

Extraction of Contact List from CDR



By

NC Hina Aziz (Group Leader)

NC Amna Feroze

NC Aida Zahid

Submitted to the Faculty of Computer Science
National University of Sciences and Technology, Rawalpindi in partial fulfillment for the
requirements of a B.E Degree in Computer Software Engineering

June 2014

CERTIFICATE

Certified that the contents and form of project report entitled “**Extraction of Contact List from CDR (Call Detail Record)**” submitted by Hina Aziz, Amna Feroze, and Aida Zahid have been found satisfactory for the requirement of the degree.

Supervisor: _____

Dr. Seemab Latif

Department of CS

Military College of Signals

ABSTRACT

Modern Technology has revolutionized the way we communicate. The most widely used communication mechanism of modern times is the cellular networks. The cellphones used are supported with multiple services which include a way to save your contacts in your phone. But in-case your phone suffers any tragedy for example gets lost, you can be in trouble if you need to contact someone. Database queries and techniques generally used by telecom companies to retrieve ones contacts are not efficient due to large data. Analysis performed on **25 GB** call details record takes **more than 75 minutes** to extract any user's contacts from database.

The project is to focus on the user's personal data and CDR, and develop an application which can discover contact list of a particular subscriber, given the CDR for a certain period of time. In this project we are employing modern algorithms from the areas of data-mining and their parallel implementations to devise solutions which can do the job of extracting contact-lists efficiently. Efficiency and high-performance is a key goal in this project because only then the said application will be a solution that can be acceptable to the big telecommunications service providers. The software is intended to generate contact list from CDR (Call Detail Record).

The works focuses on databases searching mechanism. To perform efficient searching, Searching is done on the bases of parameters which are faster than others. Clustered indexes are defined on the database as to save data in order that it's easier to retrieve. The whole data is divided into different file groups so that while searching instead of traversing whole data only limited portion of data is scanned. Normalization and de-normalization operations are also performed on tables to improve performance. After performing all these operation on database, Parallel implementation algorithm is developed to access database simultaneously by making several connections in parallel to database. These techniques improved performance in a way that the time taken to extract any user's contact on **25 GB** data is reduced to average 2 minutes.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To all the reformed institutions of Pakistan, dedicated to Allah and his kingdom and,

In particular to the faculty, staff and students of Military College of Signals

ACKNOWLEDEMENTS

We are grateful to our parents for their unwavering faith in us, their continuous support and love without which we would not have been able to succeed.

We are extremely grateful to our project supervisor Dr. Seemab Latif from MCS who in addition to providing valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project. We are highly thankful to all of our teachers and staff of MCS who supported and guided us throughout our course and research work. Their knowledge, guidance and training enabled us to carry out this research work.

We would like to thank Dr. Zaki Murtaza from MCS for his kind help and support for our project.

In the end we would like to acknowledge the support provided by all our friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

Table of Content

1. Introduction.....	9
1.1 Introduction	9
1.2 Background.....	9
1.3 Problem Statement.....	10
1.4 Objectives.....	10
1.5 Deliverables.....	11
1.6 Technological requirements	11
2. Literature Review	12
3. System Requirements Specifications.....	16
3.1 Introduction	16
3.2 Scope	16
3.3 Product Perspective	18
3.4 Basic Assumptions	18
3.5 System Feature	19
3.5.1 Extract Contacts	19
3.5.2 Search.....	21
3.5.3 Email	22
3.5.4 Help.....	24
3.6 Nonfunctional requirements	25
3.6.1 Performance Requirements	25
3.6.2 Safety Requirements	25
3.6.3 Security Requirements	26
3.7 Software Quality Attributes.....	26
4. System Design and Development.....	28
4.1 Introduction	28
4.2 Architectural Model.....	28
4.3 Design Pattern	30
4.4 System Flow Diagram	32
4.5 Detailed Design	33
4.5.1 Database Diagram.....	33
4.5.1.1 Entity Relationship Diagram	33
4.6 Static View	36
4.6.1 System Use Case Diagram.....	36
4.6.2 Login Use Case	37
4.6.3 Extract Contacts Use Case Diagram	39
4.6.4 Search Use Case.....	42
4.6.5 Use Case Description	42
4.6.6 Email Use Case	44
4.7 CDR Class Diagram	47
4.7.1 Main Menu Class	48
4.7.2 Authenticator_DAO_interface Class.....	49
4.7.3 Authenticator_DAO_implementation Class	49
4.7.4 Connection Class.....	50
4.7.5 Extract of contact List_Dao_interface Class	50
4.7.6 Extraction of contact list_DAO_implementation Class	51
4.7.7 Search_DAO_Interface Class	51
4.7.8 Search_DAO_Implementation Class	52
4.7.9 Help Class	52
4.7.10 Table View_DAO_interface Class.....	53
4.7.11 Table View_DAO_imlementation Class.....	53
4.7.12 Email Sender_DAO_interface Class	53
4.7.13 Email Sender_DAO_implementation Class.....	54

4.8	Generator Class Diagram	54
4.8.1	Records Generator Class	55
4.8.2	Records Class	56
4.9	Dynamic View	57
4.9.1	Activity Diagrams	57
4.9.2	Sequence Diagrams	62
4.9.3	Data Flow Diagram	67
4.9.4	Human Interface Design	68
5.	System Implementation	77
5.1	Implementation introduction	77
5.2	Tools and Technologies	77
5.2.1	System Specification	77
5.3	Implementation	78
5.3.1	Contact List extraction	78
5.3.2	Search	80
5.3.3	Email	81
5.3.4	Help	82
5.4	Application Issues	82
6.	Project Analysis and Evaluation.....	82
6.1	Testing	82
6.2	Testing Levels	82
6.2.1	Unit Testing.....	83
6.2.2	Integration Testing	90
iii.	Integration Technique to be used:	99
iv.	Test configuration set-up:	100
6.2.3	System Testing	105
6.2.4	Acceptance Testing	111
6.3	Summary.....	122
6.4	Result and Analysis	122
6.4.1	Results and Analysis Introduction.....	122
6.4.2	Results.....	122
7.	Conclusion and Future Work	124
	Appendix A:Glossary	125
	Bibliography.....	130

List of Figures

Figure 2-1: Sparse Matrix	16
Figure 4-1: Data Centric Architecture	29
Figure 4-2: System Flow Diagram.....	32
Figure 4-3: Entity Relationship Diagram.....	33
Figure 4-4: Operator	34
Figure 4-5: CDR	35
Figure 4-6: Personal_info	35
Figure 4-7 : System Use case.....	36
Figure 4-8 : Login Use Case	37
Figure 4-9 : Extract Contacts Use Case	39
Figure 4-10 : Search Use case.....	42
Figure 4-11: Email Use Case	44
Figure 4-12 : CDR Class Diagram.....	47
Figure 4-13: Generator Class Diagram	54
Figure 4-14: Main Page Activity Diagram	57
Figure 4-15: Extract Contacts Activity Diagram	58
Figure 4-16: Search Activity Diagram.....	59
Figure 4-17: Email Activity Diagram	60
Figure 4-18: Help Activity Diagram.....	61
Figure 4-19: Authenticator Sequence Diagram	62
Figure 4-20: Main Page Sequence Diagram	63
Figure 4-21: Extract Contacts Sequence Diagram	64
Figure 4-22: Search Sequence Diagram.....	65
Figure 4-23: Email Sequence Diagram.....	66
Figure 4-24: Help Sequence Diagram.....	67
Figure 4-25: Data Flow Diagram	67
Figure 4-26: Login Page	68
Figure 4-27: Main Page	69
Figure 4-28: Extract Contacts	70
Figure 4-29: Tabular Output	71
Figure 4-30: File Page	72
Figure 4-31: Email Sender	73
Figure 4-32: Search Page (a).....	74
Figure 4-33: Search Page (b)	75
Figure 4-34: Print Page	75
Figure 4-35 : Help.....	76
Figure 5-1: System Methodology	79
Figure 6-1: Integration testing (a)	94
Figure 6-2 Integration testing (b).....	94
Figure 6-3 Integration testing (c)	94

1. Introduction

1.1 Introduction

The project is to focus on the user's personal data and CDR, and develop an application which can discover contact list of a particular subscriber, given the CDR for a certain period of time. We can employ modern algorithms from the areas of data-mining, pattern-recognition and their parallel implementations to devise solutions which can do the job of extracting contact-lists efficiently.

Efficiency and high-performance is a key goal in this project because only then the said application will be a solution that can be acceptable to the big telecommunications service providers. For the purpose of this project we developed our own random CDR generator that contains millions of records. Information generated by Call Detail Records includes time, date, call duration, number dialed, caller ID information, extension, call status will be useful to a business providing insight into phone use, employees, sales and traffic trends, and so much more. We will save the person contact list in file and send it to person. Others features that are required to implement is security that only administrators are allowed to use this CDR as its contains callers and called people personal details e.g. name, national identity card number so that nobody other than administrator can view it. Printing and Help features are also available. Search feature to find that whether user is authorized user and registered sim holder.

1.2 Background

Today's rapidly changing business environment is creating intense competition among corporations. Markets are changing faster now than in any other time in history. Product life cycles are shortening and businesses must compete globally. Today's market conditions therefore require that managers evaluate cellular technology as an alternative to traditional wireline telecommunications services to make their corporations more competitive. The services offered

range from basic cell-phone call to other services like sms, saving contact list etc. While providing the users with all these services, the service providers generate huge amount of data. The very primitive forms are user's data records (name, address, sim# etc) and call-data records or CDR. Both of these records is necessary to provide communication services to the users. A cell-phone service provider company can potentially generate multiple tera-bytes of this data every day, especially the CDR. Efficiency is necessary in this area .We need an algorithm that will search database efficiently .Most of the user traverses entire data to get any particular record or entry but that particular entry or record has a field that differentiates it from other records so we developed an algorithm that separates the data on the basis of that special field for example Network name in our case to avoid traversal entire data and also use parallel searching approach to efficiently use system resources and reduce searching time as much as possible.

1.3 Problem Statement

Many different software's are available in the market for the purpose of retrieval of contacts from CDR .Our software is focused on the achievement of efficiency which will be our main target. With the kind of data we have described i.e. in the order of Tera-bytes every day, this problem becomes significantly complex. It takes huge amount of time for the extraction of data which the previous software's took. Our product intends to make this task easier. Next we intend to generate the entire contact list of a user. We employ modern algorithms from the areas of data-mining in general and their parallel implementations to devise solutions which can do the job of extracting contact-lists efficiently.

1.4 Objectives

The objective our project is to design a system that enables a user to extracts Contact List from Call Detail Record (CDR) from their respective telecommunication companies, given the CDR

for a defined period of time. The software would allow an authorized operator to perform all the functionalities.

Our software is focused on the achievement of efficiency which will be our main target.

The software is intended to retrieve records from huge amount of data i.e. millions of records in minimum time possible. After the data has been retrieved the Contact List of an individual is to be generated. For that purpose we have employed various algorithms and techniques.

1.5 Deliverables

- i. First Progress Report: including SRS Document
- ii. Second Progress Report: including System Design (Hardware and Software)
- iii. Third Progress Report: Demo of integrated system
- iv. Final Report: including complete documentation of the system and user manual.

1.6 Technological requirements

Operating Systems

1. Windows

Software Packages

1. Java Netbeans 8
2. SQL Server 2008 R2
3. SceneBuilder

Hardware Components

1. Personal Computer(s) with minimum 4gb RAM
2. Hard-drive

2. Literature Review

A Call Detail Record (CDR) is a record that is produced by the system and contains the details of calls that have terminated on the system. CDRs include the date and time the call started, the number that made the call, the number that received the call, and the call duration. [1]

Field	Example
Domain	company.com
primary call id	0080-1008-25C105DA-0@D141DFC50C7AA2248
call id	0080-1008-25C105DA-0@D141DFC50C7AA2248
From	“00432638316014” <sip:+432638316014@company.com;user=phone>
To	“Steve Fuller” <sip:411@company.com>
start time	20100310095708
connect time	20100310095710
end time	20100310095720
IP address	175.23.145.1

Table 2-1

We need to know how to control Call Detail Records (CDR) generation and what information is required to save in CDR. The organizational needs of CDR and commands that are used for this process. With an appropriate configuration, we can send corresponding CDRs of outgoing and incoming calls as e-mail. Bear in mind that the mail server must be configured [2]

We need an efficient data storage and retrieval for our project to extract contact list. For that we have to employ some algorithms that can enhance our storage and extraction process. First we need to apply certain optimization techniques to efficiently store data. Book” High performance

MY SQL” teaches how to build fast, reliable systems. It teaches us how to design schemas, query indexes.[3]

The SQL Server 11.0 release (code named “Denali”) introduces a new data warehouse query acceleration feature based on a new type of index called the *column store*. This new index, combined with enhanced query processing features, improves data warehouse query performance by hundreds to thousands of times in some cases, and can routinely give a tenfold speedup for a broad range of decision support queries [4].

For better query results we can also use table partition techniques that are well described by Pinal Dave in his blog which he explains how to partition your table horizontally to increase query performance. He explains partitioning table is very simple and very efficient when used with different file groups in different tables [5]

The extraction of information about behavior of individuals and communities is ongoing in recent years by using new devices to prepare life logs. An offered method for making the social network by using the information of mobile phones which are registered chronologically in telecommunication corporations. Preparing the connection graph by this method is possible as well.

Telephone Call Detail Records (CDRs) are one important source of digital evidences that can identify suspects and their partners. Law enforcement authorities may intercept and record specific conversations with a court order and CDRs can be obtained from telephone service providers. However, the CDRs of a suspect for a period of time are often fairly large in volume. To obtain useful information and make appropriate decisions automatically from such large amount of CDRs become more and more difficult. Current analysis tools are designed to present

only numerical results rather than help us make useful decisions. In this paper, an algorithm based on fuzzy decision tree (FDT) for analyzing CDRs is proposed.

Our fuzzy decision tree algorithm for CDRs analyses introduce in the following is based on some of these algorithms.

A fuzzy decision tree takes the fuzzy information entropy as heuristic and selects the attribute which has the biggest information gain on a node to generate a child node. The nodes of the tree are regarded as the fuzzy subsets in the decision-making space. The whole tree is equal to a series of “IF THEN “rules. Every path from the root to a leaf can be a rule. The precondition of a rule is made up of the nodes in the same path, while the conclusion is from the leaves of the path. [6].

We will also look to implement already available algorithms to expose network of people who are in contact with each other through x-number of people in between which is our additional goal. Data about activities of individuals by new technology are collected and estimated. Individuals who are frequently found together are likely to be associated with each other. Several models for constructing social network from events are proposed. Finding subgroups and subsets of criminal groups such as drugs and money laundering or terrorism, find suspects are stated in these researches [7] [8]. The social networks are investigated in different groups and media and used to recognize and distinguish more strong or more weak ties between members.

CDR files (call detail records) are made by any switching system to prepare the billing information about subscribers such as call duration, originating and terminating numbers (A-numbers and B-numbers), furthermore origin and destination physical coordination are recorded. These files are worth investigation which can be used to prepare the connection graphs between subscribers that can be termed in social network as actors. So by transforming the latter files to

text files and then changing them to a matrix it can be easily linked to the graph theory's fundamentals.

A-number	B-number	Start time	End time	Call duration
number1	number2	8:51:30	8:55:12	0:03:42
number2	number5	8:55:03	9:10:34	0:15:31
number3	number2	9:00:52	9:05:34	0:04:42
number4	number5	9:02:16	9:08:44	0:06:28
number5	number3	9:05:55	9:07:45	0:01:50

Table 2-2

From such tables a lot of information can be obtained about social networking. The call duration column can be suppose as weight of graphs and every subscriber (actor) which is demonstrated by her/his MSISDN (Mobile Station Integrated Service Digital Network Number) can be assumed as nodes or vertices. The table can be converted to a matrix or sparse matrices and then can be consumed for establishing the social network graph.

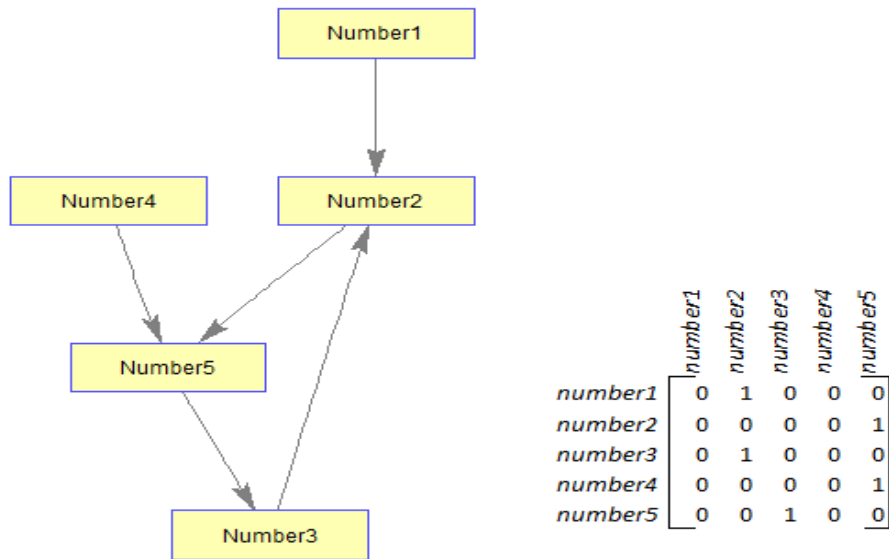


Figure 2-1: Sparse Matrix

3. System Requirements Specifications

3.1 Introduction

This chapter describes the requirements specifications for Extraction of Contact List from CDR (call detail records). The product is a system that allows an operator in a telecommunication organization to extract contact list of a particular individual on the basis of mobile number.

This chapter provides specifications that Extraction of Contact List from CDR (call detail records) has to fulfill so it can perform in an efficient manner.

3.2 Scope

The aim of this project is to focus on the user's data records and CDR, and develop an application which can create contact list of a particular subscriber, given the CDR for a defined period of time.

Objectives:

- i. To create contact list of a cell-phone user, from the CDR.
- ii. To achieve high-performance levels on the processing of the Big-data in the form of CDR. The goal is to generate a contact list for a number, from a sizable record of CDR in least amount of time possible.
- iii. To create a storage modeled on basic data-warehousing concept to store typical CDR and user's data records.
- iv. To use/modify/update/develop the algorithms to efficiently do data-mining over the input data.
- v. Integration of any third-party tools from vendors like Oracle/IBM/Open source implementations of Map Reduce etc which can help in data mining.
- vi. Provides a visual IDE (Interactive Design Environment) enabling the visualization of complex data formats that help the business understand new insights that may be possible
- vii. Ensures compliance with data privacy regulations and organization-wide data privacy mandates to dynamically mask sensitive production data from unauthorized access

End Goal:

The End Goal of this Project is to develop an application, which creates contact-lists of a specific user given an instance of CDR and data records bank of a service provider.

Additional Goal:

We will also look to implement already available algorithms to expose network of people who are in contact with each other through x-number of people in between.

The software is intended to be a lot beneficial for the telecommunication companies. In the field of telecommunications, to succeed in the goal of providing the users with all the services like marketing, billing, business intelligence a vital role is played by the use of call detail records (CDRs). The purpose of generating the complete contact list is one of the many desires of the customers.

3.3 Product Perspective

Our product intends to make the task of extracting contacts efficiently which the previous software's were unable to do. Our software generates the entire contact list of a user. We can employ modern algorithms from the areas of data-mining in general and especially in very graph processing algorithms and their parallel implementations to devise solutions which can do the job of extracting contact-lists efficiently.

Our project comprises of three major parts.

1. Extracting data from CDR
2. Data Generation
3. Identifying Patterns (Additional Goal)

3.4 Basic Assumptions

The system will have uninterrupted power 24/7 since the system is supposed to be active 24/7. The system must also be able to perform ably in all sorts of weather conditions and

terrain/background. Also the user of the system should have basic knowledge of computers and how to operate them.

3.5 System Feature

3.5.1 Extract Contacts

3.5.1.1 Extract Contacts

The user would be able to retrieve the contact list on the basis of mobile number entered through user interface the result will be displayed on screen. The user has the option to sort the list according to frequency/date. The output format can be either in the form of a table or an excel file.

3.5.1.2 Stimulus/Response Sequences

Normal Flow

Stimulus: User Clicks on the Extract contact Button.

Response: The system displays next window prompting user to enter domain, mobile number and choose the output format Tabular/File the user also has to choose the sequence of output to be either frequency or of date.

Stimulus: User Enters domain, mobile number and selects frequency and tabular and clicks on extract contacts.

Response: The system displays next window showing output of contact list in tabular form the result would be sorted according to frequency of calls made.

Stimulus: User Enters domain, mobile number and selects frequency and File and clicks on extract contacts

Response: The system will save the contact list in the form of an excel file result sorted according to frequency.

Stimulus: User Enters domain, mobile number and selects Date and tabular and clicks on extract contacts.

Response: The system displays next window showing output in tabular form the result would be sorted according to date.

Stimulus: User Enters domain, mobile number and selects Date and File and clicks on extract contacts.

Response: The system will save the contact list in the form of an excel file result sorted according to date.

Stimulus: User clicks on print

Response: The system will print the contact list of the customer.

Stimulus: User Clicks on the Email Button.

Response: The system displays next window showing email sender.

Stimulus: User Enters Id, password, Customer Id, Subject Message attaches excel file and clicks on send message

Response: The email will be send to customer.

Alternate Flow

Stimulus: User clicks in back button

Response: The main page will be displayed.

3.5.1.3 Functional Requirements

REQ-1: Software will allow valid user to retrieve the contact list.

REQ-2: Software will accept numeric entry of the mobile format i.e 11 digit number.

3.5.2 Search

3.7.2.1 Description and Priority

The user can search the CDR (call detail record) for the existence of a particular mobile number.

3.5.2.2 Stimulus/Response Sequences

Normal Flow

Stimulus: User Clicks on the Search Button.

Response: The system displays next window prompting user to enter domain, mobile number.

Stimulus: User Enters domain, mobile number and clicks Search.

Response: The system displays result of name and NIC if the person exists in database else displays person not found.

Alternate Flow

Stimulus: User clicks on back button

Response: The main page will be displayed.

3.5.2.3 Functional Requirements

REQ-1: Software will allow valid user to search the CDR.

REQ-2: Software will accept numeric entry of the mobile format i.e 11 digit number.

3.5.3 Email

3.5.3.1 Description and Priority

User can mail the contact list retrieved to the customer via email through his Gmail/Hotmail account.

3.5.3.2 Stimulus/Response Sequence

Normal Flow

Stimulus: User Clicks on the Email Button.

Response: The system displays next window showing email sender.

Stimulus: User Enters Id, password, Customer Id, Subject Message attaches excel file and clicks on send message

Response: The email will be send to customer.

Alternate Flow

Stimulus: User clicks on back button

Response: The main page will be displayed.

3.5.3.3 Functional Requirements

REQ-1: Software will allow valid user to email the contact list to the customer.

REQ-2: Software will allow the email id of operator to be either of Gmail/Hotmail and should not have two step verification.

REQ-3: The contact list would be extracted before it is mailed.

