**INDUSTRIAL TRAINING REPORT**

**On**

**TELEPHONE MANAGEMENT SYSTEM**

***Submitted in partial fulfillment of the requirements***

***for the award of degree of***

**Bachelor of Technology**

**In**

**Computer Science & Engineering**

****

**Submitted by:**

**Chander Mohan Singh Negi**

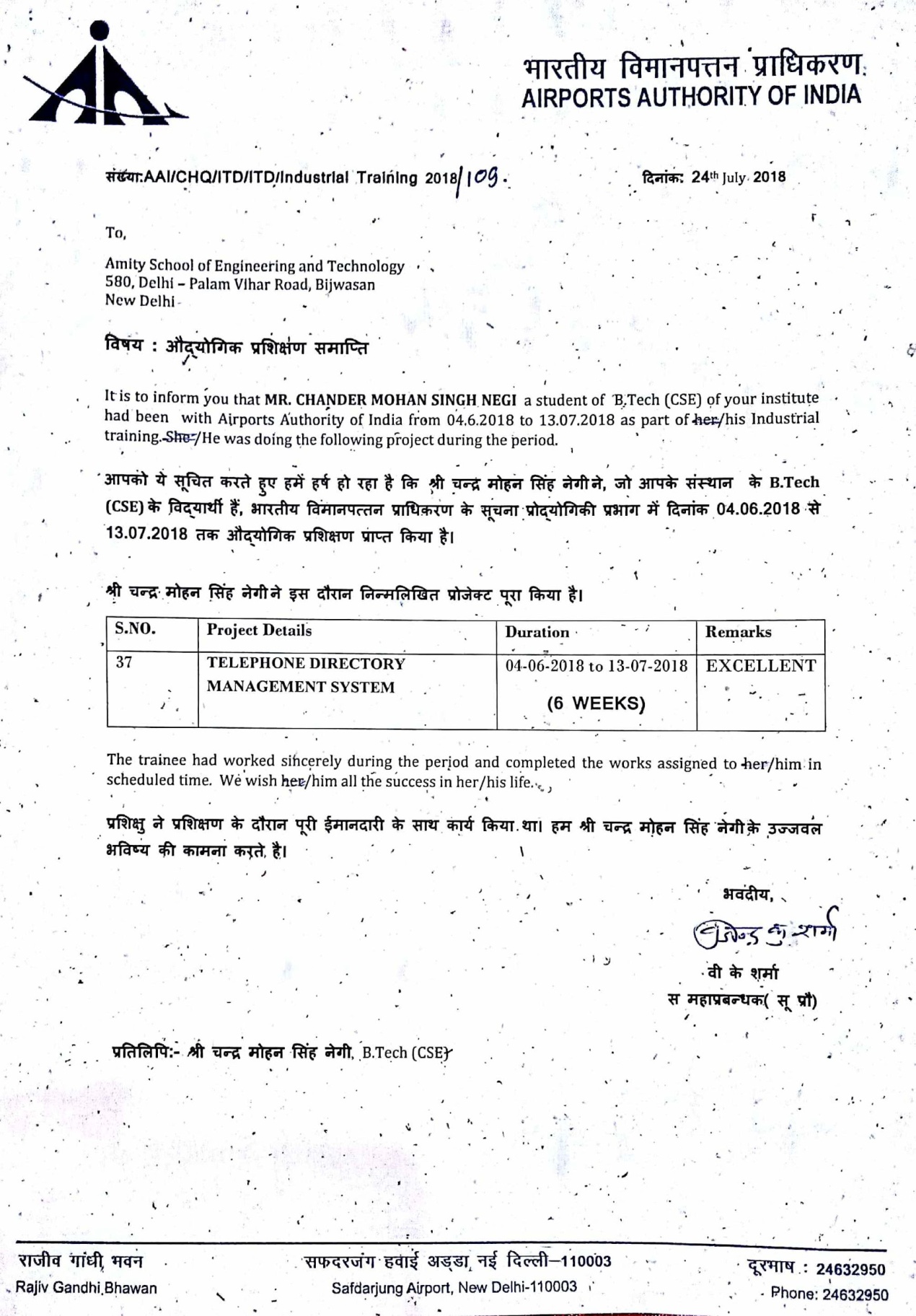
**(00810402715)**

**Department of Computer Science & Engineering**

**Amity School of Engineering and Technology**

**Guru Gobind Singh Indraprastha University, New Delhi**

**CERTIFICATE**

****

**Declaration**

I hereby declare that the Industrial Training Report entitled " **TELEPHONE MANAGEMENT SYSTEM**” is an authentic record of my own work as per the requirements of Industrial Training during the period **June 4th, 2017** to **July 13th, 2017** for the award of degree of Bachelor of Technology degree in Computer Science and Technology, Amity School of Engineering Technology, Bijwasan, New Delhi,Under the guidance of **MR. V. K. SHARMA.**

**Acknowledgement**

For the accomplishment of this project, I would like to take this opportunity to express gratitude to the people who have been a part of this project right from its inception and helped and supported me. It is to them I owe my deepest gratitude.

It gives me immense pleasure in presenting this project report on “**TELEPHONE MANAGEMENT SYSTEM**”. The success of this project is the result of sheer hard work and determination put in by me under the guidance of mentors. I hereby take this opportunity to add a special note of thanks for **Mr. V.K SHARMA**, for their constant encouragement and guidance. Without their insights and support this project wouldn’t have kick-started and neither would have reached fruitfulness.

I express my sincere gratitude to **Prof. Dr. Rekha Agarwal (Director)**, Amity School of Engineering and Technology, and **Prof. Pinky Nayak Head of Department (Department of CS/IT)**, for their constant encouragement and guidance.

Thank You

Chander Mohan Singh Negi

00810402715

**List of Figures**

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

FIG 1: UML Diagram 9

FIG 2: JAVA IDE 16

FIG 3: MySQL Workbench 6.3 CE 21

FIG 4: How data can be added into the directory 29

FIG 5 : Field of the telephone directory 30

FIG 6 : Overview of Database 30

**Table of Contents**

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

Chapter 1 ABSTRACT 1

Chapter 2 INTRODUCTION 2

Chapter 3 REQUIREMENT ANALYSIS 5

Chapter 4 SYSTEM ANALYSIS 7

Chapter 5 DESIGN PHASE 8

Chapter 6 DEVELOPMENT PHASE 10

Chapter 7 TESTING PHASE 28

Chapter 8 RESULTS 29

REFERENCES 31

APPENDIX 32

**Chapter 1**

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

**ABSTRACT**

The main intention of introducing this system is to reduce the manual work at telephone department offices. Every sort of task is performed by the system, such as registering different types of Applications, enquiries and complaints, etc. Reducing mush paper work and burden of file storage.

Also the latest information is right available for the officials and executives wherever they require. This system also facilitates the employ to enquire about their Application progress, bills and directory enquiries such as enquiry by telephone, by name, etc.

**Where the system must be placed?**

There are lot of benefits to the Airports Authority of India by placing the system at their offices. At the same time the employees are also benefited using this system. They can get latest information they require within no time.

**How to use the system?**

Using the system is as simple as using the personal computer. Since end user computing is developing in our country. It is beneficial to both organization and the employees. Every step is clearly defined and help is provided throughout the application to the user. Even the exceptions are handled well to avoid confusion. Third system can be used in a multi user environment.

**How is it beneficial to the employees?**

The employees can get much out of the system. They can get latest information they require within no time. There will be no need for them to stand hours in queues for enquiries or to application or to do any other business within the corporation. They are welcome to use various services.

**Chapter 2**

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

**INTRODUCTION**

**BRIEF OVERVIEW OF TELEPHONE MANAGEMENT SYSTEM**

To develop a Telephone Management System, to take care of employees concerns such as application enquiries, application for connections, application for surrender.

The current manual system is slow laborious and error prone to computerize the same for quicker efficient results and employees satisfaction.

The employees can get much out of the system. They can get latest information they require within no time. There will be no need for them to stand hours in queues for enquiries or to application or to do any other business within the corporation. They are welcome to use various services.

**COMPANY PROFILE**

The Government of India constituted the International Airports Authority of India (IAAT) in 1972 to manage the nation's international airports while the National Airports Authority (NAA) was constituted in 1986 to look after domestic airports.

The organization was merged in April 1995 by an Act of Parliament and was named as Airports Authority of India (AA). This new organization was to be responsible for creating, upgrading, maintaining and managing civil aviation infrastructure both on the ground and air space in the country.

**The Airports Authority of India (AAI)** under the Ministry of Civil Aviation is responsible for creating, upgrading, maintaining and managing civil aviation infrastructure in India. It provides Air Traffic Management (ATM) services over Indian airspace and adjoining oceanic areas. It also manages a total of 125 Airports, including 11 International Airports, 8 custom airports, 81 Domestic Airports and 25 Civil Conclaves at Military Airfields. AAI also has ground installations at all airports and 25 other locations to ensure safety of aircraft operations. AAI covers all major air-routes over Indian landmass via 29 Radar installations at 11 locations along with 89VOR/DVOR installations co-located with The Airports Authority of India (AAI) under the Ministry of Civil Aviation is responsible for creating, upgrading, maintaining and managing civil aviation infrastructure in India. It provides Air traffic management (ATM) services over Indian airspace and adjoining oceanic areas. It also manages a total of 125 Airports, including 11 International Airports, 8 Customs Airports, 81 Domestic Airports and 25 civil enclaves at Military Airfields. AAI also has ground installations at all airports and 25 other locations to ensure safety of aircraft operations. AAI covers all major air-routes over Indian landmass via 29 Radar installations at 11 locations along with 89 VOR/DVOR installations co-located with Distance Measuring Equipment (DME). 52 runways are provided with Instrument landing system (ILS) installations with Night Landing Facilities at most of these airports and Automatic Message Switching System at 15 Airports.

