
t2.setFont(new Font("Times New Roman",1,16));  
t3.setFont(new Font("Times New Roman",1,16));
```

```
t4.setFont(new Font("Times New Roman",1,16));  
t5.setFont(new Font("Times New Roman",1,16));  
t6.setFont(new Font("Times New Roman",1,16));  
t7.setFont(new Font("Times New Roman",1,16));  
t8.setFont(new Font("Times New Roman",1,16));  
t9.setFont(new Font("Times New Roman",1,16));
```



## COLLEGE MANAGEMENT SYSTEM



```
t10.setFont(new Font("Times New Roman",1,16));
```

```
cb1.setFont(new Font("Times New Roman",1,16));  
cb2.setFont(new Font("Times New Roman",1,16));  
cb3.setFont(new Font("Times New Roman",1,16));  
cb4.setFont(new Font("Times New Roman",1,16));
```

```
l1.setForeground(Color.blue);  
l2.setForeground(Color.blue);  
l3.setForeground(Color.blue);  
l4.setForeground(Color.blue);  
l5.setForeground(Color.blue);  
l6.setForeground(Color.blue);  
l7.setForeground(Color.blue);  
l8.setForeground(Color.blue);  
l9.setForeground(Color.blue);  
l10.setForeground(Color.blue);  
l11.setForeground(Color.blue);  
l12.setForeground(Color.blue);  
l13.setForeground(Color.blue);  
l14.setForeground(Color.blue);
```

```
t1.setForeground(Color.blue);  
t2.setForeground(Color.blue);  
t3.setForeground(Color.blue);  
t4.setForeground(Color.blue);  
t5.setForeground(Color.blue);  
t6.setForeground(Color.blue);  
t7.setForeground(Color.blue);  
t8.setForeground(Color.blue);  
t9.setForeground(Color.blue);  
t10.setForeground(Color.blue);
```

```
cb1.setForeground(Color.blue);  
cb2.setForeground(Color.blue);  
cb3.setForeground(Color.blue);
```

```
cb4.setForeground(Color.blue);
```

```
t1.addActionListener(new enterListener());  
t2.addActionListener(new enterListener());  
t3.addActionListener(new enterListener());  
t4.addActionListener(new enterListener());  
t5.addActionListener(new enterListener());
```



## COLLEGE MANAGEMENT SYSTEM



```
t6.addActionListener(new enterListener());  
t7.addActionListener(new enterListener());  
t8.addActionListener(new enterListener());  
t9.addActionListener(new enterListener());  
t10.addActionListener(new enterListener());
```

```
b1.addActionListener(new newListener());  
b2.addActionListener(new insertListener());  
b3.addActionListener(new updateListener());  
b4.addActionListener(new viewListener());  
b5.addActionListener(new deleteListener());  
b6.addActionListener(new loadListener());
```

```
l1.setBounds(30,30,145,25);  
l2.setBounds(30,70,145,25);  
l3.setBounds(30,110,145,25);  
l4.setBounds(30,150,145,25);  
l5.setBounds(30,190,145,25);  
l6.setBounds(30,230,145,25);  
l7.setBounds(30,270,145,25);  
l8.setBounds(400,30,145,25);  
l9.setBounds(400,70,145,25);  
l10.setBounds(400,110,145,25);  
l11.setBounds(400,150,145,25);  
l12.setBounds(400,190,145,25);  
l13.setBounds(400,230,145,25);  
l14.setBounds(400,270,145,25);
```

```
cb1.setBounds(190,30,185,25);  
t1.setBounds(190,70,185,25);  
cb2.setBounds(190,110,185,25);  
cb3.setBounds(190,150,185,25);  
t2.setBounds(190,190,185,25);  
t3.setBounds(190,230,185,25);  
t4.setBounds(190,270,185,25);
```

```
t5.setBounds(560,30,185,25);  
t6.setBounds(560,70,185,25);  
t7.setBounds(560,110,185,25);  
t8.setBounds(560,150,185,25);  
t9.setBounds(560,190,185,25);  
t10.setBounds(560,230,185,25);  
cb4.setBounds(560,270,185,25);
```





## COLLEGE MANAGEMENT SYSTEM



```
b1.setBounds(780,30,100,30);  
b2.setBounds(780,76,100,30);  
b3.setBounds(780,122,100,30);  
b4.setBounds(780,168,100,30);  
b5.setBounds(780,214,100,30);  
b6.setBounds(780,260,100,30);
```

```
jsp.setBounds(12,300,910,160);
```

```
p3.add(b1);  
p3.add(b2);  
p3.add(b3);  
p3.add(b4);  
p3.add(b5);  
p3.add(b6);
```

```
p3.add(l1);  
p3.add(l2);  
p3.add(l3);  
p3.add(l4);  
p3.add(l5);  
p3.add(l6);  
p3.add(l7);  
p3.add(l8);  
p3.add(l9);  
p3.add(l10);  
p3.add(l11);  
p3.add(l12);  
p3.add(l13);  
p3.add(l14);
```

```
p3.add(t1);  
p3.add(t2);  
p3.add(t3);  
p3.add(t4);  
p3.add(t5);
```

```
p3.add(t6);  
p3.add(t7);  
p3.add(t8);  
p3.add(t9);  
p3.add(t10);
```

```
p3.add(cb1);  
p3.add(cb2);  
p3.add(cb3);
```



## COLLEGE MANAGEMENT SYSTEM



```
p3.add(cb4);

cn.add(p1);
cn.add(p2);
cn.add(p3);
cn.add(p4);

l1.setBorder(edge);
l2.setBorder(edge);
l3.setBorder(edge);
l4.setBorder(edge);
l5.setBorder(edge);
l6.setBorder(edge);
l7.setBorder(edge);
l8.setBorder(edge);
l9.setBorder(edge);
l10.setBorder(edge);
l11.setBorder(edge);
l12.setBorder(edge);
l13.setBorder(edge);
l14.setBorder(edge);

t1.setBorder(edge1);
t2.setBorder(edge1);
t3.setBorder(edge1);
t4.setBorder(edge1);
t5.setBorder(edge1);
t6.setBorder(edge1);
t7.setBorder(edge1);
t8.setBorder(edge1);
t9.setBorder(edge1);
t10.setBorder(edge1);

setVisible(true);
setSize(1265,570);

setLocation(5,95);

connect();
regLoad();
}
public void regLoad()
{
    Connection con = null;
    Statement stm = null;
    ResultSet rs;
```



## COLLEGE MANAGEMENT SYSTEM



```
String qry,rg;
try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con = DriverManager.getConnection
("jdbc:odbc:adsn","system","123");
    stm = con.createStatement();
    qry = "select reg from studentdetail;";
    rs = stm.executeQuery(qry);

    while(rs.next())
    {
        rg = rs.getString("reg");

        cb1.addItem(rg);
    }
}
catch(Exception e1)
{
}
finally
{
    try
    {
        stm.close();
        con.close();
    }
    catch(SQLException e1)
    {
    }
}

}

public class newListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            t1.setText(" ");
            t2.setText(" ");
            t3.setText(" ");
            t4.setText(" ");
            t5.setText(" ");
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
t6.setText(" ");
t7.setText(" ");
t8.setText(" ");
t9.setText(" ");
t10.setText(" ");

cb1.setSelectedItem(0);
cb2.setSelectedIndex(0);
cb3.setSelectedIndex(0);
cb4.setSelectedIndex(0);

    }
    catch(Exception e1)
    {
    }
}

private class insertListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = (String) cb1.getSelectedItem();
            str2 = t1.getText();
            str3 = (String) cb2.getSelectedItem();
            str4 = (String) cb3.getSelectedItem();
            str5 = t2.getText();
            str6 = t3.getText();
            str7 = t4.getText();
            str8 = t5.getText();
            str9 = t6.getText();
            str10 = t7.getText();
            str11 = t8.getText();
            str12 = t9.getText();
            str13 = t10.getText();

            str14 = (String) cb4.getSelectedItem();

            stm.executeUpdate("insert into BbaResult
values('"+str1+"','"+str2+"','"+str3+"','"+str4+"','"+str5+"','"+str6+"','"+str7+"','"+str8+"
','"+str9+"','"+str10+"','"+str11+"','"+str12+"','"+str13+"','"+str14+"')");
```



## COLLEGE MANAGEMENT SYSTEM



```
stm.executeUpdate("commit");
JOptionPane.showMessageDialog(null,"Record
Inserted");

load();

} //end of try
catch(Exception sql)
{
JOptionPane.showMessageDialog(null,sql);
}
}

public class updateListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            str1 = (String) cb1.getSelectedItem();
            str2 = t1.getText();
            str3 = (String) cb2.getSelectedItem();
            str4 = (String) cb3.getSelectedItem();
            str5 = t2.getText();
            str6 = t3.getText();
            str7 = t4.getText();
            str8 = t5.getText();
            str9 = t6.getText();
            str10 = t7.getText();
            str11 = t8.getText();
            str12 = t9.getText();

            str13 = t10.getText();

            str14 = (String) cb4.getSelectedItem();

            stm.executeUpdate("update BbaResult set
name = '"+str2+"',course = '"+str3+"',semester = '"+str4+"',paper1 =
 '"+str5+"',paper2 = '"+str6+"',paper3 = '"+str7+"',paper4 = '"+str8+"',paper5 =
 '"+str9+"',paper6 = '"+str10+"',pract1 = '"+str11+"',pract2 = '"+str12+"',percentage =
 '"+str13+"',grade = '"+str14 +"' where reg = '"+str1+"'");
            stm.executeUpdate("commit");

JOptionPane.showMessageDialog(null,"Records Updated ");
```



## COLLEGE MANAGEMENT SYSTEM



```
load();

    }
    catch(SQLException sql)
    {
        JOptionPane.showMessageDialog(null,"Could not
Updated"+sql);
    }
}

public class viewListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("Enter
Registration number to be view");
            rs = stm.executeQuery("select * from BbaResult
where reg = '"+st+"'");
            while(rs.next())
            {

                cb1.setSelectedItem(rs.getString(1));
                t1.setText(rs.getString(2));

                cb2.setSelectedItem(rs.getString(3));
                cb3.setSelectedItem(rs.getString(4));
                t2.setText(rs.getString(5));

                t3.setText(rs.getString(6));
                t4.setText(rs.getString(7));

                t5.setText(rs.getString(8));
                t6.setText(rs.getString(9));

                t7.setText(rs.getString(10));
                t8.setText(rs.getString(11));

                t9.setText(rs.getString(12));
                t10.setText(rs.getString(13));
                cb4.setSelectedItem(rs.getString(14));

            }
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not found");
        }
    }
}

public class deleteListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            st = JOptionPane.showInputDialog("enter
Registration number to be deleted");
            stm.executeUpdate("delete from BbaResult where
reg = '"+st+"'");
            stm.executeUpdate("commit");
            JOptionPane.showMessageDialog(null,"Record is
deleted");
            load();
        }
        catch(SQLException sql)
        {
            JOptionPane.showMessageDialog(null,"Not
deleted");
        }
    }
}

protected void processWindowEvent(WindowEvent we)
{
    if(we.getID()==we.WINDOW_CLOSING)
    {
        r=JOptionPane.showConfirmDialog(null,"You want to
exit","Enter Carefully",JOptionPane.WARNING_MESSAGE);
        if(r == JOptionPane.YES_OPTION)
            dispose();
    }
}

public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```



## COLLEGE MANAGEMENT SYSTEM



```
        con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
        stm = con.createStatement();
    }
    catch(SQLException sql)
    {

JOptionPane.showMessageDialog(null,"Exception"+sql);
    }
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Not Connected");
    }
}
public class loadListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        load();
    }
}

public void load()
{
    int k;
    k=0;

    try
    {
        stm.executeUpdate("commit");
        rs = stm.executeQuery("select
reg,name,course,semester,paper1,paper2,paper3,paper4,paper5,paper6,pract1,pract
2,percentage,grade from BbaResult order by reg");

        while(rs.next())
        {
            str1 = rs.getString(1);
            str2 = rs.getString(2);
            str3 = rs.getString(3);
            str4 = rs.getString(4);
            str5 = rs.getString(5);
            str6 = rs.getString(6);
            str7 = rs.getString(7);
            str8 = rs.getString(8);
            str9 = rs.getString(9);
```





## COLLEGE MANAGEMENT SYSTEM



```
str10 = rs.getString(10);
str11 = rs.getString(11);
str12 = rs.getString(12);
str13 = rs.getString(13);
str14 = rs.getString(14);

data[k][0]=str1;
data[k][1]=str2;
data[k][2]=str3;
data[k][3]=str4;
data[k][4]=str5;
data[k][5]=str6;
data[k][6]=str7;
data[k][7]=str8;
data[k][8]=str9;
data[k][9]=str10;
data[k][10]=str11;
data[k][11]=str12;
data[k][12]=str13;
data[k][13]=str14;

k++;
}
data[k][0] = " ";
data[k][1] = " ";
data[k][2] = " ";

data[k][3] = " ";
data[k][4] = " ";
data[k][5] = " ";
data[k][6] = " ";
data[k][7] = " ";
data[k][8] = " ";
data[k][9] = " ";
data[k][10] = " ";
data[k][11] = " ";
data[k][12] = " ";
data[k][13] = " ";

JTable table = new JTable(data,colhead);
table.setFont(new Font("Times New Roman",Font.BOLD,16));
table.setForeground(Color.blue);
JScrollPane jsp = new JScrollPane(table);

jsp.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLB
AR_NEVER);
```



## COLLEGE MANAGEMENT SYSTEM



```
p3.add(jsp);
jsp.setBounds(12,300,910,160);
p3.add(jsp);
}
catch(SQLException sqle)
{
JOptionPane.showMessageDialog(null,"Record not transfer");
}
}
private class enterListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==t1)
            t2.requestFocus();
        if(e.getSource()==t2)
            t3.requestFocus();
        if(e.getSource()==t3)
            t4.requestFocus();
        if(e.getSource()==t4)
            t5.requestFocus();
        if(e.getSource()==t5)
            t6.requestFocus();
        if(e.getSource()==t6)
            t7.requestFocus();

        if(e.getSource()==t7)
            t8.requestFocus();
        if(e.getSource()==t8)
            t9.requestFocus();
        if(e.getSource() == t9)
        {
            double z1,z2,z3,z4,z5,z6,z7,z8,z9;

            z1 = Double.parseDouble(t2.getText());
            z2 = Double.parseDouble(t3.getText());
            z3 = Double.parseDouble(t4.getText());
            z4 = Double.parseDouble(t5.getText());
            z5 = Double.parseDouble(t6.getText());
            z6 = Double.parseDouble(t7.getText());
            z7 = Double.parseDouble(t8.getText());
            z8 = Double.parseDouble(t9.getText());

            z9 = (z1 + z2 + z3 + z4 + z5 + z6 + z7 + z8)/8;
```