3.5.4 Help

3.5.4.1 Description and Priority

The software provides help feature which includes instructions on how to use the software.

3.5.4.2 Stimulus/Response Sequence

Normal Flow

Stimulus: User Clicks on the Help Button.

Response: The system displays next window showing options to choose what information you require.

Stimulus: User Enters Extract Contacts.

Response: The system displays Help on how to use the extract contact functionality.

Stimulus: User Enters Search.

Response: The system displays Help on how to use the Search functionality.

Stimulus: User Enters Email.

Response: The system displays Help on how to use the Email functionality.

Alternate Flow

Stimulus: User clicks on back button

Response: The main page will be displayed.

3.5.4.3 Functional Requirements

REQ-1: Software will allow valid user to view help.

3.6 Nonfunctional requirements

3.6.1 Performance Requirements

- i. Responses to queries of system navigation shall take no longer than 10 seconds to load onto the screen after the user submits the query.
- ii. Search and Extraction queries are dependent on size of data. With our current 25GB data these queries shall take no longer than an average of 1 minute.
- iii. The system shall display confirmation messages to users almost instantly on submission of information by the user.

3.6.2 Safety Requirements

The software will not affect any database outside of its server nor will it affect any other hardware or software installed on the system.

3.6.3 Security Requirements

3.6.3.1 There will be a proper login mechanism for users involving user name and password.

3.6.3.2 The system will not allow a user with incorrect username and password to perform an action.

3.6.3.3 The CDR information like mobile numbers, details of call information shall remain classified to all.

3.6.3.4 Prevention of unauthorized write/delete access.

3.7 Software Quality Attributes

Availability: The system shall be available all the time.

Adaptability: The system shall be able to support all versions of SQL we are working in SQL 2008 R(2) other versions are also compatible ; few changes in the code might be required for connectivity.

Backup: This functionality is not applicable for our software as the data storage facility would be the responsibility of the telecommunication companies.

Capacity Requirements: The system will be able to store the data of millions of records.

Compatibility: The system will be compatible with operating systems like windows, older versions of these operating systems. The system will support all screen resolutions. The system should be compatible with third party systems which include telecommunication company software's.

Competitive Edge: The Software provides great efficiency which was not achieved by previously existing software.

Compliance: Not applicable

Documentation: A user manual would be provided along with the software.

Efficiency: The software shall be able to retrieve contact list of the individual within least possible time.

Flexibility: The layout/architecture of the software will be flexible enough for some later requirements change or application enhancement.

Interoperability: Syntactic interoperability is provided through SQL .The software gain access to the data provided in the database and extracts the contact list from the data base of the telecommunication company.

Maintainability: The software shall be easily maintainable in case of change in requirement or a bug.

Reliability: If the system goes down due to some fault or error it should be able to recover to its current state, System comprises of extensive error handling mechanism.

Robustness: If the connection between the user and the system is broken prior to a request being either confirmed or canceled, the System shall enable the user to recover an incomplete request.

Scalability: The software shall be able to handle one operator at a time the software shall be able to handle data of millions of records.

Start-up Time: Not applicable

Storage: The contact list generated are saved in file format and mailed to the customer.

Stress: The software shall be able to work efficiently with huge amount of data such as of millions of records.

Timeliness: Not applicable.

Usability: The system will have a simple user interface that will require minimal amount of training.

4. System Design and Development

4.1 Introduction

This chapter describes the design specifications of Extraction of Contact list from CDR.

The design specifications have been developed using requirements described in chapter 3.

This chapter provides details about system structure and architecture.

4.2 Architectural Model

This section provides a detailed and comprehensive architectural overview of the system.

The contact list retrieval software is broken down into two main components: a User-side java software and a database.

The User-side application is also separated into two parts: the functional component (written in Java), and the graphical component .The functional component forms the core of our software. It receives user input and constructs contact list of customer, create an

excel file, create tabular result, print the contact list, provide email functionality. It performs all the business logic required to retrieve the contacts and perform the required operations by the operator. The graphical component, as the name indicates is simply the graphical user interface. It provides all of the buttons, text boxes, and other on-screen elements which allow the user to access all of the features provided by the software.

The Database Centric Architectural Pattern is used for this project. Database Centric is used because of the project nature which involves the use of database as a crucial element. The software requires the use of database centric language i.e. SQL which is required for data extraction from the database.

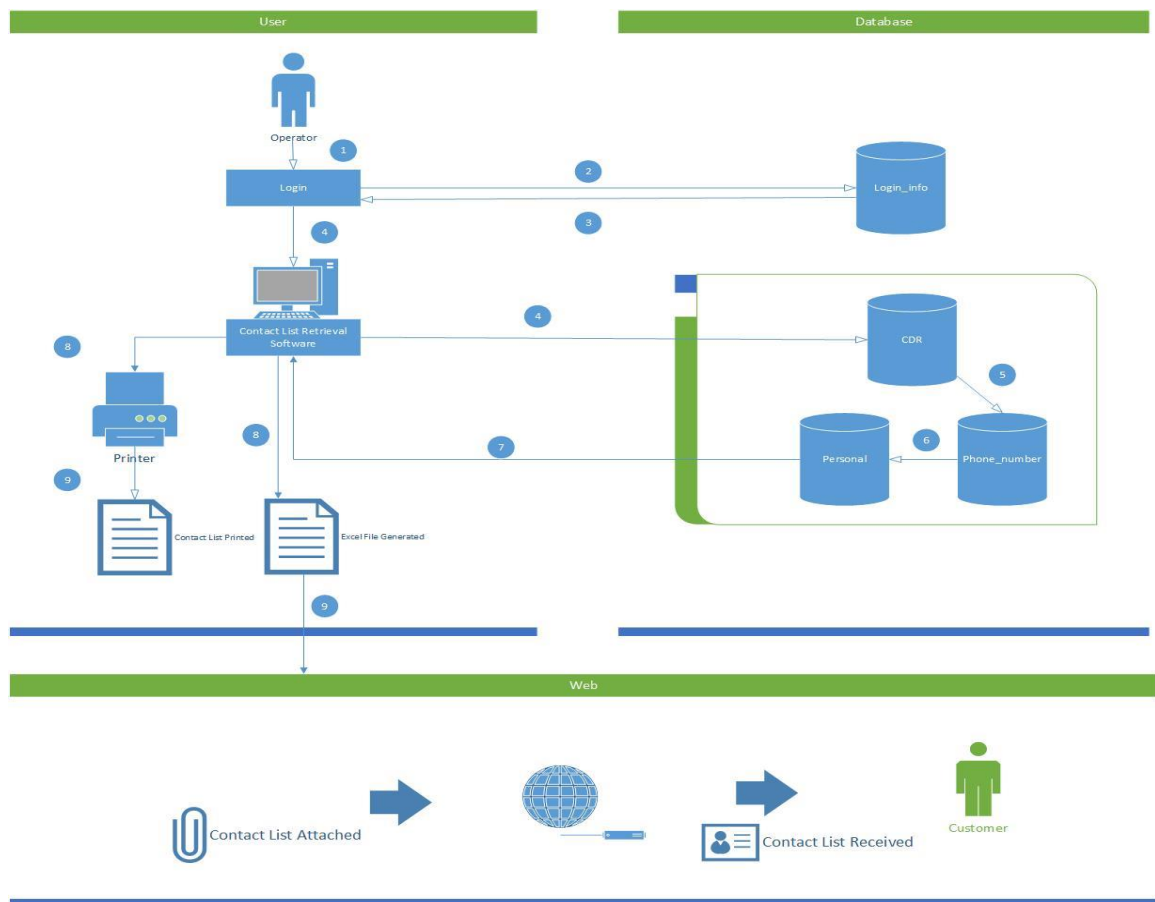


Figure 4-1: Data Centric Architecture

4.3 Design Pattern

The design pattern used here is **FETCH PATTERN** with **DAO (Data Access Object)**

Fetch Pattern

Fetch pattern is used because we have a database in which denormalization is used as value from one table is copied to another table for that purpose we have two tables one with the Name and NIC and the other involving NIC and Phone Number. NIC is used as a Foreign key.

Data Access Object

Along with this we use a Data Access Object (DAO) to abstract and encapsulate all access to the data source. The DAO manages the connection with the data source to obtain and store data. The DAO implements the access mechanism required to work with the data source. The data source is a persistent store i.e RDBMS, The business component that relies on the DAO uses the simpler interface exposed by the DAO for its clients.

DAO pattern is based on abstraction and encapsulation design principles and shields rest of application from any change on persistence layer e.g. for now we are using SQL server 2008 but later we might want to change that because of better performance of a newer version or totally a different database like Oracle. It also saves from change of persistence technology e.g. from Database to Filesystem.

The advantage of using DAO design pattern is that to access database, it would be relatively safe as we only need to make change on Data Access Layer. DAO design pattern also keeps coupling low between different parts of application. By using DAO design pattern our View Layer is completely independent to DAO layer and only Service layer has dependency on it which is also abstracted by using DAO interface.

4.4 System Flow Diagram

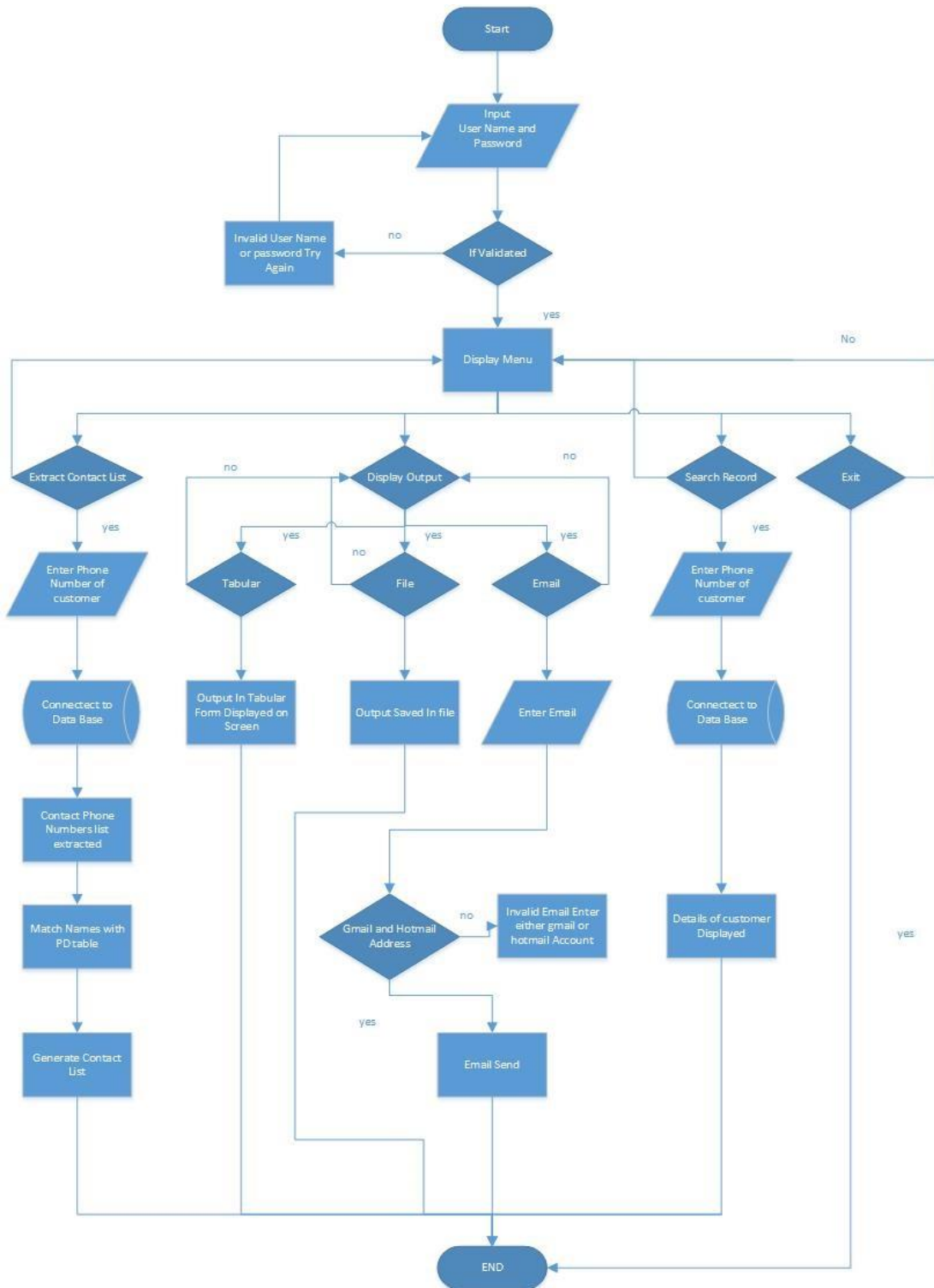


Figure 4-2: System Flow Diagram

4.5 Detailed Design

4.5.1 Database Diagram

4.5.1.1 Entity Relationship Diagram

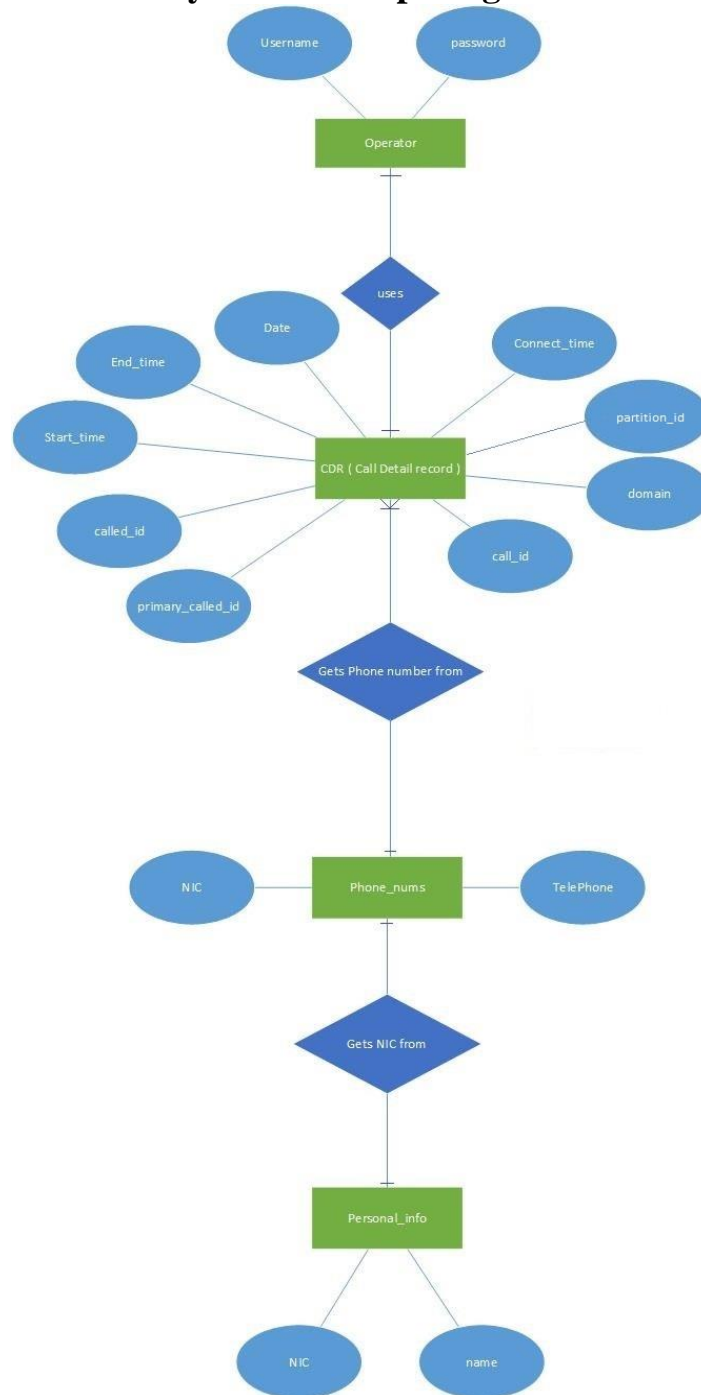


Figure 4-3: Entity Relationship Diagram

Entities & Attributes

Operator

- a) Username
- b) Password

Sr.no	Field Name	Data type	Size
1	username	varchar	255
2	password	varchar	255

Figure 4-4: Operator

CDR (Call Detail Record)

- a) Date
- b) End_time
- c) Start_time
- d) Connect_time
- e) Domain
- f) Call_id
- g) Primary_called_id
- h) Caller_id
- i) Partition_id

Sr.no	Field Name	Data type	Size
1	Call_ID	int	4
2	Domain	varchar	255
3	Primary_caller_id	varchar	255
4	Called_id	varchar	255
5	Start_time	varchar	255
6	Connect_time	varchar	255
7	End_time	varchar	255
8	Date	date	3
9	Partition_id	int	4

Figure 4-5: CDR

Personal_info

- a) NIC
- b) f_name

Sr.no	Field Name	Data type	Size
1	f_name	varchar	255
2	NIC	varchar	255

Figure 4-6: Personal_info

Phone_nums

- a) Telephone
- b) NIC

Sr.no	Field Name	Data type	Size
1	NIC	varchar	255
2	telephone	varchar	255

Table 4-1 : Phone_nums

4.6 Static View

4.6.1 System Use Case Diagram

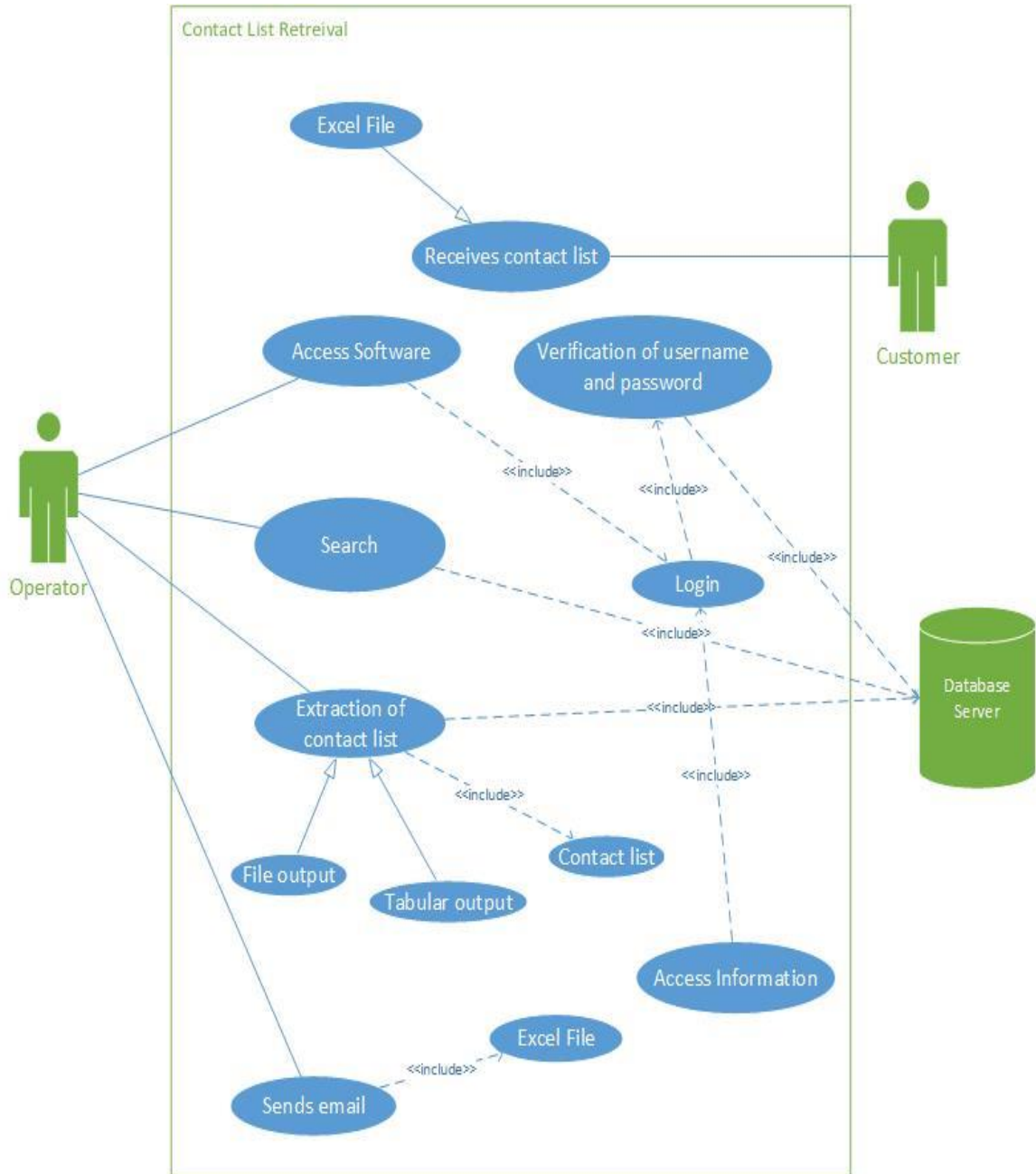


Figure 4-7 : System Use case

4.6.2 Login Use Case

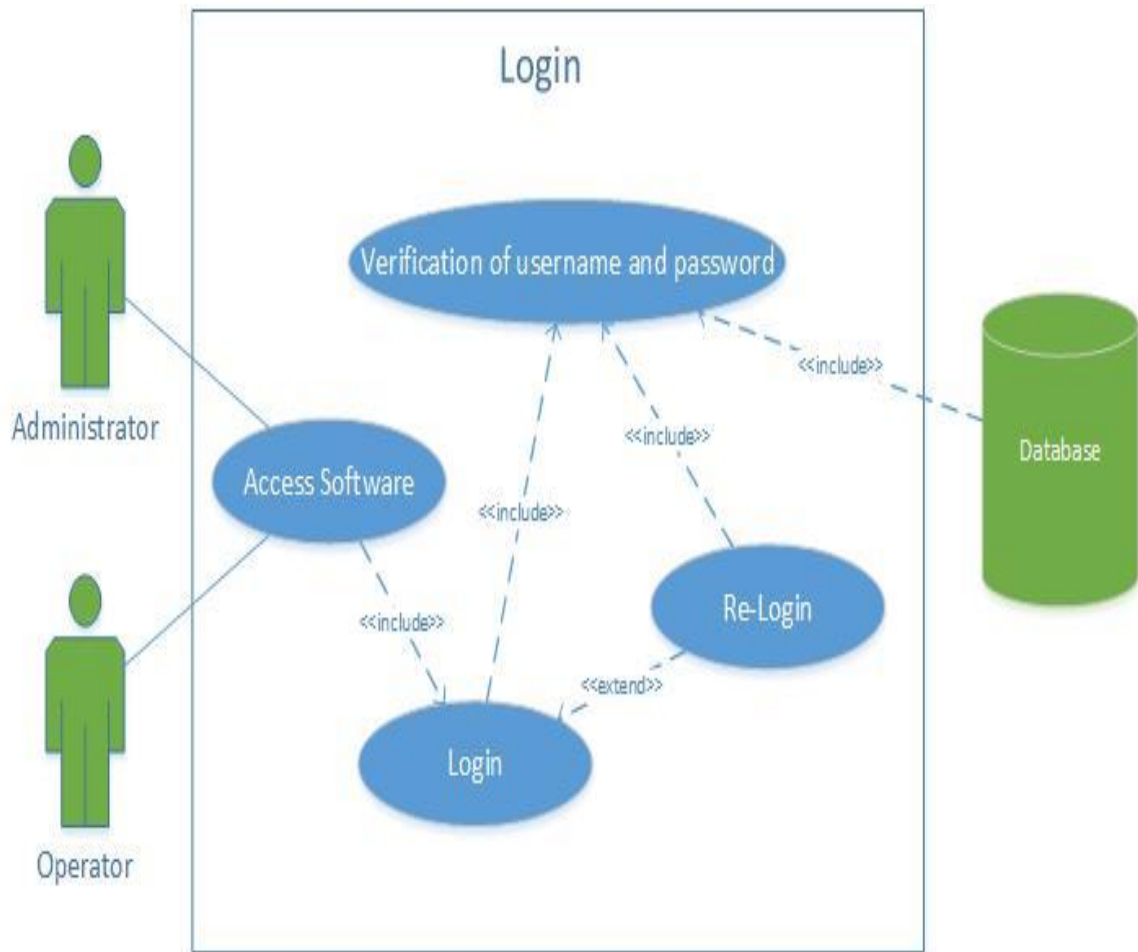


Figure 4-8 : Login Use Case

4.6.2.1 Use Case Description

This use case describes how the user logs-in to avail the functionalities provided by Contact list retrieval.

