

DISSERTATION



(A JAVA TECHNOLOGY PRODUCT)

*Submitted in partial fulfillment of the
Requirements for the award of the degree*

*Of
Bachelor of Technology
In
Computer Science & Engineering*

*By
Amanpreet Singh Chauhan(06/CSE1/2009)
Gagandeep Singh(04/CSE1/2009)
Hargeet Kaur (03/CSE1/2009)*

*Under the guidance of:
Ms. Rekha Bhatia*



**Department of Computer Science & Engineering
Guru Tegh Bahadur Institute of Technology
Guru Gobind Singh Indraprastha University
Dwarka, New Delhi
Session 2012-2013**

Sales Process Analyzer

DISSERTATION

*Submitted in partial fulfillment of the
Requirements for the award of the degree*

Of

Bachelor of Technology

In

Computer Science & Engineering

By

Amanpreet Singh Chauhan (06/CSE1/2009)

Gagandeep Singh (04/CSE1/2009)

Hargeet Kaur (03/CSE1/2009)

Under the guidance of:

Ms. Rekha Bhatia



**Department of Computer Science & Engineering
Guru Tegh Bahadur Institute of Technology**

**Guru Gobind Singh Indraprastha University
Dwarka, New Delhi
Session 2012-2013**

DECLARATION

We hereby declare that all the work presented in the dissertation entitled “**Sales Process Analyzer**” in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering**, Guru Tegh Bahadur Institute of Technology, affiliated to Guru Gobind Singh Indraprastha University Delhi is an authentic record of our own work carried out under the guidance of **Ms. Rekha Bhatia**.

Date:

Amanpreet Singh Chauhan(06/CSE1/2009)

Gagandeep Singh(04/CSE1/2009)

Hargeet Kaur(03/CSE1/2009)

CERTIFICATE

This is to certify that dissertation entitled “**Sales Process Analyzer**”, which is submitted by **Mr. Amanpreet Singh Chauhan, Mr. Gagandeep Singh and Ms. Hargeet Kaur** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering**, Guru Tegh Bahadur Institute of Technology, New Delhi is an authentic record of the candidate’s own work carried out by them under our guidance. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Rekha Bhatia

(Project Guide)

Munshi Yadav

(Head of Department)

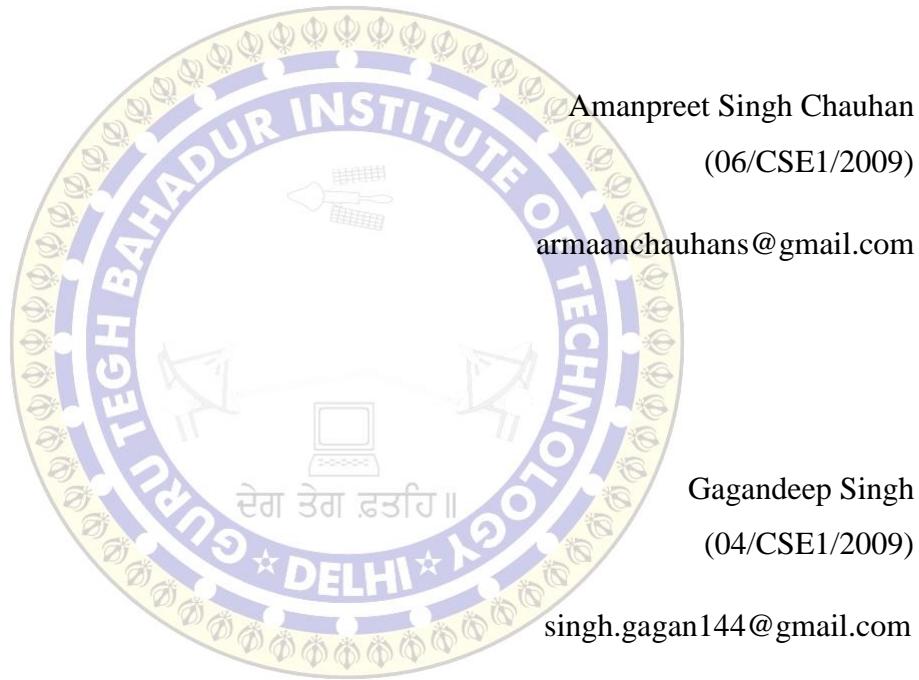
Computer Science & Engineering

Date:

ACKNOWLEDGEMENT

We would like to express our great gratitude towards our supervisor, **Ms. Rekha Bhatia** who has given us support and suggestions. Without their help we could not have presented this dissertation upto the present standard. We also take this opportunity to give thanks to all others who gave us support for the project or in other aspects of our study at Guru Tegh Bahadur Institute of Technology.

Date:



Amanpreet Singh Chauhan
(06/CSE1/2009)

armaanchauhans@gmail.com

Gagandeep Singh
(04/CSE1/2009)

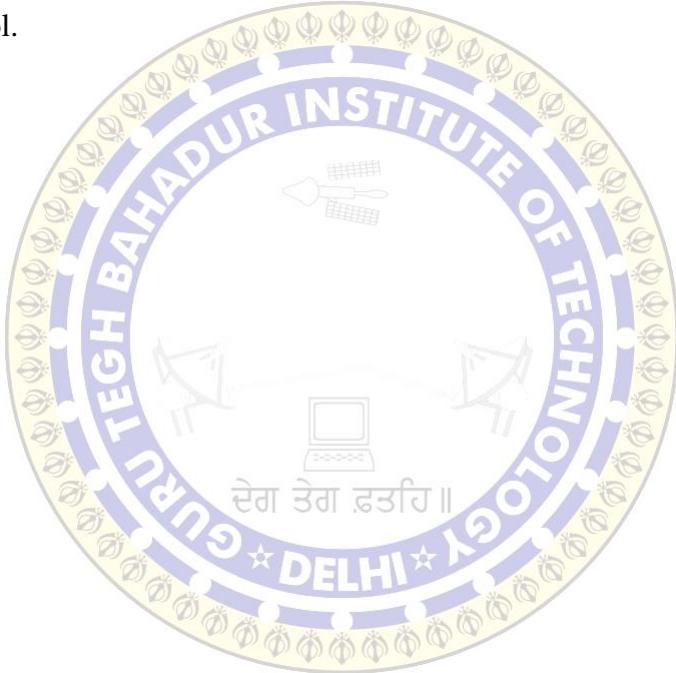
singh.gagan144@gmail.com

Hargeet Kaur
(03/CSE1/2009)

hargeet.kaur13@gmail.com

ABSTRACT

Sales Process Analyzer is a sales management tool that enables you to identify weak areas in your sales process and have more qualified prospects, leads and more sales conversions. Sales Process Management is imperative for the SMB organizations as it is for the larger ones, to take the former's business to the next level. Sales Process Management is a market proven business management technique that provides a great forecasting ability for forward looking business. The analyzer tool takes inputs based on number data & choice selection and based on logics & algorithms generates a report that identifies areas of concern for the party that takes test on the analyzing tool.



LIST OF FIGURES

Fig No	Figure Name	Page
4.1	Entity Relationship Diagram	20
4.2	Zero level Data Flow Diagram	25
4.3	One level Data Flow Diagram	26
4.4	Use Case Diagram	27
4.5	Class Diagram	29
4.6	State Chart Diagram	31
4.7	Activity Diagram of Booting Activity	33
4.8	Activity Diagram of Diagnose Activity	35
4.9	Activity Diagram of Database Checking Activity	38
4.10	Activity Diagram of Log Checking Activity	40
4.11	Deployment Diagram	41
5.1	Black box Testing	45

LIST OF TABLES

Table No	Table Name	Page
3.1	Additional Software Required	12
3.2	List of input data items	14
3.3	List of output data items	16
4.1	Database Schema	23
5.1	Functional Testing	45
5.2	OS and Processor compatibility	51
5.3	Database Connectivity compatibility	52

CONTENTS

Chapter	Page No.
Title Page	i
Declaration	ii
Certificate	ii
Acknowledgement	iii
Abstract	iv
List Of Figures	v
List Of Tables	vi
1. Introduction	2
1.1 Project Introduction	2
1.1.1 Aim	2
1.1.2 Objective	2
1.1.3 Purpose, Scope and Applicability	2
1.1.3.1 Purpose	3
1.1.3.2 Scope	3
1.1.3.3 Applicability	3
1.1.4 Features	4
2. Technology Used	6
2.1. Programming Language Used	6
2.2. Environment Used	7
2.2.1. Net Beans IDE	7
2.2.2. Adobe Photoshop CS5	7
2.2.3. Corel Draw X5	8
3. Software Requirement Specification	10
3.1 Introduction	10

3.1.1 Purpose	10
3.1.2 Scope	11
3.1.3 References	11
3.1.4 Overview	11
3.2 Overall Description	11
3.2.1 Application Perspective	12
3.2.1.1 Interfaces	12
3.2.1.2 Hardware Interface	12
3.2.1.3 Software Interface	12
3.2.1.4 Site Adaptation Requirements	13
3.2.2 Application Functions	13
3.2.3 User Characteristics	14
3.3 Specific Requirements	14
3.3.1 Non Functional Requirements	14
3.3.2 Functional Requirements	16
3.3.2.1 Validity Checks on the Inputs	16
3.3.3 Application Attributes	17
3.3.3.1 Reliability	17
3.3.3.2 Security	17
3.3.3.3 Maintainability	17
3.4 Change Management Process	18
3.5 Document Approval	18
4. System Design	20
4.1 Entity Relationship Diagram	20
4.2 Database schema	23
4.3 Data Flow Diagram	25
4.3.1 Zero Level Data Flow Diagram	25
4.3.2 First Level Data Flow Diagram	26
4.4 Use Case Diagram	27

4.5 Class Diagram	29
4.6 State Chart Diagram	31
4.7 Activity Diagram	33
4.7.1 Booting the application	33
4.7.2 Diagnose Algorithm	35
4.7.3 Database Connectivity Check Algorithm	37
4.7.4 Log Check Algorithm	39
4.8 Deployment Diagram	41
5. Testing	44
5.1 Functional Testing (Black Box Testing)	44
5.2 Non-Functional Testing	51
5.2.1 Compatibility Testing	51
5.2.2 Recovery Testing	52
5.2.3 Security Testing	53
6. Limitation and Future Scope	56
5.1 Limitations	56
5.2 Future Scope	56
References	59
Appendix A : Screen Shots	63
Appendix B : Source Code	76

INTRODUCTION

1.1 PROJECT INTRODUCTION

1.1.1 AIM

To create an analytical tool that will help the business owners to feed information and see as to where their organizations are lacking in Sales Process.

1.1.2 OBJECTIVE

Sales Process is an intelligent step by step waterfall model that any organization must follow to close a sale. Successful companies follow sales processes in totality and have a major market share in their product space. Based on sales analysis technique, SPA - Sales Process Analyzer is a sales management tool that enables you to identify weak areas in your sales process and have more qualified prospects, leads and more sales conversions

1.1.3 PURPOSE, SCOPE and APPLICABILITY

This document aims at defining the overall software requirements for the project “**Sales Process Analyzer**”. Efforts have been made to define the requirements exhaustively and accurately. The final product will have features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing this product. In case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/or will be produced.

1.1.3.1 PURPOSE

This specification document describes the capabilities that will be provided by the System. It also states the various required constraints by which the system will abide.

A business company will have monthly, quarterly and annual targets, and **Sales Process Analyzer** tells the company whether its sales force is on track.

SPA also enables the company to plan and prioritize future sales strategies and predict their results. It also analyzes past performance against targets, track the effectiveness of marketing campaigns, forecast future trends, develop sales strategies and calculate sales commissions.

We have built this project using programming language JAVA.

1.1.3.2 SCOPE

The system once functional will give users a platform to enter their past records and details as asked by the system, store the information in the database, process the sales and analyze the results. Users are enabled to derive an accurate picture of the performance of a business using available data, after integrating all the data required to assess sales.

The system built is user friendly and is highly interactive.

1.1.3.3 APPLICABILITY

Sales Process Analyzer helps a business detect trends. It is imperative to know which trends are declining and which ones are on the rise. Such analysis allows a business to forecast its sales in every given market segment at a given time.

Sales Process Analyzer also helps a business come up with a sales strategy by identifying customers through demographics, for instance. Sales analysis also helps a business know which products are selling and which are not.

1.1.4 FEATURES.

- The basic aim of the application is to take set of inputs from the user regarding sales process and company's past performance and then generates strategic information.
- The application then takes set of inputs that include personal information, company's past performance and a questioner that consist of twelve multiple choice questions to understand sales process in the client's organization.
- Next, the application applies an algorithm based on certain business dimensions and facts to process all inputs.
- It then generates output in form of graphs and text specifying weak areas and suggestions to improve them.
- The application also generates a PDF file containing all discussion.
- The application is supported by a primary database that stores information of clients consulted which later can be used for future reference. The user can select data and load it into the application for analysis.
- On the other hand, the application is also be powered by a secondary database. This database will store information in form of binary files and will be used in case of database connectivity problem. The data then can be transferred to primary database once the problem has been resolved.
- Moreover, the application will be accessed through a secured login. The user must define a password during configuration however, it can be changed later.
- A smart feature of the application includes an 'Exception Handler' that will describe any error occurred during execution and provide solution to rectify them.

TECHNOLOGY USED

2.1 PROGRAMMING LANGUAGE USED

Java is a small, simple, safe, object oriented, interpreted or dynamically optimized, byte coded, architectural, garbage collected, multithreaded programming language with a strongly typed exception-handling for writing distributed and dynamically extensible programs. Java is a high-level, third generation language like C, FORTRAN, Small talk, Pearl and many others.

The following features make it one of the best programming language:

- It is simple and object oriented
- It helps to create user friendly interfaces.
- It is very dynamic.
- It supports multithreading.
- It is platform independent
- It is highly secure and robust.

We have used the following features of Java in sales Process Analyzer:

- **Swings:** Swing is the primary Java GUI widget toolkit. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform.
- **Threads:** A thread is an independent path of execution within a program. Many threads can run concurrently within a program. Every thread in Java is created and controlled by the `java.lang.Thread` class.
- **JDBC:** This technology is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases. A

JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the JVM host environment.

- **File Handling:** An I/O Stream represents an input source or an output destination. A stream can represent many different kinds of sources and destinations, disk files, devices, other programs, etc. Streams support many different kinds of data simple bytes, primitive data types, localized characters, and objects. Data is transferred to devices by streams.

2.2 ENVIRONMENT USED

2.2.1 NETBEANS IDE

Net Beans refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Groovy, C, C++, Scala, Clojure, and others.

The Net Beans IDE is written in Java and can run anywhere a compatible JVM is installed, including Windows, Mac OS, Linux, and Solaris. A JDK is required for Java development functionality, but is not required for development in other programming languages.

The Net Beans platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the Net Beans platform (including the Net Beans IDE) can be extended by third party developers.

2.2.2 ADOBE PHOTOSHOP CS5

Adobe Photoshop is a graphics editing program developed and published by Adobe Systems Incorporated. Adobe Photoshop is released in two editions: **Adobe Photoshop**, and **Adobe Photoshop Extended**, with the Extended having extra 3D image creation, motion graphics editing, and advanced image analysis features. Adobe Photoshop

Extended is included in all of Adobe's Creative Suite offerings except Design Standard, which includes the Adobe Photoshop edition.

Photoshop has ties with other Adobe software for media editing, animation, and authoring.

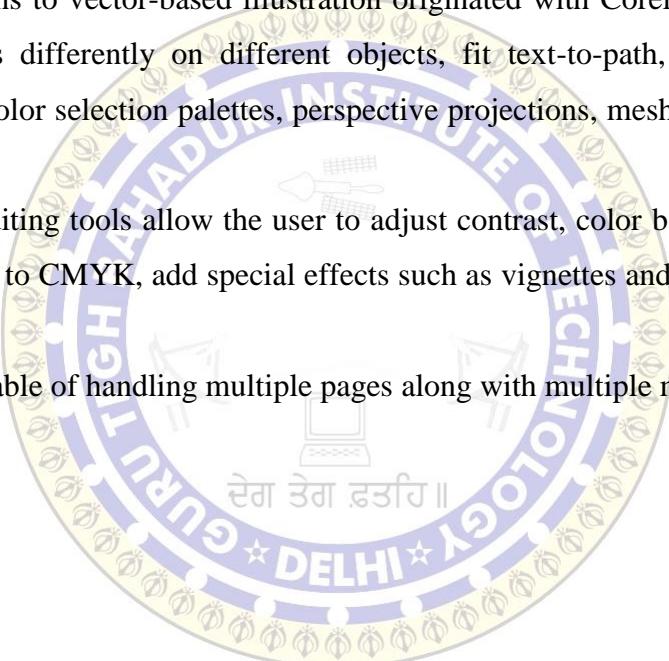
2.2.3 COREL DRAW X5

CorelDraw is a vector graphics editor developed and marketed by Corel Corporation of Ottawa, Canada. It is also the name of Corel's Graphics Suite.

