



INTERACTIVE VOICE RESPONSE SYSTEM

By

R. MAHESHI

Reg. No: 71202621019

Of

KUMARAGURU COLLEGE OF TECHNOLOGY

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

Of

MASTER OF COMPUTER APPLICATIONS

June, 2005

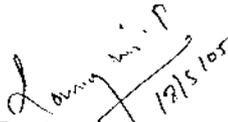
BONAFIDE CERTIFICATE

Certified that this project report titled
INTERACTIVE VOICE RESPONSE SYSTEM

Is Bonafide work of

Mr. R.MAHESH (Reg. No: 71202621019)

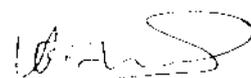
Certified that this project report titled Interactive Voice Response System is the bonafide work of Mr. R. Mahesh who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.


Project Guide


Head of the Department

The Candidate with University Register No. 71202621019 was
examined by us in the project Viva-Voce examination held on

24/06/2005


Internal Examiner


External Examiner
24/6/05

ABSTRACT

The main objective of **Interactive Voice Response System** is automating the gas booking system. In the system a consumer can use his telephone line to book the cylinder, by just dialing the agency number and then entering his consumer number.

The Interactive Voice Response System was designed to achieve maximum flexibility in booking the gas cylinder. It is a PC based automatic telephone service which enables us to interact directly with the data stored in a computer data base through telephone. This project is specially designed for consumers those who are already registered with the agency. The design and user interface are made as simple as possible enabling even a common man to interact with the system and retrieve the necessary details.

This system will ensure that it manages the details of the consumer in a better, faster, very efficient, and reliable manner. It is very cost effective and fault tolerant software providing uninterrupted quality service. The system is flexible and user friendly to the employees of the Gas Agency and the consumer.

In an Interactive Voice Response System, a call is answered automatically when it is received, and a prerecorded message is played over the call. This message usually offers callers choices, such as Press one to book the cylinder, press two to know the booking status.

TABLE OF CONTENTS

CONTENTS	PAGE NO
ABSTRACT	iii
LIST OF FIGURES	vi
1. INTRODUCTION	
1.1 Interactive Voice Response System	1
1.2 Organization Profile	3
2. REQUIREMENT SPECIFICATION	
2.1 Hardware Requirement	7
2.2 Software Requirement	7
2.3 Software Overview	7
3. ANALYSIS OF THE PROBLEM	
3.1 Problem Definition	16
3.2 System Analysis	
3.2.1 Existing system	16
3.2.2 Proposed system	17
3.3 Feasibility Study	
3.3.1 Economical Feasibility	18
3.3.2 Technical Feasibility	18
3.3.3 Operational feasibility	19
3.3.4 Justification	19

4. SYSTEM DESIGN	
4.1 Modular Design	20
4.2 Interface Design	21
4.3 Functional Independence	22
4.4 Cohesion	22
4.5 Database Design	23
4.6 Input and Output Design	24
4.7 Data Flow Diagram	27
4.8 Data Dictionary	31
5. TESTING AND IMPLEMENTATION	
5.1 Testing	33
5.2 Implementation	35
6. CONCLUSION & FURTHER SCOPE	
6.1 Conclusion	36
6.2 Further scope	36
7. APPENDICES	37
8. REFERENCES	45

LIST OF TABLES

Table No	Table Name	Page No
3.3.2.1	ConsumerDetails	31
3.3.2.2	BookingDetails	31
3.3.2.3	DeliveryDetails	32
3.3.2.4	ComplaintDetails	32

LIST OF FIGURES

Figure No	Figure Name	Page No
4.7.1	Gas Booking Process	27
4.7.2	Complaint Booking Process	28
4.7.3	Booking & Delivery Status Process	29
4.7.4	Overall Process of IVRS	30

CHAPTER 1

INTRODUCTION

1.1 INTERACTIVE VOICE RESPONSE SYSTEM

The main objective of **Interactive Voice Response System** is automating the gas booking system. In the system a consumer can use his telephone line to book the cylinder, by just dialing the agency number and then entering his consumer number.

In an Interactive Voice Response System, a call is answered automatically when it is received, and a prerecorded message is played over the call. This message usually offers callers choices, such as Press one to book the cylinder or press two to know the booking status.

The main objective behind the development of this project is to achieve maximum flexibility of booking the gas cylinder. It is a PC based Automatic Telephone service which enables to interact directly with information about a result stored in a computer data base through telephone. This project is specially designed for consumers those who are already registered in the agency. The design and user interface are made as simple as possible enabling even a common man to interact with the system and retrieve the necessary details.

It maintains all the consumer information's in a database and the consumer can interact with this database any where in the telephone network to know their details with more flexibility

PROJECT SCOPE

Purpose:

The main aim of Interactive Voice Response System is to book a cylinder through a telephone line.

Benefits:

- The consumer can book a cylinder from his home through telephone line.
- It saves a lot of time.
- The consumer can book a cylinder at any time.
- It reduce the cost.
- No manual intervention.
- Keeping the process simple

1.3 ORGANIZATION PROFILE

Rangasoftt is a group of Ranga Informatics Private Limited having diversified interests created a state of art software development facilities at Coimbatore, India to leverage the advantage of Information Technology. The value addition that happens at our process enables us to retain our customers. Mr.S.S.Sambandan, CEO and Founder, an Engineering Graduate having 20 years of experience in International Trade and services heads the Ranga group. The basic value proposition and continues efforts from CEO has put the Group companies on the top for 10 consecutive years.

Ranga group has enriched business relations around the world and has tie-up with large trading houses in Japan, Korea, Europe and USA. Apart from providing solutions covering the graphical areas including Japan, Korea, Malaysia, Singapore, Taiwan, Hong Kong, Mauritius and all parts of Europe, United States of America, Rangasoftt catered to domestic market with some innovative products. With the passion to strive for the best with established infrastructure, process and world class people Ranga is all set to take the opportunities existing in the globe.

Rangasoftt mission is to provide world-class solutions to the globe with repetitive accuracy both in quality and time. They continue to apply cutting edge technologies to enable their clients to leverage the advantage of low Cost, reliable solution with basic strategic policy of realizing maximum possible ROI within a set time frame.

Quality policy is:

Rangasoftt totally commits to excel in quality and in its operations for providing software services and products with active involvement of our global customers, employees and suppliers towards sustainable customer satisfaction.