AAI's implementation of Automatic Dependence Surveillance System (ADSS), using indigenous technology, at Kolkata and Chennai Air Traffic Control Centre's, made India the first country to use this technology in the South East Asian region thus enabling Air Traffic Control over oceanic areas using satellite mode of communication. Performance Based Navigation (PBN) procedures have already been implemented at Mumbai, Delhi and Ahmedabad Airports and are likely to be implemented at other Airports in a phased manner. AAI is implementing the GAGAN project in technological collaboration with the Indian Space Research Organization (ISRO), where the satellite based system will be used for navigation. The navigation signals thus received from the GPS will be augmented to achieve the navigational requirement of aircraft. First phase of technology demonstration system was completed in February 2008.

AAI has four training establishments viz. The Civil Aviation Training College (CATC) at Allahabad, ational Institute of Aviation Management and Research (NIAMAR) at Delhi and Fire Training Centre's (FTC) at Delhi & Kolkata An Aerodrome Visual Simulator (AVS) has been provided at CATC and non-radar procedural ATC simulator equipment is being supplied to CATC Allahabad and Hyderabad Airport. AAI has a dedicated Flight Inspection Unit (FIU) with a fleet of three aircraft nitted with flight inspection system to inspect Instrument Landing Systems up to Cat-III, VORS, DMES, NDBs, VGSI (PAPI, VASI) and RADAR (ASR/MSSR). In addition to in-house flight calibration of its navigational aids, AAI undertakes flight calibration of navigational aids for the Indian Air Force, Indian Navy, Indian Coast Guard and other private airfields in the country.

**AAT has entered into Joint Ventures at Mumbai, Delhi, Hyderabad, Bangalore and Nagpur Airports to upgrade these airports. (DME). 52 runways are provided with Instrument landing system (ILS) installations with Night Landing Facilities at most of these airports and Automatic Message Switching System at 15 Airports.**

AAI's implementation of Automatic Dependence Surveillance System (ADSS), using indigenous technology, at Kolkata and Chennai Air Traffic Control Centre's, made India the first country to use this technology in the South East Asian region thus enabling Air Traffic Control over oceanic areas using satellite mode of communication. Performance Based Navigation (PBN) procedures have already been implemented at Mumbai, Delhi and Ahmedabad Airports and are likely to be implemented at other Airports in a phased manner. AAI is implementing the GAGAN project in technological collaboration with the Indian Space Research Organization (ISRO), where the satellite based system will be used for navigation. The navigation signals thus received from the GPS will be augmented to achieve the navigational requirement of aircraft. First phase of technology demonstration system was completed in February 2008. AAI has four training establishments viz. The Civil Aviation Training College (CATC) at Allahabad, National Institute of Aviation Management and Research (NIAMAR) at Delhi and Fire Training Centre's (FTC) at Delhi & Kolkata. An Aerodrome Visual Simulator (AVS) has been provided at CATC and non-radar procedural ATC simulator equipment is being supplied to CATC Allahabad and Hyderabad Airport. AAI has a dedicated Flight Inspection Unit (FIU) with a fleet of three aircraft fitted with flight inspection system to inspect Instrument Landing Systems up to Cat-III, VORS, DMES, NDBs, VGSI (PAPI, VASI) and RADAR.

**GOALS AND NEED**

**GOAL**:

With every going day the need to be connected is increasing to an extent where inflow of connection request exceeds that which can be handled manually. Hence computerization of application request and employees complaints brings better satisfaction and service oriented -ness.

Quicker processing of application would mean better business and more employees. It would also help in having fewer complaints and thus less time is wasted on rework. Proper billing timely payments and good employees' feedback go a long way in identifying and succeeding a business and its future growth.

Towards this achievement the computerization of the Telephone industry will help greatly in meeting market requirements more satisfied employees would mean quicker growth and hence obviously greater profits towards which all organizations strive.

**NEED:**

To develop a Telephone inventory maintenance system, to take care of Employees concerns such as application enquiries and complaints. The current manual system is slow laborious and error prone to computerize the same for quicker efficient results and employees satisfaction.

**GENERAL METHODOLOGY IN DEVELOPING**

**SOFTWARE PROJECT**

The general methodology in developing a system is involved in different phases, whichdescribe the system's life cycle model for developing software project. The conceptincludes not only forward motion but also have the possibility to return that is cycle back to an activity previously completed.

This cycle back or feedback may occur as a result ofthe failure with the system to meet a performance objective or as a result of changes in redefinition of system activities. Like most systems, the life cycle of the computer based system also exhibits distinct phases.

**Chapter 3**

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

**REQUIREMENT ANALYSIS**

This phase includes the identification of the problem, in order to identify the problem, we have to know information about the problem, the purpose of the evaluation for problem to be known. We have to clearly know about the client's requirements and the objectives of the project.

The requirements also include the in-hand name and telephone details or precisely the data which needs to be added the inventory. All the data has to be collected and stored as a hard copy.

Software analyzation, installation, proper handling and professionalism is also required along with testing tools.

What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

Problem/Requirement Analysis:The process is order and more nebulous of thetwo, deals with understand the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has **been** found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The Requirement phase terminates with the production of the validate SRS document, Producing the SRS document is the basic goal of this phase.

ROLE OF SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

Requirement Check:

Given a list of contacts which exist in a phone directory. The task is to implement search query for the phone directory.Phone Directory can be efficiently implemented using [Trie](https://www.geeksforgeeks.org/trie-insert-and-search/" \t "_blank) Data Structure. We insert all the contacts into Trie.

Generally search query on a Trie is to determine whether the string is present or not in the trie, but in this case we are asked to find all the strings with each prefix of ‘str’. This is equivalent to doing a [DFS traversal on a graph](https://www.geeksforgeeks.org/depth-first-traversal-for-a-graph/). From a Trie node, visit adjacent Trie nodes and do this recursively until there are no more adjacent. This recursive function will take 2 arguments one as Trie Node which points to the current Trie Node being visited and other as the string which stores the string found so far with prefix as ‘str’.  
Each Trie Node stores a boolean variable ‘isLast’ which is true if the node represents end of a contact(word).

SOFTWARE REQUIREMENTS

Platform - Window (2000/XP) UNIX Solaris

Software - JAVA, MS SQL 2008 R2,

Visual Studio 2012

HARDWARE REQUIREMENTS

Processor:-Intel Celeron Class Processor with 2.0 GHz

RAM: 256

Hard Disk: 40 GB

Keyboard: 101 keys

Mouse - Any pointing device

**Chapter 4**

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

**SYSTEM ANALYSIS PHASE**

Feasibility analysis involves the benefits of various application approaches and the determination of the alternative application approaches a through method like questionnaires and interviews etc., different data about the project is collected and the data throughout the project is represented in the form of UML Diagrams.

In a typical Telephone service provider scenario employees gave their new connection requests to the local organizational manager. The Admin generally gives them forms to fill up which are subsequently scrutinized and verified with the DOT provided features/ services as application applicable and further verifications about the employees are made. The local Manager office also connects to the branch exchange to verify the services available and to identify if the exchange would need infrastructural up gradations Traffic analysis and availability of bandwidth and other technical validations are made. Further the branch exchange goes though sequence verification and document processing operations, which are replicated at a city and subsequently at a national exchange level. The entire process is very time consuming and involves tons of paper work- mostly manually, which is both error prone and time consuming.

The new system would have employees raising applications to the local manager which in further gets in touch with the branch office and the city exchange and all the employees' details are finally updated and stored at a nation exchange level database, apart from being replicated at each of the lower line databases. The following diagram exhibits the typical Telephone management service scenario.

The purpose of this document is to describe all external requirements for the Requirement of CNS department.

It also describes the interfaces for the system.

This document is the only one that describes the requirements of the system. It is meant for the use by the developer's, and will also be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change application approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

Analysis is the detailed study of various operation performed by system and their relationship with the outside of the system. A key question is: What must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determination whether or not candidate system could consider other related system. During analysis, data are collected on the available files, decision points, and transactions handled   by the present system

**Chapter 5**

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

**DESIGN PHASE**

SOFTWARE design is a process through which the requirements are translated into a representation of software. One of the software requirements have been analyzed and specified, the software design involves three technical activities, design, coding generation and testing.

The design of the system is in modular form i.e., the software is logically partitioned into components that perform specific functions and sub functions. The design phase leads to modules that exhibit independent functional characteristics. It even leads to interfaces that reduce the complexity of the connections between modules and with the external environment. The design phase is of main importance because in this activity, decisions ultimately affect the success of software implementation and maintenance.

**DEVELOPMENT PHASE**

The development phase includes choosing of suitable software to solve the particular problem given. The various facilities and the sophistication in the selected software give a better development of the problem.

User will enter the string character by character and we need to display suggestions with the prefix formed after every entered character.  
So one approach to find the prefix starting with the string formed is to check if the prefix exists in the Trie, if yes then call the displayContacts() function. In this approach after every entered character we check if the string exists in the Trie.  
Instead of checking again and again, we can maintain a pointer **prevNode**‘ that points to the TrieNode which corresponds to the last entered character by the user, now we need to check the child node for the ‘prevNode’ when user enters another character to check if it exists in the Trie. If the new prefix is not in the Trie, then all the string which are formed by entering characters after ‘prefix’ can’t be found in Trie too. So we break the loop that is being used to generate prefixes one by one and print “No Result Found” for all remaining characters.