Several innovations to vector-based illustration originated with CorelDraw: a node-edit tool that operates differently on different objects, fit text-to-path, stroke-before-fill, quick fill/stroke color selection palettes, perspective projections, mesh fills and complex gradient fills.

A full range of editing tools allow the user to adjust contrast, color balance, change the format from RGB to CMYK, add special effects such as vignettes and special borders to bitmaps.

CorelDraw is capable of handling multiple pages along with multiple master layers.



SOFTWARE REQUIREMENT SPECIFICATION

A Software Requirement Specification (SRS) describes a complete description of the behavior of a system to be developed. It will include a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS will also contain non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation.

Software Requirements is basically a sub-field of Software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

The **software requirement specification** (SRS) document will enlist all necessary requirements for project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with project team and the customer.

3.1 INTRODUCTION

The aim of the project is to develop a standalone application ‘Sales Process Analyzer’. The application is required to take several inputs in the form of questions to the user regarding the sales figure in his company and generate an output showing sales comparisons over years with respect to clients touched, prospects, leads and overall sales. The project is also required to generate a report for the company which highlights weak areas in the sales process.

The following subsection provides an overview of entire SRS.

3.1.1 PURPOSE: This SRS intended for ‘Sales Process Analyzer’ and its purpose is to provide the detailed description of the application specifying ‘what’ it will do without specifying ‘how’ it will do.

The SRS is intended for the developer as well for the customer who has requested the development of the application. The purpose of this SRS is achieved only if

the user is able to express his views and specification confidently about the application and the developer perceive them correctly.

3.1.2 SCOPE: The application is intended to identify weak areas in sales process and provide suggestions to improve them to have more qualified leads, prospects and sales conversion. The application will require an excel based database connection for storing the sales information of the company on a yearly basis. After analysis in addition to onscreen report, an equivalent PDF report will be generated for the client. Regarding the distribution, it is intended for the requested customer only.

3.1.3 REFERENCES :

- (1) Score Card Template.xls – Excel Document
- (2) Diagnostic Master Ver2.doc – Document file
- (3) SPA Query Analyzer Answers.docx – Document files

3.1.4 OVERVIEW: The remainder of this document is in two chapters, the first providing a full description of the application for the owner. It lists all the functions performed by the system. The final chapter concerns details of each of the application functions and actions in full for the application developers' assistance. These two sections are cross-referenced by topic, to increase understanding by both groups involved.

3.2 OVERALL DESCRIPTION

This section of SRS describes the general factors that may affect the application and its requirements. It does not contain any specific requirement instead it is meant for providing background for those requirements which are discussed in details in later section making them easier to understand.

3.2.1 APPLICATION PERSPECTIVE: The ‘Sales Process Analyzer’ will be an independent and totally self-contained application that does not require development of any additional application. However, it will require the user to explicitly create an ODBC connection and provide DSN name while configuring the application. On the other hand, since the application will be java based, it will require Java Runtime Environment.

The following subsections describe how the application operates inside various constraints.

3.2.1.1 INTERFACES: The interface of the application will be a major focus of the development. Less attention will be required on algorithmic part as compared to Graphical User Interface. The application is intended for any user and is not confined for specific users thus it must be compact, simple and easy to understand so that fresh user quickly adapt to it. Thus as a whole, the application is expected to be extremely user friendly and should not be complex, annoying users rather must create a good impression on their minds.

3.2.1.2 HARDWARE INTERFACE: No typical hardware component will be required to run this application. The stand alone application will be meant to run on any personal computer, laptops or even Notebooks that have preinstalled Java Runtime Environment.

3.2.1.3 SOFTWARE INTERFACE: The following table describes the various additional software requirements :

Table 3.1 – Additional software required

Name	Purpose/ Interface with application	Version Number	Source
Java ™ Platform	Java Platform, Standard	6 or higher	<u>Web Link :</u>

SE or Java Runtime Environment (JRE)	<p>Edition or Java SE is a widely used platform for programming in the Java language. It is the Java Platform used to deploy portable applications for general use. In practical terms, Java SE consists of a virtual machine, which must be used to run Java programs, together with a set of libraries (or packages) needed to allow the use of file systems, networks, graphical interfaces, and so on, from within those programs.</p>	version	<p>www.oracle.com/technetwork/java/javase/downloads/index.html</p> <p><u>Others:</u> If not installed, the browser will automatically download and install Java™ Platform SE. This will however take some time to download and install.</p>
--------------------------------------	--	---------	--

3.2.1.4 SITE ADAPTATION REQUIREMENTS: The most basic and important site adaption requirements include creation of System DSN for database connection through ODBC. The user must explicitly create a system DSN in ‘ODBC Data Source Administrator’ in ‘Administrative Tools’ inside ‘Control Panel’ of Microsoft Windows. This DSN must further be attached to an Excel 97-2003 workbook of extension .xls. Once created, the user must simply provide the DSN during configuration and the application will adapt it to its use automatically.

3.2.2 APPLICATION FUNCTIONS: The function of the application is simple and straight forward. The application will receive set of inputs each for clients touched, sales and revenue in the current as well as the past years. Moreover few

multiple choice questions will be asked to the user whose answers will also serve as an effective input. On the basis of these inputs, the weak points in the sales process will be assessed. The procedure of the calculation is strictly based on the algorithm that will be designed according to excel document. In the end a report will be generated showing all the relevant details.

3.2.3 USER CHARACTERISTICS: The application is strictly intended for the customer who has requested the project.

3.3 SPECIFIC REQUIREMENTS

This section contains all the requirements at a level of detailed sufficient to enable designers to design an application to satisfy those requirements and testers to test that the application satisfies those requirements. Throughout this section, every stated requirement should be perceived by users and operators. These requirements should include at a minimum description of every input into the application, every output from the application and all functions performed by the system in response to an input or in support of an output.

3.3.1 NON FUNCTIONAL REQUIREMENTS: This subsection contains a detailed description of all inputs and outputs for the application :

Table 3.2 – List of input data items

Name of item	Data Type	Valid Range	Source of input	Unit	Description
Person Name	String	–	TextField	–	Name of the Client
Person Email Address	String	Standard Email Address	TextField	–	Email Address of the Person

Person Country	String	—	TextField	—	Country of the Person
Phone Number	Long integer	Standard Phone Number	TextField	—	Phone number of the Person
Person Designation	String	—	TextField	—	Total number of walls to be constructed
Company Name	String	—	TextField	—	Name of the Company
Company Address	Long integer	Standard Fax number	TextField	—	Address of the Company
Date	String	DD/MM/YYYY	Obtained from system date	—	Date of Activation
Expected Client	Integer	—	Table	—	Expected Number of Clients
Actual Client	Integer	—	Table	—	Actual Number of Clients
Expected Revenue	integer	—	Table	^	Expected sales Revenue
Actual Revenue	Integer	—	Table	^	Actual sales Revenue
Clients Touched	integer	—	Table	^	Clients Touched in a Year
Prospects	integer	—	Table	^	Total Prospects in a Year
Leads	integer	—	Table	^	Total Leads in a Year
Sales	integer	—	Table	^	Total Sales in a Year
MCQ Answers	Integer	—	Radio Button	—	Answers to the Question Panel

Table 3.3 – List of output data items

Name of item	Data Type	Source of output	Unit	Description
Prospect List	string	Label	—	Qualified Prospect List
Lead List	string	Label	—	Qualified Lead List
Sales List	string	Label	—	Qualified Sales List
Expected vs. Actual Clients	—	Graph	—	Graph between expected vs. actual Clients
Expected vs. Actual Sales	—	Graph	—	Graph between expected vs. actual sales
Prospect vs. Clients Touched	—	Graph	—	Graph for Clients touched and prospects
Leads vs. Prospects	—	Graph	—	Graph for leads vs. prospects
Sales vs. Leads	—	Graph	—	Graph for sales vs. leads
Sales vs. Clients Touched	—	Graph	—	Graph for sales vs. clients touched

3.3.2 FUNCTIONAL REQUIREMENTS: This subsection defines the fundamental actions that must take place in the application in accepting and processing the inputs and in processing and generating the outputs.

3.3.2.1 VALIDITY CHECKS ON THE INPUTS: The input to the application would be provided using graphical components such as TextFields or Radio

buttons. In case of TextFields, the component takes input in form of String which later may be converted to proper data type as required. In such a case validity check is important before the inputs are processed. The table discussed above provides the necessary conversion required while fetching data from respective TextFields. Java provides an excellent way of converting data types and handling any exception occurring while conversion. However, in case of radio button the selected button from the group corresponds to an integer which is recognized by the algorithm as to which option was selected.

3.3.3 APPLICATION ATTRIBUTES: There are a number of quality attributes of the application that can serve as requirements for instance:

3.3.3.1 RELIABILITY: Once on live, the application must be extremely reliable. It must produce correct results independent of time. There must be no scope of user to input invalid data causing abnormal behavior. Moreover, any error occurring in database connection must immediately prompt.

3.3.3.2 SECURITY: The application must be designed by taking into account the security aspects of the application. Efforts must be taken to make the application as secure as possible, eliminating any possibility of unauthorized copying and editing. Moreover, a secure login must be included to prevent any unauthorized access.

3.3.3.3 MAINTAINABILITY: No maintenance is required if the product is not disturbed by the customer i.e. if all files are intact with the application in the similar way it was delivered.

3.4 CHANGE MANAGEMENT PROCESS

The application must be designed to be flexible enough to adapt normal changes in specification such as little changes in GUI layout, changes in investment heads etc. However, it must be noted that only minor changes will be acceptable. Change in algorithm or addition of any other sales process aspect will cause complete change in structure of the project. Non functional as well as functional part of the SRS has to be modified. This will consequently result in redesign of the algorithm as well as the layout of the application which may cause project to deviate from its intended schedule.

3.5. DOCUMENT APPROVAL

This is to declare that all the descriptions, requirements and specification stated in this SRS have been discussed and approved by both the customer and the developer. The customer agrees that the above state description is his expectation from the application and will satisfy his demand. On the other hand, the developer promises to perceive all the information stated and develop the application to its utmost accuracy and deliver it on time.

Moreover, the customer also agrees that unauthorized Copying, Reproduction, Rental, Broadcasting of this application is violation of applicable law. This application will be intended solely for customer's own organization. The customer may not decompile, reverse engineer, or disassemble the application, except as permitted by law.

SYSTEM DESIGN

4.1 ENTITY RELATIONSHIP (ER) DIAGRAM

An Entity – Relationship model (ER model for short) is an abstract way to describe a database. It usually starts with a relational database, which stores data in tables. The E-R model for Sales Process Analyzer is as follows:

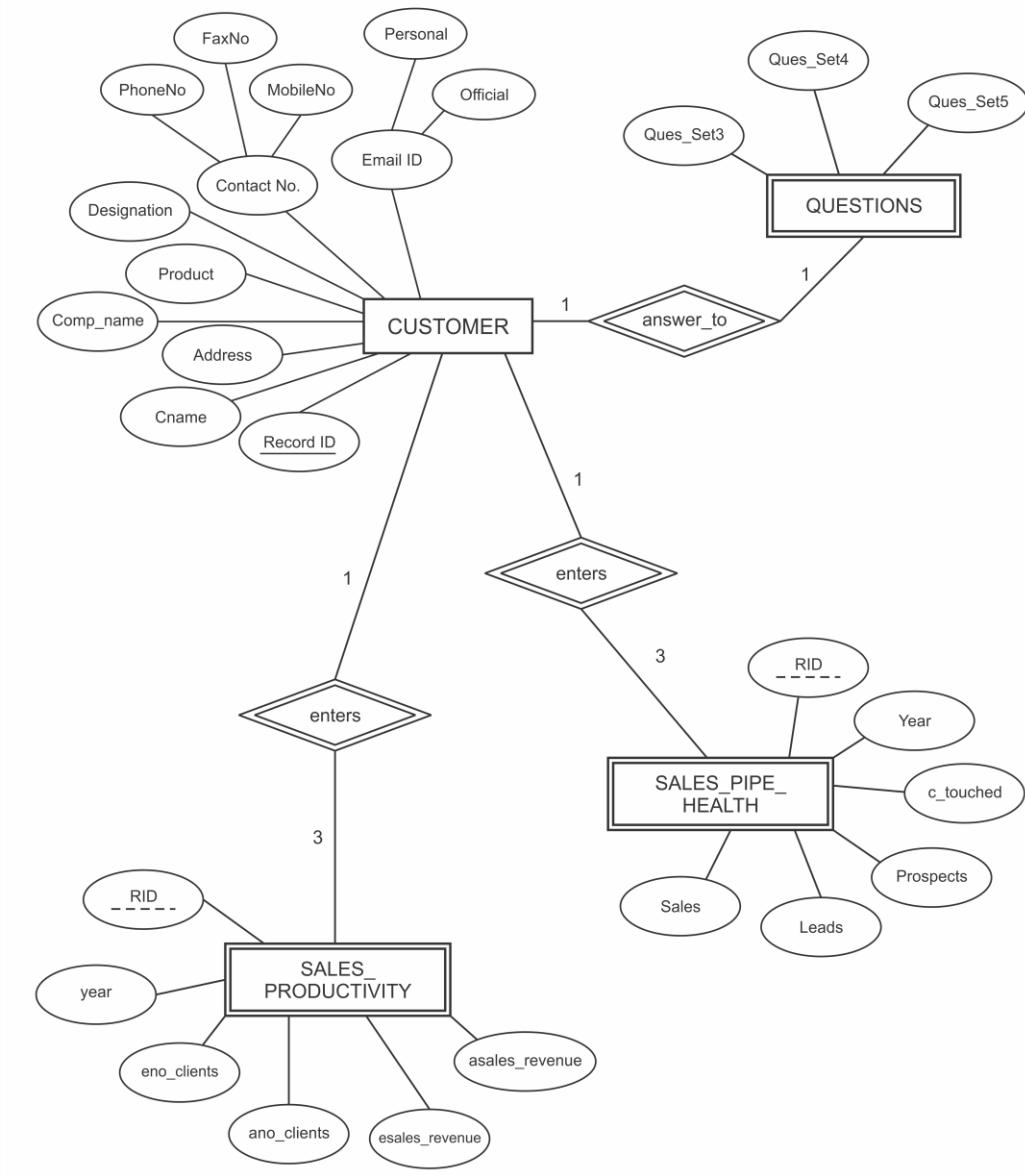


Fig 4.1 – Entity-Relationship Diagram

Legends :

- c_name: client's/customer's name
- comp_name: company's name
- ContactNo: contact number
- phone: phone number
- fax: fax number
- mobile: mobile number
- Ques_set1: Question Set 1
- Ques_set2: Question Set 2
- Ques_set3: Question Set 3
- RID: Record ID
- eno_clients: expected number of clients
- ano_clients: actual number of clients
- esales_revenue: expected sales revenue
- asales_revenue: actual sales revenue
- sales_info: sales information
- c_touched: clients touched

The above ER diagram is explained as follows:

- The entity '**Customer**' contains attributes such as Record ID (primary key), c_name, Address, comp_name, Product, Designation, Contact No, Email ID.

- Contact No. is a composite attribute containing attributes such as phone, fax and mobile.
- Email ID is also composite having personal as well as official email id.
- The entities '**Questions**', '**Sales_productivity**' and '**Sales_info**' are weak entities depending upon its parent entity '**Customer**'.
- Customer '**answers_to**' Questions. For each record in 'Customer' there is exactly 1 record in 'Questions'
- Customer '**enters**' Sales_productivity and Sales_info. For each record in 'Customer' there are exactly 3 records in 'Sales_productivity' and 'Sales_info'.
- The entity '**Questions**' contains attributes such as Ques_set1, Ques_set2 and Ques_set3.
- The entity '**Sales_productivity**' contains attributes such as RID (foreign key references primary key RecordID in 'Customer'), year, ano_clients, eno_clients, asales_revenue and esales_revenue.
- The entity '**Sales_info**' contains attributes such as RID (foreign key references primary key RecordID in 'Customer'), year, c_touched, prospects, leads and sales.

