



CHAPTER 1

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

Software Development Life Cycle (SDLC)

1.1 Introduction:-

- Software Engineering:-

Software Engineering is the establishment and use of sound engineering principles in order to obtain economically software i.e. reliable and works efficiently on machines.

Software Engineering:

1. The application of systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software.

2. The study of approaches as in (1).

- A Generic view of Software Engineering:-

The work associated with software engineering can be categorized into three generic phases, regardless of application area, project size, or complexity. Each phase addresses one or more of the questions noted previously.

The definition phase focuses on what. That is, during definition, the software engineer attempts to identify what information is to be processed, what function and performance are desired, what system behavior can be expected, what interfaces are to be established, what design constraints exist, and what validation criteria are required to define a successful system. Four types of change are encountered during the support phase:

Correction: Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects.

Adaptation: Over time, the original environment (e.g., CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

Student Record Management System

Enhancement: As software is used, the customer/user will recognize additional functions that will provide benefit. Perfective maintenance extends the software beyond its original functional requirements.

Prevention: Computer software deteriorates due to change, and because of this, preventive maintenance, often called software engineering, must be conducted to enable the software to serve the needs of its end users.

1.2 What is the SDLC?

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. The concept generally refers to computer or information system.

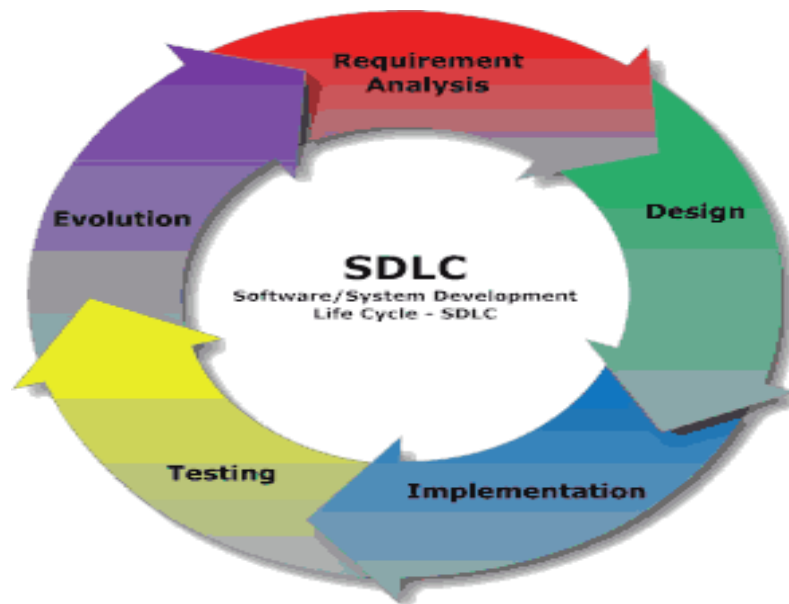


Fig 1.1 Software Development Life Cycle

In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system software development process.

1.2.1 Software requirements analysis:

The requirements gathering process is intensified and focused specifically on software. To understand the nature of the program(s) to be built, the software engineer ("analyst") must understand the information domain for the software, as well as required function, behavior, performance, and interface. Requirements for both the system and the software are documented and reviewed with the customer. Once the general requirements are gleaned from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document.

1.2.2 Design:

Software design is actually a multistep process that focuses on four distinct attributes of a program: data structure, software architecture, interface representations, and procedural (algorithmic) detail. The design process translates requirements into a representation of the software that can be assessed for quality before coding begins. Like requirements, the design is documented and becomes part of the software configuration.

1.2.3 Implementation:

Generally, Implementation can be done by code generation in the different language. The design must be translated into a machine-readable form. The code generation step performs this task. If design is performed in a detailed manner, code generation can be accomplished mechanistically.

1.2.4 Testing:

Once code has been generated, program testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals; that is, conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results.

1.2.5 Evolution :

Software will undoubtedly undergo change after it is delivered to the customer (a possible exception is embedded software). Change will occur because errors have been encountered, because the software must be adapted to accommodate changes in its external environment

(e.g., a change required because of a new operating system or peripheral device), or because the customer requires functional or performance enhancements. Software support/maintenance reapplies each of the preceding phases to an existing program rather than a new one.

1.3 Development:-

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins.

Programmers use various techniques to develop computer programs. The large transaction-oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Primary procedural programming activities include the creation and testing of source code and the refinement and finalization of test plans. Typically, individual programmers write and review (desk test) program modules or components, which are small routines that perform a particular task within an application. Completed components are integrated with other components and reviewed, often by a group of programmers, to ensure the components properly interact. The process continues as component groups are progressively integrated and as interfaces between component groups and other systems are tested.

1.3.1 Development Standards:

Development standards should be in place to address the responsibilities of application and system programmers. Application programmers are responsible for developing and maintaining end-user applications. System programmers are responsible for developing and maintaining internal and open-source operating system programs that link application programs to system software and subsequently to hardware. Managers should thoroughly understand development and production environments to ensure they appropriately assign programmer's responsibilities.

Development standards should prohibit a programmer's access to data, programs, utilities, and systems outside their individual responsibilities. Library controls can be used to manage access to, and the movement of programs between, development, testing, and production

Student Record Management System

environments. Management should also establish standards requiring programmers to document completed programs and test results thoroughly. Appropriate documentation enhances a programmer's ability to correct programming errors and modify production programs.

1.3.2 Software Documentation:

Organizations should maintain detailed documentation for each application and application system in production. Thorough documentation enhances an organization's ability to understand functional, security, and control features and improves its ability to use and maintain the software. The documentation should contain detailed application descriptions, programming documentation, and operating instructions. Standards should be in place that identify the type and format of required documentation such as system narratives, flowcharts, and any special system coding, internal controls, or file layouts not identified within individual application documentation.

1.3.3 System documentation should include:

System Descriptions – System descriptions provide narrative explanations of operating environments and the interrelated input, processing, and output functions of integrated application systems.

1.4 Maintenance:-

Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.

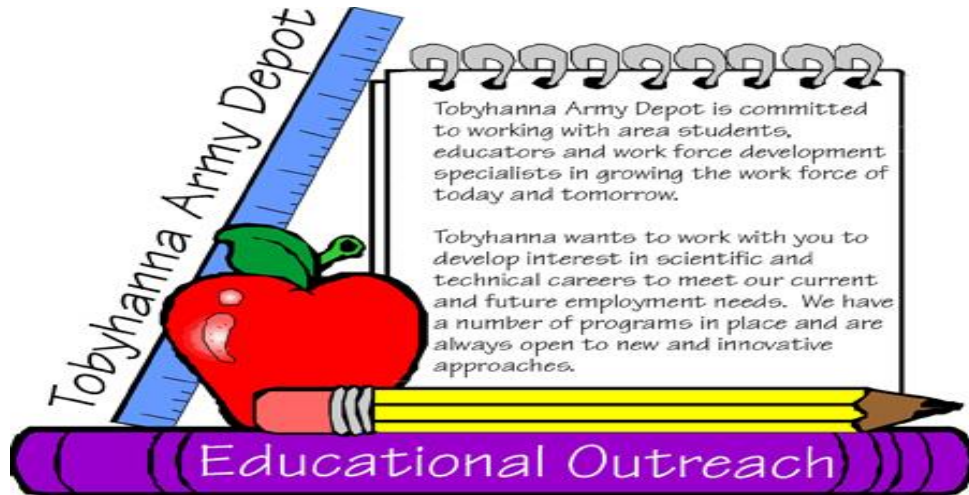
This international standard describes the software maintenance processes as:

1. The implementation processes contains software preparation and transition activities, such as the conception and creation of the maintenance plan, the preparation for handling problems identified during development, and the follow-up on product configuration management.
2. The problem and modification analysis process, which is executed once the application has

Student Record Management System

become the responsibility of the maintenance group. The maintenance programme must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and, finally, obtain all the required authorizations to apply the modifications.