**CODING PHASE**

The coding phase is for translating the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and which performs the computation specified by the design.

**TESTING PHASE**

Testing is done in various ways such as testing the algorithm, programming code, sample data debugging is also one of following the above testing.

**DESIGN CONSTRAINTS**

The telephone Management System require hug resources as hundreds of thousands of the employee’s will require the service instantly, quick response time are needed.

The database should also be very large and robust to maintain very huge employee’s data.

**UML DIAGRAM**

FIG 1: UML Diagram

**Chapter 6**

**DEVELOPMENT AND IMPLEMENTATION**

.

**MODULE DESCRIPTION**

This section attempts to describe each module of the project in brief, and the detailed description of each of these modules is spread throughout this document.

The Telephone Management System has been divided into four modules:

They are:

1. Application
2. Entries
3. Enquiries
4. **APPLICATION**

This module has been divided into five sub modules. They are:

* 1. Application for new phone Connection
  2. Application for phone Transfer
  3. Application for Modifications.
  4. Application for Cancellation

1.1 **APPLICATION FOR NEW CONNECTION**

If employees would like to take new telephone new connections, he or she has to fill in an Application from called “Application for new Phone connection” which includes Names, Address, purpose (Residence/Business/ Office,), facility (Local/STD/ISD), Employee code from any nationalized  department.

1.**2 APPLICATION FOR PHONE TRANSFER**

Some situation may occur when an employee shifts his residence from one place to another place. So, the employee address or department will be changed. At that time, they should inform the official concerned so that the further communication will be easy otherwise the phone connection will be disconnected.

If an employee wants any modifications except phone no, ref no, any address, he can get changed details for example purpose of phone from residence to business or facility from Local to STD etc.

1.3 **APPLICATION FOR CANCELLATION**

If a employees wants to cancel his telephone connection, he has to fill in an application form by specifying all details. Operator should check weather all dues are paid or not otherwise the phone connection will not be cancelled. If all dues are clear, the phone connection will be cancelled. For further purpose I am storing this employee’s record into a history file. It contains the list of all cancelled connections.

1. **ENTRIES**

This module is divided into three modules. They are:

2.1  Bill Entry

2.2 Paid Bill Entry

2.1   Bill Entry

This is an entry done by entry operator after taping phone calls from a device. The bill ID should be generated automatically by the system. He has to check how many phone calls employee’s makes and what the amount is. Amount will be calculated automatically by the system by giving calls made by specifying Local, STD etc.

2.2   Paid Bill Entry

This is an entry done by same entry operator after the bill paid by the employees to see whether it is paid or not.

1. **ENQUIRY**

This module has been divided into five sub modules. They are:

3.1 Telephone Number

This is an enquiry made by any person to know the details of employees by the telling phone Number.

* 1. Application Enquiry

Application Enquiry for New connection

Application Enquiry for phone transfer

Application Enquiry for Temporary connection

**FEASIBILITY ANALYSIS**

Feasibility study is an important phase in the software development process. Its enables the developer to have and assessment of the product being developed. It refers to the feasibility study of the product, operational use and technical support required for implementing it. It should be performed on the basis of:

* Economic Feasibility
* Operational feasibility
* Technical feasibility

**ECONOMIC FEASIBILITY:**

It refers to the benefits or outcomes I am deriving from the product as compared to the total cost I am spending for the developing the product.

**OPERATIONAL FEASIBILITY:**

It refers to the feasibility of the product to be operational. Some products may work very well at design and implementation but may fall in the real environment.

**TECHNICAL FEASIBILITY:**

It refers to whether the software that is available in the market fully supports the present Application.

**IMPLEMENTATION PLAN**

The main plan for the system developed is to mimic the existing system as it is in the proposed system.

**STUDY OF THE EXISTING SYSTEM**

The present system has obvious problems, inhibiting growth and profitability. Demand of telephone connections demand an improved, computer system to support them. By using the present system, work is done manually. So, each and every transaction takes much time to complete. Whenever employees requires any information.

Disadvantages:

1. The work is done manually so that it takes much time to transact a transaction.
2. More number of labours is needed.
3. We can’t asses the calculations accurately.

**THE PROPOSED SYSTEM**

The present system has obvious problems, inhibiting growth and profitability. Demand of telephone connections have been identified as the major growth area. By using the proposed system whole system is computerized. So, each and every transaction takes less time to complete.

Advantages:

1. Saving in staff time in entering and manipulating data.

2. Easy input, deletion and manipulation of lot, employee’s details.

3. Capable of printing both full and selected information.

Goals and Objective

Service should be provided to employees in an efficient manner. Phone number is issued instantly when employees application for connection. Enquiry details must be provided for the employees, such as Application, bills, complaints enquiries etc. Bills will be issued on time to employees and status of the bills will be observed schedule wise. Each and every employee record should be maintained in systematic manner so that the searching process will be easy.

**IMPLEMENTATION**

**INTRODUCTION**

Design is the first step in the development for any techniques and principles for the purpose of defining device, a process or system in sufficient detail to permit its physical realization.Once the software requirements have been analysed and specified the software design involves three technical activities design, coding, generation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the employee’s requirement into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

FLOW CHARTS

Before solving a problem with the help of a computer, it is essential to plan the solution in a step-by-step manner. Such a planning is represented symbolically with the help of flow chart. It is an important tool of system analysts and Programmers for tracing the information flow and the logical sequence in data processing Logic is the essence of a flow chart.A flow chart is the symbolic representation of step-by-step solution of a given problem, and it indicates flow of entire process, the sequence of the data input, operations, computations, decisions, results and other relevant information.Pertaining to a particular problem, a flow chart helps us in the complete understanding of the logical structure of a complicated problem and in documenting the method used. It would be seen that the flow chart is a very convenient method of organizing the logical steps and deciding what, when and how to proceed with various processes. The logic should be depicted in the flow charts.

KINDS OF THE FLOW CHARTS

1. SYSTEM FLOW CHARTS

The system analyst to describe data flow and operations for the data processing cycle uses these. A system flow chart defines the broad processing to be performed, output that is to generate and necessity of the offline operation.

1. PROGRAM FLOW CHART/COMPUTER PROCEDURE FLOWCHART

The programmers to describe the sequence of operation and decision of a particular problem normally use these. In this we use structure of program.

**ADVANTAGES:**

Apart from, the DFDS the flow charts has been helping the programmer to develop the programming logic and to serve as the documentation for a completed program, it has the advantages:

1. They help for the easy understanding of the logic of process or a procedure
2. It is a better communicating tool than writing in words.
3. It is easy to find the conditions, which are responsible for the actions.
4. It facilitates troubleshooting.
5. It promotes logical accuracy.

DISADVANTAGES:

1. Communication lines are not always easy to show.
2. The charts are sometimes complicated.
3. Reproduction is difficult.
4. They are hard to modify.

DEVELOPMENT PHASE

INTRODUCTION

The goal of any system development is to develop and implement the system cost effectively; user friendly and most suited to the user’s analysis is the heart of the process. Analysis is the study of the various operations performed by the system and their relationship within and outside of the system. During analysis, data collected on the files, decision points and transactions handled by the present system. Different kinds of tools are used in analysis of which interview is a common one.

INITIAL INVESTIGATION

The first step in system development life cycle is the identification of need of change to improve or enhance an existing system. An initial investigation on existing system was carried out. The present system of hospital is completely manual. Many problems were identified during the initial study of the existing system.

JAVA

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be Write Once, Run Anywhere.

Java is −

* Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
* Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
* Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
* Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
* Architecture-neutral − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
* Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
* Robust − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
* Multithreaded − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
* Interpreted − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
* High Performance − With the use of Just-In-Time compilers, Java enables high performance.
* Distributed − Java is designed for the distributed environment of the internet.
* Dynamic − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

**History of Java**

James Gosling initiated Java language project in June 1991 for use in one of his many set-top box projects. The language, initially called ‘Oak’ after an oak tree that stood outside Gosling's office, also went by the name ‘Green’ and ended up later being renamed as Java, from a list of random words.Sun released the first public implementation as Java 1.0 in 1995. It promised Write Once, Run anywhere (WORA), providing no-cost run-times on popular platforms. On 13 November, 2006, Sun released much of Java as free and open source software under the terms of the GNU General Public License (GPL). On 8 May, 2007, Sun finished the process, making all of Java's core code free and open-source, aside from a small portion of code to which Sun did not hold the copyright. Tools You Will Need For performing the examples discussed in this tutorial, you will need a Pentium 200-MHz computer with a minimum of 64 MB of RAM (128 MB of RAM recommended). You will also need the following softwares −

Linux 7.1 or Windows xp/7/8 operating system

Java JDK 8

Microsoft Notepad or any other text editor

This tutorial will provide the necessary skills to create GUI, networking, and web applications using Java.