4.2 DATABASE SCHEMA

The tables and the attributes present are as follows:

Table 4.1 – Database Schema

Table Name	Attributes Present	Attribute Type
Data	CMP_NAME DESIG PRODUCT PHONE_NO MOBILE_NO FAX_NO EMAIL_OFF EMAIL_PER R_DATE QUES3_1 QUES3_2 QUES3_3 QUES4_1 QUES4_2 QUES4_3 QUES4_4 QUES4_5 QUES5_1 QUES5_2 QUES5_3 QUES5_4	Varchar Varchar Varchar Number Number Number Varchar Varchar Date Integer Integer Integer Integer Integer Integer Integer Integer Integer Integer Integer Integer Integer Integer Integer
SaleProd	RID YEAR EXP_NO_CL	Integer Number Integer

	ACT_NO_CL	Integer
	EXP_SL_REV	Integer
	ACT_SL_REV	Integer
SalePipeH	R_ID	Integer
	YR	Number
	CT	Integer
	PROSPECTSLEADS	Integer
	SALES	Integer

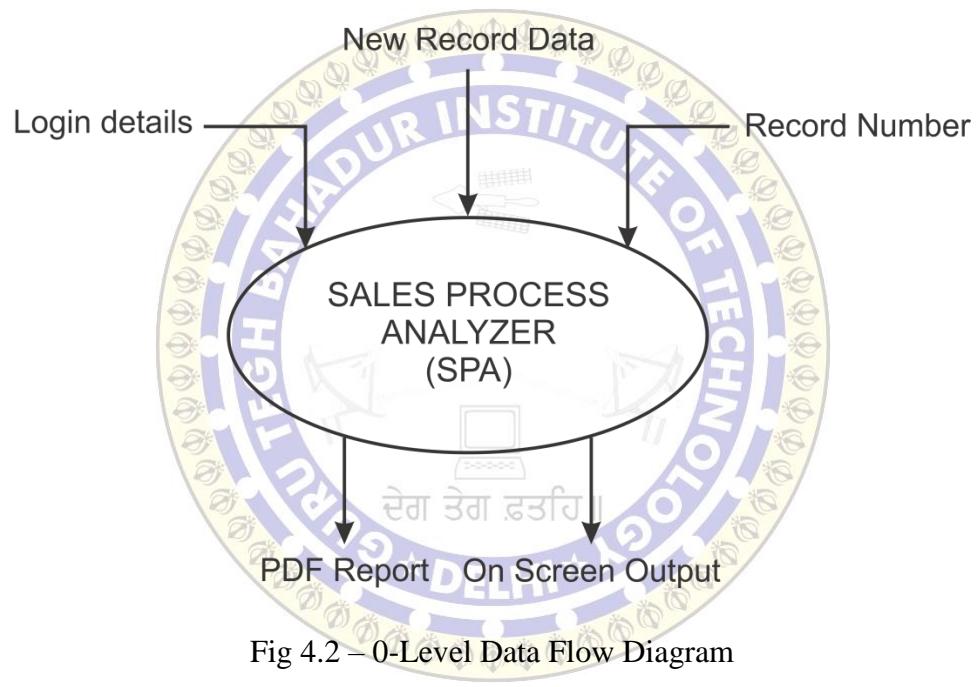
THE ABOVE SCHEMA HAS BEEN NORMALIZED TO 3 - NORMAL FORM.



4.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. The various levels DFDs of the Sales Process Analyzer are as follows:

4.3.1 ZERO LEVEL DATA FLOW DIAGRAM



In Sales Process Analyzer, the information flow is as follows:

- The login details are filled in the initial step to the application.
- The new record is then fed into the analyzer.
- Also the user may enter a record number to load data from database.
- Finally the analyzer generates a PDF report and an on-screen output.

4.3.2 ONE LEVEL DATA FLOW DIAGRAM

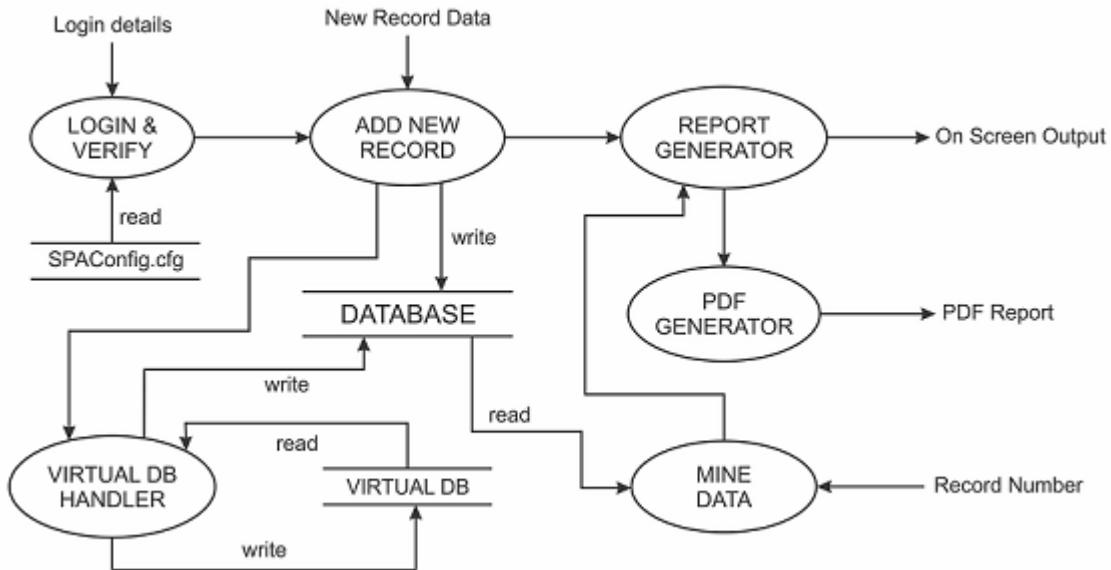


Fig 4.3 – 1-Level Data Flow Diagram

The detailed flow of data in Sales Process Analyzer is as follows:

- The login details are inputted into the Login and Verification module which verifies its data from the configuration file ‘SPAConfig.cfg’.
- A new record is added into the database with user inputs and is send to ‘Report Generator’ module for generating report. In case the database connection is lost the data is send to ‘Virtual Database Handler’ to be stored in virtual database in form of binary files.
- The ‘Report Generator’ produces an on screen output as well as leads the ‘PDF Generator’ to generate a PDF report.
- The data can also be mined from the database using the record number and is send to the ‘Report Generator’ for analysis.

4.4 USE-CASE DIAGRAM

A use case is a list of steps, typically defining interactions between an actor and a system, to achieve a goal. The Use-Case Diagram for the Sales Process Analyzer is as follows:

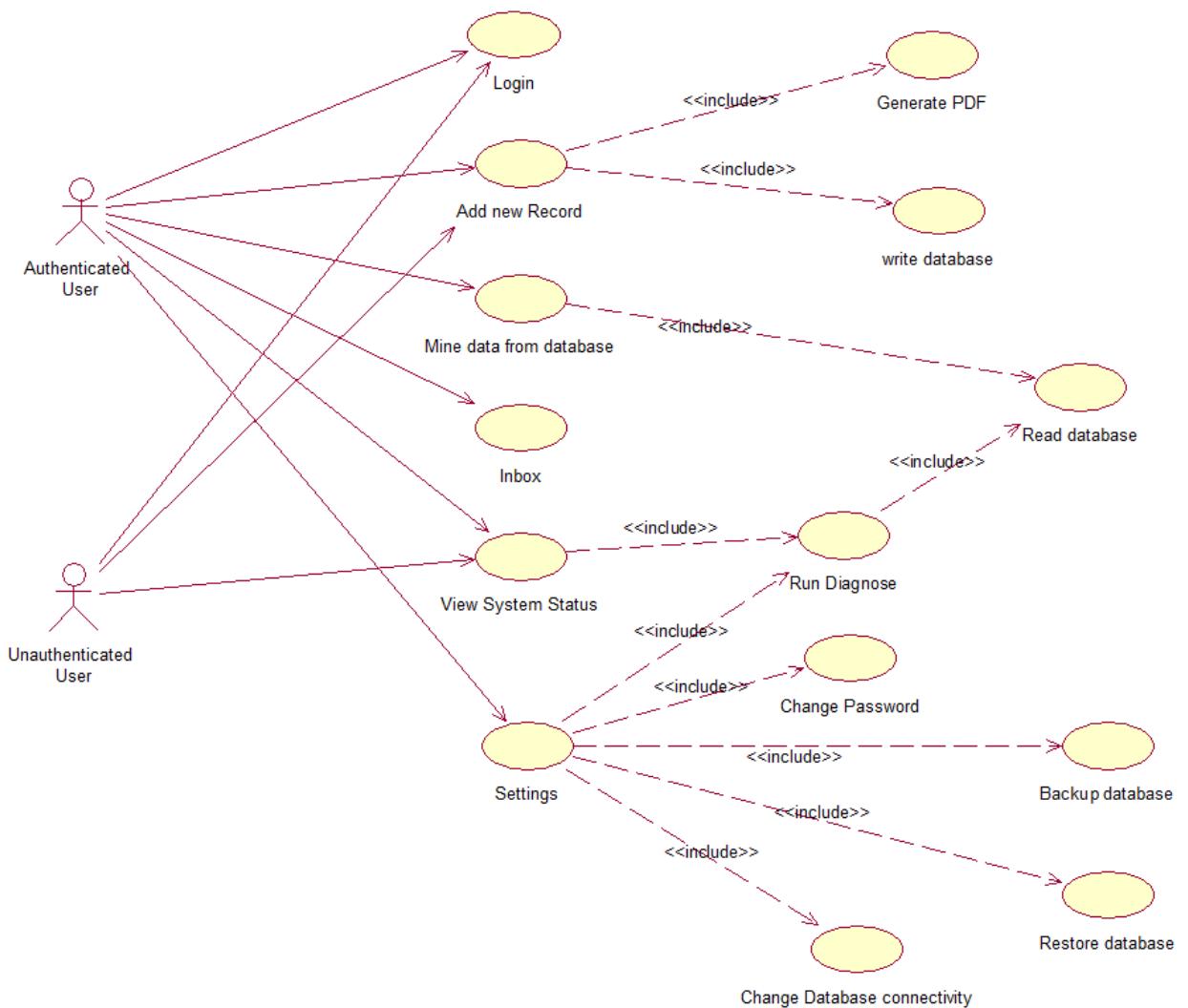


Fig 4.4 – Use Case Diagram

There are following actors in the use case:

- **Authenticated User:** The user who logs in to the application using a password.
- **Unauthenticated User:** The user who uses the application without logging into it.

Following are the use cases:

- **Login:** A login functionality to log in to the application using a password defined at the time of registration and configuration. The user must log in to access all the features of the application.
- **Add new Record:** The authenticated as well as unauthenticated user can add new record which includes **PDF generation** and writing the record to the database. For those users who log in, the data is stored in primary database whereas for the unauthenticated user the data is stored in virtual database.
- **Mine data from database:** The authenticated user can mine data from the database which includes reading the database using a record number. On successful match, the data can be loaded for analysis and report generation.
- **Inbox:** This functionality is used in case of database connectivity problems. In case where database connection is lost or tempered, the user has additional feature to store data into a virtual database in form of binary files. After the connection is made, the user can access it using inbox and can now move the data to primary database. Moreover, this feature is also used in case of unauthenticated user since the unauthenticated user is not allowed to access primary database.
- **View System Status:** Any type of user can view the system status which includes running diagnose of the whole system and identifying problem if any in critical areas of the application.
- **Settings:** The settings can only be made by an authenticated user. They include **running diagnose, changing password, maintaining backup database, restoring database and changing database connectivity.**

4.5 CLASS DIAGRAM

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. The Class Diagram for the Sales Process Analyzer is as follows:

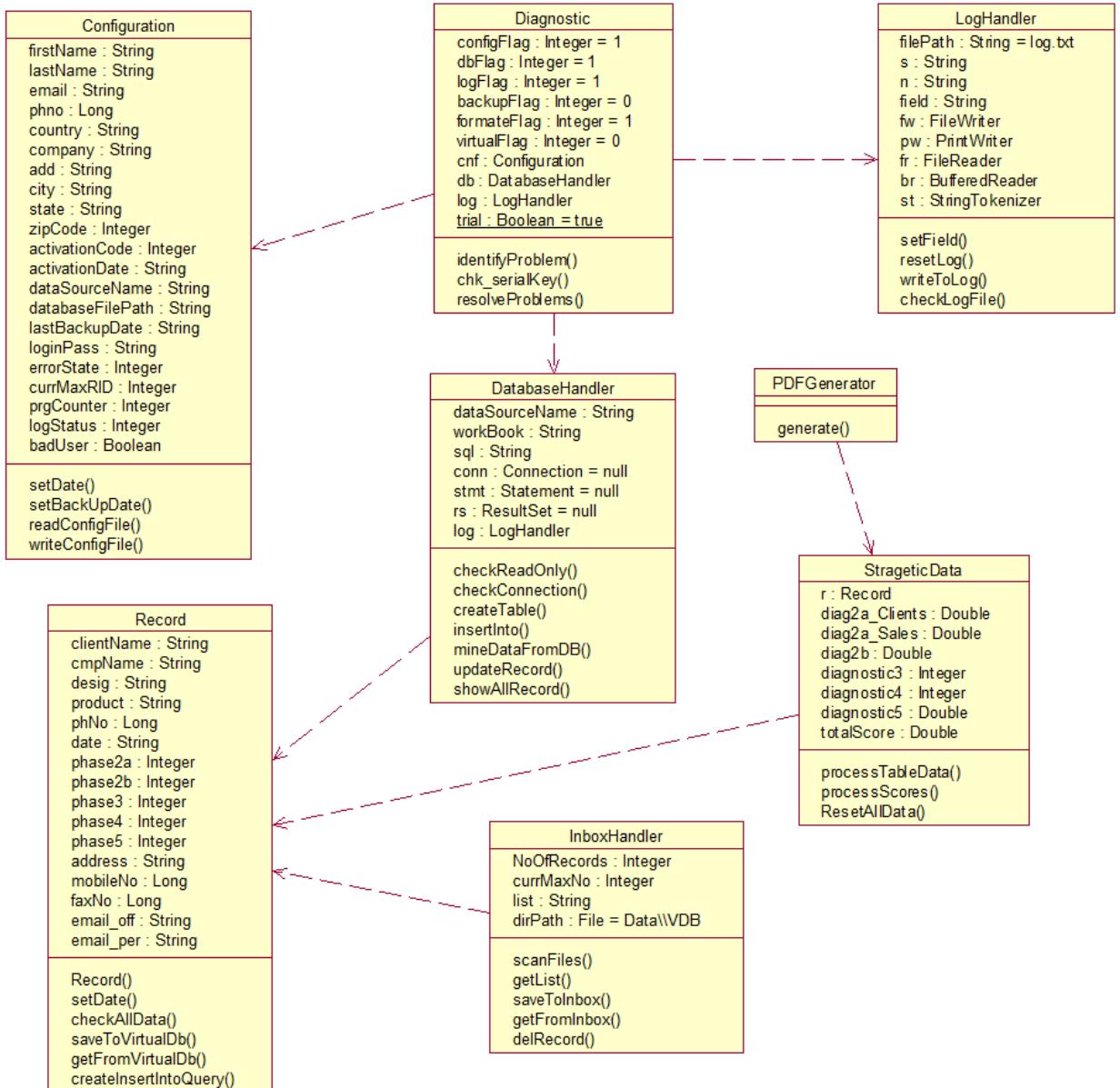


Fig 4.5 – Class Diagram

The class diagram comprises of the following classes:

- **Configuration:** The most important class of the system that contains configuration related information of the system. It is this class object that is used to read and write to and from the configuration file to configure the system accurately.
- **Diagnostic:** It is used for identifying and resolving any problem in the critical areas of the system.
- **Record:** This class contains all the inputs made by the client or read from the database. These inputs are later used for processing.
- **DatabaseHandler:** This class is used to handle all database related operations such as database connectivity check, create necessary tables, read, write and update database.
- **InboxHandler:** It is used to handle virtual database (Inbox). This class scans files and transfer data to primary database on demand.
- **LogHandler:** The responsibility of this class is to maintain a log file and constantly update it as the application performs critical operations.
- **PDF Generator:** This class is responsible for generation pdf file from processed inputs.
- **Strategic Data:** This class is responsible to hold the inputs as well as processed data to be output. These data is then used for generating pdf and on screen output.