4.6.2.2 Actors

- a) Operator
- b) Administrator
- c) Database

4.6.2.3 Pre-Conditions

User must have the software installed on his PC in the telecommunication organization.

4.6.2.4 Basic Flow of Events

- a) User starts the Contact list retrieval software.
- b) System prompts the user to enter credentials.
- c) User provides his credentials (username and password) if he/she is a valid user and chooses to log-in.
- d) Credentials are sent to the database for authentication.
- e) User gets logged-in if authenticated

4.6.2.5 Alternate Flow

Invalid Credentials: System asks the user for re-login, with a warning that indicates that either the user name or password is incorrect.

4.6.2.6 Post-Conditions

a) Successful Completion

User is logged-in. Now user can use all features provided by Contact list retrieval software depending on his/her privilege level.

b) Unsuccessful Completion

User is not logged-in.

4.6.3 Extract Contacts Use Case Diagram

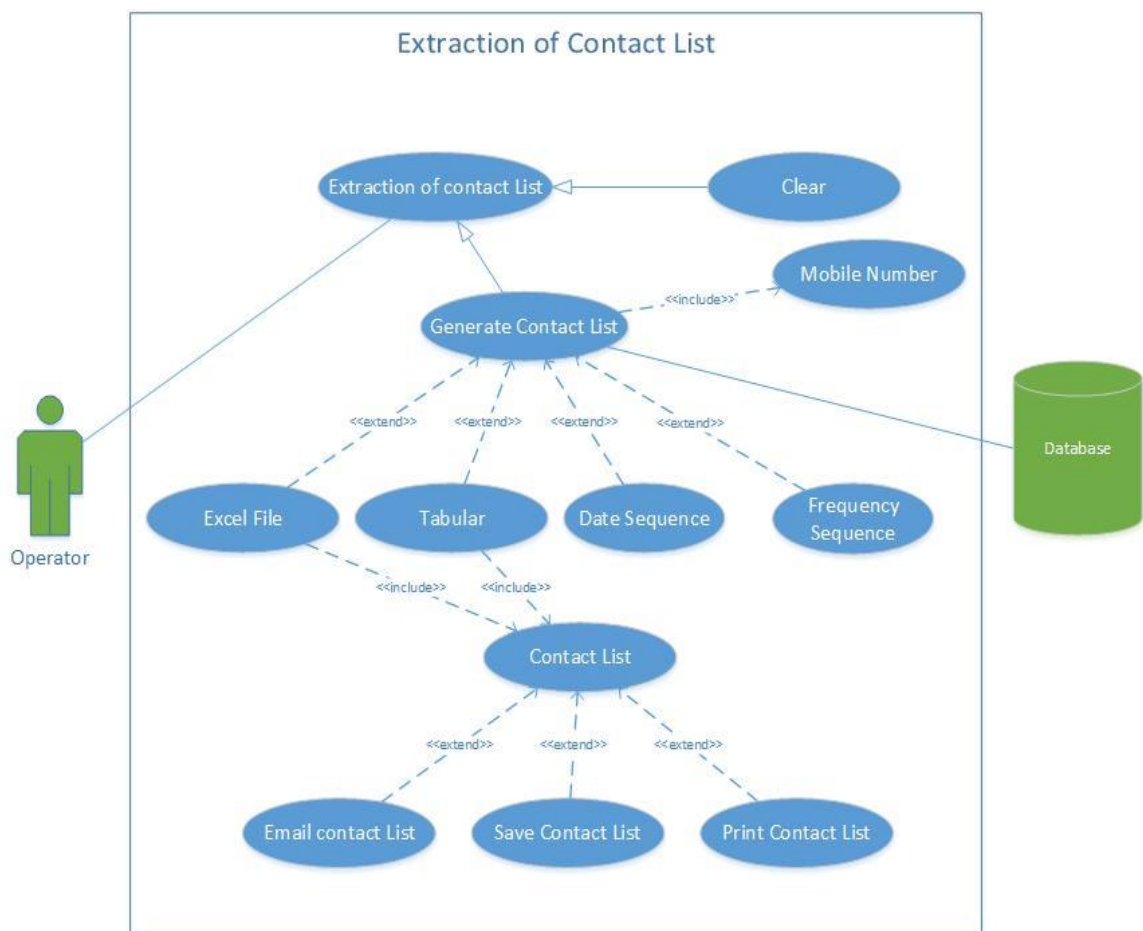


Figure 4-9 : Extract Contacts Use Case

4.6.3.1 Use Case Description

This use case describes how the operator would be able to retrieve the contact list using the extraction of contact list functionality.

4.6.3.2 Actors

- a. Operator.
- b. Database.

4.6.3.3 Pre-Conditions

- a. User must have the software installed on his PC in the telecommunication organization.
- b. User must have internet connectivity available.
- c. User must be an authenticated user.
- d. The mobile number must be present in the database.

4.6.3.4 Basic Flow of Events

- a. Operator starts the Contact List Retrieval application.
- b. Operator logs in.
- c. Operator opens Extraction of contact list.
- d. Operator chooses the domain and enters the mobile number of the customer whose contact list is to be retrieved.
- e. Operator then selects the sequence in which he wants the output to be either Date or Frequency.

- f. Operator chooses the Tabular or File format of the output.
- g. Operator then clicks on Generate.
- h. If tabular is clicked a table pops up showing the contact list and options to print, email and save.
- i. Operator clicks on print the contact list is printed.
- j. Operator clicks on save the contact list is saved in excel file format.
- k. Operator clicks on email the email functionality opens up
- l. Operator attaches the file and clicks on send email.
- m. The email is send to the customer.
- n. If File format is chosen and generate is clicked.
- o. The contact list will be saved in excel file.
- p. Operator clicks on clear all the fields become empty again.
- q. Operator clicks on back and goes to the main menu.

4.6.3.5 Alternate Flow

No connectivity: In case of no connectivity, system informs the user to check internet connections and send mail again.

Number does not exist in database: In case of the mobile number not being present in the database the contact list will not be extracted.

4.6.3.6 Post-Conditions

a. Successful Completion

Operator is able to extract the contact list.

b. Unsuccessful Completion

Operator is not able to perform the required functions.

4.6.4 Search Use Case

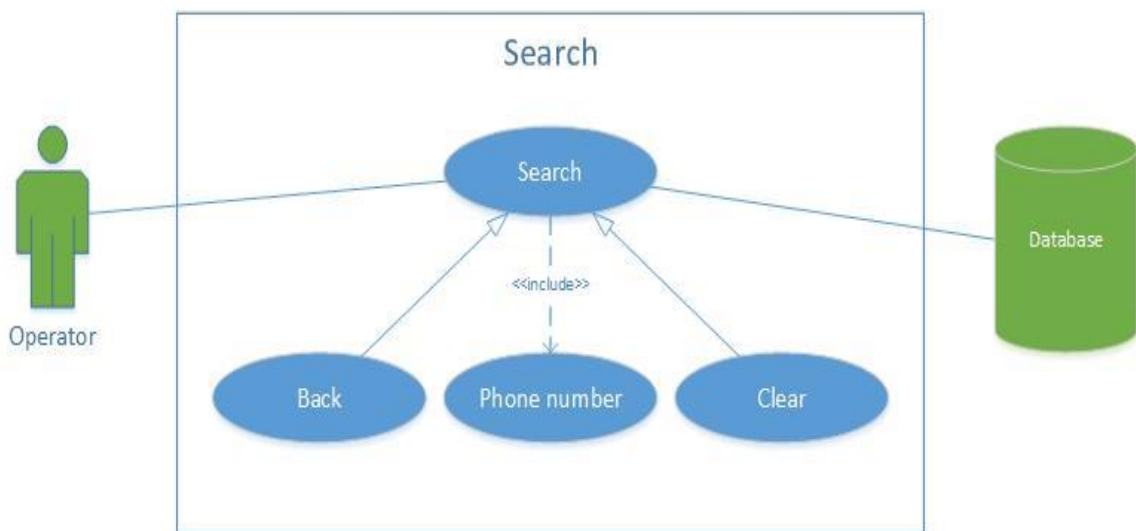


Figure 4-10 : Search Use case

4.6.5 Use Case Description

This use case describes how the Operator searches the database to check if the mobile number given by the customer exists in the database or not.

4.6.5.1 Actors

- a. Operator.
- b. Database.

4.6.5.2 Pre-Conditions

- a. User must have application installed on his PC in the telecommunication company.

- b. User must be logged in.

4.6.5.3 Basic Flow of Events

- a. Operator starts the Contact List Retrieval application.
- b. Operator logs in.
- c. Operator opens Extraction of contact list.
- d. Operator chooses the domain and enters the mobile number of the customer whose number is to be searched.
- e. Operator clicks on search.
- f. If mobile number exists “Found is displayed”.
- g. If mobile number doesn’t exist “Cannot find mobile number in the database is displayed”.
- h. Operator clicks on clear the fields become empty.
- i. Operator clicks on back and goes to the main menu.

4.6.5.4 Alternate Flow

None

Post-Conditions

a. Successful Completion

Operator is able to check if the mobile number exists or not.

b. Unsuccessful Completion

Operator is not able to check whether the mobile number exists or not.

4.6.6 Email Use Case

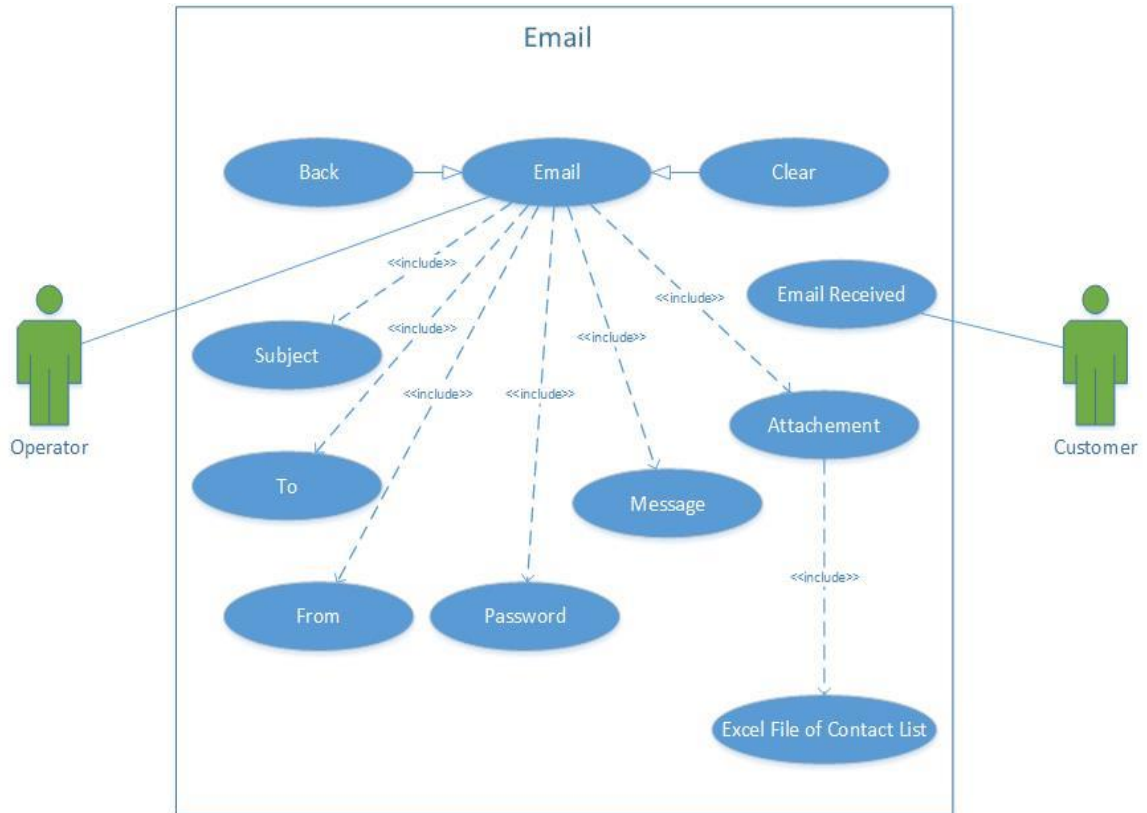


Figure 4-11: Email Use Case

4.6.6.1 Use Case Description

This use case describes how the Operator sends and email to the customer.

4.6.6.2 Actors

- a. Operator.
- b. Customer.

4.6.6.3 Pre-Conditions

- a. User must have the software installed on his PC in the telecommunication organization.
- b. User must have internet connectivity available.
- c. User must be an authenticated user.

4.6.6.4 Basic Flow of Events

- a. Operator starts the Contact List Retrieval application.
- b. Operator logs in.
- c. Operator opens extraction of contact list.
- d. Operator chooses the tabular form of output and clicks generate.
- e. Operator clicks on email.
- f. Operator enters his email address of either Hotmail or Gmail.
- g. Operator fills all the fields i.e. To (customers email address) , password, Subject , Message.
- h. Operator attaches the excel file of the customer.
- i. Operator clicks on send mail,
- j. Mail is sent to the customer.
- k. The customer receives the mail with his contact list.

4.6.6.5 Alternate Flow

Email id/Password invalid: The system prompts the user to enter a valid email id or password.

No connectivity: In case of no connectivity, system informs the user to check internet connections and send mail again.

4.6.6.6 Post-Conditions

a. Successful Completion

Operator is able to send the mail which is received by the customer.

b. Unsuccessful Completion

Email not sent.

4.7 CDR Class Diagram

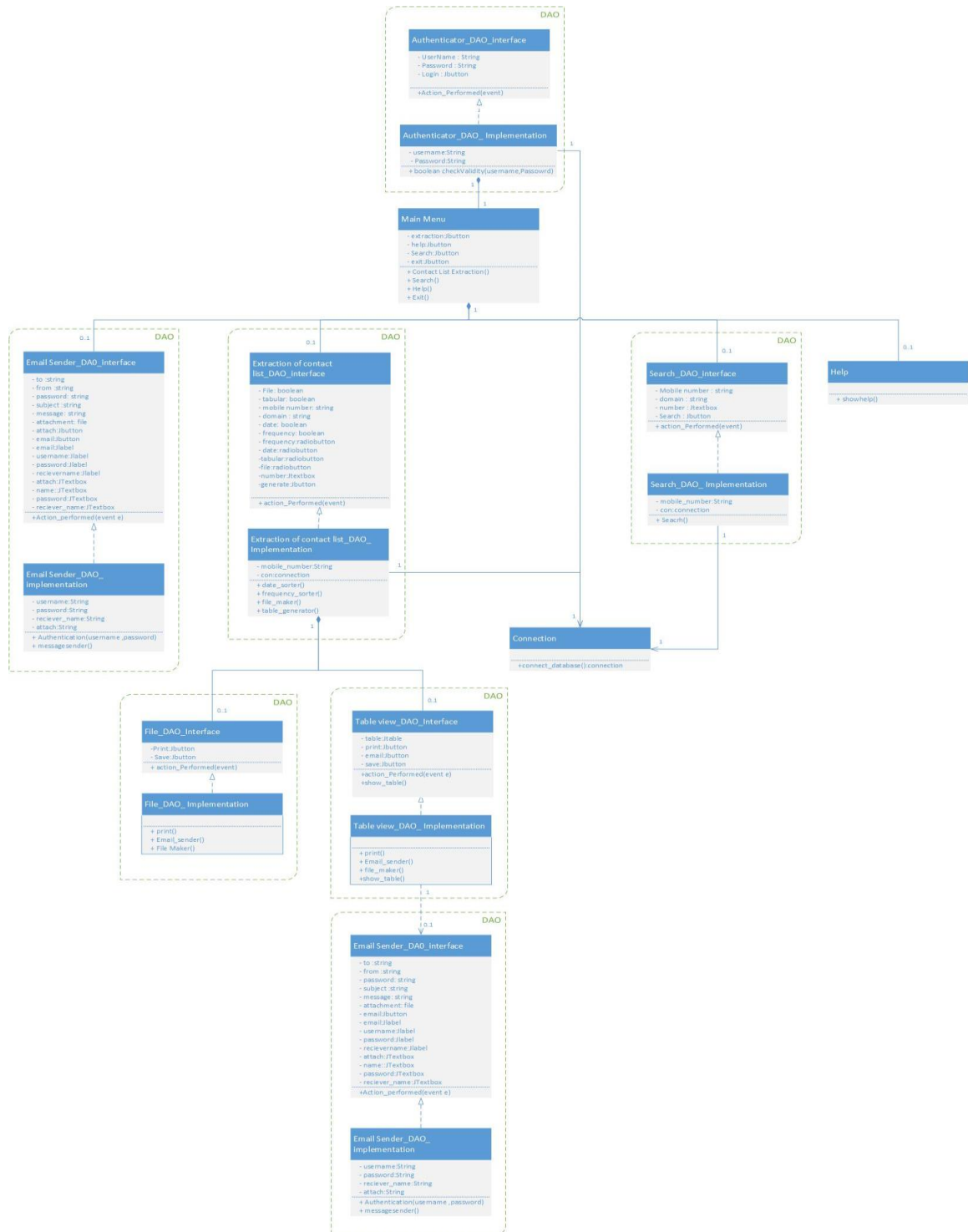


Figure 4-12 : CDR Class Diagram

Classes' Detail

4.7.1 Main Menu Class

4.7.1.1 Description

The main class of the system interface, which will handle all the communication and utilize the functionality of its sub-systems.

4.7.1.2 Variables

May include variables during development.

4.7.1.3 Functions

Contact List extraction (): To extract the contact list and perform further functionality on it.

Search (): To search the mobile number in database and display its availability.

Help (): For viewing the help provided for the software.

Exit (): To turn the software off.

4.7.2 Authenticator_DAO_interface Class

4.7.2.1 Description

The class deals with the interface for the authenticator it requires the authenticator_DAO_implementation class.

4.7.2.2 Variables

Username: Username of the Contact list retrieval software's operator/administrator.

Password: Password provided by the operator/administrator.

4.7.3 Authenticator_DAO_implementation Class

4.7.3.1 Description

This class is required to perform the functionality of authentication of the user through the database for that purpose a resource manager class is being used which acts as a helper for this class.

Uses the connection class to first make a valid connection with the correct database.

4.7.3.2 Functions

Check Availability (): Checks if the user name and password entered is correct and valid if yes then allows the user to proceed further on.

4.7.4 Connection Class

4.7.4.1 Description

Resource manager class of our system as its purpose is to create database connection and return it back. It ensures the valid connection with the database. It Acts as a helper class for all the DAO implementation classes in the system.

4.7.5 Extract of contact List_Dao_interface Class

4.7.5.1 Description

The core class of the contact list extraction software that retrieves the contact list from the database.

4.7.5.2 Variables

File: Identifies if the output needs to be an excel file.

Tabular: Identifies if the output needs to be in tabular form.

Mobile Number: The number whose contact list needs to be retrieved.

Domain: Specifies the domain of the number whose contact list needs to be retrieved.

Date: Identifies if the sequence in which the output is displayed is according to date i.e. most recent call first.

Frequency: Identifies if the sequence in which the output is to be displayed is in accordance with the frequency of calls made i.e most calls made to a number displayed at the top.

4.7.6 Extraction of contact list_DAO_implementation Class

4.7.6.1 Description

Main class for extraction and handling the contact list requires the helper class connection for connecting with the database. This class performs the functionality of retrieving the contact list given a mobile number and perform further functionality on it.

4.7.6.2 Functions

Date_sorter(): Given that the date variable is true this function sorts the contact list according to date i.e. most recent call first.

Frequency_sorter(): Given that the frequency variable is true this function sorts the contact list according to frequency i.e most dialed numbers first.

File_maker (): Given that the file variable is true this function saves the contact list in an excel file format with the Names and mobile numbers.

Table_generator(): Given that the tabular variable is true this function displays the contact list in a tabular form with the names and mobile numbers.

4.7.7 Search_DAO_Interface Class

4.7.7.1 Description

An important class to make sure that a particular number whose contact list is to be extracted is present in the database or not.

4.7.7.2 Variables

Mobile Number: The number whose contact list needs to be retrieved.

Domain: Specifies the domain of the number whose contact list needs to be retrieved.

4.7.8 Search_DAO_Implementation Class

4.7.8.1 Description

A class that checks whether a mobile number is present in the database or not requires a helper class connector to establish a connection with the database.

4.7.8.2 Functions

Search() : Uses the mobile number given and searches in the database to check if it is a valid number or not i.e If the number exists in the database or not.

4.7.9 Help Class

4.7.9.1 Description

A class that provides information regarding how to use the software

4.7.9.2 Functions

Show help (): Shows the document containing detailed information regarding how to use the software.

4.7.10 Table View_DAO_interface Class

4.7.10.1 Description

Provides an interface regarding how the tabular display is shown

4.7.11 Table View_DAO_imlementation Class

4.7.11.1 Description

A class that provides the functionality of displaying the tabular results along with other functionality requires the helper class connection for establishing a valid connection with the database.

4.7.11.2 Functions

Print() : Prints the contact list

Save(): Save the contact list in excel file format.

Email() : Opens the email functionality allowing the operator to mail the contact list to the customer.

4.7.12 Email Sender_DAO_interface Class

4.7.12.1 Description

This class provides an interface for the Email sender.

4.7.12.2 Variables

To: provides the email address of the customer.

From: provides the email address of the operator.

Password: provides the password of the operator.

Subject: provides the subject of the email.

Message: provides the message.

Attachment: The contact list in excel file attached.

4.7.13 Email Sender_DAO_implementation Class

4.7.13.1 Description

The class provides the functionality of sending an email to the customer.

