**ATTENDANCE MANAGEMENT SYSTEM**





PROJECT REPORT

ON

**ATTENDANCE MANAGEMENT SYSTEM**

**SUBMITTED IN FULFILLMENT OF THE REQUIREMENT FOR THE EXAMINATION**

**A-LEVEL BY**

AMISHA SHARMA

(REG NO. 1229913)

FROM **NIELIT**

SUBMITTED BY:

AMISHA SHARMA - “A” Level

National Institute of Electronics and Information Technology (NIELIT)

PROJECT GUIDANCE BY:

ALOK BHATT

National Institute of Electronics and Information Technology - Faculty

**PERFORM FOR A/B/C Level PROJECT CERTIFICATE FROM PROJECT GUIDE / ACCREDITED INSTITUTE**

**[For Direct as well as candidate from Accredited Institute]**

This is to certify that the Project /Dissertation entitled,\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_is a Bonafide Work done by Mr./Ms.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(NIELIT Registration No. \_\_\_\_\_\_\_\_\_\_\_\_) in partial fulfillment of A Level /B Level / C Level examination and has been carried out under my direct supervision and guidance . This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Guide/Supervisor

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place:\_\_\_\_\_\_\_\_\_\_\_\_\_

Designation:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date:\_\_\_\_\_\_\_\_\_\_\_\_

Address:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature of Center Manager**

**[in case of a candidate from Accredited Institute].**

**PREFACE**

The main objective of any computer science student is to get as much of practical knowledge as possible . Being and able to have a practical knowledge by developing a project is a lifetime experience.

As a practical knowledge by developing a project is a lifetime experience. As practical knowledge is important theoretical knowledge we are thankful of having a project .

Though the development of the project we had a great experience of various strategies that can be applied in development of .this project is stepping stone for our carrier.

We are pleased to present this project proper care has been taken while organizing the project so that it is to comprehensive . Also various software engineering concept have been implemented.

**Acknowledegment**

I have taken effort in the project. However ,it would not have been possible with the kind support and help of many individuals. I would like extend my sincere thanks to all of them.

I m highly indebted to the staff of National Institute of Electronics and information Technology ,Gorakhpur for giving their such attention and time,,especially to Mr.Alok Bhatt for his expert guidance and supervision in the project ,it was sure that without him I would not have been able to come out with this project in given time.he consistently guided me during the making of project and helped me remove flaws,thus making the project much beautiful and of supreme quality.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **S.NO.** | **TOPICS** |
| **1.** | **OBJECTIVE AND SCOPE.** |
| **2.** | **THEORITICAL BACKGROUND.** |
| **3.** | **DEFINATION OF PROBLEM.** |
| **4.** | **ABSTRACT** |
| **5.** | **INTRODUCTION** |
| **6.** | **FEASIBILITY STUDY & COST AND BENEFIT ANALYSIS.** |
| **7.** | **SYSTEM ANALYSIS AND USER REQUIREMENTS** |
| **8.** | **SYSTEM PLANNING.** |
| **9.** | **METHODOLOGY AND SYSTEM IMPLEMENTATION WATERFALL MODEL** |

|  |  |
| --- | --- |
| **10.** | **SOURCE CODE.** |
| **11.** | **TESTING** |
| **12..** | **SCREENSHORTS** |
| **13.** | BIBLIOGRAPHY |
| .  14. | **FUTURE PLANNING** |

**1.** **OBJECTIVE AND SCOPE**

I have tried my best to develop this project as more user friendly and to be used by employee user. They just have to maintain the student attendance.

This project “Attendence Management System” which is going to be implemented for NIELIT Attendance Mangement System will be automated the major operation of the student Attendance.

1)Least Paper Work: All the work done before on papers will be estimated and will be done on computer will increase efficiency of the system.

2)Time Saving: All the work is done by the computer system itself ,therefore its saves a lot of time of the user.

3)Secured System: This project is also secured , and only those people who have password can able to login to this system.

3)Cost Effective: It is also very cost effective as it reduces the huge amount of cost required in storing the files and also reduces the space required to maintain the file.

For manual attendance system, the most common problem is the class teacher need to take student daily attendance and manually filled the record in attendance book for every month. If the attendance book is missing or misplace, it could lead to big problem because the teacher need the attendance record to make analysis and generate an attendance report. Another problem is the teacher will need more time to analyze and generate the attendance report because the teacher needs to search and refer the old attendance record first.

Besides that, an error could happen when the teacher make the calculations to generate the attendance report by themselves. Even though the attendance record is hassle to keep by the class teacher, management report is required in urgent basis.

Analyzed attendance record is required by for future actions is normally being delay because of the lack of precise. Moreover, delay analyzes would leads to prolong the time to inform the parents about the truancy students. Addressing this issue, this paper discusses the strategy to analyze and generate reports related to attendance, provide trigger facilities in conjunction to pre-defined consequences and develop a prototype of attendance management system

**2. THEORETICAL BACKGROUND.**

The Attendance management system was one of the earliest changes to improve efficiency. Attendance management system eventually evolved into the storing student record in the database. A attendance management System is used for the taking attendance of a particular class and interface of the whole institute which supports institute and colleges and other student organisations in making attendance for most major part in a single system.

Attendance management Systems contain students schedules, student punctuality and records. An attendance direct distribution works within their own present system, as well as pushing out information to the whole attendance. A second type of direct distribution channel is faculty who use the applications to make their own attendance.

**INVENTORY MANAGEMENT**

Under this topic, we discuss about all aspects of our project in theoretical manner. Through this project we can modify the project “Attendance Management System” according to the attendance institute from manual to computerized System . We know that computer based system can do a job in cost effective manner . This project we can search the attendance management system.

**Fare Attendance**

The Fares data store contains fare tariffs, rule sets, , class of service tables, and some more

information that construct the punctual. Rules like daily present (e.g. minimum stay, advance, etc.) are tailored differently between different instutute pairs or zones, and assigned a class of service corresponding to its appropriate inventory bucket. Inventory control can also be manipulated manually through the availability feeds, dynamically controlling how many seats are offered for a particular price by opening and closing particular classes.

**Need of Attendance system**

A few factors that direct us to develop a new system are given below -:

1) Faster System

2) Accuracy

3) Reliability

4) Informative

**3. DEFINITION OF PROBLEM**

The web based “attendance management system” project is an attempt to stimulate the basic concepts of attendance management system. The system enables the customer to do the things such as search for attendance roll number for two college cities on a specified date, choose date based on the details, present days of the student and cancellation of attendance.

The system allows the student to search there present dates that are available between the months or the years, namely the “attendance month” and “attendance days” for a particular and present dates. The system displays all the students details such as roll no, name, phone no and duration of attendance etc.

After search the system display list of students details and allows customer to choose a particular student. Then the system checks for the availability of seats on the students. If the seats are available then the system allows the students to take a seat.

Otherwise it asks the user to choose another class. To take admission in the system asks the student to enter his details such as name, address, city, State, and contact number. Then it checks the validity of student to take the attendance and update the student database and teachers database.

The attendance system is manual system. The main drawback of the existing system is time consumption .There are number of records and job become tedious . this document required more storing space ,more manpower and job become tedious.

**4.ABSTRACT**

ABSTRACT Student attendance management system deals with the maintenance of the student’s attendance details. It is generates the attendance of the student on basis of presence in class. It is maintained on the daily basis of their attendance. the staffs will be provided with the separate username & password to make the student’s status. The staffs handling the particular subjects responsible to make the attendance for all students. Only if the student present on that particular period, the attendance will be calculated. The students attendance reports based on MONTLY and consolidate will be generated.

**5. INTRODUCTION**

Attendance management system is part of the so-called student service system, this Attendance Process is very difficult to understand in General meaning. But we are providing a Solution for that Problem. This system provides a facility to easy access towards student users. They can easily connect through it and just 3 steps. There is no requirement for any type of worker. We are giving a all this facility in one project “Attendance Management System”.

Attendance management Systems contain students schedules, student punctuality and records. An attendance direct distribution works

within their own present system, as well as pushing out information to the whole attendance. A second type of direct distribution channel is faculty who use the applications to make their own attendance.

. Inventory control is typically manipulated from here, using availability feeds to open and close classes of attendance.

The role of the taking attendance complex is to issue and store student detail records and the very small number of data that are still issued. The electronic student information is stored in a database containing the data that historically was printed on a paper attendance including items such as the components.

**Attendance** A admission for an itinerary is made in the attendance system, either .

directly with the college or by an worker. The itinerary includes all the above details needed for the issuance of student data, except the child age .

When the attendance the made, a student will be created which is used to manage the attendance and check in. It is possible to have multiple students details in a single name record.

Record:-

Having a attendance it does not entitle the student to take a particular class. Only when the institute receives the payment or a students redeems miles/points, adetailt is issued which is linked to the addmission which allows the student to take admission.

Traditionally,admission and payment are separate steps, which the time between them are defined in the fare rules when the admission is made.

Each student must hold his/her own air attendance, as shown by an individual roll number, even when attendance are linked by a single table.

**5.1 OVERVIEW OF JAVA**

JAVA is an object-oriented programming language introduced by Sun Microsystems in the year 1995. It was originally developed in the year 1991 by James Gosling and his team. The syntax and the semantics of JAVA are somewhat similar to C++. However, it is regarded as most powerful as C++ and the other high-level programming languages. In the current scenario, JAVA is the most dominant Object-Oriented programming language for developing web-based applications. Apart from web-based applications, JAVA can also be employed to develop other types of applications, such as desktop applications and embedded systems applications.

JAVA is a highly platform independent language because it uses the concept of ***just-in-line*** compilation. In this type of compilation JAVA programs are not directly compiled into the native machine code. Instead, an intermediate machine code called ***byte code*** is generated by the JAVA compiler that can be interpreted on any platform with the help of program known as ***JAVA Interpreter.***

**Characteristics of JAVA:-**

* It is a highly Object-Oriented and platform independent language.
* The programs written in this language are compiled and interpreted in two different phases.
* The programs written in this language are more robust and reliable.
* It is more secure as compared to other high-level programming languages because it does not allow the programmer to access the memory directly.
* It assists the programmers in managing the memory automatically with a feature called garbage collection.
* It also implements the concept of dynamic binding and threading in a better and efficient manner as compared to other Object-Oriented languages.

**5.2 ECLIPSE IDE**

Eclipse is an integrated development environment (IDE) used in computer programming, and is most widely used JAVA IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in JAVA and is primarily used for developing applications using the JAVA programming language; but it may also be used to develop applications in other programming languages via plug-ins, such as C/C++, Python, PERL, Ruby; etc.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments, and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, Peeved is a plug-in that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C, PHP eclipse is a plug-in to eclipse that provides complete development tool for PHP.

Steps for Installing Eclipse IDE on Windows:-

**Step 0: Install JDK**

To use Eclipse IDE for Java programming, you need to first install Java Development Kit (JDK).

**Step 1: Download Eclipse IDE**

Download Eclipse from ***https://www.eclipse.org***. For beginners, choose the 3rd entry "**Eclipse IDE for Java Developers**" (32-bit or 64-bit) (e.g., "eclipse-java-oxygen-1-win32-x86\_64.zip" 161MB) then click on Download link.

**Step 2: Unzip**

To install Eclipse IDE, simply unzip the downloaded file into a directory of your choice (e.g., "E:\myproject").

There is no need to run any installer. Moreover, you can simply delete the entire Eclipse directory when it is no longer needed (without run

ning any un-installer). You are free to move or rename the directory. You can install (unzip) multiple copies of Eclipse in the same machine.

**Eclipse IDE Versions**

Every year, since 2006, the Eclipse foundation releases the Eclipse Platform and a number of other plug-ins in June. List of Eclipse flavors released over the years,

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Platform Version** | **Projects** |
| Callisto | 30 June 2006 | 3.2 | Callisto projects |
| Europa | 29 June 2007 | 3.3 | Europa projects |
| Ganymede | 25 June 2008 | 3.4 | Ganymede projects |
| Galileo | 24 June 2009 | 3.5 | Galileo projects |
| Helios | 23 June 2010 | 3.6 | Helios projects |
| Indigo | 22 June 2011 | 3.7 | Indigo projects |
| Juno | 27 June 2012 | 3.8 and 4.2 | Juno projects |
| Kepler | 26 June 2013 | 4.3 | Kepler projects |
| Luna | 25 June 2014 | 4.4 | Luna projects |
| Mars | 24 June 2015 | 4.5 | Mars projects |
| Neon | 22 June 2016 | 4.6 | Neon Projects |
| Oxygen | 28 June 2017 | 4.7 | Oxygen Projects |

**Introduction to our system:**

Attendance management system is an integrated student processing system. This system includes:

1. attendance
2. student details
3. generate record

4.Addmission of student.

.

The Attendance management system has the following Modules:

1. **User registration module:**

This module is helpful for the registration of the new teacher.

1. **Login module:**

This module performs the login of the registered student. In this module student-id and password is verified.

1. **Attendance module:**

This module performs attendance of student to the attendance module.

1. **Cancellation module:**

This module performs the cancellation student details.

# Introduction to Object Modeling Technique:

OMT consists of building a model of an application domain and then adding implementation details to it during the design of a system.

The Methodology has the following stages:

**1. Analysis:** The analysis model is a concise, precise abstraction of what the desired system must do, not how it will be done.

**2. System** **Design:** The system designer makes high-level decisions about the overall architecture, during system design; the target system is organized into subsystems based on both the analysis structure and the proposed architecture.

**3. Object** **Design:** The object designer builds a design model based on the analysis model but containing implementation details. The designer adds details to the design model in accordance with the strategy established during system design.

**4. Implementation:** The object classes and relationships developed during object design are finally translated into a particular programming language, database, or hardware implementation.

The OMT methodology uses three kinds of models to describe a system.

**1. Object** **Model**: The object model describes the static structure of the objects in a system and their relationships. The object model contains object diagrams.

**2. Dynamic Model:** The dynamic model describes the aspects of a system that change over time. The dynamic model contains state diagram.

**3. Functional Model:** The functional model describes the data value transformations within a system. The functional model contains data flow diagrams.

**Use Case name: User Login**

• **Description:** This use case describes the scenario where the user logs into the application, with the username and password he has provided while registering with the system.

• **Actor:** User or the student.

• **Input:** The user or the teacher creates a username and password at the time of registering with the system. He then uses them to logon to the system and make reservations or view any information.

• **Output:** The application then verifies the authenticity of the username and password that the student has provided and allows the user to view the information available on the system, if the username and password are valid.

**5.3 HARDWARE & SOFTWARE REQUIREMENTS**

**Software Requirement:-**

**Operating System**: WINDOWS 10, WINDOWS 8

**Front End** : Swing

**Back End** : MySql 8.0

**Hardware Requirement:-**

**PROCESSOR** : Intel core i5

**RAM** : 8GB

**CPU** : 2.4GHZ

**HDD** : 1TB

**KEYBOARD** : STANDARD

**6. FEASIBILITY STUDY**

**&COST AND BENEFIT ANALYSIS.**

**Cost benefit analysis** is the comparative study of the **cost** verses **benefit** and savings that are expected from the proposed **system**. Since the organization is well equipped with the required hardware and software so the **Attendance managaement System** was found to be economically.

Feasibility study is to check the viability of the project under consideration. Theoretically various types of feasibilities are conducted, but we have conducted three type of feasibilities explained as under.

**6.1 ECONOMIC FEASIBILITY**

With the manual system the operating cost of the system is about 1Lacks P.A... This cost comprises salary of 25 people, stationary, building rent, electricity, water, telephone etc. But with the new system this reoccurring cost comes out to be about 2 lacks P.A. Hence the new system is economically feasible.

It is evaluation of development cost weight against the ultimate income or benefit derived from the system. Economic justification is generally much weighted (except notable system such as research programs).

Economic justification includes a board of concerns that includes cost benefits

Analysis, long term corporate income strategies, impact on other profit centers or products, cost of resources needed for development and potential market growth. The hardware and software requirement is already setup. Hence the access cost incurred is negligible. Thus making the proposed system economically feasible is very essential.

The system is economically feasible on account of the following reasons:

* There will be only one time capital investment to purchase of hardware and software package. If the software is developed in house cost of readymade software can be saved.
* Maintenance cost is reduced, as there is no need of maintaining same date in various records for various documents.
* As there is no need of extra employees and extra registers, it is cost effective

**6.2 TECHNICAL FEASIBILITY**

The new system requires only 6 trained person to work with the system and in overall 10 people per office are sufficient. So we will identify 6 best people from existing system and train them.

