**PROJECT REPORT**

****

**TELEPHONE MANAGEMENT SYSTEM**

Project work submitted in partial fulfilment

of the requirement for the award of the degree

**BACHELOR OF TECHNOLOGY**

**(COMPUTER SCIENCE ENGINEERING)**



**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY**

**DELHI**

Under the guidance of

**MR. V. K. SHARMA**

**CERTIFICATE**

This is to certify that this project entitled **“TELEPHONE MANAGEMENT SERVICE”** is a bonafide work carried out by B.tech students at **AMITY SCHOOL OF ENGINEERING AND** **TECHNOLOGY (G.G.S.I.P.University), Delhi** in partial fulfilment of the requirements for the degree of Bachelor of Technology as a part of Summer Training/Internship.

**Project Guide**

Mr. V.K. SHARMA

**ACKNOWLEDGEMENT**

“Task successful” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the person who have supported us to make it a success. Success will be crowned to people who made it a reality but the people whose constant guidance and encouragement made it possible will be crowned first on the eve of success.

This acknowledgement transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped me for the completion of our project work.

I consider myself lucky enough to get such a good project. This project would add as an asset to my academic profile.

I would like to express my thankfulness to my project guide, **Mr. V.K. SHARMA** for his constant motivation and valuable help through the project work.

**CONTENTS**

1. **INTRODUCTION**
   1. INTRODUCTION TO PROJECT
   2. COMPANY PROFILE
   3. GOAL AND NEED
   4. GENERAL METHODOLOGY IN DEVELOPING SOFTWARE PROJECT
      * 1. REQUIREMENT ANALYSIS PHASE
        2. DESIGN PHASE
        3. DEVELOPMENT PHASE
        4. CODING PHASE
        5. TESTING PHASE
2. **REQUIREMENT ANALYSIS**
   1. INTRODUCTION
   2. DATA COLLECTION
      1. OBSERVATION
   3. SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT
      1. INTRODUCTION
      2. HARDWARE REQUIREMENTS
      3. SOFTWARE REQUIREMENTS
      4. DESIGN CONSTRAINTS
3. **SYSTEM ANALYSIS**
   1. MODULE DESCRIPTION
   2. FEASIBILTY ANALYSIS
   3. STUDYING THE EXISTING SYSTEM
   4. PROPOSED SYSTEM
4. **DESIGN PHASE**
   1. INTRODUCTION
   2. FLOW CHARTS
   3. DATA FLOW DIAGRAMS
5. **SCREENS**
6. **DEVELOPMENT PHASE**
   1. FEATURES OF LANGUAGE
   2. SERVER TECHNOLOGY
7. **TESTING PHASE**
8. **CONCLUSION**
9. **BIBLIOGRAPHY**
10. **APPENDICES**

**ABSTRACT**

The main objective of this system is to store the contact details of the desired persons in a database. It is a friendly easy to use interface developed in Java with MySql as the backend to store the details. This application stores all the details like name, telephone number, address, email id and website in a database. This system was developed to reduce the errors that creep up in manual systems. It was very difficult to store all the details manually there such a system was developed. This software also allows to edit, update and search various contact details. It is secure, easy to use and reliable software system. It also provides a good level of security as there is an admin who can only edit and update details. Earlier where there was only a manual system which required a lot of paperwork but now an automated system is developed which meet all the requirements.

The telephone directory system allows storing contact details in a database which is handled by the administrator. So this provides a secure system which could easily search, edit or update any records. It also allows to easily update the contact details without modifying much. The records can be modified only by the administrator. In this, you can easily log in and view the contact details. This reduces dependence on the manual system which earlier required a lot of paperwork. This also saves time and the cost of paper. All the records are safely stored in a database using MySql.

Earlier there was a manual system which requires a lot of paperwork and required a lot of time to maintain the contact records in a database. The records stored were also not secured as anyone can view it. So there was a risk of storing records. Besides this, if any changes were to be made then all things need to be updated which was very difficult to do. There was also a risk for loss of information in this system. This also lead to errors so there was a need for the new system to be developed.

**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.

**INTRODUCTION**

* 1. **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.

**ORAGNISATIONAL PROFILE**

**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**

**1.3 GOAL 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.

**1.4 GENERAL METHODOLOGY IN DEVELOPING**

**SOFTWARE PROJECT**

The general methodology in developing a system is involved in different phases, which describe the system's life cycle model for developing software project. The concept includes 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 of the 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.

**1.4.1 REQUIREMENT ANALYSIS PHASE:**

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.

