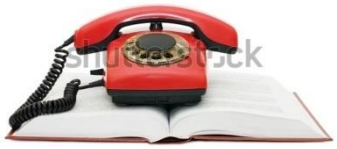
A PROJECT REPORT ON

PHONE DIRECTORY APPLICATION USING DOUBLE LINKED LIST



Submitted in partial fulfilment for the award of
Bachelor Of Computer Application
Under Berhampur University, Odisha

Submitted by
P.SUMAN REDDY
PANKAJ SAHU
SASWAT STITAPRANGYA
SANKAR NARAYAN PANDA

GUIDED BY
MR.BIBEKANAND PATTNAIK
Senior Lecturer in Department of Computer Application

DISHA COLLEGE OF MANAGEMENT AND TECHONOLOGY
JAGANATHPUR,NEAR SIDHA BHAIRABI TEMPLE

Disha College of Management and Technology

Near Sidha Bhairabi Temple, Jagannathpur, Ganjam, Odisha

CERTIFICATE

This is certified that "**Phone Directory Application Using Double Linked List"**embodies the original work done by **"P.suman Reddy, Pankaj Sahu, Saswat stitaprangya and ,Sankar Narayan Panda** during the session 2021-2022 project submission as a partial fulfillment of the requirement for the project based on " JAVA" technology For the completion of Bachelor Of Computer Application (B.C.A)at Disha College Of Management and Technology,jaganathpur

Signature Of Guide

Signature Of External

Signature Of Internal

Disha College of Management and Technology

Near Sidha Bhairabi Temple, Jagannathpur, Ganjam, Odisha

CERTIFICATE OF APPROVAL

This is certified that "**Phone Directory Application Using Double Linked List"**embodies the original work done by **"P.suman Reddy, Pankaj Sahu, Saswat stitaprangya and ,Sankar Narayan Panda** during the session 2021-2022 project submission as a partial fulfillment of the requirement for the project based on " JAVA" technology For the completion of Bachelor Of Computer Application (B.C.A)at Disha College Of Management and Technology,jaganathpur under the guidance of **"Mr Bibekanand Pattnaik"**

Signature Of principle

Disha College of Management and Technology

Near Sidha Bhairabi Temple, Jagannathpur, Ganjam, Odisha

DECLARATION

We hereby declare that the project entitled **"PHONE DIRECTORY APPLICATION USING DOUBLE LINKED LIST"**submitted to Berhampur university,is a record of an original work done by us under the guidance of **"Mr Bikeknanda pattnaik"**,HOD of computer science Dept,Disha college of management and Technology,and this project fulfilled of the requirement for the award of the degree of Bachelor of Computer Application(BCA)final year.This original work is done by us and no part of it has been ever submitted for any degree of diploma of any institution previously under our knowledge.

P.Suman Reddy

RollNo-DI43620

Regd No-41492/2019

Pankaj Sahu

Roll No-DI43820

Regd No-41494/2019

Saswat Stitaprangya

Roll No- DI45920

Regd No-41515/2019

Sankar Narayan panda

Roll No-DI46120

Regd No-41517/2019

Disha College of Management and Technology

Near Sidha Bhairabi Temple, Jagannathpur, Ganjam, Odisha

ACKNOWLEDGEMENT

We pay our deep sence of gratitude to **"Mr Bibekanand pattnaik",**HOD of BCA Department,Disha college of management and technology,Berhampur to encourage us to highest peak and to provide us to opportunity to prepare the project.we immensely obliged to our friends for their elevating inspiration encouraging guidance and kind supervision in the comletation of our project.

We feel to acknowledge indebtedness sence of gratitude to our guide **"Mr Bibekanand Pattnaik"**Lecture of BCA Department whose valuable guidance and kind supervision given to us through out the course which shaped the present work as its show.

P.Suman Reddy

RollNo-DI43620

Regd No-41492/2019

Pankaj Sahu

Roll No-DI43820

Regd No-41494/2019

Saswat Stitaprangya

Roll No- DI45920

Regd No-41515/2019

Sankar Narayan panda

Roll No-DI46120

Regd No-41517/2019

CONTENT

Introduction

INTRODUCTION:

- It is estimated that there are more than 5.07b mobile phone users around the world& the number is increasing.
- The phone book application works specifically for tracking people
- The concept of file management and data structure is obtain used in almost all function in the project.it uses functions,file management and data structure.

OBJECTIVE:

- We are determined to design a system which is intended is saving name and number of desired persons. The main objective of phone directory application is to add , search ,edit and delete various contacts.
- Phone directory application is a project that is provide by technical assignment help to us in that we get a simple SQL based solution to store our contacts.
- The names are presents in user-sated formats like alphabetical order, according to time date ,recently etc . So that user also can easily find the required person along with their address and telephone numbers and use search option.

System Analysis

Ref	Function	Category
R1.11	The System should display all the links, services & functionalities provided by the code directory	Evident
R1.12	The System should show the links to following modules • User Module • Register mobile no with email id and address • Multi- Phone no support	Evident
R1.13	The System should have a link from all the modules to their respective pages. • Personal no is the main key of the application • Side by side add two numbers work and home	Evident
R1.14	The System should provide facility for registering the appropriate authority. To be use. • Search by user • Store address • Stored Additional Email id	Evident

Feasibility Study

FEASIBILITY STUDY:

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The system has been tested for feasibility in the following points.

1. Technical Feasibility
2. Economical Feasibility
3. Operational Feasibility.

1. Technical Feasibility

The project entitles "Phone Directory System" is technically feasibility because of the below mentioned feature. The project was developed in Java which Graphical User Interface

It provides the high level of reliability, availability and compatibility. All these make Java an appropriate language for this project. Thus the existing software Java is a powerful language.

2. Economical Feasibility

The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90% approximately. The costs incurred of not creating the system are set to be great, because precious time can be wanted by manually.

3. Operational Feasibility

In this project, the management will know the details of each project where he may be presented and the data will be maintained as decentralized and if any inquires for that particular contract can be known as per their requirements and necessaries.

Implementation:

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation.

Requirement Specification

HARDWARE AND SOFTWARE CONFIGURATION

The hardware and software should be chosen carefully keeping following point in mind

- The System must be user friendly
- The System must be able to handle large number of data.
- Processing speed of the system must be fast

Hardware configuration

Main processor : Pentium PIV or Higher
Random access memory : 512 MB or Higher
Hard disk capacity : 40GB or higher

Software configuration

Operating system : Windows (xp2 or Higher)

Programming specification : java

Documentation tools : ms-office

Framework/ID : J Creator Pro

Database : SQLServer

3.1 SDLC Methodology:

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

The steps for Spiral Model can be generalized as follows:

1. The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.

2. A preliminary design is created for the new system.

3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

4. A second prototype is evolved by a fourfold procedure:

a. Evaluating the first prototype in terms of its strengths, weakness, and risks.
b. Defining the requirements of the second prototype.
c. Planning adesigning the second prototype.
d. Constructing and testing the second prototype.

Advantages:

a. Estimates(i.e. doctor avalabity, schedule etc .) become more relistic as work progresses, because important issues discoved earlier.
b. It is more able to cope with the changes that are software development generally entails.
c. Software engineers can get their hands in and start woring on the core of a project earlier.