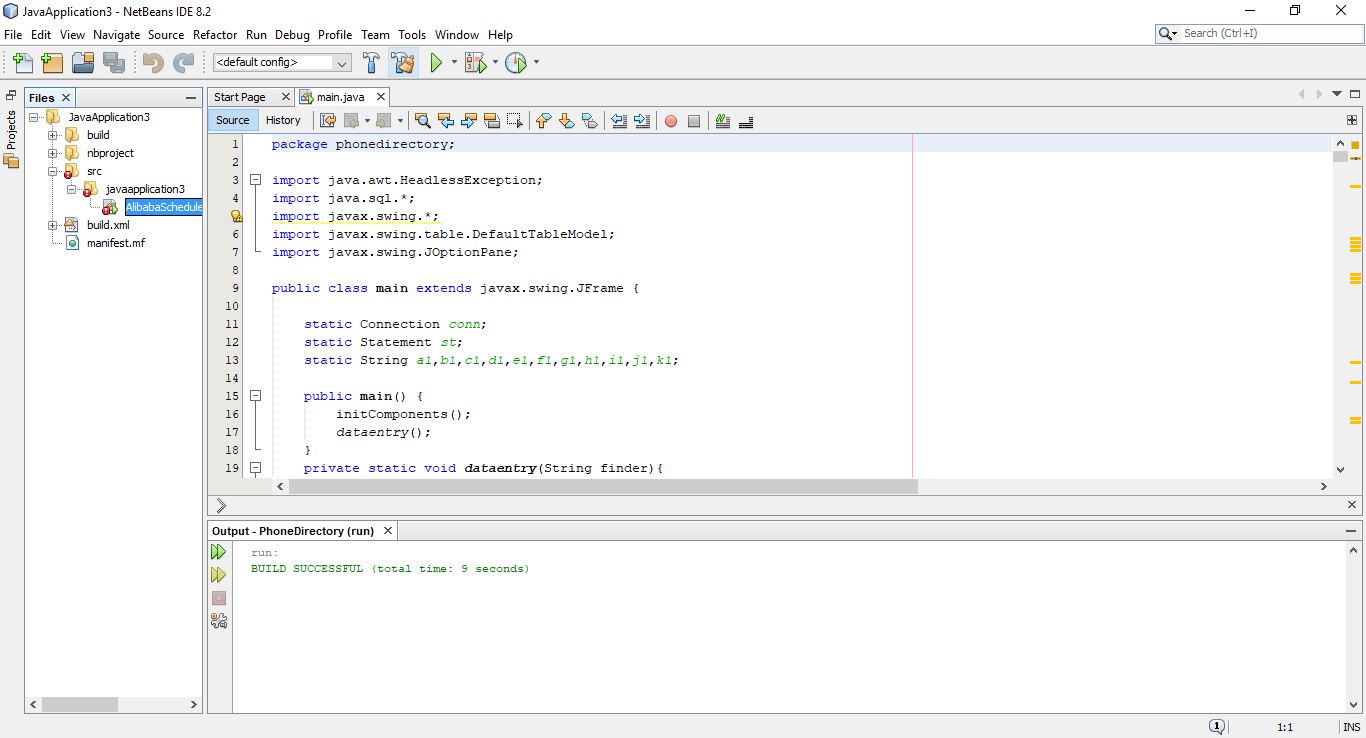


FIG 2: JAVA IDE

**Overview of MySQL**

MySQL has a large portion of the online market. Where Microsoft SQL Server and Oracle were once dominant, MySQL's free architecture and improved interface through the years have made it one of the top three databases used across the world. It's used by some of the largest websites including Facebook and Pinterest. Small startups that use WordPress inherently use MySQL since the application is integrated with the database. Webmasters, ecommerce startups, bloggers, and programmers can all benefit by understanding and learning MySQL administration.

What is MySQL?

There are two regions in a website design: the front-end and the back-end. The front-end is the site design and interface users see when they open the site. The back-end is the database. The database is the power horse for an application. It houses all the data and information needed by the front-end. Webmasters have several options when choosing a database, and one of these options is MySQL.

Before you understand MySQL, it helps to understand database functionality. A database is made up of tables, stored procedures, and functions. These three parts drive the back-end of your application.

Tables are the database engine's storage components. Architecture of database storage is extremely technical, but an easy way to envision a database table is to visualize a spreadsheet. Spreadsheets have columns and rows. Where these columns and rows intercept are fields. The fields contain one item of data.

When you set up a spreadsheet, your columns make up the components of your data. For instance, a spreadsheet that contains customer information has a first name, last name and address column. Each row has a column of information. These rows represent your records. Using the customer example, each row is a customer.

This is a simplified way to explain a MySQL table, but it helps new database users envision how data is stored and represented. If you look at a MySQL table, it's laid out like a spreadsheet of information. The trick is to understand how to retrieve, filter, edit and delete data.

Stored procedures are the programming objects within your database. For instance, suppose you want to select the last customer who ordered, and then send that customer an email. You could write a stored procedure. You can view stored procedures as small functions for your application. They perform certain functionality, which usually results in displayed, editing, inserting or updating data.

Functions are similar to regular programming functions. They act as standard procedures that you use regularly in your SQL programming. For instance, suppose you need to add two numbers each time you query your MySQL tables. Instead of writing stored procedure code each time you add the numbers, you can create a function and use it to save you programming time.

**Why is MySQL So Popular?**

MySQL isn't popular just because it's free. Most free software isn't associated with high-end enterprise solutions, but MySQL is an exception to this rule. MySQL can be used with small startups or high-traffic enterprise applications. Banks, social networks, universities and government agencies use MySQL as their database solution.

Because companies can go from small startup to large enterprise without changing its database solution, MySQL is scalable. Scalability is often disregarded when startups have tight budgets, but since MySQL is free, it's a good option for a startup that expects growth. It's also high performance. Performance is paramount to application stability. Small databases with only a few records will normally function well even if they are poorly designed. However, if the database tables grow to millions of records, application performance and stability can be affected, which then affects your customers and employees.

MySQL is also secure. Security is a sensitive issue, since the protection of your customer and employee records should be a major concern. With MySQL, administrators can encrypt data and set up authentication to protect all company assets.

There are several more open-source database applications on the market that were deployed after MySQL. MySQL's success is partly due to the numerous hosting companies that offer services with unlimited MySQL database services. Oracle and SQL Server are expensive platforms, and most hosting companies only offer 1 SQL Server if you choose to work with it instead of an open-source database solution. Since MySQL is free and open-source, hosting companies offer unlimited databases with a lower cost than Microsoft platforms. This type of marketing gave MySQL its strong foundation in application development in global cloud and desktop applications.

Relational Data Integrity

MySQL is a relational database. These systems offer data integrity. They are distinct from systems such as Mongo that relies on document style records. Relational databases are used in numerous applications, but the major reason they are used to create applications is for data integrity and transactional style record manipulation.

Relational databases work on the concept of a primary and foreign key relationship. The primary key is the main unique field that identifies a record. This unique identifier is then stored in other tables to build a relationship between a main table and a secondary related table. Each customer has its own unique identifier, and then this identifier is stored in the order table. When you need to query MySQL with a list of customers and related orders, you join the two tables using specific SQL syntax.

MySQL is also a transactional database, which means that you can roll back changes to your database. For instance, suppose you want to delete a customer but you don't want to delete the customer if there are active order records associated with the customer. You attempt to delete the customer, but MySQL's relational foreign key constraints give you an error when you attempt to delete it. You can then roll back any changes you made based on commit and rollback features. You commit changes if there were no errors found, or you can roll back previous statements if an error is thrown by the database server.

Data integrity is what sets MySQL and other relational databases apart from more modern databases such as NoSQL. NoSQL databases such as Mongo are used for analytical data and capturing any number of unstructured data. MySQL requires your data to be more structured, so it's a reliable database system for people who want to secure the structure and relationship between tables.

In bigger businesses, MySQL and NoSQL databases work together. The MySQL database stores structured data such as orders and customer information, and the NoSQL database stores unstructured data such as marketing and traffic numbers. You can export data from a MySQL database to a NoSQL database to work with them both for their best features.

We review relationships such as primary and foreign keys, transactional changes, committing and rolling back changes, and how structured data is stored in the MySQL database. It's important to understand relationships and relational data to effectively build and maintain a MySQL database architecture. If it's engineered incorrectly, you can have difficulties with performance and maintaining your data. We discuss configuring and maintaining performance on your database. We also discuss replication and how you can configure replicated database servers to create a mirrored backup of your existing MySQL database. We also discuss how you can manage replication for the best performance.

Installing MySQL

Installation Choices and Platforms

You might wonder how MySQL can generate revenue to support its continued service. While the database engine itself is free, MySQL makes its money through support options. This is the common open-source option for developers.

The basic installation options are the database engine and the Workbench. The engine is what stores and queries your data sets. The Workbench is the main user interface for designing and maintaining your databases. You can also use the command line features in MySQL, but new people find the Workbench easier to user when they are learning MySQL functionality.

If you choose a hosting provider, they normally offer MySQL databases as an addition to hosting services. Support services are by yearly subscription. Standard edition is offered for $2,000, Enterprise edition is $5,000 and Cluster Carrier Grade edition is $10,000. Most basic enterprise application solutions use Standard edition with internal support.

For people learning MySQL, the database platform has a strong user community. They also offer a developer community where new database designers and programmers can collaborate.

There are two main installation environments with MySQL: WAMP and LAMP. Linux, Apache, MySQL and PHP (LAMP) is the most common type of installation. This type of installation uses any distribution of Linux and Apache is the web server software. MySQL and PHP work seamlessly with Linux and Apache, so people familiar with the Linux environment will not have many problems installing and configuring MySQL.

Windows, Apache, MySQL, and PHP (WAMP) is the other option for developers and administrators familiar with Windows. When MySQL was first deployed, it was more compatible with the Linux and Unix environments. MySQL developers have expanded the platform and now offer Windows developers and administrators the option to install the open-source database instead of struggling to move it to a Linux environment. MySQL has an MSI installer package that you can use to install the database system on your Windows server.