As our existing system is purely manual, so we need a onetime investment of Rs 1 Lacks for the purchase of 2 computers, 1 printers, a laser printer, AC and networking etc. It requires 2Lacks PA as a operating cost.

With the above details our system is technically feasible as after investing 2of the attenjdance or record rate information. In the past, attendance issued paper attendance; since 2008, IATA has been supporting a resolution to move to 100% electronic attendance. So far, the istitute has not been able to comply due to various technological and some limitations. The institute is at 98% electronic attendance issuance today, although electroninc processing for MCOs was not available in time Lacks in a year, the company is still saving Rs 2 Lacks PA.

It is a study of functions, performance and constrains that may affect the ability to achieve an acceptable system. Technical feasibility is frequently the most difficult area to assess at this stage of system development process. The considerations that are normally associated with technical feasibility include.

 **Development Risks**:

Can the system element be designed so that the necessary functions and performance are achieved?

 **Resource Availability**: Are the necessary resources (Hardware & Software) available to build the system?

 **Technology**: Has the relevant technology progressed to a state that will support the system?

It can thus be the misjudgment in technical feasibility study. The system is technically feasible on account of the following reasons:

* The new hardware and software are available to the company. So the company has installed new hardware and software.
* The system runs on single user environment.
* Company has purchased the scanner to store specimen of signature of Shareholders for the verification and to store documents for easy accessibility.

**6.3 OPERATIONAL FEASIBILITY**

The new solution is feasible in all since but operationally it is not. The new system demands the expulsion of at least 15 people from the company. It creates an environment of joblessness and fear among the employees. It can lead to an indefinite strike in the company also. So the management must take corrective actions prior in advance in order to start the further proceedings.

**The proposed system is operational due to the following reason:**

* It provides operating requirements of organization effectively; it gives queries and reports.
* The designed system uses systematic approach and is also menu driven.
* The end user need not necessarily be a computer professional to operate the system and no special training is required for the operating system.
* System is user friendly and at every step in the system help messages lists are displayed. Any person can acquaint himself to the system with little practice.
* Easy access to the different types of information relating to vehicle from time to time immediate and time saving is possible.

**6.4 Legal Feasibility:**

It is a determination of any infringement, violation or liability that could result from development of system. Legal feasibility encompasses a board of concerns that include contracts, liabilities, infringement and myriad other traps frequently unknown to the technical staff.

**7. SYSTEM ANALYSIS AND USER REQUIREMENTS**

**INCEPTION PHASE**

The Inception Phase is the first phase of a software development life cycle. The main objective of the inception phase is to establish the business case for the system and define the scope of the system. The inception phase of the Attendance management System project mainly focuses on defining the project requirements. The primary documents created in the inception phase consist of the Vision document, Software Quality Assurance Plan and the Project Plan.

The Vision document, which is one of the outcomes of the inception phase, mainly focuses on core project requirements and the key features. It also discusses the main features of the project along with the interfaces of the project. The next document which is an outcome of the inception phase is the project plan document. The project plan document as the name says is mainly used to document the schedule of the project as well as the time required for each phase of the project.

This plan, gives us an estimate of when the project will be completed. The Software Quality Assurance Plan, which is also an outcome of the inception phase documents the standards and conventions that need to be followed in order to ensure good quality of the end product. At the end of the inception phase, the developer will give a presentation to the

Supervisory committee after submitting all the documents necessary. This phase will be marked complete, once all the documentation for the phase I is reviewed and approved by the committee.

**ELABORATION PHASE**

The next phase in the software development lifecycle is the elaboration phase. The main purpose of the elaboration phase is to establish a strong architectural foundation for the Attendance management System project. It is the most critical phase of all the phases of the software development lifecycle. The entire architectural design of the Attendance Management System will be documented using the appropriate UML diagrams. In this phase, revisions on the initial versions of the Vision document, Project Plan document will be made based on the suggestions made by the Supervisory Committee members. In the elaboration phase, each component in the Attendance management System architecture will be documented at the interface level. Another deliverable of this phase is the Test Plan which documents all the testing activities that will be carried out on the project and also states how to report and track the test results. The two technical inspectors of the project also perform an architectural review on the project and provide feedback by submitting the formal technical inspection letters which are based upon their findings this was the most important phase of my project life cycle .It had connected my maximum time.

The block diagram given bellow depicts various fact which were understood by one during the analysis phase.

**7.1 Existing System**

Attendance management System is a System including Inventory, Fares, Enquiries, and attendance etc. All user/agents are allocated a SINE code which is used during sign-on and then appended to all transactions carried out by the agent for security purpose.

**It has the following Disadvantages:**

1. Attendance does not support up to 1 year.
2. Accessing and updating the system is slow.

**7.2 Proposed System:**

Using the distributed technology we can handle these problems easily. In general a distributed process means that a program in execution makes use of resources in other machine. The technologies for distributed processing available are J2EE. In the “Attendance management system” J2EE is used for managing distributed systems. So that even if the Attendance system has lot of branches and they are located at different places, we can handle the management of service and guarantee.

The proposed system for the problem is “Attendance management system”, a sytem based system that allows attendance.

.

**It has the following advantages:**

1. Attendance are supported up to 1 year.
2. Accessing and Updating the system is fast.
3. It is implemented using the concepts of RDBMS.

**7.3 Problem Statement**

The objective of the project is to design and implement the software which helps the Attendance System students to issue attendance for various courses and maintain the records of various students and provide quick services to the students. It provides the following services:

* Maintains the student’ records.
* Reports about the monthly attendance System.
* Quick Response to the students.
* Automation and integration of Attendance management functions.
* Higher productivity and effective management
* Security and protection of confidential data.
* Including teachers details.

**8. SYSTEM PLANNING**

The object model describes the structure of objects in a system. Their identity, their relationships to other objects, Their attributes, and their operations. The object model provides the essential framework into which the dynamic and functional models can be placed. Our goal in constructing an object model is to capture those concepts from the real world that are important to an application.

The object model is represented graphically with object diagrams containing object classes. Classes are arranged into hierarchies sharing common structure and behavior and are associated with other classes. Classes define the attribute values carried by each object instance and the operations which each object performs or undergoes

**8.2 Identification of Associations:**

Any dependencies between two or more classes are an association. A reference from one class to another is association. Association often corresponds to verb phrases. These include physical location, directed actions, communication, ownership, or satisfaction of some condition. Extract all the candidates from the problem statement.

**Following are the Associations:**

* Only the authorized students are entered into the system.
* The system is responsible for controlling the operations and giving the confidential details to only authorized teachers.

**8.4. Data Dictionary**

A data dictionary is a structured repository of data about data. The data dictionary describes each object class and it also describes the scope of the class within the current problem, including any assumptions or restrictions on its membership or use. The data dictionary also describes associations, attributes, and operations

**8.5. Object Diagram**

Object diagrams provide a formal graphic notation for modeling objects, classes, and their relationship to one another. Object diagrams are useful both for abstract modeling and for designing actual programs. Object diagrams are concise, easy to understand, and work well in practice. New concepts are illustrated by object diagrams to introduce the notation and clarify our explanation of concepts.

**There are two types of Object diagrams**: Class diagrams and Instance diagrams.

* A class diagram is a schema, pattern or template for describing many possible instances of data. A class diagram describes object classes.
* An instance diagram describes how a particular set of objects relate to each other. An instance diagram describes object instances.

**8.6. Dynamic Modeling:**

The dynamic model describes the aspects of a system that change over time. The dynamic model is used to specify and implement the control aspects of a system. The dynamic model contains state diagrams. A state diagram is a graph whose nodes are states and whose arcs are transitions between states caused by events.

The dynamic model is insignificant for a purely static data repository. Such as a database. The dynamic model is important for interactive systems. For most problems, logical correctness depends on the sequences of interactions, not the exact times of interactions.

Dynamic modeling is a description of aspects of a system concerned with control, including time, sequencing of operations, and interaction of object.

**Following steps are performed in constructing a dynamic model.**

* Prepare scenarios of typical interaction sequences.
* Identify events between objects.
* Prepare an event trace for each scenario
* Built a state diagram
* Match events between objects to verify consistency.
* New student enters the system and makes the registration and gets a student\_id.
* student who are already registered enter the system.
* student makes the attendance operation and gets the response.
* student makes the query for attendance operations & gets back the response.
* student makes the admission cancellation operation and gets the appropriate response.

**8.7Event Flow Diagram:**  Response

**Attendance management system**

**student**

**Attendance**

**FACT FINDING**

After obtaining background knowledge, the analyst being to collect data in the

Existing system’s output input and costs. The tools used for the data collection are as follows:

* Review of written documents
* On site observation
* Interviews & Questionnaires

 **Review of written documents:**

When available all documentation on data carries (Forms, Records, Manual, etc.) is organized and include in procedures manual are the requirements of the system, which helps n determining to what extent they are met by the present system. Unfortunately all manuals are not up to date or may not be readable. Day to day problems may have force changes that are not reflected in the manual.

Furthermore, people have the tendency to ignore procedures and find shortcuts as long as the output is satisfactory. Regarding existing forms, analyst to find out how they are filled out, how useful they are to be read.

 **Onsite observation:**

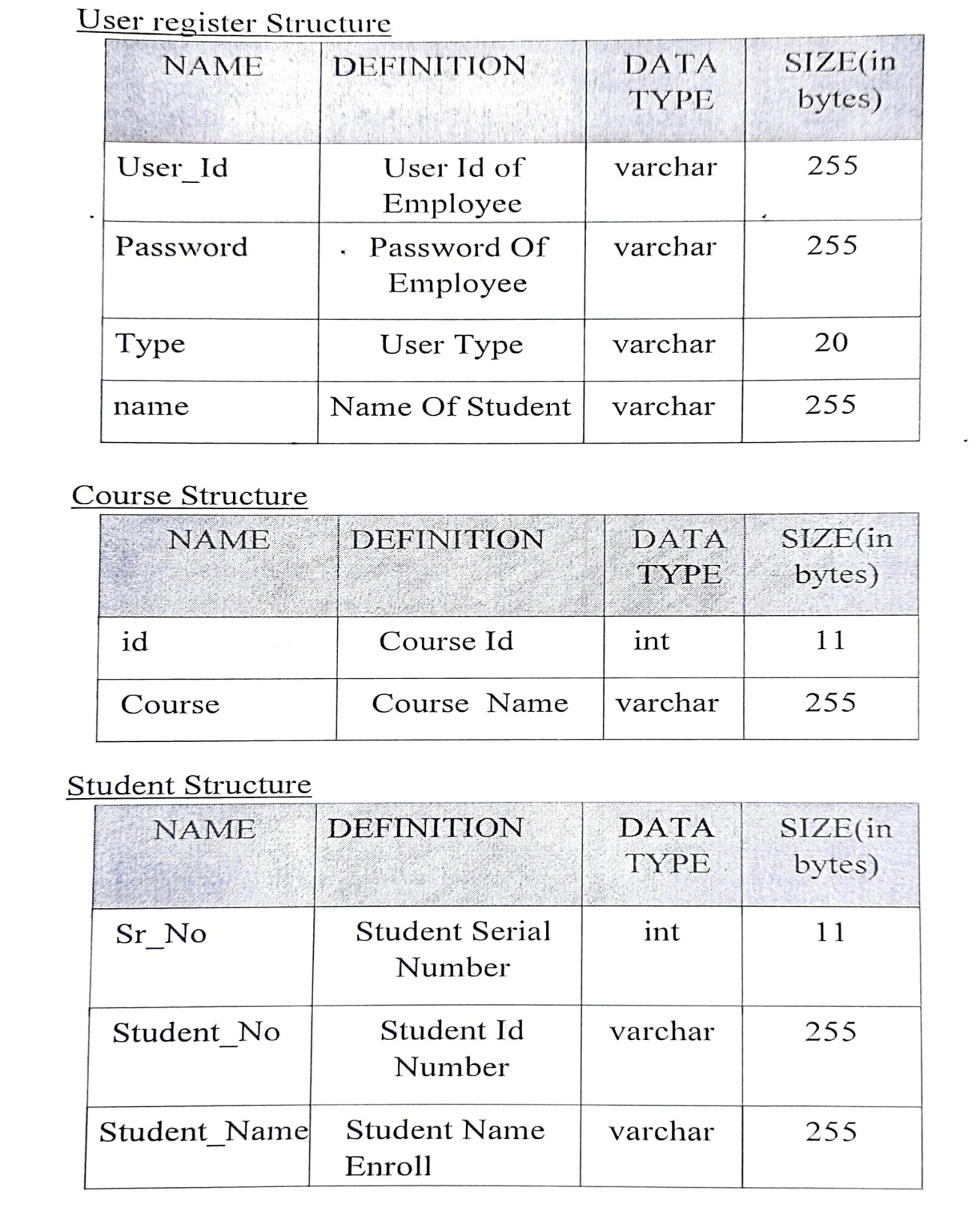
Another fact finding method is used by the system’s analyst is onsite or direct observation. The analyst’s role is that of an information seeker. One purpose of onsite observation is to get as close as possible to the “real” system being studies. As an observer the analyst follows a set of rules. Onsite observation is the most difficult fact finding technique. It requires in to the user area

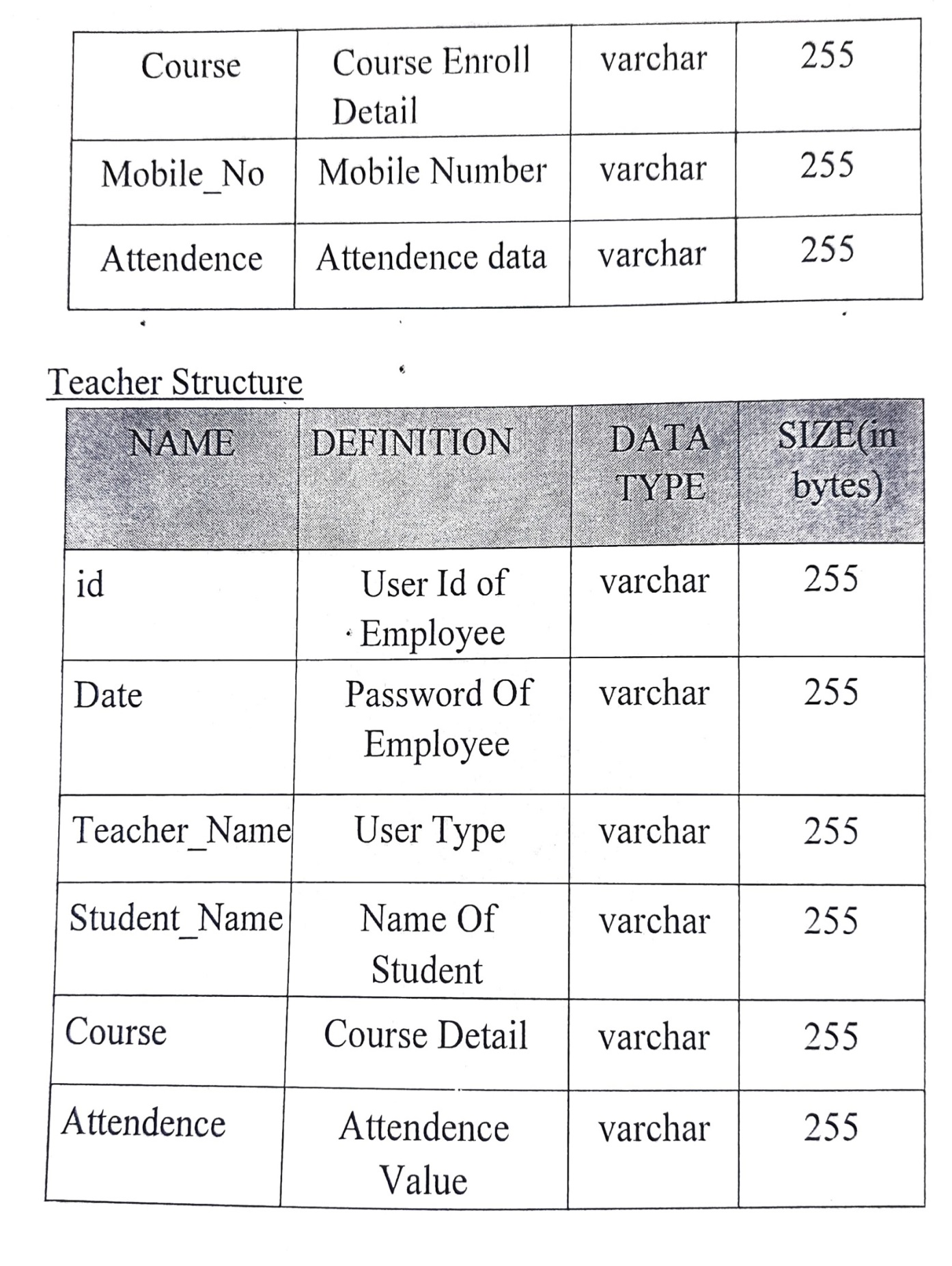
and can cause adviser reaction by the user’s staff if not handled properly. The analyst observes the physical layout of system, location and movement of people and the work flow. He/ she is alert to the behavior of user staff and the people with whom they come into contact. A change in behavior observed in perspective. If onsite observation is to be done properly in a complex situation can be time consuming. Proper sampling procedures is to be used to ascertain the stability, interface drawn from small of behavior can prove inaccurate and therefore unreliable.