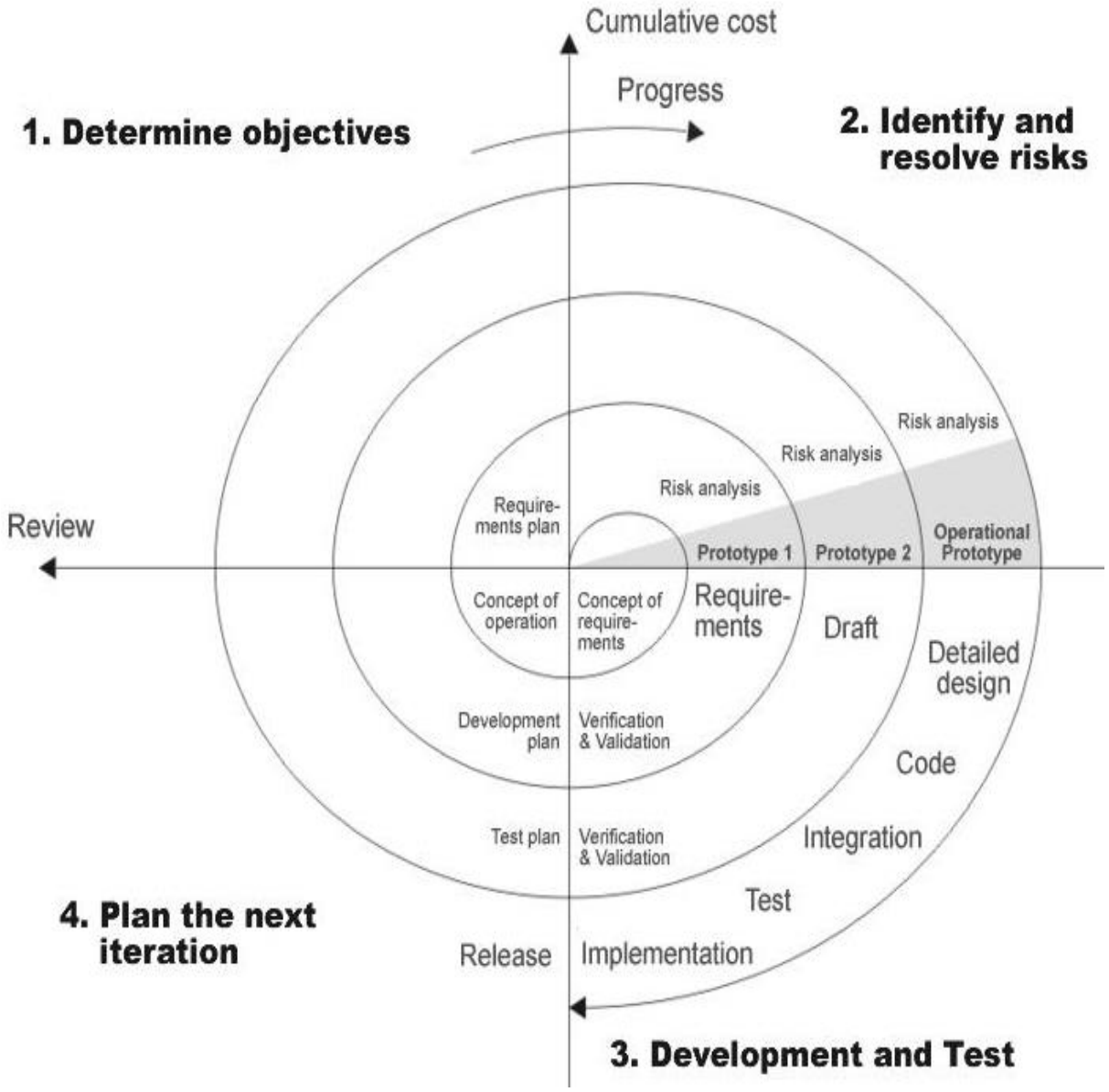


Figure 3.1 Spiral Model

System Design

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance 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 customer's requirements 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. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

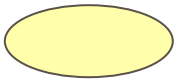
UML Diagrams:

Actor:

A coherent set of roles that users of use cases play when interacting with the use `cases.

Use case:

A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.



UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

There are various kinds of methods in software design:

They are as follows:

- Use case Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- State chat Diagram

USECASE DIAGRAMS:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor.

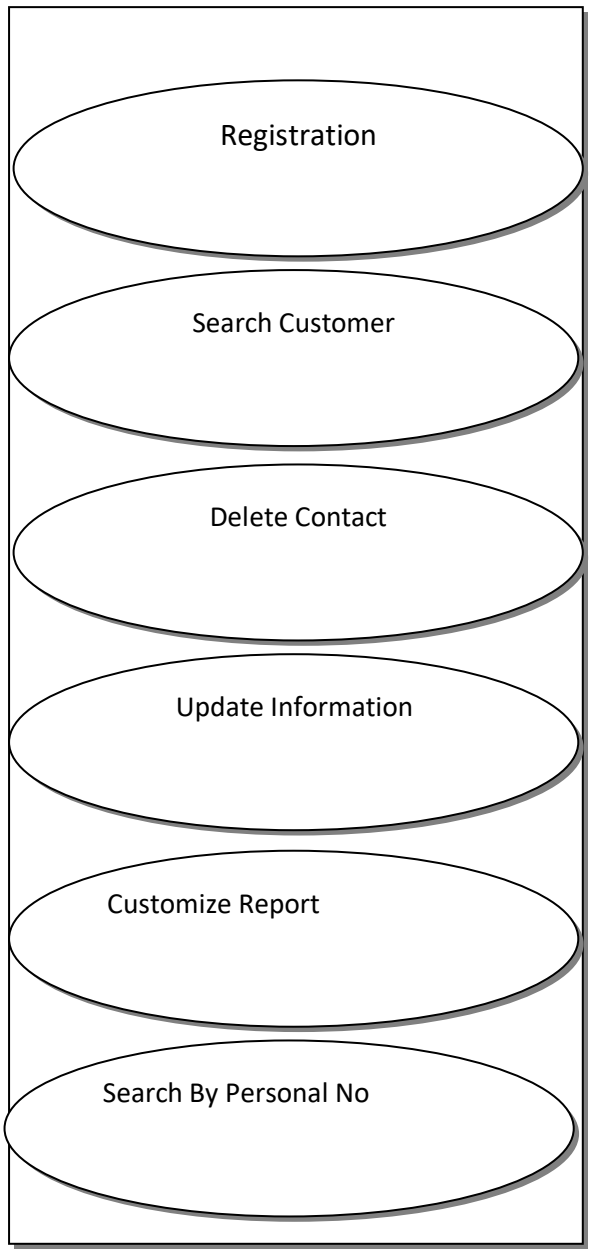
Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do.Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

USECASE DIAGRAM:

A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

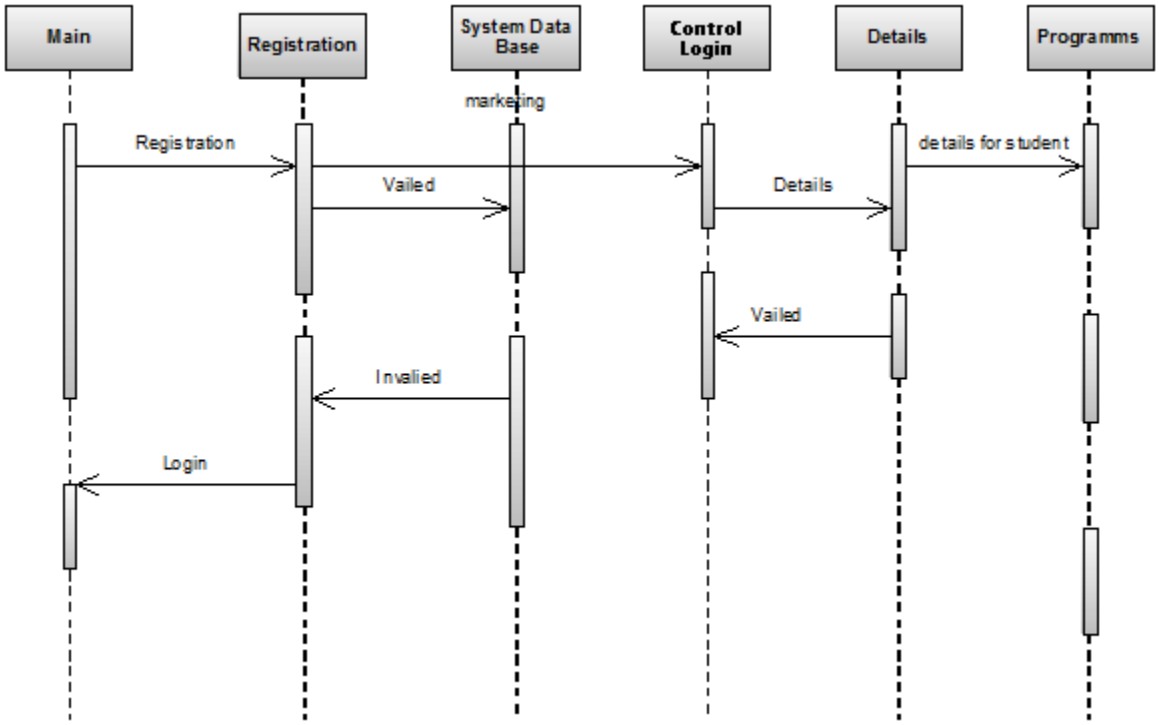
Use Case Diagram



Sequence diagram

Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.