You don't need Apache either, if you want to use just a PHP and MySQL environment. MySQL is just the database system, so you can choose any number of web services to host the application environment. You can also use PHP with Windows Internet Information Services (IIS), so you aren't limited to just Apache. The environment you choose determines the structure of your application and infrastructure, so choose your platform carefully with your technology consultant.

Finding the MySQL Download

The MySQL website is found at MySQL.com. The download link is a tab on their home page. You have two options: Enterprise or Cluster CGE. Enterprise edition is typically what most site owners can use. Cluster edition is used when you have a large farm of database engines that you need to work together to support a large application.

When you download the installation files for MySQL, ensure that you only download them from an official source. In most cases, this is only on the main MySQL website itself. In rare occasions, your host or provider might give you a link to the site or a fully installed server without installing the database yourself. However, you should still install MySQL to your development environment. When you build SQL queries, you should always first build them in development and then move these queries to a production environment.

Another advantage of MySQL is that the database engine runs on all three main operating systems: Linux, Windows, and Mac. This article will cover installing MySQL after you've successfully downloaded the binary files.

Installing MySQL for Windows

Normally, you install MySQL on a server, but the database also runs on a regular desktop operating system. You must install MySQL as an administrator. For Windows, right-click the install file and select "Run as Administrator." The install file is an MSI, so it's a native Windows Installer package. This option elevates privileges on newer Windows OS versions such as 7, 8 or 2012.

If you have a web server hosted in Windows Server 2012, you have an optional way to install MySQL. You can first download Microsoft Web PI (Platform Installer). This installer can be found at Microsoft's website by searching "Microsoft Web PI."

Once you download the executable, right-click on "wpilauncher.exe" and select "Run as Administrator." Choose MySQL Windows 5.1 and click "Install."

The database engine asks you to enter a root password. The "root" user is just like the administrator in Windows. It has full control of the database, and you'll use it to add new users, design databases, back up your data, and perform administration queries against the server.

Installing MySQL in Linux

Linux is a command line based system, at least more than Windows. It's recommended that MySQL is installed using RPM Package Manager. You'll need to log in as the root user on your Linux server or desktop.

Some Linux distributions come with MySQL packages. They are separated into several components including the ending, client, developer libraries and benchmark and performance tools. You just need the database engine to run MySQL, so all the others are extras you can use to learn how to work with MySQL.

After you log in to root, change the current directory path to the one containing the RPM packages. Once you're there, use the following command

[root@host]# rpm -i MySQL-5.0.9-0.i386.rpm

Replace the version above with the MySQL version you downloaded.

Install MySQL on Mac

Installing MySQL on a Mac is similar to Windows except the file extensions are different. You can download the packaged executables for Mac and use the Installer to install the database. Just like Linux and Windows, client tools should be installed on the machine you want to use to create queries and manage the database. Using these tools will make it much easier for you to navigate and understand MySQL database maintenance and programming rather than focus on specific command line options. Client tools also give you a GUI to work with.

Understanding MySQL Directory Contents

After you install the database engine, the next step is to understand the engine and files associated with the engine.

You should know where the data is stored. You can use this information to automatically back up your database. In Windows, you can change the location of your data using the my.ini file in the MySQL installation directory. Unless you gave the installation program a custom location, MySQL is installed into the "Program Files\mySQL\MySQL Server x.x" directory.

In the my.ini file, look at the "datadir" value. This is where your data is stored. By default, it's stored in the "All Users\Application Data\MySQL\MySQL Server x.x\Data" directory.

Linux uses the same variable, but the file name that stores variables and preferences is my.cnf. The default data location is the "/var/lib/mysql" folder.

There are three main components in MySQL. The first is development. This section is where you connect to a database, write SQL scripts, edit data and manage existing connections. This section is also where you design tables and write your stored procedures.

 The next section is server administration. Administration and development work together, but they have defined differences. The administration side of MySQL is more for maintaining your database. You can import and export data, manage server instances, add security, and add new MySQL instances.

The data modeling tool lets you create objects based on visual graphics. Instead of drawing a model in third-party software such as Visio, you can create data models that are then turned into your database design. Data modeling tools are one reason MySQL is popular over other competitor products.

MySQL and Applications

MySQL powers huge systems across the world, but it's also a scalable and dynamic way to build a small system. MySQL uses the SQL language to retrieve, edit and remove information. You can use any number of application languages to interface with the database. The most commonly used is PHP.

PHP has internal functions that work directly with MySQL. Coders don't need to create complex PHP layers to work with the data. Instead, coders can call functions that handle the technological communication between the application and the database.

We'll use PHP to show you how to connect your applications to the database. You still need to create SQL statements in PHP. You also need to test and secure these applications, but knowing basic SQL will help you build applications that retrieve, edit and remove records from the database.

This brief overview of MySQL is just the beginning. MySQL is a powerful database that can support any of your applications. Once you have the engine installed, you can start programming stored procedures, designing tables, querying records, and storing your data.

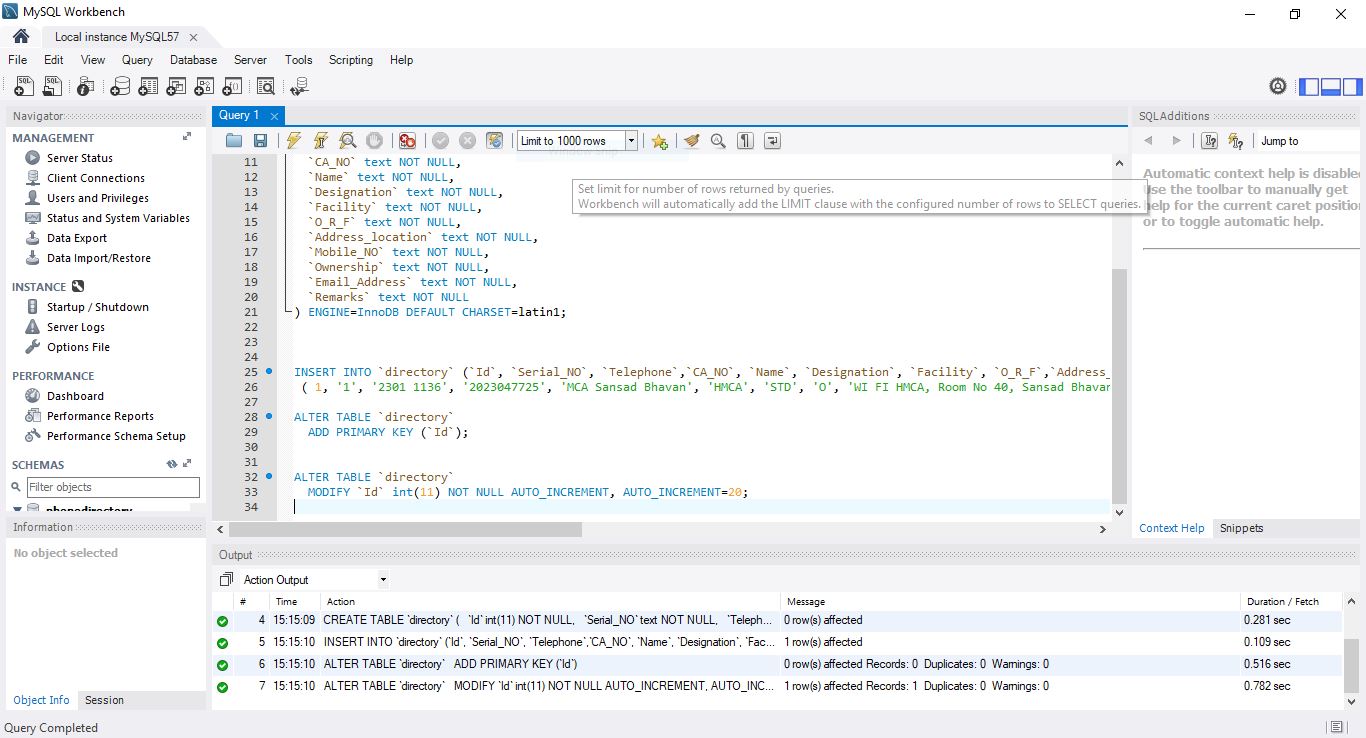


FIG 3: MySQL Workbench 6.3 CE

**EXCEL**

Microsoft Office contains a variety of tools that help people accomplish many personal and professional objectives. Microsoft Excel is perhaps the most versatile and widely used of all the Office applications. No matter which career path you choose, you will likely need to use Excel to accomplish your professional objectives, some of which may occur daily. This chapter provides an overview of the Excel application along with an orientation for accessing the commands and features of an Excel workbook.

**MAKING DECISIONS WITH EXCEL**

Taking a very simple view, Excel is a tool that allows you to enter quantitative data into an electronic spreadsheet to apply one or many mathematical computations. These computations ultimately convert that quantitative data into information. The information produced in Excel can be used to make decisions in both professional and personal contexts. For example, employees can use Excel to determine how much inventory to buy for a clothing retailer, how much medication to administer to a patient, or how much money to spend to stay within a budget. With respect to personal decisions, you can use Excel to determine how much money you can spend on a house, how much you can spend on car lease payments, or how much you need to save to reach your retirement goals. We will demonstrate how you can use Excel to make these decisions and many more throughout this text.

Figure 1.1 shows a completed Excel worksheet that will be constructed in this chapter. The information shown in this worksheet is top-line sales data for a hypothetical merchandise retail company. The worksheet data can help this retailer determine the number of salespeople needed for each month, how much inventory is needed to satisfy sales, and what types of products should be purchased.