3. The process considering the implementation of the modification itself.
 4. The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.
- Corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered problems.
 - Adaptive maintenance: Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.
 - Perfective maintenance: Modification of a software product after delivery to improve performance or maintainability.



CHAPTER 2

STUDENT RECORD MANAGEMENT SYSTEM

Student Record Management System

2.1 Introduction to Management:-

Management in business and organizations means to coordinate the efforts of people to accomplish goals and objectives using available resources efficiently and effectively. Management comprises planning, organizing, staffing, leading or directing, and controlling an organization or initiative to accomplish a goal. Resourcing encompasses the deployment and manipulation of human resources, financial resources, technological resources, and natural resources.

Since organizations can be viewed as systems, management can also be defined as human action, including design, to facilitate the production of useful outcomes from a system. This view opens the opportunity to 'manage' oneself, a prerequisite to attempting to manage others. The verb 'manage' comes from the Italian *maneggiare*, which derives from the Latin word *manus* (hand). The French word *mesnagement* (later *ménagement*) influenced the development in meaning of the English word *management* in the 17th and 18th centuries.

2.1.1 Definitions

Views on the definition and scope of management include Management defined as the organization and coordination of the activities of an enterprise in accordance with certain policies and in achievement of clearly defined objectives.

Management involves the manipulation of the human capital of an enterprise to contribute to the success of the enterprise. This implies effective communication : enterprise environment (as opposed to a physical or mechanical mechanism), implies human motivation and implies some sort of successful progress or system outcome. As such, management is not the manipulation of a mechanism (machine or automated program), not the herding of animals, and can occur in both a legal as well as illegal enterprise or environment. Based on this, management must have humans, communication, and a positive enterprise endeavor. Plans, measurements, motivational psychological tools, goals, and economic measures (profit, etc.) may or may not be

Student Record Management System

necessary components for there to be management. It considers management to consist of six functions:

1. Forecasting
2. Planning
3. Organizing
4. Commanding
5. Coordinating
6. Controlling

2.1.2 Basic functions

Management operates through various functions, often classified as planning, organizing, staffing, leading/directing, controlling/monitoring and motivation.

- **Planning:** Deciding what needs to happen in the future (today, next week, next month, next year, over the next five years, etc.) and generating plans for action.
- **Organizing:** (Implementation)pattern of relationships among workers, making optimum use of the resources required to enable the successful carrying out of plans.
- **Staffing:** Job analysis, recruitment and hiring for appropriate jobs.
- **Leading/directing:** Determining what must be done in a situation and getting people to do it.
- **Controlling/monitoring:** Checking progress against plans.
- **Motivation:** Motivation is also a kind of basic function of management, because without motivation, employees cannot work effectively. If motivation does not take place in an organization, then employees may not contribute to the other functions (which are usually set by top-level management).
- **Communicating:** is giving, receiving, or exchange information.
- **Creating:** ability to produce original Idea , thought through the use of imagination

2.2 Problem Domain

Our project is totally depends on the management of all records of an student . If we does not create any system software for managing the record , Than it is very difficult to manage it as in file and registers. Above all the function of management are used in the system.

2.3 Student Record Management System

A student record management system is used to manage, catalogue, and keep safe all student data, from personal information to academic information. Types of student record management systems include MES Hybrid Document System.

These systems vary in size , scope and capability , from packages that are implemented in relatively small organizations to cover student records alone, to enterprise-wide solutions that aim to cover most aspects of running large multi-campus organizations with significant local responsibility. Many systems can be scaled to different levels of functionality by purchasing add-on "modules" and can typically be configured by their home institutions to meet local needs.

In larger enterprise solutions that have student data at their core, further functions include Student financial aid management and more may be customized by the developer. Where national or government systems exist for student finance or statistical return purposes , student records system often provide functionality that caters for this, by way of modules or core elements that handle the production of required files , or deal with the formatted transfer of information.

2.4 Project Objective

The main objective of our system to organize the internal and external record related to college for the different purposes. We have to maintain the student record as per the authentication and the student registration. To Study the JDBC Techniques in Java for the use of connectivity of our source code project to the database for storing the student record in the database. To Study the Swings Techniques of the GUI components which helps to design the Frames in our project . And about linking technique to link this frames.

2.5 Scope and Limitations

Scope:

Our project scope is not a big. Its an only prefer to the college level . Its not an web based application ,Its an desktop based application which helps to provide an interactive purpose for the college.

Limitation:

Our System is only for the limited student which are belongs to our college and the department. Our System can only store the record about the college session. it does not contain the data about the personal details of the student. The security should be limited only the registration of the student in terms of their authentication.

2.6 ANALYSIS OF OUR SYSTEM:

Feasibility Study:

Our System does not provide more security to the student identification like biological security. It provide only an security of student identification by the help of their image and the signature.

Existing System:

We should only prefer for these System is to the database System with the java language like JDBC.

Problem Definition:

If we did not take the help of these system than we can face many problem in management of student record. Because this System provides an interactive feature for registering , deleting , modifying the student data.

2.7 USER GOALS FOR A SYSTEM:

User goals for a “ Student Records Management System”

- **Practicality:** The system is quite stable and can be operated by the people with average intelligence.
- **Efficiency:** We tried to involve accuracy, timeliness and comprehensiveness of the system output.
- **Flexibility:** We have tried that the system should be modifiable depending on the changing needs of the user.

Such modifications should entail extensive reconstructing or recreation of software. It should also be portable to different computer systems. **SECURITY PURPOSE** Allow only authorized faculty and staff to insert, save & show student records. Allow only administrator for insert, update, delete, save, show student records.

The information store in the database should be secured by the database security. The database security provides an interactive possibility to enter the password in the form than we can retrieve the record from the database.

2.8 TOOLS USED:

1.JDBC Technology:

Java is a Purely object – oriented Language . In the Java there are the certain important tools are used in the our System. JDBC is an important tools are used for connecting To the database which is used for storing the student data. JDBC is a Java database connectivity API that is a part of the Java Enterprise APIs from Java Soft . From a developer's point of view, JDBC is the first standardized effort to integrate relational databases with Java programs. JDBC has opened all the relational power that can be mustered to Java applets and applications. Database clients use database drivers to send SQL statements to database servers and to receive result set and other responses from the servers. JDBC drivers are used by Java applications and applets to communicate with database servers. Officially, Sun says that JDBC is an acronym that does not stand for anything. However, it is associated with "Java database connectivity".

Student Record Management System

JDBC is a Java database connectivity API that is a part of the Java Enterprise APIs from Java Soft. From a developer's point of view, JDBC is the first standardized effort to integrate Relational databases with Java programs. JDBC has opened all the relational power that can be mustered to Java applets and applications.

Model

JDBC is designed on the CLI model. JDBC defines a set of API objects and methods to interact with the underlying database. A Java program first opens a connection to a database, makes a statement object, passes SQL statements to the underlying DBMS through the statement object, and retrieves the results as well as information about the result sets. Typically, the JDBC class files and the Java applet reside in the client. To minimize the latency during execution, it is better to have the JDBC classes in the client. As a part of JDBC, Java Soft also delivers a driver to access ODBC data sources from JDBC. This driver is jointly developed with Intersolv and is called the JDBC-ODBC bridge. The JDBC-ODBC bridge is implemented as the `JdbcOdbc` class and a native library to access the ODBC driver. For the Windows platform, the native library is a DLL (`JDBCODBC.DLL`). As JDBC is close to ODBC in design, the ODBC Bridge is a thin layer over JDBC. Internally, this driver maps JDBC methods to ODBC calls and, thus, interacts with any available ODBC driver. The advantage of this bridge is that now JDBC has the capability to access almost all databases, as ODBC drivers are widely available. The JDBC-ODBC Bridge allows JDBC driver to be used as ODBC drivers by converting JDBC method calls into ODBC function calls.