A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis



DATA FLOW DIAGRAMS:

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software.

Data objects represented by labeled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

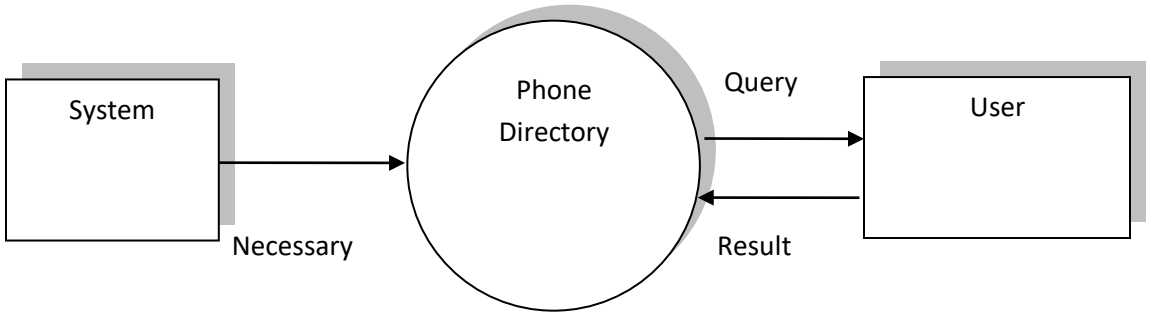
The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

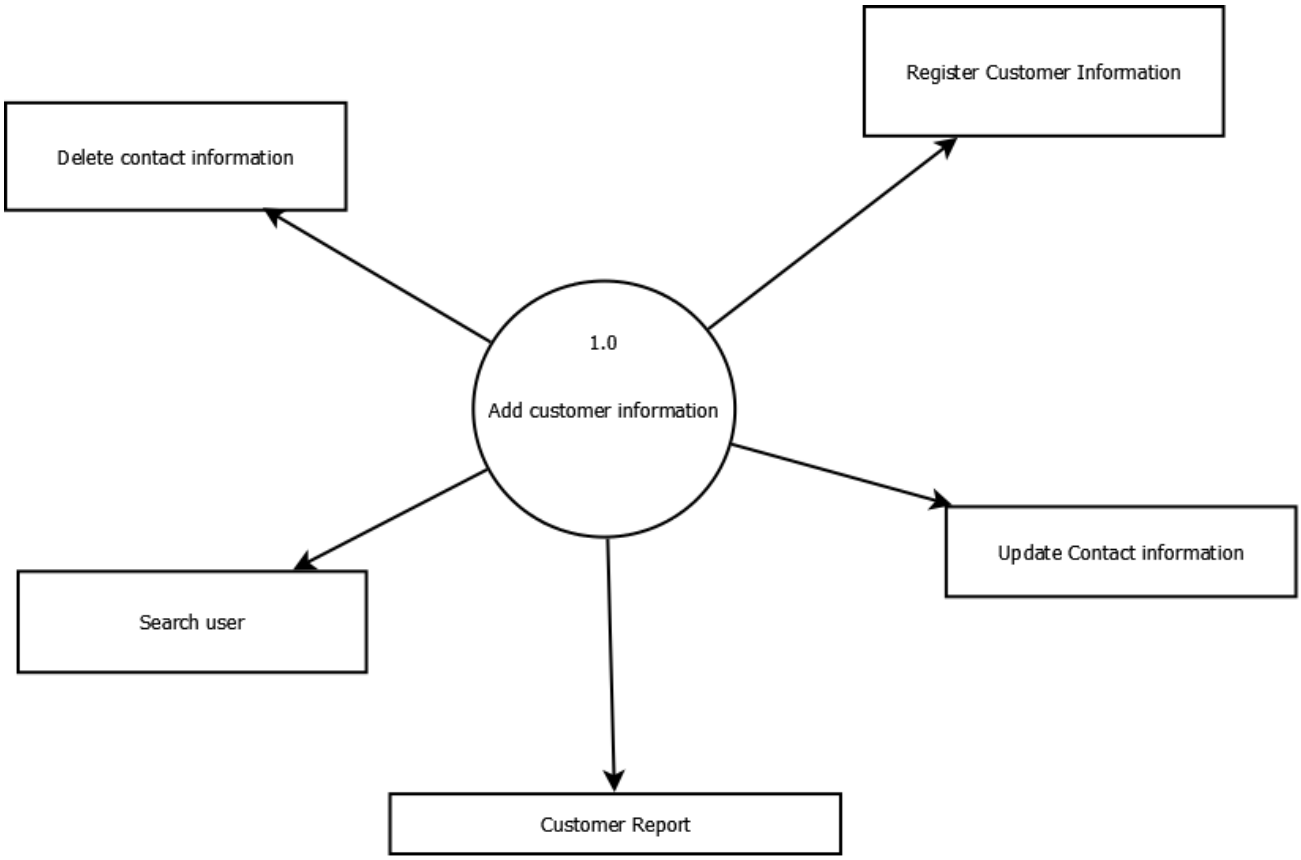
RULES FOR DFD:

- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions
- Reads left to right and top to bottom.
- Identify all inputs and outputs.
- Label data flow on each arrow
- Do not indicate hardware and ignore control information..
- There must not be unnamed process.
- Use details flow on each arrow
- Number each occurrence of repeated external entities..

DFD (0th Label)



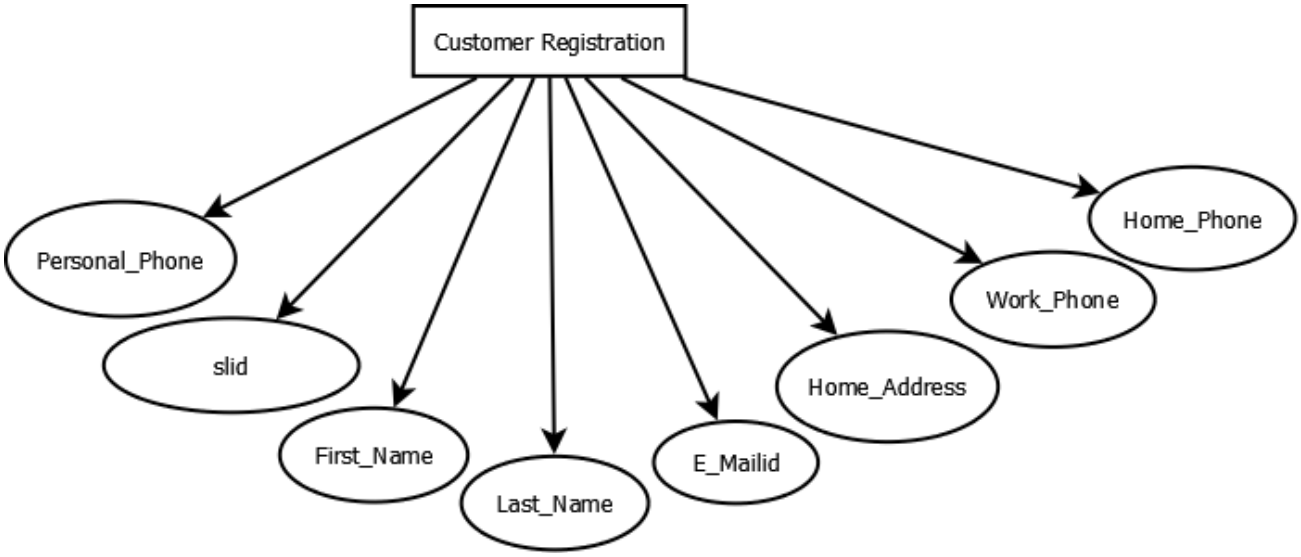
DFD(1th) Level



E-R Diagrams:

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represents data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design For the database designer, the utility of the ER model is:

- it maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- it is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.



