HOSTEL MANAGEMENT SYSTEM

A PROJECT REPORT SUBMITTED TO
THE

BABA MASTNATH UNIVERSITY

FOR PARTIAL FULFILLMENT OF THE DEGREE OF

BACHELOR OF COMPUTER APPLICATIONS (BCA)

By
Jitender
222349

SUBMITTED TO

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

BABA MASTNATH UNIVERSITY ASTHAL
BOHAR, ROHTAK (124010)

YEAR 2025

ANNEXURE A

CERTIFICATE FROM GUIDE

Certified that this Project entitled as "HOSTEL MANAGEMENT SYSTEM", submitted in partial fulfilment of the degree of BACHELOR COMPUTER APPLICATIONS
(BCA) to Baba Mastnath University, Asthal Bohar, Rohtak, done by Mr. Jitender , Roll No. 222349 is an authentic work carried out by him under Dr. Neha guidance.

To the best of my knowledge and belief, the matter embodied in this Project Report is an original and authentic work and has not been submitted earlier for the award of any degree or diploma.

Signature of the student

Signature of the guide

ANNEXURE B

CERIFICATE FROM COMPANY/ INDUSTRY/ ORGANISATION

Certified that the project titled "HOSTEL MANAGEMENT SYSTEM", submitted by Mr. Jitender , Roll no. 222349, was carried out in this organization as part of the curriculum for the Bachelor of Computer Applications (BCA) degree at Baba Mastnath University. The duration of the project was three months (from January 2025 to March 2025).

To the best of my knowledge and belief, the matter embodied in this Project Report is an original and authentic work and has not been submitted earlier for the award of any degree or diploma.

Authorised Signatory

PREFACE

"Almost a century ago a spate of invention ushered into the world first through the industrial revolution within a short span of time many countries became industrialized."
Now the world is in another industrial revolution. The major causes of first industrial Revolution were the invention of many mechanical devices and machines. The cause of second industrial revolution is electronic devices. Man invented many electronic devices and the computer is one of these but the computer has made a greater impact on society than any other single device and became major cause of second industrial revolution .They have made sufficient significant contribution to the society during the last three decades .Now no country can assume these developments without the use of the computers .They are versatile and are almost indispensable to engineers, scientists, business, executives, managers, administrators, accountants, teachers, and students.

A computer has some powerful features and characteristics as following---

- Perform complex and respective calculations at a very high speed and accuracy.
- Store large amount of data, information and subsequent manipulation.
- Make decisions under predefined instructions and conditions.
- Provide information to the user.
- Automatically correct or modify certain parameters of any system under control.
- Draw and print graphs.
- Print information and record effectively.
- Converse the user through terminal.

Though the computers are used in many areas but now a days 80% use of computer and automation of industries and business system is prevalent. So, keeping the above facts in mind we prepare our project report.

ACKNOWLEDGEMENT

I feel very privileged that through these words I be able to flaunt my thought and show my gratitude and a sense deep contentment for the people who have been part of this project in numerous ways. From those people I have got relentless support right from the stage the project idea was conceived to its completion. In Particular, I would like to shower special praise on my guide Dr. Neha and I am very thankful for his endless help during the development of entire project.

There are times in such projects when the clock beats you time and you run out of energy and you just want to finish it once and forever. My near & dear one's inducing self-belief in me to endure adverse times and make me mentally strong to cope with any situation instead of succumbing to pressure.

With great pleasure, I would like to express my sincere gratitude to **Dr. Neha, Assistant Professor, BMU** for guidance during the period.

I want to give my heartiest thanks to my parents whose principle and morale helped me to rise to this position. I would also like to thank all the friends who provided useful suggestions time to time. I also like to thank almighty god for what he has given till today.

ABSTRACT

As the name specifies “HOSTEL MANAGEMENT SYSTEM” is a software developed for managing various activities in the hostel. For the past few years, the number of educational institutions is increasing rapidly. Thereby the number of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system Which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

· Less human error

· Strength and strain of manual labour can be reduced

· High security

· Data redundancy can be avoided to some extent

· Data consistency

· Easy to handle

· Easy data updating

· Easy record keeping

· Backup data can be easily generated

TABLE OF CONTENTS

Index:

- **OBJECTIVE AND SCOPE**
- **DEFINITION OF THE PROBLEM**
- **INTRODUCTION TO SQL DATABASE**
- **INTRODUCTION TO JAVA**
- **SOFTWARE ENGINEERING PARADIGM**
- **SOFTWARE ENGINEERING METHODOLOGY AND PROCESS MODEL**
- **ESTIMATION**
- **PERT CHART EXPLANATION**
- **PERT CHART**
- **GANTT CHART**
- **COST BENEFIT ANALYSIS**
- **ENTITY RELATIONSHIP MODEL**
- **DATA FLOW DIAGRAM**
- **HARDWARE AND SOFTWARE REQUIREMENT**
- **PROGRAM FLOW CHART**
- **EVOLUTION**
- **TESTING TECHNIQUE & STRATEGIES**
- **IMPLEMENTATION**
- **MAINTENANCE**
- **CODE EFFICIENCY**
- **DATA DICTIONARY**
- **OUTPUT SCREENS AND CODING**
- **CONCLUSION**
- **BIBLOGRAPHY AND REFERENCE**

OBJECTIVE AND SCOPE

Objective:

The chief objective of "Hostel Management System"- for different hostel is to store the entire database at one place. From this the record retrieval, storing and manipulation are fast and auditing can be handled quite easily.

The services like check-in, registration, booking, checkout, room information, employee details, salary etc and queries of customer and other concerned persons are answered immediately by computerized system. We can reduce the manpower, which is used to maintain the record.

Scope:

The scope of hostel management is very vast. It includes; efficiency of the hostels, securing benefits of the transactions through practical measures, clarification of the functions of the hostels, coordination of the powerful programs, sound beneficial planning, good direction, efficient and systematic execution. It provides close collaboration and sense of sharing responsibilities, organized purpose and dynamic approach.

Any organization plays a vital role in the life of human being. It plays different functions like; brings efficiency, guide people to receive right direction from the right managers, enables the employees to get profit from their hard work, bring coordination of all members. It provides well defined policies and programs, favorable hostel running situation, growth and development of human beings, make use of appropriate materials, effective development of human qualities, execution of the programs, arrangement of the activities, efforts for attainment of the objectives etc.

In hostel organization, there is a great role of economy, men, and material to achieve the desired objectives. Following there is a great role of different subject which is called the scope in the hostel organization.

DEFINITION OF THE PROBLEM

Generally, in now days, all the banks take a step to computerized their working environment because traditional working that is manual is tough and time consuming. Generally, the record searching or verification of a student is a cumbersome process, for the entire working separate registers, ledgers etc are to be maintained.

Auditing process by management level or higher authorities is very tedious and it take a lot of time therefore we have to require a computerized system because through computers processing, searching and sorting of records is fast.

Introducing a Computerized system will reduce the burden on hostel's administrative department.

Definition:

The software system "Hostel Management System" provides different facilities according to the requirement of hostels.

Responsibilities overview:

My prime responsibility is that the performance of software never erodes in any environment; each and every query will respond in an optimal time and satiate the prospective seeker. All the responses are carried out in a minimum time in each particular situation.

INTRODUCTION TO JAVA

JAVA was developed by James Gosling at Sun Microsystems Inc in the year 1991, later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs.

Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to C/C++.

History

Java's history is very interesting. It is a programming language created in 1991. James Gosling, Mike Sheridan, and Patrick Naughton, a team of Sun engineers known as the Green team initiated the Java language in 1991. Sun Microsystems released its first public implementation in 1996 as Java 1.0. It provides no-cost -run-times on popular platforms. Java1.0 compiler was re-written in Java by Arthur Van Hoff to strictly comply with its specifications. With the arrival of Java 2, new versions had multiple configurations built for different types of platforms.

The principles for creating java were simple, robust, secured, high performance, portable, multi-threaded, interpreted, dynamic, etc. James Gosling in 1995 developed Java, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

INTRODUCTION TO SQL DATABASE

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

Structure Query Language (SQL) is a programming language used for storing and managing data in RDBMS. SQL was the first commercial language introduced for E.F Codd's Relational model. Today almost all RDBMS (MySQL, Oracle, INFOMIX, Sybase, MS Access) uses SQL as the standard database language. SQL is used to perform all type of data operations in RDBMS.

SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks."[10] Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and Data Control Language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SOFTWARE ENGINEERING PARADIGM

Software engineering occurs as a consequence of a process called "System Engineering" instead of concentrating solely on software, system engineering focuses on a variety of elements, analysing, designing and organizing those elements into a system that can be a product, service or a technology for the transformation of information or control. The system engineering process takes on different forms depending on the application domain in which it is applied.

A software system exists for one reason: to provide value to its users. All decision should be made with this in mind.

Keep It Simple,

Software design is not a haphazard process. There are many factors to consider in any design effort. All design should be as simple as possible but no simpler. These facilities having a more easily understood and easily maintained system. This is not to say that features, even internal features should be discarded in the name of simplicity. Indeed, the more elegant designs are usually the simple ones. Simple also does not mean "quick and dirty". In fact, it often takes a lot of thought and work over multiple iterations to simplify. The payoff is software that is more maintainable and less error prone.

Maintain the vision:

A clear vision is essential to the success of a software project. Without one, a project almost unfailingly ends up being "of two minds" about itself. Without conceptual integrity, a system threatens to become a patchwork of incompatible designs, held together by the wrong kind of screws. Compromising the architectural vision of a software system weakens and will eventually break down the well-designed system.

What You Produce, Others will consume:

The audience for any project of software development is potentially large. So always specify, design, and implement knowing someone else will have

to understand what you are doing. Design keeping the implementers in mind. Code with concern for those who must maintain and extend the system. Someone may have to debug the code you write, and that makes them a user of code. Making their job easier adds value to the system.

Be Open to the Future:

A system with a long lifeline has more value. In today's computing environment, were specifications change on a moment's notice and hardware platform are obsolete after just a few months, software lifetimes are typically measured in months instead of years. However true "industrial-strength" software systems must endure for longer. To do this successfully, these systems must be ready to adapt to these and other changes. Systems that do this successfully are those that have been designed this way from the start. Never design yourself into a corner. Always ask "what-if" and prepare for all possible answers by creating systems that solve the general problem, not just the specific one.

Plan Ahead for Reuse:

Reuse saves time and effort. Achieving a high level of reuse is arguably the hardest goal to accomplish in developing a software system. The reuse of code designs has been proclaimed as a major benefit of using object-oriented technologies. However, the return on this investment is not automatic. To leverage the reuse possibilities that object-oriented programming provides require fore thought and planning. There are many techniques to realize reuse at every level of the system development process. Those at the detailed designs and code level are well known and documented. New literature is addressing the reuse of design in the form of software patterns. However, this is just part of the battle. Communicating opportunities for reuse to others in the organization is paramount. How can you reuse something that you don't know exists? Planning ahead for reuse reduces the cost and increases the value of both the reusable components and the systems into which they are incorporated.

Think:

Placing clear, complete thought before action almost always produces better results. When you think about something, you are more likely to do it right. You also gain knowledge about to do it right again. If you do think about something and still do it wrong, it becomes valuable experience. Aside effect of thinking is learning to recognize when you don't know something, at which point you can research the answer. When clear thought has gone into a system, value comes out.

Software has become the key element in the evolution of computer-based system and products and one of the most important technologies on the world stage. Over the past 50 years. Software has evolved from a specialized problem solving and information analysis tool to an industry in itself. Yet we still have trouble developing high quality software on time and within budget. Software-programs, data and documents-addresses a wide array of technology and application areas, yet all software evolves according to a set of laws that remained the same for over 0 years. The intent of software engineering is to provide a framework for building higher quality software.