2. Swing Technology:

Swing did not exist in the early days of Java. Rather, it was a response to deficiencies presenting Java's original GUI subsystem: the Abstract Window Toolkit. The AWT defines a basic set of controls, windows, and dialog boxes that support a usable, but limited graphical interface. One reason for the limited nature of the AWT is that it translates its various visual components into their corresponding, platform-specific equivalents, or peers. This means that the look and feel of a component is defined by the platform, not by Java. Because the AWT components use native code resources, they are referred to as heavyweight. The use of native peers led to several problems. First, because of variations between operating systems, a

Student Record Management System

component might look, or even act, differently on different platforms. This potential variability threatened the overarching philosophy of Java: write once, run anywhere. Second, the look and feel of each component was fixed (because it is defined by the platform) and could not be (easily) changed. Third, the use of heavyweight components caused some frustrating restrictions. For example, a heavyweight component is always rectangular and opaque.

Swing Components Are Lightweight

With very few exceptions, Swing components are lightweight. This means that they are written entirely in Java and do not map directly to platform-specific peers. Because lightweight components are rendered using graphics primitives, they can be transparent, which enables nonrectangular shapes. Thus, lightweight components are more efficient and more flexible. Furthermore, because lightweight components do not translate into native peers, the look and feel of each component is determined by Swing, not by the underlying operating system. This means that each component will work in a consistent manner across all platforms.



CHAPTER 3

SYSTEM DESIGN AND DEVELOPMENT

System Design and Development

(Student Record Management System)

Design is a meaningful engineering representation of something that is to be built. It can be traced to a customer's requirements and at the same time assessed for quality against a set of predefined criteria for "good" design. In the software engineering context, design focuses on four major areas of concern: data, architecture, interfaces, and components. Software engineers design computer-based systems, but the skills required at each level of design work are different. At the data and architectural level, design focuses on patterns as they apply to the application to be built. At the interface level, human ergonomics often dictate our design approach. At the component level, a "programming approach" leads us to effective data and procedural designs. Design begins with the requirements model. We work to transform this model into four levels of design detail: data structure, system architecture, the interface representation and the component level detail.

3.1 Design of System:

Our Project System of Student Record Management includes security of our departmental purpose. Designing the Front end with the help of swing technology. Swing technology is used to create different frames which helps to provide the GUI (Graphical User Interface). Frames provide a better user interaction which helps the user to understand the purpose of our system.

In our System, we use the different menus for managing all records (internal and external) of the student related to college session. Firstly it focuses on the security. It requires a user name which is non-editable (default). The user name is "GP Gondia". Means whole Authentication is based on the password which is "gpg2014". After that main Menu is come.

Student Record Management System

The Component Available to maintain the Student Record are as follow:

1. Admission of an Student
2. External Marks of an Student
3. Internal Marks of an Student
4. Report Generation of an Student
5. About

3.1.1 Admission of an Student

This is the main menu of an system . This menu is responsible for keep take entry of an student in the system . if student entry are not present in this menu. Than this student record are not present in the system specified by other menu . The ” Enrollment no” is an unique identity for each student. Every Student will identified by thereunique ”Enrollmentno”.These menu contained an some submenu which is responsible for adding , deleting and modifying the student record. These submenus are as follow:

1. New Student Record.
2. Remove Student Record
 1. Remove one
 2. Remove all
3. Modify Student Record
4. Exit

3.1.2 External Marks of an Student

This menu is responsible for keep the entry of an external records of an student. External Record contain information about the external theory marks of an student and related external oral marks. This menu is only contain those student record whose entry are keep in the admission menu.

Student Record Management System

3.1.3 Internal Marks of an Student

This menu is responsible for keep the entry of an internal records of an student. Internal Record contain information about the internal term work of an student. It also involves the information about the internal oral , practical marks. This menu is only contains those student record whose entry are keep in the admission menu.

3.1.4 Report Generation of an Student

This menu is responsible for generating the final report of an student. Firstly, it take the marks details from Internal and External menu of an System. For every semester and for every student, the report will be generated. The report contains all the internal and external marks of an all subject related to particular semester of student. It will also calculate the total marks and generate the percentage of an student. Simply, it will generate the mark sheet of an student.

3.1.5 About

This menu are not used for any purpose of an student. It simply give the details of an developer whose develop this system.

Normally , Above all menus are present in the front end. But in Back end we use Oracle 10g database to store all details which is given by user in the front end. We creates an number of tables in the database for storing details and link them with the java programs using pure JDBC technology . We does not use the JDBC-ODBC concept .Because it does not support by every operating system. Certain operating system does not contain the ODBC driver which are require by JDBC-ODBC concept. So we use the PURE JDBC concept. The System are only available for information of CO1G , CO2G , CO3G ,CO4G ,CO5E and CO6E semesters .It does not contain information about another semester details.

3.1.6 Making Frames:

- Making an different frames for establishing Graphical user interface using swing . There are different classes of AWT component are used in the project . Ex. Jbutton , JTextField.
- A **graphical user interface (GUI)** presents a user-friendly mechanism for interacting with an application. We cover other GUI components as they're needed throughout the

rest of the book.

- The most widely used events are those defined by the AWT and those defined by Swing. This chapter focuses on the AWT events . we use event handling interfaces like ActionListener .

Swing supports a pluggable look and feel (PLAF). Because each Swing component is rendered by Java code rather than by native peers, the look and feel of a component is under the control of Swing . This fact means that it is possible to separate the look and feel of a component from the logic of the component , and this is what Swing does. Separating out the look and feel provides a significant advantage: it becomes possible to change the way that a component is rendered without affecting any of its other aspects. In other words, it is possible to “plug in” a new look and feel for any given component without creating any side effects in the code that uses that component . Moreover , it becomes possible to define entire sets of look-and-feels that represent different GUI styles . To use a specific style, its look and feel is simply “plugged in.” Once this is done , all components are automatically rendered using that style. Pluggable look-and-feels offer several important advantages. It is possible to define a look and feel that is consistent across all platforms. Conversely, it is possible to create a look and feel that acts like a specific platform. For example, if you know that an application will be running only in a Windows environment, it is possible to specify the Windows look and feel. It is also possible to design a custom look and feel. Finally, the look and feel can be changed dynamically at run time.

3.2 Hardware Requirements:-

- 1) **Processors :-** Pentium III onwards
- 2) **Main Memory:-** Minimum 64 MB
Recommended 128MB
- 3) **Hard Disk:-** Minimum 10.2GB
Recommended 40 GB

Student Record Management System

4) **Clock Speed:-** 866 MHZ or above

5) **Virtual Memory:-** 32bits

The preferred hardware configuration is:

- a. Intel(R) 845GVSR motherboard
- b. Pentium(R) 4 CPU 2.40GHz
- c. 256MB of DDR RAM

6) **Monitor:-** VGA/SVGA.

7) **Keyboard:-** 104 keys.

8) **Mouse:-** 2/3 Button.

3.3 Software Requirement:-

1. Operating System-

This software runs on windows operating system, it runs on operating systems such as:-

- **Windows 98**
- **Windows XP**
- **Windows 07**
- **Windows 08 etc.**

2. Front End:-

In front end we use java framework version 6.0 and link those frames using swings. Creates an better user interaction with frames.