Software Tools and Languages used

Front-End Specification

A.1 Overview of Java Language

JAVA is a Platform Independent Programming Language. JAVA was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991. It took 18 months to develop the first working version. This language was initially called “Oak” but was renamed “JAVA” in 1995.

A.2 Object Oriented Programming and JAVA

Object-oriented Programming was developed because of limitations found in earlier approaches of programming. To appreciate what OOP does, we need to understand what these limitations are and how they arose from traditional programming.

A.3 the Object Oriented Approach

The fundamental idea behind object-oriented languages is to combine into a single unit both data and the functions that operate on that data. Such a unit is called an object. An object’s functions, called member methods in JAVA, typically provide the only way to access its data. If you want to read the item and return the value to you, you call a member function in the object. It will read the item and return the value to you. You can’t access the data directly. The data is hidden, so it is safe from accidental modification. Data and its functions are said to be encapsulated into a single entity. Data encapsulation and data hiding are key terms in the description of object oriented languages. If you want to modify the data in an object, you know exactly what functions interact with it: the member functions in the object. No other functions can access the data. This simplifies writing, debugging, and maintaining the program. A JAVA program typically consists of a number of objects, which communicate with each other by calling one another’s members functions. We should mention that what are called member functions in C++ are called methods in JAVA. Also, data items are referred to as instance variables. Calling an object’s member function is referred to as sending a message to the object.

A.4 An Analogy

You might want to think of objects as departments—such as sales, accounting, personnel, and so on—in a company. Departments provide an important approach to corporate organization. In most companies (except very small ones), people don't work on personnel problems one day, the payroll the next, and then go out in the field as sales people the week after. Each department has its own personnel, with clearly assigned duties. It also has its own data: payroll, sales figures, personnel records, inventory, or whatever, depending on the department.

The people in each department control and operate on those departments data. Dividing the company into departments makes its easier to comprehend and control the company's activities, and helps them maintain the integrity of the information used by the company. The payroll department, for instance, is responsible for the payroll data. If you are from the sales department, and you need to know the total of all the salaries paid in the southern region in July, you don't just walk into the payroll department and start rummaging through file cabinets.

You send a memo to the appropriate person in the department, and then you wait for that person to access the appropriate person in the department, and then you wait for that person to access the data and send you a reply with the information you want. This ensures that the data is accessed accurately and that it is not corrupted by inept outsiders. (This view of corporate organization is show in figure). In the same way, objects provide an approach to program organization, while helping to maintain the integrity of the programs data.

A.5 OOP: An Approach to Organization

Keep in mind that object-oriented programming is not primarily concerned with the details of program operation. Instead, it deals with the overall organization of the program.

A.6 Characteristics of Object-Oriented Languages

Let's briefly examine a few of the major elements of object-oriented languages in general and JAVA in particular.

A.6.1 Objects

When you approach a programming problem in an object oriented language, you no longer ask how the problem will be divided into functions, but how it will be divided into objects. Thinking in terms of objects, rather than functions, has a surprisingly helpful effect on how easily programs can be designed and objects in the real world.

What kinds of things become objects-oriented programs? The answer to this is limited only by your imagination, but there are some typical categories to start you thinking:

A.6.2 Physical Objects

- Automobile in a traffic-flow simulation
- Electrical components in a circuit design to a program
- Countries in an economics model
- Aircraft in an air-traffic control system

Elements of the Computer-User Environment

- Windows
- Menus
- Graphics objects(lines ,rectangles, circles)
- The mouse and the keyboard

Programming Constructs

- Customized arrays
- Stacks
- Linked lists

Collection of Data

- An inventory
- A personnel file
- A dictionary

A table of the latitudes and longitudes of world cities.

User Defined Data Types

- Time
- Angles
- Complex numbers
- Points on the plane

Components in a Computer Games

- Ghosts in maze game
- Positions in a board game (chess, checkers)
- Animals in an ecological simulation
- Opponents and friends in adventure games

The match between programming objects and real-world objects us the happy result of combining data and functions: the resulting objects offer a revolution in program designing, no such close match between programming constructs and the items being modeled exists in a procedural language.

A.6.3 Classes

In OOP we say that objects are members of classes. What does this mean? Let's look at an analogy. Almost all computer languages have built-in data types. For instance, a data type int, meaning integer is pre-defined in JAVA. You can declare as many variables of type int as you need in your program:

- Int day;
- Int count;
- Int divisor;
- Int answer;

. A class is thus a collection of similar objects. This fits our non technical understanding of the word class, Prince, sting etc., and are members of the class of rock musicians. There is no person called rock musician but specific people with specific names are members of this class if they possess certain characteristics.

A.6.4 Abstraction

An essential element of object-oriented programming is abstraction. Humans manage complexity through abstraction. For example, people do not think of a car as a set of tens of thousands of individual parts. They think of it as a well-defined object with its own unique behavior. This abstraction allows people to use a car to drive to the grocery store without being overwhelmed by the complexity of the parts that form the car. They can ignore the details of how the engine, transmission, and braking systems work. Instead they are free to utilize the object as a whole.

A.6.5 Encapsulation

Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse. One way to think about encapsulation is as a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside the wrapper. Access to the code and data inside the wrapper is tightly controlled through a well-defined interface.

A.6.6 Inheritance

Inheritance is the process by which one object acquires the properties of another object. This is important because it supports the concept of hierarchical classification. As mentioned earlier, most knowledge is made manageable by hierarchical (that is, top-down) classifications. For example, a Golden Retriever is part of the classification dog, which in turn is part of the mammal class, which is under the larger class animal. Without the use of hierarchies, each object would need to define all of its characteristics explicitly. However, by use of inheritance, an object need only define those qualities that make it unique within its class. It can inherit its general attributes from its parent. Thus, it is the inheritance mechanism that makes it possible for one object to be a specific instance of a more general case.

A.6.7 Polymorphism

Polymorphism (from the Greek, meaning “many forms”) is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation. Consider a stack (which is a last-in, first-out list). You might

have a program that requires three types of stack. One stack is used for integer values, one for floating-point values, and one for characters. The algorithm that implements each stack is the same, even though the data being stored differs. In a non-object-oriented language, you would be required to create three difference sets of stack routines, with each set using different names. However, because of polymorphism, in JAVA you can specify a general set of stack routines that all share the same names.

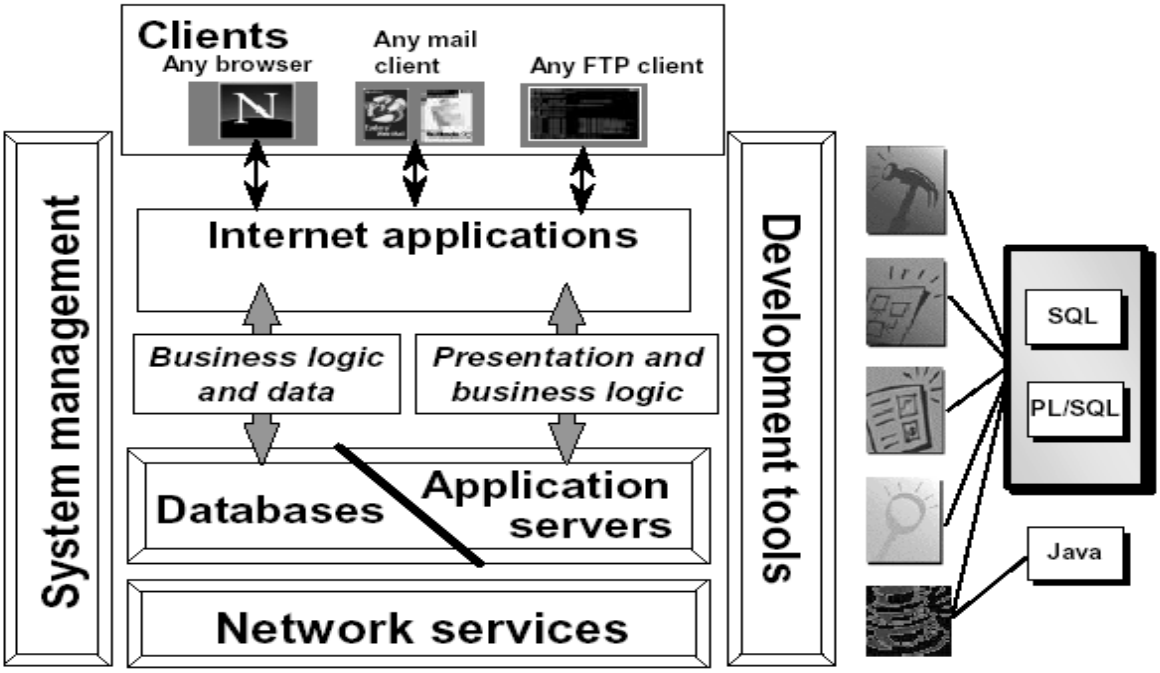
A.6.8 Polymorphism, Encapsulation, and Inheritance Work Together

