SYSTEM DESIGN DOCUMENT

SYSTEM DESIGN DOCUMENT

1 INTRODUCTION

1.1 Purpose and Scope

To develop an interface for name searching and indexing.Microsoft Office Access is used to maintain the whole database. Data will be loaded in the central database for display

1.2 Project Executive Summary

1.2.1 System Overview

PRODUCT PRESPECTIVE:

- When a name is given as the input, the program should be able to search for similar sounding names and return those names.
- The time taken for execution should be minimum.
- The program should be able to handle exceptions i.e., when the user gives a number or a special character as the input, the program must be able to handle it efficiently.
- The program should not be case-sensitive

1.2.2 Design Constraints

- GUI is only English
- Users should can type the information in the column provided for them and click search.
- The system is working for single server
- There is always a backup of data available. So availability will never get affected.
- User is required to type the alphabets (strictly alphabets)on which he intends to search for the name
- If user types numerical or alphanumerical or symbols, error will be raised by the system

1.2.3 Future Contingencies

TIE THE WEBSITE WITH EXISTING SYSTEM:

Any existing websites which requires searching and indexing can be linked with website

1.3 Document Organization

OVERALL DESCRIPTION:

Will describe major components of the system and external interface.

SPECIFIC REQUIREMENTS:

Will describe the functions at different levels of execution and the algorithm used.

1.4 Points of Contact

TEAM MEMBERS:

Anusha Venkat
Narendiran
Uma Maheshwari

1.5 Glossary

- **S/W:**

  Software.

- **H/W:**

  Hardware.

- **JSP:**

  Java Server Page.

- **HTTP:**

  hyper text transfer protocol is a trasaction oriented client/server protocol between web browser and web server.

- **HTTPS:**

  secure HTTP over SSL(secure socket layer)
  .

- **USER:**

  Access the interface for searching names.

- **JDK:**

  Java Development Kit.

- **MS ACCESS:**

  MicroSoft Access. Database software developed by Microsoft.

- **APACHE TOMCAT:**

  Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications and provides a "pure Java" HTTP web server environment for Java code to run.

- **TCP/IP:**

  Transmission control protocol/ Internet protocol, the suite of communication protocols used to connect hosts on the internet. Tcp/Ip uses several protocols, the main one being TCP/IP.

- **ADMINISTRATOR:**

  Responsible for updating the data in the database repository.