Software engineering is a discipline that integrates process, methods, and tools for the development of computer software. All the different process models for software engineering define a set of umbrella activities that span the entire process. Personal and team models for the software process have been proposed, both emphasize measurement, planning, and self-direction as key ingredients for successful software process.

An agile philosophy for software engineering stresses four key issues:

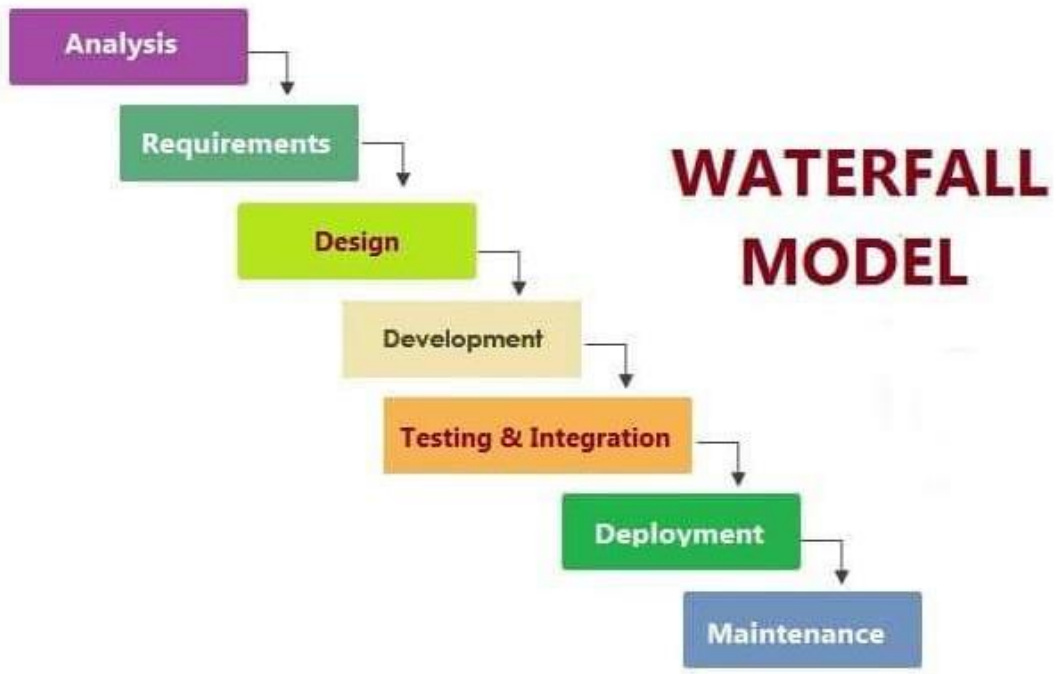
- The importance of self-organizing teams that have control over the work they perform.
- Communication and collaboration between team members and between practitioners and their customers.
- Recognition that changes represents an opportunity
- An emphasis on rapid delivery of software that satisfies the customer.

SOFTWARE ENGINEERING METHODOLOGY AND PROCESS MODEL

A production process of software is a sequence of steps. Each step performs a well-defined activity leading toward the satisfaction of the project goals with the output of one step forming the input of the next one. How to perform the activity of the particular step or phrase is an issue addressed by methodologies for that activity.

The process model I have used is Waterfall Model. In these phases are organized in a linear order.

Project begins with feasibility analysis and project planning begins. The design starts after analysis is complete and coding begins after the design is complete. Then coding is integrated and testing is done. After this regular operation and maintenance of the system takes place.



Waterfall Model

ESTIMATION

Software is the most expensive elements of virtually all computer-based systems. For example, custom systems; a large cost estimation error can make the difference between profit and loss. Cost overrun can be disastrous for the developer. Although software engineering effort is a dominant elements of project cost. It is important to remember that others costs (e.g., development environment and tools, travel, training, office space, hardware) must also be considered.

Software cost and effort estimation will never be an exact science, but a combination of good historical data and systematic technique can improve estimation accuracy. Too many variable –human, technical, environmental, political-can affect the ultimate cost of software and effort applied to develop it. However, software project estimation can be transformed from a black art to series of systematic steps that provide estimates with acceptable risk. To achieve reliable cost and effort estimates, a number of options arise.

- Delay estimation until late in the project (obviously we can achieve 100% accurate estimates after the project is completed).
- Base estimates on similar projects that have already been completed.
- Use relatively simple decomposition techniques to generate project cost and effort estimates.
- Use one or more empirical method for software cost and effort estimation.

Unfortunately, the first option however attractive is not practical. Cost estimates must be provided "up front". However, we should recognize that longer we wait, the more we know, the less likely we are to make serious errors in our estimates.

The second option can work reasonably well, if the current project is quite similar to past efforts and other project influences are roughly equivalent. Unfortunately, past experience has not always been a good indicator of future results.

The remaining options are viable approaches to software project estimation Ideally, the technique noted for each option should be applied in tandem; each used a cross check for the other. Decomposition techniques take a “divide and conquer” approaches to software project estimation. By decomposing a project into major functions and related software engineering’s activities, cost and effort estimation can be performed in a stepwise fashion.

Empirical estimation models can be used to complement decomposition techniques and offer a potentially valuable estimation approach in their own right.

Each of the viable software cost estimation options is only as good as the historical data used to seed the estimate. If no historical data exists, costing rests on a very shaky foundation.

PERT CHART EXPLANATION

The Program Evaluation and review Technique commonly abbreviated as PERT is a model for project management invented by United States Department of Defense's US Navy special projects Office in 1958 as part of the Polaris mobile submarine-launched ballistic missile project. The project was a direct response to the sputnik crisis.

PERT is basically a project scheduling method for analysing the tasks involved in completing a given project, especially the time needed to complete each task, and identifying the minimum time needed to complete the total project. It is driven by the information aid on already developed in earlier project planning activities:

- Estimates of effort
- A decomposition of the appropriate process model and task set
- Decomposition of tasks

Interdependencies among tasks may be defined using a task network. tasks sometimes called the project work breakdown structure (WBS), are defined for the product as a whole or for individual functions.

PERT provide quantitative tools that allow the software planner to

- Determines that the critical path-the chain of tasks that determines the duration of the project
- Establish 'most likely" time estimates for individual tasks by applying statistical models.
- Calculate "boundary times" that define a time "window' for a particular task

The objective of project scheduling tools is to enable a project manager to define work tasks, establish their dependencies, assign human resources to tasks, and develop a variety of graphs, charts, and tables that aid in tracking and control of the software project. It was able to incorporate uncertainty in the sense that

It was possible to schedule a project not knowing precisely the details and durations of all the activities. It is more of an event-oriented technique rather than start and completion oriented.

TERMINOLOGY USED IN PERT CHART

The various terms and terminology used in PERT CHART and their meaning are as follows:

ANALYSIS:

In analysis phase detailed study of user's requirements has been done and then gets the idea of the proposed system to build.

DESIGN:

There are various type designs:

1. External design
2. Internal design
3. Architectural design

In external designing phase externally observable characteristics of software was conceived, planned out and specified. These characteristics include user displays, report format; external data source etc. External design was started during analysis phase.

INTERNAL DESIGN:

In internal designing phase following things has been done. Specification of internal structure and details, elaboration of test plans and development of a blueprint.

ARCHITECTURAL DESIGN:

In architectural designing phase following things has been done. Identification of internal processing functions, decomposition high level functions into publications, definition of internal data streams and datastores, establishment of relationships among functions, data streams and datastores.

CODING:

In the coding phase proper are written which performs the required tasks.

TESTING:

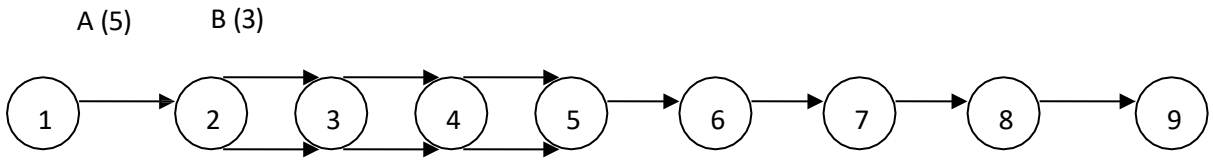
1. UNIT TESTING: In unit testing individual modules are tested. It means internal logic has been checked and required changes have been made in the code.

2. INTEGRATED TESTING: In this phase modules have been integrated and it is checked that after integration these modules are working effectively or not.

3. ACCEPTANCE TESTING: In acceptance testing it is checked that whether the software is fulfilling the user's requirement or not.

PERT CHART

	Activities	Predecessor Activity	Duration (Days
A	Preparation for designing	_	5
B	Designing DFD & ERD	A	3
C	. Designing Structure Chart	A	2
D	Designing Input Forms Layouts	B, C	15
E	Designing Report Layouts	B, C	15
F	Testing and Verification of forms	D, E	7
G	Designing Database Files	D, E	10
H	Designing Flow Chart	G, H	15
I	Coding	H	60
J	Testing	J, I	5
K	Implementation	J	5



Total Development Time=5+3=2+10+15+7+15+15+60+5+5=142 Days

GANTT CHART

A Gantt chart is a popular type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Some Gantt charts also show the dependency (i.e. Precedence network) relationships between activities. Basically, it is an activity scheduling method as a rudimentary aid to plot individual tasks against time.

The initial format of the chart was developed by Henry Gantt (1861-1919) in 1910.The Advent of personal computers in the 1980s resulted in the ability to produce and edit elaborate Gantt charts. These desktop applications were primarily intended for project managers and project schedulers. In the late 1990s and 2000s, Gantt charts became a common feature of web-based applications, including collaborative groupware. Although now considered a common charting technique, Gnatt charts were considered quite revolutionary at the time they were introduced. In recognition of Henry Gantt's contributions, the Henry Laurence Gantt medal is awarded for distinguished achievement in management and services to the community.

Gantt charts have become a common technique for representing the phases and activities of a project work breakdown structure, so a wide audience can understand them. A Gantt chart is easily comprehended for small projects that fit on a single sheet or screen, but they can become quite unwisely for projects that fit on a single sheet or screen, but they can become quite unwisely for projects with more than about 30 activities. Larger Gantt charts may not be suitable for most computer displays. A related criticism is that Gantt charts communicate relatively little information per unit area of display. That is, projects are often considerably more complex than can be communicated effectively with a Gantt chart. They only represent part of the triple constraints of projects because they focus primarily on schedule management.

Although project management software can show schedule dependencies as lines between activities, displaying a large number odd dependencies may result in a cluttered or unreadable chart.

COST BENEFIT ANALYSIS

With the help of this project a lot of time and money can be saved. Because when the task of producing reports will be accomplished through this project then it will require less amount of money.

While making this project cost benefit analysis has been done and following conclusions have been drawn.

Through this project via one time investment a specific task can be done several times. If we will perform same task manually then it will require a lot of cost and time and obviously user will have to do same sort of task again and again.

Cost of Hardware and software required for this project have to be paid only once but if we use manpower or some other option for this purpose then it will be very expensive and this is obviously not required.

Cost Benefit analysis is a tool for evaluating the effectiveness of the project. It may be used by the management. The objective of cost Benefit analysis is to find out whether it is economically worthwhile to invest in the proposed project. If the return on the investment is high then the project is considered economically worthwhile.