Technical Focus :

Rangasoftt is engaged in off shore and on site Software Development activities with prestigious clients. The company is all set for the tremendous growth. Needless to say, it's their business to continuously promote effective solutions and professionally pursue low-risk, high-yield information while maintaining the highest standards.

Rangasoftt has expertise in providing solutions in Embedded System, ERP and manufacturing solutions, EA integrations, Web Enabled Systems, Security systems and Device Drivers.

Business Focus :

They are currently focus in the following areas:

- ❖ Embedded System
- ❖ Image Processing
- ❖ Workflow customization and integration
- ❖ ERP
- ❖ Internet / Intranet Solutions
- ❖ EAI
- ❖ Device Driver Developments

Domain

Industrial Automation :

- ❖ Vision Systems
- ❖ SCADA and data acquisition
- ❖ Automation control system
- ❖ Web Interfaces for existing controller / Data acquisition system
- ❖ Testing equipments
- ❖ Customized embedded system
- ❖ Wireless control

Process Instrumentation :

Process flow monitoring and control.

- ❖ PID controls
- ❖ Security and alert system
- ❖ Wireless remote security and control

Internet Domain :

- ❖ EAI
- ❖ Workflow management

General Services :

- ❖ ERP
- ❖ Any custom development
- ❖ Embedded
- ❖ Firmware development
- ❖ Device driver development
- ❖ RTOS implementation for various boards

CHAPTER 2

REQUIREMENT SPECIFICATION

The Software Requirements Specification is a technical specification of requirements for the software product. The goal of software requirements definition is to completely and consistently specify the technical requirements for the software products in a concise and unambiguous manner.

The Software Requirements Specification is based on the system definition high-level requirements specified during initial planning are elaborated and more specific in order to characterize the features that the software product will incorporate. The requirement specification is primarily concerned with functional and a performance aspect of the software product and emphasis is placed on specifying product characteristics without implying how the product will provide those characteristics.

Desirable properties of a Software Requirement Specification

- Correct
- Complete
- Consistent
- Unambiguous
- Functional
- Verifiable
- Traceable.

2.1 HARDWARE REQUIRMENT

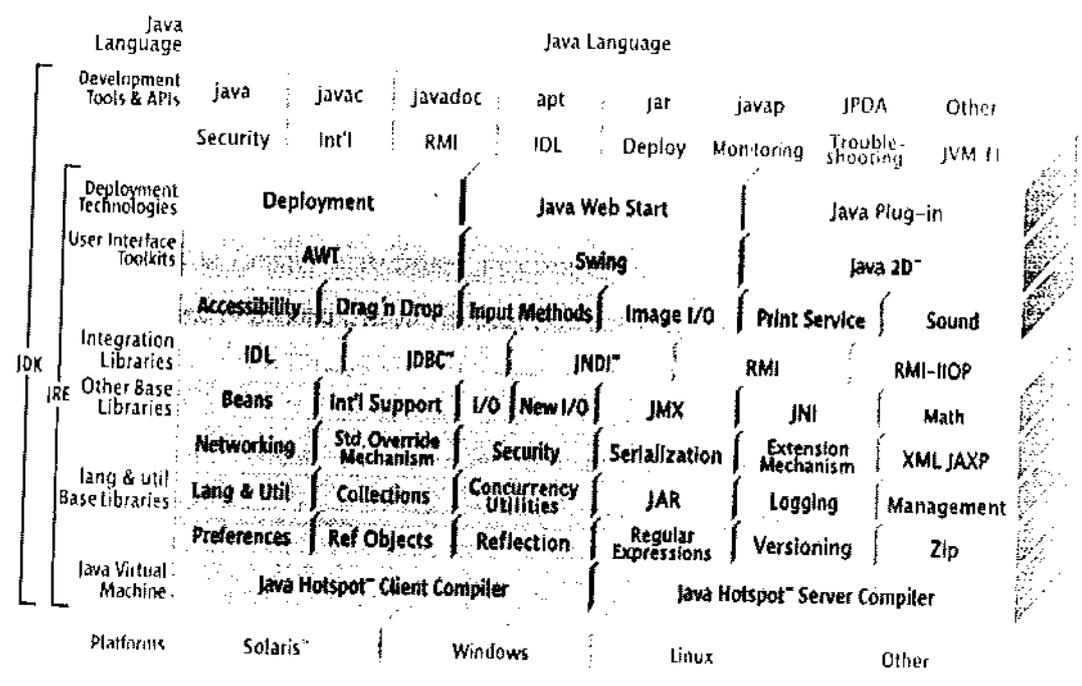
- Pentium processor
- 128MB RAM
- Voice Modem
- Telephone Connection

2.2 SOFTWARE REQUIRMENT

- Java
- JTAPI Package
- Oracle

2.3 SOFTWARE OVERVIEW

Java™ 2 Platform Standard Edition 5.0



Java™ Language Features

Generics

This long-awaited enhancement to the type system allows a type or method to operate on objects of various types while providing compile-time type safety. It adds compile-time type safety to the Collections Framework and eliminates the drudgery of casting.

Enhanced for Loop

This new language construct eliminates the drudgery and error-proneness of iterators and index variables when iterating over collections and arrays.

Autoboxing / Unboxing

This facility eliminates the drudgery of manual conversion between primitive types (such as int) and wrapper types (such as Integer).

Typesafe Enums

This flexible object-oriented enumerated type facility allows you to create enumerated types with arbitrary methods and fields. It provides all the benefits of the Typesafe Enum pattern without the verbosity and the error-proneness.

Varargs

This facility eliminates the need for manually boxing up argument lists into an array when invoking methods that accept variable-length argument lists..

Static Import

This facility lets you avoid qualifying static members with class names without the shortcomings of the "Constant Interface anti pattern".

Java language provides a lot of useful packages. The various packages used in this project are

➤ **java.awt**

It contains all of the classes for creating user interfaces and for painting graphics and images. A user interface object such as a button or a scrollbar is called, in AWT terminology, a component. The Component class is the root of all AWT components.

Some components fire events when a user interacts with the components. The AWTEvent class and its subclasses are used to represent the events that AWT components can fire.

A container is a component that can contain components and other containers. A container can also have a layout manager that controls the visual placement of components in the container. The AWT package contains several layout manager classes and an interface for building our own layout manager.