**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.

**1.4.2 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.

**1.4.3 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.

**1.4.4 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.

**1.4.5 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.

**REQUIREMENT ANALYSIS**

**2.1 INTRODUCTION**

**2.2 DATA COLLECTION:**

**Observation:**

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.

**2.3 SYSTEM REQUIREMENT SPECIFICATION DOCUMENT**

**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:**

1. **Problem/Requirement Analysis:** The process is order and more nebulous of thetwo, deals with understand the problem, the goal and constraints.
2. **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.

**2.3.1 INTRODUCTION**

**2.3.1.1 PURPOSE:**

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.

**2.3.1.2 SCOPE:**

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.

**2.3.2 SOFTWARE REQUIREMENTS**

Platform - WINDOWS 10

Software - JAVA, MY SQL

**2.3.3 HARDWARE REQUIREMENTS**

Hard Disk – 2 GB.

RAM required – 1 GB (minimum)

Processor – Dual Core or Above.

**2.3.4 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.

**SYSTEM ANALYSIS**

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.

**3.1 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. Complaints

**APPLICATION**

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

Application for new phone Connection

Application for phone Transfer

Application for Modifications.

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.

**2. ENTRIES**

This module is divided into three modules. They are-

1. Bill Entry

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.

**3. 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.

**3.2 Application Enquiry:**

1. Application Enquiry for New connection
2. Application Enquiry for phone transfer
3. Application Enquiry for Temporary connection
4. **COMPLAINTS**

If employee’s has any complaints, he has to come to offer and register the complaint by specifying his phone number and name. According to complaints, those will be responded immediately.

1. Line Disturbance

2. Phone Dead

**3.2 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

1. **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.

1. **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.

1. **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.

* 1. **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.

**3.4 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.

**DESIGN PHASE**

**4.1 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.

**4.2 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 FLOW CHART**

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 form, 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.

**DISADVANTAGE:**

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

**UML DIAGRAM**

Application

Application

New phone

Telephone

Model

New phone connection

Telephone

Caller

Telephone connection

Update connection

**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.

**EXTENSIBILITY**

Visual Studio allows developers to write extensions for Visual Studio to extend its capabilities.

These extensions “plug into” Visual Studio and extend its functionality. Extensions come in the form of macros, add-ins, and packages. Macros represent repeatable tasks and actions the developers can record programmatically for saving, replaying, and distributing. Macros, However , cannot be used to implement new commands or create tool windows. They are written using Visual basic and are not compiled Add-Ins can be used to implement new functionality and can add new tool windows. Add-Ins are plugged in to the IDE via COM and can be created in any COM-compliant language. Packages are created using the Visual Studio SDK and provide the highest level of extensibility. It is used to create designers and other tools, as well as to integrated other programming languages. The Visual studio SDK provides both unmanaged as well as a managed API to accomplish these tasks. However, the managed API isn’t as comprehensive as the unmanaged one. Extensions are supported in the standard ( and higher) version of Visual Studio 2005. Express Editions do not support hosting extensions.

Visual Studio 2008 introduced the Visual Studio Shell that allows for development of a customized version of the IDE. The Visual Studio Shell defines a set of VS packages that provide the functionality required in any IDE. On top of that, other packages can be added to customize the installation. The isolated mode of the shell creates a new Application Id where the packages are installed.

It is aimed for development of custom development environments, either for a specific language or a specific scenario. The Integrated mode installs the packages into the Application Id of the professional/Standard/Team System editions, so that the tools integrate into these editions. The Visual Studio shell is available as a free download. After release of Visual Studio 2008, Microsoft created the Visual Studio Gallery. It serves as the centre location for posting information about extensions to Visual Studio community developers as well as commercial developers can upload information about their extension to Visual Studio 2008. Employees of this can rate and review the extensions to help assess the quality of extension being posted, RSS feeds to notify employees on updates to the sites and tagging features are also planned.

By design, C# is the programming language that most directly reflects the underlying Common Language infrastructure (CLI). Most of its intrinsic types correspond to value – types implemented by the CLI framework. However, the language specification does not state the code generation requirement intermediate compiler must target a common language runtime , or generate Common intermediate Language (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or FORTRAN. In practice allexisting compiler implementations target CIL.