2. REQUIREMENTS:

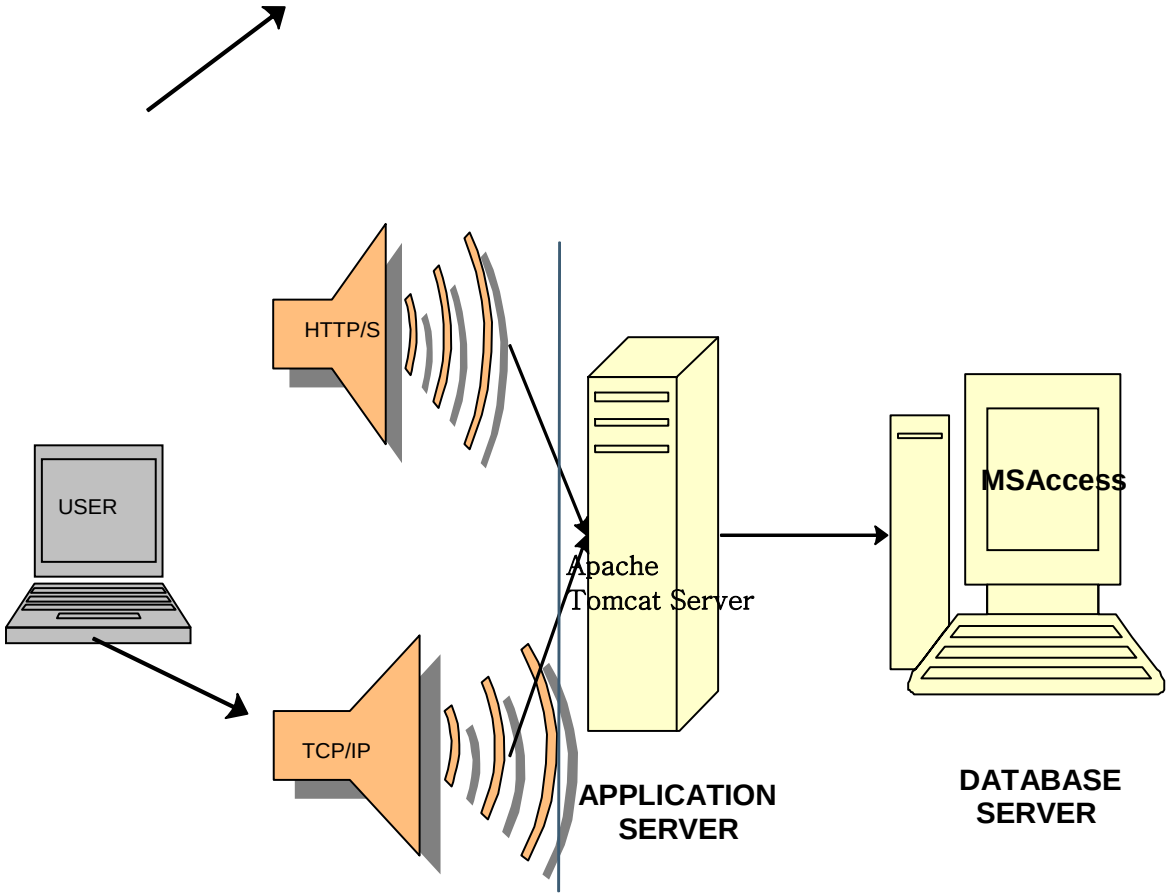
FUNCTIONAL REQUIREMENTS:

- When a name is given as the input, the program should be able to search for similar sounding names and return those names.
- The time taken for execution should be minimum.
- The program should be able to handle exceptions i.e., when the user gives a number or a special character as the input, the program must be able to handle it efficiently.
- The program should not be case-sensitive.

NON-FUNCTIONAL REQUIREMENTS:

- Only authenticated person should be allowed to edit and update the database.

3 SYSTEM ARCHITECTURE



- The web pages (HTML, .jsp) are present to provide the user interface. Communication between user and server is provided through HTTP/HTTPS and TCP/IP.
- On the server side web server is for interlinking database and users and database server is for storing the information.

3.1 System Hardware Architecture

CLIENT SIDE	PROCESSOR	RAM	DISK SPACE
Web browser	Pentium2 and higher models	128 MB (min)	512 MB
SERVER SIDE	**PROCESSOR**	**RAM**	**DISK SPACE**
Apache Tomcat Server	Any processor with 233MHz speed or higher	128MB(min)	30MB

MS Access server	Any processor with 233MHz speed or higher	128MB(min)	30MB

3.2 System Software Architecture

- **CLIENT ON INTERNET:**

  Web browser, Operating system(all Microsoft operating systems)

- **CLIENT ON INTRANET:**

  Client software, web browser, operating system (all Microsoft operating systems)

- **WEB SERVER:**

  Apache Tomcat Server, operating system (all Microsoft operating systems)

- **DATABASE SERVER:**

  MS Access 2007, operating system(all Microsoft operating systems)

- **DEVELOPMENT END:**

  JDK 1.6, operating system(windows), web server.

3.3 Internal Communications Architecture

- Client on internet will be using HTTP/HTTPS protocol.
- Client on intranet will be using TCP/IP protocol.

3.4 Database Management System Files

- Microsoft Access is the database used.
- One File is kept which contains one table
- Table in turn contains four fields.
- Three fields to store the first name, middle name and the last name of the algorithm
- The last field to store the encoded code which is the output of the algorithm applied

- Names in the database is encoded using the algorithm and saved in the field code.