When properly applied, polymorphism, encapsulation, and inheritance combine to produce a programming environment that supports the development of far more robust and scaleable programs than does the process-oriented model. A well-designed hierarchy of classes is the basis for reusing the code in which you have invested time and effort developing and testing. Encapsulation allows you to migrate your implementations over time without breaking the code that depends on the public interface of your classes. Polymorphism allows you to create clean, sensible, readable, and resilient code.

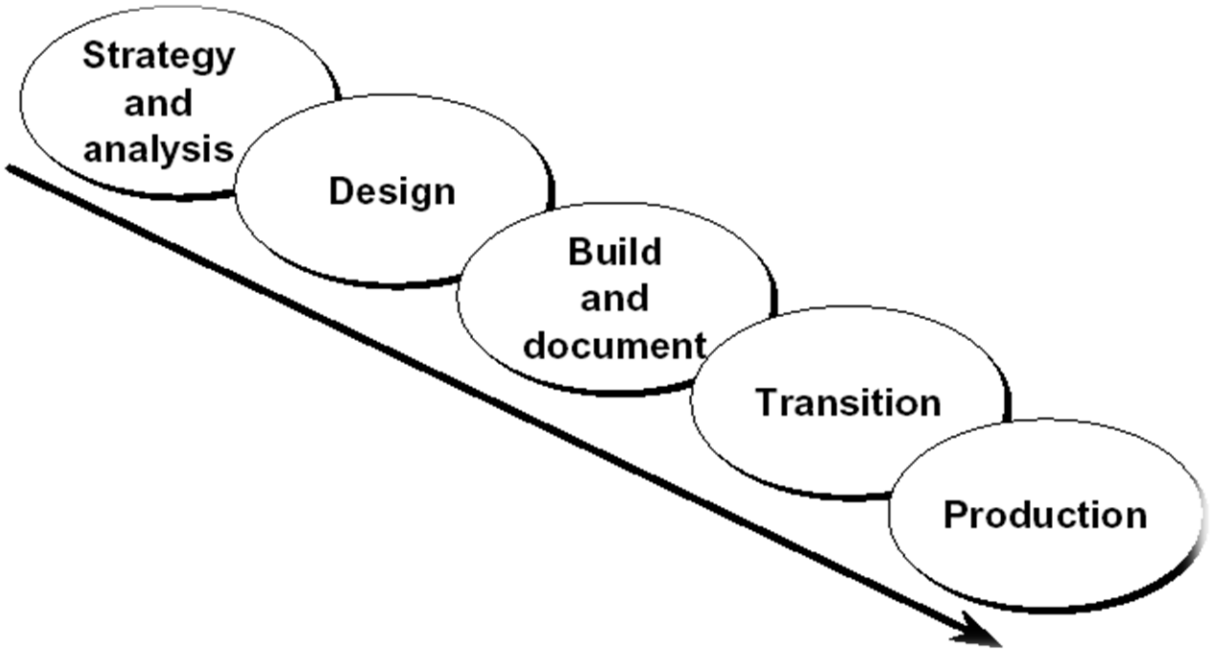
A.7 Why JAVA is Important to Internet

The internet helped catapult JAVA to the forefront of programming, and JAVA, in turn, has had a profound effect on the internet. The reason is quite simple: JAVA expands the universe of objects can move about freely in cyberspace. In a network two very broad categories of objects are transmitted between the server and client: passive information and dynamic, active programs. For example when we read our email, we are viewing passive data. However a second type of object can be transmitted to the client's computer yet is initiated by the server. For example, a program might be provided by the server to display properly the data that the server is sending.

Development tools



System Development Life Cycle



System Development Life Cycle

System Development Life Cycle

From concept to production, you can develop a database by using the system development life cycle, which contains multiple stages of development. This top-down, systematic approach to database development transforms business information requirements into an operational database.

Strategy and Analysis

- Study and analyze the business requirements. Interview users and managers to identify the information requirements. Incorporate the enterprise and application mission statements as well as any future system specifications.
- Build models of the system. Transfer the business narrative into a graphical representation of business information needs and rules. Confirm and refine the model with the analysts and experts.

Design

Design the database based on the model developed in the strategy and analysis phase.

Build and Document

- Build the prototype system. Write and execute the commands to create the tables and supporting objects for the database.
- Develop user documentation, Help text, and operations manuals to support the use and operation of the system.

Transition

Refine the prototype. Move an application into production with user acceptance testing, conversion of existing data, and parallel operations. Make any modifications required.

Project Schedule (Gantt chart)

WHAT ANALYSIS MEANS ? (A BRIEF INTRODUCTION)

Any kind of manual system when required to be computerized then the first and foremost job is to study the existing system and the activities in the existing system and to identify the limitation of the system. It may not happen that every system being computerized is likely to yield better performance and efficiency as well as profitability. Although it is an establish fact that computerization normally results overall efficiency in a particular system. In order to gauge the profitability in terms of computerizing a system; first of all we go for a feasibility study which is otherwise called as cost / benefit analysis.

The project is going to be evaluated in terms of:

1. Technical viability: - Description of different methods
2. Economic viability: - Description of different methods

   Once the viability test is conducted and the outcome of the test is positive then the decision is taken in favor of computerizing the system. If the result is negative then the objective of computerization the system is dropped.

   Once the study passes the evaluation test successfully then the system development is done through certain distinct identified steps. All of the system combined together is known as System Development Life Cycle (SDLC).

The SDLC contains 7 distinct phases in it these are:

1. Feasibility study
2. Determination of system's requirements
3. Design of the system
4. Development of software
5. System testing
6. System implementation