## COLLEGE MANAGEMENT SYSTEM



```
t10.setText(String.valueOf(z9));

if(z9>=91 && z9 <=100)
    cb4.setSelectedIndex(1);

if(z9>=81 && z9 <=90)
    cb4.setSelectedIndex(2);

if(z9>=71 && z9 <=80)
    cb4.setSelectedIndex(3);

if(z9>=61 && z9 <=70)
    cb4.setSelectedIndex(4);

if(z9>=51 && z9 <=60)
    cb4.setSelectedIndex(5);

if(z9>=41 && z9 <=50)
    cb4.setSelectedIndex(6);

if(z9>=33 && z9 <=40)
    cb4.setSelectedIndex(7);

if(z9>=21 && z9 <=32)

    cb4.setSelectedIndex(8);

if(z9>=00 && z9 <=20)
    cb4.setSelectedIndex(9);
    }
}

private class timerListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        l17.setLocation(r,40);
        l16.setLocation(r,20);
        l15.setLocation(r,0);
        r = r+1;
        if(r>290)
            r = 0;
    }
}

private class timerListener1 implements ActionListener
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    public void actionPerformed(ActionEvent e)
    {
        title.setLocation(1,10);

        l = l - 1;
        if( l < 0)
            l = 1240;
    }
}
public class exitListener implements ActionListener
{
    public void actionPerformed(ActionEvent e)
    {
        dispose();
    }
}
}
```



## **/\* Affiliation Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Affiliation_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead = {"Affiliation no.", "Affiliated to", "Course", "Affiliation
date", "Registered no."};
    JPanel p1, p2, p3;
    JButton b1, b2;
    JLabel l1;
    JTextArea ta1;

    public Affiliation_Report()
    {
        super("REPORT");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");

        l1 = new JLabel("Report of Affiliation");
        l1.setFont(new Font("Monotype
Corsiva", Font.BOLD+Font.ITALIC, 27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from affiliation");
            while(rs.next())
                c++;
        }

        catch(Exception e)
        {

```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"message 1
"+e);
}
Object data[][] = new Object[c][5];
try
{
    ResultSet rs1;
    rs1 = stm.executeQuery("select
afflno,afflto,courses,to_char(affldt,'dd-mon-yyyy'),reg from affiliation");
    for(int i=0;i<c;i++)
    {
        rs1.next();
        data[i][0] = rs1.getString(1);
        data[i][1] = rs1.getString(2);
        data[i][2] = rs1.getString(3);
        data[i][3] = rs1.getString(4);
        data[i][4] = rs1.getString(5);
    }
}
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null,"message 2
"+sql);
}
Box box = Box.createHorizontalBox();
box.add(l1);
box.add(b1);
box.add(b2);
add(box,BorderLayout.NORTH);

ta1 = new JTextArea();
ta1.setLineWrap(true);
ta1.setWrapStyleWord(true);
ta1.setEditable(false);
ta1.setText("There are "+c+" Record Found");
getContentPane().add(ta1,BorderLayout.SOUTH);

JTable table = new JTable(data,colhead);
table.setAutoResizeMode(1);
JScrollPane resultspane = new JScrollPane(table);
table.setShowGrid(false);

table.setEnabled(false);
table.setBackground(new Color(250, 216, 217));
getContentPane().add(resultspane,BorderLayout.CENTER);
```



## COLLEGE MANAGEMENT SYSTEM



```
b1.addActionListener(new printButton(table));
b2.addActionListener(this);

pack();
setSize(1050,560);
setLocation(70,100);
setVisible(true);
}
public class printButton implements ActionListener
{
    JTable table1;
    public printButton(JTable table)
    {
        table1 = table;
    }
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            table1.print();
        }
        catch(Exception ex)
        {
        }
    }
}
public void actionPerformed(ActionEvent evt)
{
    if(evt.getSource() == b2)
    {
        dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con = DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,sqle);
        }
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex);
    }
}
```





## **/\* Staff Detail Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class StaffDetail_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead =
{"eno","ename","addr","city","state","pin","phone","dob","dept","bp","gender","e
mail","doj","mstatus","designation"};
    JPanel p1,p2,p3;
    JButton b1,b2;
    JLabel l1;
    JTextArea ta1;
    public StaffDetail_Report()
    {
        super("REPORT");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");

        l1 = new JLabel("Report of Staff Detail");
        l1.setFont(new Font("Monotype
Corsiva",Font.BOLD+Font.ITALIC,27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from staffdetail");
```



## COLLEGE MANAGEMENT SYSTEM



```
        while(rs.next())
            c++;
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"message 1
"+e);
    }
    Object data[][] = new Object[c][15];
    try
    {
        ResultSet rs1;
        rs1 = stm.executeQuery("select
eno,ename,address,city,state,pin,phone,to_char(dob,'dd-mon-
yyyy'),dept,bp,gender,email,to_char(doj,'dd-mon-yyyy'),mstatus,designation from
staffdetail");

        for(int i=0;i<c;i++)
        {
            rs1.next();
            data[i][0] = rs1.getString(1);
            data[i][1] = rs1.getString(2);
            data[i][2] = rs1.getString(3);
            data[i][3] = rs1.getString(4);
            data[i][4] = rs1.getString(5);
            data[i][5] = rs1.getString(6);
            data[i][6] = rs1.getString(7);
            data[i][7] = rs1.getString(8);
            data[i][8] = rs1.getString(9);
            data[i][9] = rs1.getString(10);
            data[i][10] = rs1.getString(11);
            data[i][11] = rs1.getString(12);
            data[i][12] = rs1.getString(13);
            data[i][13] = rs1.getString(14);
            data[i][14] = rs1.getString(15);
        }
    }

    catch(Exception sql)
    {
```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"message 2  
"+sql);  
}
```

```
Box box = Box.createHorizontalBox();
```

```
box.add(l1);  
box.add(b1);  
box.add(b2);  
add(box, BorderLayout.NORTH);
```

```
ta1 = new JTextArea();  
ta1.setLineWrap(true);  
ta1.setWrapStyleWord(true);  
ta1.setEditable(false);  
ta1.setText("There are "+c+" Record Found");  
getContentPane().add(ta1, BorderLayout.SOUTH);
```

```
JTable table = new JTable(data,colhead);  
table.setAutoResizeMode(1);  
JScrollPane resultspane = new JScrollPane(table);  
table.setShowGrid(false);  
table.setEnabled(false);  
table.setBackground(new Color(250, 216, 217));  
getContentPane().add(resultspane, BorderLayout.CENTER);
```

```
b1.addActionListener(new printButton(table));  
b2.addActionListener(this);
```

```
pack();  
setSize(1050,560);  
setLocation(70,100);  
setVisible(true);
```

```
}
```

```
public class printButton implements ActionListener
```

```
{
```

```
JTable table1;  
public printButton(JTable table)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    table1 = table;
}
public void actionPerformed(ActionEvent e)
{
    try
    {
        table1.print();
    }
    catch(Exception ex)
    {
    }
}
}
public void actionPerformed(ActionEvent evt)
{
    if(evt.getSource() == b2)
    {
        dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,sqle);
        }
    }
    catch(Exception ex)
    {
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,ex);  
    }  
}  
}
```



## **/\* Student Detail Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class StudentDetail_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead = {"Reg no.", "Roll", "Student name", "Father's
name", "Mother's name", "Date of birth", "Gender", "Enroll course", "Enroll
date", "Enroll session", "Temporary address ", "Permanent address", "Contact
no.", "Email", "Remarks"};
    JPanel p1,p2,p3;
    JButton b1,b2;
    JLabel l1;
    JTextArea ta1;
    public StudentDetail_Report()
    {
        super("REPORT");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");
        l1 = new JLabel("Report of Student Detail");
        l1.setFont(new Font("Monotype
Corsiva",Font.BOLD+Font.ITALIC,27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from studentdetail");

            while(rs.next())
                c++;
        }
        catch(Exception e)
        {

```



## COLLEGE MANAGEMENT SYSTEM



```
OptionPane.showMessageDialog(null,"message 1
"+e);
}
Object data[][] = new Object[c][15];
try
{
    ResultSet rs1;
    rs1 = stm.executeQuery("select
reg,roll,sname,fname,mname,to_char(dob,'dd-mon-
yyyy'),gender,ecourse,to_char(edate,'dd-mon-
yyyy'),esession,temp,parm,contact,email,remarks from studentdetail");
    for(int i=0;i<c;i++)
    {
        rs1.next();
        data[i][0] = rs1.getString(1);
        data[i][1] = rs1.getString(2);
        data[i][2] = rs1.getString(3);
        data[i][3] = rs1.getString(4);
        data[i][4] = rs1.getString(5);
        data[i][5] = rs1.getString(6);
        data[i][6] = rs1.getString(7);
        data[i][7] = rs1.getString(8);
        data[i][8] = rs1.getString(9);
        data[i][9] = rs1.getString(10);
        data[i][10] = rs1.getString(11);
        data[i][11] = rs1.getString(12);
        data[i][12] = rs1.getString(13);
        data[i][13] = rs1.getString(14);
        data[i][14] = rs1.getString(15);
    }
}
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null,"message 2
"+sql);
}
Box box = Box.createHorizontalBox();
box.add(l1);
box.add(b1);
box.add(b2);
add(box, BorderLayout.NORTH);

ta1 = new JTextArea();

ta1.setLineWrap(true);
ta1.setWrapStyleWord(true);
```



## COLLEGE MANAGEMENT SYSTEM



```
ta1.setEditable(false);
ta1.setText("There are "+c+" Record Found");
getContentPane().add(ta1, BorderLayout.SOUTH);

JTable table = new JTable(data,colhead);
table.setAutoResizeMode(1);
JScrollPane resultspane = new JScrollPane(table);
table.setShowGrid(false);
table.setEnabled(false);
table.setBackground(new Color(250, 216, 217));
getContentPane().add(resultspane, BorderLayout.CENTER);

b1.addActionListener(new printButton(table));
b2.addActionListener(this);

pack();
setSize(1050,560);
setLocation(70,100);
setVisible(true);
}
public class printButton implements ActionListener
{
    JTable table1;
    public printButton(JTable table)
    {
        table1 = table;
    }
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            table1.print();
        }
        catch(Exception ex)
        {
        }
    }
}
public void actionPerformed(ActionEvent evt)
{
    if(evt.getSource() == b2)
    {
        dispose();
    }
}
```





## COLLEGE MANAGEMENT SYSTEM



```
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,sqle);
        }
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex);
    }
}
```



## **/\* Student Fee Detail Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class StudentFeeDetail_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead = {"Reg no.", "Student name", "Course", "Course
fee", "Misc fee", "Other", "Total dues", "Deposit fee", "Deposit date", "Remarks"};
    JPanel p1,p2,p3;
    JButton b1,b2;
    JLabel l1;
    JTextArea ta1;

    public StudentFeeDetail_Report()
    {
        super("REPORT");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");

        l1 = new JLabel("Report of Student Fee Detail ");
        l1.setFont(new Font("Monotype
Corsiva",Font.BOLD+Font.ITALIC,27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from studentfeedetail");

            while(rs.next())
                c++;
        }

        catch(Exception e)
        {

```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"message 1
"+e);
}
Object data[][] = new Object[c][10];
try
{
    ResultSet rs1;
    rs1 = stm.executeQuery("select
reg,sname,course,cfee,mfee,other,tdue,deposite,to_char(ddate,'dd-mon-
yyyy'),remarks from studentfeedetail");
    for(int i=0;i<c;i++)
    {
        rs1.next();

        data[i][0] = rs1.getString(1);
        data[i][1] = rs1.getString(2);
        data[i][2] = rs1.getString(3);
        data[i][3] = rs1.getString(4);
        data[i][4] = rs1.getString(5);
        data[i][5] = rs1.getString(6);
        data[i][6] = rs1.getString(7);
        data[i][7] = rs1.getString(8);
        data[i][8] = rs1.getString(9);
        data[i][9] = rs1.getString(10);

    }
}
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null,"message 2
"+sql);
}
Box box = Box.createHorizontalBox();

box.add(l1);
box.add(b1);
box.add(b2);
add(box,BorderLayout.NORTH);

ta1 = new JTextArea();
ta1.setLineWrap(true);
ta1.setWrapStyleWord(true);

ta1.setEditable(false);
ta1.setText("There are "+c+" Record Found");
getContentPane().add(ta1,BorderLayout.SOUTH);
```



## COLLEGE MANAGEMENT SYSTEM



```
JTable table = new JTable(data,colhead);
table.setAutoResizeMode(1);
JScrollPane resultspane = new JScrollPane(table);
table.setShowGrid(false);
table.setEnabled(false);
table.setBackground(new Color(250, 216, 217));
getContentPane().add(resultspane,BorderLayout.CENTER);

b1.addActionListener(new printButton(table));
b2.addActionListener(this);

pack();
setSize(1050,560);
setLocation(70,100);
setVisible(true);
}
public class printButton implements ActionListener
{
    JTable table1;
    public printButton(JTable table)
    {
        table1 = table;
    }
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            table1.print();
        }
        catch(Exception ex)
        {
        }
    }
}
public void actionPerformed(ActionEvent evt)
{
    if(evt.getSource() == b2)
    {
        dispose();
    }
}
public void connect()
{
    try
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
        stm = con.createStatement();
    }
    catch(SQLException sqle)
    {
        JOptionPane.showMessageDialog(null,sqle);
    }
}
catch(Exception ex)
{
    JOptionPane.showMessageDialog(null,ex);
}
}
```



## **/\* Billing Master Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BillingMaster_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead = {"Reg no.", "Roll", "Student name", "Session", "Course
fee", "Exam fee", "Admin fee", "Lab fee", "Lib fee", "Misc fee", "Others", "Total"};
    JPanel p1, p2, p3;
    JButton b1, b2;
    JLabel l1;
    JTextArea ta1;

    public BillingMaster_Report()
    {
        super("REPORT");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");

        l1 = new JLabel("Report of Billing Master");
        l1.setFont(new Font("Monotype
Corsiva", Font.BOLD+Font.ITALIC, 27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from billingmaster");

            while(rs.next())
                c++;
        }

        catch(Exception e)
        {

```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"message 1
"+e);
}

Object data[][] = new Object[c][12];

try
{
    ResultSet rs1;

    rs1 = stm.executeQuery("select
stdreg,stdroll,stdname,sesion,coursefee,examfee,adminfee,labfee,libfee,mis
cfee,others,total from billingmaster");
    for(int i=0;i<c;i++)
    {
        rs1.next();

        data[i][0] = rs1.getString(1);
        data[i][1] = rs1.getString(2);
        data[i][2] = rs1.getString(3);
        data[i][3] = rs1.getString(4);
        data[i][4] = rs1.getString(5);
        data[i][5] = rs1.getString(6);
        data[i][6] = rs1.getString(7);
        data[i][7] = rs1.getString(8);
        data[i][8] = rs1.getString(9);
        data[i][9] = rs1.getString(10);
        data[i][10] = rs1.getString(11);
        data[i][11] = rs1.getString(12);
    }
}
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null,"message 2
"+sql);
}

Box box = Box.createHorizontalBox();

box.add(l1);
box.add(b1);

box.add(b2);
add(box, BorderLayout.NORTH);
```



## COLLEGE MANAGEMENT SYSTEM



```
ta1 = new JTextArea();

ta1.setLineWrap(true);
ta1.setWrapStyleWord(true);
ta1.setEditable(false);
ta1.setText("There are "+c+" Record Found");
getContentPane().add(ta1,BorderLayout.SOUTH);


JTable table = new JTable(data,colhead);
table.setAutoResizeMode(1);
JScrollPane resultspane = new JScrollPane(table);
table.setShowGrid(false);
table.setEnabled(false);
table.setBackground(new Color(250, 216, 217));
getContentPane().add(resultspane,BorderLayout.CENTER);

b1.addActionListener(new printButton(table));
b2.addActionListener(this);

pack();
setSize(1050,560);
setLocation(70,100);
setVisible(true);
}
public class printButton implements ActionListener
{
    JTable table1;
    public printButton(JTable table)
    {
        table1 = table;
    }

    public void actionPerformed(ActionEvent e)
    {
        try
        {
            table1.print();
        }
        catch(Exception ex)
        {
        }
    }
}
```





## COLLEGE MANAGEMENT SYSTEM



