

Contents

1. DDU Profile	2
1.1 Introduction:.....	2
1.2 Mission:.....	3
1.3 Achievements:.....	3
2. Project Profile	5
3. Planning	6
3.1 Project Definition:.....	6
3.2 Preliminary Investigation:	6
3.3 Activity Chart.....	7
4. System Analysis.....	8
4.1 Requirement Analysis:	8
4.2 Software Requirement Specification:	11
5. System Design	13
5.1 Architecture.....	13
5.1.1 Activity Diagram:	14
5.1.2 Use-Case Diagram:	15
5.1.3 Sequence Diagram:	16
5.2 Interfaces.....	17
6. Implementation	27
7. Testing	59
7.1 Test Plan.....	59
7.2 Testing Strategies.....	59
7.3 Test Cases, Test Data and Test Results.....	59
8. User Manuals	61
9. Future Expansion	65
10. Bibliography & References.....	66

1.DDU Profile

1.1 Introduction:

From an institute established in **1968 by Dharmsinh Desai**, then **Member of Parliament** and well respected educationist and social worker; to an **Autonomous status in 1991**, to a **Deemed University** with effect from **June 2000** and finally a **State university in April 2005**, all in a span of **forty-one years** has been possible due to a combine of many and varied factors, with discipline and teamwork being the undisputed leader. Up till now being known popularly as **DDU**, this institution has, down the years, grown from strength to strength. The institute was earlier affiliated to Gujarat University.

From a single filed specialization institute, to a multi faceted university offering eight fields of specialization both at the graduate and post graduate level, **DDU** has many firsts to its long list of achievements. From the academic year **2005-06**, a degree course in **Bachelor of Dental Surgery** has also started with intake capacity of 100.

Here, the entire academic process is certified for the registration of **ISO 9001:2000 by KPMG-Holland**. It is noteworthy that this is the only grant-in-aid institution in the state to receive **World Bank Assistance of Rs. 18.6 Crores**.

Besides formal education, the university also conducts the continuing education programmes not simply for the professionals but also for the under privileged sections of society. E.g. **Computer Awareness** for rural women of this region. The university organizes the **Blood donation camp** in every semester; it is appreciable that the highest numbers of blood bottles are contributed by the students to the Red Cross society every year in the region.

With firm resolve to move ahead, DDU has established an **R & D centre** taking up real life **IT based projects from industry and research organizations**. The teaching & examination work are governed as per the preplanned annual calendar: & because of which it becomes possible to hold the **Convocation Programme on 9th May** every year just after fifteen days to the final semester exam.

All the students undertake the full time Industry based project training in their final semester of the course which enables them for employment through campus interview even before course completion. The syllabus is reviewed and updated as per the latest trends in the field. Yoga - Meditation and Personality Development are taught as integral part of the curriculum. The academic environment is totally free from ragging & drugs inclusive of tobacco.

Spread over an area of 42 acres and in lush green surroundings, DDU is located just 50 kilometers from Ahmedabad and Vadodara and 500 kilometers from Mumbai. The University library has open-access system, computerized records, Photocopy facility, and a seating facility for about 350 readers at a time. Number of Indian periodicals is fifty, international ones are thirty one. The overall annual budget is Rs. 5 lacs against the books-purchase and the journal subscription.

With each of the departments, having their own computer facilities, DDU has over 650 computers, thus enabling each of its students to use the facility independently. All the laboratories are having the most modern and sophisticated instruments and equipments.

PhD facilities are available in the areas of Engineering /Technology & Management related fields. About 24 eminent experts from renowned institutions of academics / research serve as guide, under whom 44 students have got registered as research students.

A gymnasium, basketball court, tennis court and a vast playground offer the students excellent facility for sports.

1.2 Mission:

“The University shall undertake programs and projects for development of human resources, both through formal and non formal delivery systems, in areas of professional pursuits in all walks of human endeavors, with accent on relevance, value addition, societal needs and futuristic pilot projects.”

1.3 Achievements:

- The Government of Gujarat has given a grant of Rs. 3.5 crores to establish the Shah-Schulman Centre for Surface Science and Nanotechnology at DDU. March 2009.
- The Gujarat Council for Science and Technology (GUJCOST) has declared the Department of Chemical Engineering as a Centre of Excellence and given a grant of Rs. 28.5 lakhs for research on nanocatalysis – development and applications, to Prof. P.A. Joshi, Dean, Faculty of Technology, and Professor of Chemical Engineering, DDU.

Statistical Voice Calculator

- The Industries Commissionerate, Government of Gujarat has given a grant of Rs. 10-crore to DDU and naming it as Anchor Institute to provide manpower training programs in the Chemical & Petrochemical Sector for the state of Gujarat.