4.7.13.2 Functions

Authentication (): Checks if the email address and password are valid.

Message Sender (): Sends the mail to the customer.

4.8 Generator Class Diagram

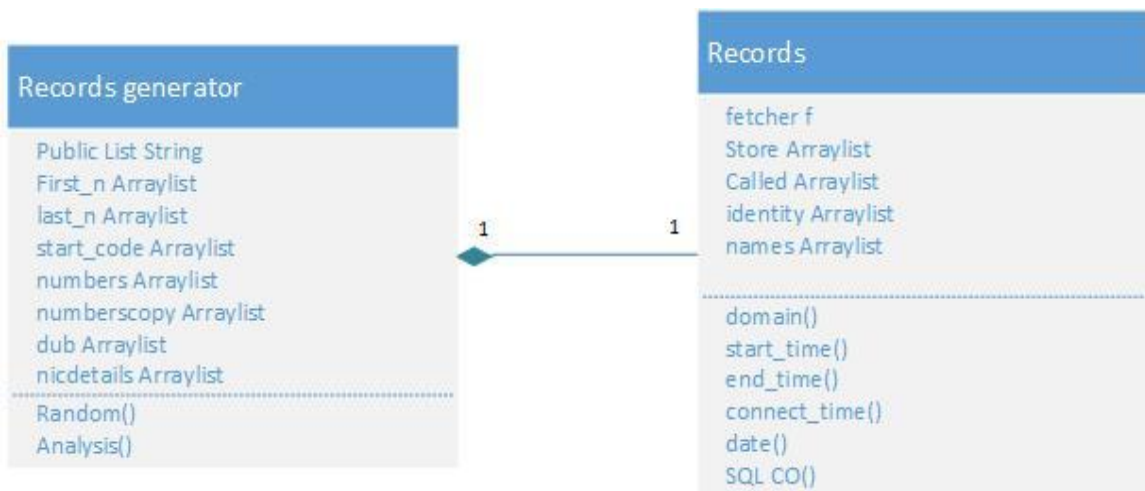


Figure 4-13: Generator Class Diagram

4.8.1 Records Generator Class

4.8.1.1 Description

This class reads data from file and insert personal information into the database.

4.8.1.2 Variables

List: List comprises of first,n, last_n , startcode, numbers ,nicdetails

First_n: contains the first part of the names read from the text file

Last_n: it contains the second part of all the names read from the text files containing name

Start_code: contains the possible domain codes for the mobile numbers read from the text file.

Numbers: contains the phone numbers generated through code and the domain codes received through text file

nicdetails: contains all the NIC numbers generated from the code.

4.8.1.3 Functions

Random (): This function is used to generate NIC numbers and phone numbers randomly.

It makes a connection to the database to create personal details table containing the names and the generated NIC and the phone numbers table containing the randomly generated NIC and phone numbers.

Analysis (): This function reads the file containing names and splits them to make of first and last names. This function also reads another file containing domain codes and adds them to another array list.

4.8.2 Records Class

4.8.2.1 Description

This class fetches the phone numbers generated in final project class using the **FETCH PATTERN** and then places it within the cdr table along with the other required fields like start_time, end_time,date,domain etc.

4.8.2.2 Variables

Store: contains the phone numbers generated in FINAL PROJECT class.

4.8.2.3 Functions

Domain: This function receives the domain code as a parameter and returns the respective telecom company name to which this code belongs to e.g. on receiving 0321 it will return WARID.

start_time: This function on its call generates the random time in the "hour:min:second" format which is the start time of the call which is filled into the cdr table.

end_time: This function returns the end time for a call in the hour:min:sec format and makes sure that it is after the start time of the call.

connect_time: This function returns the time in hour:min:sec format by adding the time required for connecting the call to the start time of that call.

Date: This function returns the date in months/days/year format which is the date a call was made.

SQL_con: This function gets the phone numbers through fetcher class and use them as phone numbers of calling person and to whom call was made by making sure they are not the same. This function also calls other functions of this class to get other required fields' values of cdr like start time,end time etc. This way it generates the records to be placed within the cdr table.