```
public void actionPerformed(ActionEvent evt)
{
    if(evt.getSource() == b2)
    {
        dispose();
    }
}

public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,sqle);
        }
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex);
    }
}
```



## **/\* PayRoll Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class PayRoll_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead = {"eno","ename","month","year","basic
pay","hra","da","ma","ta","pf","esi","advance","lic","other","net
salary","remarks"};
    JPanel p1,p2,p3;
    JButton b1,b2;
    JLabel l1;
    JTextArea ta1;

    public PayRoll_Report()
    {
        super("REPORT");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");

        l1 = new JLabel("Report of Payroll");
        l1.setFont(new Font("Monotype
Corsiva",Font.BOLD+Font.ITALIC,27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from payroll");
            while(rs.next())
                c++;
        }

        catch(Exception e)
        {

```



## COLLEGE MANAGEMENT SYSTEM



```
JOptionPane.showMessageDialog(null,"message 1
"+e);
}
Object data[][] = new Object[c][16];
try
{
    ResultSet rs1;
    rs1 = stm.executeQuery("select
eno,ename,month,year,bp,hra,da,ma,ta,pf,esi,advance,lic,other,nsalary,rem
arks from payroll");
    for(int i=0;i<c;i++)
    {
        rs1.next();

        data[i][0] = rs1.getString(1);
        data[i][1] = rs1.getString(2);
        data[i][2] = rs1.getString(3);
        data[i][3] = rs1.getString(4);
        data[i][4] = rs1.getString(5);
        data[i][5] = rs1.getString(6);
        data[i][6] = rs1.getString(7);
        data[i][7] = rs1.getString(8);
        data[i][8] = rs1.getString(9);
        data[i][9] = rs1.getString(10);
        data[i][10] = rs1.getString(11);
        data[i][11] = rs1.getString(12);
        data[i][12] = rs1.getString(13);
        data[i][13] = rs1.getString(14);
        data[i][14] = rs1.getString(15);
        data[i][15] = rs1.getString(16);
    }
}
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null,"message 2
"+sql);
}

Box box = Box.createHorizontalBox();

box.add(l1);
box.add(b1);

box.add(b2);
add(box, BorderLayout.NORTH);
```



## COLLEGE MANAGEMENT SYSTEM



```
ta1 = new JTextArea();

ta1.setLineWrap(true);
ta1.setWrapStyleWord(true);
ta1.setEditable(false);
ta1.setText("There are "+c+" Record Found");
getContentPane().add(ta1,BorderLayout.SOUTH);


JTable table = new JTable(data,colhead);
table.setAutoResizeMode(1);
JScrollPane resultspane = new JScrollPane(table);
table.setShowGrid(false);
table.setEnabled(false);
table.setBackground(new Color(250, 216, 217));
getContentPane().add(resultspane,BorderLayout.CENTER);

b1.addActionListener(new printButton(table));
b2.addActionListener(this);


pack();
setSize(1050,560);
setLocation(70,100);
setVisible(true);
}
public class printButton implements ActionListener
{
    JTable table1;
    public printButton(JTable table)
    {
        table1 = table;
    }

    public void actionPerformed(ActionEvent e)
    {
        try
        {
            table1.print();
        }
        catch(Exception ex)

        {
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
}
public void actionPerformed(ActionEvent evt)
{
    if(evt.getSource() == b2)
    {
        dispose();
    }
}
public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,sqle);
        }
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex);
    }
}
}
```



## **/\* BCA Result Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BcaResult_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead = {"Reg no.", "Student
name", "Course", "Semester", "Paper1", "Paper2", "Paper3", "Paper4", "Paper5", "Paper
6", "Practical1", "Practical2", "Percentage", "Grade"};
    JPanel p1,p2,p3;
    JButton b1,b2;
    JLabel l1;
    JTextArea ta1;

    public BcaResult_Report()
    {
        super("REPORT");

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");

        l1 = new JLabel("Report of BCA Result");
        l1.setFont(new Font("Monotype
Corsiva",Font.BOLD+Font.ITALIC,27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from bcaresult");
            while(rs.next())
                c++;
        }
        catch(Exception e)
```



## COLLEGE MANAGEMENT SYSTEM



```
{
    JOptionPane.showMessageDialog(null,"message 1
"+e);
}

Object data[][] = new Object[c][14];
try
{
    ResultSet rs1;

    rs1 = stm.executeQuery("select
reg,name,course,semester,paper1,paper2,paper3,paper4,paper5,paper6,pract1,pract
2,percentage,grade from bcaresult");
    for(int i=0;i<c;i++)
    {
        rs1.next();

        data[i][0] = rs1.getString(1);
        data[i][1] = rs1.getString(2);
        data[i][2] = rs1.getString(3);
        data[i][3] = rs1.getString(4);
        data[i][4] = rs1.getString(5);
        data[i][5] = rs1.getString(6);
        data[i][6] = rs1.getString(7);
        data[i][7] = rs1.getString(8);
        data[i][8] = rs1.getString(9);
        data[i][9] = rs1.getString(10);
        data[i][10] = rs1.getString(11);
        data[i][11] = rs1.getString(12);
        data[i][12] = rs1.getString(13);
        data[i][13] = rs1.getString(14);
    }
}
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null,"message 2
"+sql);
}
Box box = Box.createHorizontalBox();

box.add(l1);
box.add(b1);

box.add(b2);
add(box, BorderLayout.NORTH);
```



## COLLEGE MANAGEMENT SYSTEM



```
ta1 = new JTextArea();

ta1.setLineWrap(true);
ta1.setWrapStyleWord(true);
ta1.setEditable(false);
ta1.setText("There are "+c+" Record Found");
getContentPane().add(ta1, BorderLayout.SOUTH);

JTable table = new JTable(data,colhead);
table.setAutoResizeMode(1);
JScrollPane resultspane = new JScrollPane(table);
table.setShowGrid(false);
table.setEnabled(false);
table.setBackground(new Color(250, 216, 217));
getContentPane().add(resultspane, BorderLayout.CENTER);

b1.addActionListener(new printButton(table));
b2.addActionListener(this);

pack();
setSize(1050,560);
setLocation(70,100);
setVisible(true);
}
public class printButton implements ActionListener
{
    JTable table1;
    public printButton(JTable table)
    {
        table1 = table;
    }

    public void actionPerformed(ActionEvent e)
    {
        try
        {
            table1.print();
        }
        catch(Exception ex)
        {
        }
    }
}
```





## COLLEGE MANAGEMENT SYSTEM



```
public void actionPerformed(ActionEvent evt)
{
    if(evt.getSource() == b2)
    {
        dispose();
    }
}

public void connect()
{
    try
    {
        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
            stm = con.createStatement();
        }
        catch(SQLException sqle)
        {
            JOptionPane.showMessageDialog(null,sqle);
        }
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(null,ex);
    }
}
```



## **/\* BBA Result Report \*/**

```
package report;
import java.sql.*;
import java.net.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BbaResult_Report extends JFrame implements ActionListener
{
    private Container cn;
    Connection con;
    Statement stm;
    ResultSet rs;
    int c;
    String[] colhead = {"Reg no.", "Student
name", "Course", "Semester", "Paper1", "Paper2", "Paper3", "Paper4", "Paper5", "Paper
6", "Practical1", "Practical2", "Percentage", "Grade"};
    JPanel p1,p2,p3;
    JButton b1,b2;
    JLabel l1;
    JTextArea ta1;

    public BbaResult_Report()
    {
        super("REPORT");

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        b1 = new JButton("Print");
        b2 = new JButton("Close");

        l1 = new JLabel("Report of BBA Result");
        l1.setFont(new Font("Monotype
Corsiva",Font.BOLD+Font.ITALIC,27));
        l1.setForeground(Color.blue);

        c = 0;
        connect();
        try
        {
            rs = stm.executeQuery("select * from BbaResult");

            while(rs.next())

                c++;
        }
    }
}
```



## COLLEGE MANAGEMENT SYSTEM



```
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,"message 1
"+e);
}

Object data[][] = new Object[c][14];

try
{
    ResultSet rs1;

    rs1 = stm.executeQuery("select
reg,name,course,semester,paper1,paper2,paper3,paper4,paper5,paper6,pract1,pract
2,percentage,grade from BbaResult");
    for(int i=0;i<c;i++)
    {
        rs1.next();

        data[i][0] = rs1.getString(1);
        data[i][1] = rs1.getString(2);
        data[i][2] = rs1.getString(3);
        data[i][3] = rs1.getString(4);
        data[i][4] = rs1.getString(5);
        data[i][5] = rs1.getString(6);
        data[i][6] = rs1.getString(7);
        data[i][7] = rs1.getString(8);
        data[i][8] = rs1.getString(9);
        data[i][9] = rs1.getString(10);
        data[i][10] = rs1.getString(11);
        data[i][11] = rs1.getString(12);
        data[i][12] = rs1.getString(13);
        data[i][13] = rs1.getString(14);

    }
}
catch(Exception sql)
{
    JOptionPane.showMessageDialog(null,"message 2
"+sql);
}

Box box = Box.createHorizontalBox();

box.add(l1);
```



## COLLEGE MANAGEMENT SYSTEM



```
box.add(b1);
box.add(b2);
add(box, BorderLayout.NORTH);
```

```
ta1 = new JTextArea();
```

```
ta1.setLineWrap(true);
ta1.setWrapStyleWord(true);
ta1.setEditable(false);
ta1.setText("There are "+c+" Record Found");
getContentPane().add(ta1, BorderLayout.SOUTH);
```

```
JTable table = new JTable(data,colhead);
table.setAutoResizeMode(1);
JScrollPane resultspane = new JScrollPane(table);
table.setShowGrid(false);
table.setEnabled(false);
table.setBackground(new Color(250, 216, 217));
getContentPane().add(resultspane, BorderLayout.CENTER);
```

```
b1.addActionListener(new printButton(table));
b2.addActionListener(this);
```

```
pack();
setSize(1050,560);
setLocation(70,100);
setVisible(true);
}
```

```
public class printButton implements ActionListener
{
```

```
    JTable table1;
    public printButton(JTable table)
    {
        table1 = table;
    }
```

```
    public void actionPerformed(ActionEvent e)
    {
```

```
        try
        {
            table1.print();
        }
```



## COLLEGE MANAGEMENT SYSTEM



```
        catch(Exception ex)
        {
        }

    }

    public void actionPerformed(ActionEvent evt)
    {
        if(evt.getSource() == b2)
        {
            dispose();
        }
    }

