

**PROJECT REPORT**  
**ON**  
**MOBILINK CUSTOMER DATABASE**

Submitted in the partial fulfillment of the requirements for the award of the degree  
of

**Master of Computer Sciences**  
(2009-2011)

**Submitted by:**

**Mr. Haider Khalaf Jabbar**

**Enroll NO: 2009-525-012**

**Under of Supervision of:**

**Mr. Ihtiram Raza Khan**



**Department of Computer Science**  
**Faculty of Management Studies and Information Technology**  
**Jamia Hamdard, Delhi**



# Jamia Hamdard

(Hamdard University)

(Declared as Deemed-to-be University under Section 3 of  
the UGC Act, 1956 vide Notification No. F.9-18/85-U.3  
dated 10.5.1989 of the Government of India)


Phone : 2605 9688 (12 Lines)  
Extn. : 5858, 5899  
Fax : 00-91-11-26059663  
E-mail : anlam@jamiyahamdard.ac.in  
Website: www.jamiyahamdard.edu

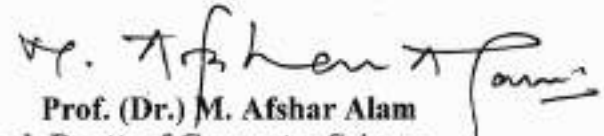
HAMDARD NAGAR  
NEW DELHI - 110062

DEPARTMENT OF COMPUTER SCIENCE  
Faculty of Management Studies & Information Technology

## CERTIFICATE

This is to certify that the Project work entitled "**MOBILINK CUSTOMER DATABASE**" being submitted by **Mr. Haider Khalaf Jabbar (2009-525-012)** in partial fulfillment of requirements for the award of the degree in **Master of Computer Science (M.Sc)** Jamia Hamdard (Hamdard University) New Delhi-110062 is a record of the students own work carried out under the supervision and guidance of **Mr. Ihtiram Raza Khan**.

  
Mr. Ihtiram Raza Khan.  
Project Supervisor

  
Prof. (Dr.) M. Afshar Alam  
Head, Deptt. of Computer Science



DEPTT. OF COMPUTER SCIENCE  
JAMIA HAMDARD (HAMDARD UNIVERSITY)  
NEW DELHI-110062 (INDIA)

## TABLE OF CONTENTS

S. No.	CONTENT	PAGE No.
1.	DECLARATION	3
2.	CERTIFICATE	4
3.	ACKNOWLEDGEMENT	6
4.	LIST OF TABLE	7
5.	LIST OF FIGURE	8
6.	LIST OF ABBREVIATION	9
7.	INTRODUCTION	10-12
8.	PROBLEM SELECTION	13-15
9.	PROJECT MONITORING SYSTEM	16-23
10.	SYSTEM STUDY	24-30
11.	SYSTEM ANALYSIS	31-33
12.	SYSTEM DESIGN ✓ Data Flow Diagram ✓ E-R Diagram	34-44
13.	SYSTEM TESTING & IMPLEMENTATION	45-50
14.	DOCUMENTATION	51-101
15.	CONCLUSION	102-105
16.	SCOPE OF PROJECT	106-107
17.	REFERENCES	108

Table 1.1

### **CERTIFICATE**

This is to Certify that MR. HAIDER KHALAF JABBAR Enroll NO:2009-525-012 has worked under my supervision to prepare his project report entitled MOBILINK CUSTOMER DATABASE PROJECT .The work embodied in this report is original and was conducted at Sciences Department of Computer Science Jamia Hamdard University From 10 February to 25 April. This work has not been submitted in part or full to this or any other university for the award of any other degree or diploma.

**Signature of Supervisor:**

**Name:**

**Designation:**

## **DECLARATION**

I hereby declare that the PROJECT work entitled MOBILINK CUSTOMER DATABASE PROJECT Submitted for the MSC VI sem(2009-2011) is my original work carried out by me under the guidance of **Mr. Ehtiram Raza Khan** for the partial fulfillment of the award of the degree of the Masters of Computer science. The matter embodied in this report has not submitted anywhere else for the award of any other degree/diploma.

**Place: Department of Computer Science**

**Date:30 April 2011**

**Signature of the Candidate:**

## ACKNOWLEDGEMENT

The project work mentioned in this report, is the result of cumulative efforts over a period of an entire year in course of which, I have received intellectual support from various sources. It is my pleasure to express my profound sense of gratitude to all those who have contributed richly to this project and have been highly instrumental in making this a success.

our special thanks to Head Of The Department of computer science

**Prof. M. AFSHAR ALAM**

I am highly indebted to my teacher **Mr. Ehtiram Raza Khan** for giving me an opportunity to work on this project. He has been an invaluable source of guidance and motivation at all stages of my work.

**NAME: HAIDER KHALAF JABBAR**  
**ENROLL NO: 2009-525-012**

### List of Table

SNo.	Table no.	Description of Table	Page no.
1.	5.1	List of figure	8
2.	6.1	List of abbreviations	9
3.	12.1	Customer table	39

### List of Figures

SNo.	Table no.	Description of Table	Page no.
1.	12.1	0 level DFD	35
2	12.2	1 level DFD	36
3.	12.3	2 level DFD	37
4.	12.4	ER DIAGRAM	47

Table 5.1



### List of Abbreviations

Sno.	Abbreviations	Word
1	Emp	Employee
2	Rec	Records
3	RA	Requirement Analysis
4	Ackn	Acknowledgment
5	SSADM	Structured System Analysis Methodology
6	Prg	Programs
7	WF	Waterfall Model
8	PS	Preliminary Survey
9	H/w	Hardware
10	S/w	Software
11	RS	Requirement Specification
12	Transaction	Trans
13	SA	System Analysis
14	DFD	Data Flow Diagram
15	ERD	Entity Relationship Diagram
16	DD	Data Dictionary
17	UT	Unit Testing
18	ST	System Testing

Table 6.1

# MOBILINK CUSTOMER DATABASE

## **Abstract**

Before the introduction of computers in office, the office work was done manually by keeping the information in registers. It is also requires searching of the registers for particular information. searching the information manually is a time consuming process. In a computerized system, the user can do the things in a much easier way as compared to the manual system of keeping the records. for that I designed this new program that will be do all the job automatically But it need first to input the main data depend on the programming way and that will save us from the manually way. And it be have high level of security .

## **Aim of the project**

The aim of the project is to deliver a working system that will allow the higher level management to keep the track records of the customer.

## **Objective of the project**

In a computerized system, the user can do things in a much easier way as compared to the manual system of keeping the records. This software is made to computerized the working of office automation or office management software for keeping the track records of employees computerized. to produce a simple, yet suitably effective user friendly interface, and a system that is easily maintainable, efficient and practical and keep the records in well manner & stores it.

## **Application Area**

The project undertaken is actually to develop an Office Management System for the higher level authority. The application will be in C++ & the concept of files has been included to save the records. so, the application area of the project will include all the tasks involving the items of the employees.

## **Problem selection**

The current system is not organized. It is a system based on more of common sense. The main idea behind this system is "as it has been going on fine". The speed of the system depends on human speed and mood.

## **Problem Definition**

Present system is based more on manual record keeping methodology, which makes the overall working of the system very slow, time consuming and thus it is proved to be inefficient. hence software was required which could make the entire system fast and efficient the conventional system in the office is manual where all the work is done manually by entering the records in the various files and folders.

It is quite cumbersome to proceed through the manual system because, maintaining the transactions regular employees takes ample time. Maintaining these files requires lot of space as

well, since it is totally depending on human, efficiency is low. There is less security as all the information is basically kept as hard copy in a register. The computer back up is also in the file system.

#### **IDENTIFICATION OF NEED**

This step is initiation of System Analysis. An overview of the client's requirement has been done. The basic need of the client to opt for such kind of project is analyzed. as per current marketing scenario, an entire system was required to track day-to-day transactions, and timely generation of reports is the basic features of this project.

Client was following a Manual Process, which is not at all compatible with its current working conditions. It was not only time consuming, but also lacks accuracy. Security point of view the manual system was failed to hide the information from any unauthenticated staff or any outside person. Therefore, there was an urgent requirement of such Computerized System, which can fulfill all of its current as well as future requirements.

## **INTRODUCTION**

## **INTRODUCTION**

Before the introduction of computers in office, the office work was done manually by keeping the information in registers. It is also requires searching of the registers for particular information.

Searching the information manually is a time consuming process. In a computerized system, the user can do the things in a much easier way as compared to the manual system of keeping the records.

- AIM
- OBJECTIVE
- APPLICATION AREAS

### **Aim of the project**

The aim of the project is to deliver a working system that will allow the higher level management to keep the track records of the customer.

### **Objective of the project**

In a computerized system, the user can do things in a much easier way as compared to the manual system of keeping the records. This software is made to computerized the working of office automation or office management software for keeping the track records of employees computerized.

To produce a simple, yet suitably effective user friendly interface, and a system that is easily maintainable, efficient and practical and keep the records in well manner & stores it.



### **Application Area**

The project undertaken is actually to develop an Office Management System for the higher level authority. The application will be in C++ & the concept of files has been included to save the records.

So, the application area of the project will include all the tasks involving the items of the employees.

### **About Office Management**

Mobile Database was founded by Mr. Narayan in 2001 with support of his wife Ms. Swati Narayan. The first office was started in Bangalore followed by 6 outlets in various parts of the country till now.

Mr. Narayan is a postgraduate in M.SC from Pune while Ms.Narayan is a graduate in Business Administration. Both started their career from a small a office outlet in the Delhi.

Today Narayan and corporate acquires a reputed position in Delhi. The organization offers quality Structure for the employees. Initially the office track records system was carried out manually by the admin who made it tedious and time consuming. Also the employees demanded for a registered office management

Keeping in mind all the pitfalls of the manual record keeping system, a computerized system was demanded for all the outlets.