**Interviews and Questionnaires:**

As we have discussed onsite observation is directed toward describing and understanding events and behavior as they occur. This method however is less effective for learning about people’s perception, feelings and motivations. The alternatives are the personal interview and the questionnaires. In either method heavy reliance is placed on the interview’s reports for information about job, the present system or experience. The quality of response judged in terms of its reliability and validity. Reliability means that the information gathered is dependable enough to be used for making decision about the system being studied. Validity means that the questions asked are so worded as to elicit the intended information. So, reliability and validity of data gathered depend on the design of the interviews or questionnaires and the manner in which each instrument is administered. In an interview, since the analyst and the person in the interviews meet face to face, there is an opportunity for greater flexibility in eliciting information

**8.9. DATA DICTIONARY**



****

**9. METHODOLOGY AND SYSTEM IMPLEMENTATION**

**WATERFALL MODEL:**

**PROCESS MODEL** We have used LINEAR SEQUENTIAL **MODEL** or **WATERFALL MODEL** for our **system** which is a software **development model** in which **development** is seen as flowing steadily downwards (like a **waterfall**) through the phases of requirements analysis, design, implementation, testing

In this phase initially I had designed E-R diagram of the processes, in order to identify various entities and relationship set, entity set, attributers, link attributes The Diagram of this process as under. After this step we had tried design the data base for the new system and normalized it The tables motivated in data dictionaries enclosed as annex II is an outcome of this step

**The symbol given below:-**

Student\_info

**M**

**1**

**M**

attendance\_info

**1**

Course\_name

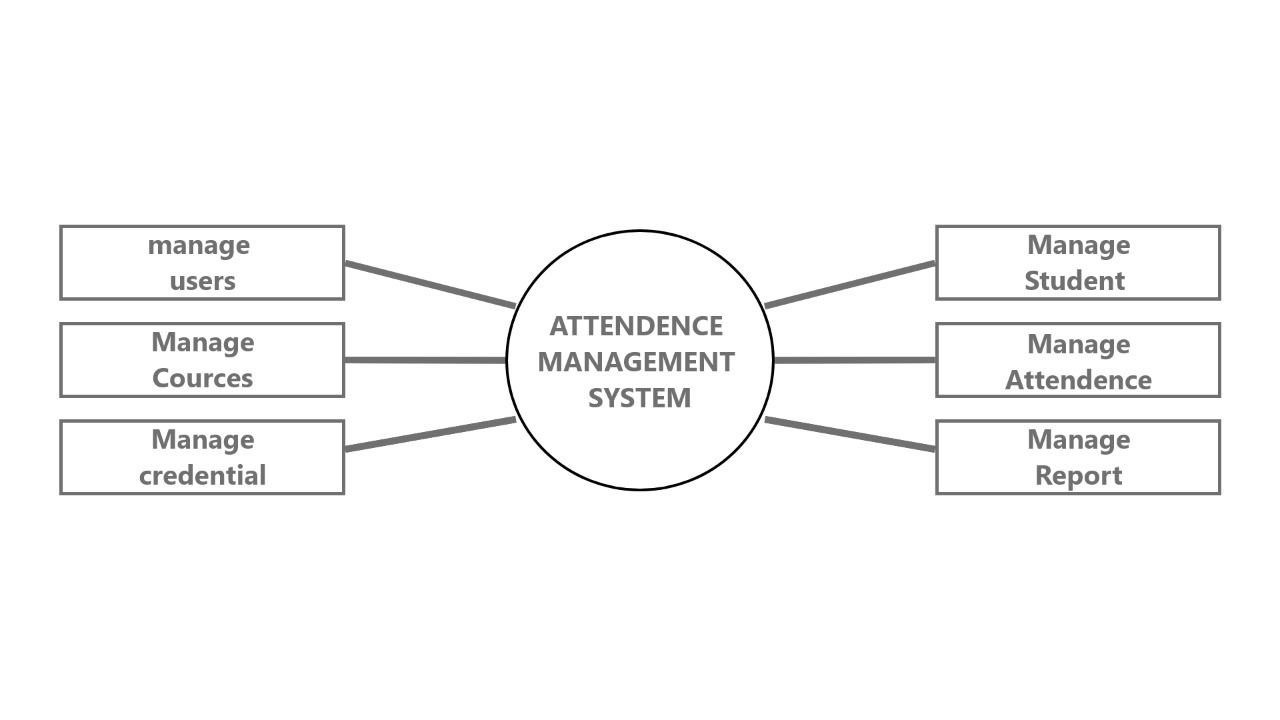
**E-R DIAGRAM**

Login\_info

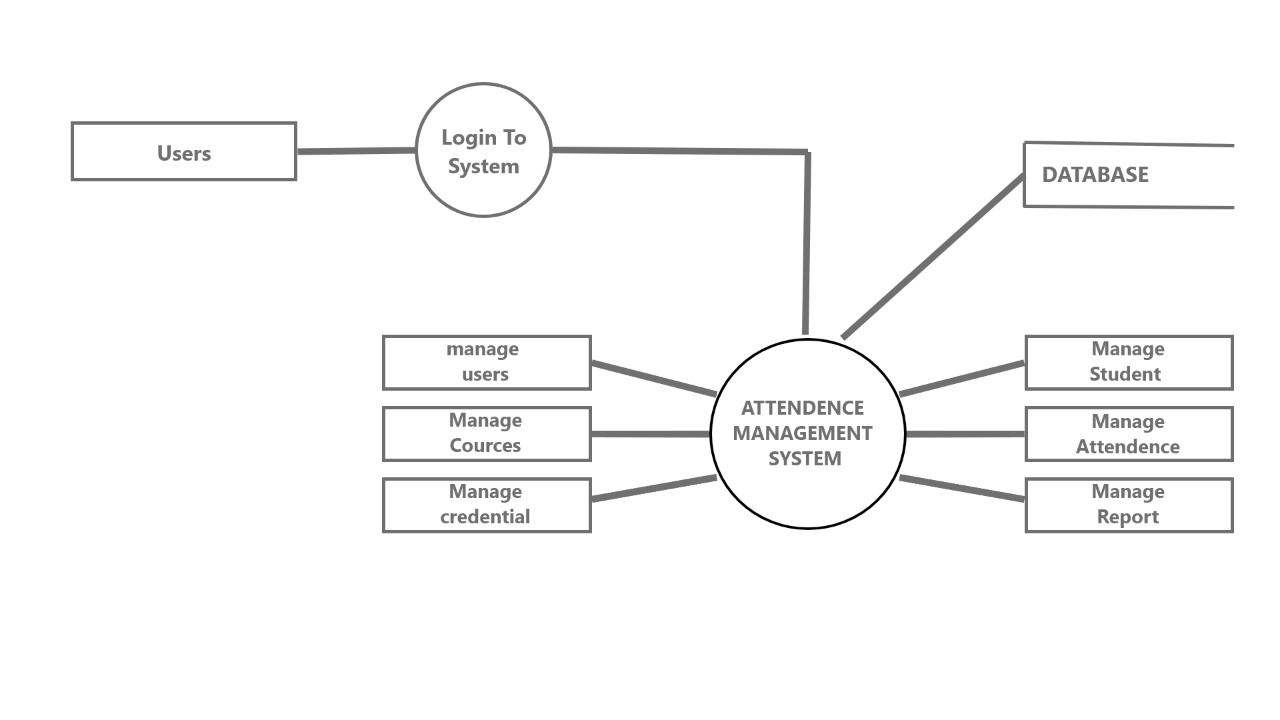
**9.3. DATA FLOW DIAGRAM**

In order to design a better solution. I had designed the DFD for system including all technical processing details is given bellow.

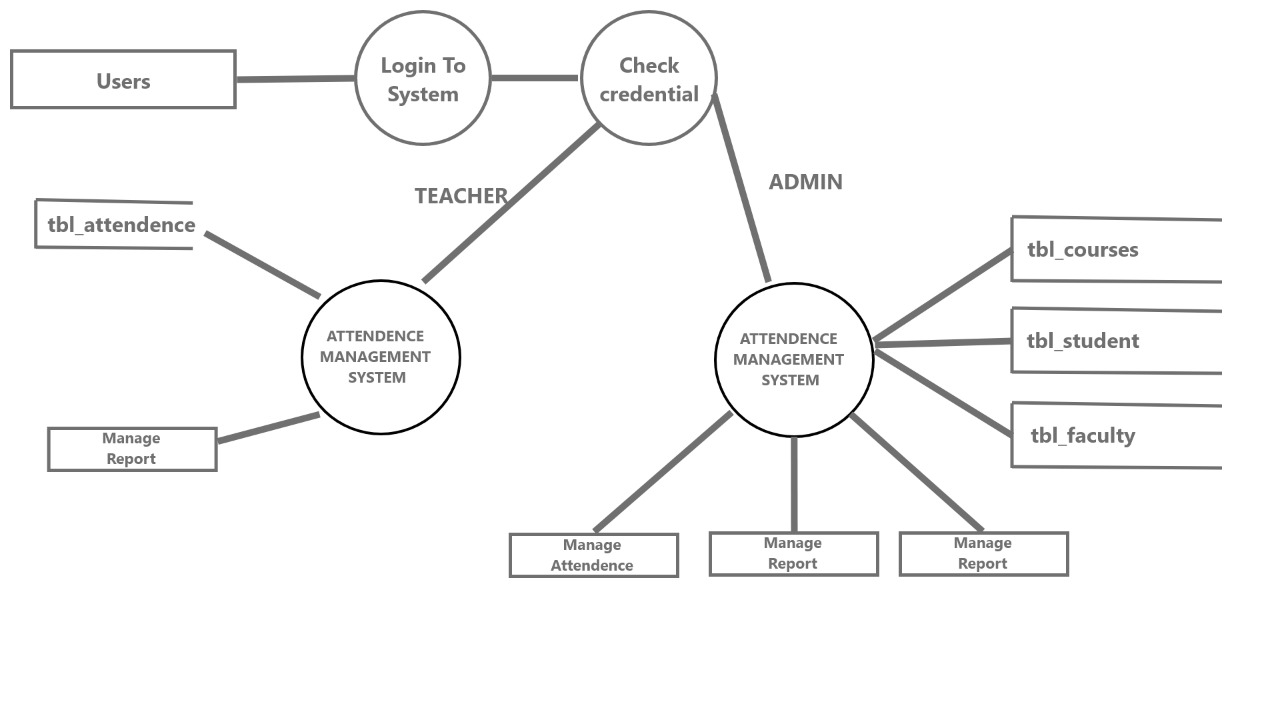
**LEVEL CONTEXT LEVEL DIAG**



**LEVEL 1 DATA FLOW DIAGRAM**



**LEVEL 2 DFD**



The problem analysis is the most important phase in any project. Only after knowing precisely what the problem is could we successfully eliminate it. The identification of the root problem is necessary.

We were able to discuss with the personnel of various departments and gather information and we got a clear picture of what the existing problem were and what our jobs was to eliminate them by redesigning a new design.

Design is a multi step process that focuses on data structure, software architecture, Procedural details (algorithms etc) and interface between the modules. The design process also translates the requirements into the representations of the software that can be assessed for quality before coding begins.

Computer software design changes continually as new methods, better analysis and border understanding evolve. Software design is at a relatively early flexibility and quantitative nature that is normally associated with more classical engineering design disciplines. However, techniques for software design to exist criteria for design qualities are available and design notation can be applied.

Once the software requirements have been analyzed and specified, software design is the first of three technical activities- Design code and test that are required to build and verify the software. Each activity transforms information in a manner that ultimately results in validation of the computer software.

The importance of the software design can be started with a single word quality. Design is the place where quality fostered in software development. Design provides us with representations of the software that can be accessed for quality.

Design is the only way that we can accurately translate a customer’s requirements into a finished software product or system. Without design, risk of building an unstable system exists one that will fail when small changes are made. One that may be difficult to test.

**Thus the system design includes following three types of design:**

* **Data Design:** The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software.
* **Architectural Design:** The architectural design defines the relationship among the major structural components of the program.
* **Procedural Design:** The procedural design transforms structural components into a procedural description of the software. Source code is generated and testing is conducted to integrate and validate the software.

Thus, system design is a solution, a “how to” approach to the creation of the new system.

**9.3 TABLES**

**Database Design:**

The database design converts the data model developed in logical design to a database definition that is supported by database software. Database design proceeds through a number of steps.

The first step is independent of the kind of DBMS used. This step converts the conceptual entity relationships model to a set of record type is known as the logical record structures. (LRS)

The next database design step converts the LRS to a database definition. These steps use techniques that depend on the DBMS. DBMS dependent techniques are needed here because different DBMS support different kind of links between the records. Such links are used to retrieve records by following the link from one record to another. Database design depends on the structure supported by DBMS and uses techniques appropriate to these structures.

DBMS dependent design proceeds in two stages. The first step is logical design. Logical design defines the DBMS record types and the links between them. The next step is physical design. This step chooses a physical organization that supports the methods uses to accesses the databases.

The first thing in system design the input and output screen design according to the needs of the user the input and output design are related with each other in the sense that the accuracy of output data depends on the accuracy of the input data and processing of the input data and processing of the input data. Thus for this proposed system the input and output design are in the form of the forms. In the forms based interface design, the user can easily input data without any difficulty.

**10. SOURCE CODE**

**Login Page**

**package attendanceSystem;**

**/\***

**\*@author AMISHA SHARMA**

**\*/**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.ResultSet;**

**import java.sql.Statement;**

**public class login extends JFrame implements ActionListener{**

**JLabel l1,l2,l3;**

**JTextField t1;**

**JButton b1,b2;**

**JPanel p1,p2,p3,p4;**

**JPasswordField t2;**

**login()**

**{**

**super("Admin Login");**

**setSize(400,250);**

**setResizable(false);**

**setLocation(500,250);**

**/\* ImageIcon img1=new ImageIcon("Electricity\_billing/image/screen2.jpg");**

**setIco nImage(img1.getImage());\*/**

**l1=new JLabel("User Name");**

**l2=new JLabel("Password");**

**ImageIcon i1=new ImageIcon(ClassLoader.getSystemResource("attendanceSystem/image/logo4.jpg"));**

**Image img=i1.getImage().getScaledInstance(170, 170, Image.SCALE\_DEFAULT);**

**ImageIcon i2=new ImageIcon(img);**

**l3=new JLabel(i2);**

**t1=new JTextField(14);**

**t2=new JPasswordField(14);**

**ImageIcon i3=new ImageIcon(ClassLoader.getSystemResource("attendanceSystem/image/logo7.png"));**

**Image img2=i3.getImage().getScaledInstance(20, 20, Image.SCALE\_DEFAULT);**

**b1=new JButton("Login",new ImageIcon(img2));**

**ImageIcon i4=new ImageIcon(ClassLoader.getSystemResource("attendanceSystem/image/logo8.png"));**

**Image img3=i4.getImage().getScaledInstance(16, 16, Image.SCALE\_DEFAULT);**

**b2=new JButton("Cancle",new ImageIcon(img3));**

**Font f;**

**f =new Font("Arial",Font.BOLD,16);**

**l1.setFont(f);**

**l2.setFont(f);**

**b1.setFont(f);**

**b2.setFont(f);**

**t1.setFont(f);**

**t2.setFont(f);**

**p1=new JPanel();**

**p2=new JPanel();**

**p3=new JPanel();**

**p4=new JPanel();**

**setLayout(new BorderLayout());**

**p2.add(l1);**

**p2.add(t1);**

**p2.add(l2);**

**p2.add(t2);**

**add(p2,BorderLayout.CENTER);**

**p1.add(l3);**

**add(p1,BorderLayout.WEST);**

**p3.add(b1);**

**p3.add(b2);**

**add(p3,BorderLayout.SOUTH);**

**b1.addActionListener(this);**

**b2.addActionListener(this);**

**setVisible(true);**

**}**

**//main method**

**public static void main(String arg[]) {**

**new login();**

**}**

**public void actionPerformed(ActionEvent ev)**

**{**

**try {**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection cn=DriverManager.getConnection("jdbc:mysql://localhost:3306/attendance\_system","root","");**