STARTING EXCEL

Locate Excel on your computer.

Click Microsoft Excel to launch the Excel application and present you with workbook options.

Click the first option; “Blank Workbook”.

**THE EXCEL WORKBOOK**

Once Excel is started, a blank workbook will open on your screen. A workbook is an Excel file that contains one or more worksheets (sometimes referred to as spreadsheets). Excel will assign a file name to the workbook, such as Book1, Book2, Book3, and so on, depending on how many new workbooks are opened. Figure 1.2 shows a blank workbook after starting Excel. Take some time to familiarize yourself with this screen. Your screen may be slightly different based on the version you’re using.

Your workbook should already be maximized (or shown at full size) once Excel is started, as shown in Figure 1.2. However, if your screen looks like Figure 1.3 after starting Excel, you should click the Maximize button, as shown in the figure.

**NAVIGATING WORKSHEETS**

Data are entered and managed in an Excel worksheet. The worksheet contains several rectangles called cells for entering numeric and nonnumeric data. Each cell in an Excel worksheet contains an address, which is defined by a column letter followed by a row number. For example, the cell that is currently activated in Figure 1.3 is A1. This would be referred to as cell location A1 or cell reference A1. The following steps explain how you can navigate in an Excel worksheet:

Place your mouse pointer over cell D5 and left click.

Note:  Your highlighted column letter and row number may be different than figure shown.

Move the mouse pointer to cell A1.

Click and hold the left mouse button and drag the mouse pointer back to cell D5.

Release the left mouse button. You should see several cells highlighted, as shown in Figure 1.5.

This is referred to as a cell range and is documented as follows: A1:D5. Any two cell locations separated by a colon are known as a cell range. The first cell is the top left corner of the range, and the second cell is the lower right corner of the range.

At the bottom of the screen, you’ll see worksheets. Depending on your version of Excel, you will see either three as displayed above or just one. If you only have one sheet, click the “Insert Worksheet” to add a worksheet. Depending on your version, you instead may have a + sign; a click on the + adds an additional worksheet as well. This is how you open or add a worksheet within a workbook. Add another worksheet so that you now have three sheets displaying here.

Click the Sheet1 worksheet tab at the bottom of the worksheet to return to the worksheet shown in Figure 1.5.

Keyboard Shortcuts

Basic Worksheet Navigation

Use the arrow keys on your keyboard to activate cells on the worksheet.

Hold the SHIFT key and press the arrow keys on your keyboard to highlight a range of cells in a worksheet.

Hold the CTRL key while pressing the PAGE DOWN or PAGE UP keys to open other worksheets in a workbook.

**THE EXCEL RIBBON**

Excel’s features and commands are found in the Ribbon, which is the upper area of the Excel screen that contains several tabs running across the top. Each tab provides access to a different set of Excel commands. Figure 1.6 shows the commands available in the Home tab of the Ribbon. Table 1.1 “Command Overview for Each Tab of the Ribbon” provides an overview of the commands that are found in each tab of the Ribbon.

The Ribbon shown in Figure 1.6 is full, or maximized. The benefit of having a full Ribbon is that the commands are always visible while you are developing a worksheet. However, depending on the screen dimensions of your computer, you may find that the Ribbon takes up too much vertical space on your worksheet. If this is the case, you can minimize the Ribbon by clicking the button shown in Figure 1.6. When minimized, the Ribbon will show only the tabs and not the command buttons. When you click on a tab, the command buttons will appear until you select a command or click anywhere on your worksheet.

Keyboard Shortcuts

Minimizing or Maximizing the Ribbon

Hold down the CTRL key and press the F1 key.

Hold down the CTRL key and press the F1 key again to maximize the Ribbon.

QUICK ACCESS TOOLBAR AND RIGHT-CLICK MENU

The Quick Access Toolbar is found at the upper left side of the Excel screen above the Ribbon. This area provides access to the most frequently used commands, such as Save and Undo. You also can customize the Quick Access Toolbar by adding commands that you use on a regular basis. By placing these commands in the Quick Access Toolbar, you do not have to navigate through the Ribbon to find them. This will open a menu of commands that you can add to the Quick Access Toolbar. If you do not see the command you are looking for on the list, select the More Commands option.

In addition to the Ribbon and Quick Access Toolbar, you can also access commands by right clicking anywhere on the worksheet. Figure 1.8 shows an example of the commands available in the right-click menu.

**THE FILE TAB**

The File tab is also known as the Backstage view of the workbook. It contains a variety of features and commands related to the workbook that is currently open, new workbooks, or workbooks stored in other locations on your computer or network. Figure 1.9 shows the options available in the File tab or Backstage view. To leave the Backstage view and return to the worksheet, click the arrow in the upper left-hand corner as shown below.

Included in the File tab are the default settings for the Excel application that can be accessed and modified by clicking the Options button. Figure 1.10 shows the Excel Options window, which gives you access to settings such as the default font style, font size, and the number of worksheets that appear in new workbooks.

**SAVING WORKBOOKS (SAVE AS)**

Once you create a new workbook, you will need to change the file name and choose a location on your computer or network to save that file. It is important to remember where you save this workbook on your computer or network as you will be using this file in the Section 1.2 “Entering, Editing, and Managing Data” to construct the workbook shown in Figure 1.1. The process of saving can be different with different versions of Excel.  Please be sure you follow the steps for the version of Excel you are using. The following steps explain how to save a new workbook and assign it a file name.

**SAVING WORKBOOKS IN EXCEL 2013**

If you have not done so already, open a blank workbook in Excel.

When saving your workbook for the first time, click the File tab.

Click the Save As button in the upper left side of the Backstage view window. This will open the Save As dialog box

Click in the File Name box at the bottom of the Save As dialog box and use the BACKSPACE key to remove the current default name of the workbook.

Type the file name:  CH1 GMW Sales Data.

Click the Desktop button on the left side of the Save As dialog box if you wish to save this file on your desktop. If you want to save this workbook in a different location, such as a USB drive, select your preferred location.

Click the Save button on the lower right side of the Save As dialog box.

As you continue to work on your workbook, you will want to Save frequently by click either the Save button on the Home ribbon; or by selecting the Save option from the File menu.

**SAVING WORKBOOKS IN EXCEL 2016**

If you have not done so already, open a blank workbook in Excel.

Click the File tab and then the Save As button in the left side of the Backstage view window. This will open the Save As dialog box.

Determine a location for saving on your computer by clicking Browse on the left side to open the Save As dialog box.

Click in the File Name box near the bottom of the Save As dialog box. Type the new file name: CH1 GMW Sales Data

Review the settings in the screen for correctness and click the Save button.

Keyboard Shortcuts

Save As

Press the F12 key and use the tab and arrow keys to navigate around the Save As dialog box. Use the ENTER key to make a selection.

Or press the ALT key on your keyboard. You will see letters and numbers, called Key Tips, appear on the Ribbon. Press the F key on your keyboard for the File tab and then the A key. This will open the Save As dialog box.

Skill Refresher

Saving Workbooks (Save As)

Click the File tab on the Ribbon.

Click the Save As option.

Select a location on your PC.

Click in the File name box and type a new file name if needed.

Click the down arrow next to the “Save as type” box and select the appropriate file type if needed.

Click the Save button.

**THE STATUS BAR**

The Status Bar is located below the worksheet tabs on the Excel screen. It displays a variety of information, such as the status of certain keys on your keyboard (e.g., CAPS LOCK), the available views for a workbook, the magnification of the screen, and mathematical functions that can be performed when data are highlighted on a worksheet. You can customize the Status Bar as follows:

Place the mouse pointer over any area of the Status Bar and right click to display the “Customize Status Bar” list of options (see Figure 1.13).

Select the Caps Lock option from the menu (see Figure 1.13).

Press the CAPS LOCK key on your keyboard. You will see the Caps Lock indicator on the lower right side of the Status Bar.

Press the CAPS LOCK on your keyboard again. The indicator on the Status Bar goes away.

**EXCEL HELP**

The Help feature provides extensive information about the Excel application. Although some of this information may be stored on your computer, the Help window will automatically connect to the Internet, if you have a live connection, to provide you with resources that can answer most of your questions. You can open the Excel Help window by clicking the question mark in the upper right area of the screen or ribbon. With newer versions of Excel, use the query box to enter your question and select from helpful option links or select the question mark from the dropdown list to launch Excel Help windows.

Keyboard Shortcuts

Excel Help

Press the F1 key on your keyboard.

Key Takeaways

Excel is a powerful tool for processing data for the purposes of making decisions.

You can find Excel commands throughout the tabs in the Ribbon.

You can customize the Quick Access Toolbar by adding commands you frequently use.

You can add or remove the information that is displayed on the Status Bar.

The Help window provides you with extensive information about Excel.

**CATEGORIES OF DATA TYPES**

CTS separates data types into two categories

1. Value types
2. References types

Value types are plain aggregation of data instances of value types do not have referential identity not referential comparison semantics – equality and inequality comparisons for value types compares the actual data types values within the instances, unless the corresponding operators are overloaded. Value types are derived from system value types, always have default value, and can always be created and copied. Some other limitations on value types are that they cannot derive from each other (but can implement interfaces) and cannot have a default (without parameters) constructor Examples of value type are some primitive types such as (a 16- bit Unicode codepoint), and system DateTime (identifies a specific point in time with millisecond precision).

**DESIGN DOCUMENT**

The entire system is projected with a physical diagram which specifies the actual storage parameter that are physically necessary for any database too.