## **PROBLEM SELECTION**

## **PROBLEM SELECTION**

### **Problem selection**

The current system is not organized. It is a system based on more of common sense. The main idea behind this system is "as it has been going on fine". The speed of the system depends on human speed and mood.

### **2.1 Problem Definition**

Present system is based more on manual record keeping methodology, which makes the overall working of the system very slow, time consuming and thus it is proved to be inefficient.

Hence software was required which could make the entire system fast and efficient

The conventional system in the office is manual where all the work is done manually by entering the records in the various files and folders.

It is quite cumbersome to proceed through the manual system because, maintaining the transactions regular employees takes ample time. Maintaining these files requires lot of space as well.

Since it is totally depending on human, efficiency is low. There is less security as all the information is basically kept as hard copy in a register. The computer back up is also in the file system.



## **2.2 IDENTIFICATION OF NEED**

This step is initiation of System Analysis. An overview of the client's requirement has been done. The basic need of the client to opt for such kind of project is analyzed.

As per current marketing scenario, an entire system was required to track day-to-day transactions, and timely generation of reports is the basic features of this project.

Client was following a Manual Process, which is not at all compatible with its current working conditions. It was not only time consuming, but also lacks accuracy. Security point of view the manual system was failed to hide the information from any unauthenticated staff or any outside person. Therefore, there was an urgent requirement of such Computerized System, which can fulfill all of its current as well as future requirements.

## **PROJECT MONITORING SYSTEM**

## PROJECT MONITORING SYSTEM

C++ is a general-purpose, high-level programming language with low-level facilities. It is a statically typed free-form multi-paradigm language supporting procedural programming, data abstraction, object-oriented programming, generic programming and Runtime Type Identification (RTTI). Since the 1990s, C++ has been one of the most popular commercial programming languages.

Bjarne Stroustrup developed C++ (originally named "*C with Classes*") in 1983 at Bell Labs as an enhancement to the C programming language. Enhancements started with the addition of classes, followed by, among other features, virtual functions, operator overloading, multiple inheritance, templates, and exception handling. The C++ programming language standard was ratified in 1998 as *ISO/IEC 14882:1998*, the current version of which is the 2003 version, *ISO/IEC 14882:2003*.

### History

Stroustrup began work on *C with Classes* in 1979. The idea of creating a new language originated from Stroustrup's experience in programming for his Ph.D. thesis. Stroustrup found that Simula had features that were very helpful for large software development, but the language was too slow for practical use, while BCPL was fast but too low-level and unsuitable for large software development. When Stroustrup started working in Bell Labs, he had the problem of analyzing the UNIX kernel with respect to distributed computing. Remembering his Ph.D. experience, Stroustrup set out to enhance the C language with Simula-like features. C was chosen because it is general-purpose, fast, and portable. Besides C and simula, some other languages which inspired him were ALGOL 68, Ada, CLU and ML. At first, the class, derived class, strong type checking, inlining, and default argument features were added to C via Cfront. The first commercial release occurred in October 1985.

In 1983, the name of the language was changed from C with Classes to C++. New features were added including virtual functions, function name and operator overloading, references, constants, user-controlled free-store memory control, improved type checking, and a new single-line comment style with two forward slashes (*//*). In 1985, the first edition of *The C++ Programming Language* was released, providing an important reference to the language, as there was not yet an official standard. In 1989, Release 2.0 of C++ was released. New features included multiple inheritance, abstract classes, static member functions, const member functions, and protected members. In 1990, *The Annotated C++ Reference Manual* was published. This work became the basis for the future standard. Late addition of features included templates, exceptions, namespaces, new casts, and a Boolean type.

As the C++ language evolved, a standard library also evolved with it. The first addition to the C++ standard library was the stream I/O library which provided facilities to replace the traditional C functions such as `printf` and `scanf`. Later, among the most significant additions to the standard library, was the Standard Template Library.

After years of work, a joint ANSI-ISO committee standardized C++ in 1998 (*ISO/IEC 14882:1998*). For some years after the official release of the standard in 1998, the committee processed defect reports, and published a corrected version of the C++ standard in 2003. In 2005, a technical report, called the "Library Technical Report 1" (often known as TR1 for short) was released. While not an official part of the standard, it gives a number of extensions to the standard library which are expected to be included in the next version of C++. Support for TR1 is growing in almost all currently maintained C++ compilers.

While the C++ language is royalty-free, the standard document itself is not freely available

### Features introduced in C++

Compared to the C language, C++ introduced extra features, including declarations as statement, function-like casts, new/ delete, reference type, inline functions, default arguments, function overloading, namespaces, classes( including all class-related features such as inheritance, member function, virtual function, abstract classes, and constructors), operator overloading, templates, the :: operator, exception handling, and runtime type identification.

### **Platform**

The platform I used to develop my project "Office Management System" is in C++. During the development of this project I had several options of the platform to use. After thorough study and discussion with my faculty and guide, I decided to use OOPS concept using C++ language. It acts as front-end as well as back-end. Back – end deals with the file-handling concept of

The C++

It is very easy and reliable platform as it is based on real world programming. It has very exciting features that are very similar to our world and life.

These features are:-

- Ø Identity
- Ø Classification
- Ø Encapsulation
- Ø Data Abstraction
- Ø Polymorphism
- Ø Inheritance

All these features of C++ can run on any hardware or system that has 'Windows XP Professional' Operating System. It has many features and tools that will make our computer easy, effective and entertaining. For example, we can use remote desktop to



access our work computer and its resources from home, as well as view fields and document on our computer's desktop from a co-worker's computer.

I developed my project in the OOPS environment using C++ language. OOPS stands for "Object Oriented Programming System". It is a way of design software in which software is discrete collection of objects. As C++ is a structural programming, It is based on modularization.

Object-oriented programming (OOP) is the most dramatic innovation in software development in the last decade. It ranks in importance with the development of the first higher-level languages at the dawn of the computer age. Sooner or later, every programmer will be affected by the object-oriented approach to program design.

### **Advantages of OPP**

Why is everyone so excited about OOP? The chief problem with computer programs is complexity. Large programs are probably the most complicated entities ever created by humans. Because of this complexity, programs are prone to error, and software errors can be expensive and even life threatening (in air-traffic control, for example). Object-oriented programming offers a new and powerful way to cope with this complexity. Its goal is clearer, more reliable, more easily maintained programs.

Languages and development platform Of the object-oriented programming languages, C++ is by far the most widely used.

In past years the standards for C++ have been in a state of evolution. This meant that each compiler vendor handled certain details differently. However, in November 1997, the ANSI/ISO C++ standards committee approved the final draft of what is now known as Standard C++. (ANSI stands for American National Standards

Institute, and ISO stands for International Standards Institute). Standard C++ adds many new features to the language, such as the Standard Template Library (STL).

The most popular development environments for C++ are manufactured by Microsoft and Borland and run on the various flavors of Microsoft Windows

OOP involves concepts that are new to programmers of traditional languages such as Pascal, Basic, and C. These ideas, such as classes, inheritance, and polymorphism, lie at the heart of object-oriented programming.

### **Software and Hardware**

This project requires the latest version of either the Microsoft or the Borland C++ compiler.

Both products come in low priced. The computer should have enough processor speed, memory, and hard disk space to run the compiler we've chosen. We can check the manufacturer's specification to determine these requirements

The fundamental idea behind object-oriented languages is to combine into a single unit both data and the function that operate on that data. Such a unit is called an object.

An object's functions, called member function in C++, typically provide the only way to access its data. If we want to read a data item in an object, we call a member function in the object. It will access the data and return the value to us. We can't access the data directly. The data is hidden, so it is safe from accidental alteration. 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 we want to modify the data in an object, we know exactly what function 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 C++ program typically consists of a number of objects, which communicate with each other by calling one another's member functions

We should mention that what are called member functions in C++ are called methods in some other object-oriented (OO) languages (such as Smalltalk, one of the first OO languages). Also, data items are referred to as attributes or instance variables. Calling an object's member function is referred to as sending a message to the object. These terms are not official C++ terminology, but they are used with increasing frequency, especially in object-oriented design

The database in the C++ is created using the stream classes. A stream is a general name given to a flow of data. In C++ a stream is represented by an object of a particular class. So far we've used the cin and cout stream objects. Different streams are used to represent different kinds of data flow. For example, the ifstream class represents data flow from input disk files

C++ programmers may wonder what advantages there are to using the stream classes for I/O, instead of traditional C functions such as printf( ) and scanf( ), and for files fprintf( ), fscanf( ), and so on.

One reason is simplicity. If we've ever used a %d formatting character when we should have used a %f in printf( ), you'll appreciate this. There are no such formatting characters in streams, since each object already knows how to display itself. This removes a major source of errors.

Another reason is that you can overload existing operators and functions, such as the insertion (<<) and extraction (>>) operators, to work with classes that you create. This makes our own classes work in the same way as the built-in types, which again makes programming easier and more error free.

Streams I/O are important if we plan to program in an environment with a Graphics User Interface such as Windows, where direct text output to the screen is not used. We still need to know about C++ stream because they are the best way to write data files, and also to format data in memory for later use in text input/output windows and other GUI elements





# **SYSTEM STUDY**

## SYSTEM STUDY

System Study is to measure that how beneficial or practical the development of an information system will be to an organization. It is also known as the Feasibility study.

The project shall be feasible, given unlimited resources and infinite time. The objective of the feasibility is not to solve the problem, but to acquire a sense of its scope. It is always necessary to carry out a feasibility study for the development of a new product system from the following viewpoints:-

- What shall be the problems with the conventional system?
- What shall be the solution available?
- What shall be the goals and sub goals of the proposed system?
- What will the proposed system achieve?
- Who all shall be involved with the operations of the system?
- What benefit will the system give of the conventional system?
- What will be the estimated cost of the implementation?

Feasibility analysis is a cross life cycle activity, which has to be continuously performed throughout the system development. By using the creeping commitment approach, feasibility is measured at different times in the system development. This evaluation ensures that the project is beneficial and practical to an organization.