4.6 STATE CHART DIAGRAM

State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented in series of events that could occur in one or more possible states. The State Chart Diagram for the Sales Process Analyzer is as follows:

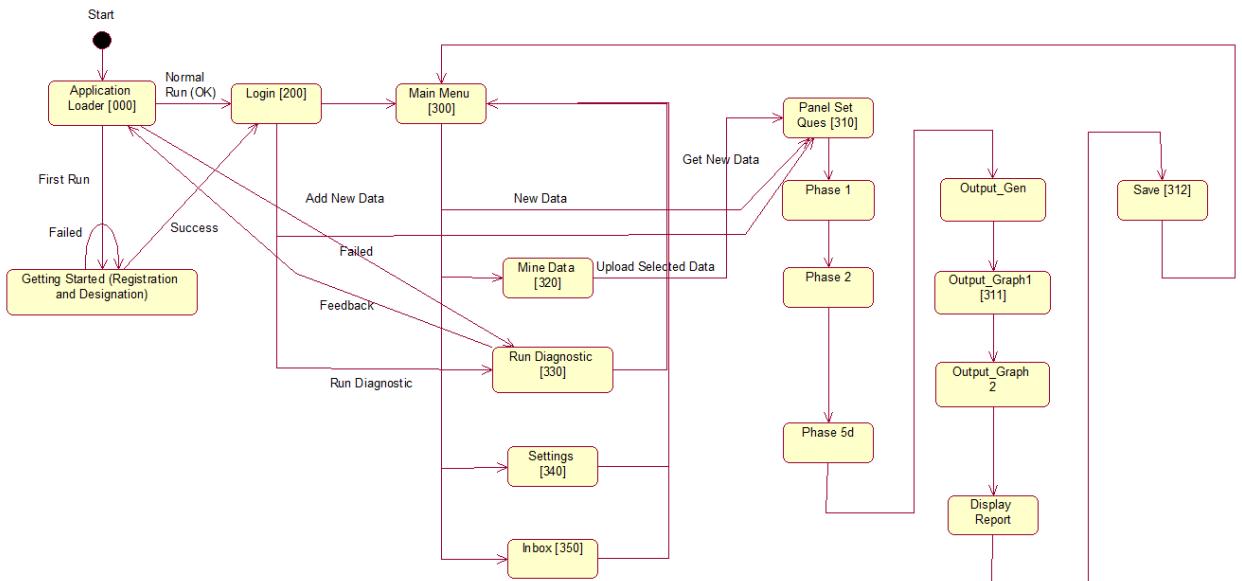
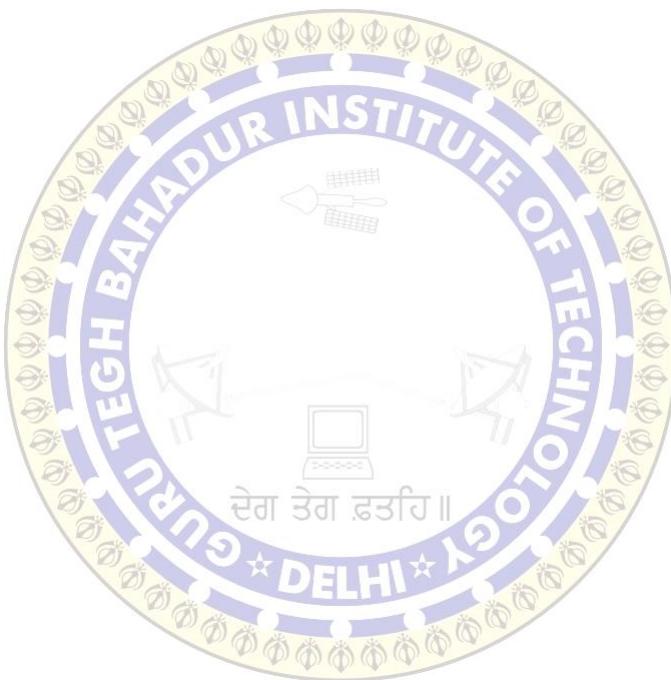


Fig 4.6 – State Chart Diagram

The state chart diagram represents the states as follows:

- At the start, the ‘Application Loader’ is initiated which on normal run leads to ‘Login’, else jumps to the ‘Getting Started’ state in case it is the very first execution of the application. The ‘Application Loader’ first goes to ‘Run Diagnostic’ state to scan the system and in turn receives a feedback.
- On successful login the ‘Login’ state leads to ‘Main Menu’ which can further lead to ‘Mine Data’, ‘Run diagnostic’, ‘Settings’ or ‘Inbox’ state depending upon user choice. In case of unsuccessful login the system remains in ‘Login’ state.
- After adding new data ‘Login’ as well as ‘Main Menu’ lead to a ‘Panel Set Ques’ which has ‘Phase1’, ‘Phase2’, ‘Phase3’, ‘Phase4’ and ‘Phase5’.

- This further leads to the ‘Output_Gen’ state which is followed by the ‘Output_Graph1’, ‘Output_Graph2’, ‘Display Report’ and ‘Save’ state. After ‘Save’ state the system comes back to ‘Main Menu’ state.



4.7 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The various Activity Diagrams for the Sales Process Analyzer are as follows:

4.7.1 BOOTING THE APPLICATION

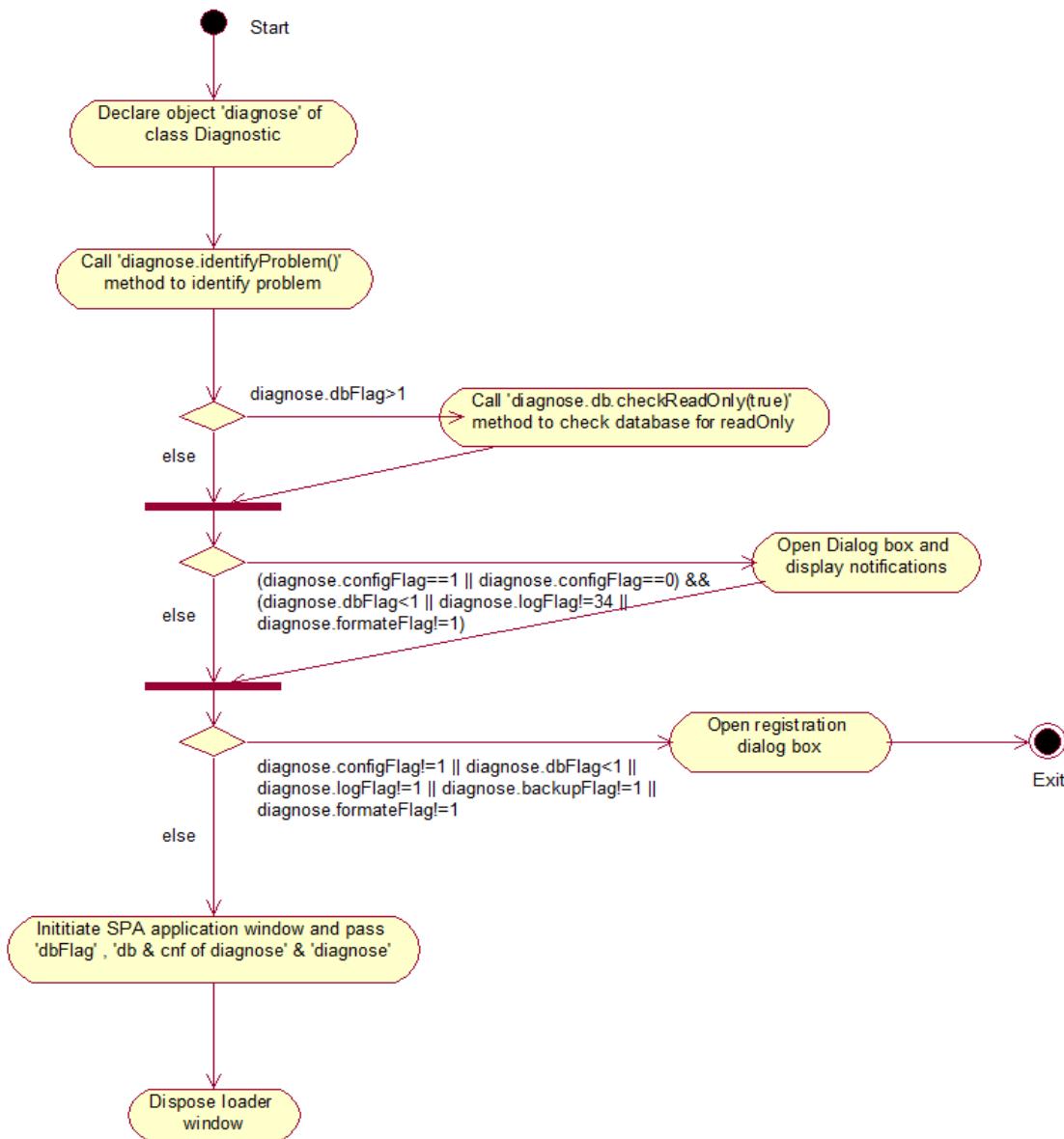
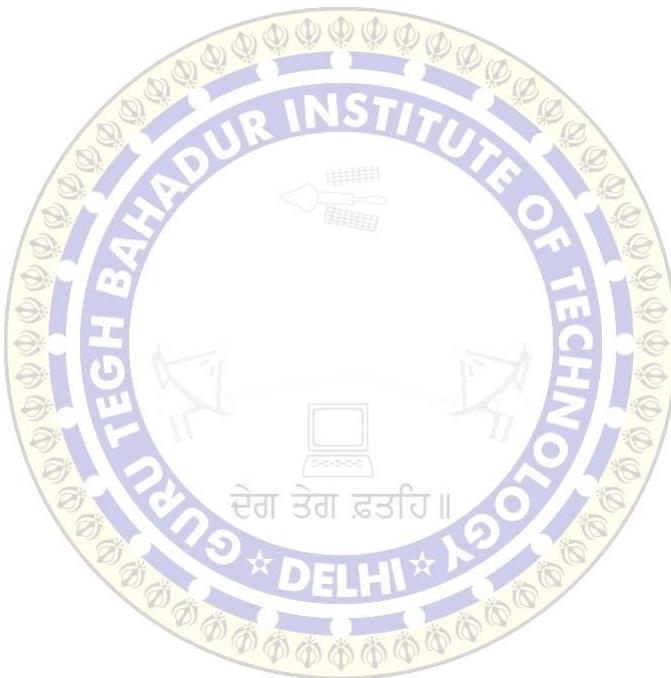


Fig 4.7 – Activity Diagram for Booting the application

- During booting a loader window is first to be displayed that executes a diagnosis algorithm.
- An object of class ‘Diagnostic’ is created which calls a method to identify the problem while booting. The results of the diagnosis are saved in flag variables.
- The values of the flag variables determines the state of error which later can be used for resolving the problem.
- After the diagnosis algorithm has been executed, any errors are displayed and the SPA application window gets initiated and the loader window is disposed.



4.7.2 DIAGNOSE ALGORITHM

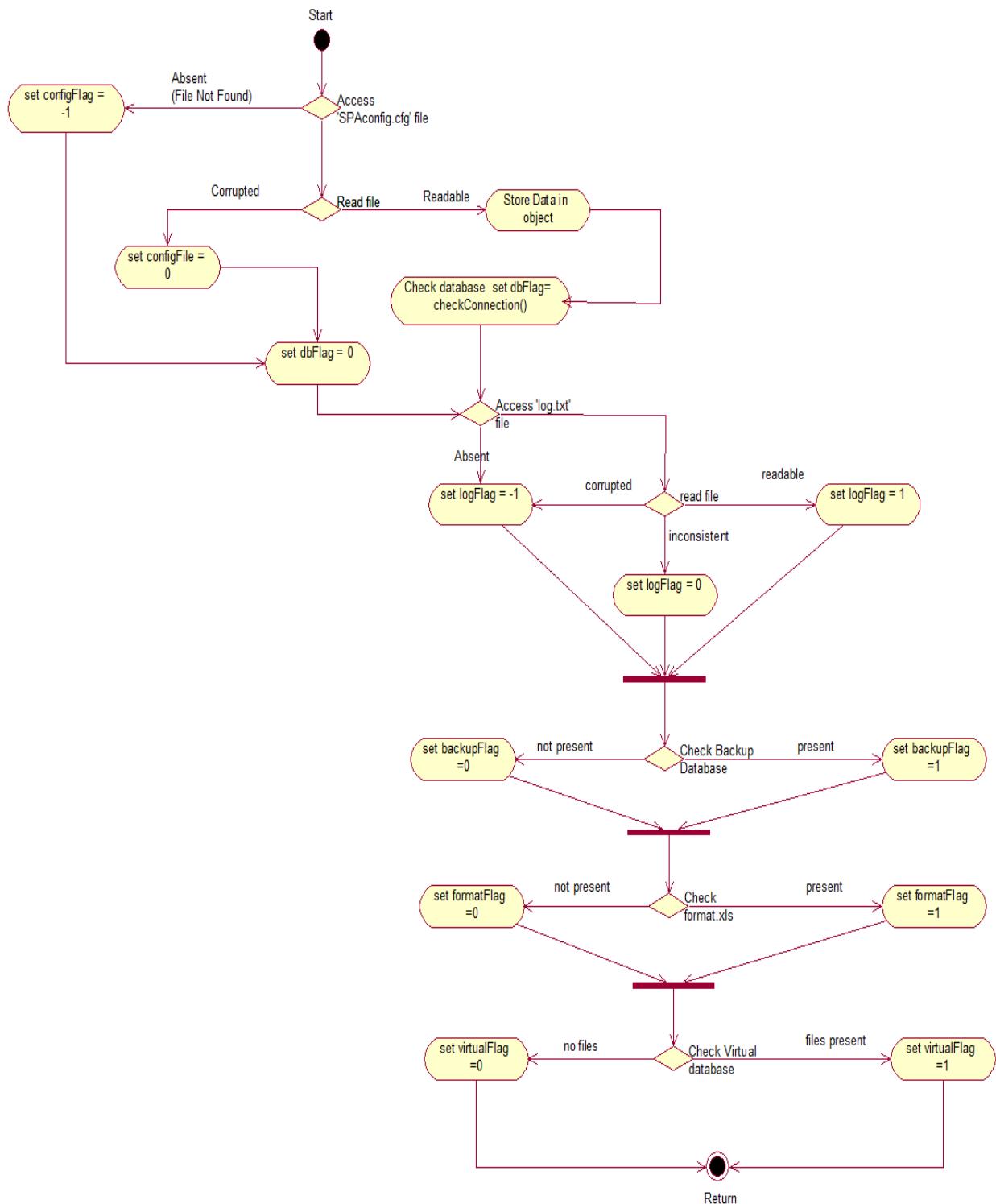
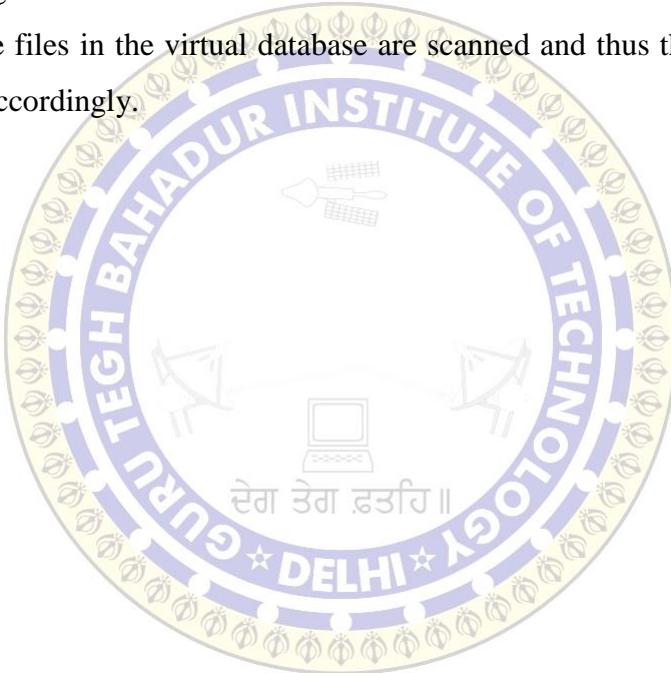