GANTT CHART

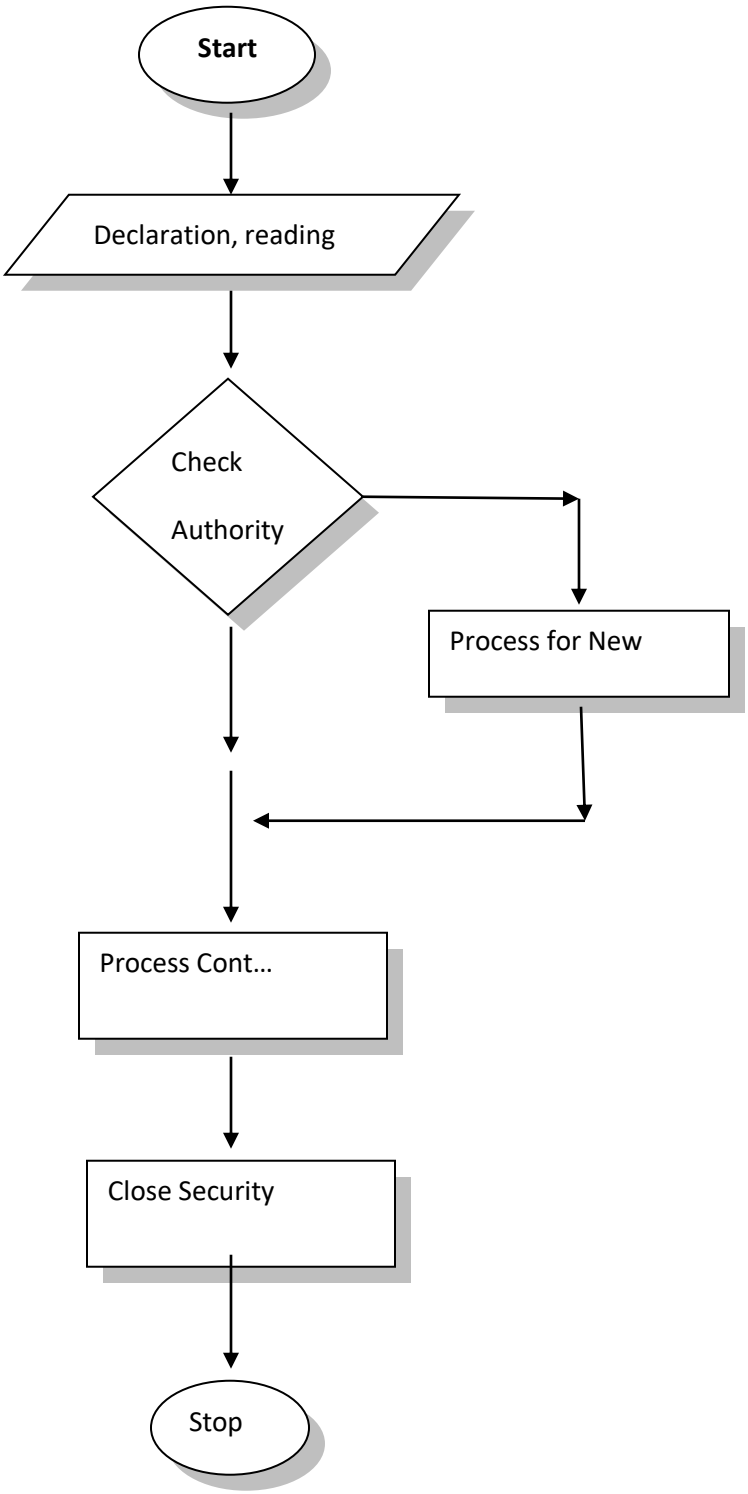
It is also knows as Bar chart is used exclusively for Hennry L. Gantt developed by scheduling purpose. It is a project controlling technique. It is used for scheduling. Budgeting and resourcing planning. A Gantt is a bar chart with each bar representing

activity. The bars are drawn against a time line. The length of time planned for the activity.This application software package for AUTOMATION OF OCA'S estimate number of days required for each of the seven tasks are as follows.

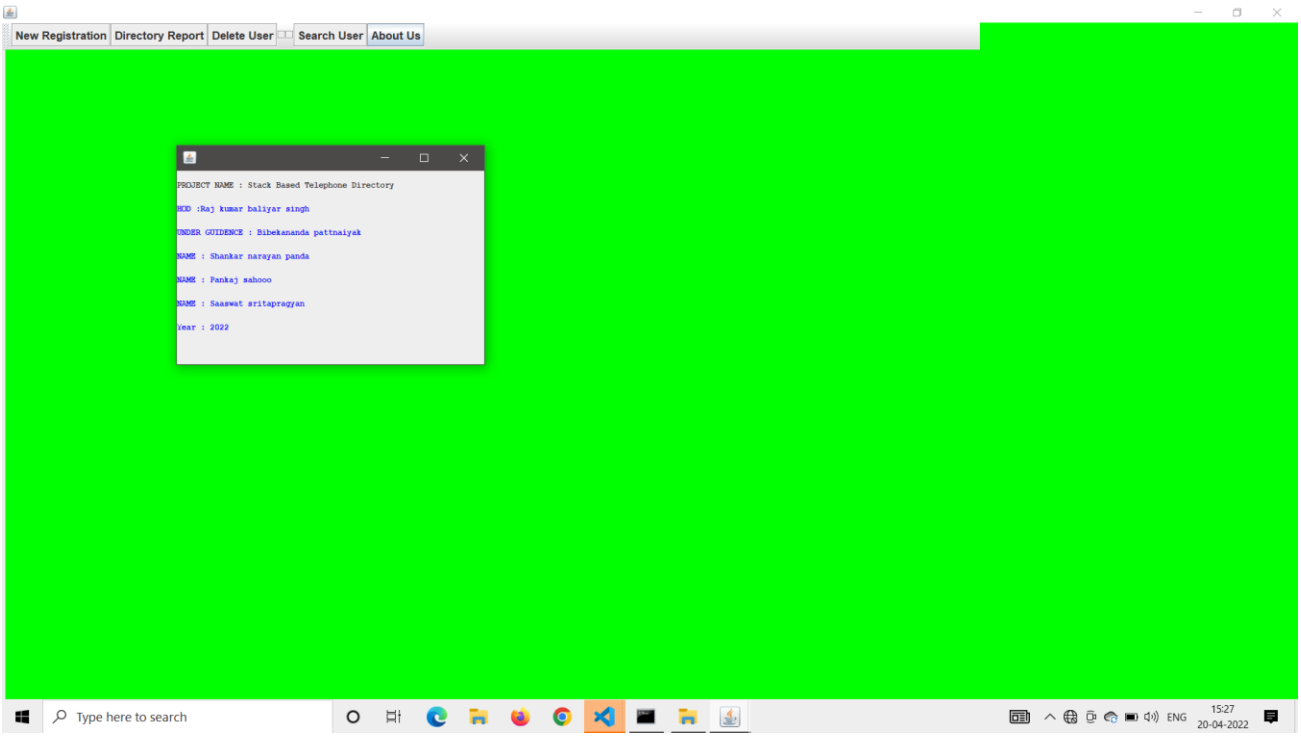
1. Review of the current system, interviews with people and review of documentation have taken approximately five person ten days to complete.
2. Designing tasks 60 days, which includes analysis of facts determination of feasibility, determination of user requirements and completion of system analysis reports
3. Building up database 20 days
4. Linking of database
5. Code generation 60 days

Integration and testing 30 days DEVELOPMENT	Day duration														
	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16
Structure															
Development															
Testing															
Documentation															