Student Record Management System

3.Back End:-

In back end we use oracle 10g for storing the Student record and connect them with the java programs by using Pure JDBC technology.

4.Design Tool:-

AWT components are used for design an user interaction with swing provided by java. All the AWT components are used for graphical user interface(GUI).

3.4System Design:-

User login- To start the interaction with software the user has to login with password.

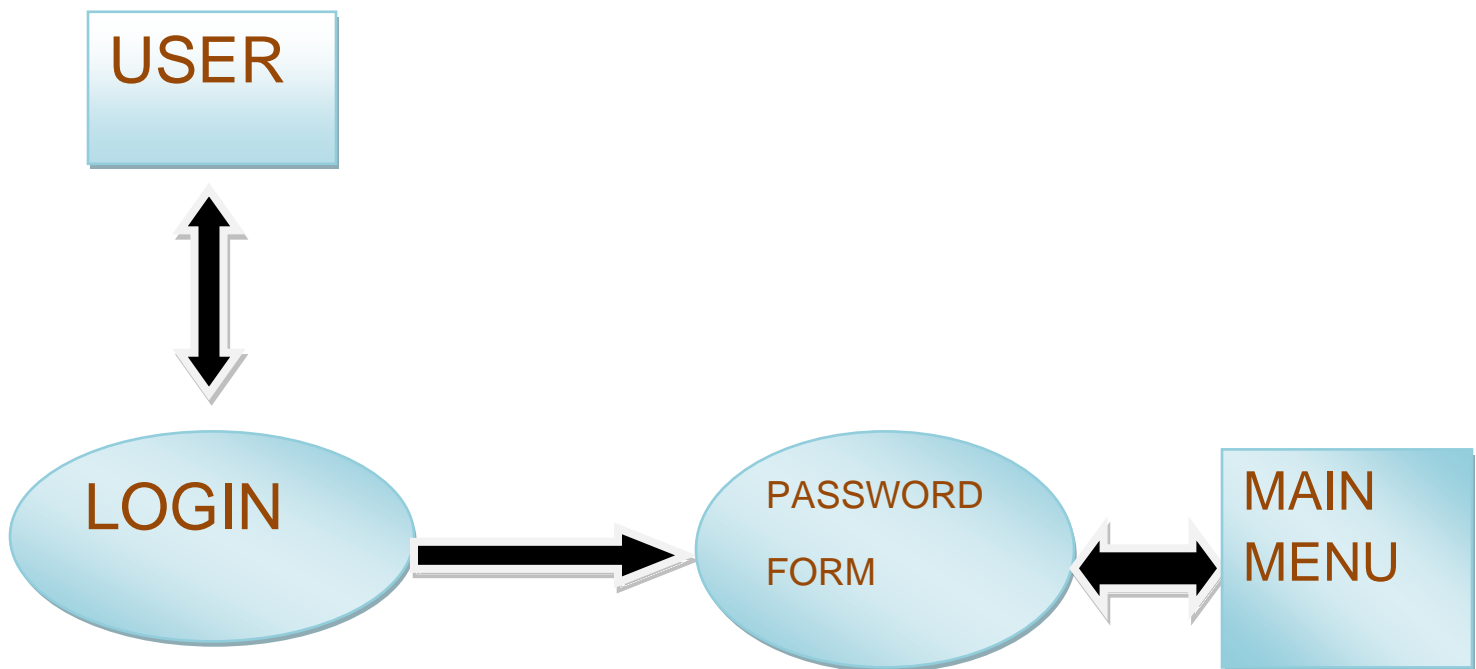
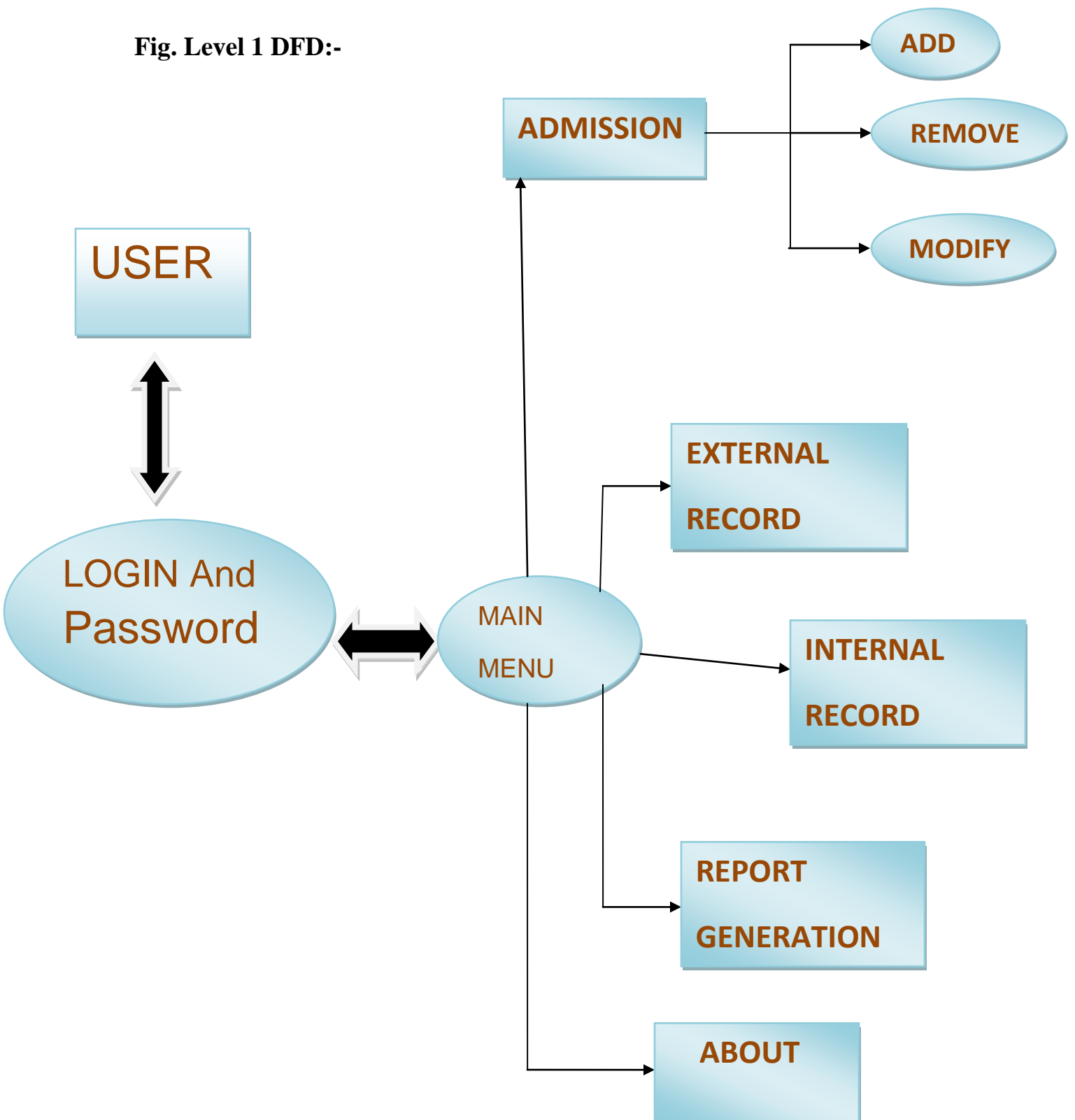