This is a four year project which has commenced from August 2009.

- The Department of Science and Technology (DST), New Delhi, has chosen Dr. Manish Mishra, of the Department of Chemistry and Chemical Engineering (and Shah-Schulman Centre for Surface Science and Nanotechnology) for the Young Scientist Fast-Track Scheme and will be funding his research (about Rs 21lakh) proposal on acid catalysts. April 2010.
- The GSFC Science Foundation has funded the project on "Nanotechnology to Clean Water in Developing Nations: Poor Man's Filter" submitted by Dr. Premal R. Shukla , Professor & Head, Department of Chemical Engineering (and Shah-Schulman Centre for Surface Science & Nanotechnology), DDU for a period of three years for Rs. 10.27 lakhs.
- Ministry of Human Resources Development, Government of India, has sanctioned Community Development Polytechnic to us with an outlay of Rs 71 lakh over a period of 5 years commencing from April 2010.
- University Grant Commission, New Delhi, has sanctioned M Tech (Chem Engg) program to Dept of Chem. Engg. with specialization in Surface Science & Nanotechnology with a total budget of Rs 41.5 lakh which will commence from July 2010.
- Department of Science & Technology, Government of India, has approved Innovation and Entrepreneurship Development center for the Faculty of Technology with a total budget outlay of Rs 45.3 lakh over four year period commencing from July 2010.
- The CSI student branch chapter of the Faculty of Technology has been awarded the best student chapter of the Region III for the year 2008-09
- In the Golden Jubilee Celebration of Government of Gujarat in Sep 2009, both the participating teams of the Computer Engg Dept., FoT were among 3 winners of the state, ; out of 40 teams only 8 could complete the assigned project in time and both the teams of Computer Engg Dept were among the winners
- In GATE 2010 exams, 24 Computer Engg. students have scored more than 90 percentile score., 9 students have scored more than 98 percentile and 5 students have scored more than 99 percentile.
- In the current batch of M.E. II, 15 papers have been selected for presentation and subsequent publication at International conference; 4 papers have been presented at national conference.
- VSAT enabled MeLT (Mobile e-Learning Terminals) Project promoted by National Mission on Education through Information and Communication Technology of Ministry of Human Resource Development Government of India with initial resources worth Rs 14 lakh have been awarded in July, 2010.

2.Project Profile

Project Name	:	“Statistical Voice Calculator”
Project Description	:	“Statistical Voice Calculator” is a desktop based application to find Mean, Median and Mode of a given data by help of Voice Commands. With this application user can find Mean, Median and Mode of Individual Data, Discrete Data and Grouped Data.
Development Team	:	<ul style="list-style-type: none">• Amin Shivang K.• Patel Dipen.• Patel Ketul A.
Internal Guide & Designation	:	Mr. J. J. Shah, Lecturer.
Platform	:	Windows
Technology Used	:	Java
Tools Used	:	NetBeans 9.6 Sphinx 4

3.Planning

3.1 Project Definition:

“Statistical Voice Calculator” is a Java based desktop application for performing calculation of Measures of Central Tendency like Mean, Median and Mode by accepting data in form of voice from the User of the System.

Using this application User can find Mean, Median and Mode of:

- Individual Data
- Discrete Data
- Grouped Data

For Individual Data, User need to specify the x_i values for calculation.

For Discrete Data, User need to specify the x_i and corresponding f_i values for calculation.

For Grouped Data, User needs to specify the lower and upper limits and the corresponding frequency value.

For getting the answer User needs to speak “Answer” to get the answer of the inputted values.

3.2 Preliminary Investigation:

Project Feasibility:

Feasibility Study is an evaluation or analysis of the impact of a proposed project.

In terms of Feasibility our system requires to pay attention to following feasibilities.

(1) Economic Feasibility:

Determines what all resources and tools we need to buy to complete the project. But the resources needed to complete our project are handy so economic wise our project is feasible.

(2) Operational Feasibility:

Operational feasibility is a measure of how well a proposed system solves the problems, taking advantage of the problems identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Statistical Voice Calculator

(3) Schedule Feasibility/Time Feasibility:

Estimating how long the project development will take, will the project will be completed before the specified deadline.

Project Development Strategy:

The Development strategy we use is “Iterative Development Strategy”. Iterative Development is a development approach that "cycles" through the development phases, from gathering requirements to delivering functionality in a working Environment.