The three types of cost benefits are-

1. Tangible or Intangible cost and benefits.
2. Fixed or Variable cost and benefits.
3. Direct or Indirect cost and benefits

ENTITY RELATIONSHIP MODEL

In computer science, an entity-relationship model (ERM) is a model providing a high-level description of a conceptual data model. Data modelling provides a graphical notation for representing such data models in the form of entity-relationship diagrams (ERD). The first stage of information system design uses these models to describe information needs or the type of information that is to be stored in a database during the requirement analysis. The data modelling technique can be used to describe any ontology (i.e., an overview and classifications of used terms and their relationships) for a certain universe of discourse (i.e. area of interest). In the case of the design of an information system that is based on the database, the conceptual data model is at a later stage (usually called logical design), mapped to a logical model, such as the relational model, this in turn is mapped to a physical model ical during "physical design.".

There are a number of conventions for entry-relationship diagrams (ERDs). The classical notation is described in the remainder of this article, and mainly relates to conceptual modelling. There are a range of notations more typically employed in logical and physical database design, including information engineering, IDEF1x (ICAM Definition Language) and dimensional modelling.

An entity represents a discrete object. Entities can be thought of (roughly) as nouns. Example: a computer, an employee, a song, a mathematical theorem. A relationship captures how two or more entities are related to one another. Relationships can be thought of (again, roughly) as verbs. Examples: an owns relation between a company and a computer, a supervises relation between an employee and a departmental performs relation between an artist and a song, a proved relation between a mathematician and a theorem. Entities are drawn as rectangles, relationships as diamonds.

Entities and relationships can both have attributes. Examples: an employee entity might have a social security number (in the US); the proved relation may have a date attribute. Attribute are drawn as ovals connected to their owning entity sets by a line. Every entity (unless it is a weak entity) must have a minimal set of uniquely identifying attributes. This set is called the

entity's primary key.Entity-Relationship diagrams don't show single entities or single instances of relations. Rather, they show entity sets and relationship sets displayed as rectangles and diamonds respectively). Example: a particular song is an entity. The collection of all songs in a database is an entity set. The proved relationship between Andrew Wiles and Fermat's last theorem is a single relationship. The set of all such mathematicians-theorem relationships in a database is a relationship set.

Lines are drawn between entity sets and the relationship sets they involved in. If all entities set must participate in a relation in the relationship set, a thick line is drawn. This is called a participation constraint. If each entity of the entity set can participate in at most one relationship in the relationship set, an arrow is drawn from the entity set to the relationship set. This is called a key constraint. To indicate that each entity set is involved in exactly one relationship, a thick arrow is drawn. Associative entity is used to solve the problem of two entities with a many-to-many relationship.

Unary relationship – A unary relationship is a relationship between the rows of a single table.

A weak entity is an entity that can't be uniquely identified by its own attributes alone, and therefore must use as its primary key both its own attributes and the primary key of an entity it is related to. A weak entity set is indicated by a bold rectangle (the entity) connected by a bold arrow to a bold diamond (the relationship). Double lines can be used instead of bold ones.

Attribute in an ER model may be further described as multi-valued, composite or derived. A multi-valued attribute, illustrated with a double-line ellipse, may have more than one value for at least one instance of its entity. For example, a piece of software (entity=application) may have the multivalued attribute "platform" because at least one instance of that entity runs on more than one operating system. A composite attribute may itself contain two or more attributes and it indicated as having at least contributing attributes of its own. For example, addresses usually are composite attributes, composed of attributes such as street address, city and so forth. Derived attributes are attributing whose value is entirely dependent on other attributes and are indicated by dashed ellipses. For example, if we have an employee database with an employee entity with an age attribute, the age attribute would be derived from a birth date attribute.

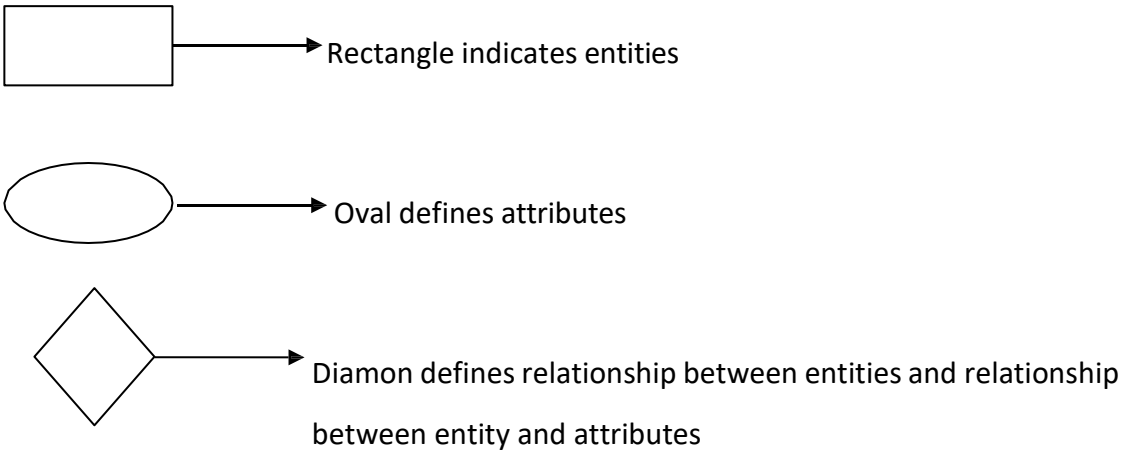
Sometimes two entities are more specific subtype of a more general type of entity. For example, programmers and marketers might both be types of employees at a software company. To indicate this, a triangle with "ISA" on the inside is drawn. The super class is connected to the point on top and two (or more) subclasses are connected to the base.

A relation and all its participating entity sets can be treated as a single entity set for the purpose of taking part in another relation through aggregation, indicated by drawing a dotted rectangle around all aggregated entities and relationships.

Classification:

Entity relationship models can be classified in BERMs (Binary Entity Relationship Model) and GERMs (General Entity Relationship Model) according to if only binary relationships are allowed. A binary relationship is a relationship between two entities. Thus, in a GERM, relationships between three or more entities are also allowed.

ERD Symbols:



DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context –level DFD first, which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modelled.

Data flow diagram were invented by Larry Constantine, the original developer of structured design, based on Martin and Estrin's data flow graph" data flow graph" model of computation. Data flow diagram (DFD) are one of the three essential perspectives of SSADM.The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system evolution. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish and how the system will be implemented. Old system dataflow diagrams can be drawn up and compared with the new systems dataflow diagrams to draw comparisons to implement a more efficient system. Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input, ultimately has an effect upon the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram.

In analyzing a business, several sets of DFDs are drawn. Initial DFDs might model the existing system (flaws and all), while later DFDs may model a solution to the problem being analyzed. For these solutions DFDs a logical and physical DFD is drawn. Physical DFDs represents physical files and transactions, while logical or conceptual DFDs can be used to represents business functions or processes.

A dataflow diagram illustrated the processes, data stores, and external entities in a business or other system and the connecting data flows.

The four components of a data flow diagram (DFD) are:

- External Entities/ Terminators/Sources/Sinks (represented by a square or oval)
- Processes (represented by a circle or rounded rectangle)
- Data Flows (represented by a arrow)
- Data Stores (represented by two parallel lines, sometimes connected by a vertical line)

External Entities/Terminators are outside of the system being modelled Terminators represent where information comes from and where it goes. In designing a system, we have no idea about what these terminators do or how they do it. Processes modify the inputs in the process of generating the outputs. Data stores represent a place in the processes, so a data store might be a place to accumulate data over a year for the annual accounting process. Data flows are how data moves between terminators, processes and data stores.

Every page in DFD should contain fewer than 10 components. If a process has more than 10 components, then one or more components (typically a process) should be combined into one and another DFD be generated that describes that component in more detail. Each component should be numbered, as should each subcomponent, and so on.

Data Process:

A data process represents the transformation of data in the system. Generally, this represents something that happens in the system. Such as "Student enrollment". Data that flows into a process should be different from the data that flows out of the process.

Data Store:

A data store is a repository for data. Data stores can be manual, digital or temporary.

External entities:

An external entity represents the source or sink of data external to the system. When modelling a DFD, the designer is not interested in the inner workings of the external entity, but only what data is produced/needed by the entity. It flows Duplication External entities and data stores can be duplicated in the system for more clarity, while processes cannot. External

entities that have been replicated are marked by an asterisk (*) in the lower right part of the oval that represents that entity. Data stores a double line of the left side of their box.

Developing a DFD:

Top-Down Approach

The system designer makes a context level DFD, which shows the interaction (data flows) between the system (represented by one process) and the system environment (represented by terminators). The system is decomposed in lower level DFD into a set of processes, data stores, and the data flows between these processes and data stores.

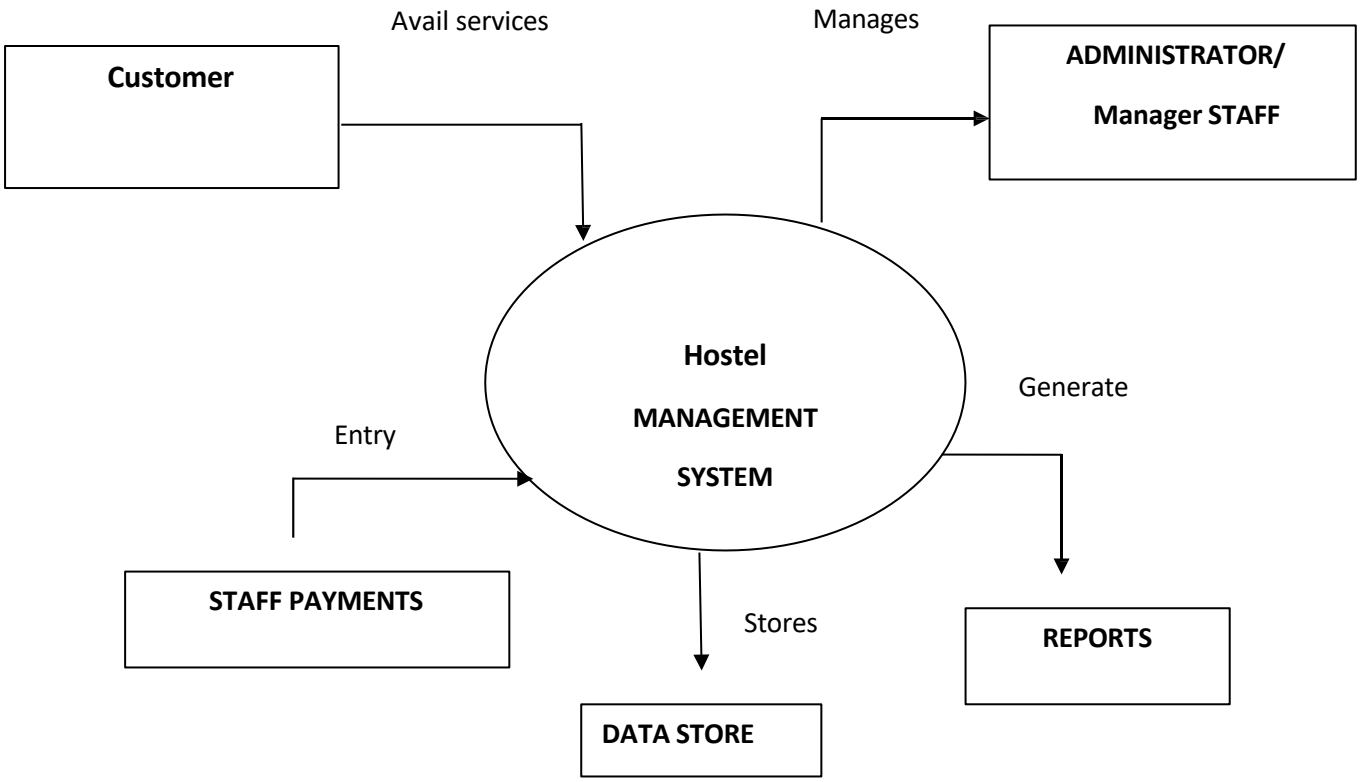
Each process is then decomposed into an even lower diagram containing its sub processes. This approach then continues on the subsequent sub processes, until a necessary and sufficient level of detail is reached.