Fig. Level 0 DFD

Fig. Level 1 DFD:-





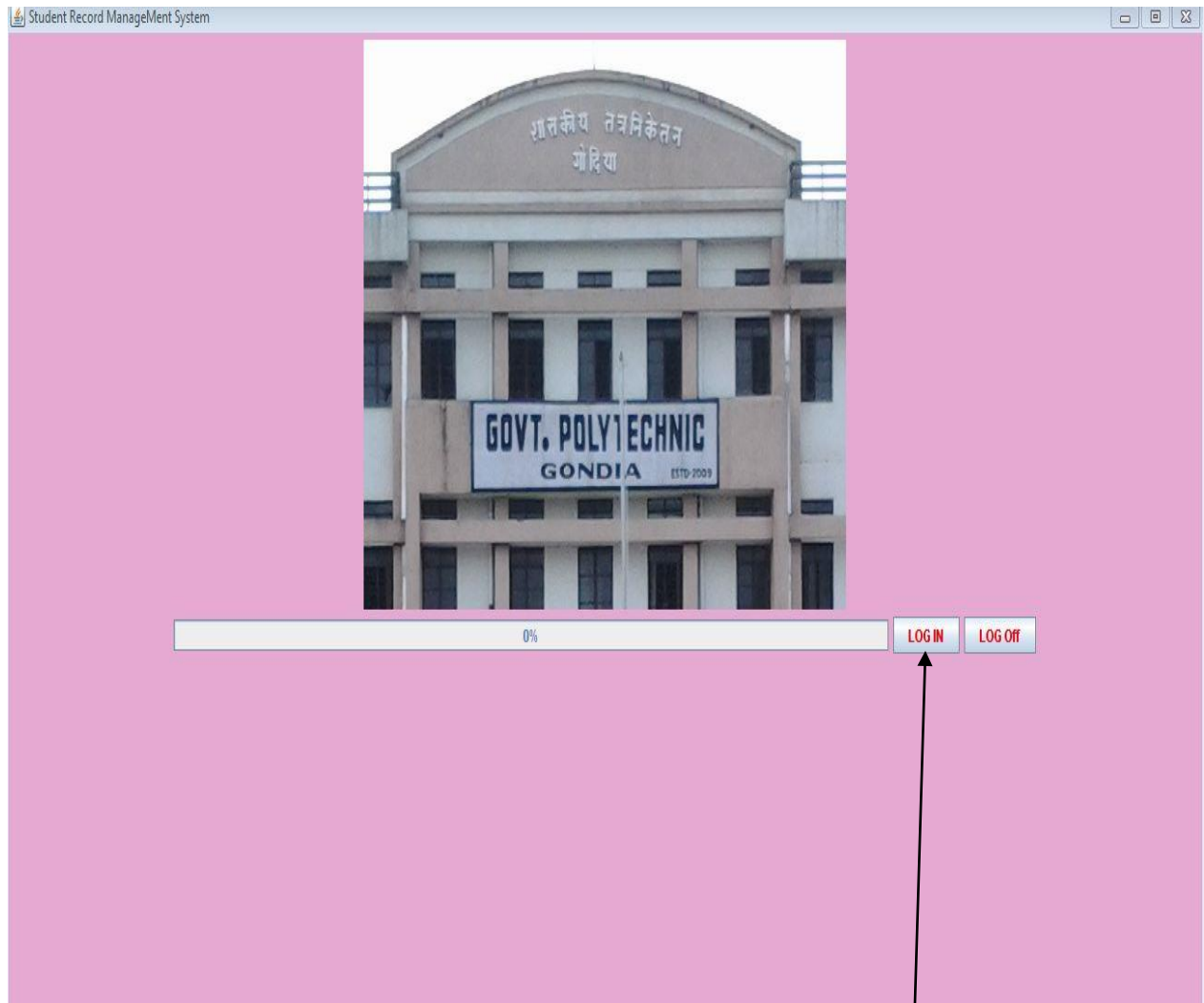
Chapter 4

RESULTS

Results

4.1 User Interface:-

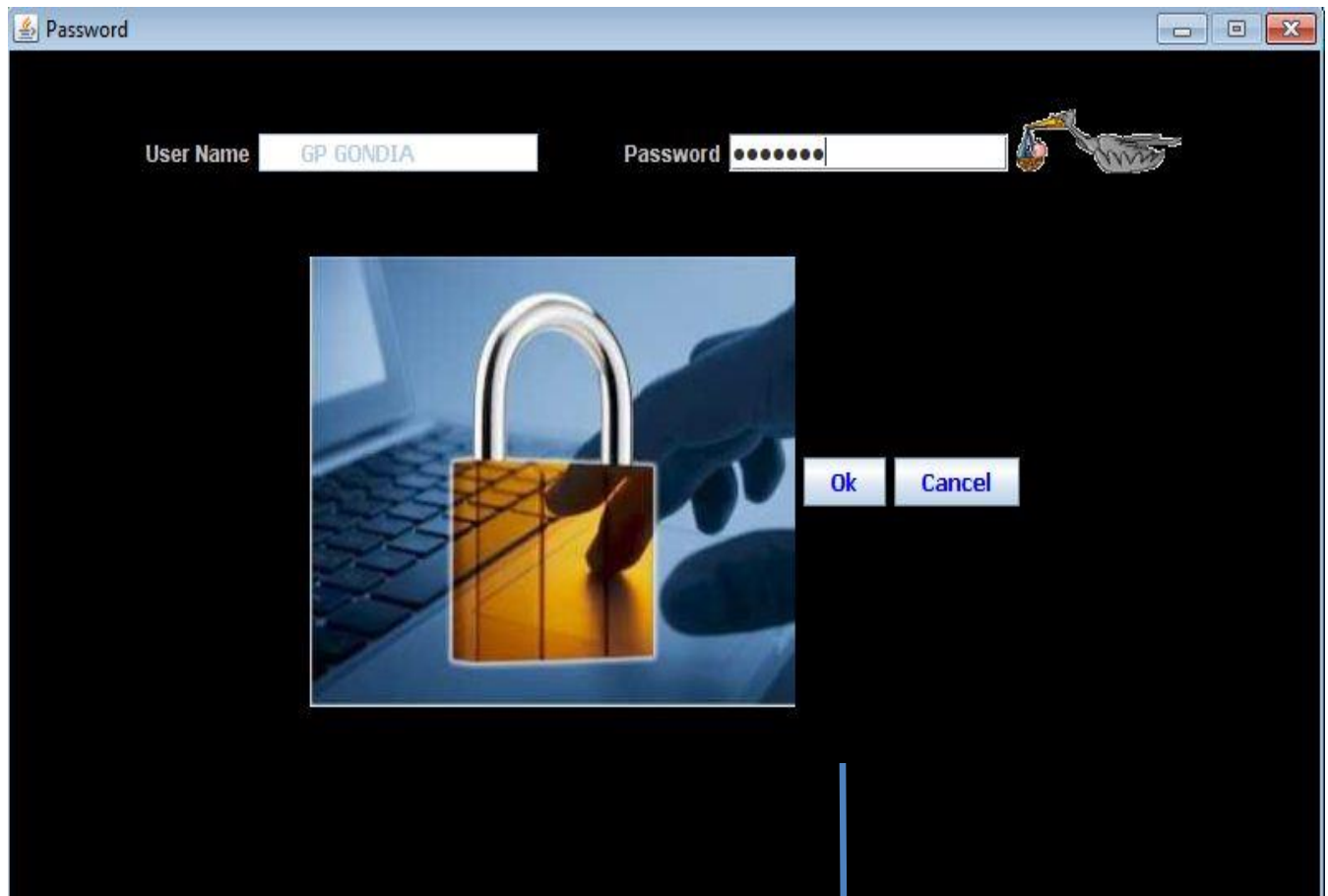
1. User Login Window:-



Press Login to
enter setup

Student Record Management System

2.After Login Password Form is opened:-

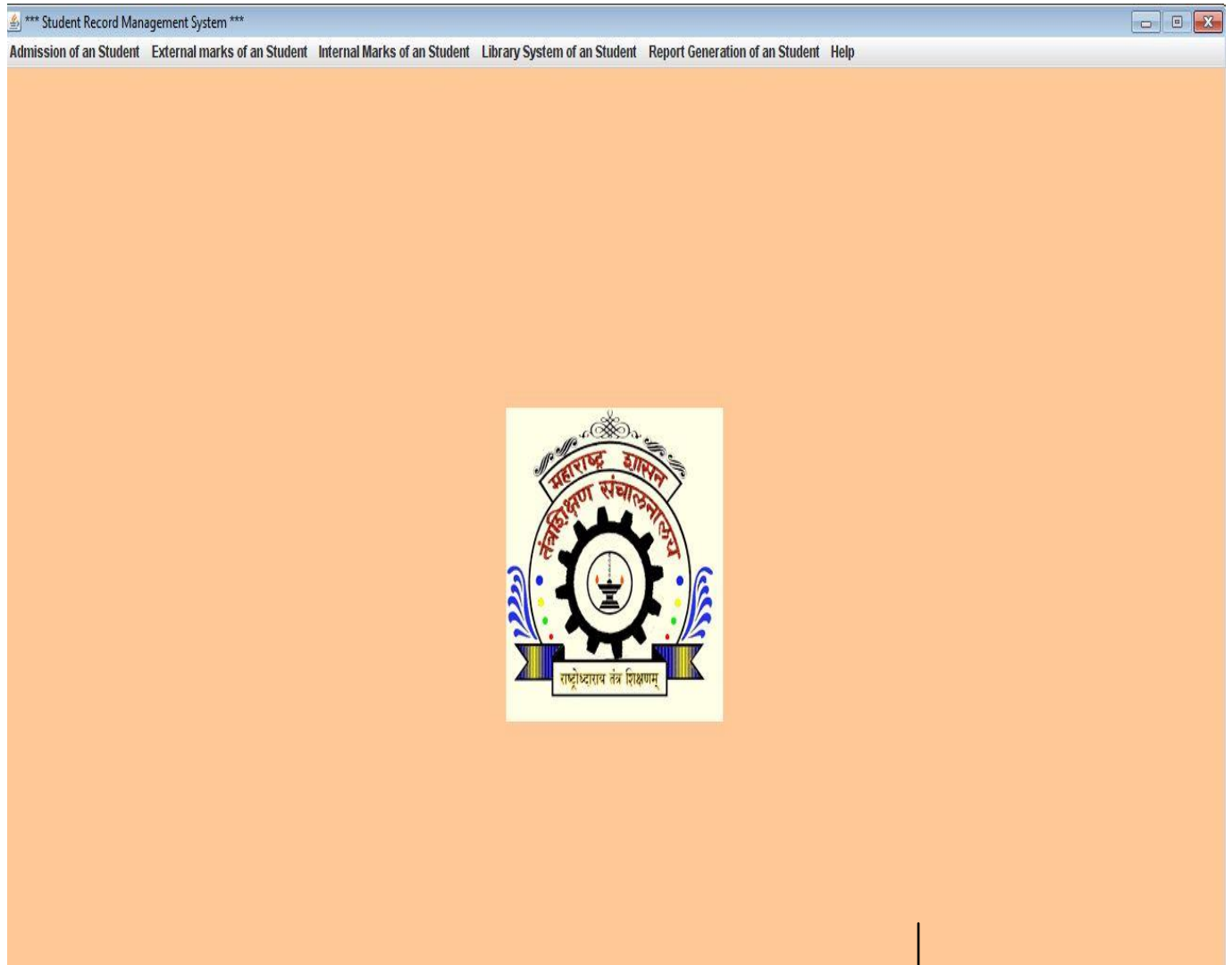


The screenshot shows a window titled "Password" with a black background. At the top, there are two input fields: "User Name" containing "GP GONDIA" and "Password" containing seven dots. To the right of the password field is a small icon of a hand holding a key. Below these fields is a large image of a hand holding a yellow padlock over a laptop keyboard. To the right of this image are two buttons: "Ok" and "Cancel".