Some notable C# distinguishing features are:

* There are no global variables or functions. All methods and members must be declared within classes. Static members of public classes can substitute for global variable and functions.
* Local variables cannot shadow variables of enclosing block, unlike C and C++ variable shadowing is often considered confusing by C++ texts.
* C# supports a strict Boolean data types, Boolean. Statement that take condition such as while and if, require an expression of a Boolean type. While C++ also has Boolean type, it can be freely convertible to Boolean, allowing to be an integer, as pointer. C# disallows this “integer meaning true or false” application approach or ground that forcing programmers to use expression that returns exactly Boolean prevent certain types of the programming mistakes such as if (a= b)(use of= instead of type).
* In C# memory address pointers can only be used within blocks specifically marked as unsafe, and programs with unsafe code need application appropriate permissions to run. Most object access is done through safe object references, which always either point to a ”live” object or have the well defined null value, it is impossible to obtain a references to a “dead” object (one which has been garbage collected), or to random block or memory.
* Managed memory cannot be explicitly freed instead, it is automatically garbage collected. Garbage collection, address memory leaks by freeing the programmer of responsibility for releasing memory with is no longer needed C# also provides direct support for deterministic finalization with the using statement ( supporting the resource Acquisition is initialization idiom).
* Multiple inheritances are not supported, although a class can implement any number of interfaces. This was a design decision by the languages lead architect to avoid complication, avoid dependency hell and simplify architectural requirements throughout CLI.
* Enumeration members are placed in their own scope.
* C# provides properties as syntactic sugar for a common pattern in which pair of methods, indexers encapsulate operation on a single attribute of a class.
* Full type reflection and discovery is available.
* C# currently (as of 3 June 2015 ) has 113 reserved method.

**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).

In contrast, reference types have the notion of referential identity – each instance of reference type is inherently distinct from every other instance, even if the data with in both instances in the same. This is reflected in default equality and inequality comparisons for reference types, which test for referential rather than structural equality, unless the corresponding operates are overloaded (such as the case for system string). In general, it is not always possible to create an instance of references type, nor to copy an existing instance, or perform a value comparison on two existing instances, though specific reference types can provides such services by exposing a public constructor or implementing a corresponding interface(such as ICIoneable or IComparable). Examples or references types are object (the ultimate base class for all other C# classes). System string (a string of Unicode characters), and system array(a base class for all C#arrays).

Both types categories are extensible with user-defined types.

**HISTORY**

After the release of Internet Information Service 4.0 in 1997, Microsoft began researching possibilities for a new web Application model that would solve common complaints about ASP, especially with regard to separation of presentation and content and being able to write “clean” code mark Anders, a manage on the IIS team, and soon Guthrie, who had joined Microsoft in 1997 after graduating from Duke University, were developed over the course of the two months by Anders and Guthrie, and Guthrie coded the initial prototypes during the Christmas holidays in 1997. Scott Guthrie in 2007.

The initial prototypes was called “XSP”, Guthrie explained in a 2007 interview that, ”people would always ask what the X stood for. At the time it really didn’t stand for anything XML started with that, XSCT started with that everything cool seemed to start with an X so that what we originally named it, ”the initial prototypes of XSP was done using java |3| but it was soon decided to build the new platform on the top of the common Language Runtime (CLR), as it offered an object-oriented programming environment, garbage collection and other features that were seen as desirable features that Microsoft’s Component Object Model Platform didn’t support Guthrie described the decision as a “Huge risk” as the success of their new web development platform would be tied to the success of the CLR, which like XSP was still in the early stages of development, so much so that XSP team was the first team at Microsoft to target the CLR.

SP+, as by this point the new platform was seen as being the successor to Active Server Pages, and the intention was to provide an easy migration path for ASP developers.

Mark Anders first demonstrated ASP+ at the ASP Connection conference in phoenix, Arizona on may 2, 2000. Demonstration to wide public and initial beta release of ASP+ (and the rest of the NET Framework) came at the 2000 in Orlando, Florida During Bill Gates’s keynotes presentation, Fujitsu demonstrated ASP+ being used in conjunction with COBOL, and support for a variety of other languages was announced, including Microsoft’s new visual basic NET and C# languages, as well as python and perl support by way of interoperability tools created by Active State.

Once the “NET” branding was decided on in the second half 2000, it was decided to rename ASP+ to ASP.NET. Mark and Anders explained on an application appearances on the MSDN show that, “the net initiative is really about a number of factors, it’s about delivering software as a service, it’s about XML and web service really enhancing the internet in terms of what it can do we really wanted to bring its name more in line with rest of the platform pieces that make up the.NET framework.”

Net pages, knows officially as “web forms”, are the main building block for Application development. Web forms are contained in files with an “.aspx” extension; in programming jargon, these files typically contain static (X)HTML markup, as well as markup defining server-side Web controls and user controls where the developers place all the required static and dynamic content for web page. Additionally, dynamic code which runs on the server can be placed in the pages with in a block <%--dynamic code --%> ehich is similar to other web development technologies such as PHP,JSP and ASP, but this practice is generally discouraged except for the purposes of data binding since it requires more calls when rendering the page.