First Name	Middle Name	Last Name	Code
Birt		**Wisle**	B632
Booth		**Davis**	**B312**

4.1 Algorithms

Soundex Algorithm,
Phonex Algorithm,
Metaphone Algorithm and
Daitch-Mokotooff Algorithm

4.1.1.SOUND-EX ALGORITHM :

DESCRIPTION :

1. Retain the first letter of the string
2. Remove all occurrences of the following letters, unless it is the first letter: a, e, h, i, o, u, w, y
3. Assign numbers to the remaining letters (after the first) as follows:

B F P V	coded as 1
C G J K Q S X Z	coded as 2
D T	coded as 3
L	coded as 4
M N	coded as 5
R	coded as 6

ADVANTAGES :

- Soundex recognises that vowels are some of the most variable aspects of surnames (for which it was originally used). This can also be extended to place names to see that Soundex may have similar abilities when dealing with a geographical corpus.

- Soundex recognises that certain consonants are similar sounding and groups them together. This means that if Soundex was used as an aid in searching, the user could still spell a surname, or place name for that matter, incorrectly, and as a result of Soundex, possibly still have the correctly spelt name/place suggested.
- Execution Time of sound-Ex algorithm is less.

LIMITATIONS :

- Soundex does not work for all cases. For example, given a name such as "Leighton", one could pronounce it as "Layton" or "Leeton". This means that the Soundex for Leighton is potentially ambiguous. Also, ideally, we would want either Leeton(L350) or Layton(L350) to have the same code as Leighton(L235). However, Soundex fails to capture this. At the same time, although Layton and Leeton *sound* different, Soundex gives them the same code. This is evidence of its extremely crude phonetic model.

- Inadequate at dealing with Spelling Errors

- SoundEx is an inherently unreliable interface. For this reason, SoundEx is only usable in applications that can tolerate high false positives (when words that don't match the sound of the inquiry are returned) and high false negatives (when words that match the sound of the inquiry are NOT returned).

4.1.2.METAPHONE ALGORITHM :

DESCRIPTION :

The 16 consonants: B X S K J T F H L M N P R 0 W Y
„th“ = 0
Exceptions:
Beginning of word: "ae-", "gn", "kn-", "pn-", "wr-" ----> drop first letter
"x" ----> change to "s"
"wh-" ----> change to "w"
Transformation ---->

B B, unless at the end of word after "m"
C X (sh), if "-cia-" or "-ch-"
S if "-ci-", "-ce-", or "-cy-"
SILENT if "-sci-", "-sce-", or "-scy-"
otherwise, including in "-sch-"
D J if in "-dge-", "-dgy-", or "-dgi-"
T otherwise
F F
SILENT if in "-gh-" and not at end or before a vowel
in "-gn" or "-gned"
G in "-dge-"
J if before "i", or "e", or "y" if not double "gg"
K otherwise
H SILENT if after vowel and no vowel follows
or after "-ch-", "-sh-", "-ph-", "-th-", "-gh-"
H otherwise
J J
K SILENT if after "c"
K otherwise
L L
M M
N N
P F if before "h"
P otherwise
Q K
R R
S X (sh) if before "h" or in "-sio-" or "-sia-"
S otherwise
T X (sh) if "-tia-" or "-tio-"
0 (th) if before "h"
silent if in "-tch-"
T otherwise
V F

W SILENT if not followed by a vowel
W if followed by a vowel
X KS
Y SILENT if not followed by a vowel
Y if followed by a vowel
Z S

ADVANTAGES :

- This algorithm is definitely better compared to SOUNDEX because it is more precise compared to the fixed 4 character code of SOUNDEX
- This algorithm produces variable length keys as its output, as opposed to Soundex's fixed-length keys. Similar sounding words share the same keys.

LIMITATIONS :

- Metaphone algorithm is not accurate and not precise.
- When it comes to systems where matching of names and identities or persons and companies, this may become a very critical matter. Companies may actually loose money when the system they rely on produces dab results.
- Execution time of this algorithm is very high.