4.9 Dynamic View

4.9.1 Activity Diagrams

4.9.1.1 Main Page

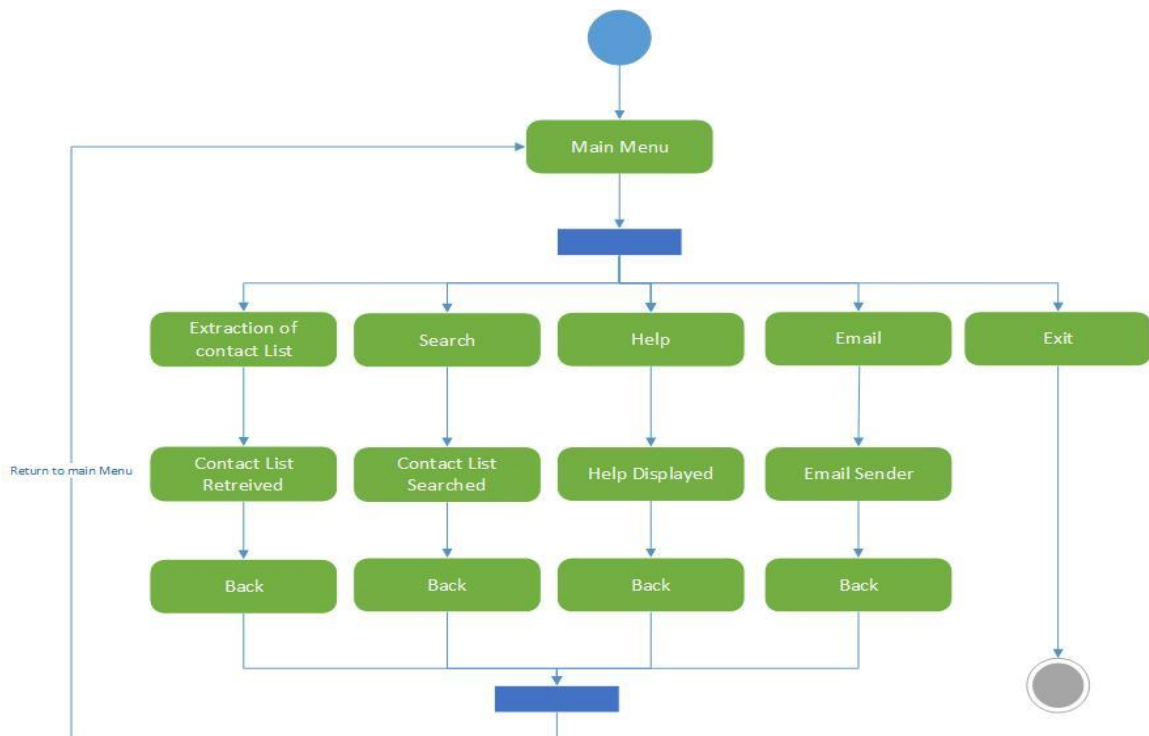


Figure 4-14: Main Page Activity Diagram

4.9.1.2 Extract Contacts

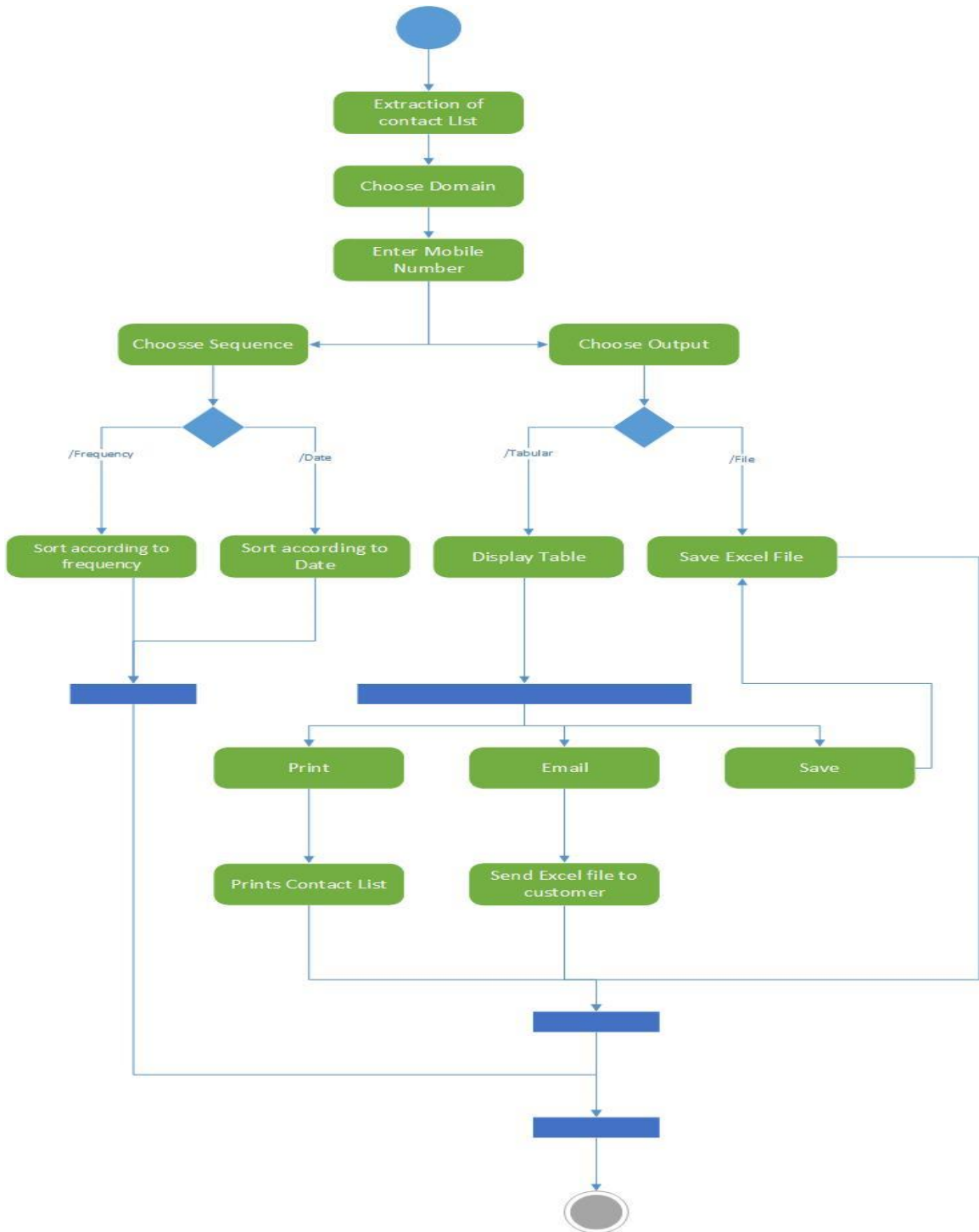


Figure 4-15: Extract Contacts Activity Diagram

4.9.1.3 Search

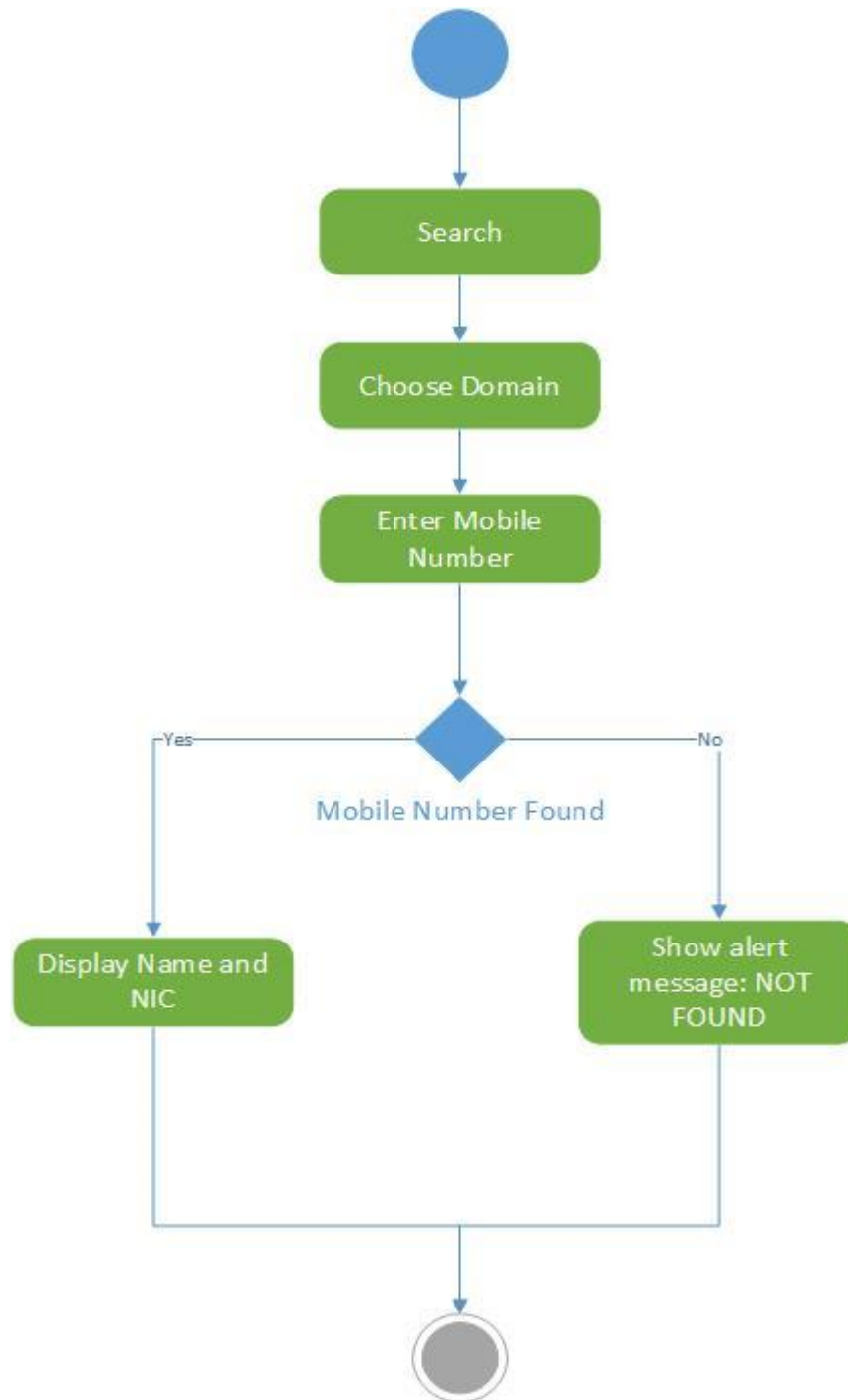


Figure 4-16: Search Activity Diagram

4.9.1.4 Email

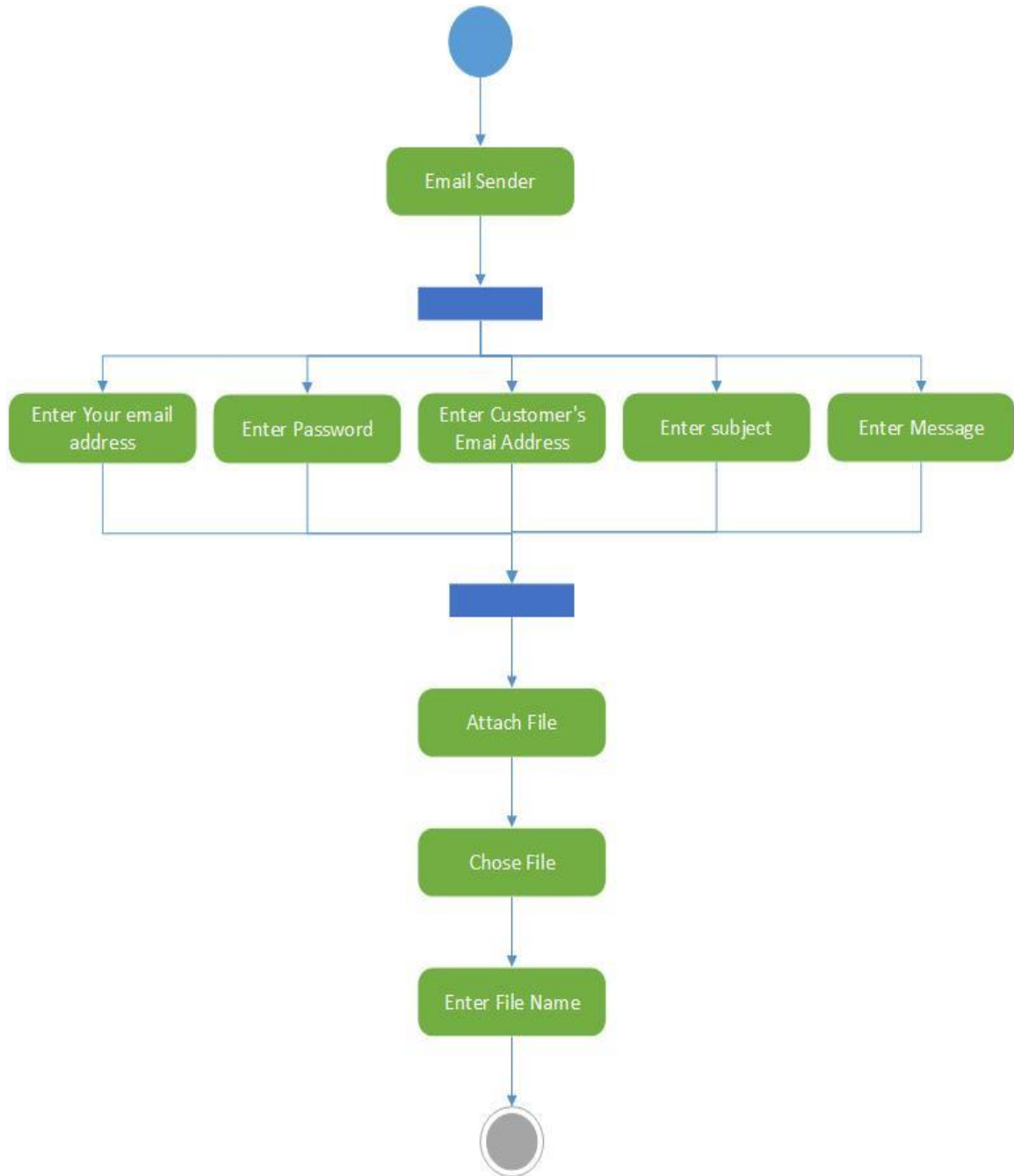


Figure 4-17: Email Activity Diagram

4.9.1.5 Help



Figure 4-18: Help Activity Diagram

4.9.2 Sequence Diagrams

4.9.2.1 Authenticator Sequence

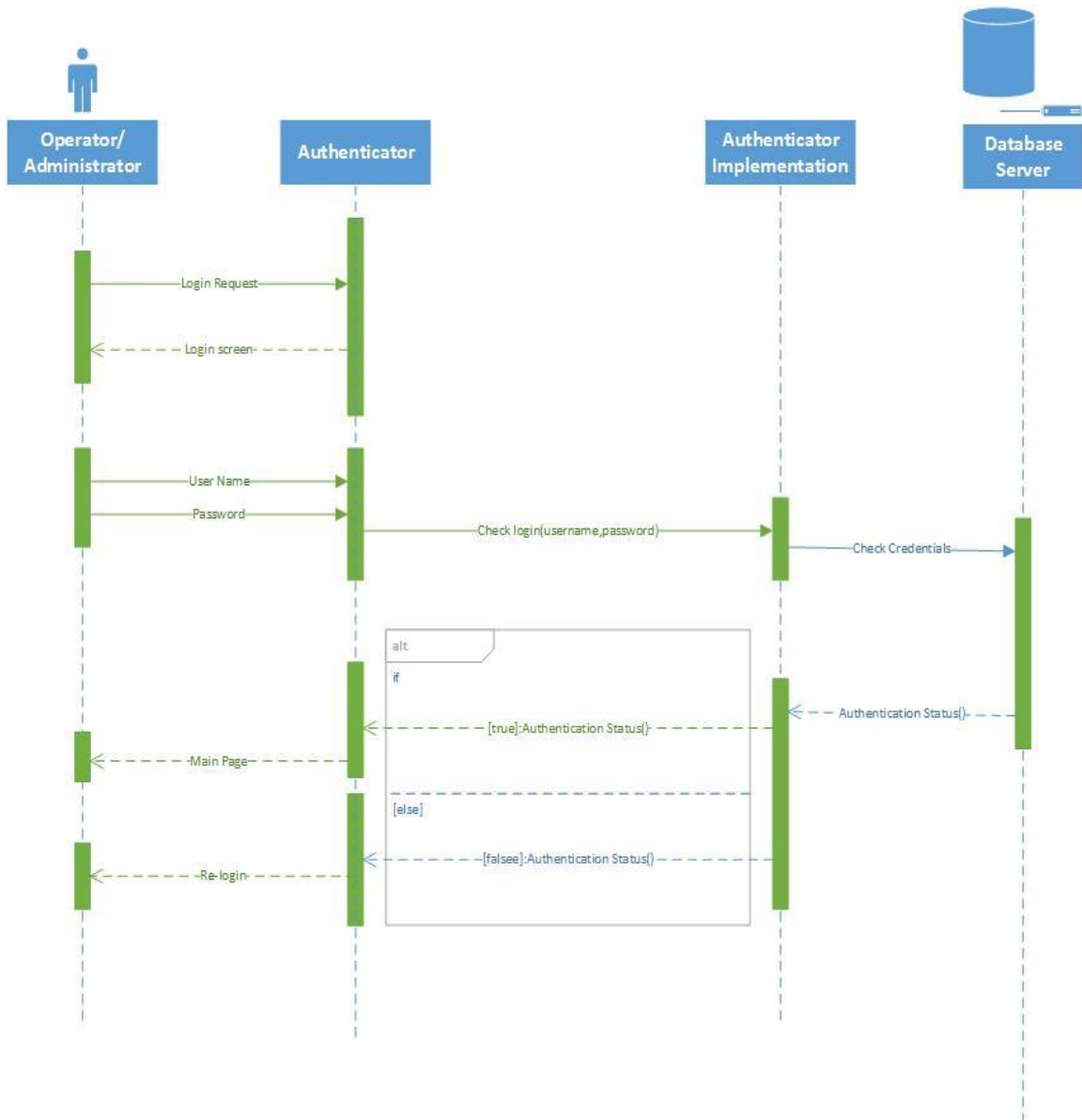


Figure 4-19: Authenticator Sequence Diagram

4.9.2.2 Main Page Sequence

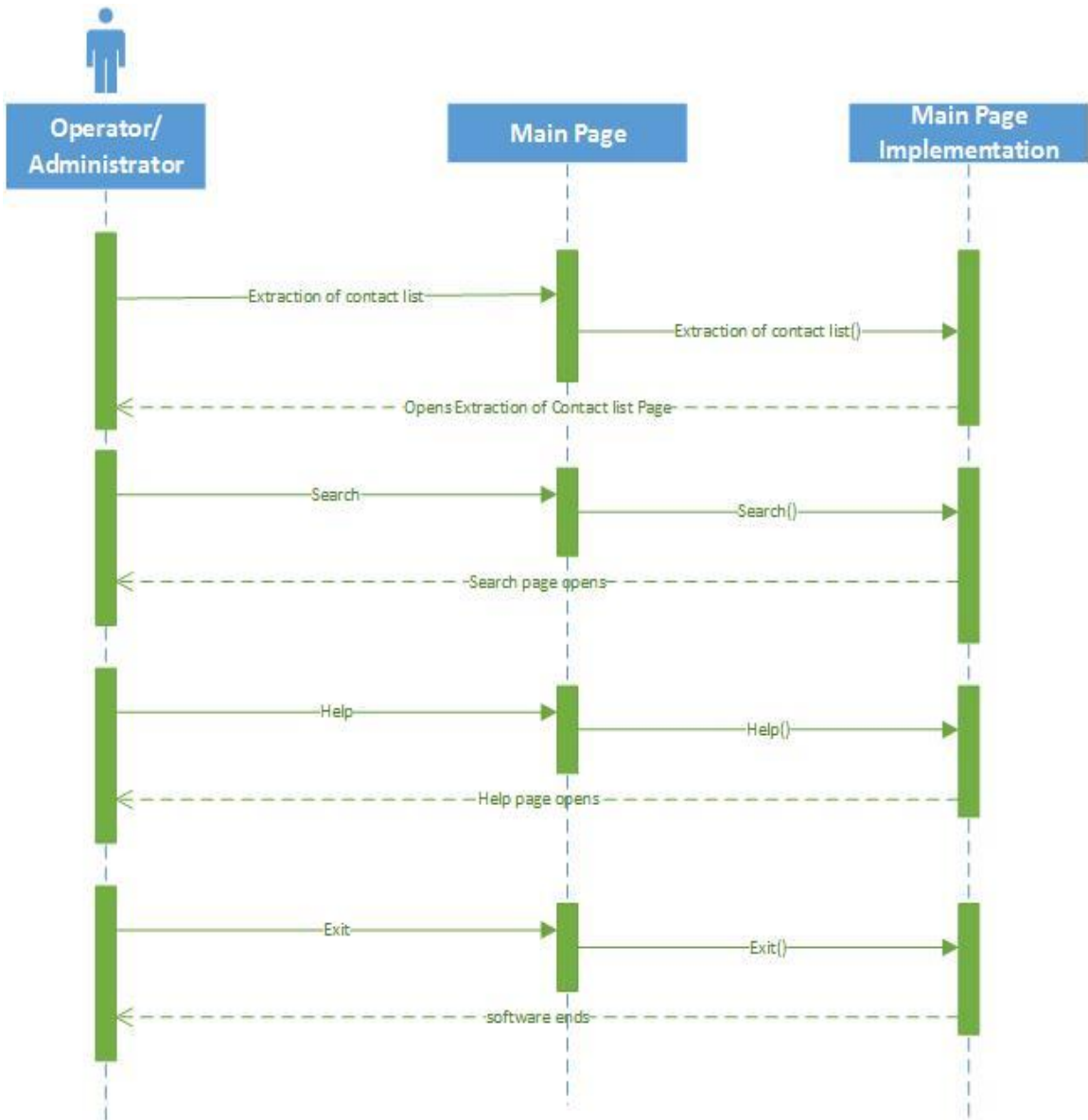


Figure 4-20: Main Page Sequence Diagram

4.9.2.3 Extract Contacts Sequence Diagram

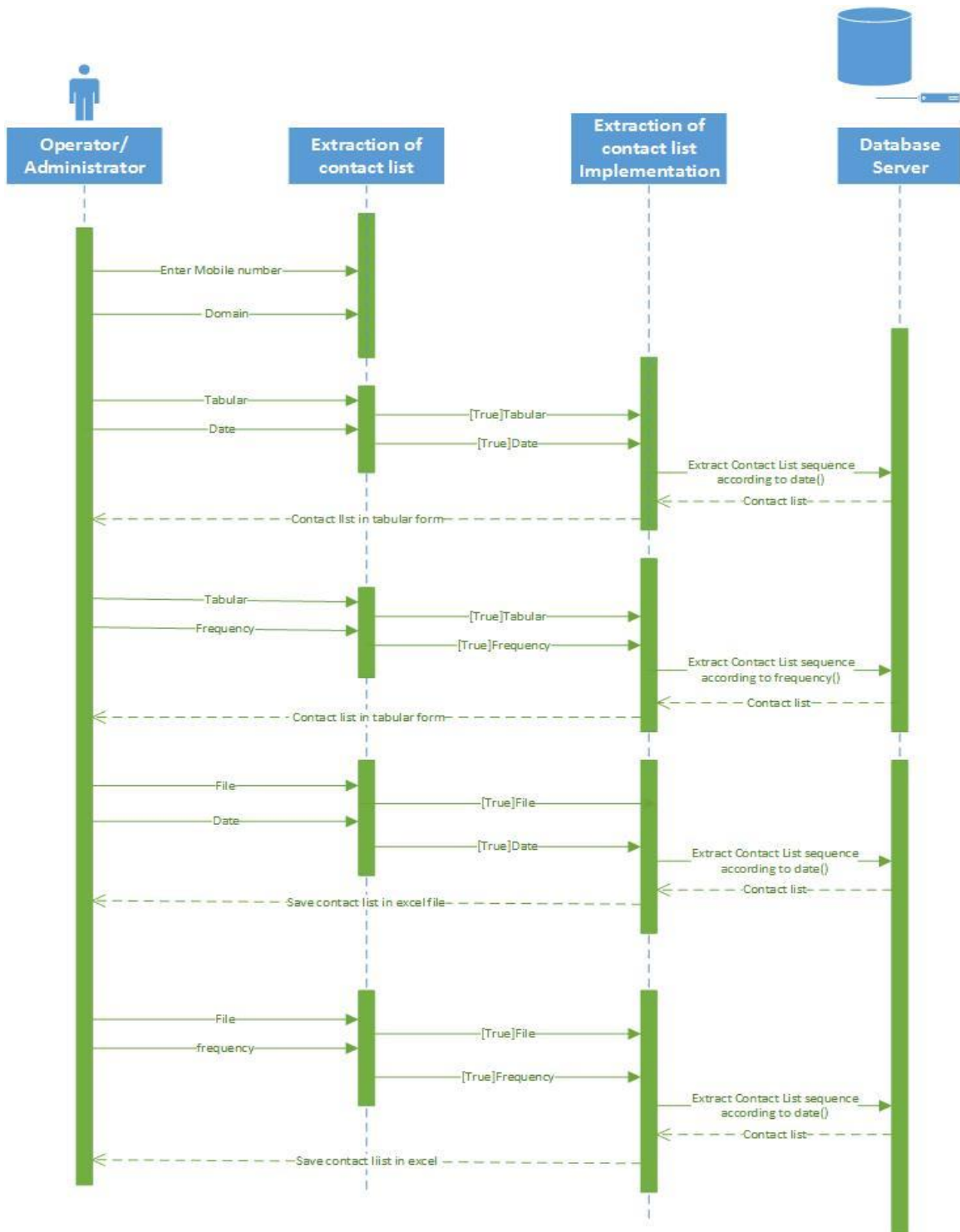


Figure 4-21: Extract Contacts Sequence Diagram

4.9.2.4 Search Sequence

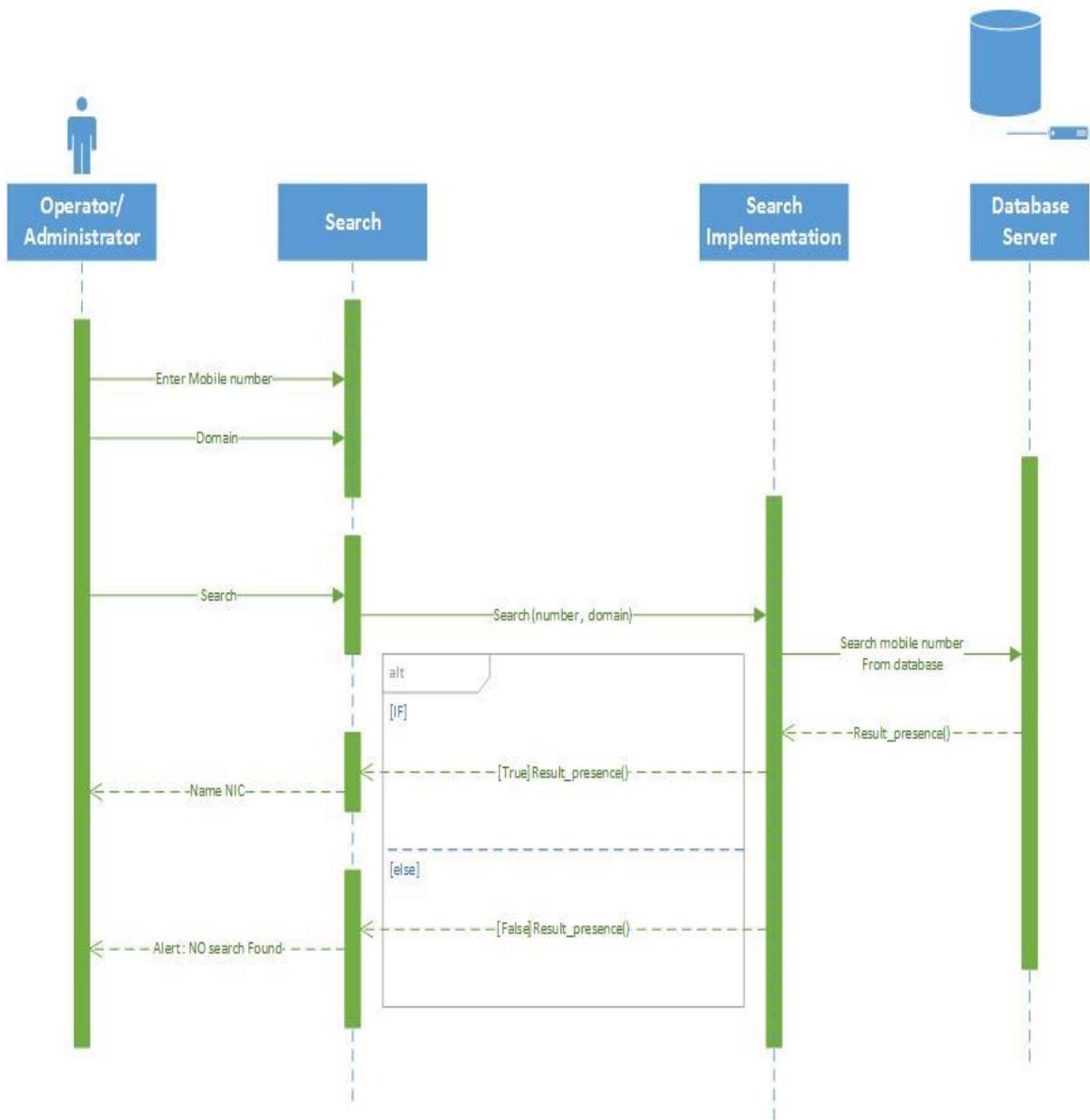


Figure 4-22: Search Sequence Diagram

4.9.2.5 Email Sequence

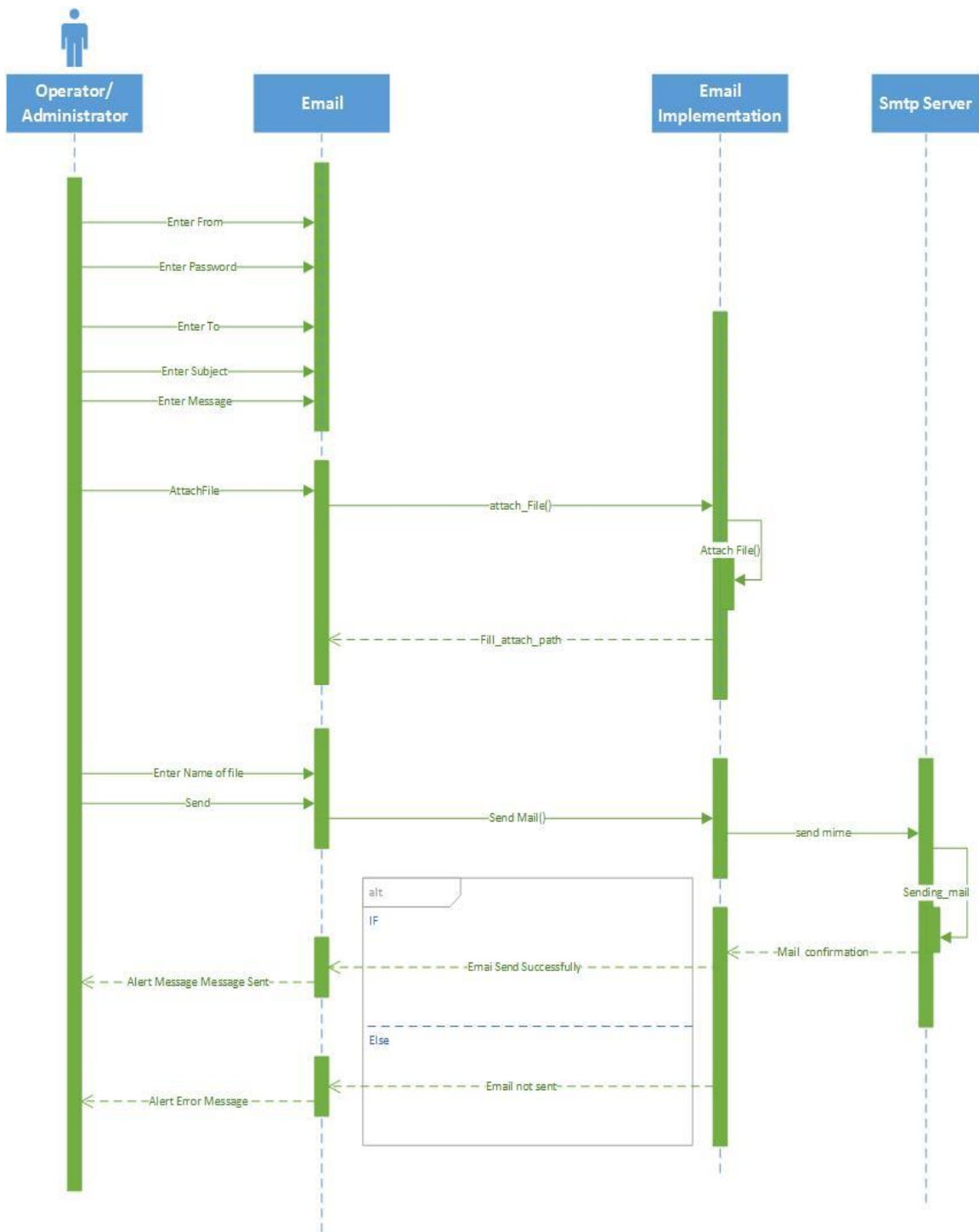


Figure 4-23: Email Sequence Diagram

4.9.2.6 Help Sequence

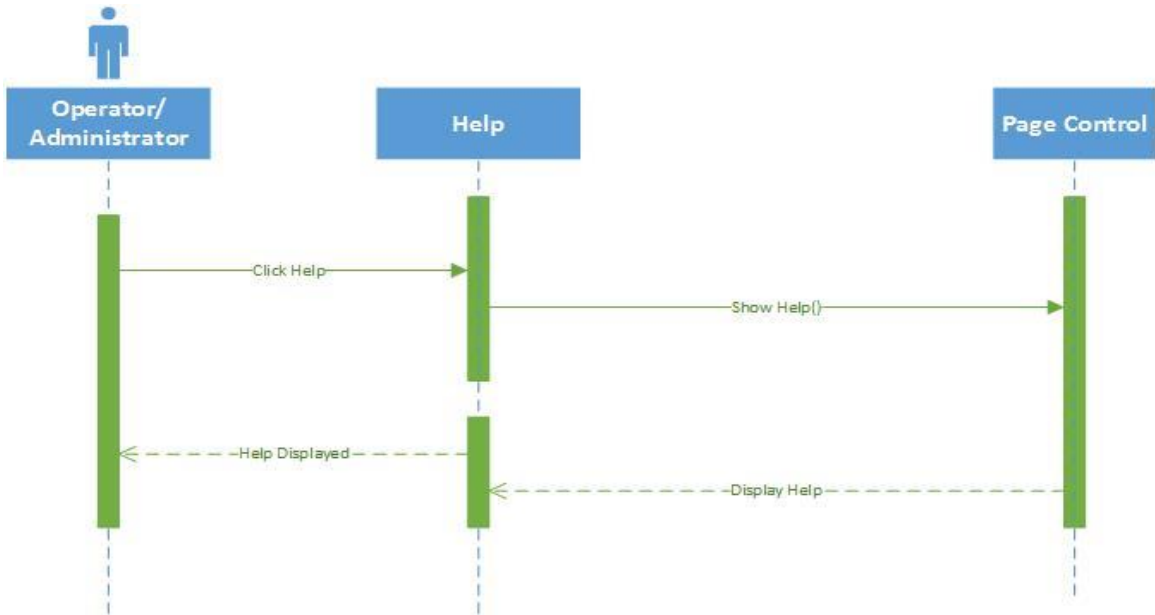


Figure 4-24: Help Sequence Diagram

4.9.3 Data Flow Diagram

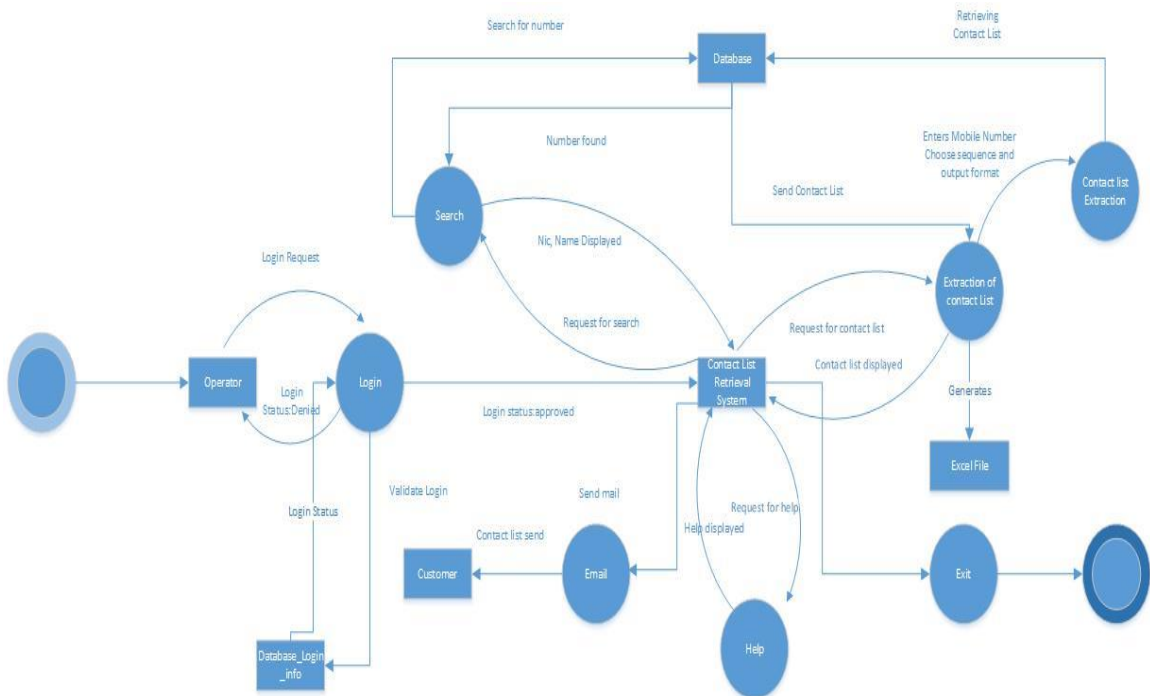
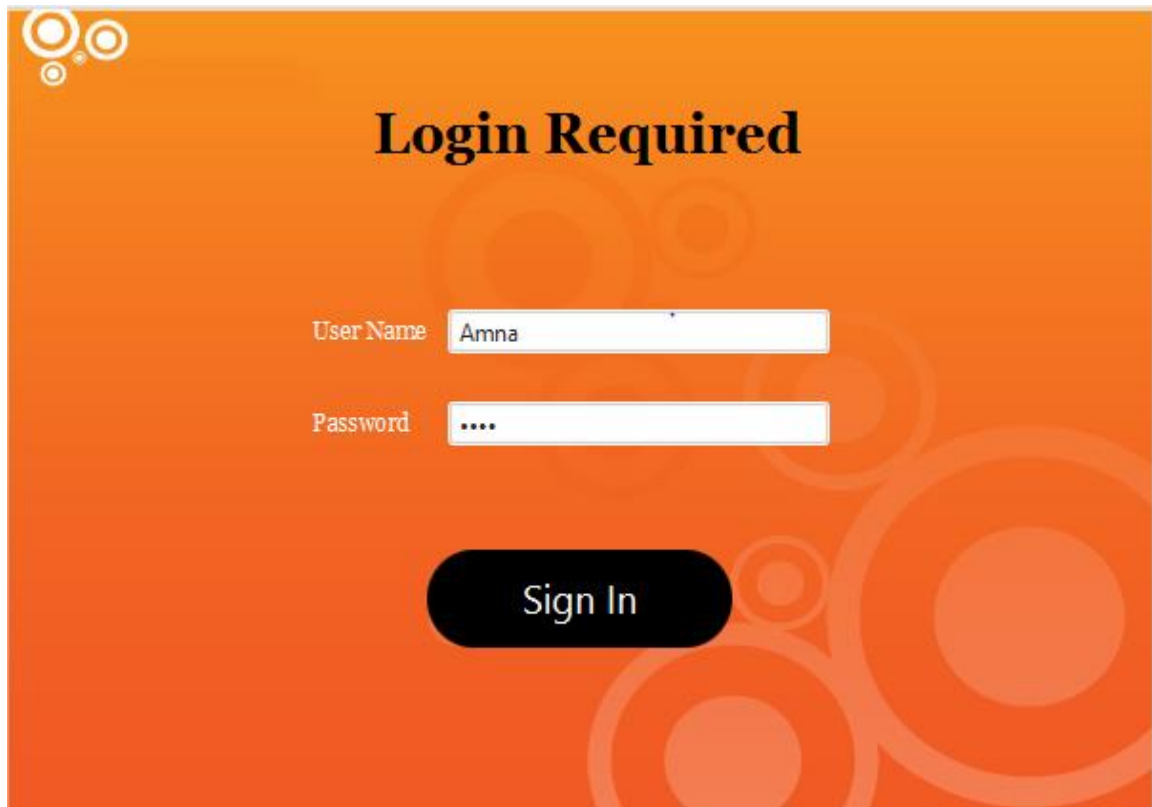


Figure 4-25: Data Flow Diagram

4.9.4 Human Interface Design

4.9.4.1 Login Page



The image shows a login page with an orange background. In the top left corner, there is a logo consisting of three overlapping circles. The main heading is "Login Required" in a large, bold, black serif font. Below the heading, there are two input fields: "User Name" with the text "Amna" and "Password" with four dots. A black rounded rectangular button with the text "Sign In" in white is positioned below the password field.