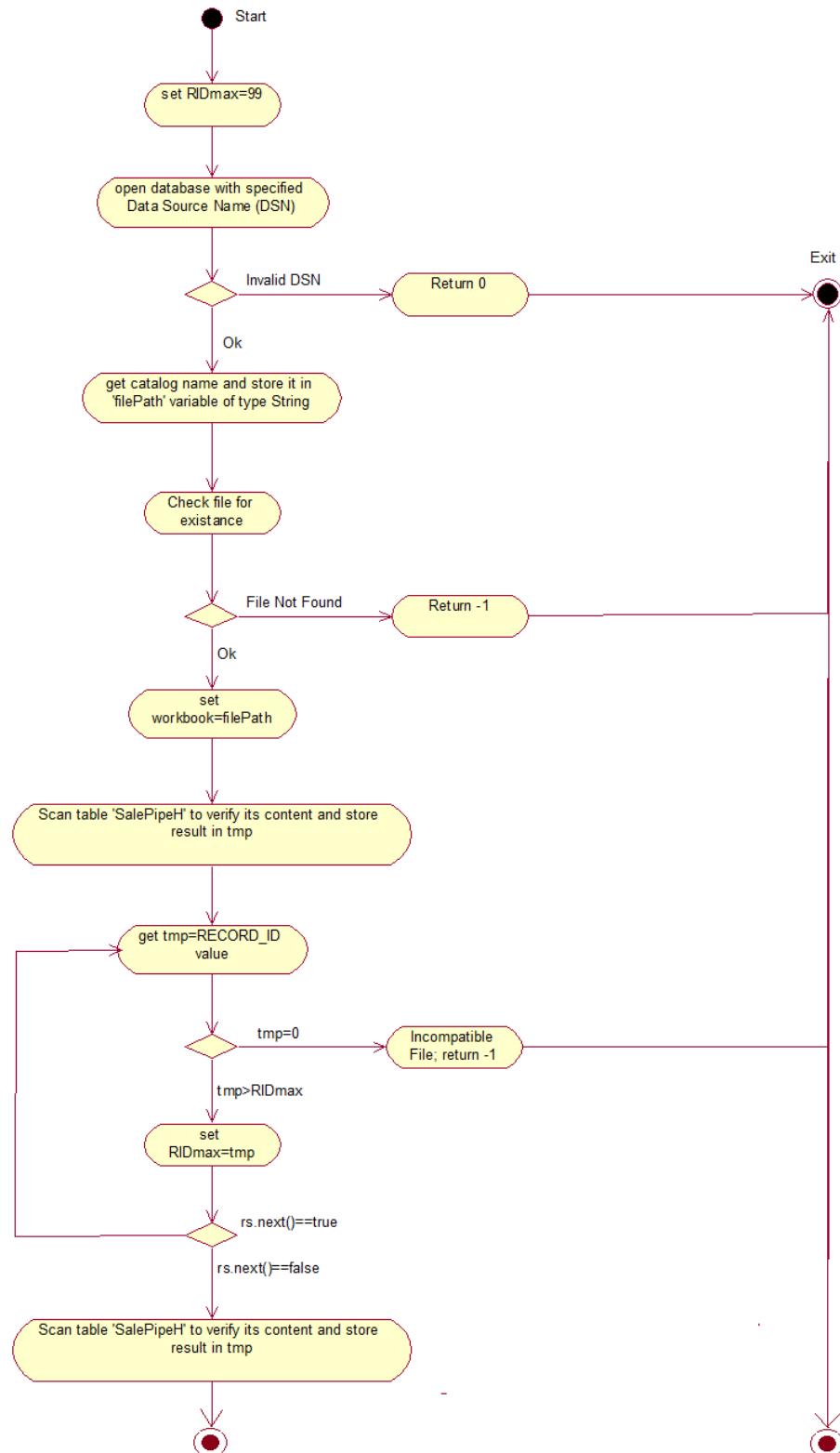
Fig 4.8 – Activity Diagram for Diagnosis algorithm

The following sequence of activities occurs in the above activity diagram:

- In the diagnose activity, initially, the SPAconfig.cfg file is checked for whether it is absent or present. If present, it is checked if it is corrupted or readable. Accordingly the configFlag and the dbFlag are set.
- Then the log.txt file is checked for whether it is absent or present. If present, it is checked if it is corrupted or readable. Accordingly the logFlag is set.
- Next the backup database is checked and accordingly backupFlag is set.
- Then the format.xls file is checked if found or not and accordingly the values for formatFlag are decided.
- Finally the files in the virtual database are scanned and thus the virtualFlag gets its value accordingly.



4.7.3 DATABASE CONNECTIVITY CHECK ALGORITHM



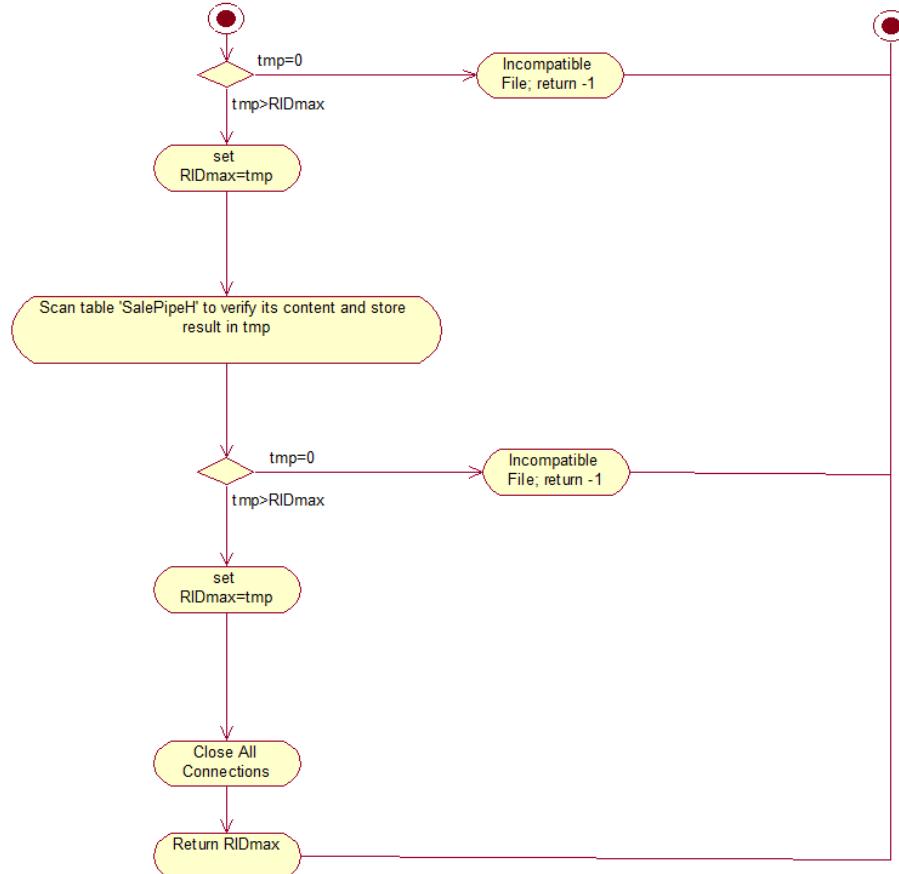
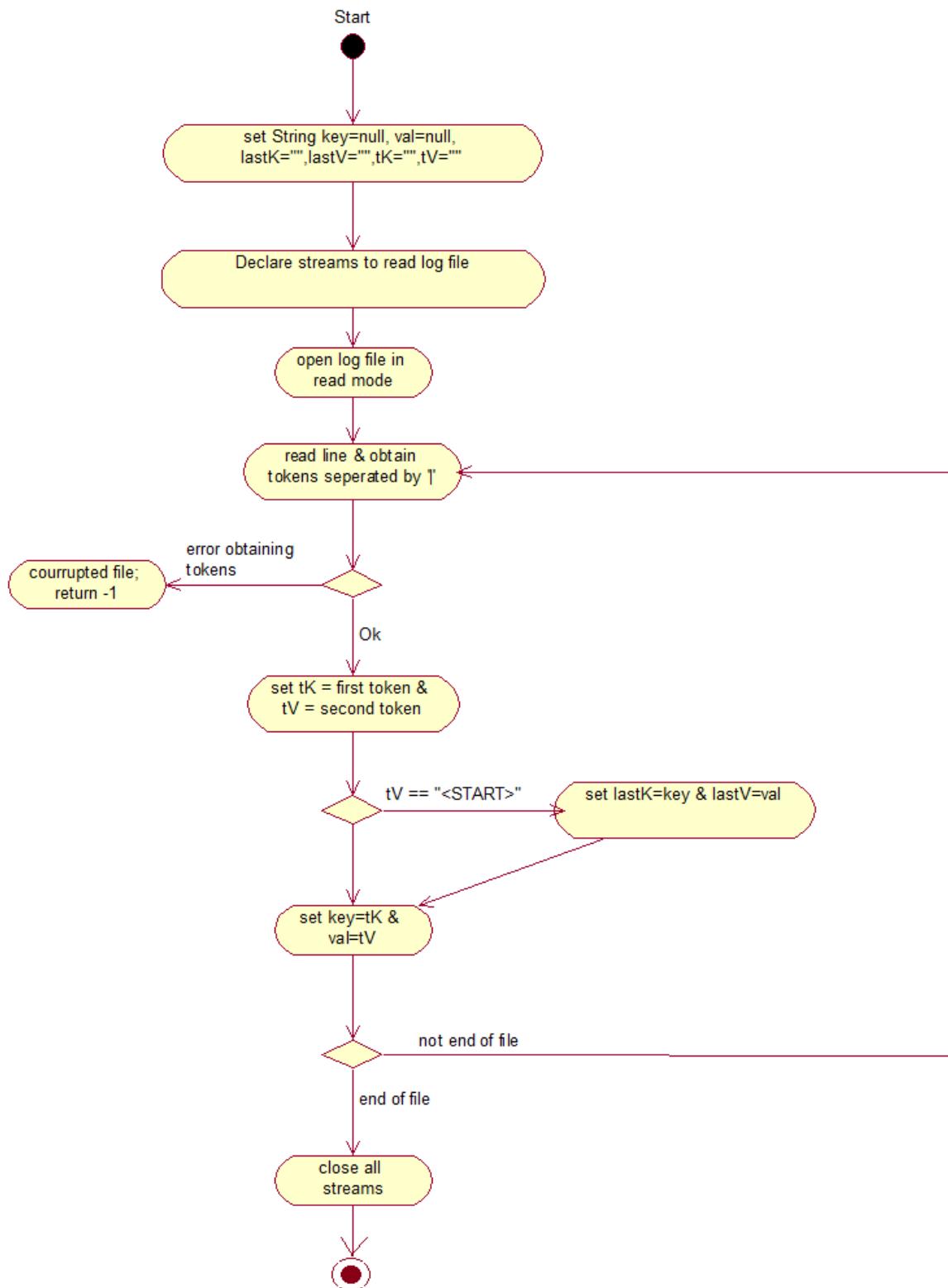


Fig 4.9 – Activity diagram for database connectivity check algorithm

The above activity diagram proceeds as follows:

- First the DSN is checked for so as to open the database.
- Then the file is checked for its existence using its catalog name.
- All tables are scanned for verification.
- Each row is scanned and the result is stored in a temporary file.
- The number of records is counted.
- All connections are closed.
- Finally the value of the last records is returned.

4.7.4 LOG CHECK ALGORITHM



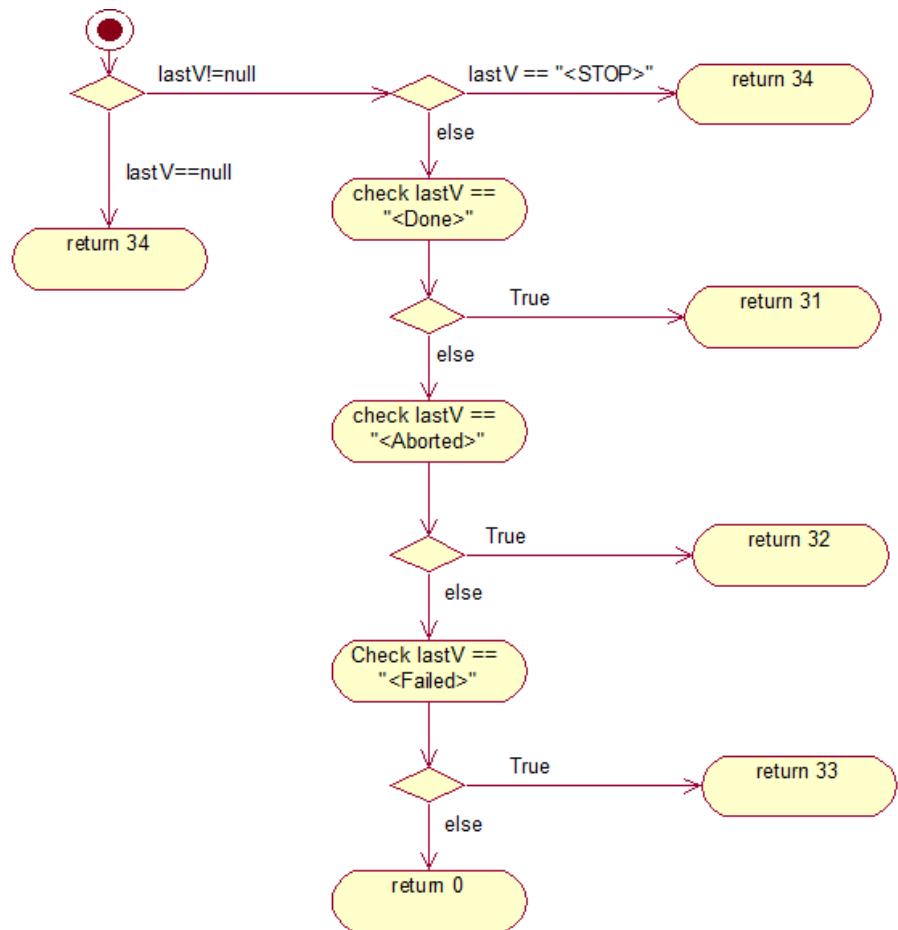


Fig 4.10 – Activity diagram for log check algorithm

The following sequence of activities takes place in log checking activity:

- The log file is opened in the read mode.
- The log file is read by obtaining tokens separated by '|'.
- The next token and the previous token are checked at each step and steps performed accordingly.
- The tokens are checked for ‘Start’, ‘Stop’, ‘Done’, ‘Aborted’ and ‘Failed’ and the next state to be taken by the system is returned accordingly.

4.8 DEPLOYMENT DIAGRAM

A deployment diagram models the physical deployment of artifacts on nodes. The Deployment Diagram for the Sales Process Analyzer is as follows:

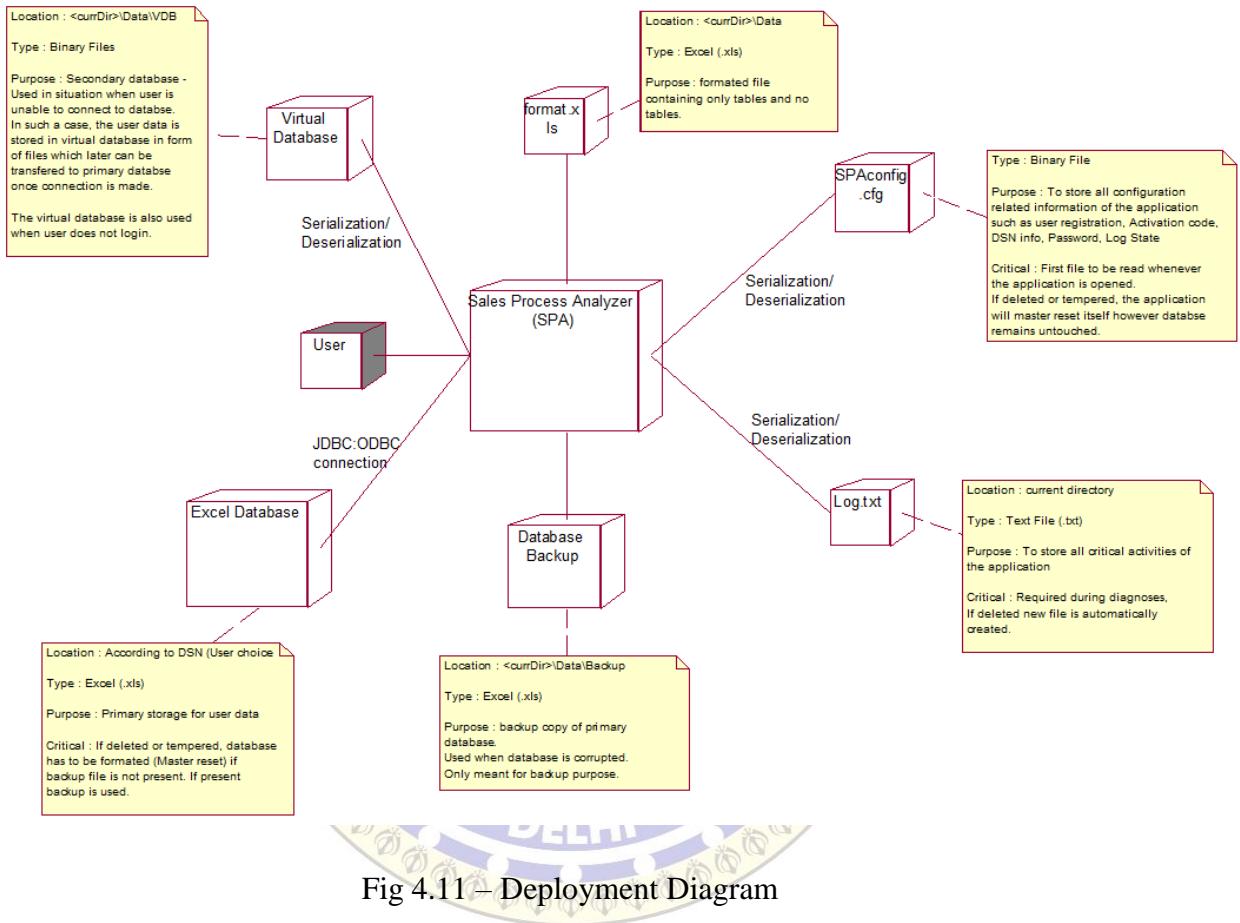
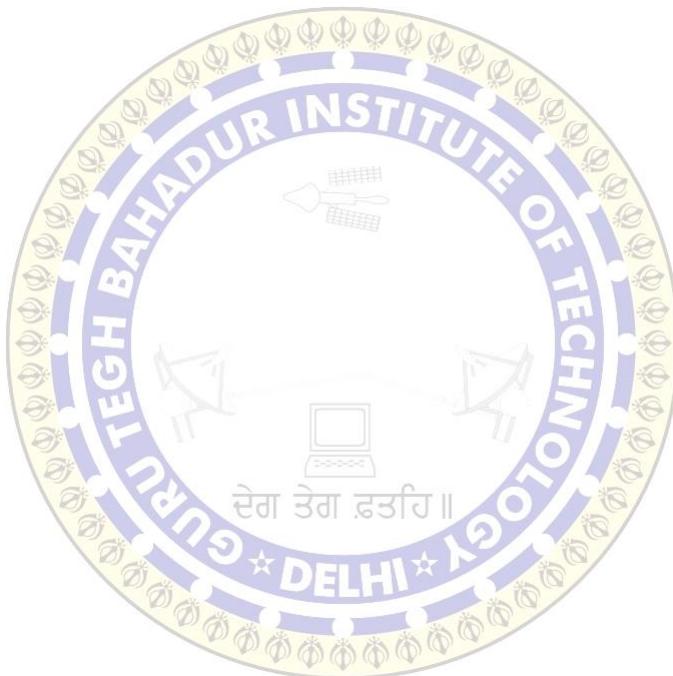