➤ **java.awt.event**

It provides interfaces and classes for dealing with different types of events fired by AWT components. Events are fired by event sources. An event listener registers with an event source to receive notifications about the events of a particular type. This package defines events and event listeners, as well as event listener adapters, which are convenience classes to make easier the process of writing event listeners.

➤ **java.awt.image**

It provides classes for creating and modifying images. Images are processed using a streaming framework that involves an image producer, optional image filters, and an image consumer. This framework makes it possible to progressively render an image while it is being fetched and generated. Moreover, the framework allows an application to discard the storage used by an image and to regenerate it at any time. This package provides a number of image producers, consumers, and filters that you can configure for your image processing needs.

➤ **java.io**

It provides for system input and output through data streams, serialization and the file system.

- **BufferedReader**

Read text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.

- **BufferedWriter**

Write text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.

- **ByteArrayInputStream**

A `ByteArrayInputStream` contains an internal buffer that contains bytes that may be read from the stream.

- **ByteArrayOutputStream**

This class implements an output stream in which the data is written into a byte array.

- `DataInputStream`

A data input stream lets an application read primitive Java data types from an underlying input stream in a machine-independent way.

- `DataOutputStream`

A data output stream lets an application write primitive Java data types to an output stream in a portable way.

- `File`

It's an abstract representation of file and directory pathnames.

- `FileDescriptor`

Instances of the file descriptor class serve as an opaque handle to the underlying machine-specific structure representing an open file, an open socket, or another source or sink of bytes.

- `FileInputStream`

A `FileInputStream` obtains input bytes from a file in a file system.

- `FileOutputStream`

A file output stream is an output stream for writing data to a `File` or to a `FileDescriptor`.

- `InputStream`

This abstract class is the super class of all classes representing an input stream of bytes.

- ObjectInputStreaaam

An ObjectInputStream deserializes primitive data and objects previously written using an ObjectOutputStream.

- ObjectOutputStream

An ObjectOutputStream writes primitive data types and graphs of Java objects to an OutputStream.

- ObjectStreamClass

Serialization's descriptor for classes.

- OutputStream

This abstract class is the super class of all classes representing an output stream of bytes.

- OutputStreamWriter

An OutputStreamWriter is a bridge from character streams to byte streams: Characters written to it are encoded into bytes using a specified charset.

- PrintStream

A PrintStream adds functionality to another output stream, namely the ability to print representations of various data values conveniently.

- PrintWriter

It is used to print formatted representations of objects to a text-output stream.

➤ **java.lang**

It provides the classes that are fundamental to the design of the Java programming language. The most important classes are Object, which is the root of the class hierarchy, and Class, instances of which represent classes at run time.

Frequently it is necessary to represent a value of primitive type as if it were an object. The wrapper classes Boolean, Character, Integer, Long, Float, and Double serve this purpose. An object of type Double, for example, contains a field whose type is double, representing that value in such a way that a reference to it can be stored in a variable of reference type. These classes also provide a number of methods for converting among primitive values, as well as supporting such standard methods as equals and hashCode. The Void class is a non-instantiable class that holds a reference to a Class object representing the primitive type void.

The class Math provides commonly used mathematical functions such as sine, cosine, and square root. The classes String and StringBuffer similarly provide commonly used operations on character strings.

Classes ClassLoader, Process, Runtime, SecurityManager, and System provide "system operations" that manage the dynamic loading of classes, creation of external processes, host environment inquiries such as the time of day, and enforcement of security policies.

➤ **javax.swing**

It provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms

➤ **javax.swing.event**

It provides methods for events fired by Swing components. It contains event classes and corresponding event listener interfaces for events fired by Swing components in addition to those events in the java.awt.event package.

ADVANTAGES OF SOFTWARE

JAVA

- Java is a platform independent language.
- Java has a wide array of scope in all fields of IT.
- Java is a highly efficient language for network based application.
- Java comes with a tool called javadoc that can make it easier to produce the documentation in a readable and organized format.
- Java is a Architecture-Neutral Language. Their goal is "write once: run anywhere, any time, forever."
- Java is made possible by pluggable look-and-feel.
- Java is a High Performance and Distributed language.

JTAPI

- The Java Telephony API (JTAPI) is a portable, object-oriented application programming interface for Java-based computer-telephony application.
- Brings simplicity to the most basic telephony applications.
- Provides a scalable framework that spans desktop applications to distributed call center telephony applications.
- Interfaces applications directly to service providers or acts as a Java interface to existing telephony APIs, such as SunXTL, TSAPI, and TAPI
- Based on a simple core that is augmented with standard extension packages.
- Runs on a wide range of hardware configuration, wherever java run-time can be used.

Oracle

Oracle is a powerful RDBMS product, that provides efficient and effective solutions for major database features. This includes

- Large database and space management control.
- Many concurrent database users.
- High transaction processing performance.
- High availability.
- Controlled availability.
- Industry accepted standards.
- Manageable security.
- Database enforceability.
- Client/Server environment.
- Distributed database systems.
- Portability.
- Compatibility.
- Connectivity.

CHAPTER 3

ANALYSIS OF THE PROBLEM

3.1 PROBLEM DEFINITION

The existing system has major drawbacks in calculations of consumer's next booking dates, delivery dates and generating the reports. The main draw back of existing system is, it is not user friendly. This may cause time delay, loss of accuracy and some times workers could not give the correct information about consumer. My project will overcome these problems.

3.2 SYSTEM ANALYSIS

3.2.1 EXISTING SYSTEM

In the existing system the consumer directly goes to the agency to book a cylinder.

DRAWBACKS OF THE EXISTING SYSTEM:

- Time consuming and Laborious.
- The booking process can be done only during working hours of the agency.
- Difficult to check the status of the transaction.
- More labor-intensive operations are required.
- Information retrieval is difficult.
- Information may not be accurate.

PROPOSED SYSTEM

Interactive Voice Response System is used to automate the gas booking system. In the system a consumer can use his telephone line to book the cylinder, by just dialing the agency number and then entering his consumer number.

The sequence of operations to be carried to book a cylinder is

- Consumer dials the agency number.
- On receiving the call from the consumer the system gives a welcome message and asks the consumer to enter the consumer number.
- The consumer types the consumer number in a telephone.
- The system validates the consumer number and asks the consumer to enter the type of operation to be performed.
- Perform the operation.
- Close the transaction.