**if(ev.getSource()==b1)**

**{**

**String name=t1.getText();**

**String pass = t2.getText();**

**if(name.isEmpty() || pass.isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Please Enter your username and password as well...");**

**}else {**

**String q="select \* from login\_info where usr\_name='"+name+"' and passwd='"+pass+"'";**

**Statement stm=cn.createStatement();**

**ResultSet set=stm.executeQuery(q);**

**if(set.next())**

**{**

**// JOptionPane.showMessageDialog(null, "Login successfull");**

**new home().setVisible(true);**

**this.setVisible(false);**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null,"invalid login");**

**}**

**}**

**}**

**else**

**{**

**setVisible(false);**

**new mainPage().setVisible(true);**

**}**

**}**

**catch(Exception e)**

**{**

**e.printStackTrace();**

**System.out.println("Error.........");**

**}**

**}**

**}**

**Home page:**

**package attendanceSystem;**

**/\***

**\*@author Satya Narayan Mishra**

**\*/**

**import attendanceSystem.student\_details;**

**import attendanceSystem.generate\_list;**

**import attendanceSystem.home;**

**import attendanceSystem.new\_student;**

**import javax.swing.\*;**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.ResultSet;**

**import java.sql.SQLException;**

**import java.text.DateFormat;**

**import java.text.SimpleDateFormat;**

**import java.time.LocalDateTime;**

**import java.time.format.DateTimeFormatter;**

**import java.util.Calendar;**

**import java.util.Date;**

**import com.jgoodies.forms.layout.FormLayout;**

**import com.jgoodies.forms.layout.ColumnSpec;**

**import com.jgoodies.forms.layout.RowSpec;**

**import com.mysql.cj.protocol.Resultset;**

**import net.miginfocom.swing.MigLayout;**

**import com.toedter.calendar.JCalendar;**

**import com.toedter.calendar.JDateChooser;**

**//Table creation class================================================**

**class tableCreation{**

**private JScrollPane sp;**

**private JTable t1;**

**private JCheckBox cb;**

**String tableData;**

**int counter=0,flag=0,i=0,j=0;**

**void create(String course,JLabel welcomeMsg1,JLabel welcomeMsg2,JFrame f) {**

**connection conObj=new connection();**

**String query="select \* FROM `student\_info` where `course\_name`='"+course+"'";**

**try {**

**ResultSet data=conObj.stm.executeQuery(query);**

**while(data.next()) {**

**flag=1;**

**counter+=1;**

**}**

**//Dynamic Table creation**

**String x[]= {"S.N","Roll.NO","Student's Name","Father's Name","Attendance Date","Attendance"};**

**String y[][]=new String[counter][6];**

**// students attendance, dynamic data creation**

**t1=new JTable(y,x);**

**sp=new JScrollPane(t1);**

**sp.setBounds(230, 150, 900, 400);**

**f.add(sp);**

**sp.setVisible(false);**

**if(flag==1) {**

**// new home().setVisible(true);**

**// f.hide();**

**welcomeMsg1.setVisible(false);**

**welcomeMsg2.setVisible(false);**

**ResultSet dbData=conObj.stm.executeQuery(query);**

**while(dbData.next()) {**

**Integer sq=i+1;**

**y[i][j++]=sq.toString();**

**y[i][j++]=dbData.getString("roll\_no");**

**y[i][j++]=dbData.getString("stu\_name");**

**y[i][j++]=dbData.getString("father");**

**y[i][j++]=dbData.getString("regi\_date");**

**i++;**

**j=0;**

**}**

**System.out.print(y[0][4]);**

**counter=0;**

**i=0;j=0;**

**sp.setVisible(true);**

**}else {**

**//showing text (No data found)**

**new home().setVisible(true);**

**JOptionPane.showMessageDialog(null, "Opps ! no any student available about this course", "Warning Message", JOptionPane.WARNING\_MESSAGE);**

**f.hide();**

**}**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**// Main Class===============================================**

**public class home extends JFrame implements ActionListener {**

**private Choice choice;**

**private JDateChooser dateChooser;**

**private JButton takeAttendance,refresh,update;**

**private JLabel welcomeMsg1,welcomeMsg2,noData;**

**home()**

**{**

**//------------------------------------ Title --------------------------------------------------------------**

**setTitle("Attendance Management System");**

**setBounds(0,5,1365,730);**

**setResizable(false);**

**JMenuBar mb=new JMenuBar();**

**JMenu master=new JMenu("Master");**

**JMenuItem mi1=new JMenuItem("New Student");**

**JMenuItem mi2=new JMenuItem("Student Management & Details");**

**master.setForeground(Color.BLUE);**

**Font f=new Font("monospaced",Font.PLAIN,16);**

**mi1.setFont(f);**

**ImageIcon img\_mi1=new ImageIcon(ClassLoader.getSystemResource("attendanceSystem/image/logo1.png"));**

**Image image\_mi1=img\_mi1.getImage().getScaledInstance(16,20,Image.SCALE\_DEFAULT);**

**mi1.setIcon(new ImageIcon(image\_mi1));**

**mi1.setBackground(Color.WHITE);**

**mi2.setFont(f);**

**ImageIcon img\_mi2=new ImageIcon(ClassLoader.getSystemResource("attendanceSystem/image/logo7.png"));**

**Image image\_mi2=img\_mi2.getImage().getScaledInstance(16,20,Image.SCALE\_DEFAULT);**

**mi2.setIcon(new ImageIcon(image\_mi2));**

**mi2.setBackground(Color.WHITE);**

**mi1.addActionListener(this);**

**mi2.addActionListener(this);**

**JMenu report=new JMenu("Management");**

**JMenuItem r1=new JMenuItem("Generate Record");**

**r1.setFont(f);**

**r1.addActionListener(this);**

**JMenu exit=new JMenu("Exit");**

**JMenuItem ex=new JMenuItem("Exit");**

**ex.setFont(f);**

**ex.setBackground(Color.WHITE);**

**ex.addActionListener(this);**

**master.add(mi1);**

**master.add(mi2);**

**report.add(r1);**

**exit.add(ex);**

**mb.add(master);**

**mb.add(report);**

**mb.add(exit);**

**setJMenuBar(mb);**

**setFont(new Font("Senserif",Font.BOLD,16));**

**getContentPane().setLayout(null);**

**JLabel Title = new JLabel("Attendance Management System");**

**Title.setHorizontalAlignment(SwingConstants.CENTER);**

**Title.setForeground(new Color(0, 139, 139));**

**Title.setBackground(new Color(72, 209, 204));**

**Title.setBounds(10, 0, 1349, 52);**

**Title.setFont(new Font("Cambria Math", Font.BOLD, 20));**

**getContentPane().add(Title);**

**JLabel titleUnderline = new JLabel("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**titleUnderline.setHorizontalAlignment(SwingConstants.CENTER);**

**titleUnderline.setBounds(0, 34, 1349, 14);**

**getContentPane().add(titleUnderline);**

**//-------------------------------------------- Date and courses ---------------------------------------------------**

**JLabel date = new JLabel("Date:");**

**date.setFont(new Font("Tahoma", Font.BOLD, 15));**

**date.setBounds(807, 83, 46, 14);**

**getContentPane().add(date);**

**JLabel course = new JLabel("Course :");**

**course.setFont(new Font("Tahoma", Font.BOLD, 15));**

**course.setBounds(997, 83, 63, 14);**

**getContentPane().add(course);**

**// JList list = new JList();**

**// list.setBounds(1201, 89, 63, -10);**

**// getContentPane().add(list);**

**choice = new Choice();**

**choice.add("CCC");**

**choice.add("Tally");**

**choice.add("O-Level");**

**choice.add("A-Level");**

**choice.add("B-Level");**

**choice.add("C-Level");**

**choice.add("M.Tech");**

**choice.setBounds(1066, 83, 105, 20);**

**getContentPane().add(choice);**

**dateChooser = new JDateChooser();**

**dateChooser.setBounds(848, 83, 139, 20);**

**getContentPane().add(dateChooser);**

**dateChooser.setDateFormatString("yyyy-MM-dd");**

**//By default set today date**

**Date d= new Date();**

**dateChooser.setDate(d);**

**takeAttendance = new JButton("Take Attendance");**

**takeAttendance.setFont(new Font("Tahoma", Font.BOLD, 11));**

**takeAttendance.setBounds(1177, 83, 144, 23);**

**getContentPane().add(takeAttendance);**

**welcomeMsg1 = new JLabel("Welcome to Attendance Management System");**

**welcomeMsg1.setFont(new Font("Cambria Math", Font.BOLD, 35));**

**welcomeMsg1.setForeground(new Color(128, 0, 0));**

**welcomeMsg1.setHorizontalAlignment(SwingConstants.CENTER);**

**welcomeMsg1.setBounds(134, 266, 1173, 52);**

**getContentPane().add(welcomeMsg1);**

**welcomeMsg2 = new JLabel("Please select today's date and course then you can take a attendance.");**

**welcomeMsg2.setHorizontalAlignment(SwingConstants.CENTER);**

**welcomeMsg2.setFont(new Font("Sitka Small", Font.PLAIN, 15));**

**welcomeMsg2.setBounds(120, 329, 1187, 23);**

**getContentPane().add(welcomeMsg2);**

**takeAttendance.addActionListener(this);**

**setVisible(false);**

**setDefaultCloseOperation(this.EXIT\_ON\_CLOSE);**

**}**

**public static void main(String args[])**

**{**

**new home().setVisible(true);**

**}**

**@Override**

**public void actionPerformed(ActionEvent ev) {**

**// TODO Auto-generated method stub**

**String msg=ev.getActionCommand();**

**if(msg.equals("New Student")){**

**new new\_student().setVisible(true);**

**}**

**else if(msg.equals("Student Management & Details")){**

**new student\_details().setVisible(true);**

**}**

**else if(msg.equals("Generate Record")){**

**new generate\_list().setVisible(true);**

**}**

**else if(msg.equals("Exit")){**

**System.exit(0);**

**}**

**// --------------------------------------- Students Attendance ----------------------------------------------**

**if(ev.getSource() == takeAttendance) {**

**DateFormat df=new SimpleDateFormat("yyyy-MM-dd");**

**String Fdate=df.format(dateChooser.getDate());**

**if(Fdate.isEmpty()) {**

**// when date is not selected then do nothing.**

**}else {**

**String arr\_date[]=Fdate.split("-");**

**int date=Integer.parseInt(arr\_date[2]);**

**int month=Integer.parseInt(arr\_date[1]);**

**int year=Integer.parseInt(arr\_date[0]);**

**int flag=0,counter=0;**

**String course=choice.getSelectedItem().toLowerCase();**

**connection conObj=new connection();**

**String query="select \* FROM `student\_info` where `course\_name`='"+course+"'";**

**try {**

**ResultSet data=conObj.stm.executeQuery(query);**

**while(data.next()) {**

**flag=1;**

**counter+=1;**

**}**

**if(flag==1) {**

**// Open another window for taking attendance......**

**new Attendance(date,month,year,course,counter).setVisible(true);**

**}else {**

**//showing text (No data found)**

**new home().setVisible(true);**

**JOptionPane.showMessageDialog(null, "Opps ! no any student available about this course", "Warning Message", JOptionPane.WARNING\_MESSAGE);**

**this.hide();**

**}**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**}**

**}**

**Main page:**

**package attendanceSystem;**

**import java.awt.EventQueue;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import java.awt.BorderLayout;**

**import javax.swing.ImageIcon;**

**import java.awt.Font;**

**import javax.swing.SwingConstants;**

**import javax.swing.JButton;**

**public class mainPage extends JFrame implements ActionListener{**

**/\*\***

**\* Launch the application.**

**\*/**

**public static void main(String[] args) {**

**EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**try {**

**mainPage window = new mainPage();**

**window.setVisible(true);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**});**

**}**

**/\*\***

**\* Create the application.**

**\*/**

**public mainPage() {**

**initialize();**

**}**

**/\*\***

**\* Initialize the contents of the frame.**

**\*/**

**private JButton login,registration;**

**private void initialize() {**

**setBounds(90, 30, 1200, 700);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**getContentPane().setLayout(null);**

**JLabel lblNewLabel\_1 = new JLabel("Attendance Management System");**

**lblNewLabel\_1.setBounds(0, 0, 1184, 30);**

**lblNewLabel\_1.setHorizontalAlignment(SwingConstants.CENTER);**

**lblNewLabel\_1.setFont(new Font("Cambria Math", Font.PLAIN, 25));**

**getContentPane().add(lblNewLabel\_1);**

**login = new JButton("Login");**

**login.setFont(new Font("Tahoma", Font.BOLD, 11));**

**login.setBounds(93, 552, 129, 30);**

**getContentPane().add(login);**

**login.addActionListener(this);**

**registration = new JButton("Registration");**

**registration.setFont(new Font("Tahoma", Font.BOLD, 11));**

**registration.setBounds(232, 552, 129, 30);**

**getContentPane().add(registration);**

**registration.addActionListener(this);**

**JLabel lblNewLabel = new JLabel("");**

**lblNewLabel.setBounds(0, 30, 1184, 631);**

**lblNewLabel.setIcon(new ImageIcon(mainPage.class.getResource("/attendanceSystem/image/theme2.png")));**

**getContentPane().add(lblNewLabel);**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**if(e.getSource().equals(login)) {**

**this.setVisible(false);**

**new login().setVisible(true);**

**}else if(e.getSource().equals(registration)) {**

**this.setVisible(false);**

**new Registration().setVisible(true);**

**}**

**}**

**}**

**Adding new student:**

**package attendanceSystem;**

**import javax.swing.\*;**

**/\***

**\*@author Satya Narayan Mishra**

**\*/**

**import java.awt.\*;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.ResultSet;**

**import com.jgoodies.forms.layout.FormLayout;**

**import com.jgoodies.forms.layout.ColumnSpec;**

**import com.jgoodies.forms.layout.RowSpec;**

**import com.jgoodies.forms.layout.FormSpecs;**

**public class new\_student extends JFrame implements ActionListener{**

**private Choice Troll\_no;**

**private JTextField Tname;**

**private JTextField Tfather;**

**private JTextField Tmobile;**

**private JTextField Taddress;**

**private JButton register;**

**private Integer dbRollno=0,flag=0;**

**private Choice courseChoice,chooseState;**

**private ButtonGroup bg;**

**private JRadioButton male,female,other;**

**new\_student(){**

**getContentPane().setFont(new Font("Tahoma", Font.PLAIN, 13));**

**setBounds(380,150,600,350);**

**setTitle("New Student Registration");**

**SpringLayout springLayout = new SpringLayout();**

**getContentPane().setLayout(springLayout);**

**setResizable(false);**

**JLabel reg\_form = new JLabel("Registration Form");**

**springLayout.putConstraint(SpringLayout.NORTH, reg\_form, 10, SpringLayout.NORTH, getContentPane());**

**springLayout.putConstraint(SpringLayout.EAST, reg\_form, -212, SpringLayout.EAST, getContentPane());**

**reg\_form.setForeground(new Color(0, 128, 128));**

**reg\_form.setFont(new Font("Cambria Math", Font.BOLD, 20));**

**getContentPane().add(reg\_form);**

**//Roll no-------------------**

**JLabel roll\_no = new JLabel("Roll No");**

**roll\_no.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**roll\_no.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(roll\_no);**

**Troll\_no = new Choice();**

**springLayout.putConstraint(SpringLayout.WEST, Troll\_no, 213, SpringLayout.WEST, getContentPane());**

**springLayout.putConstraint(SpringLayout.SOUTH, Troll\_no, -231, SpringLayout.SOUTH, getContentPane());**

**springLayout.putConstraint(SpringLayout.EAST, Troll\_no, -63, SpringLayout.EAST, getContentPane());**

**getContentPane().add(Troll\_no);**

**//Add roll number in choice button from database**

**connection conObj=new connection();**

**try {**

**String sql="select `roll\_no` from `student\_info`";**

**ResultSet data2=conObj.stm.executeQuery(sql);**

**while(data2.next()) {**

**dbRollno=data2.getInt(1);**

**flag=1;**

**}**

**if(flag==0) {**

**dbRollno=1;**

**}else {**

**dbRollno+=1;**

**}**

**}catch(Exception ex) {**

**ex.printStackTrace();**

**}**

**Troll\_no.add(dbRollno.toString());**

**//Student name--------------**

**JLabel stu\_name = new JLabel("Student Name ");**

**springLayout.putConstraint(SpringLayout.NORTH, stu\_name, 87, SpringLayout.NORTH, getContentPane());**

**springLayout.putConstraint(SpringLayout.SOUTH, roll\_no, -10, SpringLayout.NORTH, stu\_name);**

**springLayout.putConstraint(SpringLayout.EAST, stu\_name, -406, SpringLayout.EAST, getContentPane());**

**springLayout.putConstraint(SpringLayout.WEST, roll\_no, 0, SpringLayout.WEST, stu\_name);**

**stu\_name.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**stu\_name.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(stu\_name);**