Feasibility Test: There are three categories of feasibility tests:

1. Operational Feasibility
2. Technical Feasibility
3. Economic Feasibility.

**OPERATIONAL FEASIBILITY;** - is how well the solution will work in the organization and how the end-users and managers feel about the system. This people oriented test measures the urgency of problem or the acceptability of a solution. i.e. Is the problem worth solving?

PIECE is used as the basis for analyzing the urgency of problem or the Effectiveness of a solution.

**PIECE stands for:**

**P – Performance**

**I – Information**

**E – Economics**

**C – Control**

**E – Efficiency**

**TECHNICAL FEASIBILITY** centers on the existing computer system and to what extent it can support the proposed system.

**ECONOMICAL FEASIBILITY** includes the cost – benefit analysis. That is to determine whether the procedure to be applied is benefits and savings are expected from the candidate system.

After gathering all the details of the existing system a lot of problems come to light. The user stated problems are as follows:

1. Difficulty in finding any information about the food items at any instance.
2. Lack of appropriate report to submit the management on demand due to large manual processing done on various relevant registers.
3. Difficulty in calculation of total bill.
4. Long processing taken at the peak hours. As a result poor performance of the system causes dissatisfaction.
5. Difficulty in updating of the items purchased.

### **SOFTWARE PROCESS MODEL**

There are various orders in which the six phases of the Structured Systems Analysis Design Methodology (SSADM) are implemented. These are what we know as software process models.

A software process model is a development strategy we incorporate to describe how a set of activities (the six phases of SSADM) must be performed and in which order to get the desired output. We already have a few process models defined and described.

They are:

- 1.1. Code and Fix Model
- 1.2. Waterfall Model
- 1.3. Evolutionary Model
- 1.4. Transformation Model
- 1.5. Spiral Model

To select the best model for the project all the above mentioned models is to be evaluated. After evaluation the following assessment is realized.

The **Code and Fix Model** can actually be considered no model at all as there is nothing rigid and the developer has to follow his inspirations.

The **Evolutionary model** is increment driven and a kind of trial and error method.

The **Transformation model** can be called specification driven, as the development process occurs through iterative refinement of formal specifications.

**Spiral model** is a Meta model that may be called risk driven.

Since, there is time constraint, trial and error and risk driven models are eliminated. The project must have documentation and this feature is facilitated only by the Waterfall model.

The model is briefly described now which will facilitate the reader to follow the exact path I am going to follow in the development of the system.

## **REQUIREMENT ANALYSIS**

- OBJECTIVE
- PRELIMINARY SURVEY
- REQUIREMENT SPECIFICATIONS
- PERFORMANCE CRITERIA
- SYSTEM HARDWARE AND SOFTWARE REQUIRED

## **Objective**

The problem forwarded by the client is actually to computerize their already existing manual Record keeping System or employees.

## **Preliminary Survey**

The client already had a set system which was manual. This manual system was although efficient, but very time consuming and tedious. Also, the different modules were not fully integrated and needed an absolute understanding of the whole system to give the desired Results. The main problems found out were bottlenecks at the user end because of slow data flow rate in a manual system. So, a preliminary survey was done check whether any simple modification could actually stimulate the system.

The present system seemed to be perfect if only data flow could be automated. The same modules could be implemented in the software.

## **Requirement Specifications**

After the preliminary survey and listening to the client's requirements following requirements were listed:

- To develop a **Computerized Record Keeping** of the employees.
- The present manual system being employed should be selected as the client system.
- The system should ensure that a quick response is given to user queries in areas of requests.
- A simple user interface should provide to the user to easy access to requested information.
- The front-end should have graphical user interface.
- Every transaction should be dealt with in minimum delay.



### **Performance Criteria**

After the initial investigation of the present system and the client's requirements some performance criteria were deducted. These criteria cover both the software and the hardware aspect of the project.

### **System Hardware and Software Required**

After knowing the performance criteria, it became fairly easy to decide on the hardware and software platform.

The next step was to decide which software to use in order to develop and support the system. Microsoft Windows XP professional was a fair choice. Then, to choose the language various alternatives were checked. Programming languages such as C or C++ could be used, but as system was graphical interface, C++ was the inevitable choice for that and for back end concept of **FILES** has been used is the best.

Software Required	:	Turbo C
RAM	:	Maximum 256 Mb
Operating Platform	:	Windows xp Professionals
Hard Disk	:	Minimum 20 GB
Processor	:	Pentium Series



# **SYSTEM ANALYSIS**

## SYSTEM ANALYSIS

- OVERALL STRATEGY
- PROBLEMS WITH THE CURRENT SYSTEM
- ADVANTAGES OF COMPUTERIZATION
- DATA FLOW ANALYSIS OF THE CURRENT SYSTEM
- DATA FLOW DIAGRAM
- DATA DICTIONARY

### System Analysis

The goal of the analysis stage is to build and understand the scenario involved and to create a description of just what is desired and what will eventually be built. To understand the problem we must have input to get output. This can take the form of interviews, specifications regarding to level of performance and random data.

In order to have a structured approach to the analysis stage I had to choose a methodology to follow to have a complete analysis stage. The analysis stage is probably the most important as a mistake or missed requirement from analysis phase may cost much more time and money to fix later than if it had been caught in the analysis phase. Therefore it is imperative that at the analysis stage the best job possible is made.

This section outlines the strategies followed in organizing the analysis phase.

To understand the current processes for the MOBILINK CUSTOMER DATABASE PROJECT, I developed the following strategy:

Sample all documents and records of information.

Observe actual procedures followed.

### **Problems with the Current System**

The current system is not organized. It is a system based on more of common sense. The main idea behind this system is “as it has been going on fine”.

The speed of the system depends on human speed and mood.

Since it is totally depending on human, efficiency is low.

There is less security as all the information is basically kept as hard copy in a register. The computer back up is also in the file system.

### **Advantages of Computerization**

Upon computerizing the present system, it will become an organized one. Efficiency will increase and so will data flow speed.

Another advantage of computerization is security. Since main login process will have a password, unauthorized person will not be able to access any information.

# **SYSTEM DESIGN**

## **DESIGN ANALYSIS**

After understanding the business functions, purpose of the database, the performance needs of the application and the ease of maintenance, the designing part begins. A good design is the foundation to a correct, easily expandable and maintainable system. A maintainable system is one that is flexible and is simple to modify, if the system cannot adapt easily to the changing business environment, exceedingly efforts will be required to keep it serviceable. Design is a procedure for creating the blueprints for a system like the architects draft for the constructor to create the building.

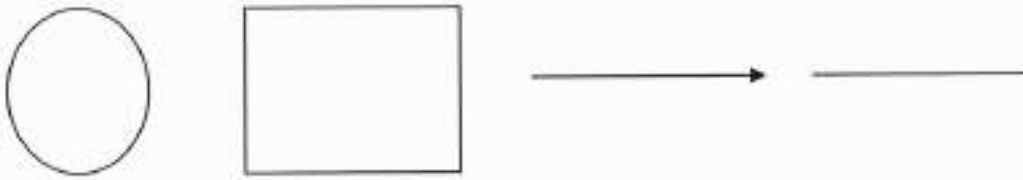
### **Data Flow Analysis For The Current System:**

After completion of the deep study of the current system, I am now ready to do the data flow analysis of the client.

This Data Flow Analysis consists of the Data Flow Diagrams (DFDs) which describe processes and data movement in the system and Data Dictionary (DD), which defines the systems' elements. Both provide useful information for designing the candidate system.

### **Data flow diagram (DFD)**

DFD or Data Flow Graphs are the cornerstone for structured systems analysis and design. It views the system as a collection of functions that transforms the inputs as desired outputs but does not go into the details of the transformation. This process modeling tool depicts flow of data through a system and processing work performed by the system. It has diagram of processes, data stores, data stores view, entities and data flows.



**Process:** Processes show what the system does. Each process has one or more data inputs and produces one or more data outputs. Processes are represented by circles in DFD.

**Data Stores:** A file or data store is repository of data. They contain data that is retained in the system. Processing can enter data into a data store or retrieve data from the data store. Each data store is represented by a thin line in the DFD and each data store has unique name.

**External Entities:** External entities are outside the system but they either supply input data into the system or they use the system output. The designer has no control over the external entities. External entities are represented by a square or rectangle.

Although data can be organized in several ways:

- stored in data repositories
- flow in data flows

Transferred to and from the external environment.

DFD is preferred here as it can be expressed by means of an attractive graphical notation that makes them easy to use and understand.

The four entities that must be represented are:

Data flows - movement of data in the system

Data stores - data repositories for data that is not moving

Processes - transforms of incoming data flow(s) to outgoing data flow(s) .