ADVANTAGES OF THE PROPOSED SYSTEM:

- The consumer can book a cylinder from his home through a telephone line.
- It saves time.
- The consumer can book a cylinder at any time.
- It reduces the cost.
- No manual intervention.
- User- friendly.

3.3 FEASIBILITY STUDY

All projects are feasible given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility listed below are equally important. The following feasibility techniques have been used in this project.

3.3.1 ECONOMIC FEASIBILITY

In Economic feasibility, we should check whether there are sufficient benefits in creating the system to make the cost acceptable or not. If the cost of the project is very high than its benefits, then it will not satisfy the economic feasibility. In this project, the benefit is well satisfied than the cost needed to develop it.

The Interactive Voice Response System is designed in such a way that the financial benefits exceeds or equals the cost. It has been decided to develop a system at reasonable costs. The system is sure to be good investment for the organization, since it requires minimal cost of implementing.

3.3.2 TECHNICAL FEASIBILITY

Technical feasibility, we should check whether the project can be done with the current equipment and existing software technology. So that it is technically feasible. In this project we used the existing software technology. Going for a new software will be technically complex.

The proposed system is designed to operate under minimal technical requirements. There is a wide range of future enhancements that can be implemented in the system. The tools used are highly reliable and guarantee case of access, data security and accuracy. The proposal is technically feasible.

3.3.3 OPERATIONAL FEASIBILITY

In operational feasibility, we should check whether the software will be used if it is developed and implemented. We are doing it for Gas Agency and this will be used as the best software because the users gave their complete system requirements and they are satisfied with the software that is developed.

The proposed system will be very useful for the organization since an efficient **Interactive Voice Response System** would contribute for reducing the expenses as well as acquiring good products from well-reputed companies. Thereby the proposed system is highly operable.

3.3.4 JUSTIFICATION

The project has the minimal software and hardware requirements making it technically feasible. It can also be implemented in the organization with the guarantee of satisfying the user needs. The project is highly operable. The above details specify that the project is feasible to all situations.

CHAPTER 4

SYSTEM DESIGN

4.1 MODULAR DESIGN

A Modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of a system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity, that is, software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements. Modularity is the single attribute of software that allows a program to be intellectually manageable. Monolithic software (i.e., a large program comprised of a single module) cannot be easily grasped by reader. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are,

➤ **Modular Decomposability**

If a design method provides a systematic mechanism for decomposing the problem into sub problems, it will reduce the complexity of the overall problem, thereby achieving an effective modular solution.

➤ **Modular Composability**

If a design method enables existing design components to be assembled into a new system, it will yield a modular solution that does not reinvent the wheel.

Modular Understandability

If a module can be understood as a standalone unit, it will be easier to build and easier to change.

Modular Continuity

If small changes to the system requirements result in changes to individual modules, rather than system-wide changes, the impact of change-induced side effects will be minimized.

> Modular Protection

If an aberrant condition occurs within a module and its effects are constrained within that module, the impact of error-induced side effects will be minimized.

4.2 INTERFACE DESIGN

Interface design mainly focuses on the design of interfaces between software modules, external entities and the user. The design of internal program interfaces, sometimes called inter modular interface design, is driven by the data that must flow between modules and the characteristics of the programming language in which the software is to be implemented. External interface design begins with an evaluation of each external entity represented in the DFDs of the analysis model. Both internal and external interface design must be coupled with data validation and error handling algorithms within a module.

Because side effects propagate across program interfaces, it is essential to check all data flowing from module to module to ensure that the data conform to bounds established during requirements analysis. User interface design has as much to do with the study of people as it do with technology issues. Who is the user? How does the user learn to interact with a new computer-based system? So the system should be developed in a user-friendly manner.

FUNCTIONAL INDEPENDENCE

The concept of functional independence is a direct outgrowth of modularity and the concepts of abstraction and information hiding. It is achieved by developing modules with "single-minded" function and an "aversion" to excessive interaction with other modules. Independent modules are easier to maintain because secondary effects caused by the design/code modification are limited, error propagation is reduced, and reusable modules are possible. To summarize, functional independence is a key to good design.

4.4 COHESION

Cohesion is a natural extension of the information hiding concept. A cohesive module performs a single task within a software procedure, requiring little interaction with procedures being performed in other parts of a program. A module that performs a set of tasks that relate to each other loosely, if at all, is termed coincidentally cohesive. A module that performs tasks that are related logically is logically cohesive. When a module contains tasks that are related by the fact that all must be executed within the same span of time, the module exhibits temporal cohesion. Moderate levels of cohesion are relatively close to one another in the degree of module independence. When processing elements of a module are related and must be executed in a specific order, procedural cohesion exists. When all processing elements concentrate on one area of a data structures, communicational cohesion is present. High cohesion is characterized by a module that performs one distinct procedural task.

DATABASE DESIGN

Oracle is chosen for developing the database. For the optimum design of the database, to have a better response time and data integrity, avoid data redundancy, and for security of the database are some of the reasons behind the choice of Oracle.

4.5.1 NORMALIZATION

First Normal Form

It states that the domain of an attribute must include only atomic value and that the value of any attribute in a tuple must be a single value from the domain of that attribute. Hence 1NF disallows having a set of values or a combination of both as an attribute value for a single tuple. In other words 1NF disallows relations within relations or relations as attribute of tuples. The only attribute values permitted by 1NF are single atomic values. All the tables in the database satisfy the first normal form.

Second Normal Form

The test for 2NF involves testing for functional dependencies whose left hand side attribute are part of the primary key. If the primary key contains a single attribute the test need not be applied at all. A relation schema R is in 2NF if every non prime attribute A in R is functionally dependent on the primary key of R . All the tables with two attributes in the primary key field satisfy second normal form.

Third Normal Form

A functional dependency $X \rightarrow Y$ in a relation schema R is a transitive dependency if there is neither a set of attributes Z that is neither a candidate key nor a subset of any key of R and both $X \rightarrow Z$ and $Z \rightarrow Y$ hold.

4.6 I/P AND O/P DESIGN

INPUT DESIGN

Input Design is the process of converting the user originated inputs to a computer based format. The design decisions for handling input specify how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention and it includes specifying the means by which actions are to be taken.