Figure 4-26: Login Page

Description: This is the first page displayed to the user i.e. Operator/Administrator.

When the user enters their user name and password if validated they proceed to the main page.

4.9.4.2 Main Page

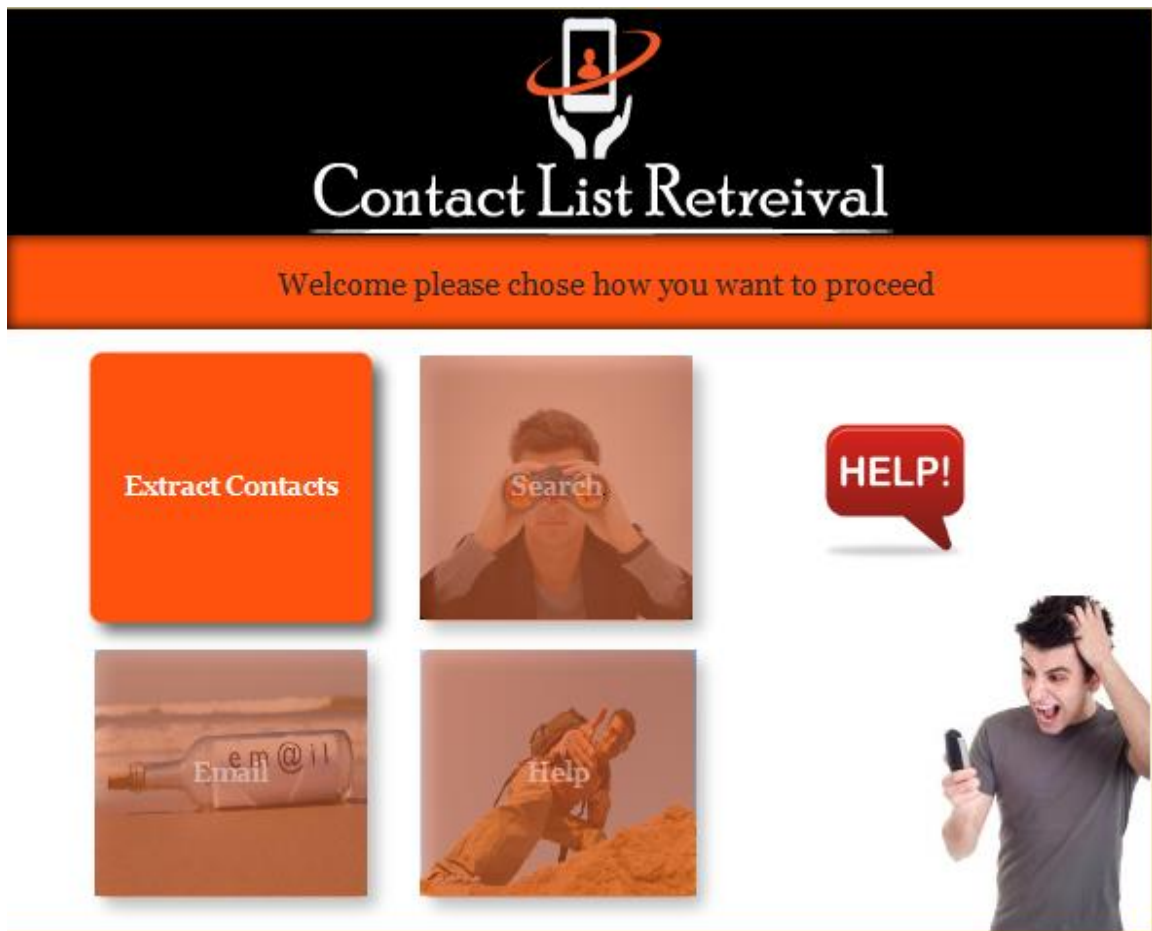


Figure 4-27: Main Page

Description: If logged in the user can now access the main features of the software which are included in the main page these include Extraction of contact list, Search, Email, Help, Exit. Each feature proceeds further on the Exit button closes the software.

4.9.4.3 Extract Contacts Page



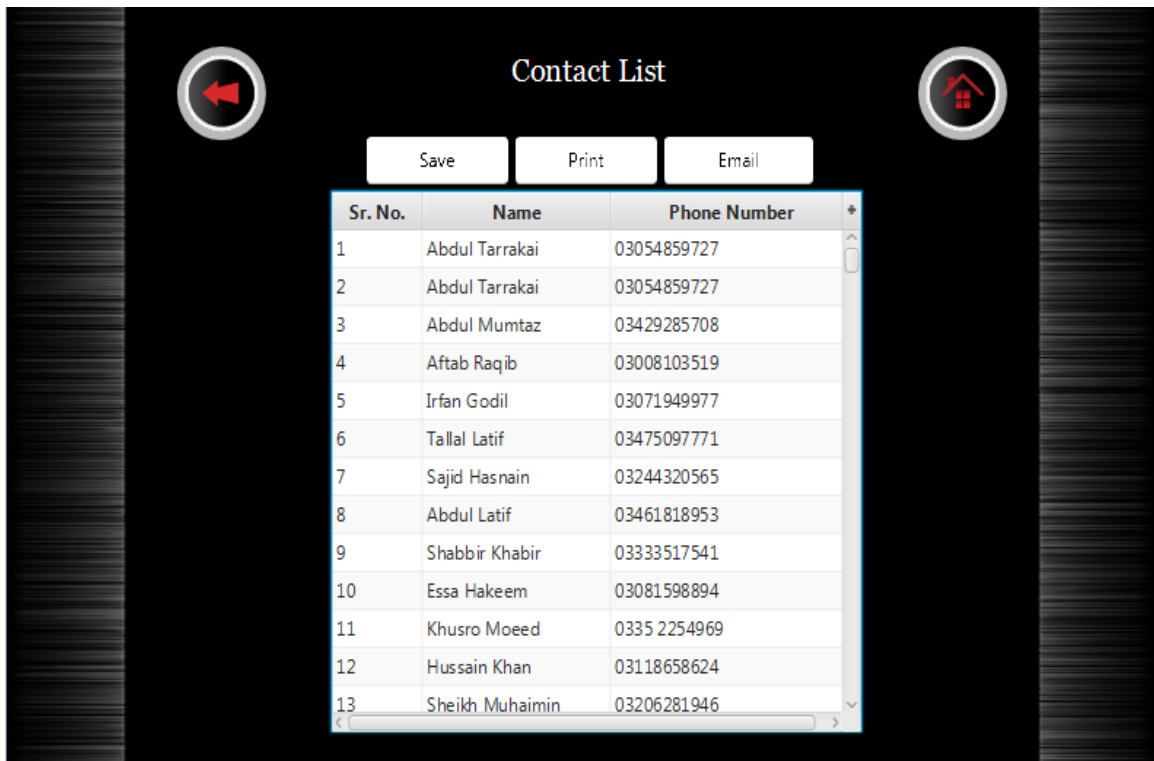
The image shows a web form titled "Extract Contact List" on a dark background with red and black decorative spheres. The form includes a "Domain" dropdown menu, a text input field containing "5650731", a "Sequence" section with radio buttons for "Frequency" and "Date", an "Output Format" section with radio buttons for "Tabular" and "File", and a prominent red "Generate" button at the bottom center.

Figure 4-28: Extract Contacts

Description: If the user chooses extraction of contact list from the main page this page is displayed. The user chooses the domain and enters the mobile number. He then has the choice of the sequence in which he wants the output. He can choose either Date/Frequency. He can then choose the form of output he wants in either tabular/File. The user can then click the Generate button to get the result. The user can use the clear

button to clear all the fields. The user can use the back button to go back to the main menu.

4.9.4.4 Tabular Output



Sr. No.	Name	Phone Number
1	Abdul Tarrakai	03054859727
2	Abdul Tarrakai	03054859727
3	Abdul Mumtaz	03429285708
4	Aftab Raqib	03008103519
5	Irfan Godil	03071949977
6	Tallal Latif	03475097771
7	Sajid Hasnain	03244320565
8	Abdul Latif	03461818953
9	Shabbir Khabir	03333517541
10	Essa Hakeem	03081598894
11	Khusro Moeed	0335 2254969
12	Hussain Khan	03118658624
13	Sheikh Muhaimin	03206281946

Figure 4-29: Tabular Output

Description: The user can view the output and also has the choice to email the output to the customer and generate an excel file.

4.9.4.5 File Output



Figure 4-30: File Page

Description: When the user chooses the file output an excel file is created and the file module is displayed that allows the user to mail or print the file.

4.9.4.6 Email Sender

Email Sender

Your Id: amna.cdr@gmail.com

Password:

Customer id: amna.feroz90@gmail.com

Subject: Contact List

Message: Dear Customer, Thankyou for using Contact List Extraction Your contact list is attached

Attach file: Give File Name: C://Data/03215650731.txt

Send Mail

Figure 4-31: Email Sender

Description: The user enters the fields i.e To, From, Password, Subject, Message, Attach_name and then he can attach the file to be send to the customer including the contact list.

4.9.4.7 Search

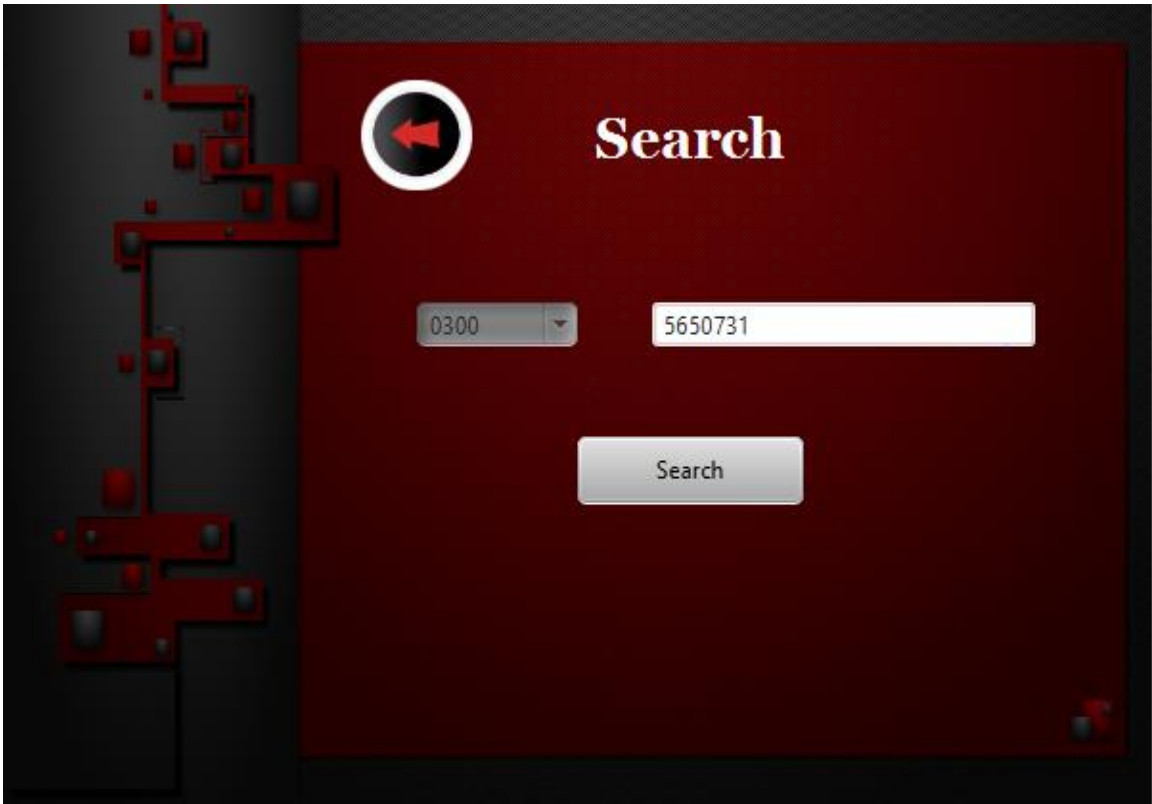


Figure 4-32: Search Page (a)

Description: The user chooses the domain and enters the mobile number. When the user clicks search if the number is present the NIC and name will be displayed as shown in figure 6.7 .If the number doesn't exist in the database an alert message would be displayed showing that the number does not exist. He can use the clear button to clear all the fields. And the back button takes the user back to the main page.

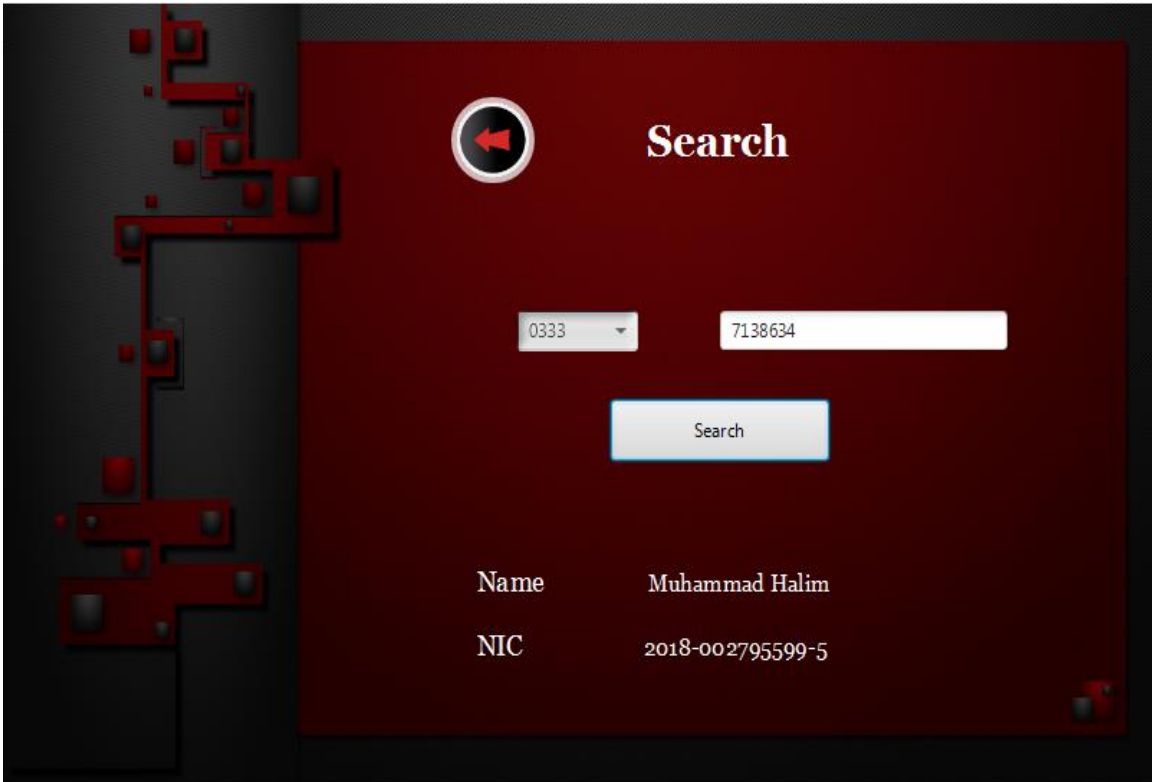


Figure 4-33: Search Page (b)

4.9.4.8 Print

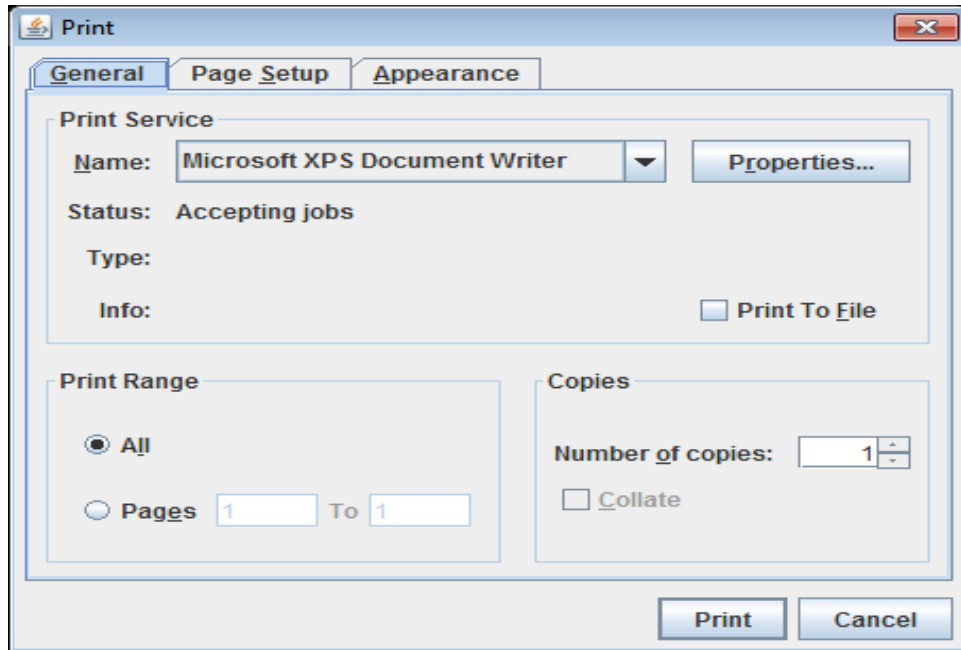


Figure 4-34: Print Page

Description: When the user clicks print the print module is displayed. User can select the printer and print the contact list.

4.9.4.9 Help

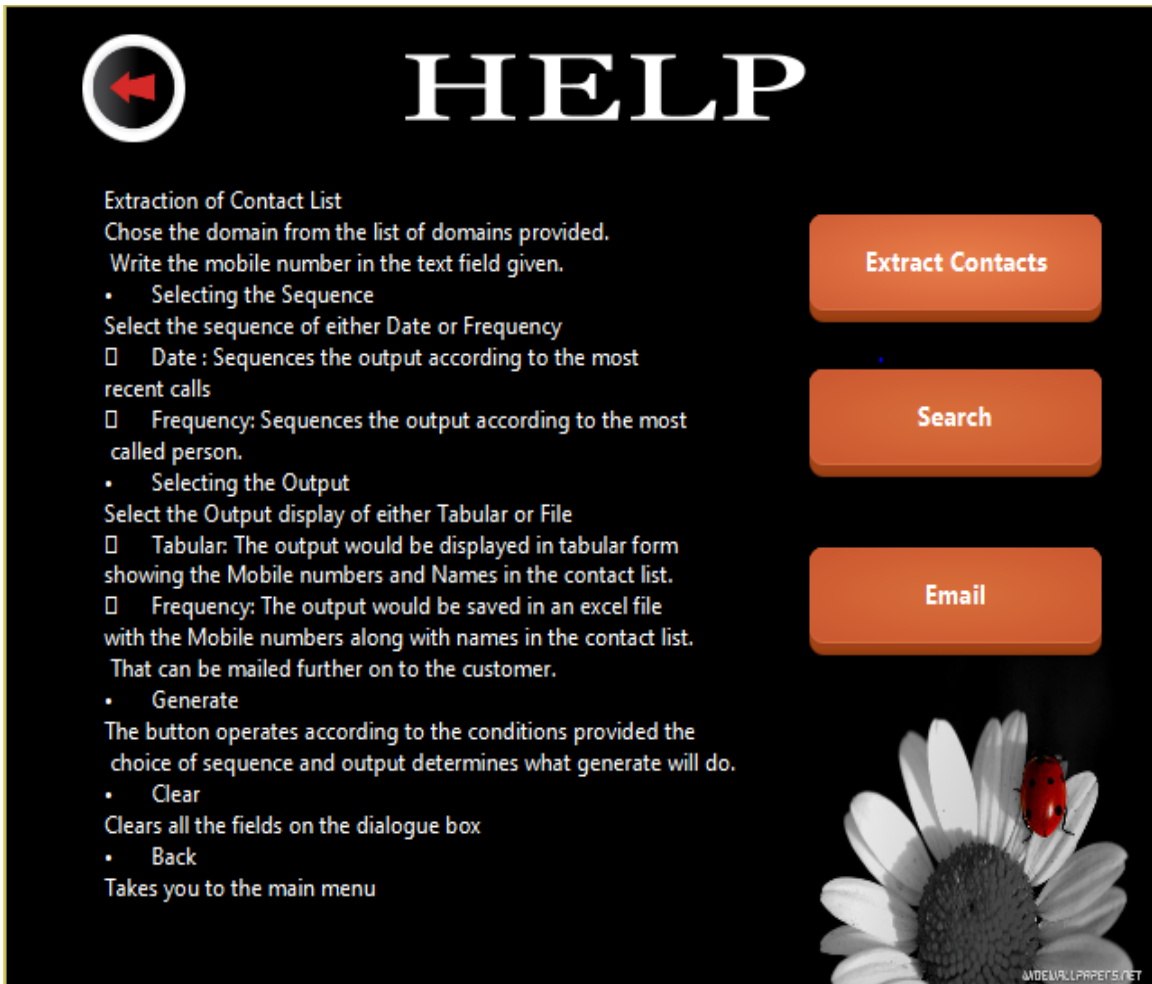


Figure 4-35 : Help

Description: When the user clicks help the document containing guidelines for user is displayed.

User can choose the help required for each feature. The back button takes the user back to the main page.

5. System Implementation

5.1 Implementation introduction

The previous chapter discussed the design; inner details of the class attributes and methods, Use case diagram, ER diagram etc. Based on that design, this chapter will concentrate on the implementation details of the system.

5.2 Tools and Technologies

1. Netbeans IDE 8.0
2. SQL Server 2008 R2
3. Scenebuilder

5.2.1 System Specification

5.2.1.1 CPU

The central processing unit used by us is a general purpose HP Intel I-5, with 8GB of RAM, printer parallel port and PCI express slot. The envisioned system would involve supercomputers.

5.2.1.2 Display

We are using a simple LCD monitor but the envisioned system would be incorporated in the telecommunication companies in their respective systems.

5.2.1.3 Harddrive

We are using 500GB internal hard drive to store the data. But for future usage external hard drive might be included as the envisioned system would include huge data warehouses.

5.3 Implementation

System consists of distinct features .Each feature has its own implementation methodology

5.3.1 Contact List extraction

5.3.1.1 Methodology

This feature extract contact list of user on the basis of contact number entered by operator. When the user enter contact number then the search mechanism on the basis of domain goes to particular partition or file group of data instead of traversing entire data. Then multithreading is implemented to make parallel implementation possible. Threads are implemented on the basis of number of cores as in case of 5 cores 5 threads are implemented such that data of particular partition is divided into 5 parts and each thread is working on different chunk of data. Contacts after getting results from all threads are combined .If frequency sorting is required hash maps are used as they sort data faster than linear searching because their complexity is $O(1)$ where date sorting is done by using comparable interface. Contact list is displayed in tabular format .Table is generated by using javafx table and then row are created dynamically based on number of

records. Contact list can also be saved in file format by creating workbook and worksheet. Cells of work sheet is created dynamically based on number of records. For creating workbook dynamically java API apache “poi” is used. The contact list can also be printed. Printing is done by using java printer library.