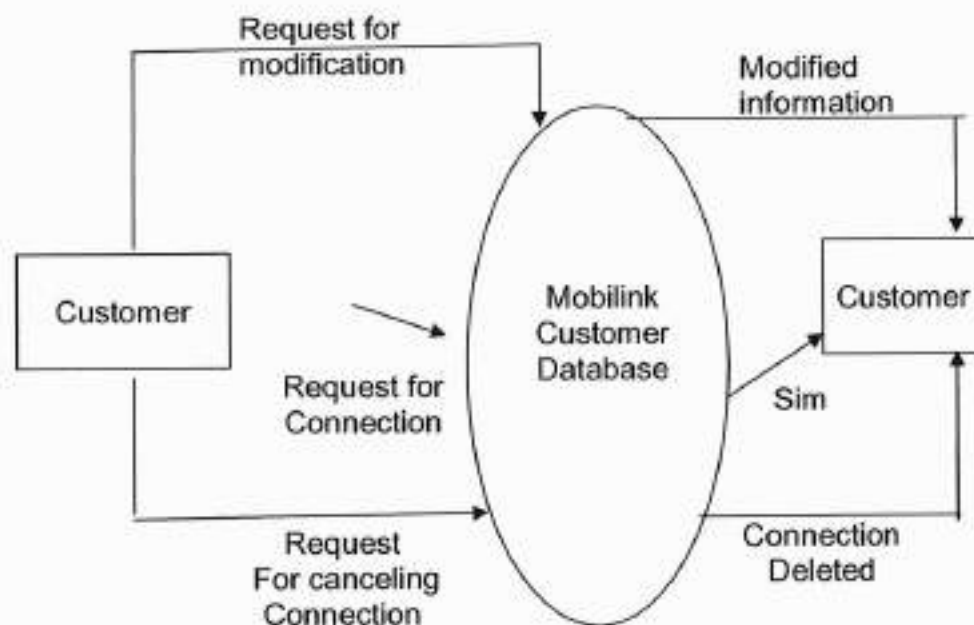
External entities - sources or destinations outside the specified system boundary

Following are the basic elements of a DFD which are used in the DFD of the project:

The function symbol (bubble)

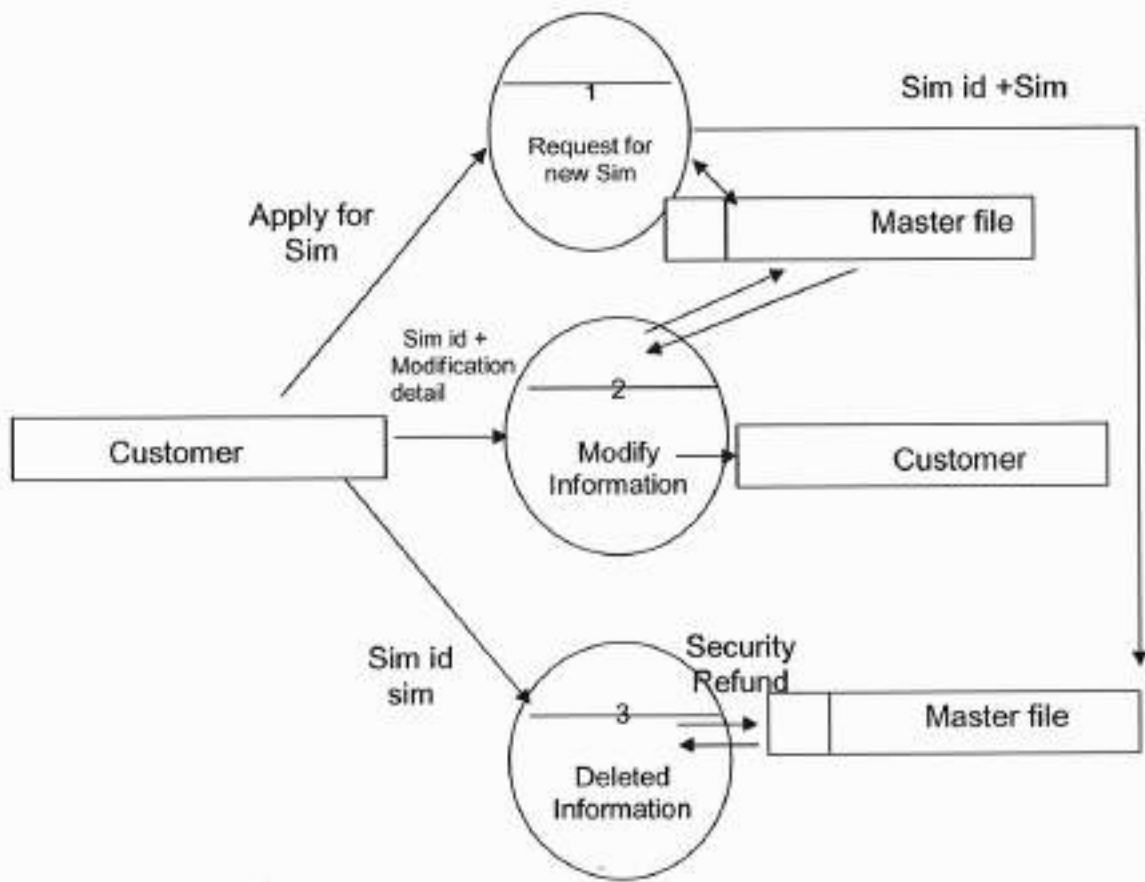
The DFDs for the current system is given here:

**O level DFD**

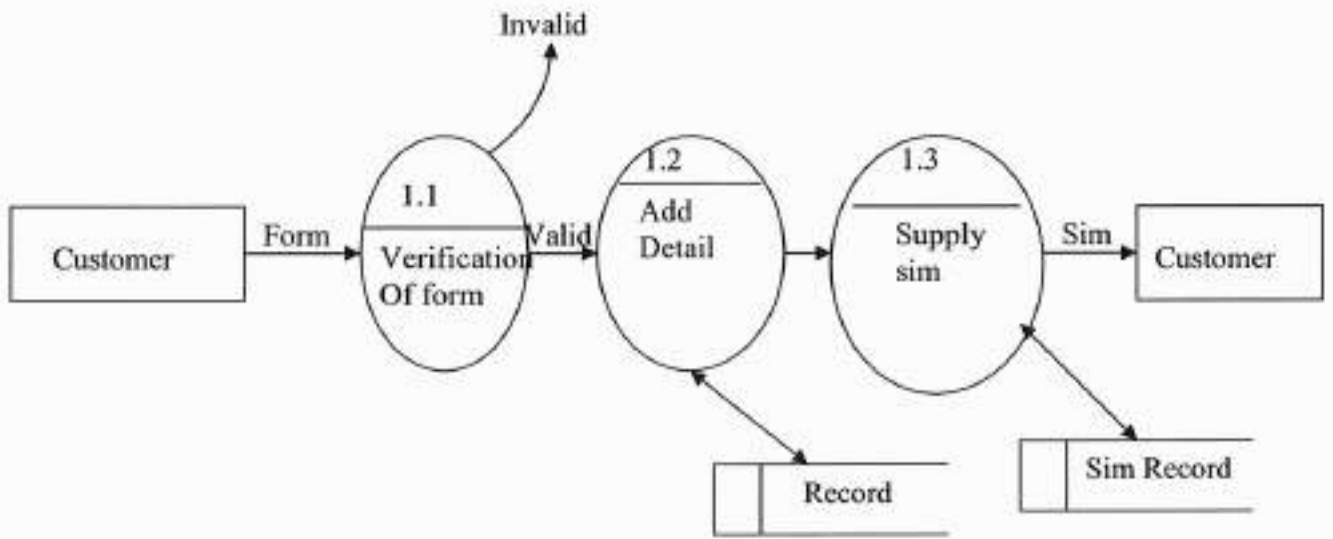


**Figure: 1.1**

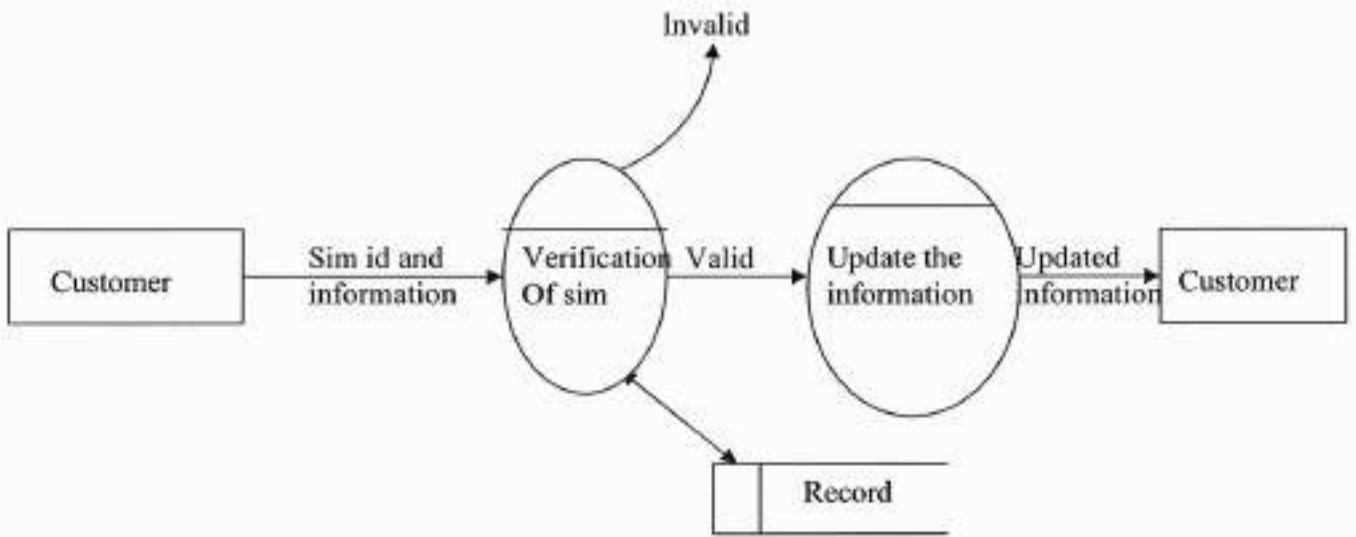




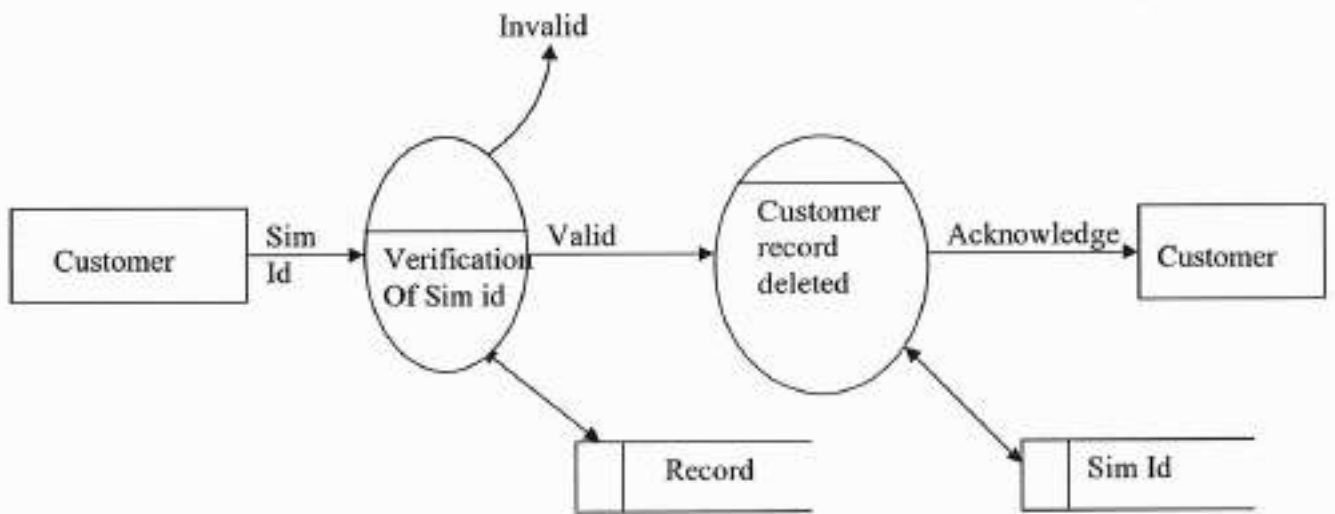




**Level 2 DFD Of DFD 1(Fig 12.1)**



**Level 2 DFD Of DFD 2(Fig 12.2)**



Level 2 DFD Of DFD 3(Fig 12.2)

## **CUSTOMER**

### **Data Dictionary**

Simply putting data dictionary (DD) is data about data. It is a

Centralized collection of definitions of all data flowing among functions and to or from data stores. Data dictionary removes redundancy and inconsistencies. The Data Dictionary for the current system is given below:

The proposed Inventory Management System is designed in the next stage following the DFDs and the DD.

<b>Attribute</b>	<b>Data Type</b>
Id	Int
Name	Char
Address	Int
Phone No	Int
Connection	Char

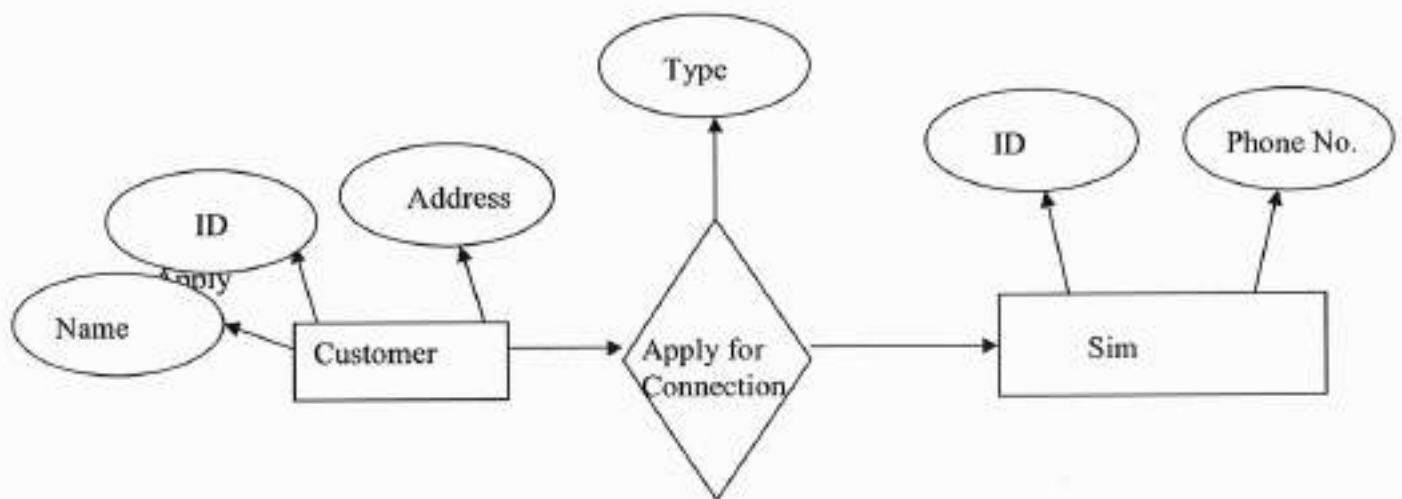
**Table 3.1**

- ✓ DESIGN ANALYSIS
- ✓ ENTITY RELATIONSHIP DIAGRAM

### Entity Relationship Diagram

Entities are related by relationships between common information, where relationship at any point in time is an association between two entities. The ER diagram is a technique that is useful in the designing phase to represent this logical part of the database. Since the entities had been identified in the analysis stage the only process left to build an ER diagram is to represent the relations.

Designing the ER diagram took several drafts before the final version was complete.



Entity Relationship Diagram

# **SYSTEM TESTING**

## TESTING AND IMPLEMENTATION

- TESTING
- UNIT TESTING
- SYSTEM TESTING
- TEST CONCLUSION

### **Testing**

The most crucial stage of software development, testing validates the application. During testing we will be concerned about the inputs and their expected outputs. We emphasize on the testing where we will input the data and compare the output with the expected results. At this stage we are not concerned about the process; we are only looking for correct outputs. Various software testing techniques exist which take different approaches to test and validate a software.

Tests done on the designed software was to verify the following properties of the software:

- Correctness(satisfaction of the specifications)
- Reliability(how well it meets the requirements)
- Portability (running in different environments)
- Usability (ease with which user can use the software)
- Maintainability (modifications after initial release), etc.

Some of them, which we used to test the software, are:

- **Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design - The software component or module. Using component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. Tests of data flow across a module interface were carried out for each module of the software. Errors in certain modules were detected and later rectified, thus correcting the dataflow path.

- **Integration Testing**

Once all the modules have been unit tested, they are interfaced together and passed through integration testing using top down integration approach.

- **Black Box Testing**

This approach enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. This approach is applied during the later stages of testing and attention is focused on matching the outputs with the expected result.

- **Debugging**

Debugging is removing the undesirable errors or bugs from the program. We implemented debugging using the C++ compiler compatible with the Windows system in which the application was developed.

During testing the program to tested is executed with the set of test cases and have the output of the program for the test cases is evaluated to determine if the program is performing as expected. Due to its approach dynamic if the program is performing as expected. Due to its approach dynamic testing can only presence of errors in the program, the exact nature of errors is not usually decided by testing. Testing forms is the process to determine errors in the program.



Once a program are tested individually then the system as a whole needs to be tested. During testing the system is used experimentally to ensure that the software does not fail i.e. it will run according to its specification. The programs executed to check for any syntax and logical errors. The errors are corrected and test is made to determine whether the program is doing what supposed to do.

**The cautions kept in mind while choosing the tests are:**

1. The testing should be based on sound and systematic techniques.
2. Testing should help locate errors, not just detect their presence.
3. Testing should be repeatable, i.e., repeating the same experiment, supplying the same input data to the same piece of code, produces the same results.

# **IMPLEMENTATION**

## IMPLEMENTAION

Once the software is delivered and developed, it enters the maintenance phase. After implementation systems need maintenance. Beyond monkey testing during Software development some errors may not appear. During its usage by the end-user with actual data certain errors may disclose. Therefore some residual errors or bugs remain in the system that must be removed as they are discovered. Many of these surfaces only after the system have been in operation sometimes for a long time. These errors once discovered need to be removed on an urgent basis for the smooth running of the system, leading to the software getting changed. Though Maintenance is not a part of software development, it is an extremely important activity in the life of a software product.

Maintenance involves understanding the existing software (code and related documents), understanding the effects of change, making the changes-to both the code and documents-testing the new parts and retesting the old part. For successful and smooth running of the system, maintenance is the prominent part of the project. Any error, which hinders the functioning of any part of the project, may lead to bad impression of the developer.

There are mainly two types of errors: Compilation error and Runtime errors.

1. Compilation errors are errors during coding and are to be taken care by the developer during development process.
2. Runtime errors are those which occur during running of the program.

**DOCUMENTATION**  
**CODING**

## CODING AND OUTPUT

```
#include <dos.h>

#include <stdio.h>

#include <ctype.h>

#include <conio.h>

#include <stdlib.h>

#include <string.h>

#include<graphics.h>

/*MACROS ARE DEFINED*/

#define NUM 5

#define PH 12

#define MIN 30

#define MAX 61

/*FUNCTION DECLARATION*/

void add();      // Ask user for input.

void view();    // Diplay all the record.

void search();  // Search for a record.

void c_name();  // Changes the name in a record.

void c_add();   // Changes the address in a record.

void c_ct();    // Changes the connection type in a record.

void c_ph();    // Changes the phone in a record.
```

```

void del();          // Deletes a record.

int date(void);     // This function is for displaying System Date,

/*VALIDATION FUNCTIONS*/

void check_id(char *,int); //Checks for the length and validity of SIM ID

void check_name(char *,int); //Checks for the length and validity of NAME

void check_add(char *,int); //Checks for the length and validity of ADDRESS

void check_phone(char *,int); //Checks for the length and validity of PHONE
NUMBER

void check_ct(int *); //Checks for the CONNECTION TYPE

int duplicate(char idno[]); //Checks for the Duplication of the SIM ID

/*MISC FUNCTION*/

void empty();

/*STRUCTURE DECLARATION*/

struct customer
{
    char id[NUM];          /* ID of Length 5*/
    char name[MIN];       /* Name of Length 30*/
    char address[MAX];    /* Address of Length 61*/
    char phone[PH];       /* Phone Number of Length 10*/
    int connection;      /* Connection Type of length 2*/
    int day;              /* It is used to display the day of length 2*/
    int month;            /* It is used to display the month of length 2*/
    int year;             /* it is used to display the year*/
}rec;