A system user interfacing through a workstation must be able to tell the system Whether to accept input, produce a report, or end the processing. The collection of input data is regarded to be the most expensive part of the overall system design. Since the inputs have to be planned in such a manner so as to get the relevant information, extreme care is taken to obtain the pertinent information. If the data fed into the system is incorrect, then the processing as well as the outputs will magnify these errors .The arrangement of messages and comments in online conversations, as well as, the placement of data, headings, and titles on display screens or source documents, is also part of the input design. An improper design that leaves the screen blank will confuse a user about what action to be taken next. Online systems include a dialogue or conversation between the user and the system. The input design consists of developing specifications and procedures for data preparation, those steps necessary to put transaction data into usable form for processing and data entry, the activity of putting the data into the computer for processing.

The objectives of input design are

- i) Controlling the amount of design.
- ii) Avoiding delay.
- iii) Controlling errors.
- iv) Keeping the steps simple.

Controlling the Amount of Input

The major reasons for controlling the amount of input are, first, as the cost of labor is high, the cost of preparing the data and entering the data is also high. Second, the input phase of computing can be a slow process that can take many times longer than the time needed by computers to carry out their tasks. By reducing input requirements, the analyst can speed the entire process from data capture to processing to providing results to users.

Avoiding Delay

A processing delay results from data preparation or data entry operations is called a bottleneck. Avoiding bottleneck should always be one objective of the analyst in designing input.

Avoiding Errors

The third objective deals with errors. In one sense, the rate at which errors occur depends on the quantity of data, since the smaller the amount of data to input, the fewer the opportunities for errors. The manner in which the data is entered also affects the occurrence of errors. Another aspect of avoiding error is the need to detect the error when they do occur. These are done using input validation techniques.

Avoiding Extra Steps

When the volume of transaction cannot be reduced, the analyst must be sure the process is as efficient as possible. The analyst must also avoid input designs that cause extra steps.

Keeping the process simple

Simplicity works and it is accepted by user's. In contrast it takes work to get users to accept complex or confusing input designs, and there is no guarantee of success in installing a complex system. In my module Inventory Management System all the above discussed points are given due importance while designing the input.

OUTPUT DESIGN

Output design refers to the results and information that are generated by the system. Output is the main reason for developing the system and it is on the basis of the outputs generated that the usefulness of the application is evaluated. The significant points that have to be considered for the output design are as follows.

- Determine what information to present.
- Decide whether to display, print, or voice the information and select the medium of output.
- Arrange the presentation of information in an acceptable format.

Decide how to distribute the output to intended recipients.

The arrangement of information on a display, or a printed document is termed as a layout. The output design is specified on layout forms, sheets that describe the location characteristics and the format of the column headings and pagination. Output design phase of system is concerned with the convergence of information to the end user in user-friendly manner. The output design should be efficient, intelligible so that the system relationship with the end user is improved considerably and thereby, enhancing the decision-making process. The contents of the output are then defined in a detailed manner during the physical design of outputs. Certain data are edited in a desirable manner.

4.7 DATA FLOW DIAGRAM

This is one of the methods to show the flow of data within a module. The data flow represented by means of the arrow with the indication of the direction of flow with the help of the arrow pointer.

Gas Booking Process:

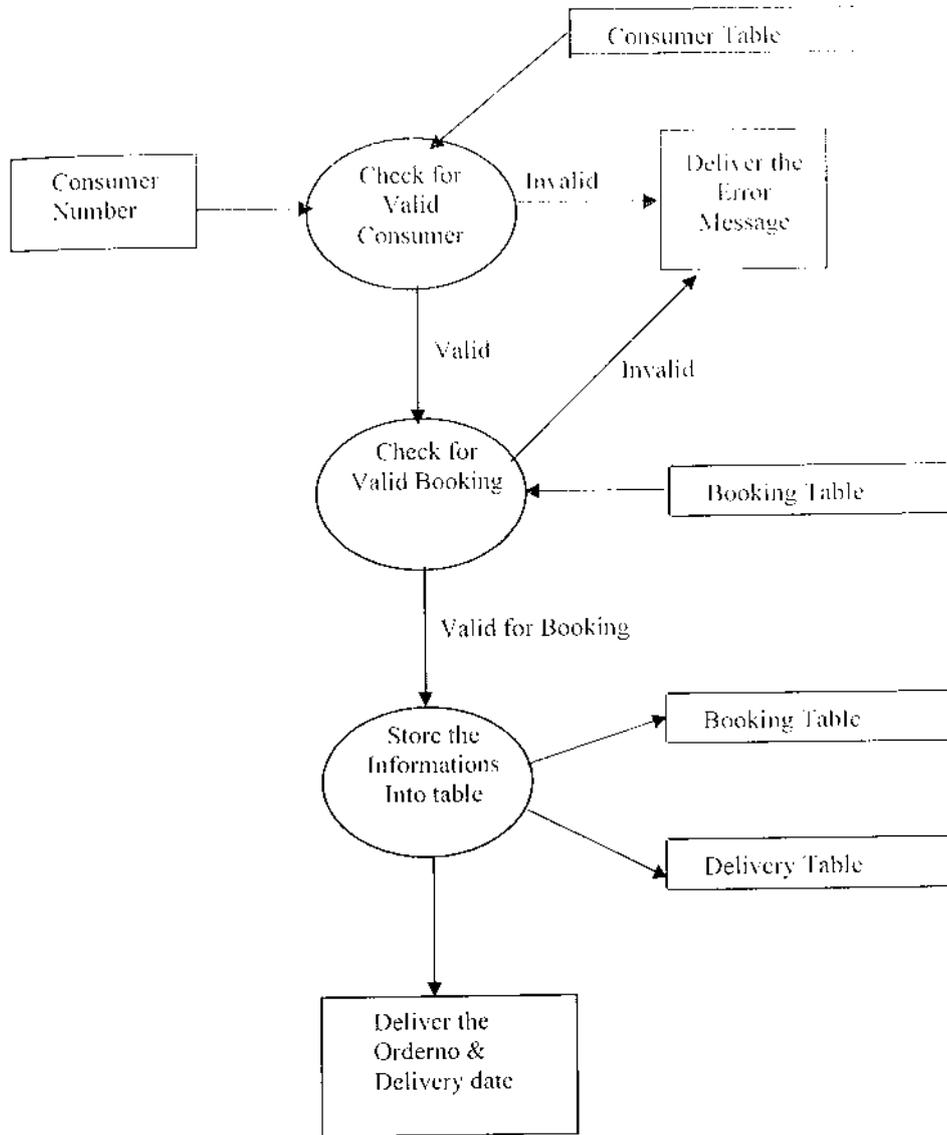


Figure 4.7.1 Gas Booking Process

Complaint Booking Process:

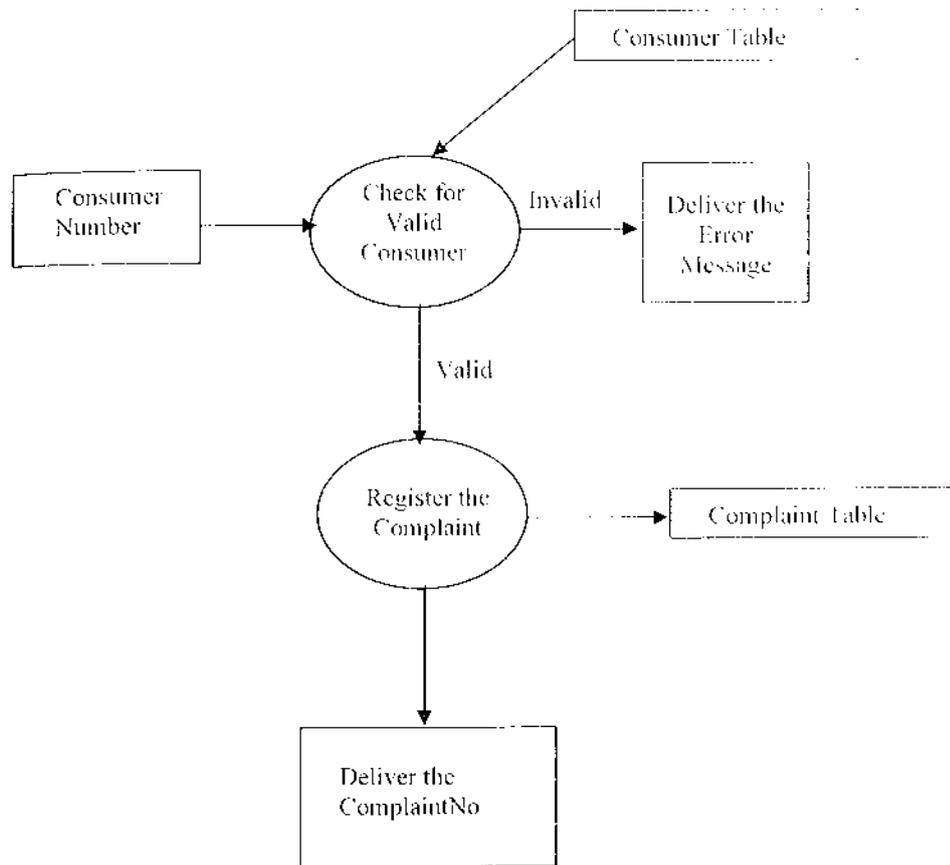


Figure 4.7.2 : Complaint Booking Process

Booking & Delivery Status Process:

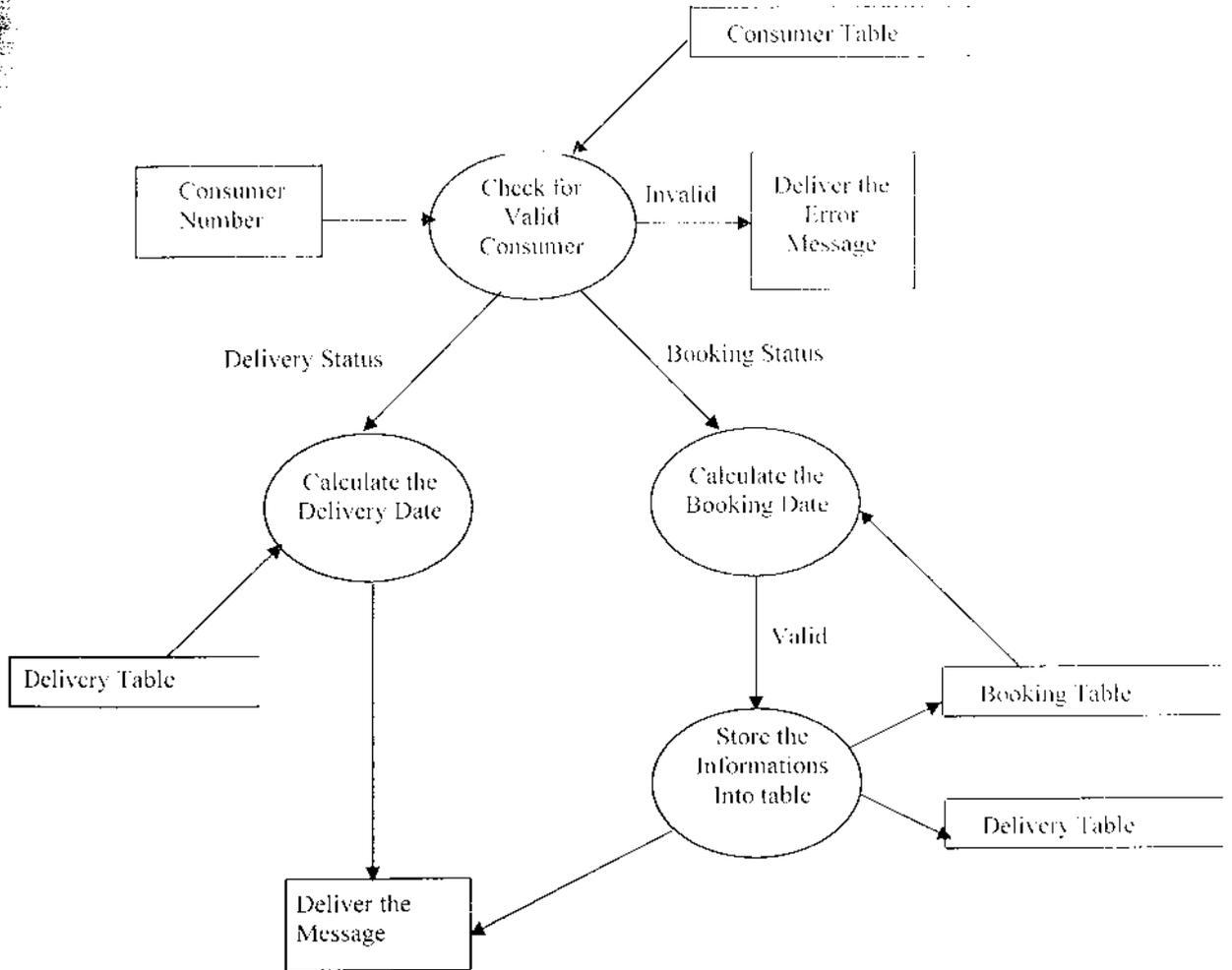


Figure 4.7.3 Booking & Delivery Status Process

Overall Process of IVRS:

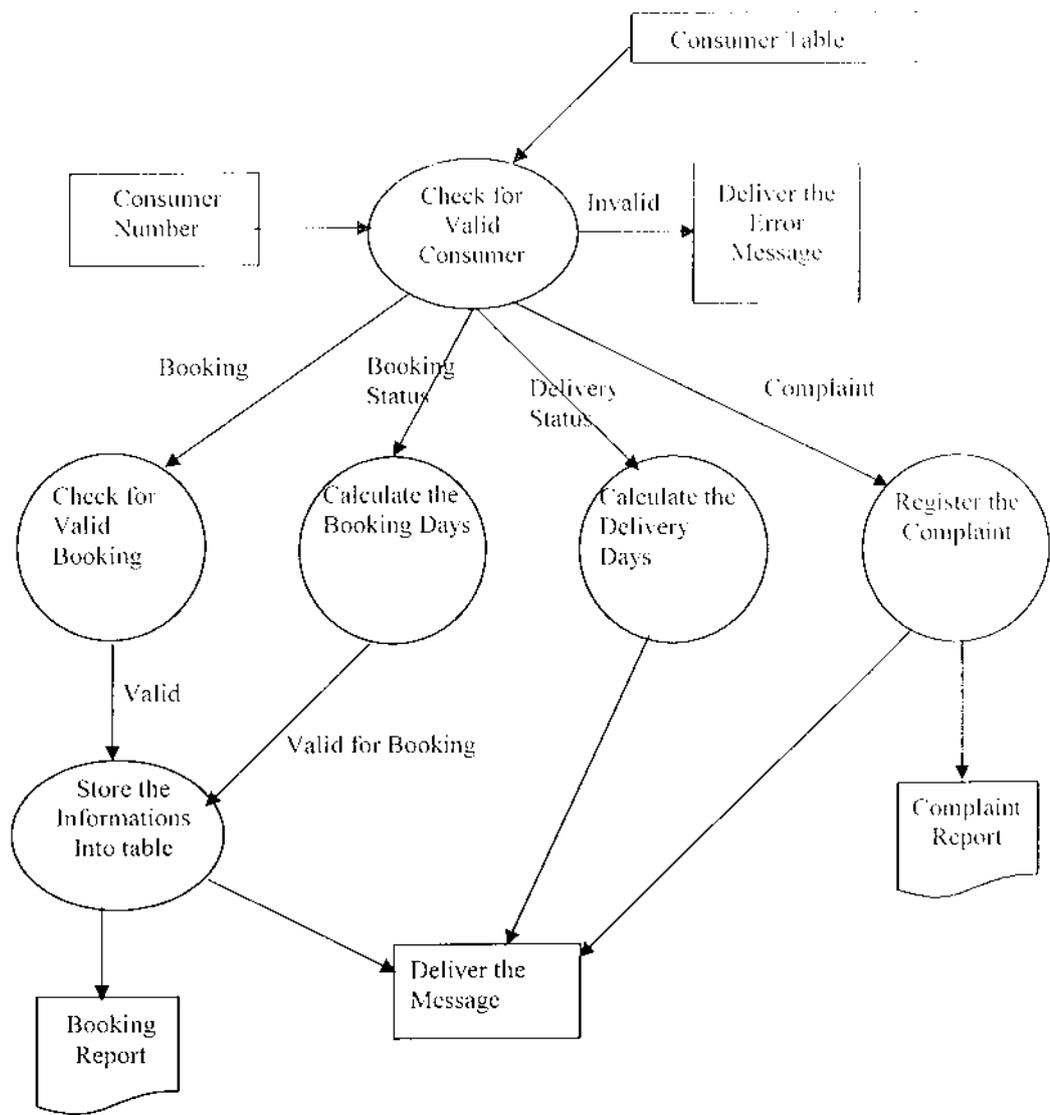


Figure 4.7.4 Overall Process of IVRS

4.8 DATA DICTIONARY

Table 4.8.1 ConsumerDetails

FIELD NAME	DATA TYPES	SIZE
ConsumerNo	NUMBER	5
ConsumerName	VARCHAR	25
ComsumerAddress	VARCHAR	50
NoCylinder	NUMBER	1
ConsumerType	VARCHAR	15

Table 4.8.2 : BookingDetails

FIELD NAME	DATA TYPES	SIZE
ConsumerNo	NUMBER	5
LastBookingDate	DATE	
RecenetBookingDate	DATE	

Table 4.8.3: DeliveryDetails

FIELD NAME	DATA TYPES	SIZE
ConsumerNo	NUMBER	5
OrderNo	NUMBER	5
BookingDate	DATE	
DeliveryDate	DATE	

Table 4.8.4 : ComplaintDetails

FIELD NAME	DATA TYPES	SIZE
ConsumerNo	NUMBER	5
ComplaintNo	NUMBER	5
ComplaintDate	DATE	

CHAPTER 5

TESTING AND IMPLEMENTATION

5.1 TESTING

Testing Objectives

- Testing is the process of executing a program with the intent of finding an error.
- A good test case is one that has finding an undiscovered error.
- A successful test is one that uncovers an error.

Unit Testing

This test focuses on each module individually, (i.e.) each module is tested to ensure that the functionality of the module is proper as a unit. And that it meets the user requirements.

Integration Testing

Integration testing addresses the issue associated with the dual problems of verification. After the software has been integrated a set of tests are conducted to check if the system is working well as a group.

The following are the types of Integration Testing:

1. Top-down testing.
2. Bottom-Up testing.

Top-Down Testing

This method is an incremental approach to the construction of the program structure. Modules are integrated by moving downward through the control hierarchy, beginning with program module. The module subordinates to the main program are incorporated into the structure in either a depth first or breath first manner.