Fig 4.11 – Deployment Diagram

In the above deployment diagram:

- **Excel Database** is the primary database for user data that stores only inputs made by the user.
- The **Virtual Database** is secondary database used to store user data in the form of files during database connectivity problems. The data can later be move to primary database.
- **format.xls** is the formatted file containing only tables and no data.

- **SPAconfig.cfg** is a binary file used to store all information related to the configuration of the system.
- **Log.txt** is the current directory which stores all critical activities of the application.
- **Database Backup** is the backup copy of the primary database and used in case of database corruption.



TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing can be stated as the process of validating and verifying that a software program/application/product:

- meets the requirements that guided its design and development;
- works as expected
- Can be implemented with the same characteristics.

Different software testing techniques have been discussed in this section to investigate and verify that “Sales Process Analyzer” meets the various requirements and works efficiently as expected.

5.1 Functional Testing (Black Box Testing)

A functional testing is testing technique based on the functionality of the program and involves only observation of the output for certain input values. There is no attempt to analyze the code, which produces the output. We ignore the internal structure of the code and therefore, functional testing is also referred to as **black box testing** in which contents of the black box are not known. Functionality of the black box is understood completely in terms of input and output. Thus, in functional or black box testing, we are interested in functionality rather than internal structure.

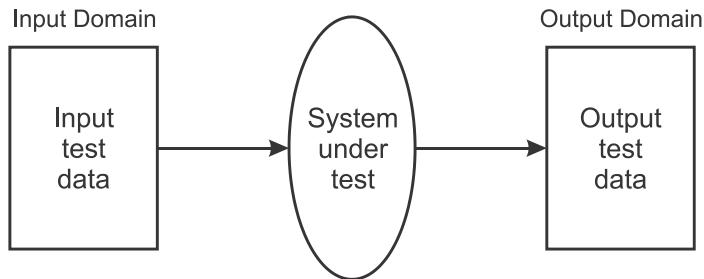


Fig 5.1 – Black box Testing

The following table describes the functional testing of various modules of Sales Process Analyzer ;

Table 5.1 – Functional Testing

Test ID	Test Case Scenario	Test Steps	Expected Results	Actual Result	Pass/Fail
Module 1 : Application Loader (Diagnosis)					
01	To verify the response of application boot up when it is run for the every first time on a fresh machine.	Run the application on a fresh machine.	The application must pop a registration window rather than showing the login screen.	The application ask for registration and a restart after completion	Pass
02	To verify the response of the application boot up when	Run the application on a registered machine with	The application must jump to login screen	Registration window is displayed.	Fail (Flags where not

	there are no errors.	all necessary files intact.	without any unusual message report.		updated)
03				Login screen appears after loading completes.	Pass
04	To verify the response of application boot up in error state.	Delete one or more critical files such as backup.xls, format.xls, or delete ODBC connection.	A dialogue box must appear describing the summary of diagnosis with indication of error states.	The application response as expected.	Pass

Module 2 : Registration

05	To examine registration module.	(a) Enter wrong activation Key.	The application must on accept the registration and ask for validation of activation key.	The application does not accept the registration.	Pass
06		(b) Enter correct details	Completes the	Response as expected.	Pass

		and activation key.	registration process, display message and ask for restart.		
07		(c) To verify obsolete trail version activation key.	The application must on accept again the trail activation key.	The trail activation key is again accepted.	Fail (condition to check obsolete key was not correct)
08		(d) To verify obsolete trail version activation key.	The application must on accept again the trail activation key.	The trail activation key is indicated as invalid key and ask for re-entry.	Pass

Module 3 : Login

09	To test the login module.	(a) Enter wrong password.	Login error message in a popup bubble must appear.	Pop bubble appears and ask to re-enter the	Pass
----	---------------------------	---------------------------	--	--	------

				password.	
10		(b) Enter correct password as specified during registration.	The application jumps to main menu.	Main Menu appears.	Pass

Module 4 : Report Generation

11	To test the input processing module that accepts inputs from made by the user, applies the designed algorithm and generates output.	(a) Enter all entries ask by the GUI. Leave no entry as blank.	The module must generate output as expected.	All outputs were accurate and upto the mark.	Pass
12		(b) Skip those entries that are permitted by the GUI.	The module must generate output as expected.	Previously entered values were retained in place of empty values.	Fail (empty values must be set to null or any legal default value)
13				All outputs were	Pass

				accurate and upto the mark.	
--	--	--	--	-----------------------------	--

* Note : The upper bound, lower bound as well as the validation of the inputs is the responsibility of the GUI components used to take the inputs. If any value is against the set of valid inputs, the reports the error immediately and does not allow user to proceed until entries have been rectified.

Module 5 : PDF Report Generation

14	To test the generation of PDF file with report summary.	After all input data has been processed and output has been displayed, select the option to create PDF file. Select path and click save.	A PDF extension file must be created to the selected path containing summary of inputs and outputs.	PDF file was successfully created however text were not properly aligned.	Fail (Alignment must be corrected)
15				PDF file was successfully created.	Pass

Module 6 : Database Handling

16	To test the insertion, updation and retrieval of	(a) Insertion	All inputs must be successfully inserted into	All inputs were inserted. In case of	Pass
-----------	--	---------------	---	--------------------------------------	------

	record to and from database.		the database.	database failure the record was shifted to virtual database.	
17	(b) Update	If a record has been uploaded from database, changed and then saved back, the intended record must be updated instead of creating new record into the database.	The record was updated	Pass	
18	(c) Retrieval	The record must be retrieved according to the record id which is hidden from	Correct record was retrieved.	Pass	

			the user.		
--	--	--	-----------	--	--

* Note : All the test in above module were conducted after many hit n trail over establishing connection to the database as well as running correct sql statements. The above tests are conducted over finalized module.

Module 7 : Virtual Database Handling

19	To examine the activity of virtual database. (creation of new file and transfer)	(a) Creation of new binary file.	A new binary file containing record data must be created in case of database connectivity error.	A new binary file was created. Data was saved from being lost.	Pass
20		(b) Transfer to primary database	The selected data can be transferred once the database issues have been resolved.	The record was successfully transferred and deleted from virtual database.	Pass
21		(d) Deletion	The record can be deleted.	The record was successfully deleted.	Pass

5.2 Non-functional Testing

Non-functional testing is the testing of a software application for its non-functional requirements. The names of many non-functional tests are sometimes often used interchangeably because of the overlap in scope between various non-functional requirements. Some of the non-functional testing of ‘Sales Process Analyzer’ are as follows :

5.2.1 Compatibility Testing: is testing conducted on the application to evaluate the application's compatibility with the computing environment. It determines how well software performs in a particular hardware / software / operating system / network environment and different combinations of above.

(a) Operating System and Processor: With respect to operating system and processor the compatibility of ‘Sales Process Analyzer’ can be studied on the bases of ‘average boot-up time of the application’. Once successfully loaded the response time of the application is nearly negligible.

Table 5.2 – OS and Processor compatibility

Operating System	Processor	Average Boot-Up Time (seconds)
Windows 7 Ultimate	Intel® Core™ i3 CPU 540 @ 3.07GHz	5 second appx.
Windows Xp SP2	Intel® Core™ i3 CPU 540 @ 3.07GHz	5 second appx.
Windows Xp SP2	Intel® Pentium 4 @ 2.0 Ghz	9 seconds appx.

Conclusion: The application is compatible with almost every operating system and processor. However, it is recommended to work with atleast Windows Xp as OS and Dual core processors.

ODBC Connection: The following table describes the names of the ODBC connections have been tested to check the compatible of ‘Sales Process Analyzer’ with:

Table 5.3 – Database Connectivity compatibility

Driver	File Extension	Compatible
Driver do Microsoft Excel	.xls	Yes
Microsoft Access Driver	.mdb , .accdb	No
SQL Server	-	No
Any other	-	No

Conclusion: The application is compatible only with ODBC Connection with Driver do Microsoft Excel (.xls). For other connection the application reports incompatible database connection.

5.2.2 Recovery Testing: Recovery testing is the forced failure of the software in a variety of ways to verify that recovery is properly performed. Recovery testing should not be confused with reliability testing, which tries to discover the specific point at which failure occurs. Recovery testing is basically done in order to check how fast and better the application can recover against any type of crash or hardware failure etc. Type or extent of recovery is specified in the requirement specifications. It is basically testing how well a system recovers from crashes, hardware failures, or other catastrophic problems.

Certain recovery plans have been designed for proper working of ‘Sales Process Analyzer’ as :

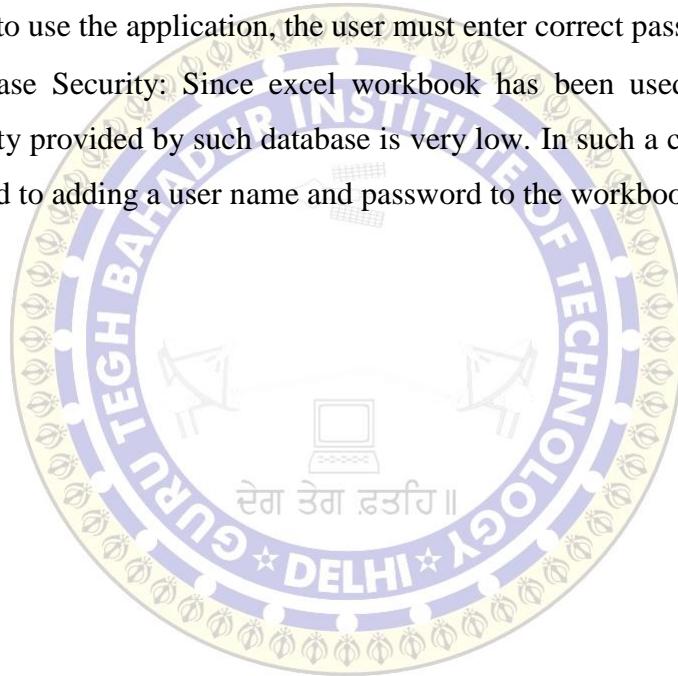
(a) Application Recovery: The application recovery is usually based on log file maintained by the application. If the application fails or crashes in the middle of execution, there will be no stop field written into the log. This informs the application on restart that a failure was occurred at last recorded activity.

(b) Database Recovery: A backup of the database is maintained each time the user tries to insert new record into the database. In case of any failure the user can run database recovery wizard to recover the database.

5.2.3 Security Testing: Security testing is a process to determine that an information system protects data and maintains functionality as intended. It determines how well the system protects against unauthorized internal or external access.

The ‘Sales process Analyzer’ provides security in terms of :

- **Secure Login:** The application provides a secure login at the start. So, in order to use the application, the user must enter correct password.
- **Database Security:** Since excel workbook has been used as database, the security provided by such database is very low. In such a case, the security is limited to adding a user name and password to the workbook only.



LIMITATION AND FUTURE SCOPE

6.1 LIMITATIONS

Following are the limitations of the project:

Sales Process Analyzer requires Java Runtime Environment (JRE) 6 or higher to run.

- The application is constrained to work with excel database only through JDBC-ODBC connectivity. It may not work with access, sql or other databases.
- The GUI of the application is restricted from 600x400 pixels to 1042x720 pixel resolution screen.
- The database is only meant to store the personal information of the client consulted as well as inputs only. No output of the application is feed into the database.
- The PDF report generated at the end does not include graphs as a part of the report.
- No other language other than English is supported.
- The project is limited only for analysis in one sector i.e. sales. It does not encompass other areas of analysis.

6.2 FUTURE SCOPE

This project has been designed to work on a large scale and in real time environments. This project can be effectively used as an analytical tool that will help the business owners to feed information and see as to where their organizations are lacking in Sales Process. It is a market proven business management technique that provides a great forecasting ability for forward looking business.

So according to our analysis the future scope of this project is bright and would have an everlasting effect on the current systems.

Various features can be included in the project to increase its analysis capabilities:

- The database capabilities of the application can be made more generic to support various types of database such as Ms-Access and sql.
- We can implement more fields of analysis using the same analytical tool, by giving more options to the user.
- The entire project can also be broken down into client-server system to increase its usability.

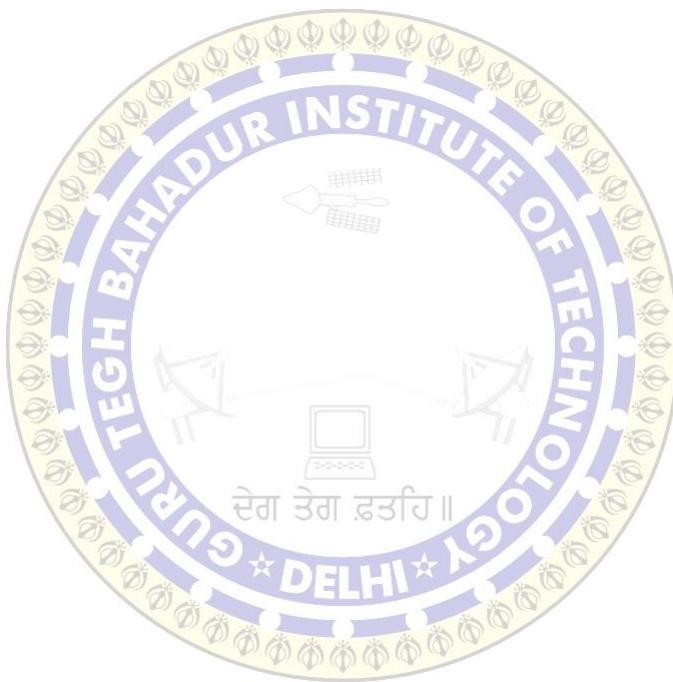


REFERENCES

- Herbert Schildt, “*The Complete Reference Java – Seventh Edition*”, Tata McGraw-Hill , 2011.
- Katy Sierra and Bert Bates, “*Head first Java*” , 2nd Edition.
- “How to Connect to an Excel Spreadsheet using JDBC in Java”,
<http://www.javaprogrammingforums.com/jdbc-database-tutorials/356-how-connect-excel-spreadsheet-using-jdbc-java.html>
- “How to make Dialogs”,
<http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>
- “How to round a number to n decimal places in java”,
<http://stackoverflow.com/questions/153724/how-to-round-a-number-to-n-decimal-places-in-java>
- “How can I connect to an Excel spreadsheet file using jdbc”,
<http://www.jguru.com/faq/view.jsp?EID=32876>
- “Sun MicroSystems - JAVA Training”, Sun Microsystems Inc.
- “Master IT – Advance Java Tutorial”, Cyclon.
- “Jframe at Center Screen”, <http://www.java-forums.org/awt-swing/3491-jframe-center-screen.html>.
- “Adding Image to Header and footer at generating Itext PDF”,
<http://www.roseindia.net/answers/viewqa/Java-Beginners/20666-Adding-Image-to-Header-and-footer-at-generating-Itext-PDF.html>
- “Adding an Itext Image to a PDF document”, <http://tutorials.jenkov.com/java-itext/image.html>

- “Copying a File or Directory ,
<http://docs.oracle.com/javase/tutorial/essential/io/copy.html>
- “Generate PDF files from Java applications dynamically”,
<http://www.ibm.com/developerworksopensource/library/os-javapdf/>
- “PDF Generation in Java using iText JAR” , <http://viralpatel.net/blogs/generate-pdf-file-in-java-using-itext-jar/>
- “Getting Cell Values in a Jtable” ,
<http://www.roseindia.net/java/example/java/swing/GetCellValues.shtml>
- “How to round a number to n decimal places in Java - Stack Overflow” ,
<http://stackoverflow.com/questions/153724/how-to-round-a-number-to-n-decimal-places-in-java>
- “Adding an IText Image to a PDF document.” , <http://tutorials.jenkov.com/java-itext/image.html>
- “Insert Row in Java,Inserting Row in JTable,Code to Add Row in Java Table Using Swing” ,
<http://www.roseindia.net/java/example/java/swing/InsertRows.shtml>
- “iText Tutorials - iText images, Java PDF API for Open, Manipulate and Creation ” , http://www.geek-tutorials.com/java/itext/itext_image.php
- “iText example at JavaRanch” , <https://www.coderanch.com/how-to/java/ItextExample>
- “pdf generation - java create pdf with header and footer - Stack Overflow” ,
<http://stackoverflow.com/questions/5551456/java-create-pdf-with-header-and-footer>
- “How to Use Modality in Dialogs” ,
<http://docs.oracle.com/javase/tutorial/uiswing/misc/modality.html>

- “PDF Generation in Java using iText JAR”, <http://viralpatel.net/blogs/generate-pdf-file-in-java-using-itext-jar>





Screen Shot 1 - UAE BootUp Window

Booting window that displays the logo of United Avenir Engineers' Logo. This is the first screen that appears whenever the application is executed.