4.1.3.PHONEX ALGORITHM :

DESCRIPTION :

The algorithm converts each word into an equivalent four-character code using the following steps:
1. Pre-process the name according to the following rules:
2. Remove all trailing 'S' characters at the end of the name.
3. Convert leading letter-pairs as follows:
KN -> N WR -> R
PH -> F
Convert leading single letters as follows:
H -> Remove
E, I, O, U, Y -> A K, Q -> C
P -> B J -> G
V -> F Z -> S
Code the pre-processed name according to the following rules:
4. Retain the first letter of the name, and drop all occurrences of A, E, H, I, O, U, W, Y in other positions.
5. Assign the following numbers to the remaining letters after the first:
B, F, P, V -> 1
C, G, J, K, Q, S, X, Z -> 2
D, T -> 3 If not followed by C.
L -> 4 If not followed by vowel or end of name.
M, N -> 5 Ignore next letter if either D or G.
R -> 6 If not followed by vowel or end of name.
6. Ignore the current letter if it has the same code digit as the last character of the code.
7. Convert to the form ‘letter, digit, digit, digit’ by adding trailing zeros (if there are less than three digits), or by dropping rightmost digits if there are more than three.

ADVANTAGES :

- This algorithm has less running time compared to SOUND-EX and METAPHONE algorithms.
- This algorithm produces variable length keys as its output.

4.1.4. DAITCH-MOKOTOFF ALGORITHM :

DESCRIPTION :

1. Translate first characters of name:
   MAC → MCC
   PH → FF
   KN → NN
   PF → FF
   K → C
   SCH → SSS
2. Translate last characters of name:
   EE → Y
   IE → Y
   DT, RT, RD, NT, ND → D
3. Translate remaining characters by following rules, incrementing by one character each time:
   EV → AF; else E, I, O, U → A
   Q → G
   Z → S
   M → N
   KN → N; else K → C
   SCH → SSS
   PH → FF
   H → previous character, if previous or next are nonvowel
   W → previous character, if previous is vowel
4. Check last character:
   if last character is S, remove it
   if last characters are AY, replace with Y
   if last character is A, remove it
5. Drop second letter of doubled letters.
6. Take first six characters.

ADVANTAGES :

- Coded names are six digits long, resulting in greater search precision (traditional Soundex uses four characters)
- coded names can be stored as numeric values, which can save space in me applications (regular Soundex encodes values as alphanumeric text)
- Several rules in the algorithm encode multiple character n-grams as single digits (American and Russell Soundex do not handle multi-character n-grams)
- Multiple possible encodings can be returned for a single name (traditional Soundex returns only one encoding, even if the spelling of a name could potentially have multiple pronunciations)

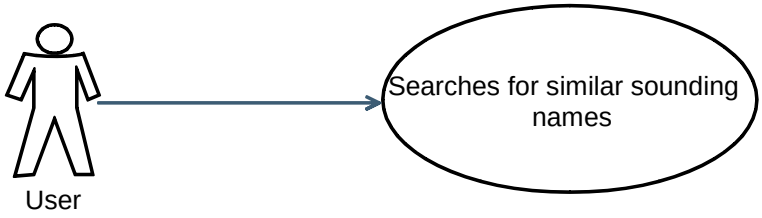
4 HUMAN-MACHINE INTERFACE

5.1. NAME OF USE CASE: USER

DESCRIPTION:

Initiates a search. Can search for similar sounding names. Any number of search can be initiated

WORKFLOW:



PRECONDITION:

- Output will be indexed only if information given is in alphabets (strictly).

NORMAL FLOW OF EVENTS:

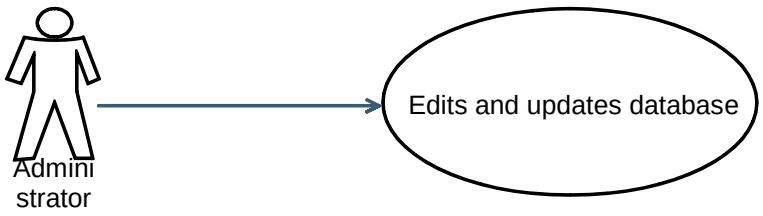
- User will type the information required for the search
- Query will be submitted
- Algorithms will be used to find the result
- Relevant output will be displayed.

5.2.NAME OF USE CASE: ADMINISTRATOR

DESCRIPTION:

Responsible for managing and updating the data required for the system.