Bottom Up Testing

This method begins with the construction and testing with the modules at the lowest level in the program structure. The bottom-up integration strategy may be implemented with the following steps:

The low-level modules are combined into clusters into clusters that perform a specific software sub-function.

Validation Testing

At the end of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and correction testing begin.

Validation Test Criteria

Software Testing and validation is achieved through series of black box tests that demonstration conformity with the requirements. A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both, the plan and procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are meet.

System Testing

System testing is series of different tests whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all the work should verify that all system elements have been properly integrated and perform allocation function.

5.2 IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing manual or automated system, or it may be a major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system, as well as improper installation will prevent any improvement.

The implementation phase involves the following tasks:

- Careful Planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of staff in the changeover phase.
- Evaluation of changeover.

CHAPTER 6

CONCLUSION AND FURTHER SCOPE

6.1 CONCLUSION

With the development of Body Scanning System, the purpose of the system required is achieved. This software has received positive response from the user department. Since no computerized monitoring system was available for body scanning system, the newly developed system has improved the functioning of the hospital. The goals that have been achieved by the developed system are

- Faster and is proved to be user- friendly.
- Requirements are processed immediately.
- Portable and flexible for further development.

6.2 FURTHER SCOPE

The below mentioned are few of the suggestions for further work in this development software:

- More customized reports can be programmed instead of the pre-defined reports.
- The speech application or, the voice-enabled features can be used to make the software even user-friendlier.

CHAPTER 6

CONCLUSION AND FURTHER SCOPE

6.1 CONCLUSION

With the development of Body Scanning System, the purpose of the system required is achieved. This software has received positive response from the user department. Since no computerized monitoring system was available for body scanning system, the newly developed system has improved the functioning of the hospital. The goals that have been achieved by the developed system are

- Faster and is proved to be user- friendly.
- Requirements are processed immediately.
- Portable and flexible for further development.

6.2 FURTHER SCOPE

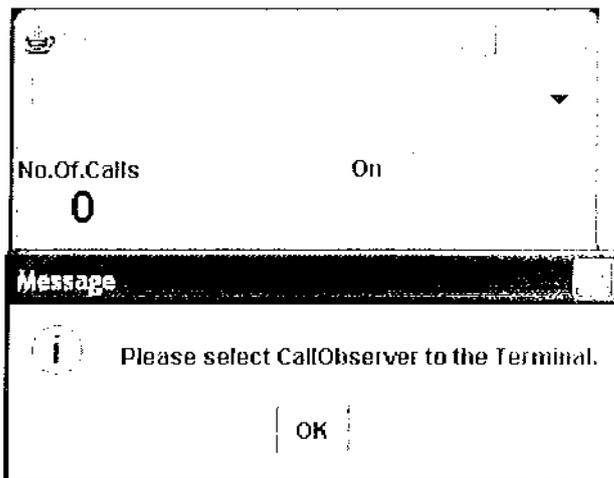
The below mentioned are few of the suggestions for further work in this development software:

- More customized reports can be programmed instead of the pre-defined reports.
- The speech application or, the voice-enabled features can be used to make the software even user-friendlier.

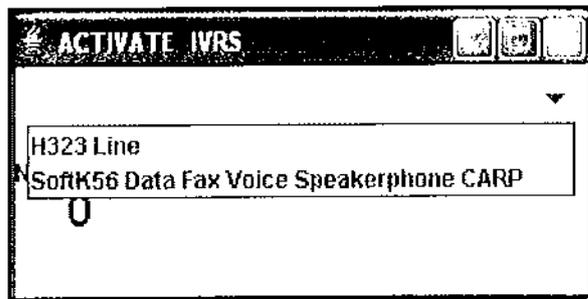
APPENDICES

SCREEN SHOTS

IVRS Activate Screen:



Terminal Selection Screen



Data Voice Speakerphone Terminal Screen

 SoftK56 Data Fax Voice Speakerphone CARP

No.Of.Calls	On
0	

Message

 Selected Terminal is : SoftK56 Data Fax Voice Speakerphone CARP

| OK |

H323 Terminal Selection Screen

The screenshot shows a terminal selection interface. At the top, there is a header bar with a logo on the left and a title 'H323 Terminal Selection Screen' in the center. Below the header, the text 'H323 Line' is displayed with a small downward arrow to its right. Underneath, the text 'No.Of.Calls' is followed by the number '0', and the text 'On' is positioned to the right. A dark message box is overlaid on the bottom half of the screen. The message box has a title bar that says 'Message' and contains an information icon (a lowercase 'i' in a circle) followed by the text 'Selected Terminal is : H323 Line'. At the bottom of the message box, there is an 'OK' button.

No Of Calls Received screen

ACTIVATE IVRS	
SoftK56 Data Fax Voice Speakerphone CARP ▼	
No.Of.Calls	Off
0	

Receive Calls from Telephone Screen

```
Project>java JAnsweringMachine2  
pressed Buttons: 9  
pressed Buttons: 0  
pressed Buttons: 5  
pressed Buttons: 4  
  
Consumer Number From Phone:9054  
Invalid Cosumer Number
```

ACTIVATE IVRS

SofIK56 Data Fax Voice Speakerphone CARP

No.Of.Calls	Off
1	

After Receiving Calls Screen

ACTIVATE IVRS	
SoftK56 Data Fax Voice Speakerphone CARP ▾	
No.Of.Calls	<input type="checkbox"/> Off
1	

Daily Report Screen

Report *Daily Report Form*

Date: 12/23/05
Time: 1:01 PM

Consumer No	Consumer Name	Consumer Address	Reading Date	Inventory Date	Order No
1001 1002	Mahesh Kumar	12, Appachi Nagar 49, Konesi Nagar	12/23/05 11/1/05	7/27/05 10/24/05	5071 1100

REFERENCES

Books

- [1]. Herbert Schildt - "The Complete Reference Java 2", Tata Mc-Graw Hill, 2000.
- [2]. JDK 1.5 Documentation.
- [3]. Alexis Leon and Mathews Leon - "SQL A Complete Reference ",
Tata Mc-Graw Hill , 1999.

URL

<http://www.sun.com>
<http://www.microsoft.com>
<http://www.ibm.com>
<http://www.sun.java.com/swing>
<http://www.sun.java.forum.com>