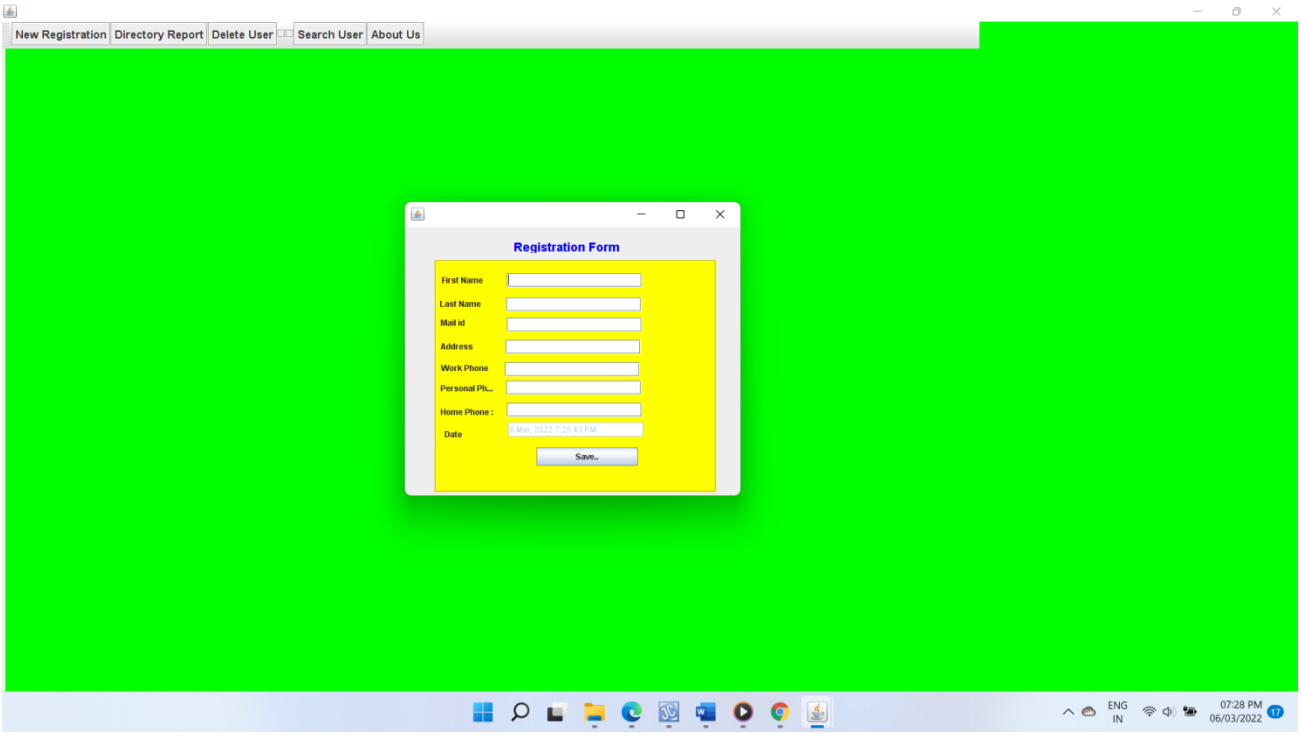
Process Flow Chart



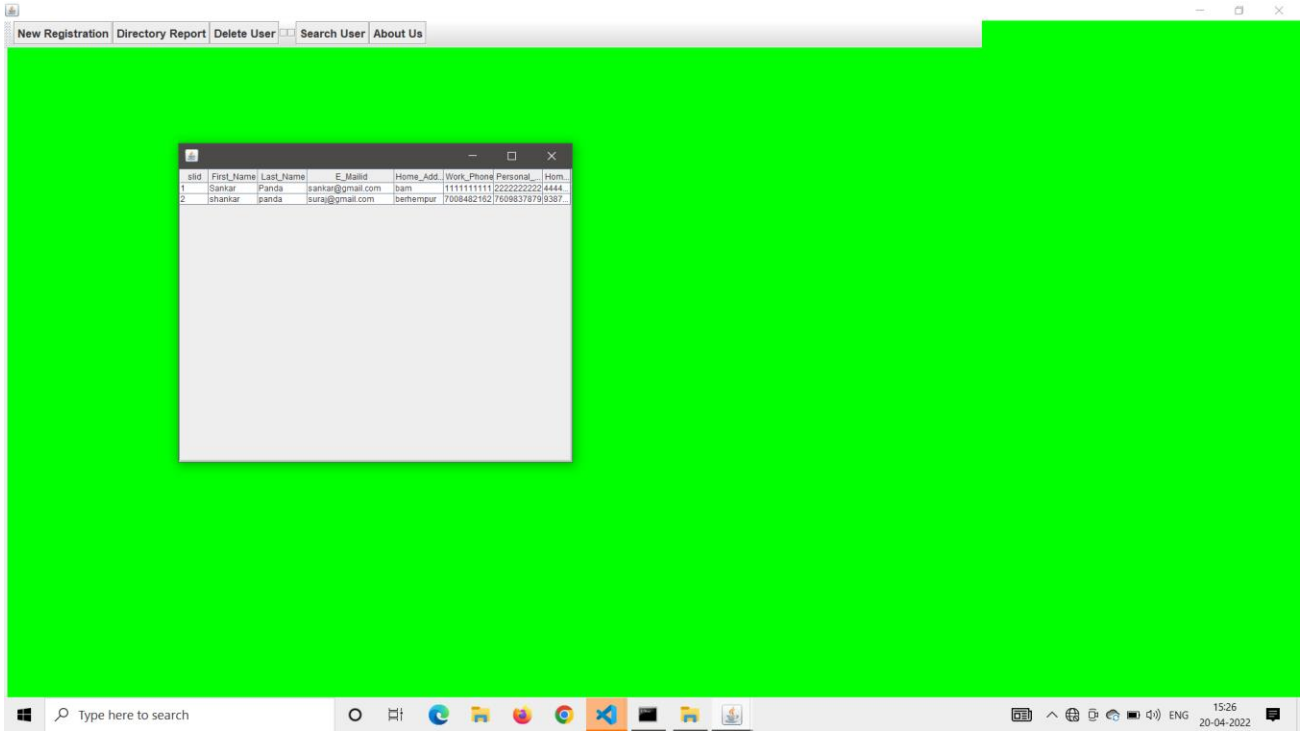
System Screen Shots


New Registration
Directory Report
Delete User
Search User
About Us
PROJECT NAME : Stack Based Telephone Directory
HOD :Raj kumar baliyar singh
UNDER GUIDENCE : Bibekananda pattnaiyak
NAME : Shankar narayan panda
NAME : Pankaj sahooo
NAME : Saaswat sritapragyan
Year : 2022
Type here to search
ENG
15:27
20-04-2022