Construct detail DFD:

The list of all events is made. For each events a process is constructed. Each process is linked (with incoming data flows) directly with other processes or via data stores, so that it has enough information to given event. The reaction of each process to a given event is modelled by an outgoing data flow.

Develop higher-level DFDs for clarity. A group of related processes is aggregated into a process of higher level. Related aggregated processes are aggregated into a process of even higher level. This process then continues until a contextual level DFD is reached.

CONTEXT LEVEL DFD



HARDWARE AND SOFTWARE REQUIREMENT

HARDWARE (Minimum Requirements):

- ✓ Pentium IV Processor
- ✓ HDD 20 GB
- ✓ RAM 128 MB
- ✓ VGA colour Monitor
- ✓ Scroll mouse
- ✓ Multimedia Keyboard

SOFTWARE (Minimum Requirements):

Front end	NetBeans IDE
Back-end	SQL Server 2008
O.S.	Microsoft windows XP

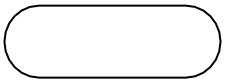
PROGRAM FLOW CHART

Anything represented in the pictorial form is easier to understand than when presented in a descriptive form. The solution to any problem is expressed as an algorithm. The Flow Chart is a graphical tool for pictorial representation of an algorithm.

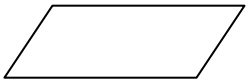
A Flowchart uses different symbols for different operations. A flowchart is not a computer program. It is just an aid to represent a program in a pictorial form. In flowchart it is read from top to bottom.

There are symbols that are used in a flow chart. Each is designed for a particular function. Flow lines, showing the order in which functions are to be carried out connect these symbols. These symbols have been recognized by ANSI for information processing.

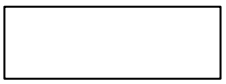
1. Terminal Symbol



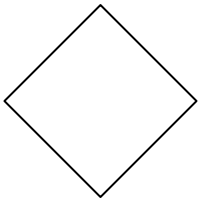
2. Input/Output Symbol



3. Processing Symbol

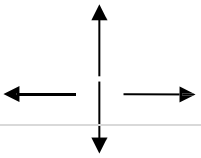


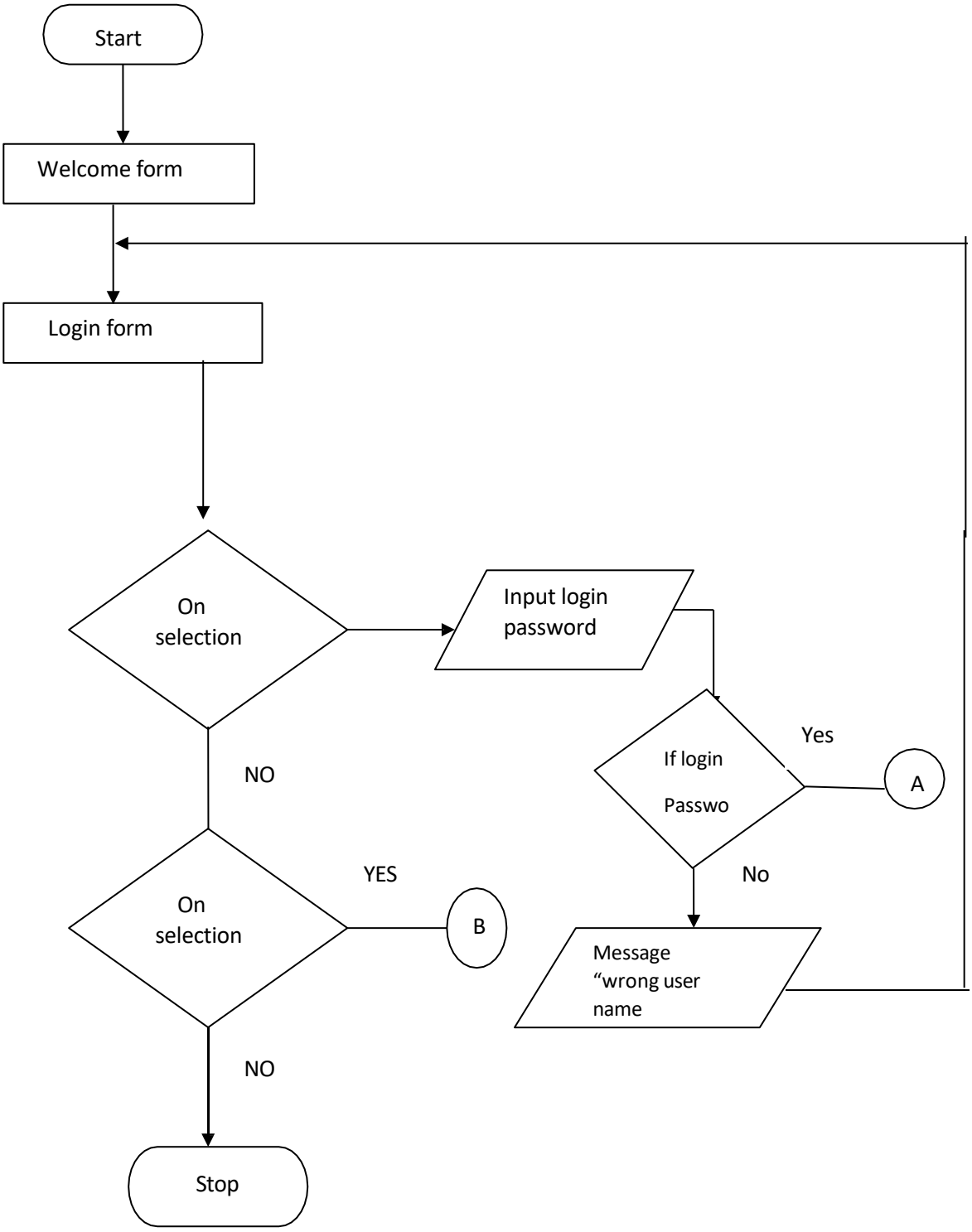
4. Decision Symbol



5. Connectors

6. Flow Lines




Start
Welcome form
Login form
On selection
Input login password
If login Passwo
Yes
A
No
Message "wrong user name
NO
On selection
YES
B
NO
Stop

EVOLUTION

Regardless of its application domain, size, or complexity, computer software will evolve over time. Change often referred to as software maintenance drives this process and occurs when errors are corrected, when the software is adapted to a new environment, when the customer requests new features or functions, and when the application is reengineered to provide benefit in a modern context.

- E-types systems are software that has been implemented in a real world-computing context and will therefore evolve over time.
- E-type must be continually adapted, or else they become progressively less satisfactory.
- As E-type system evolves its complexity increases unless work is done to maintain or reduce it.
- The E-type system evolution process is self-regulating with distribution of product and process measures close to normal.
- The average effective global act rate in an evolving E-type system is invariant over product lifetime.
- As an E-type system evolves all associated with it developers, sales, personnel, and users, for example must maintain mastery of its context and behaviour to achieve satisfactory evolution. Excessive growth diminishes that mastery. Hence the average incremental growth remains invariants as the system evolves.
- The functional context of E-type systems must be continually increased to maintain user satisfaction over the system's lifetime.
- The quality of E-type systems will appear to be declining unless they are rigorously maintained and adapted to operational environmental changes.
- E-type evolution processes continue multilevel, multiloop, multiagent systems and must be treated as such to achieve significant improvement over any reasonable base.

TESTING TECHNIQUE & STRATEGIES

Software testing accounts for the largest percentage of technical effort in the software development process. The objective of software testing is to uncover errors. To fulfill this objective, a series of test steps concentrate on functional verification of a component and incorporation of components in to the software architecture. Validation testing demonstrates trace ability to software requirements, and system testing validates software once it has been incorporated into a large system.

Each test step is accomplished through a series of systematic test techniques that assist in the design of test cases. With each testing step, the level of abstraction with which software is considered is broadened.

Unlike testing (a systematic, planned activity), debugging must be viewed as an art beginning with a symptomatic indication of a problem, the debugging activity must track down the cause of an error. Of the many resources available during

Debugging, the most valuable is caused of other members of the software engineering staff.

The requirement for higher-quality software demands a more systematic approach to testing. What is required is an overall strategy, spanning the strategic test space, quite as deliberate in its methodology as was the systematic development on which analysis, design and code were based.

Testing is vital to the success of the any system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Inadequate testing or non-testing leads to errors that may not appear until months later. This creates two problems:

The time lag between the cause and the appearance of the problem (the longer the time interval, the more complicated the problem has become)

The effect of system errors on files and records within the system.

A small system error on conceivably explodes into a much larger problem.

Effective testing early in the process translates directly into long term cost savings from a reduced number of errors.

Another reason for system testing is its utility as a user-oriented vehicle before implementation. The best program is worthless if it does not meet the user needs. Unfortunately, the user demands are often compromised by efforts to facilitate program or design efficiency in terms of processing time or memory utilization. Often the computer technician and the user have communication barriers due to different backgrounds, interest, priorities and perhaps languages. The system tester (designer, programmer or user) who has developed some computer mastery can bridge this barrier.

The planned test of a system should include a thorough auditing technique and introduce control elements unique to the system. The auditor's role is to judge the controls of the system and maintain an audit trail to ensure the integrity, reliability, and confidence of the user at all levels. Being included in the system development team makes it easier for the auditor to monitor testing procedures and consider the acceptance of new controls to replace those changed by the new design.

The primary objective for test case design is to derive a set of tests that have the highest likelihood of uncovering errors in software. To accomplish this objective, two different categories of test case design techniques-applicable top conventional and object-oriented systems are used

White-box Testing

Black-Box Testing

White-box tests focus on the program control structure. Test cases derived to ensure that all statements in the program have been executed at least once testing and that all logical conditions have been exercised. basic path testing, a white box technique, makes use of program graphs to deliver a set of linearly independent tests that will ensure coverage. Condition and data flow testing further exercise program logic, and loop testing compliments other white-box techniques by providing a procedure for exercising loops of varying degrees of complexity.

Black-Box tests are designed to validate functional requirements without regard to the internal workings of a program. Black-Box testing techniques focus on the information domain of a program in a manner that provides thorough test coverage Equivalence partitioning divides the

input domain into classes of data that are likely to exercise specific software function. Boundary value analysis probes the program's ability to handle data at limits of acceptability.

Orthogonal array testing provides an efficient systematic method for testing systems with small number of input parameters.

Integration testing can be accomplished using a use-based strategy. Use-based testing constructs the system in layers, beginning with those classes that do not use server classes. Integration test case design methods can also random and partition tests.

Specialized testing methods encompass a broad array of software capabilities and application areas. Testing for graphical user interfaces, client/server architectures, documentation and help facilities, and real- time systems each require specialized guidelines and techniques.

Experienced software developers often say, "Testing never ends, it just gets transferred from you to your customer. Every time your customer uses the programs, a test is being conducted." By optimally applying the test case design, the software engineer can achieve more complete testing and thereby uncover and correct the highest number of errors before the "customer's tests" begin.

IMPLEMENTATION

Implementation is nothing but coding, testing and installing a developed software package on the computer system. The successful implementation of the new software package is the most important part of the system Development Life Cycle. System implementation comprises the following activity.

Creating computer compatible files

Training the operating staff

Installing the necessary hardware, other necessary devices

Installing the application package and utilities

Testing and performing audit trail

Implementing a system is just like putting the pieces of jigsaw puzzle together. We know what we want and must put it together piece by piece. We have to buy the equipment plan individual sub-systems and hire and allocate people to implement the whole system.