**Tname = new JTextField();**

**springLayout.putConstraint(SpringLayout.NORTH, Tname, 6, SpringLayout.SOUTH, Troll\_no);**

**springLayout.putConstraint(SpringLayout.WEST, Tname, 35, SpringLayout.EAST, stu\_name);**

**springLayout.putConstraint(SpringLayout.EAST, Tname, 0, SpringLayout.EAST, Troll\_no);**

**Tname.setColumns(10);**

**getContentPane().add(Tname);**

**//father name------------**

**JLabel father = new JLabel("Father's Name ");**

**springLayout.putConstraint(SpringLayout.WEST, father, 0, SpringLayout.WEST, roll\_no);**

**father.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**father.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(father);**

**Tfather = new JTextField();**

**springLayout.putConstraint(SpringLayout.WEST, Tfather, 33, SpringLayout.EAST, father);**

**springLayout.putConstraint(SpringLayout.EAST, Tfather, 0, SpringLayout.EAST, Troll\_no);**

**Tfather.setColumns(10);**

**getContentPane().add(Tfather);**

**//Course----------------**

**JLabel course = new JLabel("Course's Name ");**

**springLayout.putConstraint(SpringLayout.NORTH, course, 10, SpringLayout.SOUTH, father);**

**springLayout.putConstraint(SpringLayout.WEST, course, 0, SpringLayout.WEST, roll\_no);**

**course.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**course.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(course);**

**courseChoice = new Choice();**

**springLayout.putConstraint(SpringLayout.NORTH, courseChoice, 6, SpringLayout.SOUTH, Tfather);**

**springLayout.putConstraint(SpringLayout.WEST, courseChoice, 0, SpringLayout.WEST, reg\_form);**

**springLayout.putConstraint(SpringLayout.EAST, courseChoice, 0, SpringLayout.EAST, Troll\_no);**

**getContentPane().add(courseChoice);**

**courseChoice.add("Select Course");**

**courseChoice.add("CCC");**

**courseChoice.add("Tally");**

**courseChoice.add("O-Level");**

**courseChoice.add("A-Level");**

**courseChoice.add("B-Level");**

**courseChoice.add("C-Level");**

**courseChoice.add("M.Tech");**

**//mobile-------------------**

**JLabel mobile = new JLabel("Mobile Number ");**

**springLayout.putConstraint(SpringLayout.NORTH, mobile, 10, SpringLayout.SOUTH, course);**

**springLayout.putConstraint(SpringLayout.WEST, mobile, 0, SpringLayout.WEST, roll\_no);**

**mobile.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**mobile.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(mobile);**

**Tmobile = new JTextField();**

**springLayout.putConstraint(SpringLayout.WEST, Tmobile, 26, SpringLayout.EAST, mobile);**

**springLayout.putConstraint(SpringLayout.EAST, Tmobile, 0, SpringLayout.EAST, Troll\_no);**

**Tmobile.setColumns(10);**

**getContentPane().add(Tmobile);**

**//Address---------------**

**JLabel address = new JLabel("Address ");**

**springLayout.putConstraint(SpringLayout.NORTH, address, 10, SpringLayout.SOUTH, mobile);**

**springLayout.putConstraint(SpringLayout.WEST, address, 0, SpringLayout.WEST, roll\_no);**

**address.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**address.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(address);**

**Taddress = new JTextField();**

**springLayout.putConstraint(SpringLayout.NORTH, Taddress, 228, SpringLayout.NORTH, getContentPane());**

**springLayout.putConstraint(SpringLayout.SOUTH, Tmobile, -6, SpringLayout.NORTH, Taddress);**

**springLayout.putConstraint(SpringLayout.WEST, Taddress, 70, SpringLayout.EAST, address);**

**springLayout.putConstraint(SpringLayout.EAST, Taddress, 0, SpringLayout.EAST, Troll\_no);**

**Taddress.setColumns(10);**

**getContentPane().add(Taddress);**

**//State-----------------**

**JLabel state = new JLabel("State ");**

**springLayout.putConstraint(SpringLayout.NORTH, state, 10, SpringLayout.SOUTH, address);**

**springLayout.putConstraint(SpringLayout.WEST, state, 0, SpringLayout.WEST, roll\_no);**

**state.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**state.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(state);**

**//Gender---------------**

**male = new JRadioButton("Male");**

**springLayout.putConstraint(SpringLayout.NORTH, Tfather, 4, SpringLayout.SOUTH, male);**

**springLayout.putConstraint(SpringLayout.SOUTH, male, -175, SpringLayout.SOUTH, getContentPane());**

**getContentPane().add(male);**

**female = new JRadioButton("Female");**

**springLayout.putConstraint(SpringLayout.NORTH, female, 112, SpringLayout.NORTH, getContentPane());**

**springLayout.putConstraint(SpringLayout.WEST, female, 331, SpringLayout.WEST, getContentPane());**

**springLayout.putConstraint(SpringLayout.EAST, male, -61, SpringLayout.WEST, female);**

**getContentPane().add(female);**

**other = new JRadioButton("Other");**

**springLayout.putConstraint(SpringLayout.SOUTH, other, -5, SpringLayout.NORTH, Tfather);**

**springLayout.putConstraint(SpringLayout.WEST, other, 48, SpringLayout.EAST, female);**

**getContentPane().add(other);**

**bg=new ButtonGroup();**

**bg.add(male);bg.add(female);bg.add(other);**

**JLabel gender = new JLabel("Gender");**

**springLayout.putConstraint(SpringLayout.NORTH, father, 12, SpringLayout.SOUTH, gender);**

**springLayout.putConstraint(SpringLayout.NORTH, gender, 0, SpringLayout.NORTH, male);**

**springLayout.putConstraint(SpringLayout.EAST, gender, 0, SpringLayout.EAST, roll\_no);**

**gender.setHorizontalAlignment(SwingConstants.CENTER);**

**gender.setFont(new Font("Tahoma", Font.PLAIN, 13));**

**getContentPane().add(gender);**

**//state choice**

**chooseState = new Choice();**

**springLayout.putConstraint(SpringLayout.NORTH, chooseState, 3, SpringLayout.SOUTH, Taddress);**

**springLayout.putConstraint(SpringLayout.WEST, chooseState, 0, SpringLayout.WEST, reg\_form);**

**springLayout.putConstraint(SpringLayout.EAST, chooseState, 0, SpringLayout.EAST, Troll\_no);**

**getContentPane().add(chooseState);**

**chooseState.add("Select State");chooseState.add("Uttar Pradesh");chooseState.add("Andhra Pradesh");chooseState.add("Assam");chooseState.add("Goa");**

**chooseState.add("Gujarat");chooseState.add("Hariyana");**

**//Submit button**

**register = new JButton("Register");**

**springLayout.putConstraint(SpringLayout.SOUTH, register, -10, SpringLayout.SOUTH, getContentPane());**

**springLayout.putConstraint(SpringLayout.EAST, register, -233, SpringLayout.EAST, getContentPane());**

**getContentPane().add(register);**

**//Action for register button**

**register.addActionListener(this);**

**}**

**public static void main(String arg[]) {**

**new new\_student().setVisible(true);;**

**}**

**// Tname;**

**// private JTextField Tfather;**

**// private JTextField Tcourse;**

**// private JTextField Tmobile;**

**// private JTextField Taddress;**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**if(e.getSource() == register) {**

**Integer rollNumber=dbRollno;**

**String name=Tname.getText().toLowerCase();**

**String gender=null;**

**if(male.isSelected()) {**

**gender="male";**

**}else if(female.isSelected()) {**

**gender="female";**

**}else if(other.isSelected()) {**

**gender="other";**

**}**

**String father=Tfather.getText().toLowerCase();**

**String course=courseChoice.getSelectedItem();**

**String mobile=Tmobile.getText();**

**String address=Taddress.getText().toLowerCase();**

**String state=chooseState.getSelectedItem();**

**//Check null values-----**

**if(name.isEmpty() || gender.isEmpty() || father.isEmpty() || course.equals("Select Course") || mobile.isEmpty() || address.isEmpty() || state.equals("Select State")) {**

**JOptionPane.showMessageDialog(null, "Opps ! all field required....");**

**}else {**

**connection conObj=new connection();**

**try {**

**String sql="insert into `student\_info` (`stu\_name`,`gender`,`father`,`course\_name`,`mobile`,`address`,`state`) values ('"+name+"', '"+gender+"', '"+father+"', '"+course.toLowerCase()+"', '"+mobile+"', '"+address+"', '"+state.toLowerCase()+"')";**

**conObj.stm.executeUpdate(sql);**

**JOptionPane.showMessageDialog(null, "User Registration successfull");**

**}catch(Exception exc) {**

**exc.printStackTrace();**

**}**

**//After inserting data, clearing form**

**rollNumber+=1;**

**String newRollNo=rollNumber.toString();**

**Troll\_no.removeAll();**

**Troll\_no.add(newRollNo);**

**Tname.setText(null);Tfather.setText(null);courseChoice.select("Select Course");Tmobile.setText(null);Taddress.setText(null);chooseState.select("Select State");**

**}**

**}**

**}**

**}**

**Registration page:**

**package attendanceSystem;**

**import java.awt.EventQueue;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.ResultSet;**

**import javax.swing.ButtonGroup;**

**import javax.swing.JButton;**

**import javax.swing.JComboBox;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.JTextField;**

**import javax.swing.JRadioButton;**

**import java.awt.Font;**

**import javax.swing.ImageIcon;**

**import java.awt.Color;**

**public class Registration extends JFrame implements ActionListener{**

**/\*\***

**\* Launch the application.**

**\*/**

**public static void main(String[] args) {**

**EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**try {**

**Registration window = new Registration();**

**window.setVisible(true);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**});**

**}**

**/\*\***

**\* Create the application.**

**\*/**

**public Registration() {**

**initialize();**

**}**

**private JTextField userNameField,nameField,emailField,passField,confirmPassField;**

**private JButton button;**

**private JTextField mobileField;**

**private JTextField addressField;**

**private JRadioButton male,female;**

**private JButton cancel;**

**private void initialize() {**

**setTitle("Registration Form");**

**setBounds(100, 100, 450, 350);**

**setResizable(false);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**JLabel usrNameLabel = new JLabel("User Name ");**

**JLabel nameLabel = new JLabel("Name ");**

**JLabel emailLabel = new JLabel("Email ");**

**JLabel addressLabel = new JLabel("Address");**

**JLabel lblMobileNo = new JLabel("Mobile No. ");**

**JLabel passLabel = new JLabel("Password ");**

**JLabel confirmPassLabel = new JLabel("Re-type Password ");**

**userNameField = new JTextField();**

**nameField = new JTextField();**

**emailField = new JTextField();**

**passField = new JTextField();**

**confirmPassField = new JTextField();**

**usrNameLabel.setBounds(63, 40, 72, 20);**

**userNameField.setBounds(184, 40, 233, 20);**

**nameLabel.setBounds(63, 80, 77, 20);**

**nameField.setBounds(183, 80, 234, 20);**

**emailLabel.setBounds(63, 159, 64, 20);**

**emailField.setBounds(184, 159, 235, 20);**

**passLabel.setBounds(62, 290, 66, 20);**

**passField.setBounds(184, 290, 233, 20);**

**confirmPassLabel.setBounds(63, 330, 108, 20);**

**confirmPassField.setBounds(184, 330, 236, 20);**

**lblMobileNo.setBounds(63, 248, 65, 20);**

**getContentPane().add(lblMobileNo);**

**mobileField = new JTextField();**

**mobileField.setBounds(184, 248, 233, 20);**

**getContentPane().add(mobileField);**

**addressLabel.setBounds(63, 203, 65, 20);**

**getContentPane().add(addressLabel);**

**setBounds(450, 150, 500, 450);**

**addressField = new JTextField();**

**addressField.setBounds(184, 203, 233, 20);**

**getContentPane().add(addressField);**

**JLabel genderLabel = new JLabel("Gender");**

**genderLabel.setBounds(63, 121, 72, 20);**

**getContentPane().add(genderLabel);**

**ButtonGroup bg=new ButtonGroup();**

**male = new JRadioButton("Male");**

**male.setBounds(184, 120, 109, 23);**

**getContentPane().add(male);**

**female = new JRadioButton("Female");**

**female.setBounds(295, 120, 109, 23);**

**getContentPane().add(female);**

**bg.add(male);bg.add(female);**

**button = new JButton("Ragister");**

**button.setForeground(Color.BLUE);**

**button.setFont(new Font("Tahoma", Font.BOLD, 11));**

**button.setBounds(118, 376, 127 , 23);**

**button.addActionListener(this);**

**getContentPane().add(usrNameLabel);getContentPane().add(userNameField);**

**getContentPane().add(nameLabel);getContentPane().add(nameField);**

**getContentPane().add(emailLabel);getContentPane().add(emailField);**

**getContentPane().add(passLabel);getContentPane().add(passField);**

**getContentPane().add(confirmPassLabel);getContentPane().add(confirmPassField);**

**getContentPane().add(button);**

**getContentPane().setLayout(null);**

**cancel = new JButton("Cancel");**

**cancel.setForeground(Color.RED);**

**cancel.setFont(new Font("Tahoma", Font.BOLD, 11));**

**cancel.setBounds(256, 376, 127, 23);**

**getContentPane().add(cancel);**

**cancel.addActionListener(this);**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**if(e.getSource().equals(button)) {**

**String usrName = userNameField.getText();**

**String name = nameField.getText();**

**String email = emailField.getText();**

**String address = addressField.getText();**

**String mobile= mobileField.getText();**

**String pass = passField.getText();**

**String confirmPass = confirmPassField.getText();**

**String gender="";**

**if(usrName.isEmpty() || name.isEmpty() || email.isEmpty() || address.isEmpty() || mobile.isEmpty() || pass.isEmpty() || confirmPass.isEmpty()) {**

**JOptionPane.showMessageDialog(null, "All Field required", "Warrning ", JOptionPane.WARNING\_MESSAGE);**

**}else{**

**if(male.isSelected() || female.isSelected()) {**

**if(!pass.equals(confirmPass)) {**

**JOptionPane.showMessageDialog(null, "password should be same", "Warrning ", JOptionPane.WARNING\_MESSAGE);**

**}else {**

**connection conn=new connection();**

**String sql="select `usr\_name` from `login\_info` where `usr\_name`='"+usrName+"'";**

**try {**

**ResultSet data=conn.stm.executeQuery(sql);**

**if(data.next()) {**

**JOptionPane.showMessageDialog(null,"Opps ! this username already exits", "Warrning ", JOptionPane.WARNING\_MESSAGE);**

**}else {**

**if(male.isSelected()) {**

**gender="male";**

**}else {**

**gender="female";**

**}**

**String sql2="insert into `login\_info` values ('"+usrName+"','"+name+"','"+gender+"','"+email+"','"+address+"','"+mobile+"','"+pass+"')";**

**conn.stm.executeUpdate(sql2);**

**JOptionPane.showMessageDialog(null, "Congrats ! you are register successfull..", "Registerd", JOptionPane.PLAIN\_MESSAGE);**