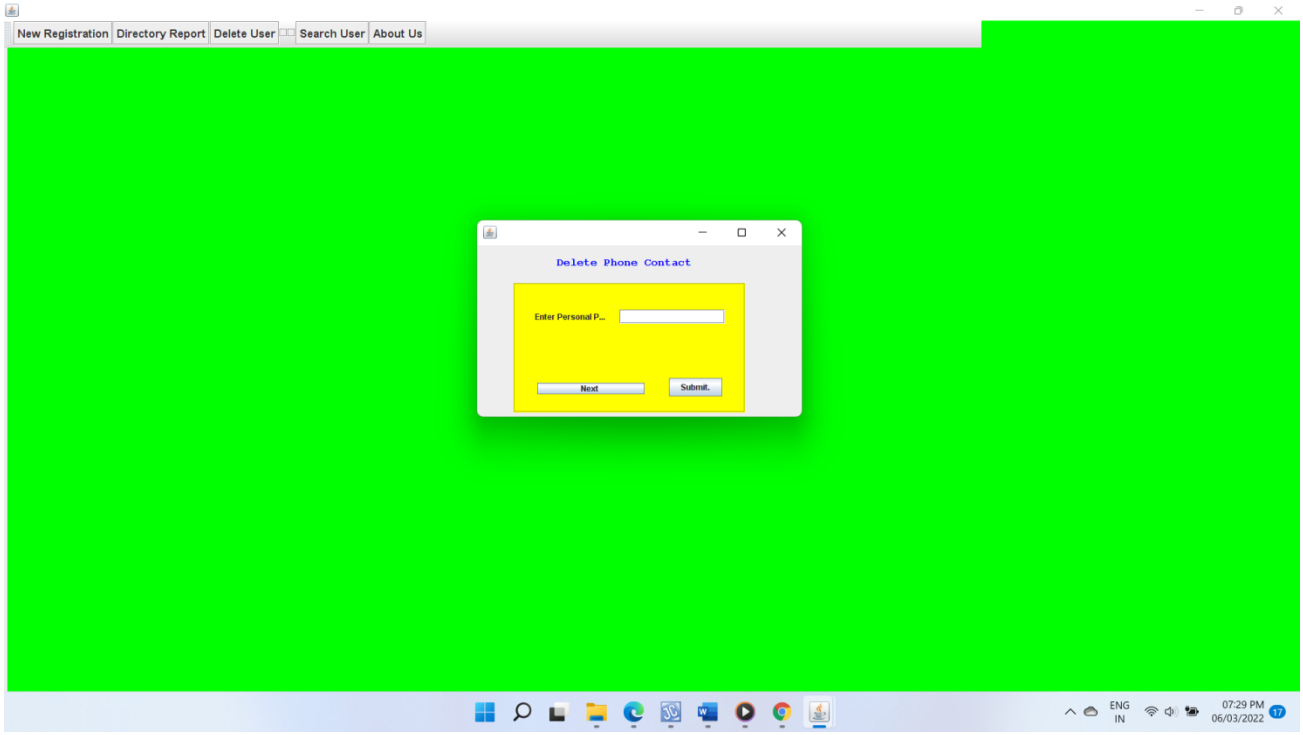
Registration Page



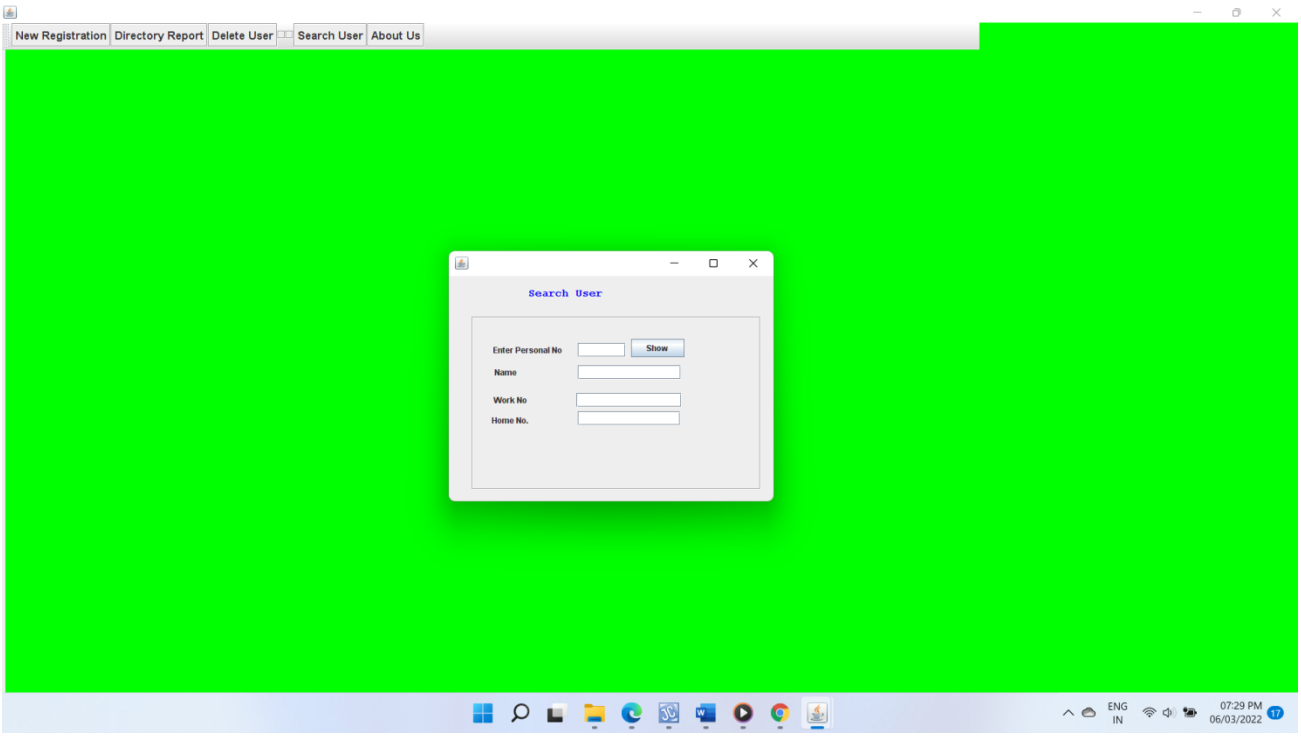
Report Page


New Registration
Directory Report
Delete User
Search User
About Us
slid
First_Name
Last_Name
E_Mailid
Home_Add..
Work_Phone
Personal_...
Hom...
1
Sankar
Panda
sankar@gmail.com
bam
1111111111
2222222222
4444...
2
shankar
panda
suraj@gmail.com
berhempur
7008482162
7609837879
9387...
Type here to search
ENG
15:26
20-04-2022

Delete contact page



Search User Page



Testing

TESTING

Volume Testing:

In this test we create as many records as would normally be produced to verify the hardware and the software that will function properly. The user is easily asked to provide test data for volume testing. In the system we develop a huge number of records are being tested and the test output shows that the system can hold an amount of data required by the firm.

Stress Testing:

The stress testing is to provide that does not malfunction under peak loads. Unlike volume testing, where time is not factor, we subject the system to a high volume of data over short time period. This simulates an –line environment where high volume of activities occurs spurts.

Recovery Testing:

A forced system failure is induced to test backup recovery procedure for file integration. Inaccurate data are to see how the system response in terms of error deduction and protection related to file integrity as a test demonstrate that data program are secure from unauthorized access.

Usability Documentation and Procedure:

The usability tests verify the friendly nature of the system. It also tests where an unknown user can handle the system freely or not. A crucial phase of the system life cycle is the successful implementation of the new system design. Implementation simply means covering the new system into operation .These involve creating computer compatible files, training operating staff and installing hardware, terminal and telecommunication network before the system is setup and running critical factor in conversion is not disrupting the functioning of the organization.

Testing of Individual Program

The individual programs are completed during the program development stage itself. Each program was tested in some test data at the time of coding a necessary changes in order to make sure that the programs working properly.

Creating test Data:

Though some test data during individual program development was not sufficient for testing the system as a whole. During the time testing all types of checking has been done depending upon situation.

System testing:

After successfully completion of the individual forms the whole system is run through a series of test to ensure the working of the system as a whole. The effects of testing the entire program are to verify that the program is working properly and according users specification that were made during the period of system studies.

User training:

User training is an important to be taken into account before implementation. Testing and implementation

Testing techniques:

White box testing

- Condition testing
- Dataflow testing

Black box testing

- Equivalence partitioning
- Boundary value analysis

Testing strategies

- Unit Testing
- Integration Testing
- Validation Testing

Testing intends the developed to discards preconceived notions of correctness of the software development and overcome the conflict of interest that occurs when error are uncovered. Testing is done in order to find the errors that are present in the system. The error occurs when the output of the software does not match with the expected output. In

der to detect the error and correct then different testing techniques and testing strategies have been used.

TESTING TECHNIQUES

The common testing techniques are

1. WHITE BOX TESTING
2. BLACK BOX TESTING

WHITE BOX TESTING

It is predicted upon the close examination of procedural detail. Logical paths through the software are tested by providing test cases that exercise specific set of condition or loops. It guarantees that

- All independent paths in the module have been executed at least once
- Execution of all logical decision on their both true and false side
- Execution of loops to their boundary values
- Exercise internal data structure to ensure their validity

CONDITIONAL TESTING:-

Conditional testing are done in order to exercise all logical condition.

DATAFLW TESTING:-

The data flow testing method select test paths of a program according to the location of definition and use of variables in the program.

LOO TESTING:-

Loop testing is a white box testing technique that focuses on the validity of the loop construction.

BLACK BOX TESTING:-

Black box testing is done in order to the detect the interface errors. It guarantees that the inputs are properly accepted and outputs are correctly produced and the integrity of the external information is maintained. It finds the following common errors.

1>Interface errors

2>Errors in data structure

3> Initialization and termination errors.

The difference Black box techniques are:

EQUIVALANCE PARTIONING: - It is the testing method that divides the input domain of the program into classes of data from which different test cases are designed.

BOUNDARY VALUE ANALYSIS:- A greater number of errors tends to occur at the boundary of the input domain. Therefore, boundary values analysis is required to derive the test that exercises boundary values.

TESTING STRTEGIES:- Various testing strategies have been proposed but only few of them have been used in this project>they are

UNIT TESTING: - It focuses verification effort on the smallest unit of the software design. Important control ate tested to uncover error within the boundary of the module.

INTEGRATION TESTING: - Integration testing is a systematic approach for constructing the program structure while at the same time conducting test to uncover error associated with interfacing.

There are two types of integration they are:

>TOP DOWN INTEGRATION

>BOTTOM UP INTEGRATION

VALIDATION TESTING:- After the culmination of integration testing, software is completely assembled as package, interfacing errors are uncovered and corrected then the validation testing begins.

SYSTEM TESTING:-

The systems are integrated to eventually from the entire system. After the system is put together system testing is performed to see of the entire requirements are met and if the systems performs as specified by the requirement.

For example, one module expected by another sub module. If field size of both sub modules are different then occurred at the time of system testing.

Future Scope AND ENHANCEMENT

- Users with average info about mobile or computers can make use of this system
- They can make use of different facilities offered by this software like they can store e-mail address, nicknames, B-Day of the contact along with phone number.
- Searching will be faster as compared to manual search.

Conclusion

- The application program has been successfully implemented using experimental cases .

- This application works for other functions that make it easy to search, delete, edit, and remember our peer information.

References

Java for Beginner (Macmilan)

Java Black Book (HerbeltSchildt)

www.google.com

www.codeproject.com