3.3 Activity Chart

Activity Schedule

ID	Task Name	Start	Finish	Duration		Aug 2010					Sep 2010				Oct 2010	
					25/7	1/8	8/8	15/8	22/8	29/8	5/9	12/9	19/9	26/9	3/10	
1	Analysis	26-07-2010	14-08-2010	20d	<div></div>											
2	Design	15-08-2010	30-08-2010	16d	<div></div>											
3	Coding	01-09-2010	28-09-2010	28d	<div></div>											
4	Testing	29-09-2010	04-10-2010	6d	<div></div>											
5	Documentation	27-07-2010	07-10-2010	73d	<div></div>											

4. System Analysis

4.1 Requirement Analysis:

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements.

Mean: The mean is the arithmetic average of a set of values, or distribution.

Median: The median is a value that lies half way in a distribution.

Mode: The mode is a value that is repeated maximum number of times.

Following are the inputs, processes and outputs that help to us to understand the functioning of the system.

Mean, Median and Mode for Individual Data

Inputs:

1. Enter x_i values for calculation.

Processes:

1. Get count of number of x_i values, i.e. “n”.
2. Calculate $\sum x_i/n$ for **MEAN**.
3. For **MEDIAN**, arrange x_i values in either ascending or descending order.
4. If count of x_i values is Odd, then put value of n in $\left(\frac{n}{2} + 1\right)$ and get the corresponding $\left(\frac{n}{2} + 1\right)$ th observation.
5. If count of x_i values is Even, then put the value of n in $\left(\left(\frac{n}{2}\right) + \left(\frac{n}{2} + 1\right)\right)/2$ and get the corresponding term.
6. For **MODE**, get the value of x_i that is repeated maximum number of times in a given data that will be the mode of data.

Output:

1. Calculated values of MEAN, MEDIAN and MODE will be obtained.

Statistical Voice Calculator

Mean, Median and Mode for Discrete Data

Inputs:

1. Enter x_i values for calculation.
2. Enter Frequency f_i .

Processes:

1. Get Summation of number of f_i values, i.e. “N”.
2. Calculate $\sum_{i=1}^n f_i x_i / N$ for **MEAN**.
3. For **MEDIAN**, calculate cumulative frequency (cf) from the given frequency.
4. Put the value of N in $\frac{N+1}{2}$ and get the corresponding $\frac{N+1}{2}$ th observation.
5. For **MODE**, get the highest value of frequency and the value of x_i against that frequency is the Mode.

Output:

1. Calculated values of MEAN, MEDIAN and MODE will be obtained.

Mean, Median and Mode for Grouped Data

Inputs:

1. Get the Lower Limits from User.
2. Get the Upper Limits from User.
3. Get the associated frequencies.

Processes:

1. Calculate mid values (x_i) of the class using formula: $(\text{lower limit} + \text{upper limit})/2$.
2. Calculate $\sum_{i=1}^n \frac{f_i x_i}{N}$ for **MEAN**.
3. For **MEDIAN**, calculate cumulative frequency (cf) from the given frequency.
4. Calculate $\left(\frac{N}{2}\right)$, where $N = \sum f_i$

Statistical Voice Calculator

5. Match value of $N/2$ in Cumulative Frequency column to see in which class **MEDIAN** lies.

6. Calculate **MEDIAN** using formula: $L + \left(\frac{\frac{N}{2} - c.f}{f} \right) * c$

Where, L = Lower Limit of Median Class.

f = Frequency of Median Class.

Cf = Cumulative frequency of class above Median Class.

C = Class Interval.

7. For **MODE**, get the class with highest frequency. That will be the mode class.

8. Calculate **MODE** using formula,

$$L + \left(\frac{f1 - f0}{2f1 - f0 - f2} \right) * c$$

Where, L = Lower Limit of Mode Class.

$f1$ = Frequency of Mode Class.

$f0$ = Frequency of class above Mode Class.

$f2$ = Frequency of class below Mode Class.

c = Class Interval.

Output:

1. Calculated values of MEAN, MEDIAN and MODE will be obtained.

About Sphinx-4:

Sphinx-4 is a speech recognition system written entirely in the Java™ programming language. Sphinx-4 is an HMM-based speech recognizer. HMM stands for Hidden Markov Models, which is a type of statistical model. In HMM-based speech recognizers, each unit of sound (usually called a phoneme) is represented by a statistical model that represents the distribution of all the evidence (data) for that phoneme. This is called the acoustic model for that phoneme. When creating an acoustic model, the speech signals are first transformed into a sequence of vectors that represent certain characteristics of the signal, and the parameters of the acoustic model are then estimated using these vectors (usually called features). This process is called training the acoustic models.