**this.setVisible(false);**

**new mainPage().setVisible(true);**

**}**

**}catch(Exception ex) {**

**ex.printStackTrace();**

**}**

**}**

**}else {**

**JOptionPane.showMessageDialog(null, "please Select your gender", "Warrning ", JOptionPane.WARNING\_MESSAGE);**

**}**

**}**

**}else if(e.getSource().equals(cancel)) {**

**this.setVisible(false);**

**new mainPage().setVisible(true);**

**}**

**}**

**}**

**Student details:**

**package attendanceSystem;**

**import javax.swing.\*;**

**/\***

**\*@author Satya Narayan Mishra**

**\*/**

**import java.awt.EventQueue;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.ResultSet;**

**import java.sql.SQLException;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import java.awt.BorderLayout;**

**import javax.swing.SwingConstants;**

**import com.mysql.cj.protocol.Resultset;**

**import java.awt.Font;**

**import java.awt.Color;**

**import java.awt.Panel;**

**import javax.swing.JScrollPane;**

**import javax.swing.JTable;**

**import javax.swing.JTextField;**

**import java.awt.Button;**

**import javax.swing.JButton;**

**import java.awt.ScrollPane;**

**import java.awt.event.KeyAdapter;**

**import java.awt.event.KeyEvent;**

**public class student\_details extends JFrame implements ActionListener{**

**private JFrame frame;**

**/\*\***

**\* Launch the application.**

**\*/**

**public static void main(String[] args) {**

**EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**try {**

**student\_details window = new student\_details();**

**window.setVisible(true);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**});**

**}**

**/\*\***

**\* Create the application.**

**\*/**

**public student\_details() {**

**initialize();**

**}**

**/\*\***

**\* Initialize the contents of the frame.**

**\*/**

**private int i=0;**

**private JTable table;**

**private JButton btnClose,btnUpdate,btnDelete;**

**private JTextField Registration;**

**private JTextField Mobile;**

**private JTextField Course;**

**private JTextField FatherName;**

**private JTextField Gender;**

**private JTextField StudentName;**

**private JTextField RollNo;**

**private JTextField Address;**

**private void initialize() {**

**setBounds(40, 80, 1300, 610);**

**JLabel header = new JLabel("Student Management & Details");**

**header.setForeground(new Color(139, 0, 0));**

**header.setFont(new Font("Cambria Math", Font.BOLD, 20));**

**header.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(header, BorderLayout.NORTH);**

**connection conn=new connection();**

**String sql="select \* from `student\_info`";**

**try {**

**ResultSet rslt=conn.stm.executeQuery(sql);**

**if(rslt.next()) {**

**// If data available**

**Panel panel = new Panel();**

**getContentPane().add(panel, BorderLayout.CENTER);**

**panel.setLayout(null);**

**Panel panel\_1 = new Panel();**

**panel\_1.setBounds(10, 11, 496, 506);**

**panel.add(panel\_1);**

**panel\_1.setLayout(null);**

**btnUpdate = new JButton("Update");**

**btnUpdate.setBounds(84, 448, 89, 23);**

**panel\_1.add(btnUpdate);**

**btnUpdate.addActionListener(this);**

**btnDelete = new JButton("Delete");**

**btnDelete.setBounds(210, 448, 89, 23);**

**panel\_1.add(btnDelete);**

**btnDelete.addActionListener(this);**

**btnClose = new JButton("Close");**

**btnClose.setBounds(335, 448, 89, 23);**

**panel\_1.add(btnClose);**

**Registration = new JTextField();**

**Registration.setText("");**

**Registration.setColumns(10);**

**Registration.setBounds(201, 396, 223, 20);**

**panel\_1.add(Registration);**

**JLabel lblRegistrationDate = new JLabel("Registration Date");**

**lblRegistrationDate.setBounds(84, 399, 91, 14);**

**panel\_1.add(lblRegistrationDate);**

**JLabel lblMobileNo = new JLabel("Mobile No.");**

**lblMobileNo.setBounds(84, 312, 91, 14);**

**panel\_1.add(lblMobileNo);**

**Mobile = new JTextField();**

**Mobile.setText("");**

**Mobile.setColumns(10);**

**Mobile.setBounds(201, 309, 223, 20);**

**panel\_1.add(Mobile);**

**Course = new JTextField();**

**Course.setText("");**

**Course.setColumns(10);**

**Course.setBounds(201, 265, 223, 20);**

**panel\_1.add(Course);**

**JLabel lblCourse = new JLabel("Course");**

**lblCourse.setBounds(84, 268, 91, 14);**

**panel\_1.add(lblCourse);**

**JLabel lblFathersName\_1 = new JLabel("Father's Name");**

**lblFathersName\_1.setBounds(84, 223, 91, 14);**

**panel\_1.add(lblFathersName\_1);**

**FatherName = new JTextField();**

**FatherName.setText("");**

**FatherName.setColumns(10);**

**FatherName.setBounds(201, 220, 223, 20);**

**panel\_1.add(FatherName);**

**Gender = new JTextField();**

**Gender.setText("");**

**Gender.setColumns(10);**

**Gender.setBounds(201, 177, 223, 20);**

**panel\_1.add(Gender);**

**JLabel lblFathersName = new JLabel("Gender");**

**lblFathersName.setBounds(84, 180, 91, 14);**

**panel\_1.add(lblFathersName);**

**JLabel lblStudentName = new JLabel("Student Name");**

**lblStudentName.setBounds(84, 136, 91, 14);**

**panel\_1.add(lblStudentName);**

**StudentName = new JTextField();**

**StudentName.setText("");**

**StudentName.setColumns(10);**

**StudentName.setBounds(201, 133, 223, 20);**

**panel\_1.add(StudentName);**

**RollNo = new JTextField();**

**RollNo.setText("");**

**RollNo.setColumns(10);**

**RollNo.setBounds(201, 86, 223, 20);**

**panel\_1.add(RollNo);**

**RollNo.addKeyListener(new KeyAdapter() {**

**@Override**

**public void keyReleased(KeyEvent e) {**

**String rollNo=RollNo.getText().strip();**

**Integer intRollNo=Integer.parseInt(rollNo);**

**String Query="select \* from `student\_info` where `roll\_no`="+intRollNo;**

**try {**

**ResultSet quickData=conn.stm.executeQuery(Query);**

**if(quickData.next()) {**

**StudentName.setText(quickData.getString("stu\_name"));**

**Gender.setText(quickData.getString("gender"));**

**FatherName.setText(quickData.getString("father"));**

**Course.setText(quickData.getString("course\_name"));**

**Mobile.setText(quickData.getString("mobile"));**

**Address.setText(quickData.getString("address"));**

**Registration.setText(quickData.getString("regi\_date"));**

**}else {**

**StudentName.setText("");**

**Gender.setText("");**

**FatherName.setText("");**

**Course.setText("");**

**Mobile.setText("");**

**Address.setText("");**

**Registration.setText("");**

**}**

**} catch (SQLException e1) {**

**// TODO Auto-generated catch block**

**e1.printStackTrace();**

**}**

**}**

**});**

**JLabel lblRollNo = new JLabel("Roll No.");**

**lblRollNo.setBounds(84, 89, 91, 14);**

**panel\_1.add(lblRollNo);**

**Address = new JTextField();**

**Address.setText("");**

**Address.setColumns(10);**

**Address.setBounds(201, 353, 223, 20);**

**panel\_1.add(Address);**

**JLabel lblAddress = new JLabel("Address");**

**lblAddress.setBounds(84, 356, 91, 14);**

**panel\_1.add(lblAddress);**

**btnClose.addActionListener(this);**

**Panel panel\_2 = new Panel();**

**panel\_2.setBounds(512, 22, 762, 495);**

**panel.add(panel\_2);**

**panel\_2.setLayout(null);**

**try {**

**String query="select \* from `student\_info`";**

**ResultSet data1=conn.stm.executeQuery(query);**

**int counter=0,i=0,j=0;**

**while(data1.next()) {**

**counter+=1;**

**}**

**String x[]= {"Roll No","Name","Gender","Father","Course","Mobile","Registration Date"};**

**String y[][]=new String[counter][8];**

**ResultSet data=conn.stm.executeQuery(query);**

**while(data.next()) {**

**y[i][j++]=data.getString("roll\_no");**

**y[i][j++]=data.getString("stu\_name");**

**y[i][j++]=data.getString("gender");**

**y[i][j++]=data.getString("father");**

**y[i][j++]=data.getString("course\_name");**

**y[i][j++]=data.getString("mobile");**

**y[i][j++]=data.getString("regi\_date");**

**i++;**

**j=0;**

**}**

**table = new JTable(y,x);**

**table.setBounds(10, 11, 403, 18);**

**JScrollPane titleSP=new JScrollPane(table);**

**panel\_2.add(titleSP);**

**titleSP.setBounds(10, 24, 742, 432);**

**} catch (SQLException e1) {**

**// TODO Auto-generated catch block**

**e1.printStackTrace();**

**}**

**}else {**

**JLabel lblNewLabel = new JLabel("Opps ! There are no any record found...");**

**lblNewLabel.setForeground(new Color(255, 0, 0));**

**lblNewLabel.setBackground(new Color(255, 0, 0));**

**lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 15));**

**lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);**

**getContentPane().add(lblNewLabel, BorderLayout.CENTER);**

**}**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**if(e.getSource().equals(btnClose)) {**

**this.setVisible(false);**

**}else {**

**String rollNo=new String(RollNo.getText().strip().toLowerCase());**

**Integer intRoll=Integer.parseInt(rollNo);**

**String name=StudentName.getText().strip().toLowerCase();**

**String gender=Gender.getText().strip().toLowerCase();**

**String father=FatherName.getText().strip().toLowerCase();**

**String course\_name=Course.getText().strip().toLowerCase();**

**String mobile=Mobile.getText().strip().toLowerCase();**

**String address=Address.getText().strip().toLowerCase();**

**if(rollNo.isEmpty() || name.isEmpty() || gender.isEmpty() || father.isEmpty() || course\_name.isEmpty() || mobile.isEmpty() || address.isEmpty()) {**

**JOptionPane.showMessageDialog(null, "Opps ! all field required..", "Warning", JOptionPane.WARNING\_MESSAGE);**

**}else {**

**connection conn=new connection();**

**if(e.getSource().equals(btnUpdate)) {**

**String sql="update `student\_info` set `stu\_name`='"+name+"', `gender`='"+gender+"', `father`='"+father+"', `course\_name`='"+course\_name+"',`mobile`='"+mobile+"', `address`='"+address+"' where `roll\_no`="+intRoll;**

**try {**

**conn.stm.executeUpdate(sql);**

**JOptionPane.showMessageDialog(null, "Record Updated successfull...", "Success", JOptionPane.WARNING\_MESSAGE);**

**this.hide();**

**new student\_details().setVisible(true);**

**} catch (SQLException e1) {**

**// TODO Auto-generated catch block**

**e1.printStackTrace();**

**}**

**}else if(e.getSource().equals(btnDelete)) {**

**int val=JOptionPane.showConfirmDialog(null, "After clicking you lose all record from database \n Do you want to delete this record?", "Conformation", JOptionPane.YES\_NO\_OPTION);**

**if(val==0) {**

**// clicked on yes button**

**String sql1="delete from `attendance\_info` where `roll\_no`="+intRoll;**

**String sql2="delete from `student\_info` where `roll\_no`="+intRoll;**

**try {**

**conn.stm.executeUpdate(sql1);**

**conn.stm.executeUpdate(sql2);**

**JOptionPane.showMessageDialog(null, "Record Deleted successfull");**

**this.hide();**

**new student\_details().setVisible(true);**

**} catch (SQLException e1) {**

**// TODO Auto-generated catch block**

**e1.printStackTrace();**

**}**

**}else {**

**// clicked on no button**

**}**

**}**

**}**

**}**

**}**

**}**

**Attendance:**

**package attendanceSystem;**

**import java.awt.EventQueue;**

**import javax.swing.JFrame;**

**import com.jgoodies.forms.layout.FormLayout;**

**import com.jgoodies.forms.layout.ColumnSpec;**

**import com.jgoodies.forms.layout.RowSpec;**

**import java.awt.GridBagLayout;**

**import javax.swing.GroupLayout;**

**import javax.swing.GroupLayout.Alignment;**

**import javax.swing.BoxLayout;**

**import java.awt.GridLayout;**

**import com.toedter.components.JSpinField;**

**import java.awt.GridBagConstraints;**

**import javax.swing.JScrollBar;**

**import javax.swing.JScrollPane;**

**import javax.swing.JDesktopPane;**

**import java.awt.Insets;**

**import javax.swing.JTabbedPane;**

**import javax.swing.JInternalFrame;**

**import javax.swing.JRadioButton;**

**import java.awt.FlowLayout;**

**import net.miginfocom.swing.MigLayout;**

**import java.awt.Panel;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.sql.ResultSet;**

**import javax.swing.JLabel;**

**import javax.swing.JOptionPane;**

**import javax.swing.SwingConstants;**

**import java.awt.Font;**

**import java.awt.Color;**

**import javax.swing.JCheckBox;**

**import javax.swing.JTable;**

**import javax.swing.JPanel;**

**import javax.swing.JButton;**

**import java.awt.CardLayout;**

**/\***

**\*@author Amisha Sharma**

**\*/**

**public class Attendance extends JFrame implements ActionListener {**

**/\*\***

**\***

**\*/**

**private static final long serialVersionUID = 1L;**

**private JTable table;**

**private JButton done;**

**private JCheckBox cb1[];**

**private JLabel numOfdata,selectedDate;**

**private JLabel LabelRollNo[];**

**/\*\***

**\* Launch the application.**

**\*/**

**public static void main(String[] args) {**

**int month=0,date=0,year = 0,counter=0;**

**String course = null;**

**new Attendance( date,month,year,course,counter).setVisible(true);**

**}**

**/\*\***

**\* Application start**

**\* @param date**

**\* @param month**

**\* @param year**

**\* @param course**

**\* @param counter**

**\*/**

**Attendance(Integer date,Integer month,Integer year,String course,Integer counter) {**

**setBounds(400, 130, 700, 530);**

**getContentPane().setLayout(null);**

**//Data container---------------------------**

**JScrollPane sp=new JScrollPane();**

**JPanel panel = new JPanel();**

**sp.setViewportView(panel);**

**//Title---------------------------**

**JLabel title = new JLabel("Attendance Date : "+date.toString()+"-"+month+"-"+year+" Course : "+course.toUpperCase());**

**title.setForeground(new Color(128, 0, 0));**

**title.setFont(new Font("Cambria Math", Font.PLAIN, 20));**

**title.setHorizontalAlignment(SwingConstants.CENTER);**

**title.setBounds(10, 11, 664, 33);**

**getContentPane().add(title);**

**// fetching data from database and setting the data into class**

**myDb dbObj=new myDb();**

**dbObj.setData(date, month, year, course, counter);**

**// this is dummy JLabel for accessing on Action Listening time**

**numOfdata=new JLabel(counter.toString());**

**selectedDate=new JLabel(date+"-"+month+"-"+year);**

**String n[][]=new String[1][4];**

**String x[]= {"A/P","S.N","Roll.NO","Student's Name"};**

**table = new JTable(n,x);**

**JScrollPane titleSP=new JScrollPane(table);**

**getContentPane().add(titleSP);**

**titleSP.setBounds(102, 53, 508, 25);**

**//inserting dynamically checkbox**

**cb1=new JCheckBox[counter];**

**LabelRollNo=new JLabel[counter];**

**int boundsY=0;**

**for(int i=0,k=0;i<counter;i++) {**

**//LabelRollNo is also dummy label for accessing on Action Listening time**

**connection conn=new connection();**

**String sql="select \* from `attendance\_info` where `roll\_no`="+dbObj.DbData[i][k+1]+" and `date`="+date+" and `month`="+month+" and `year`="+year+" and `iss\_present`="+1;**

**try {**

**ResultSet attendaceData=conn.stm.executeQuery(sql);**

**if(attendaceData.next()) {**

**cb1[i] = new JCheckBox(" "+dbObj.DbData[i][k]+" "+dbObj.DbData[i][k+1]+" "+dbObj.DbData[i][k+2].toUpperCase(),true);**

**}else {**

**cb1[i] = new JCheckBox(" "+dbObj.DbData[i][k]+" "+dbObj.DbData[i][k+1]+" "+dbObj.DbData[i][k+2].toUpperCase());**

**}**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**//setting checkbox bounds dynamically======================**