    public void connect()
    {
        try
        {
            try
            {
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                con =
DriverManager.getConnection("jdbc:odbc:adsn","system","123");
                stm = con.createStatement();
            }
            catch(SQLException sqle)
            {
                JOptionPane.showMessageDialog(null,sqle);
            }
        }
        catch(Exception ex)
        {
            JOptionPane.showMessageDialog(null,ex);
        }
    }
}
```



# DOCUMENTATION



## **INTRODUCTION AND OBJECTIVES OF THE PROJECT**

My project " **COLLEGE MANAGEMENT SYSTEM**" has been created with the help of two-master minded programming language -"JAVA" and "oracle". The former is being in the foreground and the latter in the background respectively. The background language is favorite of many programmers because it offers the more wanted structure query facilities. Oracle provides high securities at the database are stored in the logical area called "db Engine". As db engine is created in the virtual memory and end user cannot access that directly it. With these thought in mind "COLLEGE MANAGEMENT SYSTEM" has been develop after brain storming the needs and requirements of the GIS.



## **OBJECTIVES OF PROJECT**

### **1. Capability:**

- a) Improved processing Speed:** - The COLLEGE MANAGEMENT SYSTEM used virtual memory which is supported by java JVM which keep the data intact in memory and in case of virtual memory it provide better management and provide fast processing also DLL to increase inherent speed to process data, graphic files.
- b) Increased volume:** - The COLLEGE MANAGEMENT SYSTEM provides many activity and events to avail the increased volume of processing, as data available in main memory and in the virtual memory to keep data intact, and depending on increased volume within processing the data kept in the virtual memory. Thus system avail the increased volume.
- c) Faster retrieval of information:** - The system provides faster retrieval of information as this is a complex task and the system supported the RDBMS engine of the oracle. The front end Java support in





this process by bringing the more and more data in main memory and in virtual memory, and swiftly interacting with the Oracle.

## 2. Control: -

**a) Greater accuracy and consistency:** It showing details, information, graphic require greater volume accuracy and consistency these best supported as the java has math coprocessor and multithreading capability, also the consistence improved exponentially by using Oracle as a back-end.

**b) Better security:** Safeguarding sensitive and important by making it accessible only to authorized persons. There are three level securities, one is data level, second is user level and third is administrator level.

## 3. Graphic user interface:

**a) Enhanced communication:** - The front-end of the project is Java and designed in interactive graphical interface using windows type interface. So, the information from the application to another is easily possible. Such as embedding and linking method of



## COLLEGE MANAGEMENT SYSTEM



windows .It is possible to embed a part of whole document in any specific module from another application.

**b) Integration:** - The project "College Management System" built in java platform, all information regarding of processing bring to a single area of virtual memory or in buffer for fast processing in remote site. The system provide common interface through applet which is a standardized tool for integrating.



# SYSTEM ANALYSIS

## 1. Identification of Need: -

To identify the need for software we use the Principles of Requirement Engineering. Requirement engineering provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirement as they are transformed into an operational system. The requirement

engineering process can be described in five distinct steps: -

- Requirement elicitation.
- Requirement analysis & negotiation.
- Requirement specification.
- System Modeling.
- Requirement validation.
- Requirement Management.

In other words we can say that requirement analysis is a software task that bridges the gap between system level requirement engineering and software design. Requirement



## COLLEGE MANAGEMENT SYSTEM



analysis allows the software engineering to refine the software allocation and build models of the data, functional and behavioral domains that will be treated by software. Requirement analysis provides the software designer with a representation of information, function and behavior that can be translated into data, architectural, interface and component level design; finally the requirement specification provides the developer and the customer with the means to assess quality once software is built.

The most commonly used requirement elicitation technique is to conduct a meeting or interview. The first meeting between a software engineer and customer can be likened to the awkwardness of a first date between to adolescents. Neither person knows what to say or ask, Both are worried that what they do say will be misinterpreted, both are thinking about where it might lead (Both likely have radically different expectation here) Both want to get the think over with, but at the same time, both want it to be a success.

Here according to this principle the analyst starts by asking context-free-questions. That is a set of question that will lead to a basic understanding of the problem, the people who want a solution, the nature of solution



## COLLEGE MANAGEMENT SYSTEM



that is desired, and the effectiveness of the first encounter itself. The first set of Context-free question focuses on the customer, the overall goals, and the benefits. For example, the analyst might ask: -

- Who is behind the request for this work?
- Who will use the solution?
- What will be economic benefit of a successful solution?
- Is there another source for the Solution that you need?

The next set of the questions enables the software engineer to gain a better Understanding of the problem and the customer to voice his or her perceptions about a solution: -

- How would you characterize "good" output that would be generated by a successful solution?
- What problem(s) will this solution address?
- Can you show me (or describe) the environment in which the solution will be used?
- Will special performance issues or constraints affect the way of the solution is approached?

The final set of question focuses on the effectiveness of the meeting:-



## COLLEGE MANAGEMENT SYSTEM



- Are you the right person to answer these questions?  
Are your answers? Official”?
- Are my questions relevant to the problem that you have?
- Am I asking too many questions?
- Can anyone else provide additional information?
- Should I be asking you anything else?

According to the above concepts I search on different websites of Geographical Information System and website of Indian government and state government for data management.

### **2. Preliminary Investigation: -**

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the System. The purpose of the preliminary investigation is to evaluate project requests. It is neither a design study nor does it includes the collection of details top describe the business system in all respect. Here for the “COLLEGE MANAGEMENT SYSTEM”, I have worked on the preliminary investigation that accomplished the following objectives: -

- Clarify and understand the project request.
- Determine the size of the project.
- Assess costs and benefits of alternative approaches.



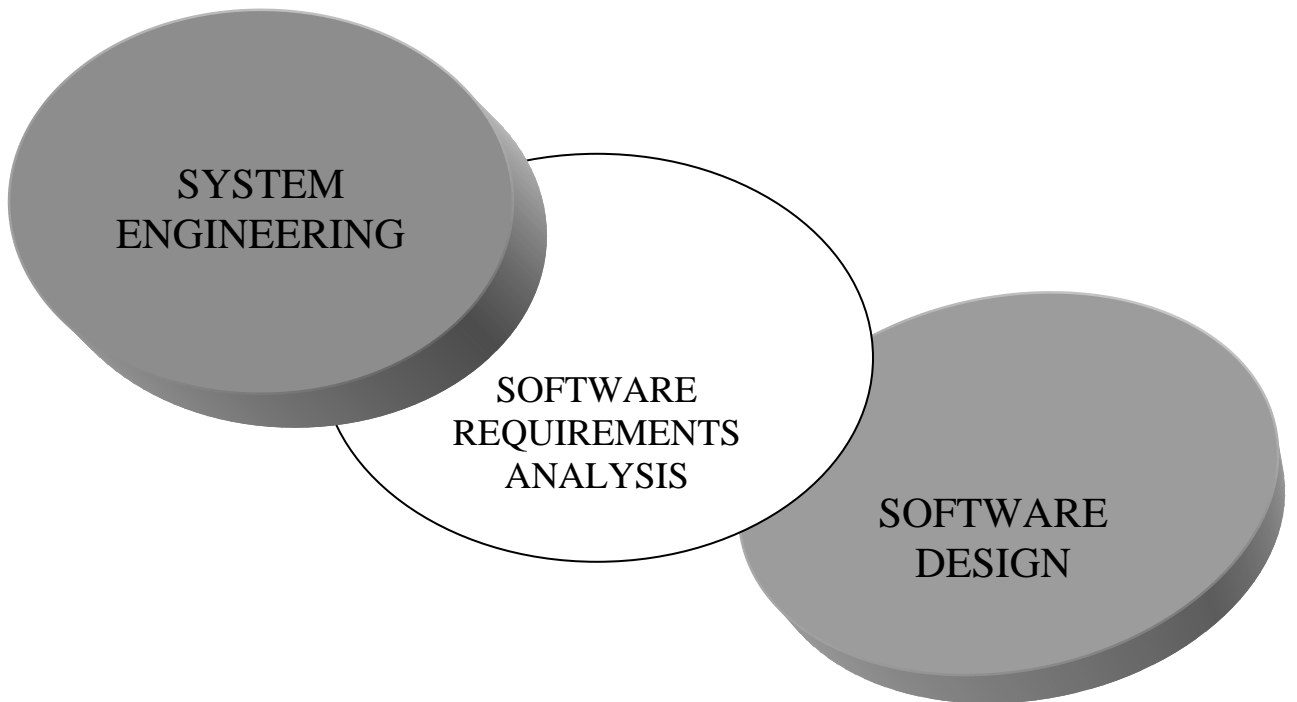
## COLLEGE MANAGEMENT SYSTEM



- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management with recommendation outlining the acceptances or rejection of the proposal.

Considering above criteria I also keep in mind that the requirements are clearly understandable when the clarification of project request is enquired. The data of the COLLEGE MANAGEMENT SYSTEM, which are collected by me during preliminary investigation, are: -

- Reviewing organization Documents
- Onsite observation and



**Figure-1: Analysis as a Bridge between System Engineering and Software Design**





## **FEASIBILITY STUDY**

Not everything imaginable is feasible, not even in software, evanescent as it may appear to outsiders. Feasibility is the determination of whether or not a project is worth doing. On the contrary, software feasibility has seven solid dimensions as below: Technical feasibility.

1. Operational feasibility.
2. Economic feasibility.
3. Social feasibility.
4. Management feasibility.
5. Legal feasibility.
6. Time feasibility.

The process followed in making this determination is called a feasibility study. This type of study determines when a project can or should be taken. Once it has been determined that the project is feasible,

And then analyst can go ahead and prepare the project specification, which finalizes project requirements. Generally, feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. The contents and recommendation



## COLLEGE MANAGEMENT SYSTEM



of such a study will be used as a sound basis for deciding whether to proceed, postpone or cancel the project. Thus, since the feasibility study may lead to the commitment of large resources, it becomes necessary that it should be conducted competently and that no fundamental errors of judgment are made.

In other words we can say that a feasibility study is conducted to select the best system that meets performance requirement. This contains an identification description, an evaluation of candidate system and the selection of best system for the job. The system required a statement of constraints; the identification of specific system objective and a description of outputs define performance.

### **1. Technical Feasibility: -**

Technical feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirements. My project is fit in technical feasibility because it requires neither too much heavy configuration system for its run nor in its development.

Following hardware and software components are used in the development of the proposed system.



## COLLEGE MANAGEMENT SYSTEM



### **HARDWARE REQUIRMENTS: -**

Processor	equivalent Intel dual core or above
Ram	256 MB
Hard disk drive	10 GB
Cache memory	256 KB
Monitor	VGA with 1024x768
Keyboard	104 keys
Mouse	Optical mouse

### **SOFTWARE REQUIREMENTS: -**

Operating system	WINDOWS 7/8/10
Back-end	oracle 10g
Front-end	Java

The same configuration system is also meets all the requirements to run or for implements to run or for implementation of the system. Though I have also tested this software for that system whose configuration is lower than the above specification.

### **2. Economic Feasibility: -**

Economic analysis is the most frequently used technique for evaluation of the effectiveness of a proposed system. More



## COLLEGE MANAGEMENT SYSTEM



commonly known as Cost/Benefit analysis, the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to

Design and implement the system. Other weight costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

Before establishing Economical Feasibility so as to justify the expenses that different organization working under this category has to bear for this Software. It is crystal clear that the user or Customer of this Software has no need to invest heavy amount for this software. As my project has been written in such a language, which is a platform Independent. It has also the feature of WORM. It means it possess the technology of Write Once & Read Many.

There is no any heavy expense on the maintenance of this Software. Also there is no need to expense a heavy Amount on the set of "Very Expert Personals" because it is very



## COLLEGE MANAGEMENT SYSTEM



easy and user-friendly software. It is fully tighten protection of **Secret Password** and **User Name** for unauthorized access of this Software.

For different types of Process or Operation, Reports or Query and Output also, there is no need of heavy expenditure.

### 3. Operational Feasibility: -

My project is also fit in Operational Feasibility Study. As I have explained before this about Technical Feasibility and Economical Feasibility, this Software is very easy and user-friendly.

I have used a very easy menu system and also used some control buttons in the easy way by which one can choose the options on his desire. Though this is fully protected with the security by word (Password and user name), if both are matched only then one can get access of this software. Various Reports, Forms and Queries can be generated on the fingertips for the user. In nutshell we can say that it has following operational features:

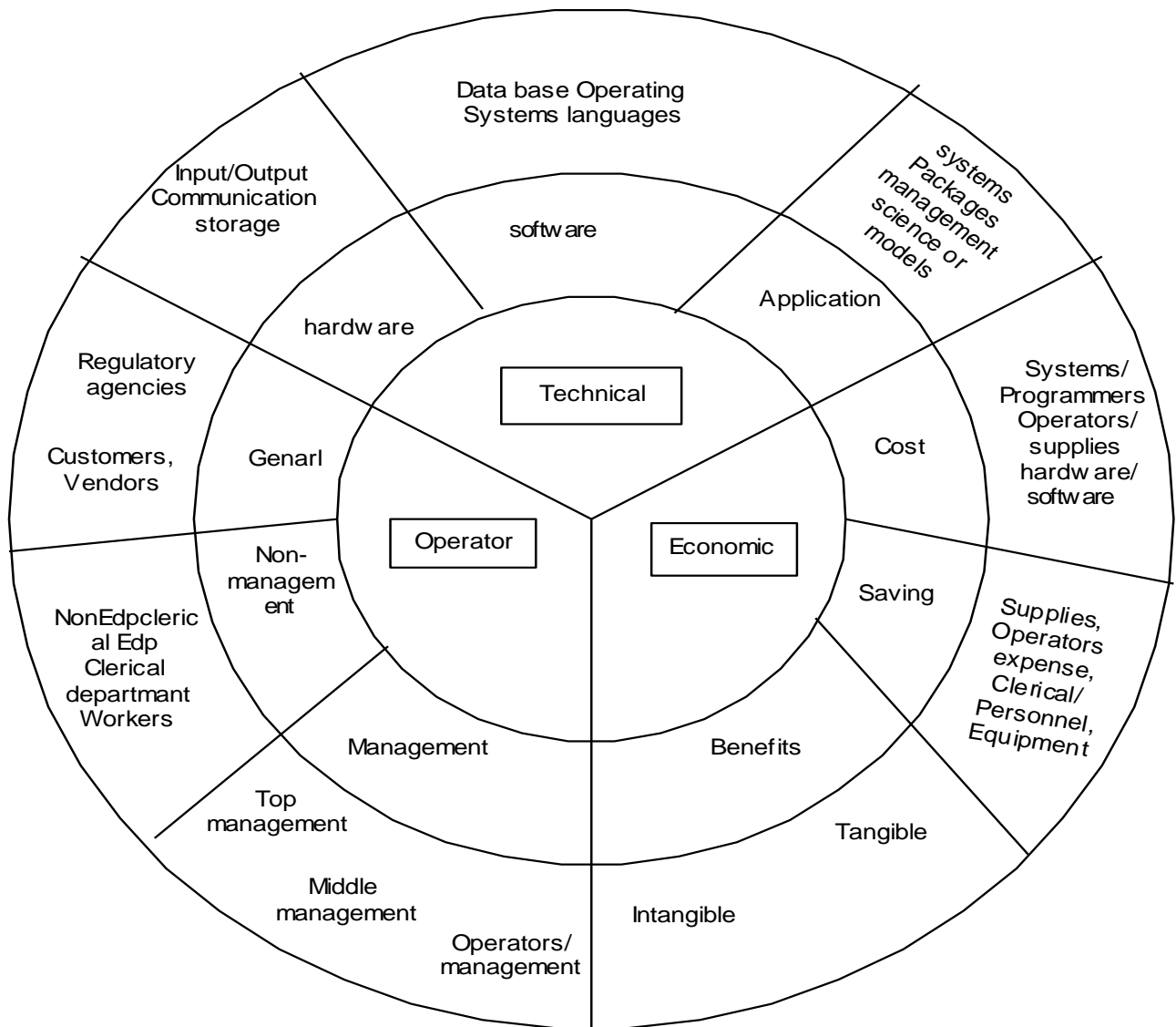
1. It is User-Friendly.
2. It is having less paperwork.
3. Efficient tractability.
4. Query can be generated.



## COLLEGE MANAGEMENT SYSTEM



5. Various Reports and Forms can be generated.
6. Fully protected by Password and User Name for unauthorized access.



**Figure 2: Application Feasibility Criteria**



## **SOFTWARE ENGINEERING PARADIGM APPLIED**

**Linear Sequential Model** has been used in carrying out this project work. The Linear Sequential Model is the oldest and the most widely used paradigm for Software Engineering. Linear Sequential Model is called sometimes the Classic Life Cycle or the Waterfall Model.

The Linear Sequential Model suggests a systematic, Sequential approach to software development that begins at the system level and progresses through Analysis, Design, Coding, Testing and Support.

Linear Sequential Model contains the following activities:

### **System / Information Engineering and Modeling: -**

Because software is always part of a larger system (or business), work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when software must interact with other elements such as hardware, people and database. System engineering and analysis encompass requirements gathering at the system level with a small amount of top-level design and analysis.





## COLLEGE MANAGEMENT SYSTEM



Information engineering encompasses requirements gathering at the strategic business level and at the business area level.

### **Software Requirements Analysis: -**

The requirements gathering process is intensified and focused specifically on software. To understand the nature of the program(s) to be built, the software engineer ("analyst") must understand the information domain for the software, as well as required function, behavior, performance, and interface. Requirements for both the system and the software are documented and reviewed with the customer.