```

```
/******MAIN FUNCTION STARTS******/
```

```
void main()
```

```
{
```

```
    int c;
```

```
    do
```

```
    {
```

```
        int i;
```

```
        window(1,1,80,25);
```

```
        textbackground(LIGHTRED);
```

```
        textcolor(LIGHTGRAY);
```

```
        clrscr();
```

```
        gotoxy(24,2);
```

```
        printf("<< MOBILINK CUSTOMER DATABASE >>");
```

```
        gotoxy(24,3);
```

```
        cprintf("<< BY-HAIDER ALLAMI >>");
```

```
        for(i=10;i<=70;i++)          //This 'FOR' loop will print a
```

combination of

```
        {          //asteriks '*' and dash '-' horizontally till
```

```
            gotoxy(i,4); //the 70th column is reached.
```

```
            printf("Ü");
```

```
            gotoxy(71,4); //the 70th column is reached.
```

```
            printf("Ü");
```

```
            gotoxy(i,7);
```

```
            printf("-");
```



```

gotoxy(i,9);
printf("-");
gotoxy(i,11);
printf("-");
gotoxy(i,13);
printf("-");
gotoxy(69,16);
printf("B");
gotoxy(71,16);
printf("B");
gotoxy(i,16);
printf("B");
}
for(i=5;i<=15;i++) //This 'FOR' loop will print asteriks '*'
{
    //vertically till the 15th row is reached.
    gotoxy(10,i);
    printf("YB");
    gotoxy(70,i);
    printf("YB");
}
for(i=6;i<=12;i++) //This 'FOR' loop will print '|' sign
{
    //vertically till the 12th row is reached.
    gotoxy(40,i);

```

```
        printf("");
    }
    gotoxy(15,6);
    printf("[1] ENTER A RECORD");
    gotoxy(15,8);
    printf("[2] VIEW RECORDS");
    gotoxy(15,10);
    printf("[3] SEARCH RECORD");
    gotoxy(15,12);
    printf("[4] DELETE RECORD");
    gotoxy(43,6);
    printf("[5] CHANGE NAME");
    gotoxy(43,8);
    printf("[6] CHANGE ADDRESS");
    gotoxy(43,10);
    printf("[7] CHANGE CONNECTION TYPE");
    gotoxy(43,12);
    printf("[8] CHANGE PHONE NUMBER");
    gotoxy(37,14);
    printf("[9] EXIT");
    gotoxy(10,20);
    textcolor(LIGHTGRAY);
    cprintf("PLEASE SELECT A CHOICE: ");
```

```

        c = (getche());

        switch (c) //This 'SWITCH' structure will ask the user for input
        from [1] to [9] and will display error on Invalid Entry.
        {

                case '1': add(); break; //It will call the add()
        funtion to allow user to add and save record.

                case '2': view(); break; //It will call the
        view() funtion to allow user to view the saved record.

                case '3': search(); break; //It will call the
        search() funtion to allow user to search records.

                case '4': del(); break; //It will call the delete()
        funtion to allow user to delete a saved record.

                case '5': c_name(); break; //It will call the
        c_name() funtion to allow user to modify name of a saved record.

                case '6': c_add(); break; //It will call the
        c_add() funtion to allow user to modify address of a saved record.

                case '7': c_ct(); break; //It will call the c_ct()
        funtion to allow user to modify connection type & date of a saved record.

                case '8': c_ph(); break; //It will call the
        c_ph() funtion to allow user to modify phone number of a saved record.

                case '9': gotoxy(30,24);printf("<<<<--THANK
        YOU-->>>>");getch();exit(0); //It will terminate the program.

                default:

                sound(500);delay(200);nosound(); //sound
        turns the PC speaker on at the specified frequency, nosound turns the PC speaker off.

                gotoxy(26,24);

                puts("<<--ENTER FROM 1-9 PLEASE-->>");
        //This message will only print on INVALID ENTRY and Will ask again for
        input.

                getch();

```

```

        }
    }
    while(c != '9');    //This 'WHILE' loop will continue till the Exit option [9]
    is entered.

    getch();
}

/***** END OF MAIN FUNCTION *****/

void add()
{
    FILE *fp;
    char ch;
    char temp[NUM];
    if(fp!=NULL)
    {
        do{

            fp=fopen("customer.rec", "ab");
            clrscr();
            gotoxy(25,2);
            printf("<<---ADDING RECORD--->>");
            do{

                gotoxy(1,4);
                printf("Enter [UNIQUE] SIM ID: ");
                check_id(temp, NUM);
            }
        }
    }
}

```

```

        }while(duplicate(temp));           //This Condition
Will keep on executng till a non duplicate ID is entered.

        strcpy(rec.id, temp);           //After verifying,
ID in 'temp' will copy to 'rec.id'

        gotoxy(1,6);

        printf("Enter NAME: ");

        check_name(rec.name,MIN);       //It will verify that
no neumerical input is given & the name input should not exceed the length limit

        gotoxy(1,8);

        printf("Enter ADDRESS: ");

        check_add(rec.address,MAX);     //It will verify that
only alpha neumeric input is entered & should not exceed the length limit

        gotoxy(1,10);

        printf("Enter PHONE Number: ");

        check_phone(rec.phone, PH);     //It will verify
phone number of a particular length

        gotoxy(1,12);

        printf("Enter CONNECTION TYPE: ");

        check_ct(&rec.connection);     //It will allow user
to select between 'STAR' & 'JAZZ' only

        {

                struct date d;

                getdate(&d);

                gotoxy(1,12);

                printf("DATE OF PURCHASE: %d %d %d
",d.da_day,d.da_mon,d.da_year);

                rec.day = d.da_day;

```

```

        rec.month = d.da_mon;
        rec.year = d.da_year;
    }
    window(1,1,80,25);
    gotoxy(1,12);
    fwrite(&rec, sizeof(rec), 1, fp);    //Record will be
written to file.

    fclose(fp);
    textbackground(BLACK);
    gotoxy(20,24);
    printf("DO YOU WISH TO CONTINUE..[Y]/[N]: ");
    ch=toupper(getche());
}while(ch == 'Y'); //Repeat loop while user inputs 'Y'
}
else
{
    gotoxy(20,24);
    printf("Unable To Open File...");
}
}
/*****
| Function: view()
|
| Purpose : To Display all the records in the file in a
|

```





```

printf(" NAME: %s",rec.name);
gotoxy(3,9);
printf(" ADDRESS: %s",rec.address);
gotoxy(3,11);
printf("     PHONE     Number:
%s",rec.phone);

printf("     CONNECTION     TYPE:
%s",rec.connection);

if(rec.connection==1)
{
    gotoxy(21,13);
    printf("AIRTEL");
}
else
if(rec.connection==2)
{
    gotoxy(21,13);
    printf("IDEA");
}
gotoxy(3,15);
printf(" DATE Of PURCHASE: %d-
%d-%d",rec.day,rec.month,rec.year);

}

```

```

        if(stremp(rec.id,"")==NULL)
        {
            clrscr();
            gotoxy(20,13);
            sound(500);delay(200);nosound();
            printf("<<===--THERE ARE NO RECORDS
TO VIEW--==>>");

            getch();
        }
        fclose(fp);
    }
    else
    {
        gotoxy(20,24);
        printf("Unable To Open File...");
    }
}

/*****END OF VIEW FUNCTION*****/

void search()
{
    int ch;
    FILE *fp;
    char sno[NUM];
    clrscr();

```

```

fp=fopen("customer.rec","rb");
gotoxy(20,13);
printf("PLEASE ENTER SIM ID ==>>>> ");
check_id(sno,NUM);
    while ( fread(&rec, sizeof(rec), 1, fp) && strcmp(sno,rec.id) );
        ch=strcmp(sno,rec.id);
        if(ch!=0)
            {
                clrscr();
                sound(500);delay(200);nosound();
                gotoxy(28,13);
                printf("??--RECORD NOT FOUND--
??");
                getch();
            }
        else
            {
                clrscr();
                gotoxy(28,2);
                printf("<<==SEARCH
SUCCESSFUL==>>");
                gotoxy(3,5);
                printf(" SIM ID: %s",rec.id);
                gotoxy(3,7);

```

```

printf(" NAME: %s",rec.name);
gotoxy(3,9);
printf(" ADDRESS: %s",rec.address);
gotoxy(3,11);
printf("     PHONE     Number:
%s",rec.phone);

printf("     CONNECTION TYPE:
%s",rec.connection);

if(rec.connection==1)
{
    gotoxy(21,13);
    printf("AIRTEL");
}
else
if(rec.connection==2)
{
    gotoxy(21,13);
    printf("IDEA");
}
gotoxy(3,15);
printf(" DATE Of PURCHASE: %d-
%d-%d",rec.day,rec.month,rec.year);

getch();
}

```

```

        fclose(fp);
    }
    /*****END OF SEARCH FUNCTION*****/
void c_name()
{
    int ch;
    FILE *fp;
    char sno[NUM];
    clrscr();
    fp=fopen("customer.rec","rb+");
    gotoxy(20,13);
    printf("PLEASE ENTER SIM ID ==->>>> ");
    check_id(sno,NUM);
        while ( fread(&rec, sizeof(rec), 1, fp) && strcmp(sno,rec.id) );
            ch=strcmp(sno,rec.id);
                if(ch!=0)
                    {
                        clrscr();
                        sound(500);delay(200);nosound();
                        gotoxy(28,13);
                        printf("??--RECORD NOT FOUND--
??");
                        getch();
                    }
}