**LabelRollNo[i]=new JLabel(dbObj.DbData[i][k+1]);**

**cb1[i].setBounds(50, boundsY, 508, 23);**

**panel.add(cb1[i]);**

**boundsY+=25;**

**}**

**//Setting Layout Dynamically------------**

**GroupLayout gl\_panel = new GroupLayout(panel);**

**gl\_panel.setHorizontalGroup(**

**gl\_panel.createParallelGroup(Alignment.LEADING)**

**.addGap(0, 490, Short.MAX\_VALUE)**

**);**

**gl\_panel.setVerticalGroup(**

**gl\_panel.createParallelGroup(Alignment.LEADING)**

**.addGap(0, boundsY, Short.MAX\_VALUE)**

**);**

**panel.setLayout(gl\_panel);**

**sp.setBounds(102, 75, 508, 355);**

**getContentPane().add(sp);**

**//Done button-----------------------------------------**

**done = new JButton("Attendance Done");**

**done.setFont(new Font("Cambria Math", Font.PLAIN, 15));**

**done.setForeground(new Color(0, 0, 0));**

**done.setBounds(102, 429, 508, 33);**

**getContentPane().add(done);**

**done.addActionListener(this);**

**for(int i=0;i<counter;i++) {**

**cb1[i].addActionListener(this);**

**}**

**}**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**if(e.getSource()==done) {**

**this.hide();**

**}else {**

**String strNumOfData=numOfdata.getText();**

**int intNumOfData=Integer.parseInt(strNumOfData);**

**String strDate=selectedDate.getText();**

**String arrDate[]=strDate.split("-");**

**int intDate=Integer.parseInt(arrDate[0]);**

**int intMonth=Integer.parseInt(arrDate[1]);**

**int intYear=Integer.parseInt(arrDate[2]);**

**for(int i=0;i<intNumOfData;i++) {**

**String strRollNo=LabelRollNo[i].getText();**

**int intRollNo=Integer.parseInt(strRollNo);**

**if(e.getSource()==cb1[i]) {**

**connection conn=new connection();**

**try {**

**String sql="select \* from `attendance\_info` where `roll\_no`="+intRollNo+" and `date`="+intDate+" and `month`="+intMonth+" and `year`="+intYear+" and `iss\_present`="+1;**

**ResultSet dbDataset=conn.stm.executeQuery(sql);**

**if(dbDataset.next()) {**

**//when something found in the database then---------------------**

**if(cb1[i].isSelected()) {**

**//when check box selected**

**String updateSql="update `attendance\_info` set `iss\_present`="+1+" where `roll\_no`="+intRollNo+" and `date`="+intDate+" and `month`="+intMonth+" and `year`="+intYear;**

**conn.stm.executeUpdate(updateSql);**

**}else {**

**//when check box unselected**

**String updateSql="update `attendance\_info` set `iss\_present`="+0+" where `roll\_no`="+intRollNo+" and `date`="+intDate+" and `month`="+intMonth+" and `year`="+intYear;**

**conn.stm.executeUpdate(updateSql);**

**}**

**}else{**

**String sql2="select \* from `attendance\_info` where `roll\_no`="+intRollNo+" and `date`="+intDate+" and `month`="+intMonth+" and `year`="+intYear+" and `iss\_present`="+0;**

**ResultSet dbDataset2=conn.stm.executeQuery(sql2);**

**if(dbDataset2.next()) {**

**//when something found in the database then---------------------**

**if(cb1[i].isSelected()) {**

**//when check box selected**

**String updateSql="update `attendance\_info` set `iss\_present`="+1+" where `roll\_no`="+intRollNo+" and `date`="+intDate+" and `month`="+intMonth+" and `year`="+intYear;**

**conn.stm.executeUpdate(updateSql);**

**}else {**

**//when check box unselected**

**String updateSql="update `attendance\_info` set `iss\_present`="+0+" where `roll\_no`="+intRollNo+" and `date`="+intDate+" and `month`="+intMonth+" and `year`="+intYear;**

**conn.stm.executeUpdate(updateSql);**

**}**

**}else {**

**//when nothing found in the database then------------------------**

**if(cb1[i].isSelected()) {**

**//when check box selected**

**String updateSql="insert into `attendance\_info` (`roll\_no`,`date`,`month`,`year`,`iss\_present`) values("+intRollNo+","+intDate+","+intMonth+","+intYear+","+1+")";**

**conn.stm.executeUpdate(updateSql);**

**}else {**

**//when check box unselected**

**String updateSql="insert into `attendance\_info` (`roll\_no`,`date`,`month`,`year`,`iss\_present`) values("+intRollNo+","+intDate+","+intMonth+","+intYear+","+0+")";**

**conn.stm.executeUpdate(updateSql);**

**}**

**}**

**}**

**}catch(Exception ex) {**

**ex.printStackTrace();**

**}**

**}**

**}**

**}**

**}**

**}**

**// for getting data from database**

**class myDb{**

**Integer i=0,j=0,k;**

**// String x[]= {"Attendance","S.N","Roll.NO","Student's Name","Father's Name"};**

**String DbData[][];**

**void setData(Integer date,Integer month, Integer year,String course,int counter) {**

**String todayDate=date.toString()+"-"+month.toString()+"-"+year.toString();**

**String y[][]=new String[counter][4];**

**connection obj=new connection();**

**String sql="select \* from `student\_info` where `course\_name`='"+course+"'";**

**try {**

**ResultSet dbResult=obj.stm.executeQuery(sql);**

**while(dbResult.next()) {**

**k=i+1;**

**y[i][j++]=k.toString();**

**y[i][j++]=dbResult.getString("roll\_no");**

**y[i][j++]=dbResult.getString("stu\_name");**

**y[i][j++]=dbResult.getString("father");**

**i++;**

**j=0;**

**}**

**setDbData(y);**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**}**

**void setDbData(String y[][]) {**

**DbData=y;**

**}**

**}**

**Connection:**

**package** attendanceSystem;

**import** java.sql.\*;

/\*

\* @author Satya Narayan Mishra

\*/

**public** **class** connection{

Connection cn;

Statement stm;

connection(){

**try** {

Class.*forName*("com.mysql.cj.jdbc.Driver");

cn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/attendance\_system", "root", "");

stm = cn.createStatement();

}**catch**(Exception e) {

e.printStackTrace();

}

}

**public** **static** **void** main(String arg[]) {

**new** connection();

}

}

**11. TESTING**

**12.1 Functional Testing**

Functional testing is a type of black box testing that bases its test cases on the specifications of the software component under test.

Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (Not like in white-box testing). Functional testing differs from system testing in that functional testing a program by checking it against ... design document or specification", while system testing "validate a program by checking it against the published user or system requirements.

**Functional testing typically involves five steps:**

1. The identification of functions that the software is expected to perform

2. The creation of input data based on the function's specifications

3. The determination of output based on the function's specifications

4. The execution of the test case

5. The comparison of actual and expected outputs

**12.2 System testing:**

Software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

**12.3 Structural Testing :**

It determines the durability and integrity of complete structures or sub-assemblies, usually in multi-axis testing systems that replicate end use conditions. The product range extends from single-axis component testing through to complex testing systems for the simulation of almost all service loads affecting a vehicle or component.

IST offers a wide range of modular standard testing systems for use in the automotive industry to test car components such as suspensions and steering systems. Alternatively, systems can be engineered to meet specific customer requirements.

**12.4 Unit Testing:** individual software components of application are tested in isolation from other part of the program.

**12.5 Integration Testing**:

* **Big-ban testing:** individual software components of an application are combined at once into system .every module is first init tested. Then the entire system is tested for communication interfaces between them.
* **Bottom-up testing:** in bottom-up integration testing begins from bottom of the module hierarchy and work up to the top to simulate higher level modules .every module is first unit tested then modules are added in ascending hierarchical order. Lower level modules are tested first then the next set of higher level modules is tested with previously tested lower level modules.
* **Top-down testing:** begins testing from top of the module hierarchy and work down to the bottom to simulate lower interfacing modules. Every module is first unit tested then the modules are added in descending hierarchical order. Higher level modules are tested first then the next set of lower level modules is tested with previously tested higher level modules.

**Testing of the Project**

This is to find if any errors are present in the system. To check for the errors an artificial made database are fed to system and been checked and the errors will be verified accordingly. **Two main ways of data verifications are:**

**Proofreading:** Proofreading (also proof-reading) traditionally means reading a proof copy of a text in order to detect and correct any errors. Modern proofreading often requires reading copy at earlier stages as well.

**Double Entry:** In the double entry system transactions are entered twice in the accounts. For example, the same car purchased will result an increase in the "vehicle" account and a decrease in "cash" account. Therefore, the difference between the two is that in single entry system, transaction is entered only once and in double entry system it is entered twice.

Typing the data twice and comparing the two database at the same time. This method is also known as double entry method.

In this phase, I had tried to check all the modules separately for there proper formatting. After this step, I had performed a unit test to check the functionality of the whole system. Further, I had come to know to add certain validation in project as given bellow:

**1)** Table name- student information

Field name -student name

Data type -Text.

When we enter number in the form then it show wrong .Because it is not number type it is a character. So it shows wrong value.

**2)** Table name \_Course name

Field name- course name

Data type- text.

When we enter number in the form then it show wrong .Because it is not number type it is a character. So it shows wrong value

3) Table name –teacher information

Field name- login-code

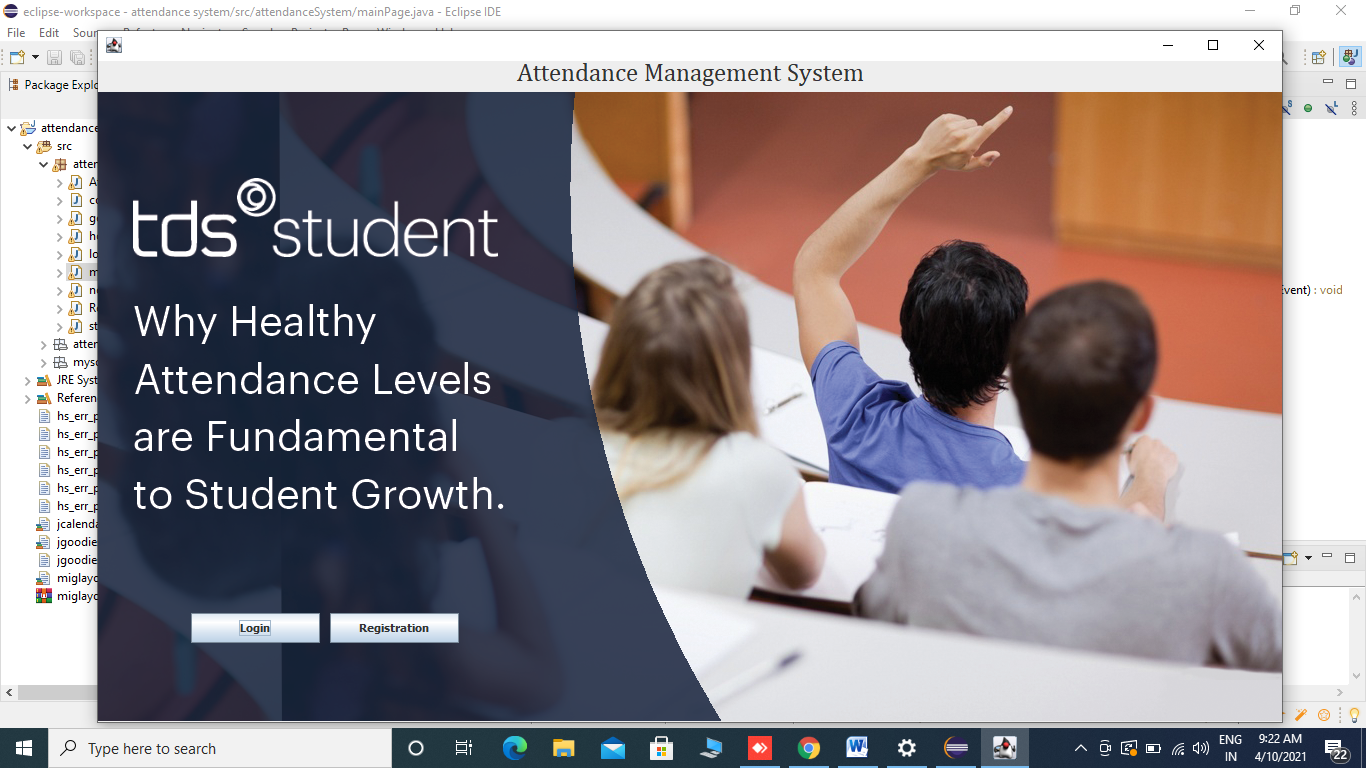
Data type- text

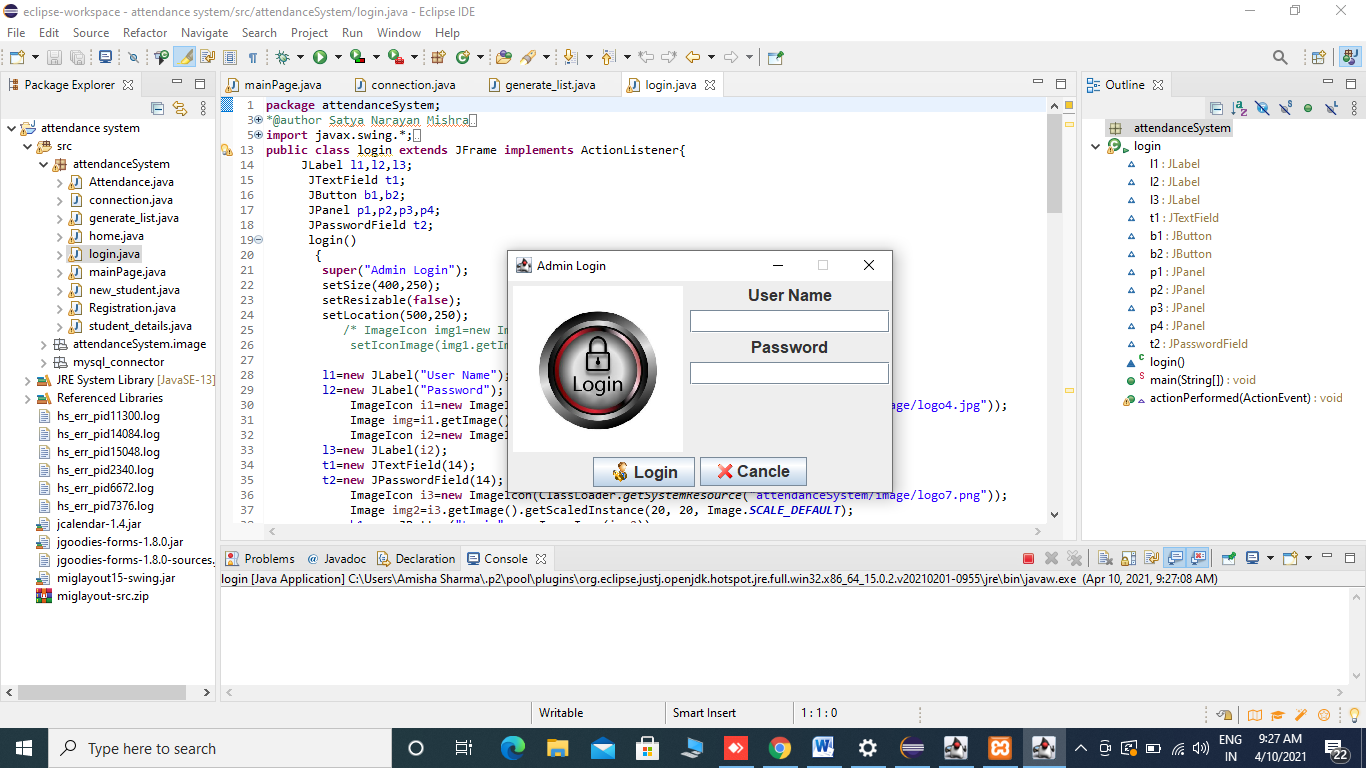
4)Table name-login,datatype number

When we put any other value or character then it ask validity check.