### **Design: -**

Software design is actually a multi-step process that focuses on four distinct attributes of a program data structure, software architecture, interface representations, and procedural (algorithmic) detail. The design process translates requirements into a Representation of the software that can be accessed for quality before coding begins. Like requirements, the design is documented and becomes part of the software configuration.



## COLLEGE MANAGEMENT SYSTEM



### **Code Generation: -**

The design must be translated into a machine-readable form.

The code generation step performs this task. If design is performed in a detailed manner, code generation can be accomplished mechanistically.

### **Testing: -**

Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested and on the functional externals, that is, conducting tests to uncover errors and ensure that defined input will product actual results that agree with required results.

### **Support: -**

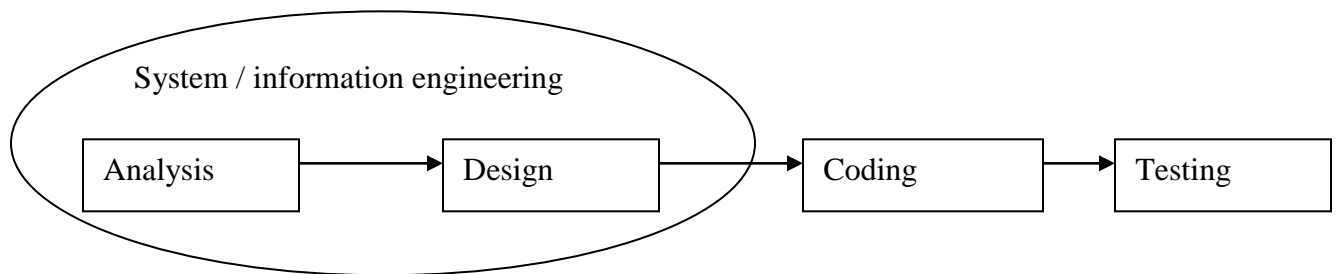
Software will undoubtedly undergo change after it is delivered to the customer (a possible exception is embedded software). Change will occur because errors have been encountered, because the software must be adapted to accommodate change in its external environment (e.g., a change required because of a new operating system or peripheral device), or because the customer requires



## COLLEGE MANAGEMENT SYSTEM



functional or performance enhancements. Software support/maintenance reapplies each of the preceding phases to an existing program rather than a new one.



**Figure 3: The Linear Sequential Model**

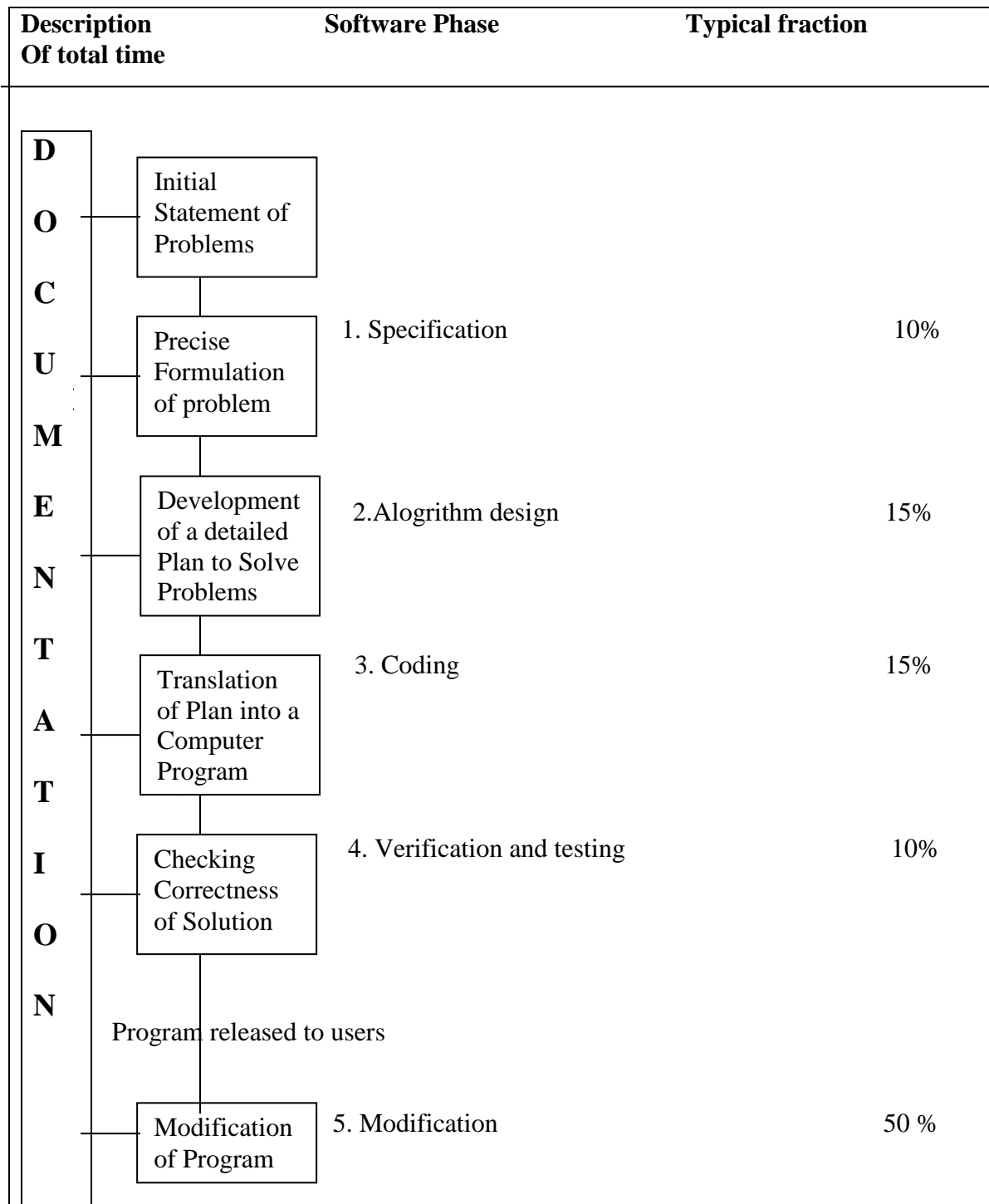
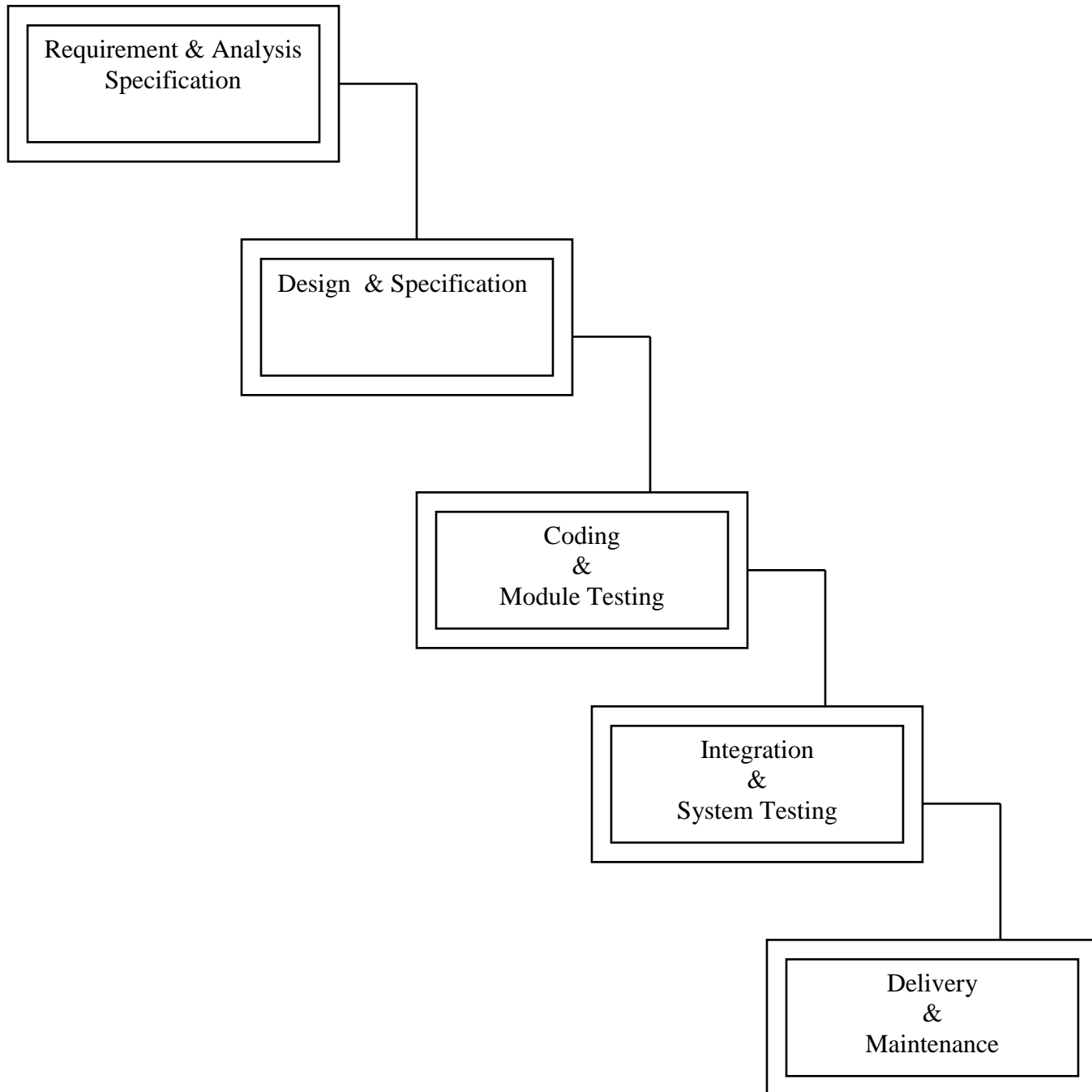


Figure 4: Software Development Life Cycle

**Figure 5: Waterfall Model of Software Life Cycle**





## **SOFTWARE & HARDWARE REQUIREMENT SPECIFICATION**

Following hardware and software components are used in development of proposed system: -

### **1. Hardware Requirements: -**

#### **Server:**

- ✓ Intel dual core or higher Processor
- ✓ SVGA Color Monitor
- ✓ At least 10 GB Hard Disc
- ✓ Minimum 256 MB RAM
- ✓ Mouse and Keyboard
- ✓ Printer
- ✓ Network Interface Card

### **2. Software Requirements: -**

#### **Server:**

- ✓ Window 7 / 8 / 10
- ✓ Oracle 10g Enterprise Edition



### **3. Description of Software & Hardware Tools: -**

#### **(A) . Database Management System (DBMS) : -**

A database management system (DBMS) consists of collection of interrelated data and a set of program to access those data. The collection of data, usually referred to as the database, contains information about one particular enterprise. The primary goal of DBMS is to provide an environment that is both convenient and efficient to use database information.

#### **Feature of Database Management System:**

DBMS permits the following facility/capability: -

1. Creating of file, addition of data, and deletion of data, creation, and addition and deletion of entire files.
2. Retrieving data collectively or selectively.
3. The data can be sorted or indexed at the user direction and discretion.
4. Various reports can be produced from the system. These may be either standardized report or that may be specifically generated according to specific user definition.



## COLLEGE MANAGEMENT SYSTEM



5. Mathematical functions can be performed and the data stored in the database can be manipulated with these functions to perform the desired calculations.

6. To maintain data integrity and database use.

### **(B). Relational Database Management System: -**

Relational Database Management System (RDBMS) have much more advantage over DBMS.

One of the main advantages of Relational Database Management System is that it is based on mathematical theory of relation. So, its database management system has the capability to recombine the data elements to form different relations thus giving great flexibility in the uses of data.

The data structure aspects of RDBMS deals with defining tables, its various attributes etc. Tables are nothing but the logical structure in relational system, not the physical structure, we can also say that tables represent an abstraction of the way the data is physically stored.

### **Capabilities of RDBMS: -**

1. Easy access of data
2. Flexibility in data modeling
3. Reduced data storage and redundancy
4. Independence of physical and logical data design.
5. A high-level data manipulation language (SQL)





## COLLEGE MANAGEMENT SYSTEM



6. It also frees the user from details of storage structure and access method.

### **(C).Oracle 10g Database Server: -**

Oracle is an object-relational database in simple way, we know, a relational database is nothing more than a collection of tables of data. An object relational database supports all the features of the relational database. It also supports object-oriented concepts and features.

In other words, an object-relational database management system (ORDBMS) extends the capability of the RDBMS to supports the object-oriented concept. Users can use ORACLE as an RDBMS or take advantage of its object-oriented features.

### **Oracle 10g Productivity and Utility: -**

ORACLE is a kernel package that has a number of tools that can be purchased separately and integrated the kernel and "add-ons". The tools allow the user to create database objects, like procedure, trigger function etc. some of tools are:

- **SQL\*PLUS**
- **PL/SQL**



## COLLEGE MANAGEMENT SYSTEM



The ORACLE RDBMS is a central ORACLE product. It includes the database manager and several tools intended to assist the users and DBA in the maintenance and use of data. The

core of RDBMS is the kernel. The kernel handles the following task:

- Manages storage and definition of data
- Controls and limits data access and concurring
- Allows backup and recovery of data
- Interprets SQL and PL/SQL

### **SQL \* PLUS: -**

SQL\*PLUS is the command line interface of the oracle database. It is a powerful oracle product that can take instruction for oracle, Check them for correctness, submit them to oracle and then modify or reformat the response oracle gives, based on orders or directions that you have set in place. It interacts with us, which means we can "talk" back. We can give directions, and it will follow them precisely.

It will tell us if it doesn't understand something we have told it to do.



As we get started, just think of SQL\*PLUS as a co-worker—an assistant follows our instructions and helps us do our work more quickly.

### **SQL: -**

SQL is a data sub-language. That is to say, the purpose of SQL is to interface to a relational database such as Oracle and all SQL statement are instructions to the database. In this it differs from general purposes programming language like C and BASIC.

The features of SQL are the following: -

- It provides sets of data as groups rather than as individual units.
- It provides automatic navigation to the data.
- It uses statements that are complex and powerful. The flow-control statements of most programming languages are absent in SQL
- They are provided in Oracle's extension to standard SQL called PL/SQL.
- SQL provides commands for a variety of tasks including:



## COLLEGE MANAGEMENT SYSTEM



- ✓ Querying data.
- ✓ Inserting, updating and deleting rows in a table.
- ✓ Creating, replacing altering and dropping objects.
- ✓ Controlling access to the database and its objects.
- ✓ Guaranteeing database consistency and integrity

### **PL/SQL: -**

PL/SQL stand for "procedural language extension to SQL"

PL/SQL available primly as an "enabling technology" within other softer products; it does not exit as a stand along Language. PL/SQL is powerful language, which incorporate many of most advance elements of procedural languages including:

- Explicit block structure.
- Conditional and sequential control statements.
- Loops of various kinds.
- Exception handlers for use in event-based error handling. Construct that help in the development of modular code - functions, procedures and Packages (collection of related program and variables); packages are often referred to as "Poor man's objects".



## COLLEGE MANAGEMENT SYSTEM



- User-defined data type such as object and nested table (with the oracle object option). PL/SQL is integrated tightly into oracle SQL language. We can execute SQL statement directly from our procedural program. Conversely, we can also call PL/SQL function from within a SQL statement.

We have use PL/SQL in the oracle relational database in the oracle server and in client-side application development tools 'such as oracle forms. PL/SQL is closely integrated into SQL language, yet it add programming construct that are not native to this standard relational data base language.

PL/SQL is a usual and an unusually powerful -Programming language. PL/SQL support packages, which allow us to perform object -Oriented design. PL/SQL provide a powerful mechanism for trapping and with exception handlers resolving errors. The tight integration of PL/SQL with SQL provides developer with the best of both worlds-declarative and procedural logic.

### **(D) . JAVA**

**JAVA** is an award-winning suite of Rapid Application

Development (RAD) tools for building enterprise-class applications that give end users a highly Graphics interface to the database.



It enables to: -

- Deploy applications in three-tier or two-tier configurations.
- Produce efficient, scalable applications.
- Increase the effectiveness of new and experienced developers.
- Build large, consistent, maintainable applications quickly.

### **TYPES OF APPLICATION ARCHITECTURE IN JAVA: -**

Applications are developed to support organizations in their business operations. The java supports three types of architecture.

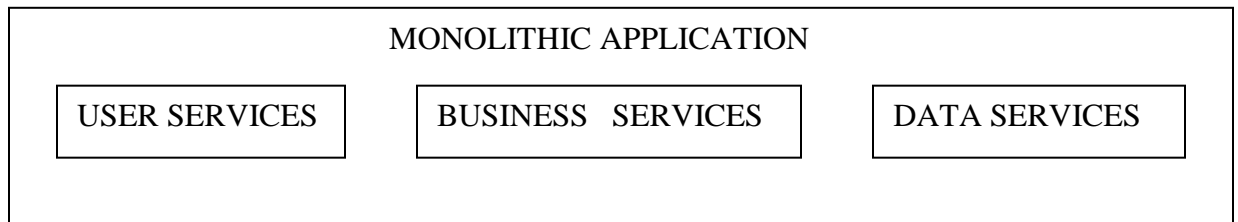
- (i) Single-tier architecture
- (ii) Two-tier architecture
- (iii) Three-tier architecture

#### **(i) Single-tier Architecture: -**

In single-tier architecture a single executable file handles all the functions relating to the user, business and data service layers. This type of application is called monolithic application.



## COLLEGE MANAGEMENT SYSTEM

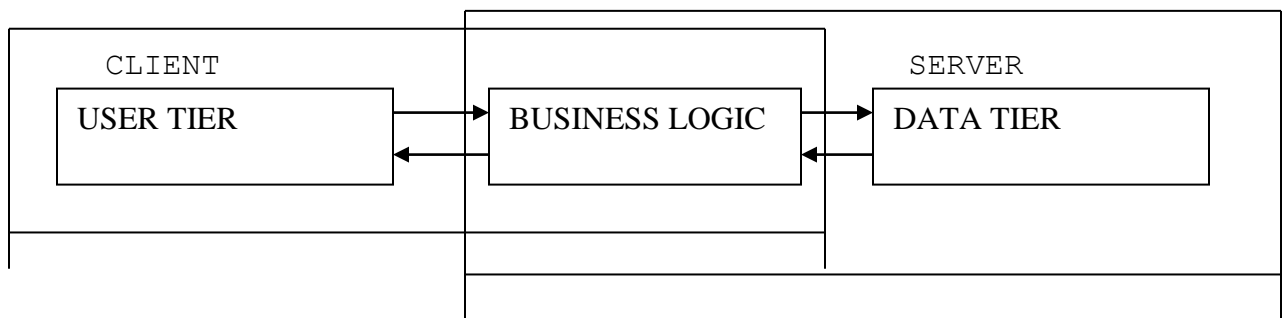


### (ii). Two-tier Architecture: -

The two-tier architecture divides an application into two components: -

(a) **Client:** Implements the users interface.

(b) **Server:** stores data.



### (iii) Three-tier Architecture: -

In three-tier architecture all the three layers reside separately, either on the same machine or on different machine. The user interface interacts with business logic.

The business logic validates the data sent by the

interfaces and forwards it to the database if it conforms to the requirements.



## COLLEGE MANAGEMENT SYSTEM



### FEATURES OF JAVA: -

There are following features introduced in JAVA

- (i).Inheritance
- (ii).Constructor and destructor
- (iii).Overloading
- (iv).Overriding
- (v).Structured exception handling
- (vi).Multithreading

### (i).Inheritance: -

Inheritance is the ability of a class to derive its characteristics from an existing class. java provides the "Inheritance" keyword to inherit the characteristics of a class "parent class" into another class derived class.





### **(ii).Constructor and destructor: -**

A constructor is a special type of method that invoked when we create a new instance of a class. A constructor is used to initialize the member of the class.

While the destructor is invoked when an object is destroyed. A destructor releases the resources used by an object.

### **(iii).Overloading: -**

Overloading enables methods to be defined with the same name but different parameters. Java provides a keyword "overloads" for this purpose.

### **(iv).Overriding: -**

Overriding is the capability of a derived class to overrides the characteristics of the parent class.java provides a keyword "override" for this purpose.

### **(v).Structures exception handling: -**

Exceptions are errors that generated at run time. Java support structure exception handling that consists of protected blocks of code and filter for the possible exceptions that can be raised by the program.

### **(vi). Multithreading: -**

Multithreading enables an application to contain one or more threads that can share the worked in an application by executing at a time.



## COLLEGE MANAGEMENT SYSTEM



### **(E) .LAN: -**

Local Area Networks, generally called LANs, are privately owned networks within a single building or campus of up to a few kilometers inside. They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information. LANs are distinguished from other kinds of networks by three characteristics:

- (i) Their size
- (ii) Their transmission technology and
- (iii) Their topology

Traditional LANs run at speeds of 10 to 100 Mbps, have low delay (tens of microseconds), and make very few errors. Newer LANs may operate at higher speeds, up to hundreds of Mbps.

### **(F) .UTP: -**

UTP stands for Unshielded Twisted Pair. Twisted pair cabling comes in several varieties, two of which are important for computer networks.

**Category 3** twisted pair consists of two insulated wires gently together. Four such pairs are typically grouped together in a plastic sheath for protection and to keep the eight wires together. This scheme allowed up to four regular



## COLLEGE MANAGEMENT SYSTEM



telephones or to multilane telephones in each office to connect the telephone company equipment in the closet.

**Category 5** twisted pair is similar to **category 3** twisted pair, but with more twists per centimeter and Teflon insulation, which results in less cross talk and a better quality signal over long distances. Both of these wiring types are often referred to as UTP.

**(G) . TCP/IP Protocol: -**

Transmission Control Protocol (TCP)/Internet Protocol (IP) is a set of protocols developed to allow computers of all sizes from different vendors, running different operating systems, to communicate or to share resources across the networks. This was designed to provide a reliable end-to-end byte stream over an unreliable inter networks. TCP is a connection-oriented protocol while IP is a connectionless protocol.



## **SYSTEM DESIGN**

The system design process is not a step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge and experience of the designer.