Traditional approach to Implementation:

Each module or small group of modules is coded tested and debugged individually.

After all the modules of the whole system are debugged, these modules are grouped into subsystems.

Subsequently each subsystem is tested and debugged.

Next the subsystems are combined to from the whole system, this is known as system integration.

Finally, the whole system is tested and debugged.

Before the proposed new system is up and running. A major concern in updating to new system is to let there be no disruption in the current functioning of the organization.

The implementation phase of software development is also concerned with translating design specification into source code. It is necessary to write source code and internal and documentation so that conformance of the code to its specification can be easily verified, and so that debugging, testing and modification can be performed easily. This can be achieved by making the source code as crystal clear and unambiguous as possible. The software implementation team should therefore be provided with detailed description of software requirements, an optimum architectural design specification and a detailed design representation.

Human behavioural factors should not be overlooked in the system implementation phase. System changes can provide unnecessary psychological stir among the potential users of the new system because they may show the signs of resistance to the new system. User training is another vital area, which is responsible for minimizing resistance to change which is inevitable and giving the new system a chance to proof its worth. Training aids, such as user –friendly manuals, data dictionary, and jobs performance enhancing aids that communicate information about the new system provides the user with a good start on the new system.

Conversion:

Conversion begins with the review of the project plan, implementation plan and system test documentation. The project team and programmers along with the users do review of these documents.

The conversion portion of the implementation plan is ultimately approved

The conversion method to be followed is decided and applied.

Required files are converted.

Output generated and operations by the new system are recorded on a special form. These are documented for future reference.

If no difficulties are encountered with the new system, it is allowed to continue its operations in the organizations.

With this the conversion is completed and plans for the post – implementation review is prepared.

After the review, if any modifications are suggested, these are implemented.

Following this, the new system is officially declared as operational.

Conversion Control:

One method for gaining control of the conversion is to use well-planned test files for testing the new system. Test file is nothing but a program that will test the new system using some test data, before the system is tested with live data. Test files must have the ability to generate initially a small number of simple records. Progressively records, which are challenging to the new system, should be generated.

The test file should provide previously determined output results to compare with those produced by the new system.

Data entry and audit control:

Many systems fail due to inadequate attention given to data entry control or protective features such as audit control. These must be part of overall plan for conversion. Before a data entry operator starts working, a data entry validation program can be written to keep track of wrong data entered. In order to exercise data entry control, number of records that can be keyed in must be conveyed to the operator. The data entry program should possess the provision for log. In order to exercise audit control trail, the user should have a copy of additions or deletions to any file. A good audit control trail is the key to detecting fraud and errors in data entry.

At the time installation of the software, it is required that SQL server must be installed on the server of the company and it must contain the data files related to the software before implementing the developed system on all machines machine connected to the server. For this the exported data

is imported to the client machine and the database is checked that all tables are properly imported to the client machine or not as the software is developed in SQL server data base and also going to be implemented in oracle there is no conversion is required.

User training:

User manual is the most potent training aid to impart information of the proposed new system to the prospective users. The user manual illustrates in easy manner, the different steps required to accomplish the task the user is intended to perform. A user manual provides essential information required to operate a system in a short and concise format. It may include the following contents:

This project is having three modules:

Hostel Management Module

Staff Module

Reports Module

MAINTENANCE

Maintenance is the enigma of system development. It holds the software industry captive, trying up programming resources. Analyst and programmers spend far more time maintaining programs than they do writing them.

Maintenance covers a wide range of activities, including correcting coding and designs errors, updating documentation and test data, and upgrading user support. More often than not most activities classified as maintenance are actually enhancements. Unlike hardware, software does not wear out; it is corrected. Although software does not war out like a piece of hardware; it ages and eventually fails to perform because of cumulative maintenance. Over time the integrity of the program, test data, and documentation degenerates as a result of modifications. Eventually, it takes more effort to maintain the application than to rewrite it.

A major problem with software maintenance is the labor-intensive nature and therefore likely hood of errors. It is error prone process that is still perceived by many as more cost effective than writing programs from scratch.

Maintenance activities begin where conversion leaves off and it handled by the same planning and control used in a formal system project. Documentation is as much a part of maintenance as it is of system development. Briefly, the maintenance staff receives a request for service from an authorized user, followed by a definition of the required modifications. The source program and written procedures for the system are acquired from the programming library. Program changes are then tested and submitted to the user for approval. Once approved, the modification documentation is filled with the library and a project complication notice is sent to the user, signaling the termination of the project.

Once a system has been installed and is fully operational, the system's maintenance task begins. It involves keeping the software operational, that is undertaking preventive measures to keep computer running, monitoring the system's operation, fixing problems, if any, and modifying programs in response to new user requirements if any

System maintenance is of the following three types:

- Corrective Maintenance means repairing processing or performance failures or making changes because of previously uncovered problems or false assumptions.
- Adaptive Maintenance means changing the program function.
- Perfective Maintenance enhancing the performance or modifying the program to respond to the user's additional or changing needs.

Of these three more time and money are spent on perfective than on corrective and adaptive maintenance together.

In summary the work of maintenance is certainly not for neophyte programmers. To put maintenance in its proper perspective requires considerable skill and experience and is an important and ongoing aspect of system development. The ability of the maintenance programmer to make sound judgement depends on his/her technical expertise and ability to identified user needs. This means understanding the operational environment and what the user is trying to accomplish with the software. Maintenance demands more orientation and training than any other programming activities.

Maintenance is expensive. One way to reduce maintenance costs is through maintenance management and software modification audits. Software modifications consists of program rewrites system level updates, and re-audits of low-ranking programs to verify and correct the soft spots. The outcome should be more reliable software, a reduced maintenance backlog, and higher satisfaction and morale among the maintenance staff.

CODE EFFICIENCY

It has been realized umpteen times that good style can overcome many of the deficiencies of a selected programming language, while poor styles ca wreak havoc on the intent of the same language. The goal of a good coding style is to provide easily understood, straightforward and elegant code. writing code with efficiency requires a combination of experience, an eye for detail, and a basic understanding of the architecture of the language and how processors work. the guidelines for coding include:

- Constraint your algorithms by following structured programming practice
- Minimum Use of unconditional branching statement i.e., "goto"
- Keep conditional logic as simple as possible
- Understand the software architecture and create interfaces that are consistent with it.

In computer science, code efficiency is used to describe several desirable properties of an algorithm or other construct. Besides clean design, functionality, etc. Code efficiency is generally contained in two properties; speed, (the time it takes for an operation to complete), and space, (the memory or non-volatile storage used up by the construct). Optimization is the process of making code as efficient as possible ,sometimes focusing on space at the cost speed or vice versa.

The speed of an algorithm is measured in various ways. The most common method uses time complexity to determine the Big-O of an algorithm: often, it is possible to make an algorithm faster at the expense of space. This is the case whenever you cache the result of an expensive calculation rather than recalculating it on demand. This is a very common method of improving speed, so much so that language often add special features to support it, such as cc++ mutable keyword.

The space of an algorithm is actually two separate but related things. The first part is the space taken up by the compiled executable on disk (or equivalent, depending on the hardware and language) by the algorithm.

mechanisms (such as virtual functions and run-time type information) over certain compile-time decision making mechanism

(Such as macro substitution and templates). This, however, comes at the cost of speed.

The other part of algorithm space measurement is the amount of temporary memory taken up during processing. For example, pre-caching results, as mentioned earlier, improves speed at the cost of this attribute.

Optimization of algorithms frequently depends on the properties of the machine the algorithm will be executed on. For example, one might optimize code for time efficiency in applications for home computers with sizable amounts of memory, while code to be placed in small, memory-tight devices may have to be made to run slower to conserve space.

One simple way to determine whether an optimization is worthwhile is as follows: Let the original time and space requirements (generally in Big-O notation) of the algorithm be O1 and O2. Let the new code require N1 and N2 time and space respectively. If N1N2<O1O2, the optimization should be carried out. However, as mentioned above, this may not always be true.

One must be careful, in the pursuit of good coding style, not to over-emphasize efficiency. Nearly all of the time, a clean and usable design is much more important than a fast, small design. There are exceptions to this rule (such as embedded systems, where space is tight and processing power minimal) but these are rarer than one might expect.

DATA DICTIONARY

The logical characteristics of current systems data stores, including name, description, aliases, contents, and organization, identifies processes where the data are used and where immediate access to information required, serves as the basis for identifying database requirements during system design.

Uses of Data Dictionary:

1. To manage the details in large systems.
2. To communicate a common meaning for all system elements.
3. To Document the features of the system.
4. To facilitate analysis of the details in order to evaluate characteristics and determine where system changes should be made.
5. To locate errors and omissions in the system.

OUTPUT SCREENS AND CODING

LOGIN PAGE -

CODE FOR THE ABOVE PAGE -

```
package hostel;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {
```