Figure 5-1: System Methodology

5.3.1.1.1 File groups /partition

Database is horizontally partitioned on the bases of domain so that data from each network is located in separate file group .Primary file group is divided into 6 files such that one file for each network. By creating separate files we reduce the search area so that instead of traversing entire data to get particular records, the database searches in one particular partition based on the domain of user 'mobile number for example Mobilink number is searched only in Mobilink partition to find its contact list. The data is logically partitioned.

For creating partition query is written in sql editor and clustered index is defined on domain and Call_id.

5.3.1.1.2 Multithreading

Multithreading is done to efficiently use system resources because in serial execution the CPU usage is very less. In order to use resources effectively we implemented java executor service by using interface class such that each thread is working on different chunk of rows belonging to particular partition. Number of threads are based of cores such that in case of 5 cores 5 threads are created .In code five threads are created by making 5 objects of "Query executor class". Each object implements different query in execute method of the interface .Query Executor class implements interface class query method execute and implements runnable interface. Partition id is defined which is unique inside partition and non-clustered index is defined on that such each execute method searches on different partition id range.

5.3.2 Search

5.3.2.1 Methodology

Search module is to find NIC and Name against particular contact number

Searching is done such that when user enter contact number then from phone number table NIC of person against that contact number is extracted which is then used to extract Name form personal details table in database

5.3.3 Email

The email module is used to send the contact list extracted in the form of an excel file to the customers via email. The mail.jar file is used. The smtp protocol is used with port number 587 that is used for both the gmail and the Hotmail/live accounts. The authentication check is set to true

If the operators email Id is of gmail than the server is set to gmail using

```
props.put("mail.smtp.host", "smtp.gmail.com")
```

Else the server is set to Hotmail using

```
props.put("mail.smtp.host", "smtp.live.com")
```

The JFileChooser is used to select the contact list excel file to be mailed to the customer

A new session is created for each email message the ID, password is authenticated once the authentication is confirmed the message subject and contact list file is made a part of the mime message body then the message is send using

```
javax.mail.Transport.send(message)
```

5.3.4 Help

The help module displays the detail of how to use the software with the detail of each step. The instructions are displayed on selecting the respective feature whose help required, this is achieved by clicking the specific button.

5.4 Application Issues

5.4.1.1 Yahoo sender address issue

Yahoo does not allow its port number to be used openly. Thus the sender id that is the operator's id cannot be that of yahoo. Google and Hotmail can be used as the operators address.

6. Project Analysis and Evaluation

6.1 Testing

To ensure quality of the product, testing is conducted. Accuracy and efficiency of tasks performed by our system had to be tested to analyze the system and verify and validate it. Software testing techniques and results obtained are discussed in the coming sections.

6.2 Testing Levels

Separate modules were developed to provide different functionalities of the system. All of these modules were tested at different levels during development and after integration. Different levels at which the system has been tested and results obtained are described in this section.

6.2.1 Unit Testing

Each module was developed and tested individually. Each software modules each and every component was tested at unit level to ensure that they were functioning properly.

6.2.1.1 Authentication Module

The User Name and Password Validity

Decision Table:

Condition	Rule 1	Rule 2
Username	Valid	invalid
Password	Valid	invalid
Action:	Case 1: Main menu screen displayed	Case 2: Invalid login message displayed

Table 6-1

Test cases:

<Username and password valid, Case 1 executed >

< Username and password invalid, Case 2 executed>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	Username and password valid	Case 1 executed	Case 1 executed	Pass
TC 2	Username and password invalid	Case 2 executed	Case 2 executed	Pass

Table 6-2

6.2.1.2 Extraction Module

Pair Wise Testing:

All Pairs Test

	Contact no	date	frequency	tabular	file
Row1	valid	selected		selected	
Row2	valid		selected	selected	
Row3	valid	selected			selected
Row4	valid		selected		selected
Row5	valid			selected	
Row6	valid				selected
Row7	valid	selected			
Row8	valid		selected		
Row9	invalid	selected		selected	
Row10	invalid		selected	selected	
Row11	invalid	selected			selected
Row12	invalid		selected		selected
Row13	invalid			selected	
Row14	invalid				selected
Row15	invalid	selected			
Row16	invalid		selected		

Table 6-3

Cases:

Case 1: Contact list sorted on the bases of date displayed in tabular format.

Case 2: Contact list sorted on the bases of frequency displayed in tabular format.

Case3: Contact list sorted on the bases of date displayed in file format.

Case 4: Contact list sorted on the bases of frequency displayed in file format.

Case 5: Contact list sorted on the bases of date displayed in tabular format.

Case 6: Contact list sorted on the bases of date displayed in file format.

Case 7: Format select message displayed.

Case 8: Format select message displayed.

Case 9: No contact list displayed.

Case10: No contact list displayed.

Case 11: Empty file created.

Case 12: Empty file created.

Case13: No contact list displayed.

Case 14: Empty file created.

Case 15: Format select message displayed

Case 16: Format select message displayed

Test cases:

<Contact no valid and date and tabular selected, Case 1 executed >

< Contact no valid and frequency and tabular selected, Case 2 executed>

< Contact no valid and date and file selected, Case 3 executed >

< Contact no valid and frequency and file selected, Case 4 executed>

< Contact no valid and tabular selected, Case 5 executed >

< Contact no valid and frequency selected, Case 6 executed>

< Contact no valid and date selected, Case 7 executed >

< Contact no valid and frequency selected, Case 8 executed>

< Contact no invalid and date and tabular selected, Case 9 executed >

< Contact no invalid and frequency and tabular selected, Case 10 executed>

< Contact no invalid and date and file selected, Case 11 executed >

< Contact no invalid and frequency and file selected, Case 12 executed>

< Contact no invalid and tabular selected, Case 13 executed >

< Contact no invalid and frequency selected, Case 14 executed>

< Contact no invalid and date selected, Case 15 executed >

< Contact no invalid and frequency selected, Case 16 executed>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	contact no valid and date and tabular selected	Case 1 executed	Case 1 executed	Pass
TC 2	contact no valid and frequency and tabular selected	Case 2 executed	Case 2 executed	Pass
TC 3	contact no valid and date and file selected	Case 3 executed	Case 3 executed	Pass
TC 4	contact no valid and frequency and file selected	Case 4 executed	Case 4 executed	Pass
TC 5	contact no valid and tabular selected	Case 5 executed	Case 5 executed	Pass
TC 6	contact no valid and frequency selected	Case 6 executed	Case 6 executed	Pass
TC 7	contact no valid and date selected	Case 7 executed	Case 7 executed	Pass
TC 8	contact no valid and frequency selected	Case 8 executed	Case 8 executed	Pass
TC 9	contact no invalid and date and tabular selected	Case 9 executed	Case 9 executed	Pass
TC 10	contact no invalid and frequency and tabular selected,	Case 10 executed	Case 10 executed	Pass
TC 11	contact no invalid and date and file selected	Case 11 executed	Case 11 executed	Pass
TC 12	contact no invalid and frequency and file selected,	Case 12 executed	Case 12 executed	Pass
TC 13	contact no invalid and tabular selected	Case 13 executed	Case 13 executed	Pass
TC 14	contact no invalid and frequency selected	Case 14 executed	Case 14 executed	Pass
TC 15	contact no invalid and date selected	Case 15 executed	Case 15 executed	Pass
TC 16	contact no invalid and frequency selected	Case 16 executed	Case 16 executed	Pass

Table 6-4

6.2.1.3 Search Module

Enter contact number

Condition	Rule 1	Rule 2
Contact no	Exists	Does not exists
Action:	Case 1: Name and NIC displayed	Case 2: No search found message displayed

Table 6-5

<Contact number exists, name and nic displayed>

< No contact number exists, no search found message displayed>

Test case ID	Input	Expected output	Actual output	Status (pass/fail)
TC 1	Contact number exists	Name and NIC displayed	Name and NIC displayed	Pass
TC 2	Contact number not exists	no search found message displayed	no search found message displayed	Pass

Table 6-6

6.2.1.4 Menu Module

Test cases

<Extracts contacts, extraction screen>

< Search, Search screen>

< Email, Email screen>

<Help, help screen >

Test Case Id	Input	Expected Result	Actual Result	Status
TC1	extracts 'contacts	Extraction screen shown	Extraction screen shown	Pass
TC2	Search	Search screen shown	Search screen shown	Pass
TC3	Email	Email screen shown	Introduction displayed	Pass
TC4	Help	Help screen displayed	Help screen displayed	Pass

Table 6-7

6.2.1.5 Print Module

Test cases:

<Print table, table printed>

<print file, file printed>

Test Case Id	Input	Expected Result	Actual Result	Status
TC1	Print Table	Table printed	Table printed	Pass
TC2	Print file	File printed	File printed	Pass

Table 6-8

6.2.1.6 Email Module

Test cases

Pair wise

Use orthogonal array to design test cases:

	Sender email id	Sender password	Receiver id
Row 1	valid	Valid	valid
Row 2	valid	Invalid	invalid
Row 3	invalid	Valid	invalid
Row 4	invalid	Invalid	valid

Table 6-9

Test cases:

Four test case scenarios

<Sender email id and sender password and receiver id valid, mail sent>

<Sender email id valid but sender password and receiver id invalid, mail not sent>

<Sender email id and receiver id invalid but sender password valid, mail not sent>

<Sender email id and sender password invalid but receiver id valid, mail not sent>

Test Case Id	Input	Expected Result	Actual Result	Status
TC1	sender email id and sender password and receiver id valid	mail sent	mail sent	Pass
TC2	sender email id valid but sender password and receiver id invalid	mail not sent	mail not sent	Pass
TC3	sender email id and receiver id invalid but sender password valid	mail not sent	mail not sent	Pass
TC4	sender email id and sender password invalid but receiver id valid	mail not sent	mail not sent	Pass

Table 6-10

6.2.1.7 Help Module

Test cases:

<extracts contacts, extraction's help open>

< Search, Search's help open>

< Email, Email's help open>

<Help, help screen >

Test Case Id	Input	Expected Result	Actual Result	Status
TC1	extracts 'contacts	Extraction 's help open	Extraction 's help open	Pass
TC2	Search	Search 's help open	Search 's help open	Pass
TC3	Email	Email 's help open	Email 's help open	Pass

Table 6-11

6.2.2 Integration Testing

This Test Plan describes the integration tests that will be conducted on the implementation of contact list retriever software.

The interfaces between the following modules will be tested:

1. Data generator
2. Search
3. Authentication
4. Extractor of contact list
5. Mailer
6. Printer
7. Menu

It is assumed that unit testing already provided.

The external output device, the printer, is not part of the software to be created. The computer the user is using will have printers available. The report on retrieved contact list can be printed on request by the user.

The internet connection will be available either by Ethernet or Wi-Fi.

SYSTEM ARCHITECTURE:

The aim is to check if the modules such as menu, search, extraction and mailer after integrating with each other perform as they were required. The user of the system should be able to login after giving valid credentials and should be able to perform functionalities like search, extract the contact details of customer, print or send it through mail to customer. The system performance would be tested under valid and invalid inputs.

Functional Testing will focus on each and every use case that is included in the version currently being worked on. Testing will mainly consists of execution of test cases written to address the gap identified. It will focus on inputs, outputs and system changes due to the actions.

1. Login authentication
2. Search number
3. Extract contact list of customer
4. Read help
5. Generate excel file
6. Display in tabular format
7. Print the file or table
8. Attach and send generated file though email.
9. Performance testing ensures scalability, throughput and endurance.

For this purpose we are using the incremental approach to integrate the modules together.

Data Generation:

We start with the two basic modules for data generation which are FINALPROJECT and records. We will first perform the test cases to validate their interaction. We will make sure that FINALPROJECT is successfully reading external resource files and records is getting all the generated data and storing it into correct database in SQL SERVER 2008 R2.

Data Extraction:

After this we can start integrating the modules related to retrieval of records.

First we will integrate the tabular module with the extraction module and make sure that it provides the correct results in tabular format. We will then integrate it with the file creation module to see if the results are correctly added to the excel file. In case of any errors, after debugging and satisfaction we will incrementally add printer module to check that if the module after integration with the already integrated modules works fine and doesn't disturb their performance as well. If the new module still prints the file as well as table of results correctly, then we will add mailer module to it and check if the file can be automatically attached and mailed to the required client/subscriber.

Then we will add menu module so that we can check if all the functionalities can be reached by user (operator/administrator) using the main menu. Next the search, help and manual attachment modules are also integrated with the menu.

At the end we will add authenticator module to the system to see if the functionalities can only be accessed via the authorized user.

The advantage is that the integration testing process can start earlier, as soon as related modules have been successfully unit tested. An incremental approach makes it easier to find errors since the application environment introduces only one (or a few) modules at a time. Finally, this approach results in more overall testing, since the earlier modules get tested repeatedly as you add new modules.

6.2.2.1 Structure of Integration levels:

6.2.2.1.1 Integration test phases:

a. First phase of testing

In this phase tests will be designed to uncover functional errors in each module after it is integrated with the system. Errors associated with local or global data structures will be uncovered with such tests.

After integration of modules of extraction and file module we will see if the tables of contact details is being generated correctly. Previously designed unit tests for checking the functionality of extraction module i.e., extracting results in table will be executed again to see if the integration has caused any errors such as finding wrong contacts.

Similarly, unit tests designed for file module like creation of a file at specified folder and location with correct fields will be performed again.

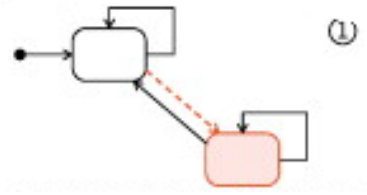


Figure 6-1: Integration testing (a)

Then integrating print module test cases designed in unit testing for checking functions of print e.g. printing the selected file will be validated. Then after integration of the mailer module it is checked if the mail can still be sent through system.

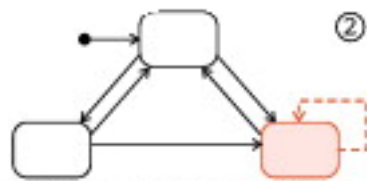


Figure 6-2 Integration testing (b)

Then menu module will be integrated with above integrated modules and again functional validity will be performed and checked whether the interface of module are in consistent with each other.

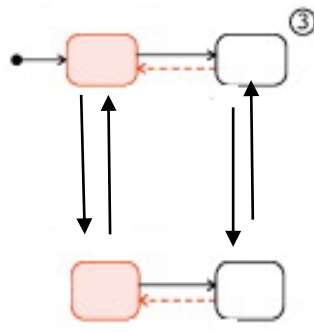


Figure 6-3 Integration testing (c)

b. Second phase of testing

Pair wise validity will be performed in which we will combine two modules together and check the functionality which is dependent on both modules.

First menu and extraction modules will be tested to see if the correct option is displayed i.e. extraction screen.

Then pair of extraction and file module would be tested. Extracted contacts for given number are passed to file module so that excel file of those extracted contacts is made at specified location.

Then print module is integrated which takes information from file module and prints automatically the recently generated file which is functionality of printer module. Similarly the mailer module is integrated and tested.

Then menu module is tested in pairs with help, search and printer (manual attachment).

At the very end menu module is integrated with authentication module to see if only after the validation of user menu is displayed to him.

In end to end testing test cases will be performed which include testing of almost all modules like user getting authenticated the chooses the extraction of contact list option from menu and saves the results in from of excel file and prints it for walk in customer.

b. Third phase of testing

In third phase endurance testing will be performed that how much stress our system can with stand i.e. how many users at a time can use the system and for how many days or hours our system can keep working without breakdown.

6.2.2.2 Modules to be integrated in each phase:

a. First phase

In first phase we need to check functional validity of all the modules. So, we will use all of our modules in this phase.

b. Second phase

In end to end tests. We will utilize all of our modules to perform the task starting from one of the module (e.g. menu) to the module that accepting information from other modules (e.g. tabular and file) produces some final results (mail or print module).

c. Third phase

For checking the stress the system can withstand we will again require our whole integrated system in place. This phase validates that critical functions will meet production performance requirements during peak transaction volumes.

6.2.2.3 Building process and Schedule for each phase:

a. First phase

The first phase starts on Monday, 22 April 2014. The phase checks the functional and interface validity of each module and takes about 3 hours. So ends on 22 April 2014.

b. Second phase

The second phase checks end to end validity. It will start on Tuesday, 23 April 2013 after the 1st phase. It will be completed in 5 hours.

c. Third phase

The endurance testing will be performed as last phase in integration testing. It can start on Wednesday, 24 April 2013 .System endurance will be checked by testing it for 3 continuous hours under different conditions and for valid and invalid inputs.

6.2.2.4 Environment to be set up and resources required in each phase:

Make sure that computer is running Windows 7 or 8 with min 4GB RAM and properly configured. Databases are available for the content store.

- i. 64bit windows OS
- ii. SQL server 2008 R2
- iii. Testers and developers set up manual and configuration guides
- iv. Printer
- v. Internet connection

6.2.2.5 Criteria for each Integration test phase n:

a. First phase:

i. Entry criteria:

Entry criteria for functional testing is that each module has been unit tested individually and all errors have been corrected. Interface integrity ensures that the two modules communicate with each other in the desired way. The entry criteria for this phase is that all the modules should have undergone unit testing and are working independently.

ii. Exit criteria:

Exit criteria for the first phase is fully functionally tested modules integrated together.

For example in our system all the modules are incrementally integrated. In the end we get a successfully **functionally** tested integrated system. The integrated system has passed all the integration tests and no defect is outstanding.

iii. Integration Technique to be used:

Incremental technique will be used

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

b. Second phase:

i. Entry criteria:

Entry criteria for End to end and pair wise validity is that we have functionally tested and integrated system to check for communication between modules. For example extraction, file, printer, mailer and menu are functionally tested and after passing through first phase can enter 2nd phase.

ii. Exit criteria:

Exit criteria is that we get a fully integrated system with no more modules to integrate, All the system integration tests for the phases have passed, all the defects found have been fixed and also documented.

iii. Integration Technique to be used:

Incremental technique will be used.

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

c. Third phase:

i. Entry criteria:

The entry criteria of this phase is that we have a validated system that works together from end to end. There are no major defects remaining while testing the end to end working of the integrated system.

ii. Exit criteria:

The exit criteria of this phase is that we have an integrated system to which when the load is applied such as using the printer, mailer etc., it can sustain the load and does not crash. Also system endurance is ensured during this phase so that in the end we have a system that can work for hours without crashing. This is important as the crashing of the system can lead to loss of the results that had already been found from database causing delays in service provided to customers.

iii. Integration Technique to be used:

Incremental technique will be used.

iv. Test configuration set-up:

Normal test conditions are required (computers and environment with normal temperature).

6.2.2.6 Test specification for each integration phase:

a. First phase:

This phase will include performing tests on integrated modules to check their interface integrity and functional validity. Same test cases as used in unit testing for modules functionality are tested again by replacing stubs with actual modules. These are given below:

i.Interface Integrity

Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
TC1	Wrong username	System has executed and login screen is opened	When screen appeared for username and password ,wrong username input from keyboard is made	Display error message	Error message	Pass
TC2	Wrong password	System has executed and login screen is opened	When screen appeared for username and password ,wrong password from keyboard is made	Display error message	Error message	Pass
TC3	Correct username and password combination	System has executed and password screen is opened	When screen appeared for password , input from keyboard is made	Main menu screen shown	Main menu screen shown	Pass
TC4	Search	User	From the main	Search screen	Search screen	Pass

	button clicked	successfully logged in	menu search button is pressed through mouse	displayed	displayed	
TC5	Extraction of contact list clicked	User successfully logged in	Select check-in option from the menu by using mouse	Extraction screen displayed	Extraction screen displayed	Pass
TC6	help	User successfully logged in	Select help option from the menu by using mouse	Help is provided	Help screen is displayed	Pass
TC7	mail	User successfully logged in	Select display option from the menu by using mouse	Screen for mailing your file is provided which allows user to attach any file from his system and send to the customer.	Mailer screen is displayed	Pass
TC8	exit	User successfully logged in	Select display option from the menu by using mouse	System exits	System exits successfully	Pass

Table 6-12

b. Second phase:

This will include designing test cases to check end to end validity. In our system this can be checked by designing test cases that will start from getting user's credentials from the login screen till the result is retrieved.

For Pair Wise:

Modules integrated	Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
Authentication and menu	TC1	Correct Username and password pair	System is in running condition with login screen displayed	Enter the correct username and password combination from keyboard	Main menu is displayed	Main menu is displayed	Pass
Menu and search	TC2	Search button clicked	User was successfully logged in	Select search option from the menu by using mouse	Search screen displayed	Search window is displayed	Pass
Menu and search	TC3	Domain and cellphone number	User is in search window	In the search screen choose the domain from drop down menu and 7 digit number is entered in text field	Customer's name and NIC is displayed on same screen	Customer's name and NIC is displayed on same screen	Pass
Menu and extraction	TC4	Extraction of contact list pressed	User successfully logged in	Select extraction option from the menu by using mouse	Extraction screen displayed	Extraction screen displayed	Pass
Extraction and tabular	TC5	Domain chosen value number entered and tabular option from radio button	Extraction window is displayed	Domain is chosen from drop down menu and a valid 7 digit number in text field and tabular is chosen from radio button	Table containing the contact details of the entered number is generated	Table containing the contact details of the entered number is generated	Pass

				options			
Extraction and file	TC6	Domain chosen value number entered and file option from radio button	Extraction window is displayed	Domain is chosen from drop down menu and a valid 7 digit number in text field and file is chosen from radio button options	File containing the contact details of the entered number is generated with the name same as provided number. Another screen with print and mail option displayed	File containing the contact details of the entered number is generated with the name same as provided number and another screen with print and mail option displayed	Pass
File and print	TC7	Print button is clicked	Print and mail option screen displayed and working printer is attached	Print option is selected from mouse which opens up the printer option window. Click ok	The contact details are printed	The contact details are printed	Pass
File and mail	TC7	Mail button is clicked and internet provided	Print and mail option screen displayed and working printer is attached	Mail option is selected from mouse which opens up the mail window. Fill the text fields with valid details and click send	The recently generated file is automatically attached and on clicking send email is sent to customer	The recently generated file is automatically attached and on clicking send email is sent to customer	Pass
Menu and mail	TC8	Mail button is clicked	Main menu is being displayed and	Mail button is clicked from main menu.	The mail window opens up with the	Mail window opened with the empty attachment	Pass

			internet connection available	Browse for the file that needs to be sent and click send	attachment field empty that should be filled on browsing the file.	filed which is filled with file path on browsing	
Menu and help	TC9	Help button pressed	User successfully logged in with main menu appeared on screen	Help is chosen option from menu	Help screen displayed	Help screen displayed	Pass

Table 6-13

For the End to End:

For end to end testing we will perform a test which will include almost all the modules so that it can be checked that they all are working as required together.

We will start from user authentication, after that extraction of contact list option is selected from main menu which will open up the extraction screen. Choose the domain and enter the digit mobile number for which we want to extract the contact details.

Choose the frequency or date as order and file or tabular format for the extracted results.