When designer starts working on system design, he will face different type of problems. Many of these will be due to constraints imposed by the user or limitation of the hardware and software available in the market. Sometimes, it is difficult to enumerate the complexity of the problems and solutions therefore since the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system-designing phase.

### **1. Design Objectives: -**

The primary objective of the design, of course, is to deliver the requirement as specified is the feasibility report. In general, the following design objectives should be kept in mind.



## COLLEGE MANAGEMENT SYSTEM



### **A. Practicality:**

The system must be stable and can be operated by people with average intelligence.

### **B. Efficiency:**

This involves accuracy, timeliness and comprehensiveness of the system output.

### **C. Cost:**

It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.

### **D. Flexibility:**

The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer system.

### **E. Security:**

This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse. In actually System Design involves first logical design and then physical construction of the



system. The logical design describes the structure and characteristics of features, like the outputs, inputs, files, databases and procedures. The physical construction, which follows the logical design, produces actual program software, files and a working system.

### **2. Design Constraints: -**

The designer normally will work under following constraints:

#### **A) Hardware:**

The existing hardware will obviously affect the system design.

#### **B) Software:**

The available software (Operating System, Utilities, Languages etc.) in the market will constrain the design.

#### **C) Budget:**

The budget allocated for the project will affect the scope and depth of design.

#### **D) Time-Scale:**

The new system may be required by a particular time (e.g. the start of a financial year). This may put a constraint on the designer to find the best design.



## COLLEGE MANAGEMENT SYSTEM



### **E) Interface with other Systems:**

The new system may require some data from another computerized system or may provide data to another system in which case the files must be compatible in the format and the system must operate with a certain processing cycle.

In nutshell we can say that system design develops the architectural details required to build a system of product the system design contains the following activities.

- Partition the analysis model into subsystems.
- Identify concurrency that is dictated by the problem.
- Allocate subsystems to processors and tasks.
- Develop a design for the user interface.
- Choose basic strategy for implementing data management.
- Identify global resources and the control mechanisms required to access them.
- Design an appropriate control mechanism for the system, including task management.
- Consider how boundary conditions should be handled.
- Review and consider trade-offs.



## COLLEGE MANAGEMENT SYSTEM



In actually the computer system design process is an exercise of specifying "How" the system will work. It is an iterative process, which is based on "What" the system will do.

Mainly, following five parts have been included in the system design process:

### **i) Output Design: -**

The starting point of the design process is the proper knowledge of system requirements, which will normally be converted in terms of output.

### **ii) Input Design: -**

Once the output requirements have been finalized, the next step is to find out what data need to be made available to the system to produce the desired outputs. The basic documents in which these data are available need to be identified. If necessary, these documents may have to be revised or new documents may have to be introduced.

### **iii) File Design: -**

Once the input data is captured in the system, these may have to be preserved either for a short or long period. These data will generally be stored in files in a logical





## COLLEGE MANAGEMENT SYSTEM



manner. The designer will have to devise the techniques of storing and retrieving data from these files.

### **iv) Procedure Design:**

This step involves specification of how processing will be performed in this, there are two aspects:

- Computer procedure.
- Non-computer procedure.
- The computer procedure will specify what Functions will be carried out on computer, what will be different programs and in what sequence the programs will be run. The non-computer procedure will specify the manual procedures for feeding input data, receiving outputs etc.

### **v) Control Design:**

The control design indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc. This will ensure that the system is functioning as per plan.

### **3. Input Design: -**

The design of input play very significant role in getting the correct output. It covers all phases of input from creation of initial data to actual entering the data to the



## COLLEGE MANAGEMENT SYSTEM



system for processing. The input design is the link that ties the information system into the world of its users. Some feature of design may vary depending on whether the system is batch oriented or online.

The various objectives of input design which focus on:

- Controlling amount of input.
- Avoiding delay.
- Avoiding errors in data.
- Avoiding extra steps.
- Keeping the process simple.

### **Major Concerns Regarding Input:**

We must consider on some important points regarding input, they are: -

- What input is needed?
- How and where is the input created?
- How should the source documents be designed?
- What format should be used for the Input records?
- What medium should be used for recording the input?

### **4. Output Design: -**

Presenting the data processed by a computer-based information system in an attractive and usable form has



## COLLEGE MANAGEMENT SYSTEM



become very essential. Success and acceptance of a system to some extent depend on good presentation. Therefore, system analysts must know fully how to design output report in an attractive way. Many new output devices are being introduced in the market because of recent development in computer technology.

There are three main reasons why outputs from the computer are required.

1. For communicating to the persons concerned.
2. For re-input to the computer for being connected with other data and further processing.
3. For permanent storage.

### **Output Design Consideration:**

An output to be produced usually depends upon the following consideration: -

#### **(a) Type of User and Purpose:**

Generally different levels of users will have different requirements from the system. Some want exception reports (e.g. when transaction fall below a certain level), some want summary reports (e.g. transaction quantity and value for each region) while some want details.



## COLLEGE MANAGEMENT SYSTEM



Again statutory reports will normally be as per requirement specified under the law and the designer will not have much flexibility to change the format.

### **(b) Content:**

The data that are needed to be included in the output. These will be related to the purpose of the input.

### **(c) Format:**

This refers to the arrangement of data on the report, size of the paper, titles, headlines, color of the paper etc.

### **(d) Frequency and Timing:**

At what frequency (monthly, annually etc.) and when (after annual closing of transaction, after the end of the financial year, before the last day of every month etc.)

### **(e) Volume:**

Often sheer volume of the output deters one from using the outputs. The sheer bulk of the report may also create problems for handling, filing or printing time.

### **(f) Sequence:**

The usefulness of an output very often depends on the sequence of data printed. A proper sequence will also help distribution of outputs to different users (e.g. pay-slips printed department wise facilitates easier payment).



### **(g) Quality:**

This relates to the content, appearance and accuracy of the output generated for external users should be given special attention in respect of its get-up, quality of paper etc.

### **(h) Type of Stationary:**

Reports can be generated on ordinary blank stationery or on specially printed stationary that is useful when most of the contents of the input are constant. This type of stationary has the following advantages: -

- Save Computer time.
- Attractive appearance convenient to use by the users.

### **5. Form Design: -**

As we know that data provide the basis for information systems. Without data there is no system, but data must be fed in correct way so that the information produced must be in a format acceptable to the user. In either case, it is still data- the basic element of a printed form.

A form is a tool with a message. It is the physical carrier of data or information. It also can constitute authority for action. It is hard to imagine a business operating without using forms. They are the vehicles for most



## COLLEGE MANAGEMENT SYSTEM



communications and the blue print for many activities. It is as important as a printed form is, however, the majority of forms are designed by poorly trained people who are puzzled by confusing forms, they ask for directions on how to read them and how to fill them out. When a form is poorly designed, it is a poor (and costly) administrative tool.

### **Classification of Forms:**

A printed form is generally classified by what it does in the system. There are three primary classifications: Entry, Query, and Report forms.

### **Entry Forms:**

The user has designed entry Forms for Data Entry work.

### **Query & Report Forms:**

A report form guides supervisors and other administrators in their activities. It provides data on a project or a job.

### **Factors to be considered in Form Design:**

Form design plays an important role in data processing. Form must have the appearance of a well-conceived and attractive design. Some of the important, factors, which should be taken care of are given below.



## COLLEGE MANAGEMENT SYSTEM



- a) Size and shape of the form should be such that it is convenient for handling, filling, sorting etc.
- b) Arrange the material in logical orders so that it becomes easy to fill it up.
- c) The form title must clearly identify its purpose.
- d) Precise contents should be recorded. Adequate and compact space should be provided for items to be recorded. Pre-printed entries should be taken care of.
- e) Special features like security and control are to be considered.
- f) Introduce emphasis by shading columns, heavy lines, etc. If the form is to be used for specific clerical operation, for example copying or checking. See that the detail is arranged and spaced to provide maximum help to the operation.
- g) The form designer should design the form in such a way so as to cover the specific needs of the purpose for which it is designed.

### **6. File Design: -**

After designing the Input and Output, the designer begins to pay his attention on the work of file designing or how data



## COLLEGE MANAGEMENT SYSTEM



should be organized around user requirements. How data are organized depends on the data and response

requirements that determine hardware configurations. System analyst is responsible for designing the files and selecting their contents, selecting from options available for organizing the data. File organization may be sequential, index sequential, inverted list or random. Each method has its own uses and abuses.

An integral approach to file design is the database. The general theme is to handle information as an integrated whole, with a minimum of redundancy and improved performance. Various software techniques are applied to manipulate, describe and manage data. Irrespective of type of data structure used, the main objectives of database are accurate and integrity, successful recovery from failure, privacy and security of data.

Files are the heart of a computer application. The basic terms used to describe file are Data item, Record, File and Database.





## COLLEGE MANAGEMENT SYSTEM



### **Data Item:**

A basic or individual element of data is called data item. Each Data item is assigned a value. Data item is sometimes referred to as a field.

### **Record: -**

The collection of related data items is called a record. The analyst also determines the length and type of each field.

### **File: -**

File is a collection of related records. Each record in a file is included because it pertains to the same entity.

### **Database: -**

The highest level in the hierarchy is the database. It is a set of inter-related files for real time processing. It contains the necessary data for problem solving and can be used for several users who are accessing data concurrently.

### **Types of Files: -**

There are various types of files in which the records are collected and maintained they are categorized as:

- Master file.
- Survey file.



## COLLEGE MANAGEMENT SYSTEM



- Table file.
- Report file.
- Back-up file.
- Dump file
- Library file.

### **Master File: -**

Master files are the most important type of file. Most file design activities concentrate here. In a business application, these are considered to be very significant, they contain the essential records for maintenance of the organizational business.

### **Survey File: -**

A Survey file is a temporary file used for two purposes. First of all, it is used to accumulate data about events as they occur. Secondly, it helps in updating master files to reflect the result of current Survey. The term Survey refers to any business event that affect the organization and about which data is captured.

### **Table File: -**

A special type of master file is included in many systems to meet specific requirements. Where data must be referenced repeatedly. Table files are permanent files containing



## COLLEGE MANAGEMENT SYSTEM



reference data used in processing transaction, updating master file of producing output. As memory space and make the program maintenance easier by storing data in a file, that otherwise would be included in programs or master file records.