Enter Password
And Click OK

Student Record Management System

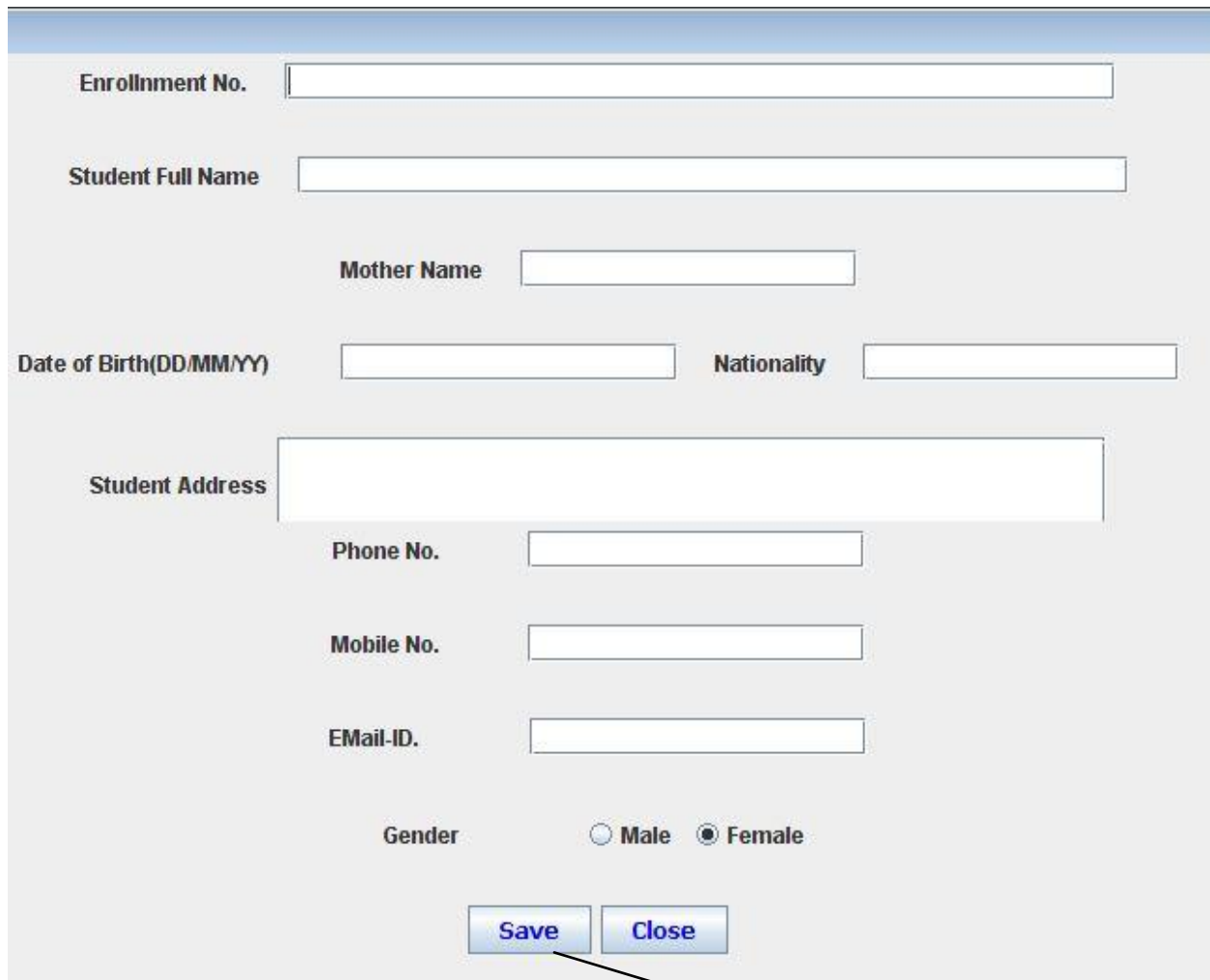
3. After Password form Main menu is opened:-



Chose Menu
From Above

Student Record Management System

4. Admission System of an student:-



A web form for student admission. It features a light blue header bar. The form fields are arranged vertically: Enrollment No. (text box), Student Full Name (text box), Mother Name (text box), Date of Birth(DD/MM/YY) (text box), Nationality (text box), Student Address (text box), Phone No. (text box), Mobile No. (text box), EMail-ID. (text box), and Gender (radio buttons for Male and Female, with Female selected). At the bottom are Save and Close buttons.