The image shows a registration form titled "Getting Started ...". The form is divided into several sections:

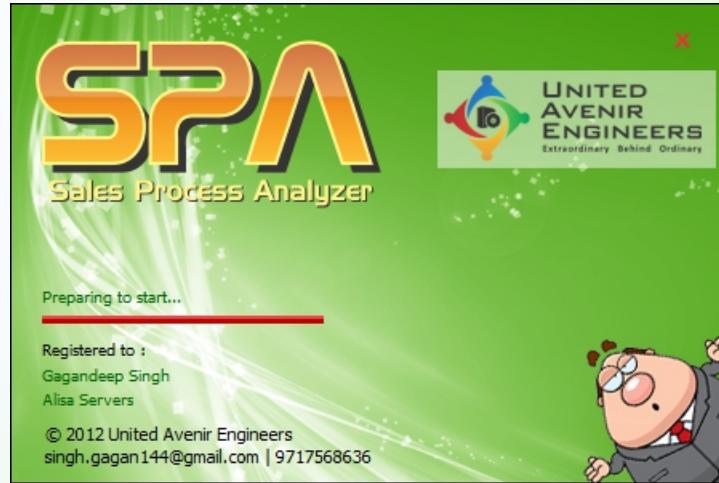
- Registration:** Contains fields for First Name (Sagandeep), Last Name (Singh), Company (Alisa Servers), E-Mail (alisaservers@gmail.com), Phone No (25793648), Address (198 Defence Colony), City (New Delhi), State (Delhi), Country (India), and Zip/Postal Code (110064).
- Database Connection:** Contains fields for Data Source Name (SPA) and WorkBook Path (Format Workbook : C:\Program Files\SF). It also has "Format" and "Process" buttons.
- Password:** Contains fields for Administrator Password and Confirm Password, both set to "*****".
- Product Activation:** Contains four activation keys and a "Check" button.

At the bottom, there are two checked checkboxes: "I am a genuine buyer of this product." and "I accept the terms and conditions in the License Agreement.". There are also "Submit" and "Exit" buttons.

Screen Shot 2 - Registration

Whenever the application is executed on a new machine, the registration screen pops up. The user must register himself before he can run the application. This screen appears after the BootUp window.

On completing, the user is required to restart the application.



Screen Shot 3 - BootUp Window

This is the second screen of bootUp window after displaying UAE logo. It is here when the application verifies the state of the application, run diagnose algorithm and check for any errors. The progress bar specifies the progress made. After completion, errors are displayed if any and application is initiated.



Screen Shot 4 - Main Menu

On successful loading, the login screen appears where the user must enter the password he had chosen while registering. Even though here, the user gets a option to add new record however this data is stored in secondary database rather than to primary database.



Screen Shot 5 - Main Menu

This is the Main Menu of the application that appears on successful login. The user can now navigate to various options. Add new record, Mine data, check inbox, view system status or even change Settings.



Screen Shot 6 - Add New Record | Personal Information

Personal information of the client being consulted is the first requirement of the application as the user adds new record. The user must enter correct values else an error message is displayed.



Screen Shot 7 - Add New Record | Definitions

This is second screen of ‘Add New Record’ where the application displays various definition that it uses. These definitions are meant to make the client comfortable with the application.



Screen Shot 8 - Add New Record | Past information

Here the user enters the past performance of the company of his client on the basis of number of clients & sales expected and gained as well as Clients touched, Prospects, leads and Sales. The data is in tabular form on yearly basis. These data is then used to obtain graphs.



Screen Shot 9 - Add New Record | Graphs for Expected Vs Actual

The above graphs are the graphical representation of the data entered in the previous screen. These graph displays the difference between the expected clients/sales and actual clients/sales. The lights above each graph specifies the warning state.



Screen Shot 10 - Add New Record | Graph for Clients Touched/Prospects/Leads/Sales

These are the second set of graphs based on the second table entered showing the various ratio between Clients Touched, Prospects, Leads and Sales. Here also the lights above each graph specifies the weak or strong areas.

Sales Process Analyzer

Based on Sales Process Analysis for better sales performance

United Avenir Engineers

SPA Sales Process Analyzer

Get Qualified Prospect List Generate Qualified Leads Generate Sales

3. Generate Qualified Prospect List:

Q3) Do you have a certain strategy to supply from qualified accounts to your sales process?

Yes, we are in a position to generate leads from our qualified accounts on a dedicated basis.

Yes, we have a database which contains our clients who are called directly on a regular basis.

Yes, we take help of external agencies from time to time and are satisfied with their performance.

No, we do not have any in-house or outsourced mechanism in our sales team & our sales team is expected to do the same.

No, we are dependent on leads required by our partners.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q2) Do you have a proper qualified account list definition / criteria? For Example: No. of employees, Turnover, Mkt, and L&B etc.

If yes, do you think it could be re-defined to zero down on precise market?

Yes, we have proper qualified account list definition for each of our offering and our sales team member have perfect understanding of the same.

Yes, we have a proper qualified account list definition for each of our offering however our sales team does not understand it properly and act accordingly.

No, we do not have a perfect qualified account list definition and wish to know how it would help us precisely.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q1) Do you study the profile of the accounts deeply to decide sales strategy?

Yes, we divide each account in our qualified account list based on its profile and make an annual sales engagement / strategy document.

Yes, we do not make any annual sales engagement / strategy document and are dependent on our partners to help us find opportunities.

No, we do not make any annual sales engagement / strategy document and are opportunity oriented.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q3) Do you have broader diversify your product / solutions / services offerings into well scripted RIC (Reasons to Call)? For Example: Advantage... Strategic Buyer, Economic Buyer & Technical Buyer.

Yes, we have well scripted RIC for all our products, offerings, solutions and services for Strategic, Technical and Economic Buyer.

Yes, we have well scripted RIC for all our products, offerings, solutions and services for Strategic, Technical and Technical.

Yes, we have RICs for R&D and R&D however the calling is not yet started for R&D.

No, we do not have RICs and R&D however our cold calling is based on email introductions.

No, we do not have RICs and R&D and our cold calling is random.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q2) Do you have and use a Reason to Call Calendar to get in touch with qualified prospect list?

Yes, we have a monthly reason to call calendar.

Yes, we have a weekly reason to call calendar.

No, we do not have a monthly reason to call calendar.

No, our calling team is not working according to reason to call calendar.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q1) How have sales team send prospect EDMs to relevant buyers and then speak to them about it and gauge their response?

Yes, our sales team regularly uses these EDMs to relevant stakeholders and engage them in discussions regarding the same.

Our sales team sometimes use these EDMs and we are not sure whether we use the same EDM for a discussion.

No, our sales team does not use them regularly to engage with the accounts.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q3) Does your sales team have been exposed to 4 key aspects of effective sales leads?

- Strategic Buyer
- Possible Objectives
- Buying Environment
- Buying Decision Makers

Yes, our sales team has been exposed to all 4 aspects of "Qualified sales lead" for each contact when we have a reason.

Yes, our sales team members take care of all 4 aspects of "Qualified sales lead" for each contact when we have a reason.

No, our sales team members take care of only 2 of the aspects for the rest they leave the decisions all depends on the team.

No, our sales team members do not have any information at once, however we quickly get all information.

No, our sales team does not have the information and we do not guide them to get the same.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q3) Do you have product / solutions / services presentations to be exhibited to Technical Buyer, Strategic Buyer & Economic Buyer - Separate one for each?

- Strategic Buyer
- Economic Buyer
- Joint

Yes, we have presentations to exhibit our products/services to our account in Strategic / Technical / Economic mode of evaluation.

Yes, we have presentations but they are not made according to Strategic, Technical and Economic buying behavior.

No, we do not have presentations.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q3) Do you have a relevant sales query form that the sales executives use to get customer feedback on presentations made to them?

Yes / No (Strategic, Technical, Economic)

- Why did they buy from us if so?
- Why did they not buy from us if so?
- What are their expectations from us?
- What they expect from us to give the sales order to us?
- Are they concerned that this product/service is right for them?

Yes, we have a relevant sales query form that it covers all these questions.

Yes, we have a relevant sales query form but it does not cover all these questions.

No, we do not have a relevant sales query form but we will try to make according to the need of the business.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q2) Do you gauge your sales forecast based on above Q2 questions?

Yes, we do our sales forecasting based on above 2 questions.

No, we do not do any sales forecasting based on above 2 questions.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Q3) Does your team use any CRM for managing sales projections? If yes, is this the way you review sales with your team? Please elaborate.

Yes, we have a CRM for managing sales projections and use it to the fullest.

Yes, we have a CRM but do not use it that aggressively.

No, we do not have a CRM or a Sales Template.

Get Qualified Prospect List Generate Qualified Leads Generate Sales

Thank you for completing our Analytic Test. Please click "Generate Report" to view your result!

Generate Report

Screen Shot 11 to 24 - Add New Record | Questioner

All the above screen shots corresponds to the next phase of the analysis where the user ask set of questions from his client and select the corresponding option from give choices. The questioner consist of 12 questions. The selected option then specifies the weak areas based on the algorithm.



Screen Shot 25 - Add New Record | Analysis Report

After the user has completed the questioner, a report is generated on the basis of the algorithm designed. The algorithm displays messages for each of the three sector considered in the questioner specifying weak areas and the actions to be taken.



Screen Shot 26 - Add New Record | Save Data

This is final step for 'Add New Record' option where the user can save data to database for future reference as well as generate PDF file for his client.



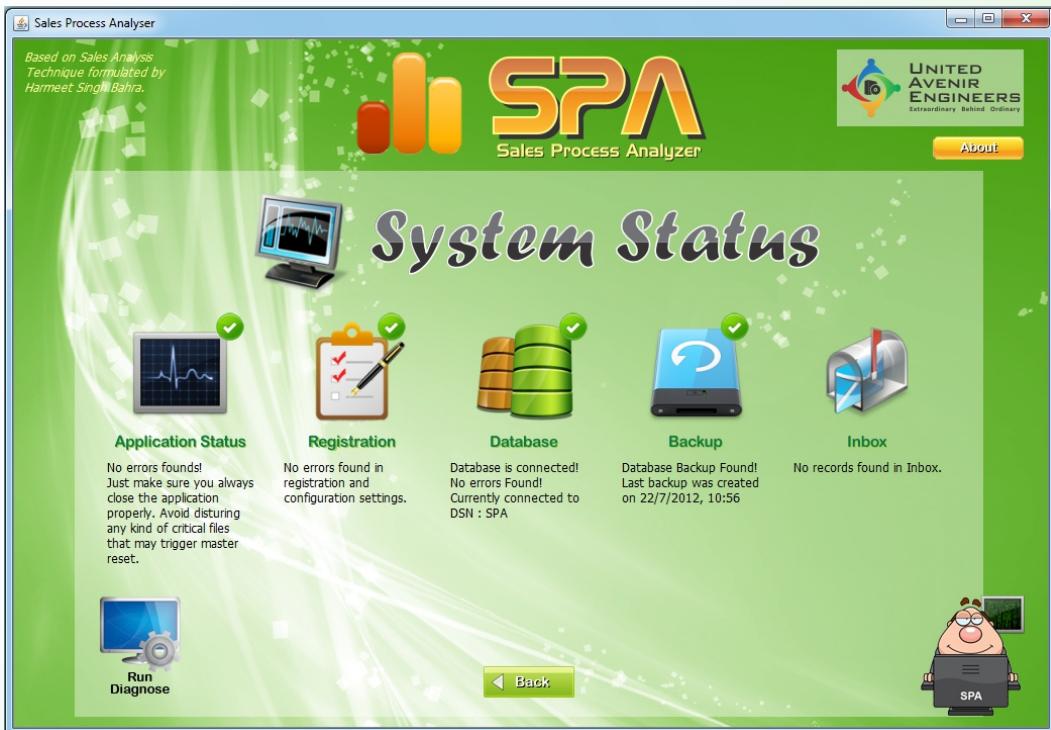
Screen Shot 27 - Add New Record | record entered

A Dialog box displaying message the all data has been saved to database. If there had been any kind of error during saving, a proper message had been displayed.



Screen Shot 28 - Mine Data

The above screen is displayed as the user clicks the ‘Mine Data’ button from main menu. Here the screen displays all the records present in the database. The user may click ‘View’ Button to view personal information or click ‘open’ button to load its contents for analysis.



Screen Shot 29 - System Status

A smart feature of the application that displays the state of various critical areas of the application. The ‘green tick’ sign indicates an Ok state while an ‘exclamation’ or a ‘cross’ sign indicate that attention is required.



Screen Shot 30 - Inbox

The application provides an additional feature for storing records to a secondary database in form of binary files in case of database connectivity error. On rectifying the problem the user can transfer the data to primary database.



Screen Shot 31 - Settings

Settings screen where the user can change database connectivity settings, change password and even create backup or restore database.



Screen Shot 32 - Settings | Database Restore Wizard

Database restore wizard to restore database from the previous backup created.



Screen Shot 33 - SPA Exception Handler

Another smart feature of the application - SPA Exception Handler. Whenever an error occurs in any part of the application, the above window pops up displaying error code, type description as well as tips to rectify the error.



Screen Shot 34 - About

The about window that displays description of the application, user, developing team and developing environment.

Personal Information

Client Name : Prithvi Rachand
Address : 201, Sector 10, Noida
Company : Techsoft Systems
Designation : Managing Director
Product : Tablet 8418 E Series
Phone Number : 25798416
Mobile Number : 9810642697
Fax Number : 46525799
Official e-mail ID : prithvi@yahoo.com
Personal e-mail ID :

Year on year Growth and Productivity

Year	Expected No. of Clients	Actual No. of Clients	Expected Sales Revenue	Actual Sales Revenue
2012	40	30	60	55
2011	30	20	55	30
2010	20	20	45	35

Analysis :

- Expected VS Actual Clients : Your company is doing some of the Sales Process related tasks correctly. However there seems to be an imperative need to identify weak areas to achieve considerable improvement and conversions. Please proceed to take analytic test.
- Expected VS Actual Sales : Your company is facing lot of challenges in Sales Process Management. We highly recommend you to take analytic test to identify weak areas and take immediate corrective actions for considerable improvement and conversions.