### **Report File: -**

Report files are collected contents of individual output reports of documents produced by the system. They are created by the system where many reports are produced by the system but printer may not be available for all the reports. This situation frequently arises when the computer carry out three functions- Input, Processing and Output simultaneously, rather than executing each function in sequence. In this case, the computer writes the report contents to a file on the disk or secondary storage device, where it remains until it can be printed. That file is called the report file, which contains the unprinted output data.

The process of creating it is known as spooling, which means that output that cannot be printed when it is produced is spooled into a report file. Then, depending on the availability of printer, the system will be instructed to read the report file and print the output on the printer.



## COLLEGE MANAGEMENT SYSTEM



### **Backup File: -**

It is a copy of master, transaction or table file that is made to ensure a copy is available if anything happens wrong to the original.

### **Dump File: -**

This is a copy of computer-held data at a particular point of time. This may be copy of master file to be retained to help recovery in the event of a possible corruption of the master file or it may be part of a program in which error is being traced.

### **Library File: -**

Library file generally contains application programs, utility programs and system software packages.



## CODE EFFICIENCY

Every step has been considered in order to achieve the code efficiency. **"Java"** has been used in order to achieve the high degree of optimality of code. It is an object oriented GUI tool that provides the facility of the event driven programming as well as supports object oriented programming (oops). The event driven and oops concept reduce the code. SQL meets only the target (what has to be done) to be specified and it does not need the steps (how) to be followed to achieve the target. The data retrieved is done by SQL engine of the RDBMS, which is very efficient, and it increases the efficiency of the application as well. As far as user interface design of the application is concerned, object oriented paradigm has been applied which reduces the lengthy codification and achieve the inheritance, which further enhance the code efficiency.



## **CODE OPTIMAZATION**

The application has been designed using two-tier client/server technology in which the functions of the application segregated between the two tiers i.e. the Back End and the Front End. Every type of data security, constraints and validation rules has been applied.

At the Back End Database itself, which is as per the code's rules, due to which less codification is needed at the Front End user interface level? Due to this approach, it is possible to optimize the codes so that it can take less memory to execute and makes execution fast.

Again PL/SQL programming language has been used which provides the facility of embedded SQL. The data retrieval power of SQL is well known. Since PL/SQL is the product of Oracle itself, which runs faster than any other, programming language on Oracle RDBMS platform. Thus processing power of PL/SQL is unmatched. The combination of PL/SQL and SQL is excellent as far as processing time and memory is concerned. Thus it reduces the requirement of storage as well as processor and the resultant code optimization or efficiency is very high.



# **VALIDATION CHECK**

As discussed earlier the application has been broken in two tiers like Back End Oracle RDBMS and the Front End user interface in Java. In this architecture every type of validation check and constraints have been applied at the database level itself as per the code rules. Apart from this every type of data security rules have also been applied by creating appropriate synonyms and user roles. Access to the user has been provided in controlled way keeping the view at user's level and their requirements. Due to which user access is restricted and the security has enforced.

Another level of validation checks has been applied at data entry form level also in the user interface. These two levels of validation checks enforce high level of data validity and only valid data can be entered and processed. The validation rules cannot be bypassed by any means.



# **SOFTWARE TESTING**

## **1. Testing Objectives: -**

- (i) Testing is a process of executing a program with the intent of finding an error.
- (ii) A good test case is one that has a high probability of finding an undiscovered error.
- (iii) A successful test is one that uncovers as yet undiscovered error.

## **2. Testing Techniques: -**

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to our customer. Our goal is to design a series of test cases that have a high likelihood of finding errors.

Software is tested from two different perspectives: -

- (a) Internal program logic is exercised using "White Box" test case design techniques.
- (b) Software requirements are exercised using "Black 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.





### **(a).White Box Testing:**

White box testing sometimes called Glass box testing, is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

1. Guarantee that all independent paths within a module have been exercised at least once.
2. Exercise all logical decisions on their true and false sides.
3. Exercise internal data structures to ensure their validity.

A reasonable question might be posed at this juncture. "Why spend time and energy worrying about (and testing) logical minutiae when we might better expend effort ensuring that program requirements have been met?" State another way, why don't we spend all of our energy on black box tests?

Logic errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed. Errors tend to creep into our work when we design and implement function, conditions, or control that is out of the mainstream.



We often believe that a logical path is not likely to be executed when, in fact, it may be executed on a regular basis.

Typographical errors are random. When a program is translated into programming language source code, it is likely that some typing errors will occur. Many will be uncovered by syntax and type checking mechanisms, but others may go undetected until testing begins. White-Box Testing is far more likely to uncover them.

### **(b). Black- Box Testing:-**

Black Box testing, also called Behavioral Testing, focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods.

Black Box testing attempts to find errors in the following categories: -

1. Incorrect or missing functions.
2. Interface errors.



## COLLEGE MANAGEMENT SYSTEM



3. Errors in data structures or external data base access.
4. Behavior or performance errors.
5. Initialization and termination errors.

Unlike White-Box testing, which is performed early in the testing process, black box testing tends to be applied during later stages of testing. Because black box testing purposely disregards control structure, attention is focused on the information domain. Tests are designed to answer the following questions: -

1. How is functional validity tested?
2. How is system behavior and performance tested?
3. What classes of input will make good test cases?
4. Is the system particularly sensitive to certain input values?
5. How are the boundaries of a data class isolated?
6. What data rates and data volume can the system tolerate?
7. What effect will specific combinations of data have on system operation?

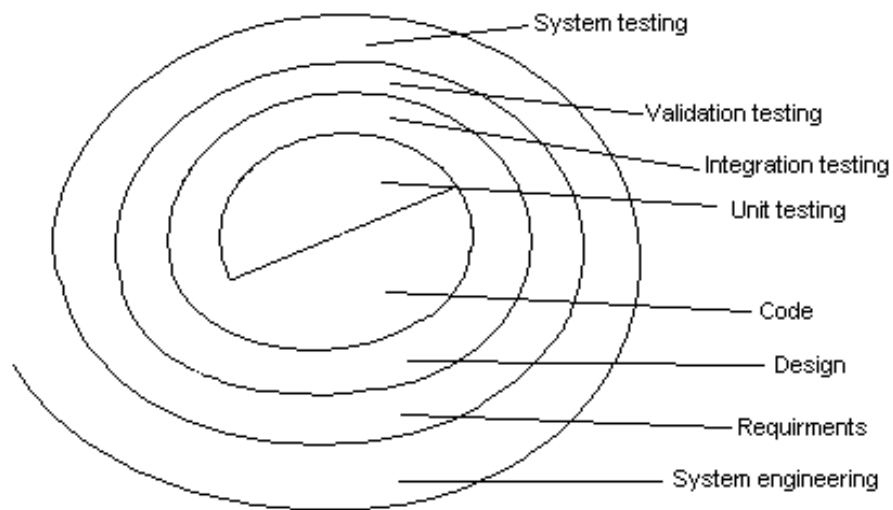
### **3. Testing Strategies: -**

System engineering defines the role of software and leads to software requirements analysis where the information domain, function, behavior, performance, constraints, and validation



criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral inward along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral (shown in figure): -



**FIGURE: -TESTING STRATEGY**

### **1. Unit Testing: -**

Unit testing begins at the vertex of the spiral and concentrates on each unit (i.e. component) of the software as implemented in source code. It focuses verification effort on the smallest unit of software design - the



## COLLEGE MANAGEMENT SYSTEM



software component or module. The unit test is white-box oriented, and the step can be conducted in parallel for multiple components.

### **Unit Test Considerations:**

- Interface
- Local data structures
- Boundary conditions
- Independent paths
- Error handling paths

### **2. Integration Testing: -**

In this testing strategy its focus is on design and construction of the software architecture. Integration testing is a systematic technique for construction of program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

### **Components of Integration Testing: -**

#### **(a) Top-down Integration: -**

Top-down integration testing is an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control (main program). Modules subordinate



## COLLEGE MANAGEMENT SYSTEM



(and ultimately subordinate) to the main control module are incorporated into the structure in either a depth-first or breadth-first manner.

### **b) Bottom-up Integration: -**

Bottom-up integration testing, as its name implies, begins construction and testing with atomic modules (i.e. components at the lowest levels in the program structure). Because components are integrated from the bottom up, processing required for components subordinate to a given level is always available and the need for stubs is eliminated.

### **c) Regression Testing: -**

It is the re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects.

### **d) Smoke Testing: -**

Smoke testing is an integration testing approach that is commonly used when "shrink-wrapped" software products are being developed. It is designed as a pacing mechanism for time-critical projects, allowing the software team to assess its project on a frequent basis.



#### **4. Validation Testing: -**

Validation can be defined in many ways but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer.

**a) Validation Test Criteria:** Two possible Conditions exist:

-

- 1) The function or performance characteristics conform to specification and are accepted
- 2) A deviation from specification is uncovered and a deficiency list is created.

It is often necessary to negotiate with the customer to establish a method for resolving deficiencies.

#### **d) Configuration Review: -**

An important element of the validation process is a configuration review. The configuration review, sometimes called an audit.

#### **c) Alpha Testing: -**

A customer conducts the alpha test at the developer's site. Alpha tests are conducted in controlled environments.

#### **d) Beta Testing: -**

The beta test is conducted at one or more customer sites by the end-user of the Software. The beta test is a "live"



## COLLEGE MANAGEMENT SYSTEM



application of the software in an environment that cannot be controlled by the developer.

### **4. System Testing: -**

In System Testing the software and other elements are tested as a whole. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify the system elements have been properly integrated and perform allocated functions.

To test computer software, we spiral out along streamlines that broaden the scope of testing with each turn.





## **IMPLEMENTATION AND MAINTENANCE**

The study has been done to know the working of "**COLLEGE MANAGEMENT SYSTEM**" and the sample data collected. The executive bodies as well as general body members have been consulted to get the sample test data. Apart from this some of beneficiary and donators have been interviewed to get the required information. I have visited some of the colleges to collect the sample data. The data, thus, collected is entered into the system and processed. The processed information in the form of MIS Report is provided to the members of executive body. Proper modification has been done in the application as per the modification in the report processed and the suggestion of members. The process has been repeated several times till the members of executive body are fully satisfied. The test case or test data has been shown in the forms of MIS Report. It can be seen in the annexure enclosed.

First of all the application has been thoroughly tested by applying the test data. After that it got approved from the executive body to be implemented. User's operational manual has been prepared and proper training to the users has been



## COLLEGE MANAGEMENT SYSTEM



provided. Once the user is well acquainted with the new computerized system it was installed and operationalized along with the existing manual system. Both the computerized system and the manual system will run side by side for at least six months. During this period if any nonconformity will be found out, that will be fully solved and incorporated. If the new system does not contradict the existing system can be fully adopted.

After system task will not over a close eye site will be kept on the functioning of the new system. If any type of problem will find out that will be properly eradicated. This type of software maintenance will be done or not less than one hour after complete operationalization of the system. During this period any type of maintenance needed or required by the management will be done. The organization will depute a technical person to manage the new system and to operate smoothly.



## **SYSTEM SECURITY MEASURES**

There are several aspects of the system security: -

1) **Hardware Security:** -

Only limited and authenticated person will be allowed to access the system. The system administrator will take care of all these things.

2) **Software Security:** -

Two level of software security can be provided, which is the operating system level and the database level. Every user will be provided unique user-id and confidential password in order to log into the system. WINDOWS 10 operating system has been adopted in the proposed system, which is highly secured and does not allow unauthorized users. Again it restricts the function of users.

By granting them proper access control mechanism. Similarly the Oracle database server can provide an additional level of software or user security also so that unauthorized user can't access the database. The authorized user can access only that part of the database for which they are granted access privileges

3) **Application Level Security:** -

User authentication and validation have also been defined additionally in the application level. The users have been