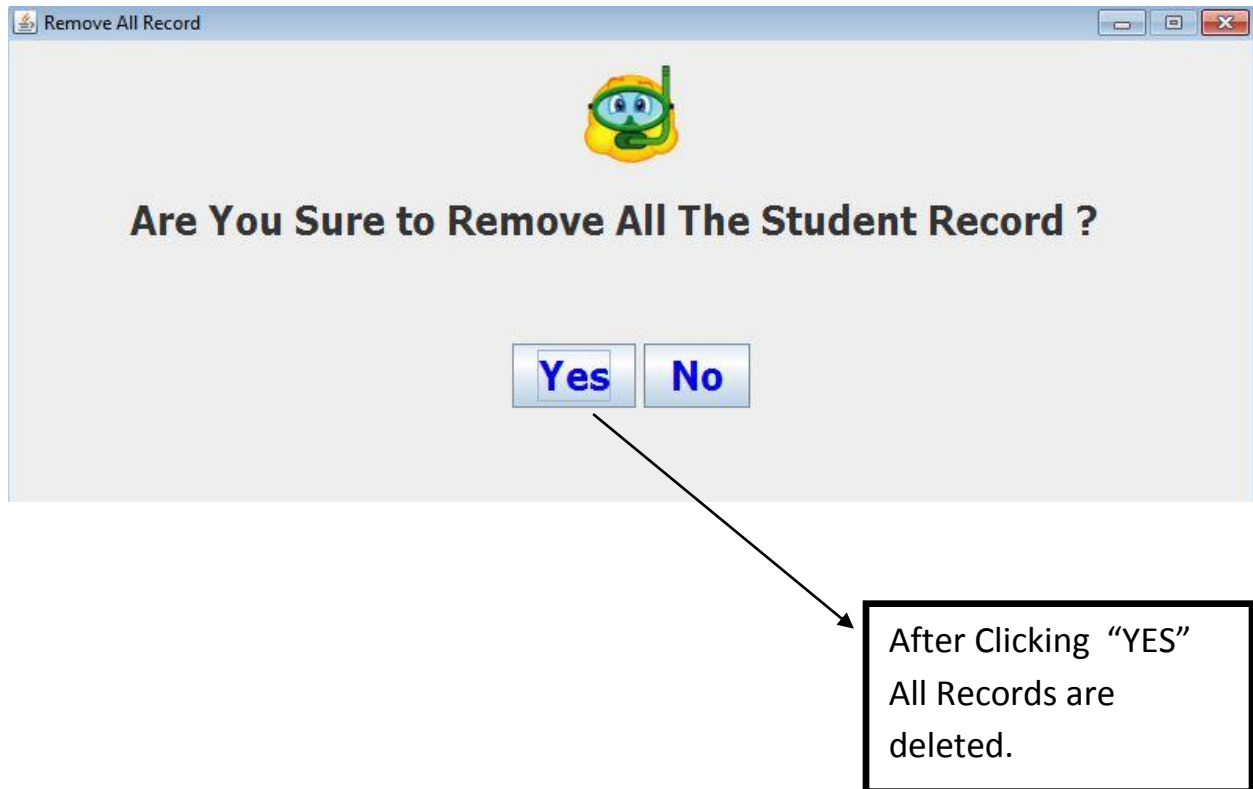
Enrollment No.	<input type="text"/>
Student Full Name	<input type="text"/>
Mother Name	<input type="text"/>
Date of Birth(DD/MM/YY)	<input type="text"/>
Nationality	<input type="text"/>
Student Address	<input type="text"/>
Phone No.	<input type="text"/>
Mobile No.	<input type="text"/>
EMail-ID.	<input type="text"/>
Gender	<input type="radio"/> Male <input checked="" type="radio"/> Female
<input type="button" value="Save"/> <input type="button" value="Close"/>	

Fill Information
And Than Click
Save

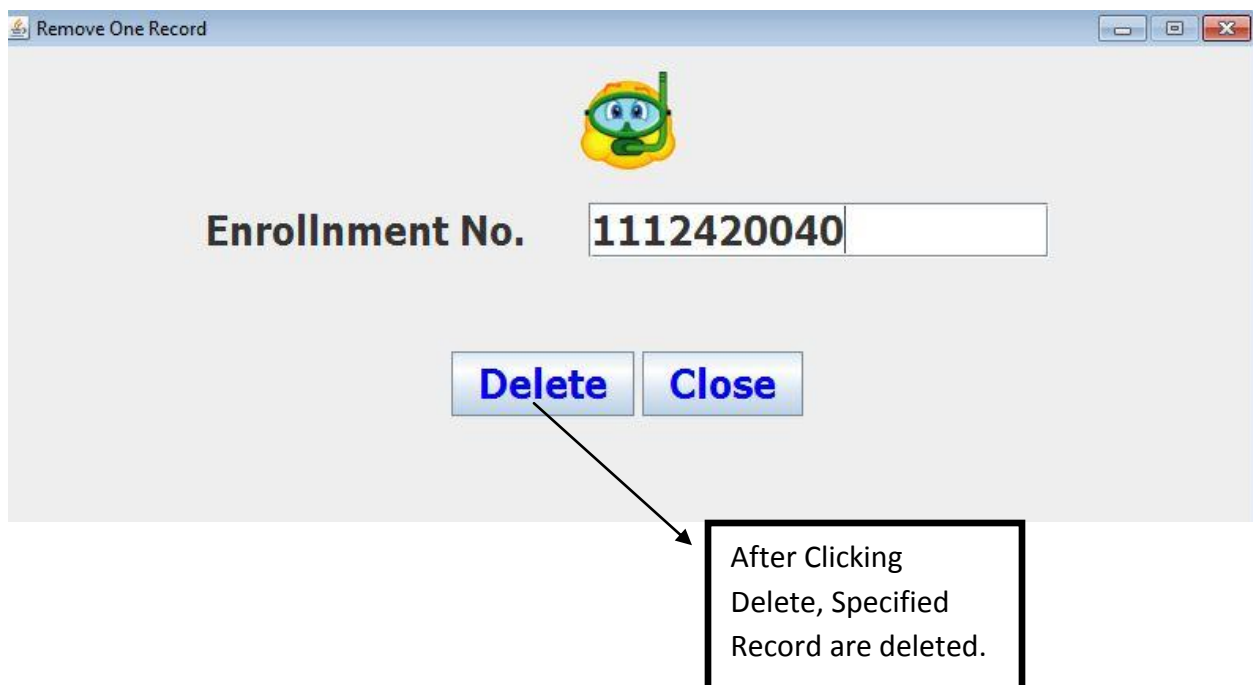
Student Record Management System

5.Removing the Student Record from System:-

Remove all:

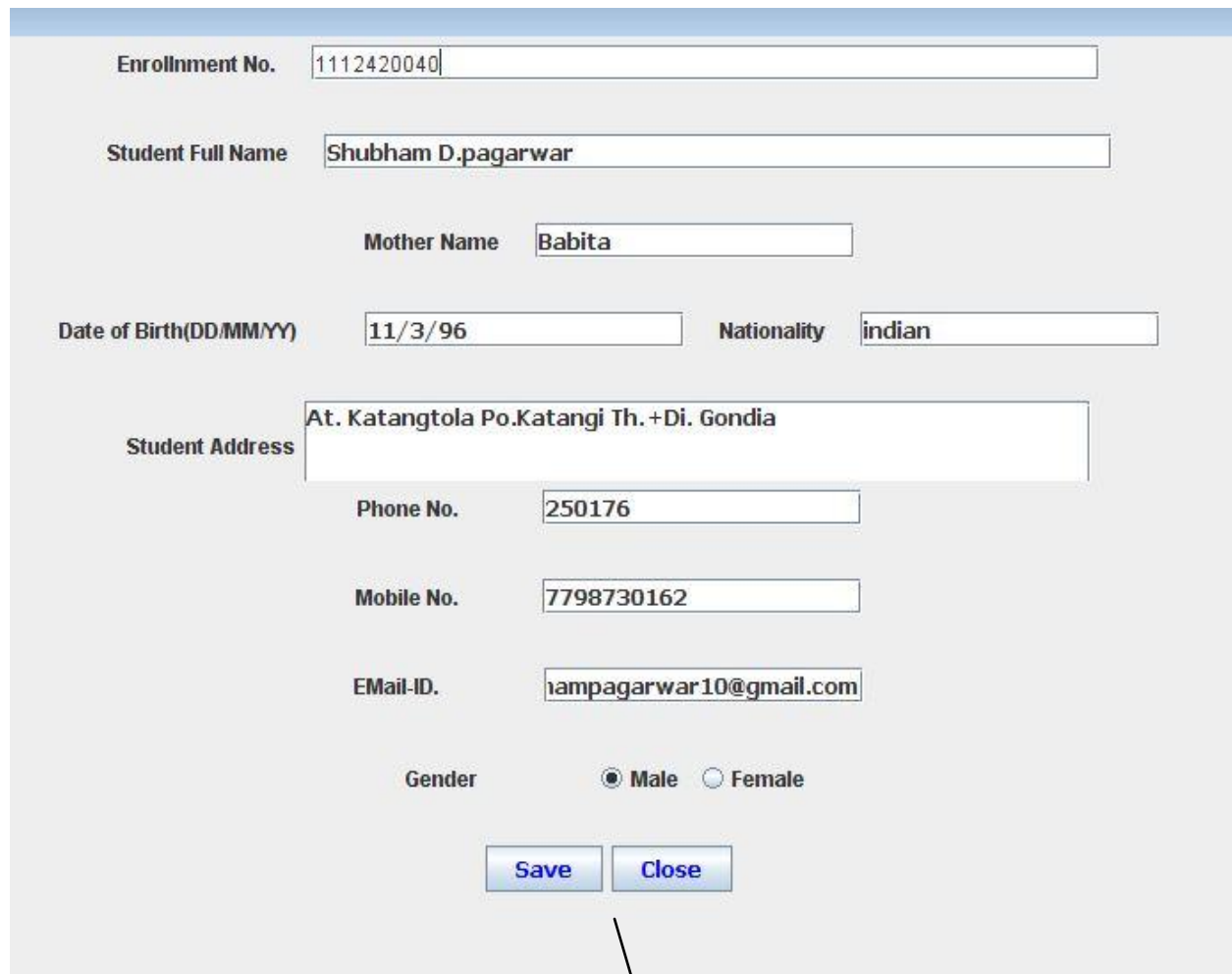


Remove One:



Student Record Management System

6.Modifying the Student Record:-



The screenshot shows a web form for modifying a student record. The form has a light blue header and a light gray background. It contains several text input fields and a radio button group. The fields are labeled as follows:

- Enrollment No. (1112420040)
- Student Full Name (Shubham D.pagarwar)
- Mother Name (Babita)
- Date of Birth(DD/MM/YY) (11/3/96)
- Nationality (indian)
- Student Address (At. Katangtola Po.Katangi Th.+Di. Gondia)
- Phone No. (250176)
- Mobile No. (7798730162)
- EMail-ID. (iampagarwar10@gmail.com)
- Gender (Male selected, Female unselected)

At the bottom of the form, there are two buttons: "Save" and "Close". An arrow points from the "Save" button to a text box below the form.

Firstly Modify the Record ,Than Click "Save" button to save it.

7 .External Marks of an Student:-

Enrollment No.= 1112420040

Student Full Name = Shubham D.pagarwar

Branch And Sem= co1g

Seat No.= 1234

Please Enter the Marks of Following Subject

English /100

Basic Physics /50

Basic Chemistry /50

Basic Mathematics /100

Fill Information
And Than Click
Save

8.Internal Marks of an Student:-

Enrollment No.= 1112420040

Student Full Name = Shubham D.pagarwar

Branch And Sem= co1g

Please Choose And Enter the Marks of Following Subject

English TermWork : /25

Basic Physics Practical : /25

Basic Chemistry Practical : /25

Computer Fundamental TermWork /25

Computer Fundamental Practical: /50

Workshop Practice TermWork : /50

Engineering Graphics Practical: /50

Engineering Graphics Termwork: /50

Sessional: /50

Save **Close**

Fill Information
And Than Click
Save

8. Report Generation of an Student:-

Enrollment No.= 1112420040			
Student Full Name = Shubham D.pagarwar			
Branch And Sem= co1g			
Seat No.= 123			
Title of Subject	Subject head	Max. Marks	Marks Obtained
English	TH	100	89
	TW	25	21
Basic Science	TH	100	92
	PR	50	46
Basic Mathematics	TH	100	100
Engineering Graphics	PR	50	48
	TW	50	50
Computer Fundamental	PR	50	23
	TW	25	24
Basic Workshop Practice	TW	50	44
Sessional	SW	50	44
Total Marks :		650	596
Percentage :		91.692314	
<input type="button" value="Close"/>			

Display an Mark sheet. Click Close to Exit.

4.2 Future Scope:-

We can add more features in it and enhanced it for future purposes. In following ways we can enhanced it:-

- We can add some other Menus like Library System, Co-operative System ,Lab System for a Student.
- We can expand scope of semester for all branches of data are stored. Like data for Information Technology , Electronics and Telecommunication and Mechanical Engineering also.
- Our Project is an desktop system, So we can add the Client server architecture to the system. So it can be access in client server network.
- In also future this software is also important to prevent from unauthorized access and users.
- We can give the option for changing the password dynamically change by the authorized user.



CHAPTER 5

CONCLUSION

Conclusion:-

The system has been developed for the given condition and is found working effectively . The developed system is flexible and changes whenever can be made easy. Using the facilities and functionalities of Java, the software has been developed in a neat and simple manner , thereby reducing the operators work. The speed and accuracy are maintained in proper way. The user friendly nature of this software developed in Java framework is very easy to work with both for the higher management as well as other employees with little knowledge of computer. The results obtained will fully satisfactory from the user point of view.

REFERENCES: -

1. Roger S. Pressman, "Software engineering: a practitioner's approach" India: McGraw Hill, 2001
2. <http://www4.comp.polyu.edu.hk>
3. <http://seminarprojects.com>
4. <http://www.freestudentprojects.com>
5. <http://www.kashipara.com>
6. Java The Complete Reference - 7th Edition [DwzRG]
7. Programming_with_Java_a_Primer3e_By_Balagurusamy
8. AdvancedJavaProgrammingwithDatabaseApplication
9. 009.01 Arlow JDBC Tutorial July 2005
10. <http://www.wikipedia.com>

▪