```



```

else
{
    clrscr();
    gotoxy(28,4);
    printf("<<<--==RECORD
FOUND==-->>>");

    gotoxy(3,8);
    printf("OLD NAME: %s",rec.name);
// This will print the old name already stored.

    gotoxy(3,12);
    printf("ENTER NAME: "); //
This will allow the user to change the old name and input a new name.

    check_name(rec.name,MIN); //
The check will validate each character entered turn by turn.

    fseek(fp, ftell(fp) - sizeof(rec),0);
    fwrite(&rec, sizeof(rec), 1, fp);
//The new name will be added to the record.

    gotoxy(34,24);
    printf("RECORD SAVED !!!");

    getch();
}

fclose(fp);
}

/*****END OF CHANGE NAME FUNCTION*****/

void c_add()
{

```

```

int ch;

FILE *fp;

char sno[NUM];

clrscr();

fp=fopen("customer.rec","rb+");

gotoxy(20,13);

printf("PLEASE ENTER SIM ID ==->>>> ");

check_id(sno,NUM);

    while ( fread(&rec, sizeof(rec), 1, fp) && strcmp(sno,rec.id) );

        ch=strcmp(sno,rec.id); // Here the result after comparing
the values of the two strings by using the 'strcmp' function is assigned to 'ch'.

SIM ID's dont compare.        if(ch!=0)    // This loop will execute if the

                                {

                                    clrscr();

                                    sound(500);delay(200);nosound();

                                    gotoxy(28,13);

                                    printf("??--RECORD NOT FOUND--

??");

                                    getch();

                                }

will be executed.        else    // If the ID's match, then this block

                                {

                                    clrscr();

```

```

                                gotoxy(28,4);
                                printf("<<<--==RECORD
FOUND==-->>>");

                                gotoxy(3,8);
                                printf("OLD          ADDRESS:
%s",rec.address);

                                gotoxy(3,12);
                                printf("ENTER NEW ADDRESS: ");
                                check_add(rec.address,MAX);
                                fseek(fp, ftell(fp) - sizeof(rec),0);
                                //fseek sets the file pointer associated with a stream to a new position.
                                fwrite(&rec, sizeof(rec), 1, fp);
                                //fwrite appends a specified number of equal-sized data items to an output
file.

                                gotoxy(34,24);
                                printf("RECORD SAVED !!!");
                                getch();
                                }

                                fclose(fp);

                                }

/*****END OF CHANGE ADDRESS FUNCTION*****/

void c_ct()
{
    int ch;
    FILE *fp;
    char sno[NUM];

```

```

clrscr();

fp=fopen("customer.rec","rb+");

gotoxy(20,13);

printf("PLEASE ENTER SIM ID ==->>>> ");

check_id(sno,NUM);

while ( fread(&rec, sizeof(rec), 1, fp) && strcmp(sno,rec.id) );

    ch=strcmp(sno,rec.id);

    if(ch!=0)

        {

            clrscr();

            sound(500);delay(200);nosound();

            gotoxy(28,13);

            printf("??--RECORD NOT FOUND--

??");

            getch();

        }

    else

        {

            clrscr();

            gotoxy(28,2);

            printf("<<<<--RECORD

FOUND==->>>>");

            gotoxy(3,6);

            printf("OLD DATE OF PURCHASE:

%d %d %d",rec.day,rec.month,rec.year); //Old Date Of purchase is displayed.

```

```

    {
        struct date d;
        getdate(&d);
        gotoxy(3,8);
        printf("NEW DATE OF
PURCHASE: %d %d %d ",d.da_day,d.da_mon,d.da_year); //New Date Of
Purchase is stored.

        rec.day = d.da_day;
        rec.month = d.da_mon;
        rec.year= d.da_year;
    }
    gotoxy(3,10);
    printf("OLD CONNECTION TYPE:
%s",rec.connection); //Old Connection type is displayed
    if(rec.connection==1)
    {
        gotoxy(24,10);
        printf("AIRTEL"); //If
option stored in record is '1' Then "STAR" will be displayed on screen.
    }
    else
    if(rec.connection==2)
    {
        gotoxy(24,10);
        printf("IDEA"); //If option
stored in record is '2' Then "JAZZ" will be displayed on screen.
    }
}

```

```

    }
    gotoxy(3,12);
    printf("NEW CONNECTION TYPE:
"); //New Connection Type is asked for Input.
    check_ct(&rec.connection); //This
check provides choices to the user to select b/w "STAR" or "JAZZ".
    fseek(fp, ftell(fp) - sizeof(rec),0);
    fwrite(&rec, sizeof(rec), 1, fp);
    window(1,1,80,25);
    gotoxy(1,12);
    gotoxy(35,24);
    printf("RECORD SAVED !!!");
    getch();
}
fclose(fp);
}
/*****END OF CHANGE CONNECTION TYPE FUNCTION*****/
void c_ph()
{
    int ch;
    FILE *fp;
    char sno[NUM];
    clrscr();
    fp=fopen("customer.rec","rb+");

```

```

gotoxy(20,13);

printf("PLEASE ENTER SIM ID ==->>>> ");

check_id(sno,NUM);

while ( fread(&rec, sizeof(rec), 1, fp) && strcmp(sno,rec.id) );

    ch=strcmp(sno,rec.id);

    if(ch!=0)

        {

            clrscr();

            sound(500);delay(200);nosound();

            gotoxy(28,13);

            printf("??--RECORD NOT FOUND--

??");

            getch();

        }

    else

        {

            clrscr();

            gotoxy(28,4);

            printf("<<<--RECORD

FOUND==->>>>");

            gotoxy(3,8);

            printf("OLD PHONE NUMBER:

%s",rec.phone); //Old phone number is displayed.

            gotoxy(3,12);

```



```

                printf("ENTER NEW PHONE
NUMBER: "); //New phone number is asked by the user.

                check_phone(rec.phone,PH);

                fseek(fp, ftell(fp) - sizeof(rec),0);

                fwrite(&rec, sizeof(rec), 1, fp);

                gotoxy(34,24);

                printf("RECORD SAVED !!!");

                getch();

            }

        fclose(fp);

    }

/*****END OF CHANGE PHONE NUMBER FUNCTION*****/

void del()

{

    int ch;

    FILE *fp;

    char sno[NUM];

    clrscr();

    fp=fopen("customer.rec","rb+");

    gotoxy(20,13);

    printf("PLEASE ENTER SIM ID ==->>>> ");

    check_id(sno,NUM);

    while ( fread(&rec, sizeof(rec), 1, fp) && strcmp(sno,rec.id) );

        ch=strcmp(sno,rec.id);

```

```

if(ch!=0)
{
    clrscr();
    sound(500);delay(200);nosound();
    gotoxy(28,13);
    printf("??--RECORD NOT FOUND--
??");
    getch();
}
else //Here the saved record is displayed.
{
    clrscr();
    gotoxy(28,2);
    printf("<<==SEARCH
SUCCESSFUL==>>");
    gotoxy(3,5);
    printf(" SIM ID: %s",rec.id);
    gotoxy(3,7);
    printf(" NAME: %s",rec.name);
    gotoxy(3,9);
    printf(" ADDRESS: %s",rec.address);
    gotoxy(3,11);
    printf("     PHONE     Number:
%s",rec.phone);
    gotoxy(3,13);

```

```
%" ,rec.connection);
```

```
printf(" CONNECTION TYPE:
```

```
if(rec.connection==1)
```

```
{
```

```
gotoxy(21,13);
```

```
printf("AIRTEL");
```

```
}
```

```
else
```

```
if(rec.connection==2)
```

```
{
```

```
gotoxy(21,13);
```

```
printf("IDEA");
```

```
}
```

```
gotoxy(3,15);
```

```
printf(" DATE OF PURCHASE: %d-
```

```
%d-%d" ,rec.day,rec.month,rec.year);
```

```
gotoxy(26,24);
```

```
printf("PRESS ANY KEY TO
```

```
DELETE RECORD.....");
```

```
getch();
```

```
empty(); //this is a Function which
```

will erase the record in memory & NOT physically.

```
fseek(fp, ftell(fp) - sizeof(rec), 0);
```

```
fwrite(&rec, sizeof(rec), 1, fp);
```

```
clrscr();
```

```

        sound(500);delay(200);nosound();
        gotoxy(33,13);
        printf("RECORD DELETED !!!");
        getch();
    }
    fclose(fp);
}
/*****END OF DELETE FUNCTION*****/
void check_id(char *p, int size)
{
    int i=0;
    char ch;
    do
    {
        ch=getch();
        if( (ch>='0' && ch<='9') && (i<size-1) )
        {
            pointer.    *p=ch;    //The value is assigned to the
            p++;        //Pointer is incremented.
            i++;        //Length counter is incremented.
            printf("%c",ch); //Validated character is printed.
        }
        else //This block of commands controls backspace.