## COLLEGE MANAGEMENT SYSTEM



provided user name and password facility. The users have been categorized into administrator officer assistant and operator. The role of each type of user has been defined and fixed and user can't violate these security rules.

#### 4) **Data Level Security:** -

Data validation rules and constraints have been properly defined both at the database level and the application level. Due to this only valid data can be entered. Apart from this, background audit mechanisms have been properly defined so as to track all the activities related to data management. For example, data insertion is recorded along with the user name and date & time of insert. Similarly data updation and deletion are also recorded with the user name and date & time in effect. A separate copy of the original data will be stored in a separate audit table during its updation & deletion. This is done along with the recording of user performing the updation & deletion and date & time. From this audit table the original data can be recorded as and when required and user can be made responsible. The application also keeps tracks of all the users' activities. Hence the data itself is completely secured.



## **COST ESTIMATION OF THE PROJECT**

Software cost and effort estimation will never been exact science. Too many variables-Human, Technical, Environmental, and Political can affect the ultimate cost of software and effort applied to develop it. However, software project estimation can be transformed from a black art to a series of systematic steps that provide estimates with acceptable risk.

To achieve reliable cost and effort estimates, a number of options arise: -

- (1) Delay estimation until late in the project (obviously, we can achieve 100% accurate estimates after the project is complete!).
- (2) Base estimates on similar projects that have already been completed.
- (3) Use relatively simple decomposition techniques to generate project cost and effort estimates.
- (4) Use one or more empirical models for software cost and effort estimation.



## COLLEGE MANAGEMENT SYSTEM



Unfortunately, the first option, however attractive, not practical. Cost estimates must be provided "Up front."

The second option can work reasonable well, if the current project is quite similar to past efforts and other influences are equivalent. Unfortunately, past experience has not always been a good indicator of future results.

The remaining options are viable approaches to software project estimation. Ideally, the techniques noted for each option should be applied in tandem, each used as a crosscheck for the other.

**Though we can estimate a project on the following basis: -**

- (i) Problem Based Estimation.
- (ii) Line of Code (LOC)-Based Estimation.
- (iii) Function Point Based Estimation.
- (iv) Process-Based Estimation.

**Estimation Table for the LOC Method: -**

FUNCTION	ESTIMATED LOC
User Interface and Control Facility (UICF)	4300
Two-Dimensional Geometric Analysis (2DGA)	5300
Three-Dimensional Geometric Analysis (3DGA)	6800
Database Management (DBM)	3350
Computer Graphics Display Facilities (CGDF)	4950
Peripheral Control Function (PCF)	2100
Design Analysis Modules (DAM)	8400

**Estimated Line of Codes**

38200

**Estimation Table for the FP-Based Method**

Measurement Parameter	Count	Simple	Average	Complex	Weight	FP Count
Number of User Inputs	24	20	24	30	4	97
Number of User Outputs	16	12	15	22	5	78
Number of User enquiries	22	16	22	28	5	88
Number of Files	4	4	4	5	10	42
Number of External Interfaces	2	2	2	3	7	15
Count						320

**Processed- Based Estimation Table**

Activity	cc	planning	Risk Analysis	Engineering		Construction release		CE	Totals
Task				Analysis	Design	Code	Test		
Function									
UICF				0.50	2.50	0.40	5.00	n/a	8.40
2DGA				0.75	4.00	0.60	2.00	n/a	7.35
3DGA				0.50	4.00	1.00	3.00	n/a	8.50
CGDF				0.50	3.00	1.00	1.50	n/a	6.00
DBM				0.50	3.00	0.75	1.50	n/a	5.75
PCF				0.25	2.00	0.50	1.50	n/a	4.25
DAM				0.50	2.00	0.50	2.00	n/a	5.00
Totals	0.25	0.25	0.25	3.50	20.50	4.50	16.50		46.00
% Effort	1%	1%	1%	8%	45%	10%	36%		

CC = Customer Communication.

CE = Customer Evaluation.

The COCOMO Model (Constructive Cost model) is one of the most widely used and discussed software cost estimation



## COLLEGE MANAGEMENT SYSTEM



model in the software industry. It has evolved into a more comprehensive estimation model, COCOMO II. Like its predecessor, COCOMO II is actually a hierarchy of estimation models that address the following areas.

- Application composition model.
- Early design stage model.
- Post architecture stage model.

Like all estimation models for software, the COCOMO II models require sizing information. Three different sizing options are available as part of the model hierarchy: -

- A. Object Points.
- B. Function Points
- C. Lines of Source Codes.

Though, simply we can calculate the cost of this project as:

Development Cost = Project Duration \* Hourly Development Rate

$$= 200 \text{ Hrs.} * 200$$

$$= 40,000$$





## **PERT CHART**

PERT (Program Evaluation and Review Technique) was developed by Navy Sponsored Research Team Composed of Messrs D.G.Malcolm, J.R.Roseboom, C.E.Clark and W.Fazar in about 1950. This is essentially management technique and if tailored properly, can be used with advantage for responsibility accounting in addition to attaining other well-defined techniques. It is a method in which we try to exercise logical discipline in planning and controlling projects. Managers have found this technique for immense value here adopted judiciously and when configuration of events and activities are correctly assessed and their times are realistically worked out.

PERT is designed for scheduling complex project that involve many inter-related task. It improves planning because:

- (1) It forms the planner to define the projects various component activities and events logically.
- (2) It provides basis for normal estimates, and yet allows for some measure of optimism or pessimism in estimating completion dates.



## COLLEGE MANAGEMENT SYSTEM



- (3) It shows the effects of changes to the overall plan as they contemplated.
- (4) It provides a built-in means for un-going evaluation of the plan.
- (5) It facilitates the process of communication between planner's management by either adhering to organizational lines or crossing over them. In essence, PERT makes the clear-cut assignment of responsibility possible.

After the project has been planned and its implementation is underway, PERT continues to be of use in controlling the project. PERT improves project control because: -

- (1) It provides all parties involved a common basis of progress reporting, both within organization and outside of it.
- (2) It identifies likely troubles pots before they are encountered.
- (3) It provides date specially tailored to each level of management.
- (4) It focuses management attention on the critical path, where it is most needed, as well as on other non-critical activities that furnish resources, essential to the completion of activities on the critical path.



- (5) It permits effects of various reallocation alternatives to be simulated such that the impact of any proposed changes in overall project can be predicted. In other words, PERT answers the "what if" questions.

Because of these planning and controlling features, PERT especially effective in projects with many distinct tasks, in which the complex interrelationships between tasks and projects with respect to personal scheduling and time constraints of critical.

### **METHODOLOGY OF PERT/CPM NETWORK**

The methodology involved in applying PERT for any project may be split into the following stages: -

#### **(1) Project Planning (Drawing the network): -**

The purpose of this is to identify all the important events, which are essential for completion as well as making up of project, and their dependence upon one another is shown explicitly in the form of a network.

#### **(2) Time Estimation (Network Analysis): -**

Estimates of the time required to perform each of the network activity are made, the estimates are based upon manpower and equipment availability and certain assumptions that may have been made in planning the project. By



## COLLEGE MANAGEMENT SYSTEM



incorporating the time required for completing each of the activity of the network, the project duration as well as the critically of the activities are found. At this stage it is also possible to compute the probability of completing the project or a part of the project by a specified time.

### **(3) Scheduling: -**

The scheduling computations give the earliest and the latest allowable start and each activity, and as a by-product, they identify the critical path through the network, and indicate the amount of the "slack" time associated with the non-critical paths.

### **(4) Time-Cost trade-offs: -**

If schedule time to complete the project as determined in step 3 is satisfactory, the project planning and scheduling may be complete, however, if one interested in determining the cost of reducing the project completion time, then time-cost trade-offs of activity performance times must be considered for those activity on the critical and near critical paths.

### **(5) Resource Allocation: -**

The feasibility of each schedule must be checked with respect to man power and equipment requirements, establishing complete feasibility of a specific schedule may require planning and rescheduling (steps: 1 and 3) or time-



## COLLEGE MANAGEMENT SYSTEM



cost trade-offs (step 4). Hence, final solution may require the performance of a number of cycles of steps 3, 4 and 5.

### **(6) Project Control: -**

When the network plan and schedule have been developed to a satisfactory extent, they are prepared to final form for use in the field. Checking progress against the schedule, assigning and scheduling manpower, equipment, and analyzing the effect of delays control the project. Whenever major changes are made in schedule, the network is revised accordingly and new schedule is computed. In other words, this stage calls for periodic updating of the network amongst other things to monitor the progress the project and making necessary changes in schedules to ensure the completion of the project as close to the target time as possible.

### **PERT NETWORK FOR THE PROJECT**

#### **Critical Path: -**

The list tasks and events are networked in a PERT chart in above figure. The arrow length is not significant, but the sequence and interconnections must give a true picture of the precedence of activities to be completed. The numbers on the activity lines are the time (in weeks) required between events.

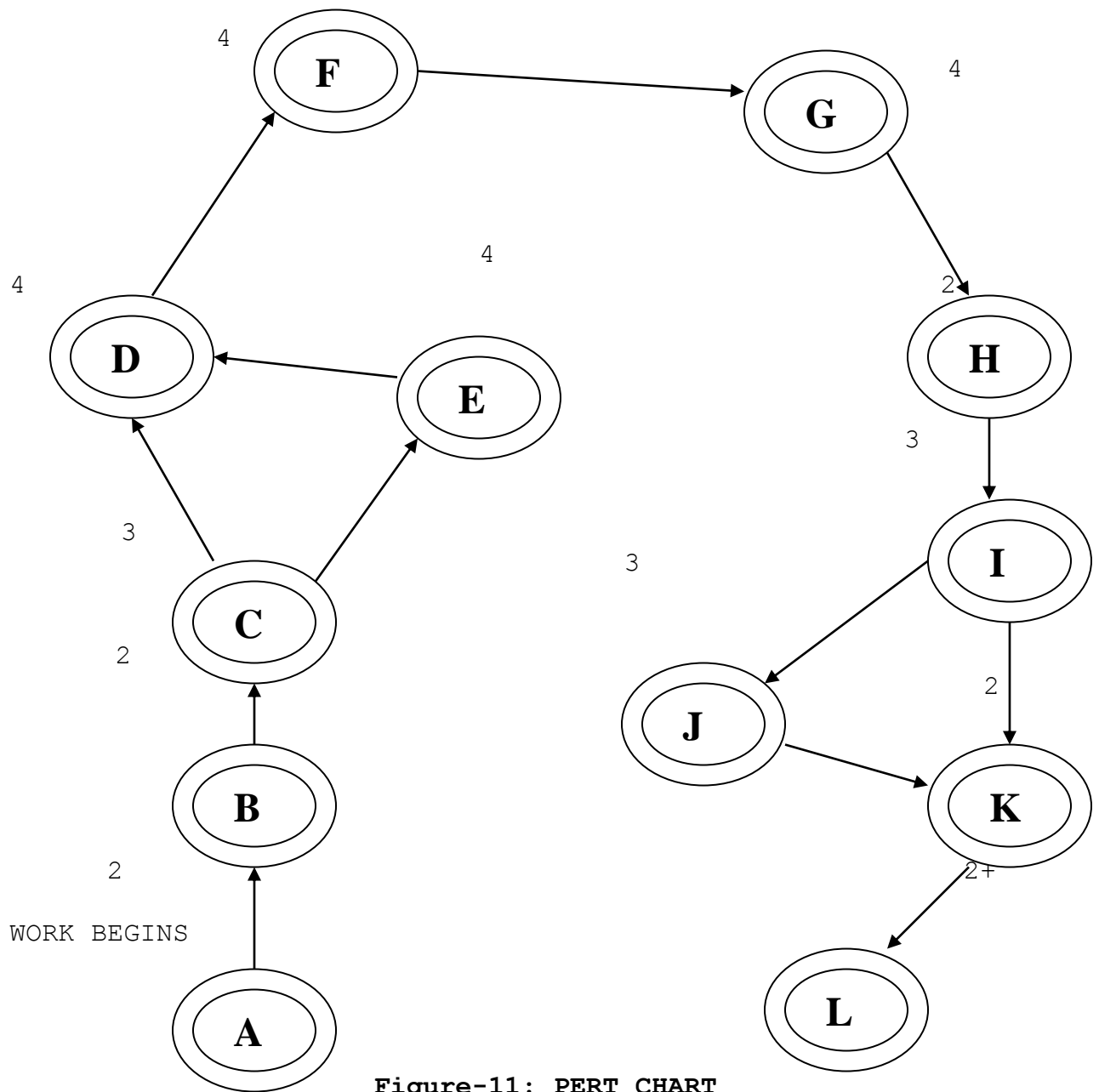


## COLLEGE MANAGEMENT SYSTEM



**Note:** No. In circle denotes events serial no., while those with the arrow denote time taken in weeks.

PERT chart is valuable when a project is being planned. When the network is finished, the next step is to determine the critical path. According to above diagram the critical path is: -





## GANTT CHARTS

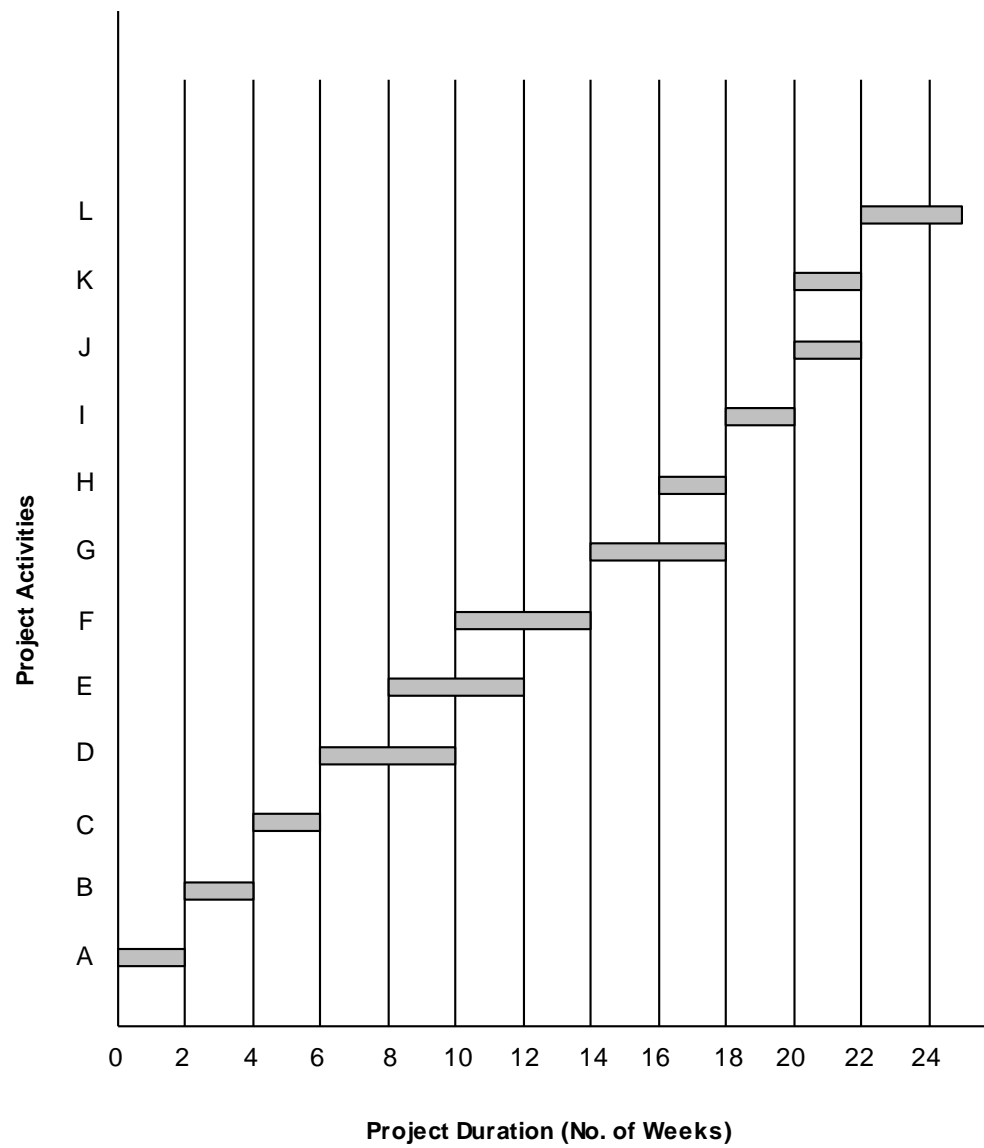
Gantt charts are a project control technique is used for including scheduling. Budgeting and resource planning. A Gantt chart is a Bar Chart, representing an activity (task).

A Gantt chart helps in scheduling the activities of a project, but it does not help in identifying them. They indicate the completion of a particular task and are the basis for determining whether the task and the project are on schedule. I estimate the number of weeks required for each of the 12 tasks.

Activity ID	Activity or Task	No. Of Weeks
A		2
B	<b>Identification of Needs</b>	2
C		3
D	<b>Preliminary Investigation</b>	4
E	<b>Feasibility Study</b>	4
F	<b>Analysis</b>	4
G	Project Proposal & Approval	4
H	Design	2
I	Coding	3
J	Code Optimization	3
K	Testing	2
L	Implementation	...
	Documentation	
	Maintenance	

Some of the six tasks overlapped each other or parallel to some task to reduce time.





**Figure 12: GANTT CHART (TIME LINE CHART)**



## **FUTURE SCOPE AND ENHANCEMENT**

This project has been prepared to maintain an up-to-date planning and environmental inventory for evaluation of development proposals keeping in mind the standard data of Road Complain Management System .This project (Road Complaint Management System) is versatile and flexible project and can be easily modified to suit the requirements in future. The project has been designed in such a way that hierarchy modification / additional capabilities can be effected at any level. The dynamic screen of the project can also be modified easily to meet the specific requirement.



## **LIMITATION OF PROJECT**

There are a lot of possibilities, which could be included in the program, but due to the shortage of time and several other reasons, these are not included in the program. Some of these possibilities are:

- This software doesn't maintain the backup copy parallel.
- There is no recovery facility; means, if any data loses then one can't recover that lost data any way.
- There is no audit-trail facility; therefore one can't check if data is right or wrong.
- ODBC drivers must have installed or Oracle 10g must be installed to run the project properly.
- It is not dedicated to any organization.
- It is purely INTRANET based.
- It supports on-line backup system but doesn't support Off-line backup system.
- The maximum number of records in a table is 50,000.
- Errors generated in project are limited. Generally we used this project only for named exceptions, not for number exceptions.



## **BIBLIOGRAPHY**

### **Books: -**

1. Press Oracle: Oracle 10g Complete Reference Book.
3. Sun Micro system: Beginning Java
4. Deitel & Deitel : JAVA™ How to Program (Sixth Edition)
5. Bayross: Java and Oracle platform
6. Raja Raman: System Analysis and Design.
7. Canter: System Analysis and Design.



# GLOSSORY

- IT - Information Technology
- PERT- Program Evaluation & Review Technique
- AC- Air-condition
- WORM- Write Once Read Many
- SVGA- Super Video-Graphic Adapter
- GB- Giga Byte
- MB- Mega Byte
- RAM- Random Access Memory
- NT- New Technology
- SQL- Structured Query Language
- PL-Programming Language
- CPM-Critical Path Method
- GUI-Graphical User Interface
- LOC-Line of Code
- COCOMO-Constructive Cost Model
- MBPS-Mega Byte Per Second
- UTP-Unshielded Twisted Pair
- LAN-Local Area Network
- TCP-Transmission Control Protocol
- IP-Internet Protocol
- NIC-Network Interface Card