**CODE-BEHIND MODEL**

It is recommended by Microsoft for dealing with dynamic program code to use the code behind model, which places this code in a separate file or in a specially designated script tag code-behind files typically have names like Mypage.aspx.cs or Mypage.aspx vb [same filename as the ASPXfile, with the final extension denoting the page language ] This practice is automatic in Microsoft visual studio and other IDEs. When using this style of programming , the developer writes code to respond to different events , like the page being loaded, or a control being clicked, rather than a procedural walk through the document .

ASP NETs code-behind model marks a departure from classic ASP in that it encourages developers to build application with separation of presentation and content in mind. In theory, this would allow a web designer for example to focus on the design markup with less potential for disturbing the programming code that drives it. This is similar to the separation of the controller from the view in model-view-controller frameworks. In this case, this page\_load0 method is called every time the ASPX page is requested. The programmer can implement event handlers at several stages of the page execution process to perform processing.

**CODE :**

**MAIN.JAVA**

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(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.ActionEvent evt) {

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.ActionEvent evt) {

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.ActionEvent evt) {

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.ActionEvent evt) {

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.ActionEvent evt) {

}

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

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(?,?,?,?,?,?,?,?,?,?,?,?,?)";

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;

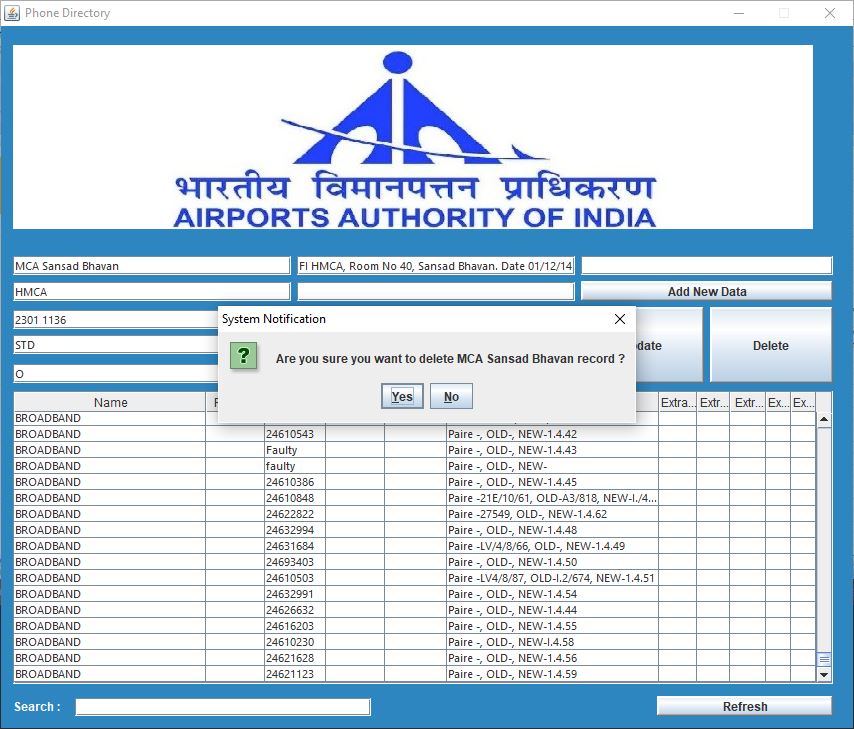
private static javax.swing.JTextField jTextField8;

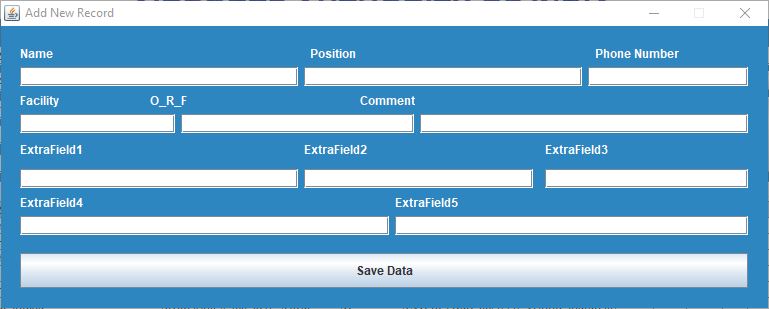
private static javax.swing.JTextField jTextField9;

}

**SCREENSHOT**







**STATE MANAGEMENT**

ASP.NET application are hosted in a web server and are accessed over the stateless HTTP protocol. As such, uses stately interaction, it has to implement state management on its own. ASP.NET provides various functionality for state management in ASP.NET Application. Conceptually, Microsoft treats ”state” as mostly GUI state, big problems may arise when an application needs to keep track of “data state” such as a finite state machine that may be in a transient state between request(lazy evaluation ) or just takes long to initialize.