WORKFLOW :

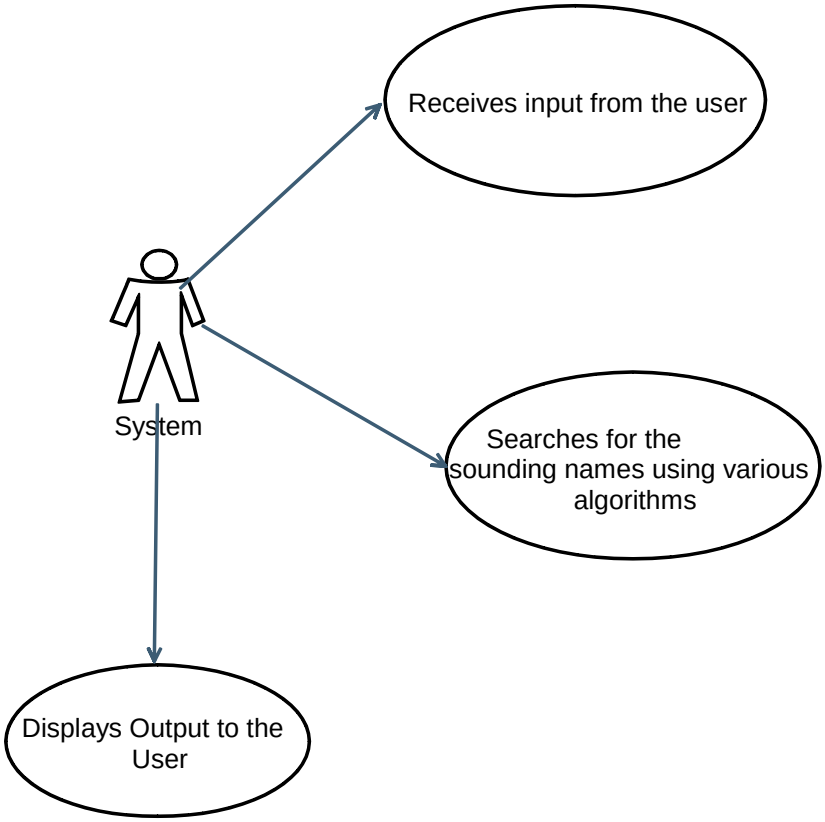


5.3.NAME OF USE CASE: SYSTEM

DESCRIPTION:

Will take the information given by the user as input and searches for the similar sounding names using algorithms given above and displays the result.

WORKFLOW:



PRECONDITION:

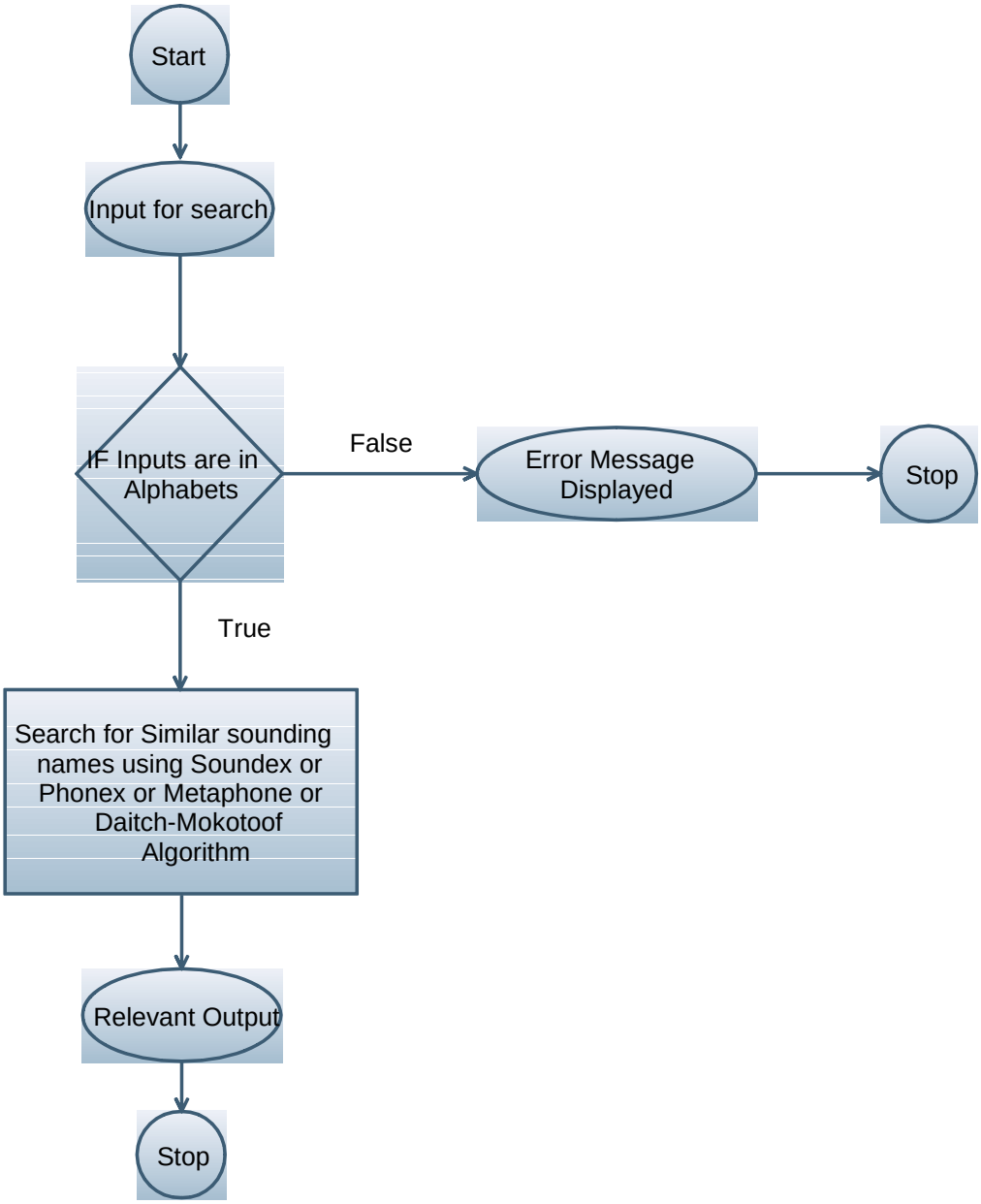
- Error message will be displayed when the input given by the user is in format other than alphabets.

NORMAL FLOW OF EVENTS:

- System will take input from the information provided by the user.
- Query will be processed using various algorithms.
- Relevant output will be indexed.

5 DETAILED DESIGN

6.1. BUSINESS FLOW DIAGRAM :

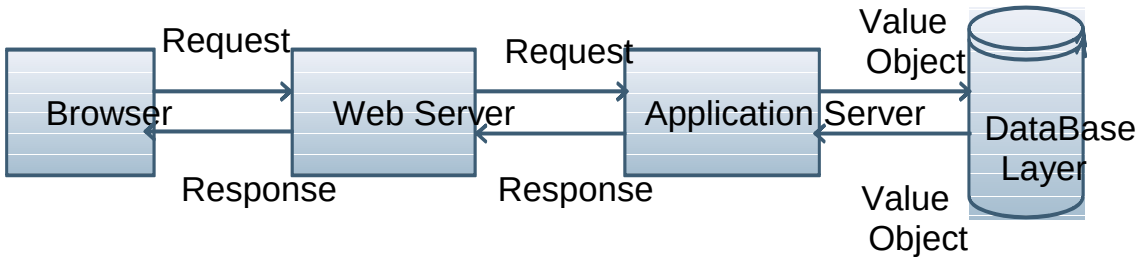


DESCRIPTION :