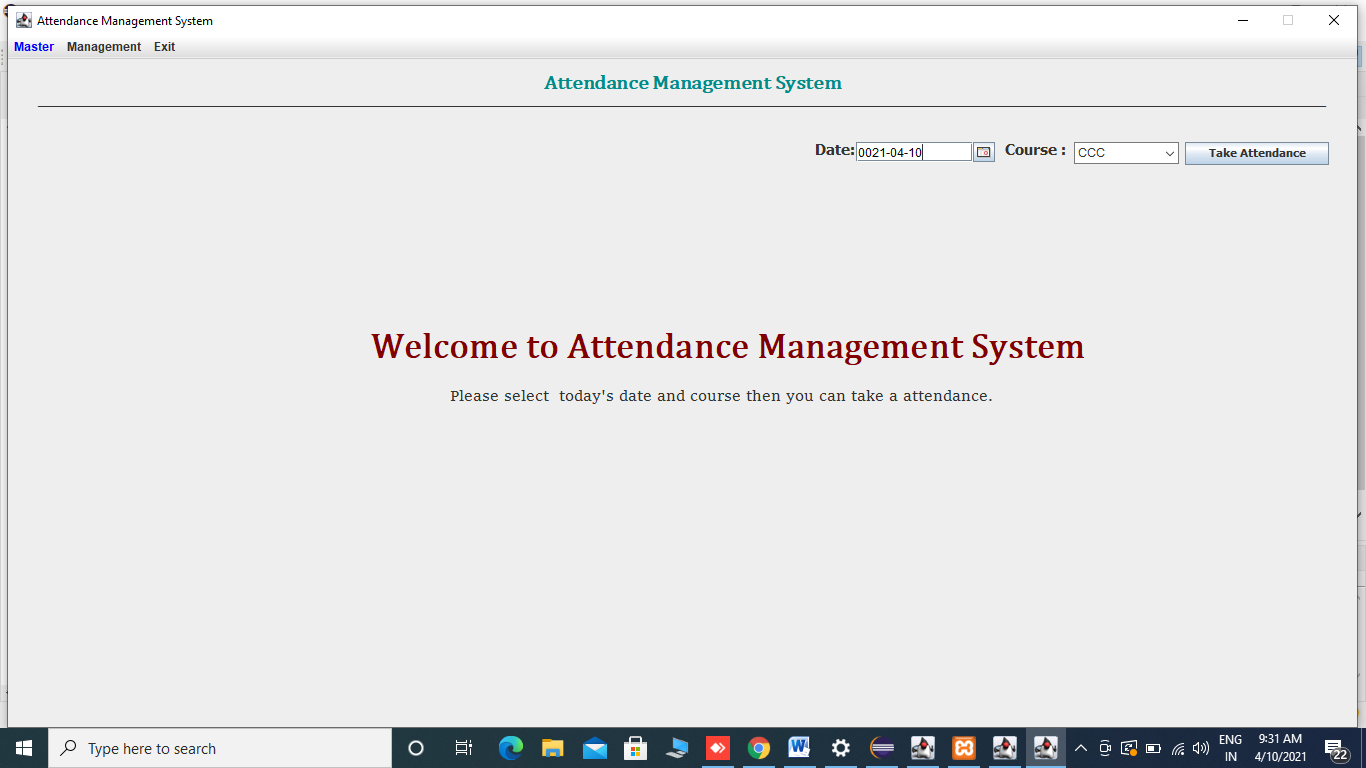
**12. SCREENSHOT**

**Main page of the project rate of Economic class:**

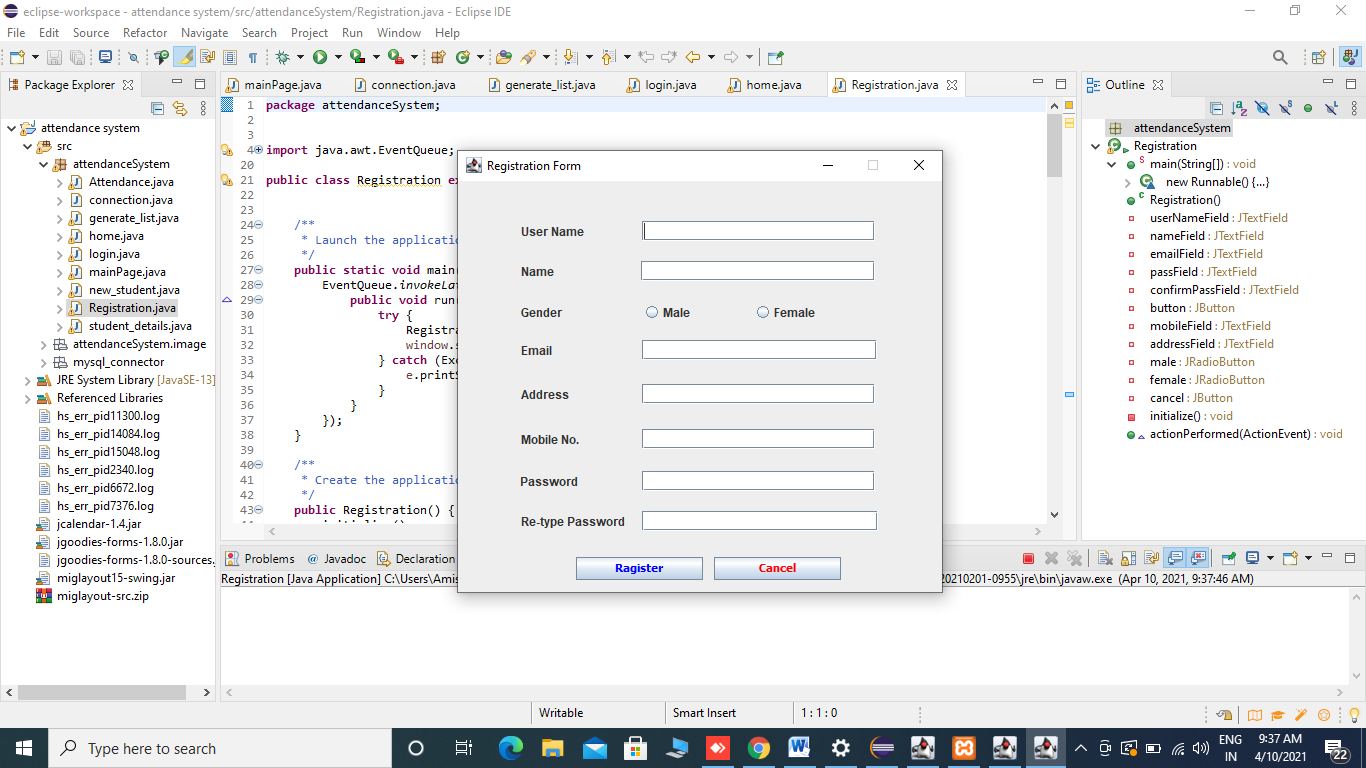




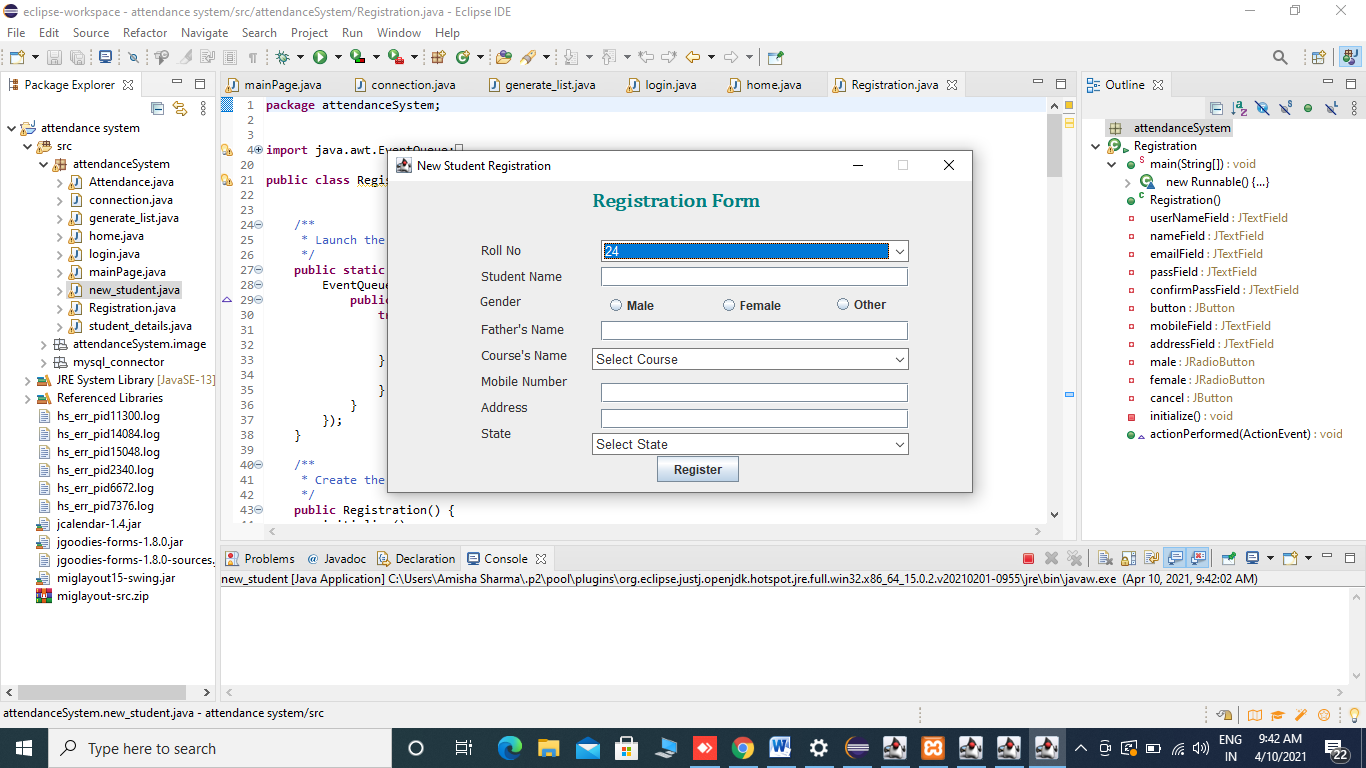
**login page:**



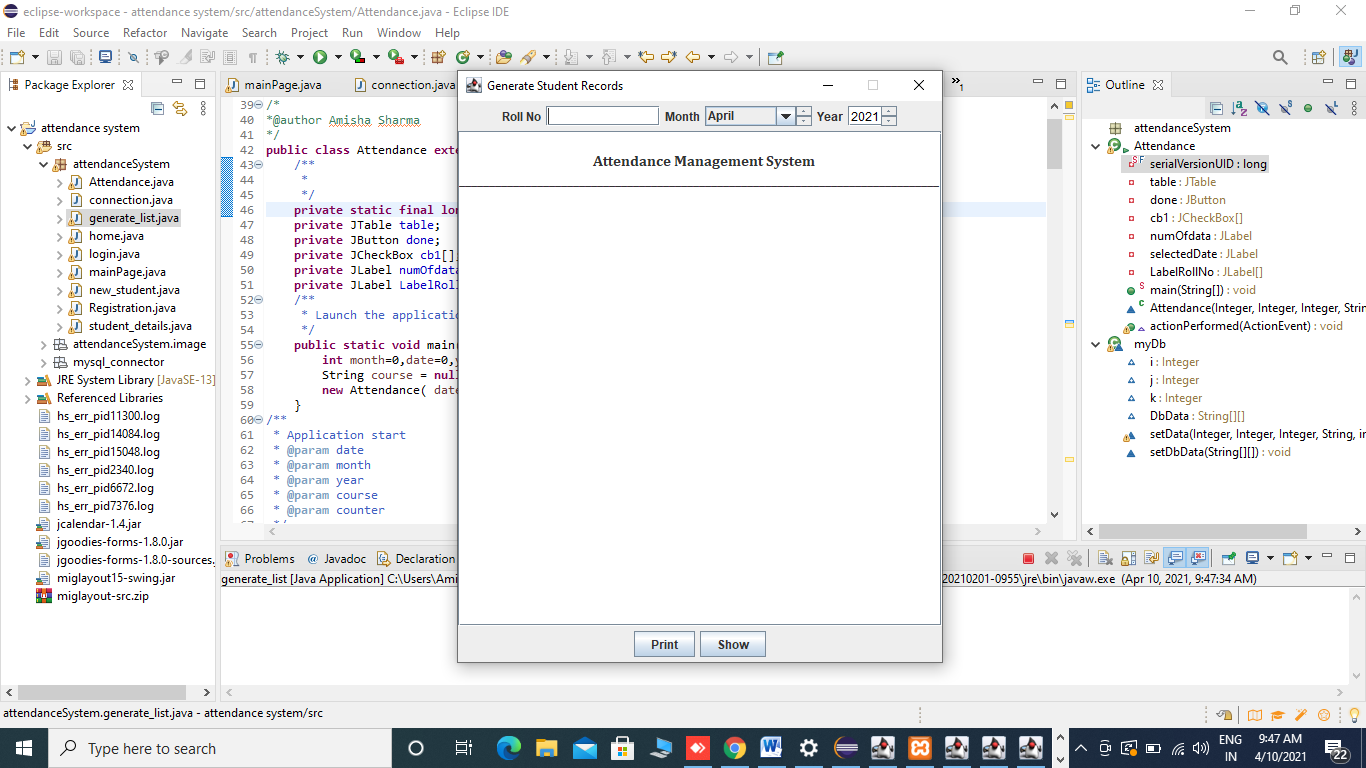
**Home page:**

****

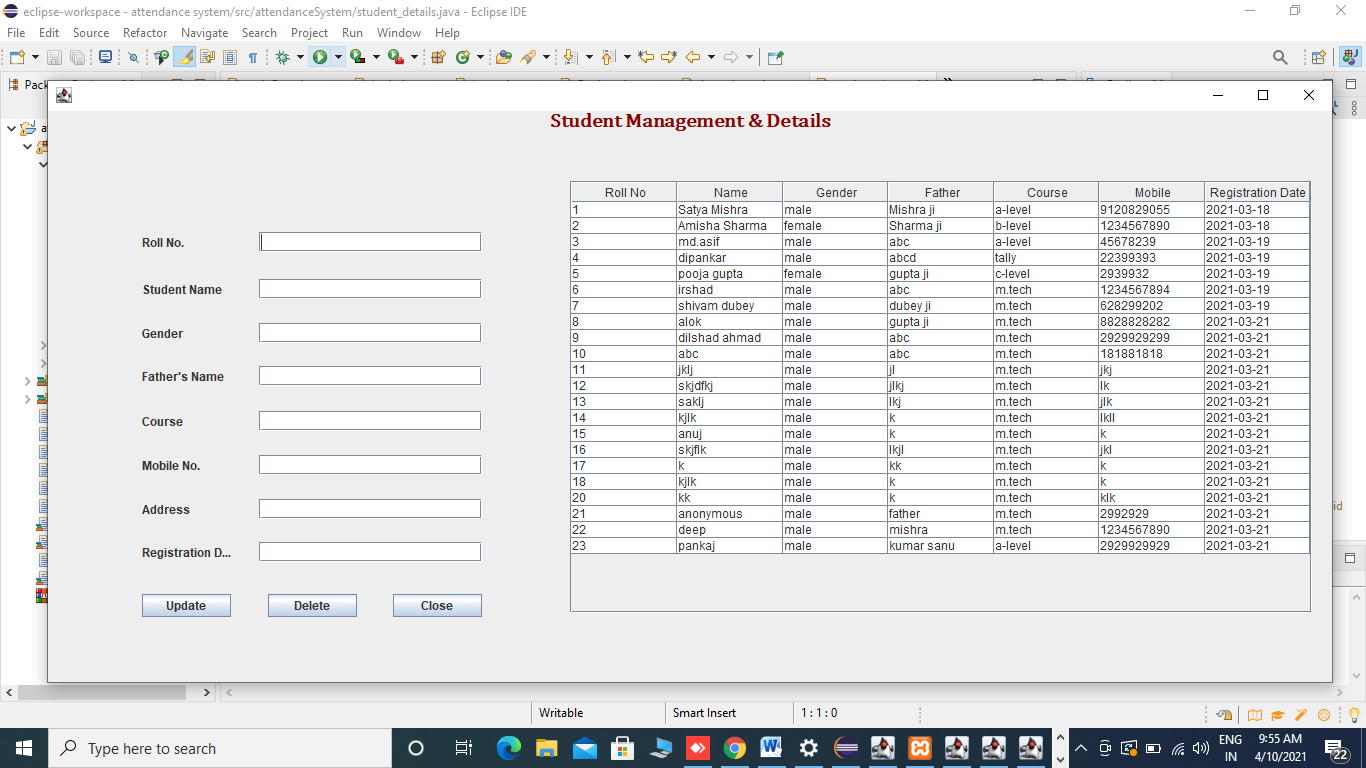
**Registration Form:**



**New Student Registration:**



**Generate Student Record:**



Student Details:

13. BIBLIOGRAPHY

The books referred during the development of this project are as

follows:

 Fundamentals of database systems by (Elmasri Navathe,

2000),

 www.tutorialspoint.com

 Website: https://www.javatpoint.com

###   [Beginning Java 8 APIs, Extensions, and Libraries](https://www.amazon.com/Beginning-Java-APIs-Extensions-Libraries/dp/1430266619?tag=javamysqlanta-20)

14. **FUTURE PLANNING AND SCOPE**

We are trying to give a daily reporting which is updated by institute so that student gets a regular attendance, Available seats, course and also planning to provide good as per theirs choice so that they can student can feel very comfortably . We will trying to provide detail reports and choice to teacher so that they can feel like their comfort and more effective fuctionality. We are

also trying to make more attention on adding on more features class people and their requirements. Our future planning is to take this project towards . So that students can easily contact to the us and they are getting quick Services from Attendance system. We also want in future to place in market so that institutes can take more advantages and saves their important time. We are also finding and approaching to companies which are using this type of software.we are planning to add a biometric function in it also for attendance purpose of the students.