* Application must be stored on to the disk .the overall systems existential idea is derived from this diagram .
* The relation upon the system is structure through a conceptual ER-diagram , which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
* The contend level DFD is provided to have an idea of the functional inputs and outputs that are achieved through the system .the system depicts the input and output standards at high levell of the systems existence.

**DATA FLOW DIAGRAMS**

Provides an indication of how the data is transformed as it moves through the systemdisputes the functions and sub functions that transforms the data flow.

* The data flow diagram provides additional infomation that is used during the analysis of the information domain , and server as a basis for the modeling of functions.
* The description of each function presented in the DFD is contained is a process specifications called as PSPEC.

**ER--DIAGRAMS**

1. The Entity Relationship Diagram (ERD) depicts the relationship between the data objects . the ERD is a notation that is used to conduct the date modelling activity, the attributes of each data object noted is the ERD can be describe as the resign data object descriptions .
2. The set of primary components that are identified by the ERD are :

* Data object
* Attributes
* Relationships
* Attributes
* Various types of indicators

The primary purpose of ERD is to represent the data objects and their relationships.

UNIFIED MODELING LANGUAGE DIAGRAM

1. The unified modeling language allows the software engineer to express an analysis model using the modelingnotation that govern by the set of syntactic semantic and pragmatic rules.
2. A UML system is represented using different views that describe the system from distinctly different perspective . Each view is define by a set of diagram which is as follows.

USER MODEL VIEW

1. This view represents the system from the employees perspective.
2. The analysis representation describes the usage scenario from the end-employees’s.

STRUCTURAL MODEL VIEW

1. In this model the data and the functionality are arrived inside the system.
2. This model view models the static structures

**Chapter 7**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TESTING PHASE**

Softwaretesting is a critical element of software quality of assurance , the ultimate review of specifications , design and coding. Testing is a exposure of the system to trial input to see whether it produced the correct output .

**TESTING PHASES** :

Software testing phase includes the following :

1. Test activities are determined and test data selected.
2. The test is conducted and the results are compared with the expected results .

**SYSTEM TESTING :**

It is mainly used if the software meets its requirement . The reference documents for this process is a requirement document.

**ACCEPTANCE TESTING :**

It is performed with realistic data of client to demonstrate that the software is working properly.

**UNIT TESTING :**

Unit testing is essentially for the verification of the ends produced during the coding phase and the goal is test the internal logic of the program .This project is thoroughly tested by exposing it to various test cases regarding correct event generation, as this project passed all the test it quality is completely assured .

**Chapter 8**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**RESULTS**

The following are snapshots to the desired results on successful completion of all the phases and implementation of code.

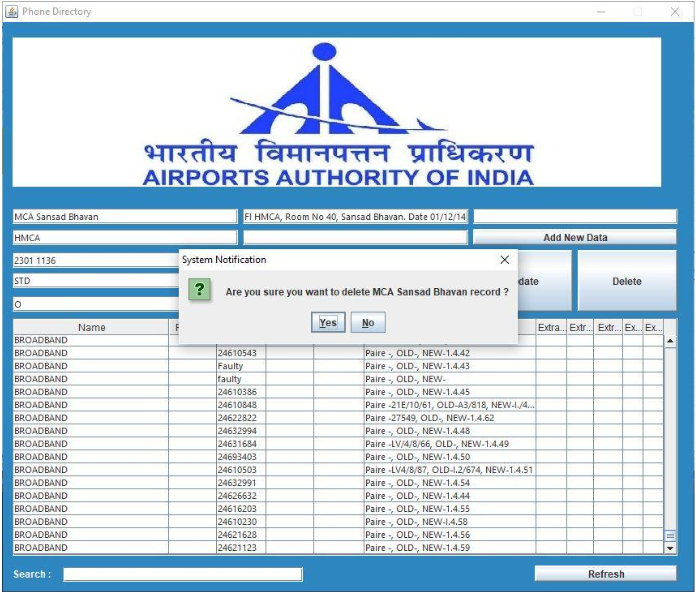


FIG 4: How data can be added into the directory.



FIG 5 : Field of the telephone directory.



FIG 6 : Overview of Database.

**REFERENCES**

[1]J. Hubbard, A. Baxter and J. Hubbard, *Programming with Java*. New York: McGraw Hill Professional, 2011.

[2]P. DuBois, *MySQL*. Upper Saddle River, NJ: Addison-Wesley, 2013.

[3]"Java Tutorial", *www.tutorialspoint.com*, 2018. [Online]. Available: https://www.tutorialspoint.com/java/. [Accessed: 08-June- 2018].

**APPENDIX**

**MAIN CODE:**

packagephonedirectory;

importjava.awt.HeadlessException;

importjava.sql.Connection;

importjava.sql.DriverManager;

importjava.sql.PreparedStatement;

importjava.sql.SQLException;

importjava.sql.Statement;

importjavax.swing.JOptionPane;

import static phonedirectory.main.a1;

public class update extends javax.swing.JFrame {

public update() {

initComponents();

}

@SuppressWarnings("unchecked")

private void initComponents() {

jButton2 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jTextField4 = new javax.swing.JTextField();

jTextField5 = new javax.swing.JTextField();

jTextField6 = new javax.swing.JTextField();

jTextField7 = new javax.swing.JTextField();

jTextField8 = new javax.swing.JTextField();

jTextField9 = new javax.swing.JTextField();

jTextField10 = new javax.swing.JTextField();

jTextField11 = new javax.swing.JTextField();

jTextField13 = new javax.swing.JTextField();

jButton3 = 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();

jLabel10 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jButton2.setText("Add New Data");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEventevt) {

jButton2ActionPerformed(evt);

}

});

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Add New Record");

setAlwaysOnTop(true);

setResizable(false);

jPanel1.setBackground(new java.awt.Color(46, 134, 193));

jTextField2.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField3.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField4.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEventevt) {

jTextField4ActionPerformed(evt);

}

});

jTextField5.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField6.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField7.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField7.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEventevt) {

jTextField7ActionPerformed(evt);

}

});

jTextField8.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField9.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField10.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField11.setFont(new java.awt.Font("Segoe UI", 0, 11));

jTextField11.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEventevt) {

jTextField11ActionPerformed(evt);

}

});

jTextField13.setFont(new java.awt.Font("Segoe UI", 0, 11));

jButton3.setText("Save Data");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEventevt) {

jButton3ActionPerformed(evt);

}

});

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setText("Name");

jLabel2.setForeground(new java.awt.Color(255, 255, 255));

jLabel2.setText("Position");

jLabel3.setForeground(new java.awt.Color(255, 255, 255));

jLabel3.setText("Phone Number");

jLabel4.setForeground(new java.awt.Color(255, 255, 255));

jLabel4.setText("Facility");

jLabel5.setForeground(new java.awt.Color(255, 255, 255));

jLabel5.setText("O\_R\_F");

jLabel6.setForeground(new java.awt.Color(255, 255, 255));

jLabel6.setText("Comment");

jLabel7.setForeground(new java.awt.Color(255, 255, 255));

jLabel7.setText("ExtraField1");

jLabel8.setForeground(new java.awt.Color(255, 255, 255));

jLabel8.setText("ExtraField2");

jLabel9.setForeground(new java.awt.Color(255, 255, 255));

jLabel9.setText("ExtraField3");

jLabel10.setForeground(new java.awt.Color(255, 255, 255));

jLabel10.setText("ExtraField4");

jLabel11.setForeground(new java.awt.Color(255, 255, 255));

jLabel11.setText("ExtraField5");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(19, 19, 19)                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()                        .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 278, javax.swing.GroupLayout.PREFERRED\_SIZE)                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)                        .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 278, javax.swing.GroupLayout.PREFERRED\_SIZE)                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)                        .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE))                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)                        .addGroup(jPanel1Layout.createSequentialGroup()                            .addComponent(jTextField11, javax.swing.GroupLayout.PREFERRED\_SIZE, 369, javax.swing.GroupLayout.PREFERRED\_SIZE)                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)                            .addComponent(jTextField13))                        .addGroup(jPanel1Layout.createSequentialGroup()                            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()                                    .addComponent(jLabel1)

.addGap(257, 257, 257)

.addComponent(jLabel2)

.addGap(239, 239, 239)                                    .addComponent(jLabel3))                                .addGroup(jPanel1Layout.createSequentialGroup()                                    .addComponent(jLabel4)                                    .addGap(91, 91, 91)

.addComponent(jLabel5)

.addGap(173, 173, 173)

.addComponent(jLabel6))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel7))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, 233, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED\_SIZE, 328, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(129, 129, 129)

.addComponent(jLabel8)))))

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED\_SIZE, 278, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel10))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField9)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel9)

.addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED\_SIZE, 203, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(97, 97, 97)

.addComponent(jLabel11)

.addGap(0, 0, Short.MAX\_VALUE))))))

.addContainerGap(20, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(19, 19, 19)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jLabel5)

.addComponent(jLabel6))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(8, 8, 8)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(jLabel8)

.addComponent(jLabel9))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel10)

.addComponent(jLabel11))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField11, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField13, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jButton3, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(20, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

pack();

setLocationRelativeTo(null);

}

private void jButton2ActionPerformed(java.awt.event.ActionEventevt) {

}

private void jButton3ActionPerformed(java.awt.event.ActionEventevt) {

Connection con = null;

Statement st = null;

try{

con = DriverManager.getConnection("jdbc:mysql://localhost/phonedirectory","root","root");

st = con.createStatement();