```

```

        if(ch==8 && i>0)
        {
                printf("%c%c%c",8,32,8);

                i--;      //Length counter is decremented.

                p--;      //Pointer is decremented.

        }

    }while(ch!=13 || i<size-1); //this loop will continue till the total given
length is reached or 'ENTER' is pressed.

    *p='\0';
}

void check_phone(char *ph, int size)
{

    int p=0;
    char s[20],ch;
    do
    {

        ch=getch();

        if( ((ch>='0' && ch<='9')||(ch=='-')) && (p<size-1) )
        {

            *ph=ch;

            ph++;

            p++;

            printf("%c",ch);

        }

    }

```

```

        else
        if( (ch==8)&&(p>0) )
        {
                printf("%c%c%c",8,32,8);

                p--;

                ph--;

        }
} while(ch!=13 || p<size-1);

s[p]='\0';
*ph='\0';
}

void check_name(char *p, int size)
{
        int j=0;
        char x[40],cj;
        do
        {
                cj=toupper(getch());
                if(((cj>='a' && cj<='z')||(cj>='A' && cj<='Z')||(cj==' ')) &&
(j<size-1))
                {
                        *p++=cj;
                        j++;
                        printf("%c",cj);

```

```

    }
    else
    if(cj==8 && j>0)
    {
        printf("%c%c%c",8,32,8);
        j--; p--;
    }
} while(cj!=13);
x[j]='0';
*p='0';
}
void check_add(char *p, int size)
{
    int l=0;
    char a[50],ad;
    do
    {
        ad=toupper(getch());
        if(((ad>='a' && ad<='z')||(ad>='A' && ad<='Z'))||(ad==' ')||
        (ad=='\n')||(ad=='\t')||(ad>='0' && ad<='9')) && (l<size-1))
        {
            *p++=ad;
            l++;
            printf("%c",ad);

```



```

    }
    else
    if( (ad==8)&&(l>0) )
    {
        printf("%c%c%c",8,32,8);
        l--; p--;
    }
}while(ad!=13);
a[l]='\0'; *p='\0';
}

```

```
void check_ct(int *p)
```

```

{
    int ab=1;
    char ch;
    int aa;
    window(24,18,51,24);
    textbackground(BLACK);
    clrscr();
    window(25,19,50,23);
    textbackground(WHITE);
    textcolor(BLACK);
    clrscr();
    gotoxy(11,2);
}

```

```
printf("AIRTEL");  
gotoxy(11,4);  
printf("IDEA");  
do{  
    ch = getch();  
    if(ch==0)  
    {  
        ch = getch();  
        if(ch==72)  
        {  
            ab=ab-1;  
        }  
    else  
        if(ch==80)  
        {  
            ab=ab+1;  
        }  
    }  
    if(ab<1)  
    {  
        ab=1;  
    }  
    else
```

```
if(ab==1)
{
    window(1,1,80,25);
    window(35,22,40,22);
    textbackground(WHITE);
    textcolor(BLACK);
    clrscr();
    printf("IDEA");
    window(1,1,80,25);
    window(35,20,41,20);
    textcolor(RED);
    clrscr();
    printf("AIRTEL");
    aa=1;
    window(1,1,80,25);
}
else
if(ab==2)
{
    window(1,1,80,25);
    window(35,20,41,20);
    textbackground(WHITE);
    textcolor(BLACK);
```

```

        clrscr();
        printf("AIRTEL");
        window(1,1,80,25);
        window(35,22,40,22);
        textcolor(BLACK);
        clrscr();
        aa=2;
        printf("IDEA");
        window(1,1,80,25);
    }
    if(ab>2)
    {
        ab=2;
    }
} while(ch != 13);
if (aa==1)
{
    gotoxy(24,12);
    printf("AIRTEL");
    *p-1;
}
else
if (aa==2)

```

```

        {
            gotoxy(24,12);
            printf("IDEA");
            *p=2;
        }
    getch();
    window(24,18,51,24);
    textbackground(LIGHTRED);
    textcolor(LIGHTGRAY);
    clrscr();
}

void empty()
{
    strcpy(rec.id, " ");
    strcpy(rec.name, " ");
    strcpy(rec.address, " ");
    strcpy(rec.phone, " ");
    rec.connection=0;
    rec.day=0;
    rec.month=0;
    rec.year=0;
}

int duplicate(char idno[])

```

```
{  
    FILE *fp;  
    fp=fopen("customer.rec","rb");  
    while(fread(&rec, sizeof(rec), 1, fp) && strcmp(rec.id, idno));  
    fclose(fp);  
    if(strcmp(rec.id, idno)==0)  
        return 1;  
    else  
        return 0;  
}  
/*****| END OF PROJECT |*****/
```

```
<< MOBILINK CUSTOMER DATABASE >>  
<< HAIDER KHALAF >>
```

[1] ENTER A RECORD	:	[5] CHANGE NAME
[2] VIEW RECORDS	:	[6] CHANGE ADDRESS
[3] SEARCH RECORD	:	[7] CHANGE CONNECTION TYPE
[4] DELETE RECORD	:	[8] CHANGE PHONE NUMBER
[9] EXIT		

PLEASE SELECT A CHOICE:



```
Turbo C++ IDE
<<-- ADDING RECORD -->
Enter UNIQUE SIM ID: 1234
Enter NAME: HAIDER KHALAF
Enter ADDRESS: KALKAJI B BLOCK 43A 1ST FLOOR
Enter PHONE Number: 07838448107
Enter CONNECTION TYPE: IDEA

DATE OF PURCHASE: 17 4 2011

DO YOU WISH TO CONTINUE..(Y)/N):
```



```
CA Turbo C++ IDE

<<<---RECORD FOUND--->>>

OLD NAME: HAIDER

ENTER NAME: KHALAF
```

<<<---RECORD FOUND--->>>

OLD ADDRESS: KALKAJI B BLOCK 43A 1ST FLOOR

ENTER NEW ADDRESS: KALKAJI O. BLOCK 40A 1ST FLOOR

<<<--==RECORD FOUND==-->>>

OLD DATE OF PURCHASE: 17 4 2011

NEW DATE OF PURCHASE: 17 4 2011

OLD CONNECTION TYPE: IDEA

NEW CONNECTION TYPE: AIRTEL\_

AIRTEL

IDEA

```
Turbo C++ IDE

<<<---RECORD FOUND--->>>

OLD PHONE NUMBER: 07838448107

ENTER NEW PHONE NUMBER: 09910215100

RECORD SAVED !!!_
```

## **CONCLUSION**

- EXTENT TO WHICH THE ORIGINAL OBJECTIVES WERE MET.
- USER FRIENDLY INTERFACE.
- PROJECT MANAGEMENT, CONSULTANCY AND COMMUNICATION
- SKILLS ACQUIRED.
- EXTENT TO WHICH USER REQUIREMENTS WERE MET.
- FINAL SUMMARY.



## CONCLUSION

This will review the initial objectives and the extent to which they were fulfilled. It will summarize the project's success and weakness, and what was learnt during its duration.

### Extent to Which the Original Objectives Were Met

The project objectives, as defined we can summarize as follows. To produce a system that provides a very friendly graphical user interface, permitting the user to perform basic administration duties on a database containing records of employees. In doing so, an appreciation of project management, communication and consultancy skills should be acquired, along with a thorough understanding of the development of windows based applications using C++. I feel that all of these aims were achieved, some to greater extent than others. I will break down these objectives in the following sections, and discuss what was achieved from them.

### User Friendly Interface

The final result of the **mobilink** consolidated an easy to use and hence effective user interface, with the basic user functionality requested from the management of the client. A lot of time and careful consideration was taken to ensure that it encompassed all of the characteristics of a successful user interface, and one which tried to inspire confidence in its user. Continual communication, in the form of message and the dialogue facility, from the system to its user gives them a feeling of security and power. Minute details, like the positioning of certain details to the screen so that the user could digest them swiftly involved significant thought, and the psychology behind the design of these features of the interface is extensively explained in chapter user interface.

### **Project Management, Consultancy and Communications Skills Acquired.**

I learnt a great deal from setting myself the objective of gaining experience in these areas. This does not necessarily mean, however, that I accomplished putting these skills into practice proficiently. On the contrary, upon reflection of the project and its end result. In retrospect, I now know that these skills are on-going learning experiences, and upon the scheduling of further projects, the accuracy of estimation should increase.

### **Extent to Which User Requirements were Met**

Great emphasis was laid on the requirements as specified are incorporated into the system. The requirements demands the system to be safe and easy to use, promptly displaying any necessary requested information via graphical user interface, with guidance provided where requested. All of these requirements were carefully considered before being implemented, so that they would be successful and effective.

The requirements demanded a user friendly system that had the possibility of helping the user where appropriate. This is achieved through the facility of continual communication from the system, in the form of the messages the system displays to the user about possible mistakes, and whether processes should continue or not.

# SCOPE OF THE PROJECT

## SCOPE OF THE PROJECT

This s/w will play an important role in future user for proper inventory management. At present data from the database and report are created manually. Need for the system arise due to increasing expenditure of the company in maintaining records regarding the details of consumers as well as company (supplier) .Basic need of the project is to keep the details of the company's company, customers, and inventory.

At present details collected, maintained and inserted manually. After implementation of this project, record will be inserted and maintained technically. This will reduce manual work and improve quality and speed and facility will maintain the queue of all the records that are entered by the management of the Restaurant. It will help the management to access the data and results as and when required.

## References

- ✓ BOOK: Object Oriented Programming in Turbo C++  
AUTHOR: Robert Lafore  
PUBLISHER: Galgotia Publications pvt. Ltd.
  
- ✓ BOOK : The C++ Programing Language  
AUTHOR: Bjarne Stroustrup  
PUBLISHER : Atst Lafs
- ✓ BOOK : Expert Data Structures With C++  
AUTHOR: Rb Patel  
PUBLISHER: Khanna Books Publishing
  
- ✓ BOOK : Programming With C++  
AUTHOR:E.BalagurnSwamy  
PUBLISHER: Bpb Pulications
  
- ✓ BOOK: C++ Power Packed  
AUTHOR : By Khanna  
PUBLISHER:Tata Mcgraw Hill
  
- ✓ BOOK: Expert Data Structure Using By C++  
AUTHOR: By Sagar  
PUBLISHER: Bpb Publication
  
- ✓ BOOK: A Beginners Guide  
AUTHOR: By Schilt Herbat  
PUBLISHER: Willey India

JAMIA HAMDARD  
(Hamdard University)  
Hamdard Nagar, New Delhi-110062.

S. No.....

Application Form for Issuing of Bonafide Certificate

Dated.....

Mr./Ms.....Son/Daughter of.....

Nationality.....Program.....Year/Sem.....

Passport No.....Identity Card No.....Session.....

Address.....

.....Mobile No.....

E. Mail I.D.....

Tuition Fees Rs.....Receipt No.....

Hostel Fees Rs.....Receipt No..... and

has paid by the student. He/she applied for bonafide certificate. So please certify the above mentioned information given by him/her.

Signature and Seal of  
Head of the Department

Signature and Seal of the  
Warden (If Hosteller)

Dr. Asim Ali Khan  
Foreign Students' Advisor