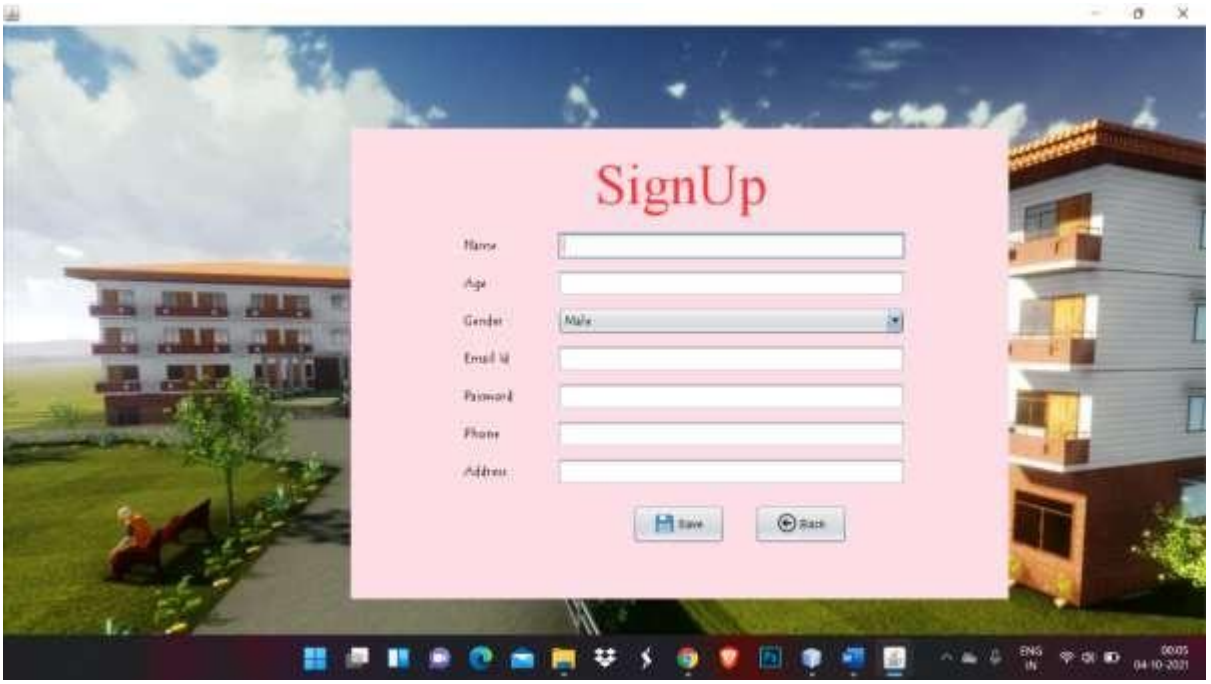
```
public Login() {

    initComponents();
}



@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    show_password = new javax.swing.JCheckBox();
    txt_user_name = new javax.swing.JTextField();
    txt_password = new javax.swing.JPasswordField();
    jLabel3 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel2.setFont(new java.awt.Font("Times New Roman", 0, 16)); // NOI18N
    jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/login ani.gif"))); // NOI18N
    getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(670, 190, 350, 120));

    jLabel1.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
    jLabel1.setText("User Name");
    getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(680, 320, 90, 30));

    jLabel4.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
    jLabel4.setText("Password");
    getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 380, 80, 30));

    jButton1.setFont(new java.awt.Font("Times New Roman", 1, 16)); // NOI18N
    jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/login.png"))); // NOI18N
    jButton1.setText("Login");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });
    getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(790, 490, -1, -1));
    jButton2.setFont(new java.awt.Font("Times New Roman", 1, 16)); // NOI18N
```

```
        jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/add.png"))); // NOI18N
        jButton2.setText("SignUp");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(890, 490, -1, -1));

        show_password.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
        show_password.setText("Show Password");
        show_password.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                show_passwordActionPerformed(evt);
            }
        });
        getContentPane().add(show_password, new org.netbeans.lib.awtextra.AbsoluteConstraints(790, 430, 130, 30));

        txt_user_name.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
        getContentPane().add(txt_user_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(780, 320, 200, 30));

        txt_password.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
        getContentPane().add(txt_password, new org.netbeans.lib.awtextra.AbsoluteConstraints(780, 380, 200, 30));

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/login background.PNG"))); // NOI18N
        getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        pack();
    }// </editor-fold>

    private void show_passwordActionPerformed(java.awt.event.ActionEvent evt) {

        if (show_password.isSelected()) {
            txt_password.setEchoChar((char) 0);
        } else {
            txt_password.setEchoChar('*');
        }
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        String user_name, password;
        user_name = txt_user_name.getText();
        password = String.valueOf(txt_password.getPassword());
```

```
    if (user_name.equals("")) {
        JOptionPane.showMessageDialog(null, "User Name is Empty");
    } else if (password.equals("")) {
        JOptionPane.showMessageDialog(null, "Password is Empty");
    } else {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hostel management", "root", "");
            Statement st;
            st = con.createStatement();
            ResultSet rs = st.executeQuery("select * from login where name='" + user_name + "' and password='" + password + "'");
            if (rs.next()) {
                JOptionPane.showMessageDialog(null, "Login Successfully ");
                new MainPage().setVisible(true);
                setVisible(false);
            } else {
                JOptionPane.showMessageDialog(null, "Invaild User Name or password");
                txt_user_name.setText("");
                txt_password.setText("");
            }

        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage());
        }
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Signup().setVisible(true);
    setVisible(false);
}



public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
```

```
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JCheckBox show_password;
private javax.swing.JPasswordField txt_password;
private javax.swing.JTextField txt_user_name;
// End of variables declaration
}
```

SIGNUP PAGE –



CODE FOR THE ABOVE PAGE -

```
package hostel;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Signup extends javax.swing.JFrame {

    public Signup() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel3 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();

        jButton2 = new javax.swing.JButton();
        txt_email_id = new javax.swing.JTextField();
        txt_password = new javax.swing.JPasswordField();
```

```
txt_gender = new javax.swing.JComboBox<>();
txt_phone = new javax.swing.JTextField();
txt_name = new javax.swing.JTextField();
txt_address = new javax.swing.JTextField();
txt_age = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMaximumSize(new java.awt.Dimension(1366, 768));
setMinimumSize(new java.awt.Dimension(1366, 768));
getContentPane().setLayout(null);

jLabel3.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
jLabel3.setText("Password");
getContentPane().add(jLabel3);
jLabel3.setBounds(490, 380, 105, 29);

jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Back.png"))); // NOI18N
jButton1.setText("Back");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
getContentPane().add(jButton1);
jButton1.setBounds(800, 510, 98, 41);

jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/save.png"))); // NOI18N
jButton2.setText("Save");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton2ActionPerformed(evt);
    }
});
getContentPane().add(jButton2);
jButton2.setBounds(670, 510, 98, 41);

txt_email_id.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
txt_email_id.setForeground(new java.awt.Color(255, 0, 0));
getContentPane().add(txt_email_id);
txt_email_id.setBounds(590, 340, 370, 29);

txt_password.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
txt_password.setForeground(new java.awt.Color(255, 0, 0));
getContentPane().add(txt_password);
txt_password.setBounds(590, 380, 370, 29);

txt_gender.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
txt_gender.setForeground(new java.awt.Color(255, 0, 0));
txt_gender.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Male", "Female", "Other" }));
getContentPane().add(txt_gender);
txt_gender.setBounds(590, 300, 370, 29);

txt_phone.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
txt_phone.setForeground(new java.awt.Color(255, 0, 0));
getContentPane().add(txt_phone);
txt_phone.setBounds(590, 420, 370, 29);

txt_name.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
txt_name.setForeground(new java.awt.Color(255, 0, 0));
getContentPane().add(txt_name);
txt_name.setBounds(590, 220, 370, 29);

txt_address.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
txt_address.setForeground(new java.awt.Color(255, 0, 0));
getContentPane().add(txt_address);
txt_address.setBounds(590, 460, 370, 29);

txt_age.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
txt_age.setForeground(new java.awt.Color(255, 0, 0));
getContentPane().add(txt_age);
txt_age.setBounds(590, 260, 370, 29);
```

```
jLabel4.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
jLabel4.setText("Age");
getContentPane().add(jLabel4);
jLabel4.setBounds(490, 260, 105, 29);

jLabel8.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
jLabel8.setText("Name");
getContentPane().add(jLabel8);
jLabel8.setBounds(490, 220, 105, 29);

jLabel9.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
jLabel9.setText("Gender");
getContentPane().add(jLabel9);
jLabel9.setBounds(490, 300, 105, 29);

jLabel10.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
jLabel10.setText("Email Id");
getContentPane().add(jLabel10);
jLabel10.setBounds(490, 340, 105, 29);

jLabel11.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
jLabel11.setText("Address");
getContentPane().add(jLabel11);
jLabel11.setBounds(490, 460, 105, 29);

jLabel12.setFont(new java.awt.Font("Tempus Sans ITC", 0, 14)); // NOI18N
jLabel12.setText("Phone");
getContentPane().add(jLabel12);
jLabel12.setBounds(490, 420, 105, 29);

jLabel5.setFont(new java.awt.Font("Times New Roman", 0, 60)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 51, 51));
jLabel5.setText("SignUp ");
getContentPane().add(jLabel5);
jLabel5.setBounds(630, 130, 220, 80);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/pages background.jpg"))); // NOI18N
getContentPane().add(jLabel1);
jLabel1.setBounds(370, 110, 700, 500);
```

```
    jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/login background.PNG"))); // NOI18N
    getContentPane().add(jLabel2);
    jLabel2.setBounds(0, 0, 1370, 770);

    pack();
}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int ch=JOptionPane.showConfirmDialog(null,"Yes ? you go to Back","Hostel Management",JOptionPane.YES_NO_OPTION);
    if(ch==0)
    {
    new Login().setVisible(true);
    setVisible(false);
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String name,age,gender,email,password,phone,address;
    name=txt_name.getText();
    age=txt_age.getText();
    gender=(String) txt_gender.getSelectedItem();
    email=txt_email_id.getText();
    password=String.valueOf(txt_password.getPassword());
    phone=txt_phone.getText();
    address=txt_address.getText();
    if(name.equals(""))
       JOptionPane.showMessageDialog(null,"Every field should be filled");
    else if(name.equals(""))
       JOptionPane.showMessageDialog(null,"Every field should be filled");
    else if(age.equals(""))
       JOptionPane.showMessageDialog(null,"Every field should be filled");
    else if(email.equals(""))
       JOptionPane.showMessageDialog(null,"Every field should be filled");
    else if(password.equals(""))
       JOptionPane.showMessageDialog(null,"Every field should be filled");
    else if(address.equals(""))
       JOptionPane.showMessageDialog(null,"Every field should be filled");
    else
    {
```

```
    try
    {
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/hostel management","root","");
    Statement st=con.createStatement();
    st.executeUpdate("insert into login values('"+name+"','"+age+"','"+gender+"','"+email+"','"+password+"','"+phone+"','"+address+"')");
    JOptionPane.showMessageDialog(null,"Successfully Inserted");
    new Login().setVisible(true);
    setVisible(false);
    }
    catch(Exception ex)
    {
      JOptionPane.showMessageDialog(null,ex.getMessage());
    }


    }

}


 public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Signup.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
```

```
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Signup().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JTextField txt_address;
    private javax.swing.JTextField txt_age;
    private javax.swing.JTextField txt_email_id;
    private javax.swing.JComboBox<String> txt_gender;
    private javax.swing.JTextField txt_name;
    private javax.swing.JPasswordField txt_password;
    private javax.swing.JTextField txt_phone;
    // End of variables declaration
}
```

NAVIGATION PAGE –

CODE FOR THE ABOVE PAGE -

```
package hostel;

import java.awt.Color;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JOptionPane;


public class MainPage extends javax.swing.JFrame {
public static int open=0;

    public MainPage() {
        initComponents();

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
```

```
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();
jButton6 = new javax.swing.JButton();
jButton11 = new javax.swing.JButton();
jButton13 = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMaximumSize(new java.awt.Dimension(1366, 768));
setMinimumSize(new java.awt.Dimension(1366, 768));
setUndecorated(true);
addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        formFocusGained(evt);
    }
});
addWindowFocusListener(new java.awt.event.WindowFocusListener() {
    public void windowGainedFocus(java.awt.event.WindowEvent evt) {
        formWindowGainedFocus(evt);
    }
    public void windowLostFocus(java.awt.event.WindowEvent evt) {
    }
});
```

```
        getContentPane().setLayout(null);

        jButton1.setFont(new java.awt.Font("Times New Roman", 0, 16)); // NOI18N
        jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/room.png"))); // NOI18N
        jButton1.setText("Manage Room");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton1);
        jButton1.setBounds(40, 640, 260, 60);
        jButton2.setFont(new java.awt.Font("Times New Roman", 0, 16)); // NOI18N
        jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/new student.png"))); // NOI18N
        jButton2.setText("New Student");
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() { public void
actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton2);
        jButton2.setBounds(40, 150, 260, 60);

        jButton3.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
        jButton3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Update & Delete Student.png"))); // NOI18N
        jButton3.setText("Update or Delete Student Details");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton3);
        jButton3.setBounds(40, 250, 260, 60);

        jButton4.setFont(new java.awt.Font("Times New Roman", 0, 16)); // NOI18N
        jButton4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/all student living.png"))); // NOI18N
        jButton4.setText("Living Student");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton4ActionPerformed(evt);
    }
});
getContentPane().add(jButton4);
jButton4.setBounds(40, 450, 260, 60);

jButton5.setFont(new java.awt.Font("Times New Roman", 0, 16)); // NOI18N
jButton5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Leaved students.png"))); // NOI18N
jButton5.setText("Leaved Student");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});
getContentPane().add(jButton5);
jButton5.setBounds(40, 540, 260, 70);
```

```
jButton6.setFont(new java.awt.Font("Times New Roman", 0, 16)); // NOI18N
jButton6.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Fees.png"))); // NOI18N
jButton6.setText("Student Fees");
jButton6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton6ActionPerformed(evt);
    }
});
getContentPane().add(jButton6);
jButton6.setBounds(40, 350, 260, 60);

jButton11.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/logout.png"))); // NOI18N
jButton11.setText("Logout");
jButton11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton11ActionPerformed(evt);
    }
});
getContentPane().add(jButton11);
jButton11.setBounds(1010, 20, 150, 70);

jButton13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Close.png"))); // NOI18N
jButton13.setText("Close");
jButton13.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton13ActionPerformed(evt);
        }
    });
    getContentPane().add(jButton13);
    jButton13.setBounds(1200, 20, 130, 70);

    jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/home background.PNG"))); // NOI18N
    getContentPane().add(jLabel1);
    jLabel1.setBounds(0, 0, 1370, 770);

    pack();
}// </editor-fold>

private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(open==0)
    {
    int ch = JOptionPane.showConfirmDialog(null, "Yes ? You want to close", "Hostel Management", JOptionPane.YES_NO_OPTION);
    if (ch == 0) {
        setVisible(false);
    }
    }
    else
    {
        JFrame frame=new JFrame();
        frame.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(frame,"One Window is alrady open");
    }
}

private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(open==0)
    {
    int ch=JOptionPane.showConfirmDialog(null,"Yes ? You want to Logout","Hostel Management",JOptionPane.YES_NO_OPTION);
    if(ch==0)
    {
        new Login().setVisible(true);
    setVisible(false);
    }
```