**APPLICATION STATE**

Application state is a collection of user-defined variables that are shared by an ASP.NET Application. These are set and initialized when the application\_Onstate event fires on the loading of the first instances of the Application and are available till the last instances exits. Application state variables are accessed using the application collection, which provides a window application for the application state variables. Application state variable are identified by names.

**SESSION STATE**

Session state is a collection of user-defined session variables, which are persisted during a user session. These variables are unique to different instances of a user session, and are accessed using the session collection. Session variables can be set to be, even if the session does not end. At the client end, a user session is identified either by a cookie or by encoding the session ID in the URL itself.

ASP.NET supports three modes of persistence for session variable:

In process mode

The session variables are maintained within the ASP.NET process. This is the fastest way. However, in the mode variables are destroyed when the ASP.NET process is recycled or shut down. Since the Application is recycled from time to time this mode is not recommended for critical Application, rather in practice this mode is not recommended for any Application.

**ASP STATE MODE**

In this mode, ASP.NET runs a separate windows services that maintains the state variables. Because the state management host applications outside the ASP.NET process and NET Remoting must be utilized by the ASP.NET engine to access the data, this mode has a negative impact on performance in comparison to the in process mode, although this mode allows an ASP.NET. application to be load-balanced and scaled across multiple server. However, since the state management service runs independent of ASP.NET, the session variables can persist across ASP.NET process shutdown.

Same problem arises though – since session state server runs as a single instance it is a single point of failure as far as session state is concerned. This service can not be load balanced and also imposes restrictions on types that can be stored in a session variable.

Class that inherits from “System Web UI Page” , write methods here that render HTML, and the make the pages in their application inherit from this new class. While this allows for common elements to be reused across a site , it adds complexity and mixes source code with markup.

Furthermore , this method can only be visually tested by running the application-not while designing it. Other developers have used include files and other tricks to avoid having to implementation same navigation and the other elements in every page. ASP.NET 2.0 introduced the concept of “master pages” which allow for template-based page development. A web Application can have one or more master pages, which beginning with ASP.NET 3.5 , can be nested. Master templates have place-holder controls, called Content Place Holder to denote where the dynamics. Content goes, as well as HTML and JavaScript shared across child pages.

Child pages used those contents Place Holder control, which must be map application to the place holder of the Master Page that the content page is populating. The rest of the page is defined by shared parts of the master page, much like a mail merge in word processor. All markup and server controls in the content page must be placed. Within the content place holder control.

When a request is made for a contend page, ASP.NET merges the output of the contend page with the output of the master page , and sends the output to the user. The master page remains fully accessible to the contend page. This means that the contend page may still manipulate headers, change title , configure caching etc. If the master page exposes public properties or methods [ e.g. for setting copyright notices ] the contend page can use these as well.

**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**

1. Provides an indication of how the data is transformed as it moves through the system
2. disputes 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 modeling notation 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

**TESTING**

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 .

**CONCLUSION**

It has been a great pleasure for me to work on this exciting and challenging project. This

project proved good for me as it provided practical knowledge of not only programming in developing web enabled application and client server technology that will be great

demand in future. This will provide better opportunities and guidance in future in

developing projects independently.

BENEFITS:

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

It’s a web-enabled project.

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

The user is mainly more concerned about the validity of the data, whatever he is

There are checks on every stages of any new creation, data entry or updating so |

user cannot enter the invalid data, which can create problems at later date.

Sometimes the user finds in the later stages of using project that he needs to update

of the information that he entered earlier.

There are options for him by which he

update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.

User is provided the option of monitoring the records he entered earlier. He can see desired records with the variety of options provided by him.

Easier and faster data transfer through latest technology associated with the computer communication.

Through these features it will increase the efficiency, accuracy and transparency

LIMITATIONS:

The size of the database increases day-by-day, increasing the database back up and data maintenance activity.

**BIBLIOGRAPHY\**

* A complete guide to JAVA programming
* E Balaguruswami
* JAVA training course : Tutorials Point
* Advanced programming using Advanced JAVA
* The complete reference MYSQL
* Vikram Vaswani