During speech recognition, features are derived from the incoming speech in the same way as in the training process. The component of the recognizer that generates these features is called the front end. These live features are scored against the acoustic model. The score obtained indicates

how likely that a particular set of features (extracted from live audio) belongs to the phoneme of the corresponding acoustic model.

The process of speech recognition is to find the best possible sequence of words (or units) that will fit the given input speech. It is a search problem, and in the case of HMM-based recognizers, a graph search problem. The graph represents all possible sequences of phonemes in the entire language of the task under consideration. The graph is typically composed of the HMMs of sound units concatenated in a guided manner, as specified by the grammar of the task.

The input speech signal is transformed into a sequence of feature vectors. After the last feature vector is decoded, we look at all the paths that have reached the final exit node. The path with the highest score is the best fit, and a result taking all the words of that path is returned.

Requirements for running Sphinx-4:

- Win 32 System
- Java SE 6 Development Kit or better

4.2 Software Requirement Specification:

Scope:

The fields where statistics can be applied are as follows:

- Statistical Surveys.
- Business Statistics.
- Engineering.

Features:

Facilities that are offered by a “Statistical Voice Calculator” are as follows:

- Calculation of Mean, Median and Mode by Voice issued Commands.
- Calculation of Mean, Median and Mode of Individual data.
- Calculation of Mean, Median and Mode of Discrete data.
- Calculation of Mean, Median and Mode of Grouped data.

Hardware Requirements:

System should have following hardware for satisfying the performance constraints. They are:

- Intel Pentium Dual Core Processor or Higher.
- Inbuilt Sound Card.
- Ram 1 GB or higher.
- Inbuilt Microphone.
- Memory Space of 1 GB.

Software Requirements:

System should have following software handy for satisfying the performance constraints. They are:

- Windows Vista/ Windows 7
- Java Development Kit

Other Requirements:

- Microsoft Visio 2003 (for system designing)
- Microsoft Word 2007 (for documentation)

5. System Design

System Design refers to the process of planning of new Information System or one to replace or complement an Existing system.

5.1 Architecture

Architecture of System design includes:

- Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

In our system, First User needs to specify, the calculation method (Individual, Discrete or Grouped) by voice command. After that user has to specify the values that are required for the particular method and get the answer.

- Use-Case Diagram:

A **use case diagram** in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

In our system Use cases are:

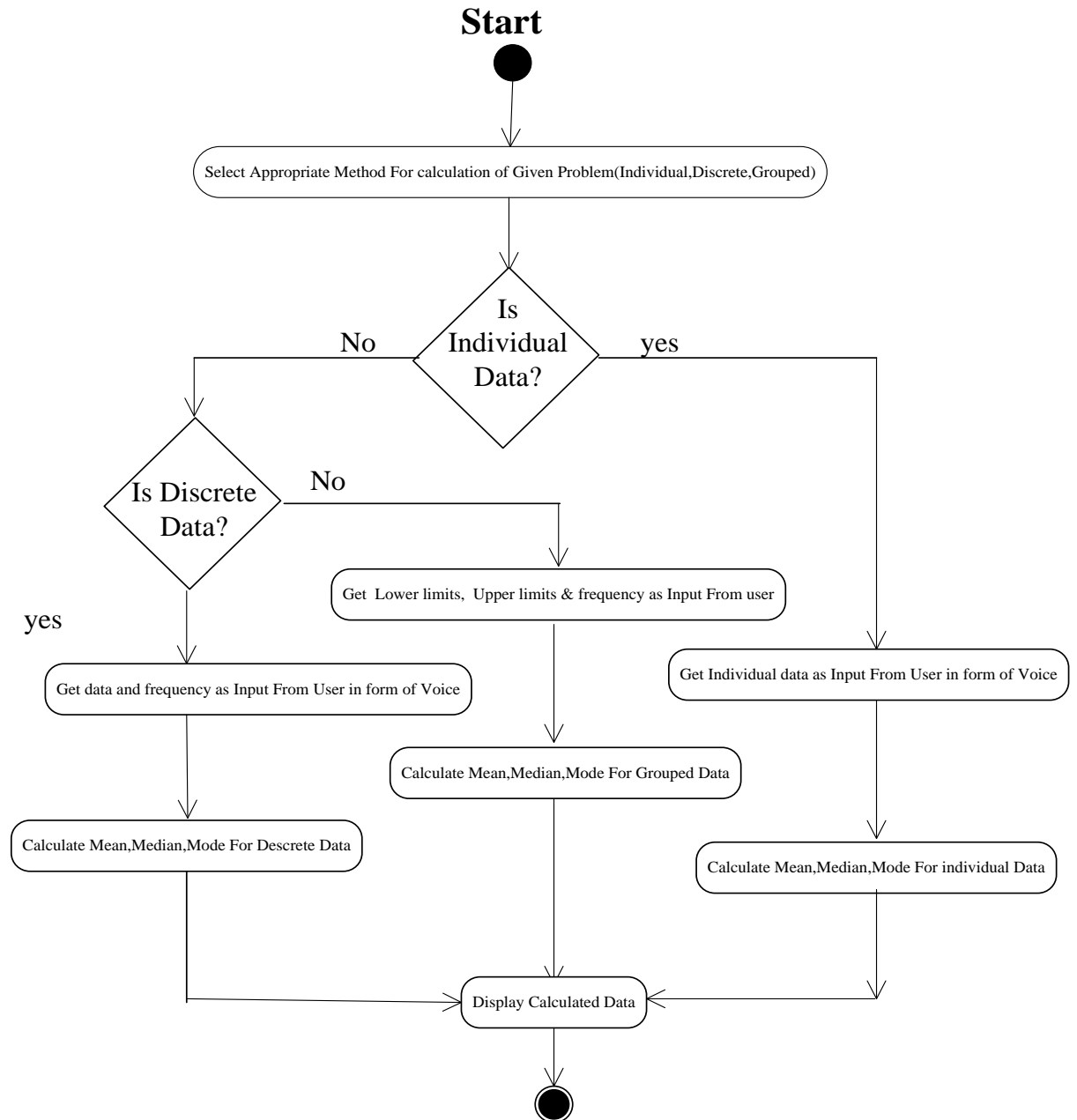
- Selection of appropriate method for calculation and give input.
- Convert text to number.
- Calculation.
- Display the answer.