(a) Your Current Sales Process (Business) Productivity :

Year	Expected No. of Clients	Actual No. of Clients	Expected Sales Revenue	Actual Sales Revenue
2012	40	30	60	55
2011	30	20	55	30
2010	20	20	45	35

(b) Clients Touched To Prospects To Leads To Sales Conversion :

Year	Clients Touched (CT)	Prospects (P)	Leads (L)	Sales (S)
2012	300	250	100	35
2011	250	200	90	30
2010	350	320	110	38

ANALYSIS REPORT

3. Generate Qualified Lead:

- You need a dedicated data mining process to supply qualified accounts for sales enablement. We can help you with the same.
- The customers are not clear our offering: Services / Products/ Solutions. Our offering is not clear to the customer from Strategic / Technical / Economic View point.

4. Generate Qualified Leads:

- You must have SSTE calling scripts with a dedicated inside sales team to generate qualified leads.
- We recommend you to have a trained inside sales team that has a forward looking RTC-Calendar in place to generate qualified leads.
- We strongly recommend you to have regular EDM/OMM communication for customer activation.
- You need to study a profile of each account and draw an annual sales strategy plan to engage with these accounts.

5. Generate Sales :

- We strongly recommend that you should convert your sales presentations addressing the Strategic, Technical and Economic aspects, separately.
- We strongly recommend you to have a sales query form that covers all aspects of Strategic, Technical and Economic Buying.
- We strongly recommend that sales forecasting should be done through SSTE qualification success rate.
- You need to use the CRM tool aggressively to manage sales and sales projections.

Total Score : 31.0 /100

Signature

R&B Process Analyzer
Developed by © 2012 United Avenir Engineers
Core Team : Gagandeep Singh, Harneet Singh Balra, Harnet Kaur Bratta, Amritpreet Singh Chauhan
singh.page114@gmail.com | 9717958636

Screen Shot 35 - PDF Report

PDF report generated by the application describing data inputs as well as the analysis report that can be send via email or printed for the client being consulted.

```

File Edit Format View Help
1/4/2012,1:29:49 |<START>
1/4/2012,1:29:51 |<Loading>
1/4/2012,1:29:52 |<diagnosis>
1/4/2012,1:29:52 |<done>
1/4/2012,1:29:52 |<Recovery>
1/4/2012,1:29:52 |<done>
1/4/2012,1:29:52 |<done>
1/4/2012,1:29:53 |<Login>
1/4/2012,1:29:56 |<done>
1/4/2012,1:29:57 |<readDB>
1/4/2012,1:29:57 |<done>
1/4/2012,1:30:4 |<ReadDB>
1/4/2012,1:30:4 |<Done>
1/4/2012,1:30:20 |<genPDF>
1/4/2012,1:30:20 |<Done>
1/4/2012,1:30:20 |<writeDB>
1/4/2012,1:30:20 |<done>
1/4/2012,1:30:23 |<STOP>
3/4/2012,9:31:37 |<START>
3/4/2012,9:33:23 |<START>
3/4/2012,9:33:25 |<Loading>
3/4/2012,9:33:26 |<diagnosis>
3/4/2012,9:33:27 |<done>
3/4/2012,9:33:29 |<Recovery>
3/4/2012,9:33:29 |<done>
3/4/2012,9:33:29 |<done>
3/4/2012,9:33:31 |<Login>
3/4/2012,9:34:12 |<done>
3/4/2012,9:34:35 |<diagnosis>
3/4/2012,9:34:35 |<done>
3/4/2012,9:35:3 |<diagnosis>

```

Screen Shot 36 - Log File

A snapshot of the log file maintained by the application. The log files plays an important role in diagnosis the system during bootUp.

Dialog_ClientInfo.java

```
package salesPackage;

import javax.swing.UIManager;

public class Dialog_ClientInfo extends javax.swing.JDialog {
    public Dialog_ClientInfo(java.awt.Frame parent, boolean modal)
    {
        super(parent, modal);

        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        label_cName = new javax.swing.JLabel();
        label_cmpName = new javax.swing.JLabel();
        label_desig = new javax.swing.JLabel();
        label_product = new javax.swing.JLabel();
        label_phNo = new javax.swing.JLabel();
        textF_cName = new javax.swing.JTextField();
        textF_cmpName = new javax.swing.JTextField();
        textF_desig = new javax.swing.JTextField();
        textF_product = new javax.swing.JTextField();
        textF_phNo = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        textF_add = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        textF_mob = new javax.swing.JTextField();
        textF_fax = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        textF_emailOff = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        textF_emailPer = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        btn_ok = new javax.swing.JButton();
        topBar = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("SPA | Client Information");
        setPreferredSize(new java.awt.Dimension(400, 500));
        setResizable(false);
        getContentPane().setLayout(null);

        label_cName.setText("Client Name :");
        getContentPane().add(label_cName);
    }
}
```

```

label_cName.setBounds(50, 85, 90, 20);

label_cmpName.setText("Company :");
getContentPane().add(label_cmpName);
label_cmpName.setBounds(50, 149, 90, 20);

label_desig.setText("Designation :");
getContentPane().add(label_desig);
label_desig.setBounds(50, 181, 90, 20);

label_product.setText("Product :");
getContentPane().add(label_product);
label_product.setBounds(50, 213, 90, 20);

label_phNo.setText("Phone No. :");
getContentPane().add(label_phNo);
label_phNo.setBounds(50, 245, 90, 20);

textF_cName.setEditable(false);
textF_cName.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173, 179)));
textF_cName.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_cName);
textF_cName.setBounds(150, 85, 190, 20);

textF_cmpName.setEditable(false);
textF_cmpName.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173, 179)));
textF_cmpName.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_cmpName);
textF_cmpName.setBounds(150, 149, 190, 20);

textF_desig.setEditable(false);
textF_desig.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173, 179)));
textF_desig.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_desig);
textF_desig.setBounds(150, 181, 190, 20);

textF_product.setEditable(false);
textF_product.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173, 179)));
textF_product.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_product);
textF_product.setBounds(150, 213, 190, 20);

textF_phNo.setEditable(false);
textF_phNo.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173, 179)));
textF_phNo.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_phNo);
textF_phNo.setBounds(150, 245, 190, 20);

jLabel1.setText("Address :");
getContentPane().add(jLabel1);

```

```

jLabel1.setBounds(50, 117, 90, 20);

textF_add.setEditable(false);
textF_add.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173,
179)));
textF_add.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_add);
textF_add.setBounds(150, 117, 190, 20);

jLabel2.setText("Mobile No. :");
getContentPane().add(jLabel2);
jLabel2.setBounds(50, 277, 90, 20);

textF_mob.setEditable(false);
textF_mob.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173,
179)));
textF_mob.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_mob);
textF_mob.setBounds(150, 277, 190, 20);

textF_fax.setEditable(false);
textF_fax.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171, 173,
179)));
textF_fax.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_fax);
textF_fax.setBounds(150, 309, 190, 20);

jLabel3.setText("Fax No. :");
getContentPane().add(jLabel3);
jLabel3.setBounds(50, 309, 90, 20);

textF_emailOff.setEditable(false);
textF_emailOff.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171,
173, 179)));
textF_emailOff.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_emailOff);
textF_emailOff.setBounds(150, 341, 190, 20);

jLabel4.setText("Official eMail ID:");
getContentPane().add(jLabel4);
jLabel4.setBounds(50, 341, 90, 20);

textF_emailPer.setEditable(false);
textF_emailPer.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(171,
173, 179)));
textF_emailPer.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(textF_emailPer);
textF_emailPer.setBounds(150, 373, 190, 20);

jLabel5.setText("Personal eMail ID :");
getContentPane().add(jLabel5);
jLabel5.setBounds(50, 373, 90, 20);

btn_ok.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/ok_1.png"))); // NOI18N

```

```

btn_ok.setBorderPainted(false);
btn_ok.setContentAreaFilled(false);
btn_ok.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
btn_ok.setFocusPainted(false);
btn_ok.setPressedIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/ok_3.png"))); // NOI18N
btn_ok.setRolloverIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/ok_2.png"))); // NOI18N
btn_ok.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btn_okActionPerformed(evt);
    }
});
getContentPane().add(btn_ok);
btn_ok.setBounds(160, 420, 90, 31);

topBar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/Mine/client_info_topBar.png"))); // NOI18N
getContentPane().add(topBar);
topBar.setBounds(0, 0, 400, 500);

pack();
}// </editor-fold>

private void btn_okActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
}

//-----MY Methods-----
void showInfo(Record ref)
{
    textF_cName.setText(ref.clientName);
    textF_add.setText(ref.address);
    textF_cmpName.setText(ref.getCmpName());
    textF_desig.setText(ref.desig);
    textF_product.setText(ref.product);
    textF_phNo.setText(String.valueOf(ref.phNo));
    textF_mob.setText(String.valueOf(ref.mobileNo));
    textF_fax.setText(String.valueOf(ref.faxNo));
    textF_emailOff.setText(ref.email_off);
    textF_emailPer.setText(ref.email_per);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[])
{
/*
 * Set the Nimbus look and feel
 */
//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
/*

```

```

* If Nimbus (introduced in Java SE 6) is not available, stay with the
* default look and feel. For details see
* http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
*/
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(Dialog_ClientInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(Dialog_ClientInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(Dialog_ClientInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(Dialog_ClientInfo.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/*
 * Create and display the dialog
 */
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        Dialog_ClientInfo dialog = new Dialog_ClientInfo(new javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {

            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    }
});
// Variables declaration - do not modify
private javax.swing.JButton btn_ok;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;

```

```

private javax.swing.JLabel jLabel5;
private javax.swing.JLabel label_cName;
private javax.swing.JLabel label_cmpName;
private javax.swing.JLabel label_desig;
private javax.swing.JLabel label_phNo;
private javax.swing.JLabel label_product;
private javax.swing.JTextField textField_add;
private javax.swing.JTextField textField_cName;
private javax.swing.JTextField textField_cmpName;
private javax.swing.JTextField textField_desig;
private javax.swing.JTextField textField_emailOff;
private javax.swing.JTextField textField_emailPer;
private javax.swing.JTextField textField_fax;
private javax.swing.JTextField textField_mob;
private javax.swing.JTextField textField_phNo;
private javax.swing.JTextField textField_product;
private javax.swing.JLabel topBar;
// End of variables declaration
}

```

Dialog_diagnosResult.java

```

package salesPackage;

import javax.swing.UIManager;

public class Dialog_diagnosResult extends javax.swing.JDialog {

    public Dialog_diagnosResult(java.awt.Frame parent, boolean modal) {
        super(parent, modal);

        try{
            UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }catch(Exception e)
        { System.out.println("Windows isn't available"); }

        initComponents();

        this.setLocationRelativeTo(null); //Open at center of screen
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        btn_ok = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        lbl_log = new javax.swing.JLabel();
        lbl_reg = new javax.swing.JLabel();

```

```

lbl_db = new javax.swing.JLabel();
lbl_bk = new javax.swing.JLabel();
lbl_form = new javax.swing.JLabel();
lbl_vir = new javax.swing.JLabel();
log = new javax.swing.JLabel();
reg = new javax.swing.JLabel();
db = new javax.swing.JLabel();
bk = new javax.swing.JLabel();
format = new javax.swing.JLabel();
vir = new javax.swing.JLabel();
bg_topBar = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("SPA | Loading Alert");
setMaximumSize(new java.awt.Dimension(400, 280));
setMinimumSize(new java.awt.Dimension(400, 280));
setResizable(false);
getContentPane().setLayout(null);

btn_ok.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/ok_1.png"))); // NOI18N
btn_ok.setBorderPainted(false);
btn_ok.setContentAreaFilled(false);
btn_ok.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
btn_ok.setFocusPainted(false);
btn_ok.setPressedIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/ok_3.png"))); // NOI18N
btn_ok.setRolloverIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/ok_2.png"))); // NOI18N
btn_ok.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btn_okActionPerformed(evt);
    }
});
getContentPane().add(btn_ok);
btn_ok.setBounds(150, 210, 90, 31);

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel2.setText("The following are the errors found while loading the application :");
getContentPane().add(jLabel2);
jLabel2.setBounds(15, 55, 370, 15);

lbl_log.setText("Application status :");
getContentPane().add(lbl_log);
lbl_log.setBounds(60, 85, 320, 16);

lbl_reg.setText("Registration :");
getContentPane().add(lbl_reg);
lbl_reg.setBounds(60, 105, 320, 16);

lbl_db.setText("Database :");
getContentPane().add(lbl_db);
lbl_db.setBounds(60, 125, 320, 16);

lbl_bk.setText("Database BackUp :");

```

```

getContentPane().add(lbl_bk);
lbl_bk.setBounds(60, 145, 320, 16);

lbl_form.setText("Format File :");
getContentPane().add(lbl_form);
lbl_form.setBounds(60, 165, 320, 16);

lbl_vir.setText("Inbox :");
getContentPane().add(lbl_vir);
lbl_vir.setBounds(60, 185, 330, 16);

log.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png"))); // NOI18N
getContentPane().add(log);
log.setBounds(32, 85, 19, 15);

reg.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png"))); // NOI18N
getContentPane().add(reg);
reg.setBounds(32, 105, 19, 15);

db.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png"))); // NOI18N
getContentPane().add(db);
db.setBounds(32, 125, 19, 15);

bk.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png"))); // NOI18N
getContentPane().add(bk);
bk.setBounds(32, 145, 19, 15);

format.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png"))); // NOI18N
getContentPane().add(format);
format.setBounds(32, 165, 19, 15);

vir.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png"))); // NOI18N
getContentPane().add(vir);
vir.setBounds(32, 185, 19, 15);

bg_topBar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/lodResult_topBar.png"))); // NOI18N
getContentPane().add(bg_topBar);
bg_topBar.setBounds(0, 0, 400, 280);

pack();
}// </editor-fold>

private void btn_okActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //this.setVisible(false);
    dispose();
}

```

```

void getStates(Diagnostic d)
{
    if(d.logFlag== -1)
    {
        log.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_error.png")));
        lbl_log.setText( lbl_log.getText()+" Corrupted");
    }else if(d.logFlag==32 || d.logFlag==33)
    {
        log.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_alert.png")));
        lbl_log.setText("One of the process was not completed during last run");
    }else if(d.logFlag==31 || d.logFlag==0)
    {
        log.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_alert.png")));
        lbl_log.setText("Application was not closed properly during last run");
    }else
    {
        log.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png")));
        lbl_log.setText( lbl_log.getText()+" Ok");
    }

    if(d.configFlag==0)
    {
        reg.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_alert.png")));
        lbl_reg.setText( lbl_reg.getText()+" Corrupted");
    }else
    {
        reg.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png")));
        lbl_reg.setText( lbl_reg.getText()+" Done");
    }

    if(d.dbFlag<1)
    {
        db.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_error.png")));
        lbl_db.setText( lbl_db.getText()+" Not Connected");
    }else
    {
        db.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png")));
        lbl_db.setText( lbl_db.getText()+" Connected");
    }

    if(d.backupFlag==0)
    {
        bk.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_alert.png")));

```

```

        lbl_bk.setText( lbl_bk.getText()+" Not Found / Not created yet");
    }else
    {
        bk.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png")));
        lbl_bk.setText( lbl_bk.getText()+" Found");
    }

    if(d.formateFlag==0)
    {
        format.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_alert.png")));
        lbl_form.setText( lbl_form.getText()+" Not Found. This file is a part of installation.");
    }else
    {
        format.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png")));
        lbl_form.setText( lbl_form.getText()+" Present");
    }

    if(d.virtualFlag>0)
    {
        vir.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png")));
        lbl_vir.setText( lbl_vir.getText()+" "+d.virtualFlag+" Records Found! Please save them into the
database!");
    }else
    {
        vir.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/salesPackage/Images/bullet_ok.png")));
        lbl_vir.setText( lbl_vir.getText()+" No Records Found");
    }
}

public static void main(String args[]) {
/*
 * Set the Nimbus look and feel
 */
/*<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
/*
 * If Nimbus (introduced in Java SE 6) is not available, stay with the
 * default look and feel. For details see
 * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
*/
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {

```

```

        javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Dialog_diagnosResult.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Dialog_diagnosResult.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Dialog_diagnosResult.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Dialog_diagnosResult.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {
    Dialog_diagnosResult dialog = new Dialog_diagnosResult(new javax.swing.JFrame(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {

        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
    dialog.setVisible(true);
}
});

// Variables declaration - do not modify
private javax.swing.JLabel bg_topBar;
private javax.swing.JLabel bk;
private javax.swing.JButton btn_ok;
private javax.swing.JLabel db;
private javax.swing.JLabel format;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel lbl_bk;
private javax.swing.JLabel lbl_db;
private javax.swing.JLabel lbl_form;
private javax.swing.JLabel lbl_log;
private javax.swing.JLabel lbl_reg;
private javax.swing.JLabel lbl_vir;
private javax.swing.JLabel log;
private javax.swing.JLabel reg;
private javax.swing.JLabel vir;
// End of variables declaration
}

```