In case of file, it should be generated on specified location with the entered number as file name and another window with print or mail option will be displayed. Selecting the mail option should open the mail window with automatically attached file ready to be sent at the provided email address.

c. Third phase:

This will include designing test cases to check system endurance:

Due to high availability, load test carries great importance. The metrics are as follows:

Metrics	Minimum	Good	Excellent
Maximum Number of Concurrent Operations Supported	2	5	10

Table 6-14

Endurance Test cases:

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	2 Weeks	System is executed	System functions are checked for 2 weeks without termination	No memory leak occurs, System resumes in case of power failure, System does not crash	No memory leak occurs, System resumes in case of power failure, System does not crash	pass
TC2	5 weeks	System is executed	System functions are checked for 5 weeks without termination	No memory leak occurs, System does not crash	No memory leak occurs	Pass

Table 6-15

6.2.2.7 Actual test results for each integration test phase:

The Actual results are already shown in the test cases specifications for each integration phase above.

6.2.3 System Testing

The goal of system testing should be to ensure that each work unit or module (those identified during unit testing) interacts with one another as designed. The document outlines the test cases, responsibilities and metrics of “Extraction of contact list from call details records” software. The current library of test cases helps us to keep development teams on track and also verify that system conforms to requirements which are described

in the requirement document of the “Extraction of contact list from call details records” software.

6.2.3.1 Basic Tests

a. Boot tests

Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
TC1	System launched by clicking button on system file that was saved in memory	Integrated and Working Version is Available	The exe file for the built system is clicked as it is the available boot option provided	The system boots up its image from memory	System gets its software image	Pass

Table 6-16

6.2.3.2 Functionality Tests

a. Communication System Tests:

The system shall be a product that requires communication with the printer. It also requires internet connection.

b. Module tests:

Same as done in integration testing.

c. GUI tests:

Test case id	Input	Initial condition	Test procedure	Expected output	Actual output	Status
TC1	Users checks the system's GUI	System has executed and screen is opened	When screen appeared. testing team checks whether the font is readable and text is interpretable	Team should find the GUI user friendly	Team found the GUI user friendly	Pass
TC2	Users calculate the system navigate time	System has executed and screen is opened	User navigates through whole GUI	The navigation time should not be more than ... minutes	The navigation time is ... minutes	Pass

Table 6-17

c. Security Tests

i. Availability and Integrity tests:

Test Case Id	Input	Initial conditions	Test procedure	Expected Result	Actual Result	Status
TC1	Valid login of operator	Integrated and Working Version is Applicable	Valid Username and password for the operator will be provided to the system	Operators will be able to log in to the Contact list retrieval software	Information validated and operator logged in	Pass
TC2	Valid System administrator's login	Integrated and Working Version is Applicable	Valid username and password for the system administrator will be entered and access to contact list retrieval software will be verified by selecting suitable option	System administrator will have access to the contact list retrieval software along with the option of adding a new operator.	User as system administrator logged in and access to given options provided	Pass

Table 6-18

ii. Confidentiality tests:

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	Invalid username and password	System is executed	Invalid login will be entered	Access to the various subsystems will be protected by a user log in screen that requires a user name and password and suitable error message displayed	Access not allowed	Pass
TC2	Operator's login information entered	System is executed	Login provided to the customer is entered and attempt to access "Adding a new operator" functionality will be made	Access to this functionality will not be provided with an error message displayed	Access denied and message displayed	Pass

Table 6-19

6.2.3.3 Scalability Tests

Not Applicable

6.2.3.4 Performance Test

Test Case Id	Input	Initial conditions	Test procedure	Expected output	Actual Result	Status
TC1	Launch button pressed	Integrated and Working Version is Available	Application will be launched several times and average load time will be calculated	The load time for user interface screens should not take more than 10 seconds.	Main screen appears in 5 seconds	Pass
TC2	Various valid login information	System is in running condition	System will be given different valid login requests and average time will be measured	The login information shall be verified in less than 5 seconds.	Main menu appears after verification of valid login in average of 3 seconds	Pass
TC3	Search option is selected and number entered	System is in running condition with operator or administrator or logged in	Query for finding personal details of customer will run	Queries shall return results within 1 minute with 25 GB data	Results are displayed in average of 45 seconds with 25GB data	Pass
TC4	Contact extraction option is selected and number entered	System is in running condition with operator or administrator or logged in	Queries for finding contact list of customer will run	Queries shall return results within average of 2 minutes on 25GB data	Results are displayed in average of 1 min with 25GB data	Pass
TC5	Mail option is selected	System is in running condition with operator or administrator or logged in and reliable internet connection is available	File to be sent is selected and attached	File should be attached after maximum of two trials and sent message should appear within 5 seconds	File is sent within 3 seconds	Pass

Table 6-20

6.2.3.5 Interoperability and Compatibility Tests

Test case id	Input	Test procedure	Expected output	Actual output	Status
TC1	Windows test	Execute the software on windows 7/8 and check every functionality that the system is compatible with windows platform	System should work on that environment	System works on that environment	Pass

Table 6-21

6.2.3.6 Robustness Tests

a. Boundary value tests:

Not applicable.

b. Power cycling tests:

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	Power glitch	System is executed	System checked that after power glitch any error occurred or not	System resumes in case of power failure, System does not crash It should not result in any hardware and software error. Database remains stable.	System resumes in case of power failure, System does not crash No hardware and software error found. Database remains consistent.	Pass

Table 6-22

c. High-Availability Tests:

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	Database is accessed	Active database is not available	System found that database is not currently available and automatically replace it with backup data	System should function properly without any apparent error	System functions properly without any apparent error	Pass

Table 6-23

6.2.3.7 Documentation Tests

Test Case Id	Input	Initial condition	Test Procedure	Expected Result	Actual Result	Status
TC1	User manual is viewed by testing team	System is not executed	Team reads the manual and execute the system side wise	System should be successfully executed and used by testing team	Team successfully used the system	Pass

Table 6-24

6.2.4 Acceptance Testing

This is acceptance test plan for “Extraction of contact list from CDR” software. By testing we want to achieve that the system matches the customer requirements and is acceptable by them. This is the first version of the software, it is built to provide ease in generating the contact list of any customer from call detail records maintained by Telecom Company. By this test plan we will test the software for the quality attributes according to the customer acceptance criteria.

Date of approval: 2nd May 2013.

i. Functional Correctness and Completeness

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Login In Feature	Is to test that the login in feature's functionality is according to what user requirement is and do the work it is programmed to do	Run the software and provide the correct username and password pair provided to user,if the menu appears,it will show that the test is successful
TC 2	Search Feature	Is to test that the search feature works as it is supposed to do according to requirements	Login , then press search from the menu interface. Enter the valid required number from database and press search. As the number was from database ,if a name and NIC against that number is retrieved correctly,it will be displayed and the test is successful
TC 3	File creation feature	Is to test that file is created containing all the records retrieved in the specified order(frequency or date)	Login , choose the extraction of contact list option from menu, enter the domain and number from database and choose file option from radio button. If file is created with retrieved records at specified location the test will be successful
TC 4	Table creation Feature	Is to test that table is created containing all the records retrieved in the specified	Login, choose the extraction of contact list option from menu, enter

		order(frequency or date)	the domain and number from database and choose tabular option from radio button. If table is displayed with retrieved records test will be successful
TC 5	Mail feature	Is to test that the file can be attached(manual or automatic) and sent through mail to customer	Login, then choose mail from menu, fill the required fields correctly and browse to attach the file. Press send. If the mail is received with the file at destination, test will be successful
TC6	Help feature	Is to test the help about the usage of the software can be made available on demand	Login, then press the help button. If the help appears, test will be successful

Table 6-25

ii. Accuracy

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Search	Is to test the accuracy of the result (name and NIC) found against the given number	Login , then press search.Search for the number from database whose details are already known.If the retrieved record is as expected,test will be successful
TC 2	Extraction of contact list in order of date	Is to test the accuracy of the number of results and the details of the result found in order of descending date	Login , then press extraction.Enter the number from database whose number of contacts and details are already known.Choose order:date from radio button and tabular or file option.If the retrieved record is as expected in descending date order ,test will be successful
TC 3	Extraction of contact list in order of frequency	Is to test the accuracy of the number of results and the details of the result found in order of decreasing frequency	Login , then press extraction.Enter the number from database whose number of contacts and details are already known.Choose order:frequency from radio button and tabular or file option.If the retrieved record is as expected in decreasing frequency order ,test will be successful

Table 6-26

iii. Data Integrity

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Call details record data	To test the data preservation i.e. data remain unchanged when retrieved by another operation for customer contact details data.	Login, press extraction and enter a number whose contact details are required in file format. Now again extract the records for the same number but in tabular format. If both operations yield same results, test is successful

Table 6-27

iv. Backup and Recoverability

Operational Environment:

The operational environment required to test the software is the operating system and database server, it requires. As it is not hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8. Backup call records and application is generated and is stored in a separate 1tb hard disk.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Personal Information records	To test that if system fails or shuts down the Personal Information record is safe and is recoverable	First login, save some personal information record then shut down the system abruptly Restart again and check if the record still safe through display feature If record same test successful. Record is saved in Database file
TC 2	Call detail records	To test that if system fails or shuts down Call detail record is safe and is recoverable	First login, save some call detail record then shut down the system abruptly Restart again and check if the record still safe through display feature If record same test successful. Record is saved in database file.
TC3	Contact list	To test that if system fails or shuts down contact list generated is safe and is recoverable	First login, generate contact list in excel format then abruptly shut down the system Restart again and check if the contact list exists there ,test successful Contact list is saved in excel file.

Table 6-28

v. Usability**Operational Environment:**

The operational environment required to test the software is the operating system and sql server 2008 it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	User Interface: Main screen	The objective is to check whether the main screen provides enough options that are easy to use so that the user can navigate through the system.	Login this will take user to the main screen. Various options will be present through which the user can select the desired option. If the user clicks extract's contact he will go to the extraction screen. Pressing search will take the user to the search screen, help click open application's help, for email user press email button.
TC 2	User Interface: Extract contact	Determine whether the user can perform the contact list extraction function with relative ease.	Login and then select extract contact from main menu ,it will take user to extraction screen where user select domain, enter contact number ,select format and order of resultant contact list ,then resultant contact list will be generated accordingly. Examine whether the required fields have been provided or not.
TC 3	User Interface: search	Determine whether the user can perform the search user function with relative ease.	Login and then select search from main menu ,then search screen will be opened where user enter contact number and select domain the name and NIC of person if exists in personal records file will be displayed. Examine whether the required fields have been provided or not.
TC4	User Interface: help	Determine whether the user can access help with relative ease.	Login and press help from main menu .User select help of particular feature that he wants to access, Examine whether user performs the operation with ease
TC5	User Interface: email	Determine whether the user can email the person contact list with relative ease.	Login and then press email, then email screen will open, user enter sender email id and receiver email id and attach file to be sent. Will be displayed. Examine whether the required fields have been provided or not.

Table 6-29

vi. Performance

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows XP/7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Display response time	To test the systems displaying response time according to the customer criteria	Login , then press the option to extract contact list or search then check the response time until display the data
TC 2	Opening response time	To test the systems opening time according to the customer criteria	Open the software and check the response time until login screen shows up.

Table 6-30

vii. Robustness

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Login Screen	Determine how the system reacts when erroneous inputs are given.	Login invalid input generates message of invalid login and asks to re-enter
TC 2	Extract contact list	Determine how the system reacts when erroneous inputs are given.	Enter contact number. If format from file or tabular is not selected it displays an error message.
TC 3	Search	Determine how the system reacts when erroneous inputs are given.	Login and enter the main menu screen. Press search button and enter the invalid number i.e. less or more than 7 digits. Observe the behavior of the system. The system displays "invalid format "message.
TC 4	Extraction of contact list	Determine how the system reacts when erroneous inputs are given.	Login and then press extraction. Enter the wrong/invalid number. The system gives the "invalid format "message.
TC 5	Operating time	Determine for how long the system can work reliably for an extended period of time.	Login and then give various inputs to the system. Measure the time the system can operate without crashing down.

Table 6-31

viii. Confidentiality

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Login	Determine the protection of data from unauthorized disclosure.	Start the system. It will ask the user to enter username and password on the login screen. If the password and username pair is correct the main menu screen will be displayed. Otherwise the system will display the error message to re-enter thus protecting data from unauthorized disclosure.

Table 6-32

ix. Compatibility and interoperability**Operational Environment:**

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

Test Case ID	Title	Objective	Procedure
TC 1	Windows test	To test whether or not system work on the platform	Execute the software on windows 7/8 and check every functionality that the system is compatible with windows platform

Table 6-33

x. Load Testing

Operational Environment:

The operational environment required to test the software is the operating system it requires. As it is not a hardware related software so it only requires a computer and operating system to be executed i.e. Windows 7/8.

Test Case Specifications:

The software is not a server base software so maximum no. of users that can operate it at a time is one and all the test cases described will also check for the load testing as it will be running on maximum load.

Schedule:

The system is medium and no. of test cases are acceptable so it will take one to two weeks for the acceptance testing to complete.

Human Resource:

The following testers will carry acceptance testing for the Call details records software

1. Aida Zahid
2. Hina Aziz
3. Amna Feroze

6.3 Summary

Testing not only maintains the software and system quality but also improves over all usability and stability of the project. At different stages of development suitable testing techniques were used to ensure product worked accurately and efficiently. Almost all the errors detected during testing were removed.

6.4 Result and Analysis

6.4.1 Results and Analysis Introduction

The Extraction of Contact List system has been developed to work in real time environment. The system was intended to generate records efficiently.

6.4.2 Results

The results were as per our expectations we had considerably reduced the time of searching by applying parallel implementation and use of file groups. With available resources the retrieval time is reduced considerably which is well depicted by the tables shown below.

Results with different resources on 25GB data:

Core: i7 16GB RAM

Serial execution	Parallel execution with 2 threads	Parallel execution with 5 threads	Parallel execution with 2 threads with file groups	Parallel execution with 5 threads with file groups	Parallel execution with 7 threads with file groups
17 mins 52 sec	12min	8 mins	35 secs	29 sec	20 sec

Table 6-34

Core: i3 4GB RAM

Serial execution	Parallel execution with 2 threads	Parallel execution with 5 threads	Parallel execution with 2 threads with file groups	Parallel execution with 3 threads with file groups
65 mins	55 mins 40 sec	38 mins 35 sec	13 mins	11 min 22 sec

Table 6-35

Core: i5 8GB RAM

Data record: 14GB 15 Million Records

Serial execution	Parallel execution with 2 threads	Parallel execution with 5 threads	Parallel execution with 2 threads with file groups	Parallel execution with 5 threads with file groups
18 min	12 min 22 sec	10 min 20 sec	42 sec	36sec

Table 6-36

Data record: 24GB 255 Million Records

Serial execution	Parallel execution with 2 threads	Parallel execution with 5 threads	Parallel execution with 2 threads with file groups	Parallel execution with 5 threads with file groups
40 min	34 min 38 sec	18 min 20 sec	1min 20 sec	40sec

Table 6-37

7. Conclusion and Future Work

To implement efficient and fast retrieval of data techniques such as parallel implementation and partitions implemented in this particular project can be used in any data warehouse comprising of huge amount of data to improve the searching mechanism.

The data set we have can be used for multipurpose .It is analyzed to determine which type of packages/services should be provided to users for example night packages or special call rates at particular time in a day when usage is high and so many other special offers and services. The data set can also be used to determine the indirect network of people who is connected to whom with x number of peoples in between. This feature can be very useful for security agencies to find the network of criminals.

We hope that one day our nation will be among the fore-runners in the field of science, research and information technology and Pakistan will earn its rightful place among the international community. Aamin!

Appendix A: Glossary

Algorithms: A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps.

Assumption: A statement that is believed to be true in the absence of proof or definitive knowledge.

Availability: Present and ready for use; at hand; accessible.

Billing: The process of sending a bill (also called an *invoice* or *accounts receivable*) to customers for goods or services.

Capacity: The ability to receive, hold, or absorb.

CDR: Call Detail Record comprising of information about the calls made by a person including *call id*, *Domain*, *primary call id*, *called id*, *Start time*, *connect time*, *end time*, *date*.

Class: A description of a set of objects having common properties and behaviors, which typically correspond to real-world items (persons, places, or things) in the business or problem domain.

Class diagram: An analysis model that shows a set of system or problem domain classes and their relationships.

Compatibility: Capable of orderly, efficient integration and operation with other elements in a system with no modification or conversion required.

Constraint: A restriction that is imposed on the choices available to the developer for the design and construction of a product.

Database: A collection of data arranged for ease and speed of search and retrieval. Also called *data bank*.

Data Extraction: The act or process of retrieving data out of data sources.

Data flow diagram: An analysis model that depicts the processes, data collections, and flows among them that characterize the behavior of a business process or of a software system.

Data retrieval: In database management, data retrieval involves extracting the wanted data from a database.

Data Mining: The automatic extraction of useful, often previously unknown information from large databases or data sets.

Data Warehousing: Massive database (typically housed on a cluster of servers, or a mini or mainframe computer) serving as a centralized repository of all data generated by all departments and units of a large organization.

Dependency: A reliance that a project has on an external factor, event, or group outside its control.

Driver: A software component used to interact with hardware devices, or used in electronic or pneumatic equipment as a force for cohesive equipment or individual machinery components.

External interface requirement: A description of an interface between a software system and a user, another software system, or a hardware device.

Feature: A set of logically related functional requirements that provides a capability to the user and enables the satisfaction of a business objective.

Firewall: A firewall establishes a barrier between a trusted, secure internal network and another network.

Flowchart: An analysis model that shows the processing steps and decision points in the logic of a process or of a program.

Functional requirement: A statement of a piece of required functionality or a behavior that a system will exhibit under specific conditions.

Hardware Interface: The logical and physical characteristics of each interface between the software product and the hardware components of the system.

IBM: The **International Business Machines Corporation** (commonly referred as **IBM**) is an American multinational technology and consulting corporation. IBM manufactures and markets computer hardware and software, and offers infrastructure, hosting and consulting services in areas ranging from mainframe computers to nanotechnology.

IEEE: The Institute of Electrical and Electronics Engineers. A professional society that maintains a set of standards for managing and executing software and systems engineering projects.

Implementation: execution of a plan, idea, model, design, specification, standard, algorithm, or policy.

Integrate: To make into a whole by bringing all parts together; unify.

Interactive Design: A user-oriented field of study that focuses on meaningful communication of media through cyclical and collaborative processes between people and technology.

Interface: A point where two systems, subjects, organizations, etc., meet and interact.

Issue List: This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.

JDBC: A Java-based data access technology from Oracle Corporation. This technology is an API for the Java programming language that defines how a client may access a database.

Map Reduce: A programming model for processing large data sets with a parallel, distributed algorithm on a cluster.

Marketing: is the process of communicating the value of a product or service to customers, for the purpose of selling the product or service. It is a critical business function for attracting customers.

Nonfunctional requirement: A description of a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behavior.

Open Source: In production and development, **open source** as a development model promotes a universal access via free license to a product's design or blueprint.

Operating Environment: The circumstances surrounding and potentially affecting something that is operating.

Operating System: A collection of software that manages computer hardware resources and provides common services for computer programs.

Oracle: The **Oracle Database** (commonly referred to as **Oracle RDBMS** or simply as **Oracle**) is an object-relational database management system produced and marketed by Oracle Corporation.

Perspective: The way in which objects appear to the eye.

Port: An application-specific or process-specific software construct serving as a communications endpoint in a computer's host operating system. A port is associated with an IP address of the host, as well as the type of protocol used for communication.

Post condition: A condition that describes the state of a system after a use case is successfully completed.

Precondition: A Condition that must be satisfied before a use case may begin.

Procedure: A written description of a course of action to be taken to perform a given activity, describing how the activity is to be accomplished.

Process: A sequence of activities performed for a given purpose. A *process description* is a documented definition of those activities.

References: List of any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

Resolution: The fineness of detail that can be distinguished in an image, as on a video display terminal.

Requirement: A statement of a customer need or objective, or of a condition or capability that a product must possess to satisfy such a need or objective. A property that a product must have to provide value to a stakeholder.

Requirements specification: See software requirement specification and specification, requirement.

Scope: The portion of the ultimate product vision that the current project will address. The scope draws the boundary between what's in and what's out for the project.

Server: A system (software and suitable computer hardware) that responds to requests across a computer network to provide, or help to provide, a network service.

Service provider: A company that provides organizations with consulting, legal, real estate, education, communications, storage, processing, and many other services.

Software requirements specification: A collection of the functional and nonfunctional requirements for a software product.

Specification, requirements: The process of documenting a system's requirements in a structured, shareable, and manageable form. Also, the product from this process.

SQL: Structured Query Language is a special-purpose programming language designed for managing data held in a relational database management system.

Stimulus: Something causing or regarded as causing a response.

Super Computing: A computer at the frontline of contemporary processing capacity-- particularly speed of calculation.

Supplementary Information: Something added to complete the information.

System requirement: A top-level requirement for a product that contains multiple subsystems, which could be all-software or software and hardware.

TCP/IP: The **Internet protocol suite** is the networking model and a set of communications protocols used for the Internet and similar networks. It is commonly known as **TCP/IP**, because its most important protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP) were the first networking protocols defined in this standard.

Telecommunication: **Telecommunication** is communication at a distance by technological means, particularly through electrical signals or electromagnetic waves.

Telecommunication Services: A company that provides telecommunications services such as telephony and data communications access.

TB: Tera Byte.

Usability: Fit for use; convenient to use.

Use case: A description of an interaction between an actor and a system that results in an outcome that provides value to the actor.

Use case diagram: An analysis model that identifies the actors who can interact with a system to accomplish valuable goals and the various use cases that each actor will perform.

User class: A group of users for a system who have similar characteristics and requirements for the system.

User requirement: User goals or tasks that users must be able to perform with a system, or statements of the user's expectations of system quality.

Validation: The process of evaluating a work product to determine whether it satisfies customer requirements.

Verification: The process of evaluating a work product to determine whether it satisfies the specifications and conditions imposed on it at the beginning of the development phase during which it was created.

Bibliography

- [1] Call-detail-records-cdrs
,<https://help.sumofiber.com/index.php?/Knowledgebase/Article/View/47/16/call-detail-records-cdrs>
- [2] Document: Generating Call Detail Records (Teles a Gate) Rev 1.3 December 2010.
- [3] High Performance MY SQL (Baron Schwartz, Peter Zaitsev)
- [4] SQL Server Technical Article: Column store Indexes for Fast Data Warehouse Query Processing in SQL Server 11.0 (Eric N. Hanson)
- [5] SQL Server -2005-Database Table Partitioning Tutorial- How to Horizontal Partition Database Table.
- [6] Analysis of Telephone Call Detail Records Based on Fuzzy Decision Tree (Liping Ding, Jian Gu, Yongii Wang, Jingzheng Wu).
- [7] Social Network Modelling: Retrieval Correlated Graphs By Mobile Phone's Chronological Billing Files (Hamid Zargari Asl, *hamid_zargari@yahoo.com*) Rev 1.3 December 2010.
- [8] Mining Social Network from Spatio-Temporal Events. Computational & Mathematical Organization Theory (Hady W. Lauw, Ee-Peng Lim, Teck-Tim Tan, Hwee-Hwa Pang) Vol.11, No 2, July, 2005, pp.97-118.