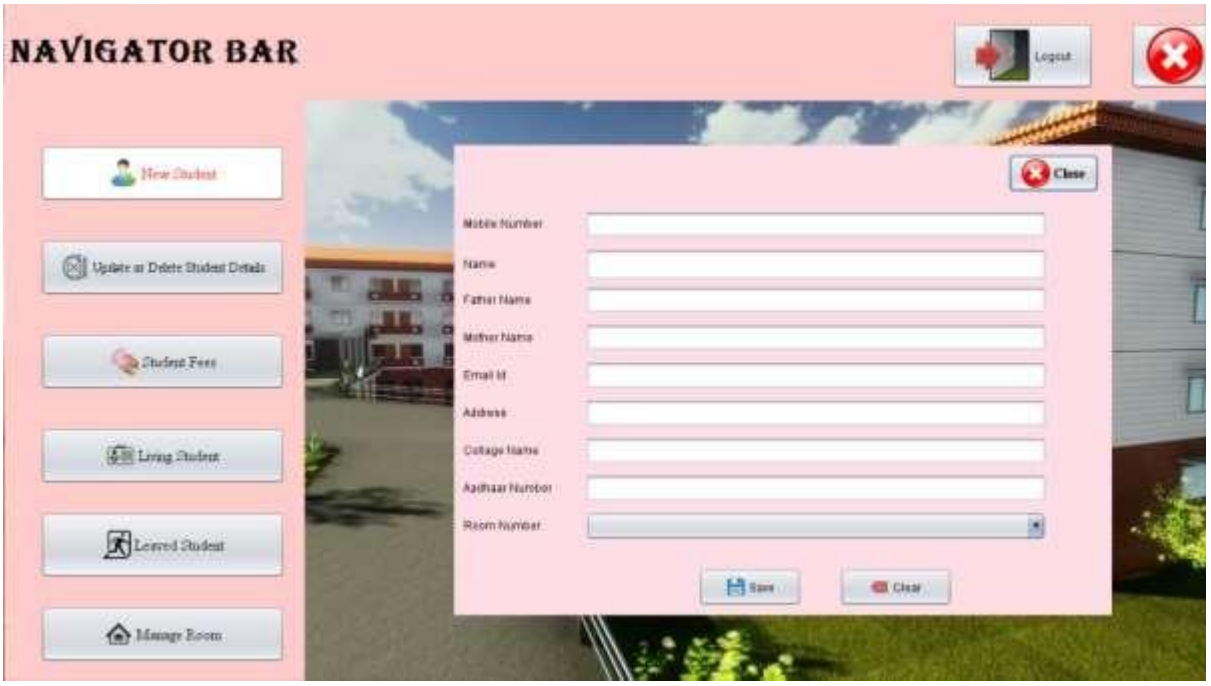
```
    }
     else
    {
        JFrame frame=new JFrame();
      frame.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(frame,"One Window is alrady open");
    }
}

  private void formFocusGained(java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:

  }
  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(open==0)
    {
    jButton1.setBackground(new Color(255,255,255));
    jButton1.setForeground(Color.red);
    new ManageRoom().setVisible(true);
    open=1;
    }
    else
    {
        JFrame frame=new JFrame();
      frame.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(frame,"One Window is alrady open");
    }
}

  private void formWindowGainedFocus(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    jButton1.setBackground(new JButton().getBackground());
    jButton1.setForeground(new JButton().getForeground());

    jButton2.setBackground(new JButton().getBackground());
    jButton2.setForeground(new JButton().getForeground());

    jButton3.setBackground(new JButton().getBackground());
    jButton3.setForeground(new JButton().getForeground());
```

```
    open=1;
    }
    else
    {
       JFrame frame=new JFrame();
     frame.setAlwaysOnTop(true);
       JOptionPane.showMessageDialog(frame,"One Window is alrady open");
    }
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(open==0)
    {
    jButton3.setBackground(new Color(255,255,255));
    jButton3.setForeground(Color.red);
    new Update_and_delete().setVisible(true);
    open=1;
    }
    else
    {
       JFrame frame=new JFrame();
     frame.setAlwaysOnTop(true);
       JOptionPane.showMessageDialog(frame,"One Window is alrady open");
    }
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     if(open==0)
    {
    jButton6.setBackground(new Color(255,255,255));
    jButton6.setForeground(Color.red);
    new StudentFees().setVisible(true);
    open=1;
    }
    else
    {
       JFrame frame=new JFrame();
     frame.setAlwaysOnTop(true);
       JOptionPane.showMessageDialog(frame,"One Window is alrady open");
```

```
    }
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     if(open==0)
    {
    jButton4.setBackground(new Color(255,255,255));
    jButton4.setForeground(Color.red);
    new LivingStudent().setVisible(true);
    open=1;
    }
    else
    {
        JFrame frame=new JFrame();
      frame.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(frame,"One Window is alrady open");
    }
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
     if(open==0)
    {
    jButton5.setBackground(new Color(255,255,255));
    jButton5.setForeground(Color.red);
    new LeavedStudent().setVisible(true);
    open=1;
    }
    else
    {
        JFrame frame=new JFrame();
      frame.setAlwaysOnTop(true);
        JOptionPane.showMessageDialog(frame,"One Window is alrady open");
    }
}



 public static void main(String args[]) {
     /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
```

```
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(MainPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(MainPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(MainPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(MainPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MainPage().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton11;
    private javax.swing.JButton jButton13;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

NEW STUDENT PAGE –



CODE FOR THE ABOVE PAGE -

```
package hostel;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class NewStudent extends javax.swing.JFrame {

    public void insertRoomNumber() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hostel management", "root", "");
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery("select * from room");
```

```
        while (rs.next()) {
            txt_room_number.addItem(rs.getString(1));
        }

    } catch (Exception ex) {

    }
}
public NewStudent() {
    initComponents();
    insertRoomNumber();
    setAlwaysOnTop(true);
}
   @SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jButton1 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    txt_address = new javax.swing.JTextField();
    txt_mobile_number = new javax.swing.JTextField();
    txt_name = new javax.swing.JTextField();
    txt_email_id = new javax.swing.JTextField();
    txt_mother_name = new javax.swing.JTextField();
    txt_father_name = new javax.swing.JTextField();
    txt_aadhaar = new javax.swing.JTextField();
    txt_collage_name = new javax.swing.JTextField();
    txt_room_number = new javax.swing.JComboBox<>();
    jButton2 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
    jLabel10 = new javax.swing.JLabel();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
setLocation(new java.awt.Point(480, 150));
setUndecorated(true);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Close all jframe.png"))); // NOI18N
jButton1.setText("Close");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 10, -1, 40));

jLabel1.setText("Name");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 112, 120, 29));

jLabel2.setText("Father Name");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 150, 120, 29));

jLabel3.setText("Address");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 270, 120, 29));

jLabel4.setText("Email Id");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 230, 120, 29));

jLabel5.setText("Collage Name");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 310, 120, 29));

jLabel6.setText("Aadhaar Number ");
getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 350, 120, 29));

jLabel7.setText("Room Number");
getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 390, 120, 29));

jLabel8.setText("Mobile Number");
getContentPane().add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 69, 120, 29));

jLabel9.setText("Mother Name");
getContentPane().add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 190, 120, 29));
```

```
        getContentPane().add(txt_address, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 270, 490, 30));

        getContentPane().add(txt_mobile_number, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 70, 490, 29));
        getContentPane().add(txt_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, 490, 34));
        getContentPane().add(txt_email_id, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 230, 490, 30));
        getContentPane().add(txt_mother_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 190, 490, 29));
        getContentPane().add(txt_father_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 150, 490, 29));
        getContentPane().add(txt_aadhaar, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 350, 490, 30));
        getContentPane().add(txt_collage_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 310, 490, 30));

        getContentPane().add(txt_room_number, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 390, 490, 30));

        jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/save.png"))); // NOI18N
        jButton2.setText("Save");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(260, 450, 110, 40));

        jButton3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/clear.png"))); // NOI18N
        jButton3.setText("Clear");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 450, 120, 40));

        jLabel10.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/pages background.jpg"))); // NOI18N
        getContentPane().add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        setAlwaysOnTop(false);
        int ch = JOptionPane.showConfirmDialog(null, "Yes ? You want to close", "Hostel Management", JOptionPane.YES_NO_OPTION);
        if (ch == 0) {
            MainPage.open = 0;
            setVisible(false);
```

```
        }

        setAlwaysOnTop(true);
    // TODO add your handling code here:
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        txt_mobile_number.setText("");
        txt_name.setText("");
        txt_father_name.setText("");
        txt_mother_name.setText("");
        txt_email_id.setText("");
        txt_address.setText("");
        txt_collage_name.setText("");
        txt_aadhaar.setText("");

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setAlwaysOnTop(false);
        String mobile, name, father, mother, email, address, collage, aadhar;
        mobile = txt_mobile_number.getText();
        name = txt_name.getText();
        father = txt_father_name.getText();
        mother = txt_mother_name.getText();
        email = txt_email_id.getText();
        address = txt_address.getText();
        collage = txt_collage_name.getText();
        aadhar = txt_aadhaar.getText();
        String room = (String) txt_room_number.getSelectedItem();
        String status="Living";
        if (mobile.equals("")) {
            JOptionPane.showMessageDialog(null, "All field are requerd");
        } else if (name.equals("")) {
            JOptionPane.showMessageDialog(null, "All field are requerd");
        } else if (father.equals("")) {
            JOptionPane.showMessageDialog(null, "All field are requerd");
        } else if (mother.equals("")) {
            JOptionPane.showMessageDialog(null, "All field are requerd");
        } else if (email.equals("")) {
```

```
            JOptionPane.showMessageDialog(null, "All field are requerd");



                    } else if (address.equals("")) { JOptionPane.showMessageDialog(null, "All field are requerd");
        } else if (collage.equals("")) {

            JOptionPane.showMessageDialog(null, "All field are requerd");

        } else if (aadhar.equals("")) {

            JOptionPane.showMessageDialog(null, "All field are requerd");

        } else {

            try {

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

                Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hostel management", "root", "");

                Statement st = con.createStatement();

                st.executeUpdate("insert into student values('" + mobile + "','" + name + "','" + father + "','" + mother + "','" + email + "','" + address + "','" + collage
+ "','" + aadhar + "','" + room + "','"+status+"')");

                JOptionPane.showMessageDialog(null, "Successfully Inserted");

                txt_mobile_number.setText("");

                txt_name.setText("");

                txt_father_name.setText("");

                txt_mother_name.setText("");

                txt_email_id.setText("");

                txt_address.setText("");

                txt_collage_name.setText("");

                txt_aadhaar.setText("");