In our system, User is a Actor who will interact with the system.

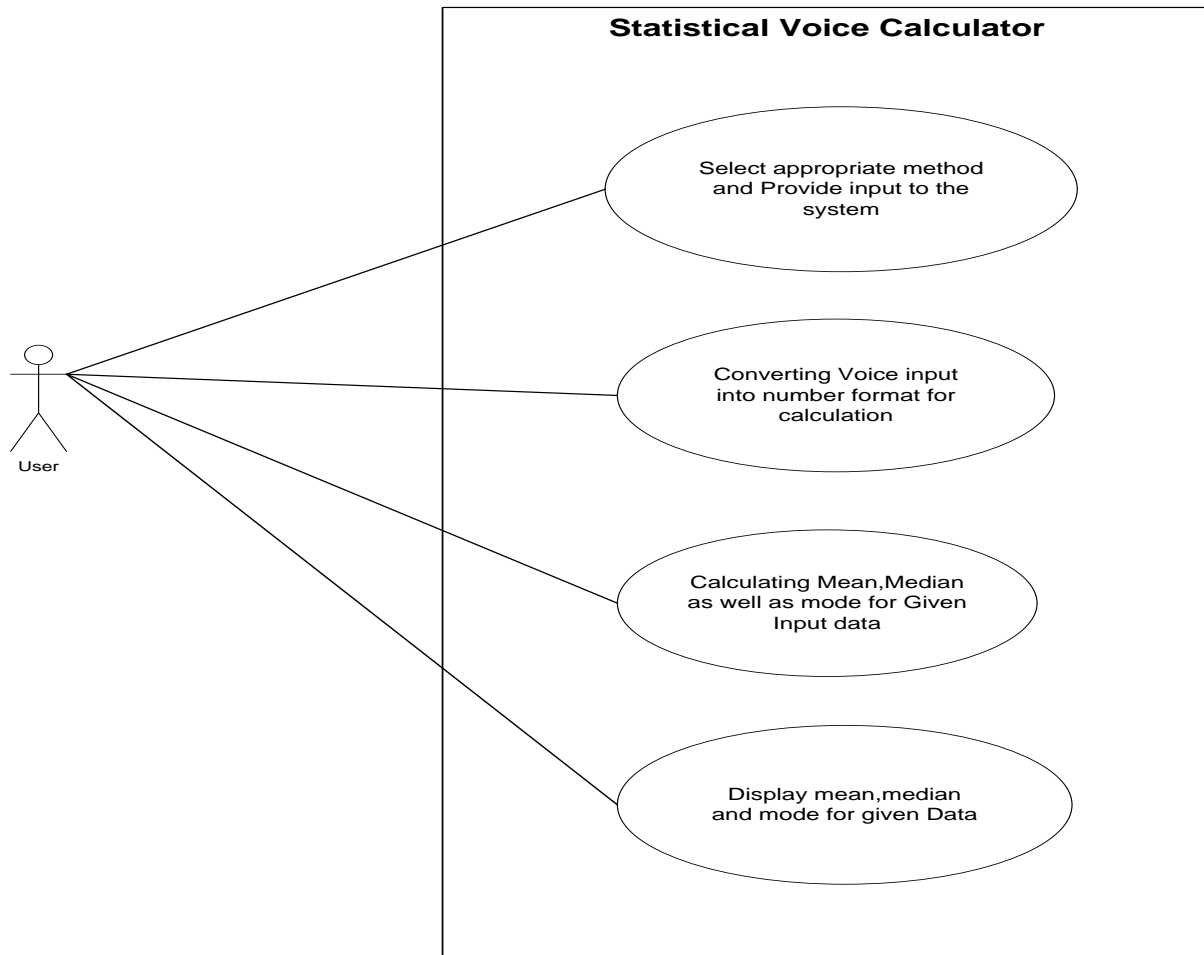
- Sequence Diagram:

A **sequence diagram** in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.

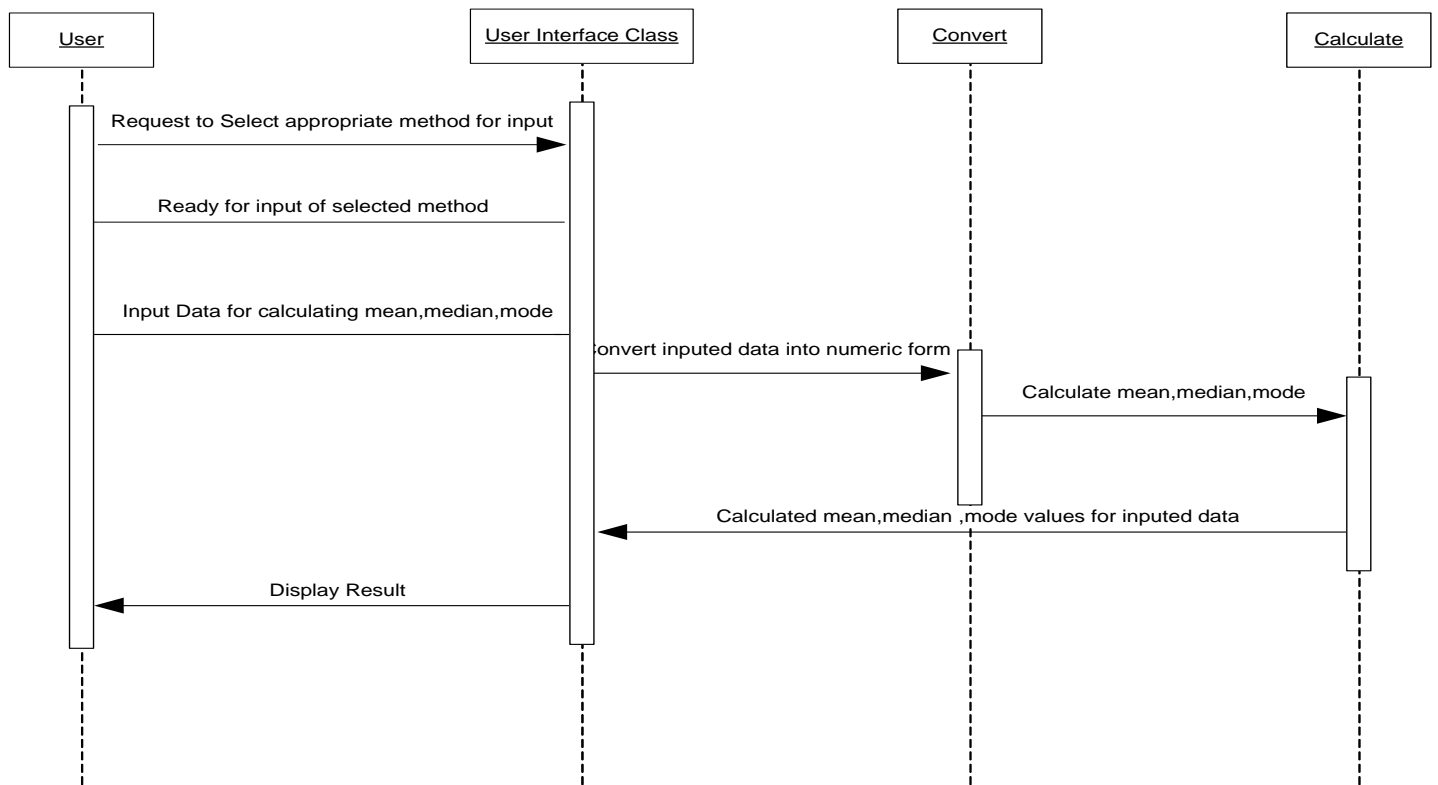
5.1.1 Activity Diagram:



5.1.2 Use-Case Diagram:

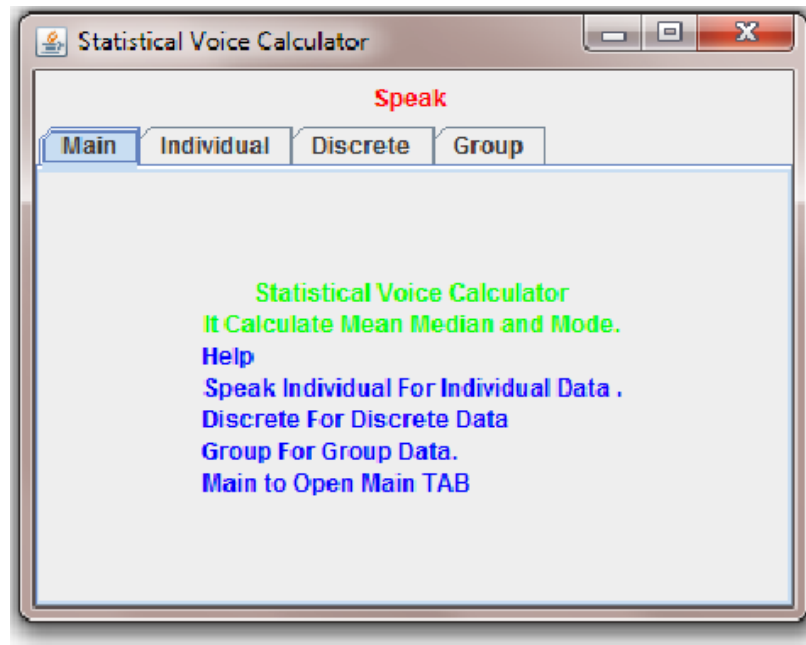


5.1.3 Sequence Diagram:



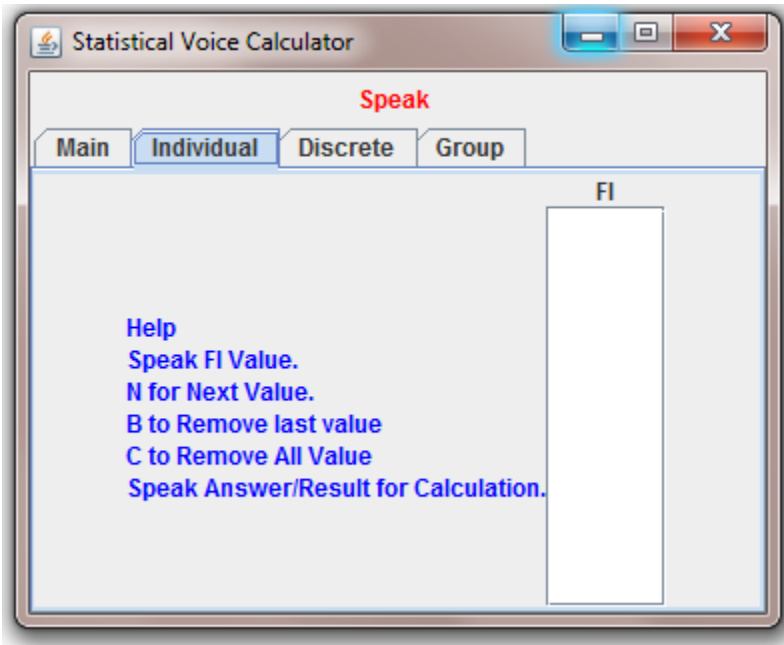
5.2 Interfaces

1. Home Screen:



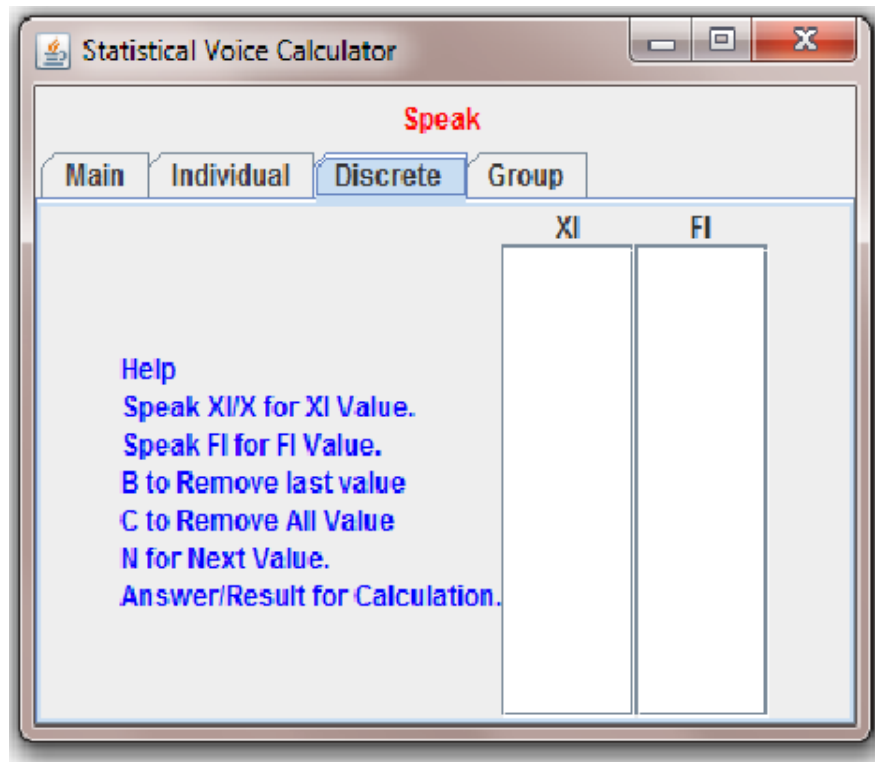
- This is the home page of the “Statistical Voice Calculator”, The home page is the page from where you can navigate to “Individual” tab for calculation of Mean, Median and Mode of “Individual Data”, “Discrete” tab for calculation of Mean, Median and Mode of “Discrete Data”, ”Group” tab for the calculation of Mean, Median and Mode of “Group Data”.