String sqlconn = "INSERT INTO directory(Id ,Name,Position,PhoneNumber,Facility,O\_R\_F,Comment,ExtraField1,ExtraField2,ExtraField3,ExtraField4,ExtraField5,ExtraField6) VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatementpreparedStmt = con.prepareStatement(sqlconn);

preparedStmt.setString (1,null);

preparedStmt.setString (2, jTextField2.getText());

preparedStmt.setString (3, jTextField3.getText());

preparedStmt.setString (4, jTextField4.getText());

preparedStmt.setString (5, jTextField5.getText());

preparedStmt.setString (6, jTextField6.getText());

preparedStmt.setString (7, jTextField7.getText());

preparedStmt.setString (8, jTextField8.getText());

preparedStmt.setString (9, jTextField9.getText());

preparedStmt.setString (10, jTextField10.getText());

preparedStmt.setString (11, jTextField11.getText());

preparedStmt.setString (12, jTextField13.getText());

preparedStmt.setString (13, "");

preparedStmt.execute();

this.dispose();

}catch(HeadlessException | SQLException ex){

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

}

}

private void jTextField4ActionPerformed(java.awt.event.ActionEventevt) {

}

private void jTextField7ActionPerformed(java.awt.event.ActionEventevt) {

}

private void jTextField11ActionPerformed(java.awt.event.ActionEventevt) {

}

public static void main(String args[]) {

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(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

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

public void run() {

new update().setVisible(true);

}

});

}

privatejavax.swing.JButton jButton2;

privatejavax.swing.JButton jButton3;

privatejavax.swing.JLabel jLabel1;

privatejavax.swing.JLabel jLabel10;

privatejavax.swing.JLabel jLabel11;

privatejavax.swing.JLabel jLabel2;

privatejavax.swing.JLabel jLabel3;

privatejavax.swing.JLabel jLabel4;

privatejavax.swing.JLabel jLabel5;

privatejavax.swing.JLabel jLabel6;

privatejavax.swing.JLabel jLabel7;

privatejavax.swing.JLabel jLabel8;

privatejavax.swing.JLabel jLabel9;

privatejavax.swing.JPanel jPanel1;

private static javax.swing.JTextField jTextField10;

private static javax.swing.JTextField jTextField11;

private static javax.swing.JTextField jTextField13;

private static javax.swing.JTextField jTextField2;

private static javax.swing.JTextField jTextField3;

private static javax.swing.JTextField jTextField4;

private static javax.swing.JTextField jTextField5;

private static javax.swing.JTextField jTextField6;

private static javax.swing.JTextField jTextField7;

private static javax.swing.JTextField jTextField8;

private static javax.swing.JTextField jTextField9;

}

**MySQL CODE:**

CREATE database if not exists Phonedirectory;

use Phonedirectory;

drop table if exists directory;

CREATE TABLE `directory` (

`Id` int(11) NOT NULL,

`Serial\_NO` text NOT NULL,

`Telephone` text NOT NULL,

`CA\_NO` text NOT NULL,

`Name` text NOT NULL,

`Designation` text NOT NULL,

`Facility` text NOT NULL,

`O\_R\_F` text NOT NULL,

`Address\_location` text NOT NULL,

`Mobile\_NO` text NOT NULL,

`Ownership` text NOT NULL,

`Email\_Address` text NOT NULL,

`Remarks` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `directory` (`Id`, `Serial\_NO`, `Telephone`,`CA\_NO`, `Name`, `Designation`, `Facility`,`O\_R\_F`,`Address\_location`,`Mobile\_NO`, `Ownership`, `Email\_Address`,`Remarks`) VALUES

( 1, '1', '2301 1136', '2023047725', 'MCA Sansad Bhavan', 'HMCA', 'STD', 'O', 'WI FI HMCA, Room No 40, Sansad Bhavan. Date 01/12/14', ' M No.', ' M Ownership', 'Email-', 'Remarks');

ALTER TABLE `directory`

ADD PRIMARY KEY (`Id`);

ALTER TABLE `directory`

MODIFY `Id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=20;

**JAVA UPDATE CODE:**

package phonedirectory;

import java.awt.HeadlessException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

import static phonedirectory.main.a1;

public class update extends javax.swing.JFrame {

public update() {

initComponents();

}

@SuppressWarnings("unchecked")

private void initComponents() {

jButton2 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jTextField4 = new javax.swing.JTextField();

jTextField5 = new javax.swing.JTextField();

jTextField6 = new javax.swing.JTextField();

jTextField7 = new javax.swing.JTextField();

jTextField8 = new javax.swing.JTextField();

jTextField9 = new javax.swing.JTextField();

jTextField10 = new javax.swing.JTextField();

jTextField11 = new javax.swing.JTextField();

jTextField13 = new javax.swing.JTextField();

jButton3 = 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();

jLabel10 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jButton2.setText("Add New Data");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Add New Record");

setAlwaysOnTop(true);

setResizable(false);

jPanel1.setBackground(new java.awt.Color(50, 148, 212));

jTextField2.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField3.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField4.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField4ActionPerformed(evt);

}

});

jTextField5.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField6.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField7.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField7.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField7ActionPerformed(evt);

}

});

jTextField8.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField9.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField10.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField11.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jTextField11.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField11ActionPerformed(evt);

}

});

jTextField13.setFont(new java.awt.Font("Segoe UI", 0, 11)); // NOI18N

jButton3.setText("Save Data");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setText("Serial\_NO");

jLabel2.setForeground(new java.awt.Color(255, 255, 255));

jLabel2.setText("Telephone");

jLabel3.setForeground(new java.awt.Color(255, 255, 255));

jLabel3.setText("CA\_NO");

jLabel4.setForeground(new java.awt.Color(255, 255, 255));

jLabel4.setText("Name");

jLabel5.setForeground(new java.awt.Color(255, 255, 255));

jLabel5.setText("Designation");

jLabel6.setForeground(new java.awt.Color(255, 255, 255));

jLabel6.setText("Facility");

jLabel7.setForeground(new java.awt.Color(255, 255, 255));

jLabel7.setText("O\_R\_F");

jLabel8.setForeground(new java.awt.Color(255, 255, 255));

jLabel8.setText("Address\_location");

jLabel9.setForeground(new java.awt.Color(255, 255, 255));

jLabel9.setText("Mobile\_NO");

jLabel10.setForeground(new java.awt.Color(255, 255, 255));

jLabel10.setText("Ownership");

jLabel11.setForeground(new java.awt.Color(255, 255, 255));

jLabel11.setText("Email\_Address");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(19, 19, 19)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 278, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 278, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jTextField11, javax.swing.GroupLayout.PREFERRED\_SIZE, 369, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField13))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(257, 257, 257)

.addComponent(jLabel2)

.addGap(239, 239, 239)

.addComponent(jLabel3))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel4)

.addGap(91, 91, 91)

.addComponent(jLabel5)

.addGap(173, 173, 173)

.addComponent(jLabel6))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel7))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, 233, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED\_SIZE, 328, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(129, 129, 129)

.addComponent(jLabel8)))))

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED\_SIZE, 278, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel10))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField9)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel9)

.addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED\_SIZE, 203, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(97, 97, 97)

.addComponent(jLabel11)

.addGap(0, 0, Short.MAX\_VALUE))))))

.addContainerGap(20, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(19, 19, 19)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jLabel5)

.addComponent(jLabel6))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(8, 8, 8)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(jLabel8)

.addComponent(jLabel9))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel10)

.addComponent(jLabel11))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField11, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField13, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jButton3, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(20, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

pack();

setLocationRelativeTo(null);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// Save

Connection con = null;

Statement st = null;

try{

con = DriverManager.getConnection("jdbc:mysql://localhost/phonedirectory","root","root");

st = con.createStatement();

String sqlconn = "INSERT INTO directory(Id ,Serial\_NO,Telephone,CA\_NO,Name,Designation,O\_R\_F,Facility,Address\_location,Mobile\_NO,Ownership,Email\_Address,Remarks) VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?)";

PreparedStatement preparedStmt = con.prepareStatement(sqlconn);

preparedStmt.setString (1,null);

preparedStmt.setString (2, jTextField2.getText());

preparedStmt.setString (3, jTextField3.getText());

preparedStmt.setString (4, jTextField4.getText());

preparedStmt.setString (5, jTextField5.getText());

preparedStmt.setString (6, jTextField6.getText());

preparedStmt.setString (7, jTextField7.getText());

preparedStmt.setString (8, jTextField8.getText());

preparedStmt.setString (9, jTextField9.getText());

preparedStmt.setString (10, jTextField10.getText());

preparedStmt.setString (11, jTextField11.getText());

preparedStmt.setString (12, jTextField13.getText());

preparedStmt.setString (13, "");

preparedStmt.execute();

this.dispose();

}catch(HeadlessException | SQLException ex){

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

}

}

private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jTextField7ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jTextField11ActionPerformed(java.awt.event.ActionEvent evt) {

}

public static void main(String args[]) {

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(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(update.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

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

public void run() {

new update().setVisible(true);

}

});

}

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

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.JPanel jPanel1;

private static javax.swing.JTextField jTextField10;

private static javax.swing.JTextField jTextField11;

private static javax.swing.JTextField jTextField13;

private static javax.swing.JTextField jTextField2;

private static javax.swing.JTextField jTextField3;

private static javax.swing.JTextField jTextField4;

private static javax.swing.JTextField jTextField5;

private static javax.swing.JTextField jTextField6;

private static javax.swing.JTextField jTextField7;

private static javax.swing.JTextField jTextField8;

private static javax.swing.JTextField jTextField9;

}