            } catch (Exception ex) {

                JOptionPane.showMessageDialog(null, ex.getMessage());

            }

        }

        setAlwaysOnTop(true);

    }

    /**

     * @param args the command line arguments

     */

    public static void main(String args[]) {


        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

                new NewStudent().setVisible(true);

            }

        });

    }


    // Variables declaration - do not modify
```

```
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JTextField txt_aadhaar;
    private javax.swing.JTextField txt_address;
    private javax.swing.JTextField txt_collage_name;
    private javax.swing.JTextField txt_email_id;
    private javax.swing.JTextField txt_father_name;
    private javax.swing.JTextField txt_mobile_number;
    private javax.swing.JTextField txt_mother_name;
    private javax.swing.JTextField txt_name;
    private javax.swing.JComboBox<String> txt_room_number;
    // End of variables declaration
}
```

UPDATE OR DELETE STUDENT DETAILS –

CODE FOR THE ABOVE PAGE –

```
package hostel;

public class UpdateAndDeleteEmployee extends javax.swing.JFrame {

    public UpdateAndDeleteEmployee() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel8 = new javax.swing.JLabel();
        txt_mobile_number = new javax.swing.JTextField();
```

```
jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();
txt_father_name = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
txt_living = new javax.swing.JComboBox<>();
txt_aadhaar = new javax.swing.JTextField();
txt_collage_name = new javax.swing.JTextField();
txt_address = new javax.swing.JTextField();
txt_mother_name = new javax.swing.JTextField();
txt_email_id = new javax.swing.JTextField();
jButton2 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();
txt_name = new javax.swing.JTextField();
jLabel10 = new javax.swing.JLabel();
txt_room_number = new javax.swing.JTextField();
jButton4 = new javax.swing.JButton();
jButton1 = new javax.swing.JButton();
jLabel11 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setLocation(new java.awt.Point(480, 150));
setUndecorated(true);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel8.setText("Mobile Number");
getContentPane().add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 30, 120, 29));
getContentPane().add(txt_mobile_number, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 30, 290, 29));

jLabel1.setText("Name");
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, 120, 29));

jLabel2.setText("Father Name");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 110, 120, 29));
getContentPane().add(txt_father_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 110, 490, 29));
```

```
jLabel9.setText("Mother Name");
getContentPane().add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 150, 120, 29));

jLabel4.setText("Email Id");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 190, 120, 29));

jLabel3.setText("Address");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 230, 120, 29));

jLabel5.setText("Collage Name");
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 270, 120, 29));

jLabel6.setText("Aadhaar Number ");
getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 310, 120, 29));

jLabel7.setText("Living Status");
getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 390, 120, 29));

txt_living.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Leaved", "Living" }));
getContentPane().add(txt_living, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 390, 490, 30));
getContentPane().add(txt_aadhaar, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 310, 490, 30));
getContentPane().add(txt_collage_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 270, 490, 30));
getContentPane().add(txt_address, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 230, 490, 30));
getContentPane().add(txt_mother_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 150, 490, 29));
getContentPane().add(txt_email_id, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 190, 490, 30));

jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/delete.png"))); // NOI18N
jButton2.setText("Delete");
getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 440, 110, 40));

jButton3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Close all jframe.png"))); // NOI18N
jButton3.setText("Close");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
getContentPane().add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 440, 120, 40));
getContentPane().add(txt_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 70, 490, 29));

jLabel10.setText("Room Number");
getContentPane().add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 350, 120, 29));
```

```
        getContentPane().add(txt_room_number, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 350, 490, 30));

        jButton4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/save.png"))); // NOI18N
        jButton4.setText("Update");
        getContentPane().add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(340, 440, 110, 40));

        jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/search.png"))); // NOI18N
        jButton1.setText("Search");
        getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(460, 30, 128, -1));

        jLabel11.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/pages background.jpg"))); // NOI18N
        getContentPane().add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

        pack();
    }// </editor-fold>

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        txt_mobile_number.setText("");
        txt_name.setText("");
        txt_father_name.setText("");
        txt_mother_name.setText("");
        txt_email_id.setText("");
        txt_address.setText("");
        txt_collage_name.setText("");
        txt_aadhaar.setText("");
    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new UpdateAndDeleteEmployee().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
```

```
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JTextField txt_aadhaar;
    private javax.swing.JTextField txt_address;
    private javax.swing.JTextField txt_collage_name;
    private javax.swing.JTextField txt_email_id;
    private javax.swing.JTextField txt_father_name;
    private javax.swing.JComboBox<String> txt_living;
    private javax.swing.JTextField txt_mobile_number;
    private javax.swing.JTextField txt_mother_name;
    private javax.swing.JTextField txt_name;
    private javax.swing.JTextField txt_room_number;
    // End of variables declaration
}
```

STUDENT FEES –

CODE FOR THE ABOVE PAGE –

```
package hostel;

import javax.swing.JOptionPane;

public class StudentFees extends javax.swing.JFrame {

    public
        Stude
        ntFee
        s() {
```

```
        initCo
        mpon
        ents()
        ;
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed"
    desc="Generated Code"> private void
    initComponents() {
```

```
        jLabel8 = new javax.swing.JLabel();
        txt_mobile_number = new
        javax.swing.JTextField(); jButton1 =
        new javax.swing.JButton();
        jLabel1 = new
        javax.swing.JLabel();
        txt_name = new
        javax.swing.JTextField();
        jLabel4 = new
        javax.swing.JLabel();
        txt_email_id = new
        javax.swing.JTextField();
        jLabel10 = new
        javax.swing.JLabel();
        txt_room_number = new
        javax.swing.JTextField(); jLabel6 =
        new javax.swing.JLabel();
        txt_amount = new
        javax.swing.JTextField();
        jButton3 = new
        javax.swing.JButton();
        txt_month = new
        javax.swing.JTextField();
```

```
jLabel7 = new
javax.swing.JLabel();
jButton2 = new
javax.swing.JButton();
jScrollPane1 = new
javax.swing.JScrollPane(); table =
new javax.swing.JTable();
jButton4 = new
javax.swing.JButton();
jLabel2 = new
javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
);  setLocation(new java.awt.Point(480, 150));
setUndecorated(true);
getContentPane().setLayout(new  org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel8.setText("Mobile  Number");
getContentPane().add(jLabel8,  new org.netbeans.lib.awtextra.AbsoluteConstraints(50,  50,  90,  29));
getContentPane().add(txt_mobile_number,  new org.netbeans.lib.awtextra.AbsoluteConstraints(150,  50,  240,  29));

jButton1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/clear.png"))); //
NOI18N jButton1.setText("Clear");
jButton1.addActionListener(new
   java.awt.event.ActionListener() { public void
   actionPerformed(java.awt.event.ActionEvent evt)
   {
      jButton1ActionPerformed(evt);
 }

}
)
;
getContentPane().add(jButton1,  new org.netbeans.lib.awtextra.AbsoluteConstraints(410,  290,  90,  -1));
```

```
jLabel1.setText("Name");
```

```
getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 90, 90, 29));
getContentPane().add(txt_name, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 90, 490, 29));

jLabel4.setText("Email Id");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 130, 90, 29));
getContentPane().add(txt_email_id, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 130, 490, 30));

jLabel10.setText("Room Number");
getContentPane().add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 170, 90, 29));
getContentPane().add(txt_room_number, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 170, 490, 30));

jLabel6.setText("Amount to be paid");
getContentPane().add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 250, 90, 29));
getContentPane().add(txt_amount, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 250, 490, 30));

jButton3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/Close all jframe.png"))); // NOI18N
jButton3.setText("Close");
jButton3.addActionListener(new
   java.awt.event.ActionListener() { public void
   actionPerformed(java.awt.event.ActionEvent evt)
   {
      jButton3ActionPerformed(evt);
 }

}
)
;
getContentPane().add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(580, 0, 120, 40));
getContentPane().add(txt_month, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 210, 490, 30));

jLabel7.setText("Month");
getContentPane().add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(50, 210, 90, 29));

jButton2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/search.png"))); // NOI18N
jButton2.setText("Search");
getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(420, 50, 128,
-1)); table.setModel(new javax.swing.table.DefaultTableModel(
```

```
                new Object [][] {
                    {null, null, null, null},
                    {null, null, null, null},
                    {null, null, null, null},
                    {null, null, null, null}
```

```
                ,
                new String [] {
                    "Title 1", "Title 2", "Title 3", "Title 4"
 }
)
)
;
            jScrollPane1.setViewportView(table);
            getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 350, 500, 120));

            jButton4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/save.png"))); //
            NOI18N jButton4.setText("Save");
            getContentPane().add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(290, 290, -1, -1));

            jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/hostel/pages background.jpg"))); // NOI18N
            getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 700, 500));

            pack();
        }// </editor-fold>

        private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
            // TODO add your handling code here:
          setAlwaysOnTop(false);
            int ch = JOptionPane.showConfirmDialog(null, "Yes ? You want to close", "Hostel Management",
            JOptionPane.YES_NO_OPTION);  if (ch == 0) {
                MainPage.o
                pen = 0;
                setVisible(f
                alse);
            }
```

```
    setAlwaysOnTop(true);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
      txt_mobile_number.setText("");
    txt_name.set
    Text("");
    txt_month.set
    Text("");
    txt_email_id.set
    Text("");
    txt_amount.set
    Text("");
    txt_amount.set
    Text("");
}
```

```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     *
    /
    t
    r
    y
    {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {   if
            ("Nimbus".equals(info.getName())) {
```

```
                javax.swing.UIManager.setLookAndFeel(info.getClas
                sName()); break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(StudentFees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(StudentFees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(StudentFees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(StudentFees.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new
    Runnable() {
        public void run() {
            new StudentFees().setVisible(true);
 }

}
)
;
    }

    // Variables declaration -
    do not modify private
    javax.swing.JButton
    jButton1;

    private
    javax.swing.JButton
    jButton2; private
    javax.swing.JButton
    jButton3; private
```

```
    javax.swing.JButton
    jButton4; private
    javax.swing.JLabel
    jLabel1; private
    javax.swing.JLabel
    jLabel10; private
    javax.swing.JLabel
    jLabel2; private
    javax.swing.JLabel
    jLabel4; private
    javax.swing.JLabel
    jLabel6; private
    javax.swing.JLabel
    jLabel7; private
    javax.swing.JLabel
    jLabel8;
    private javax.swing.JScrollPane
    jScrollPane1; private
    javax.swing.JTable table;
    private javax.swing.JTextField
    txt_amount; private
    javax.swing.JTextField
    txt_email_id;
    private javax.swing.JTextField
    txt_mobile_number; private
    javax.swing.JTextField txt_month;
    private javax.swing.JTextField txt_name;
    private javax.swing.JTextField txt_room_number;
    // End of variables declaration
}
```

CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in C#.net and no some extent Windows Application and SQL Server, but also about all handling procedure related with "Hostel MANAGEMENT SYSTEM". It also provides knowledge about the latest technology used in developing applications that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

BENEFITS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

⇒ This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.

⇒ The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updating so that the user cannot enter the invalid data, which can create problems at later date.

⇒ Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.

BIBLOGRAPHY AND REFERENCE

Websites

- www.w3schools.com
- www.webdevelopersnotes.com
- www.tutorialpoint.com
- www.codeproject.com
- www.codeguru.com
- www.stackoverflow.com
- www.techGIG.com

Useful Books

- Effective Java (3rd Edition)
- Java: The Complete Reference (11th Edition)
- Thinking in Java (4th Edition)
- Core Java Volume I – Fundamentals (11th Edition)
- Head First Java: A Brain-Friendly(2nd Edition