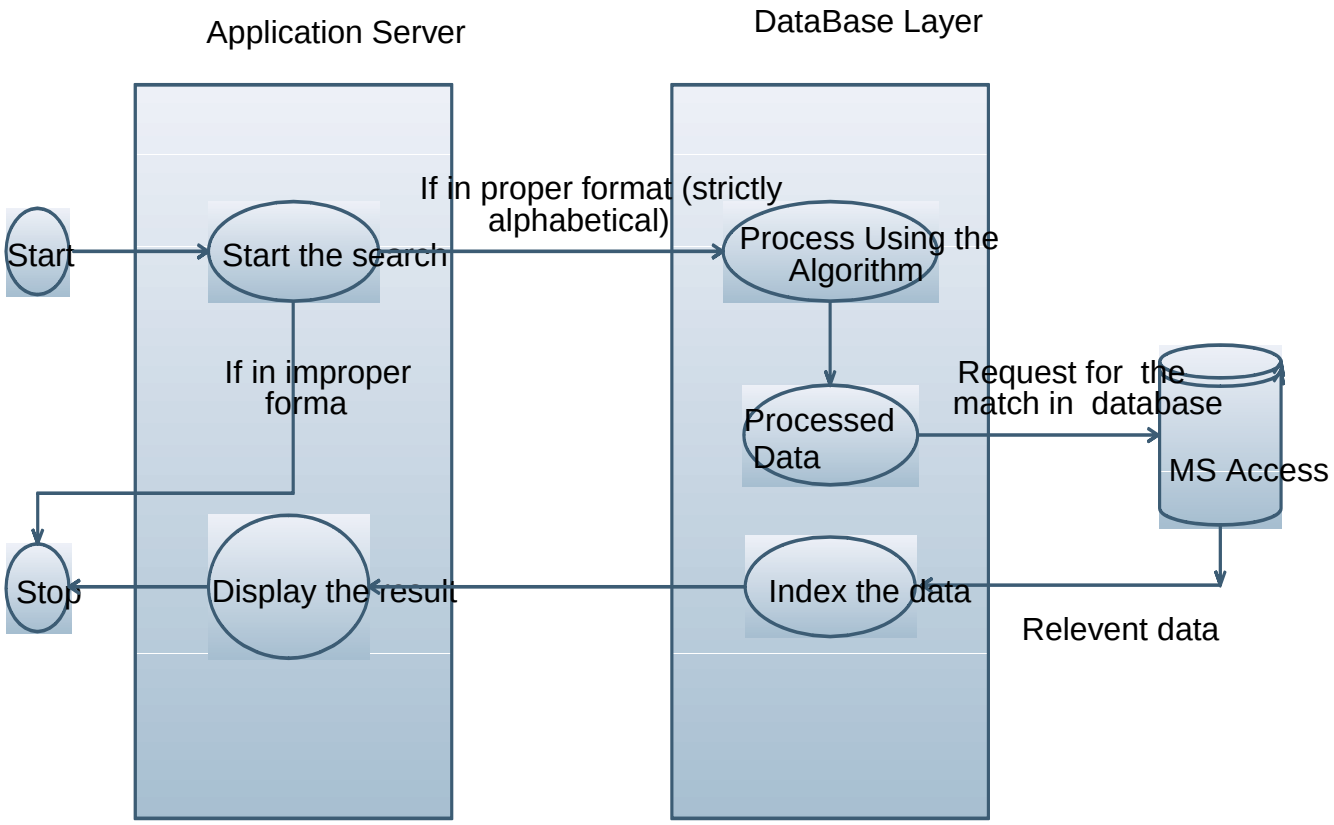
- User will type the information required for the search
- Query will be submitted

- If all the text are in alphabets, query will be processed else an error message will be displayed
- Soundex or Metaphone or Phonex or Daitch-Mokotoof algorithm will be used to process the search.
- Relevant output will be displayed.

6.2.ARCHITECTURE FLOW DIAGRAM :



6.3.DATAFLOW DIAGRAM :



7.Test Cases:

Application is tested in the following manner

- The application is executed many times (minimum of 500 times)
- System is checked for Exception Handling
    - Input is given as numbers and checked.
    - Input is given as alphanumeric data and checked.
    - No input data is given and the system is checked.
    - Both Uppercase and Lowercase data is given and the system is checked.
- All the above cases are tested for all the four algorithms.
- The system is expected to handle exceptions and give relevant output in all these cases.