- To move to “Individual Tab”, user need to speak “Individual”, to move to “Discrete Tab”, user need to speak “Discrete” and to move to “Group Tab”, User need to speak “Group”.
- To move back to “Main Tab”, from any of the other tabs, User need to speak “Main”.

2. Individual Tab Screen:



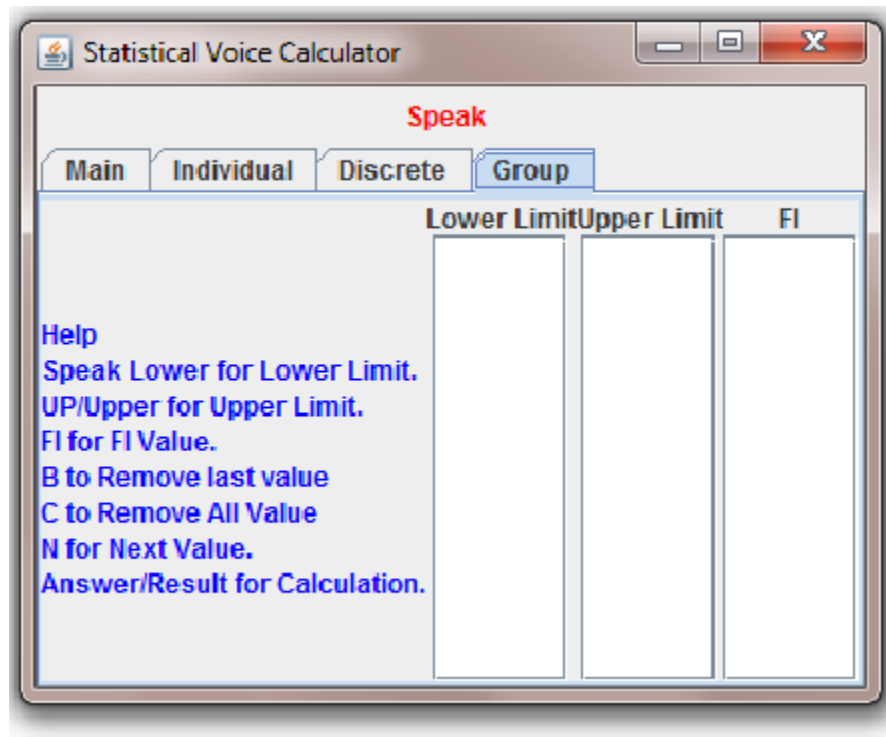
- After User speaks "Individual", The Control is shifted to this "Individual Tab".
- Here for calculation, User need to specify the " x_i " values for calculation by voice.
- All values will be specified in the list box. User will get the "Answer" after he has inputted the required values and speaks "Result/answer" for getting Answer.

3. Discrete Tab Screen:



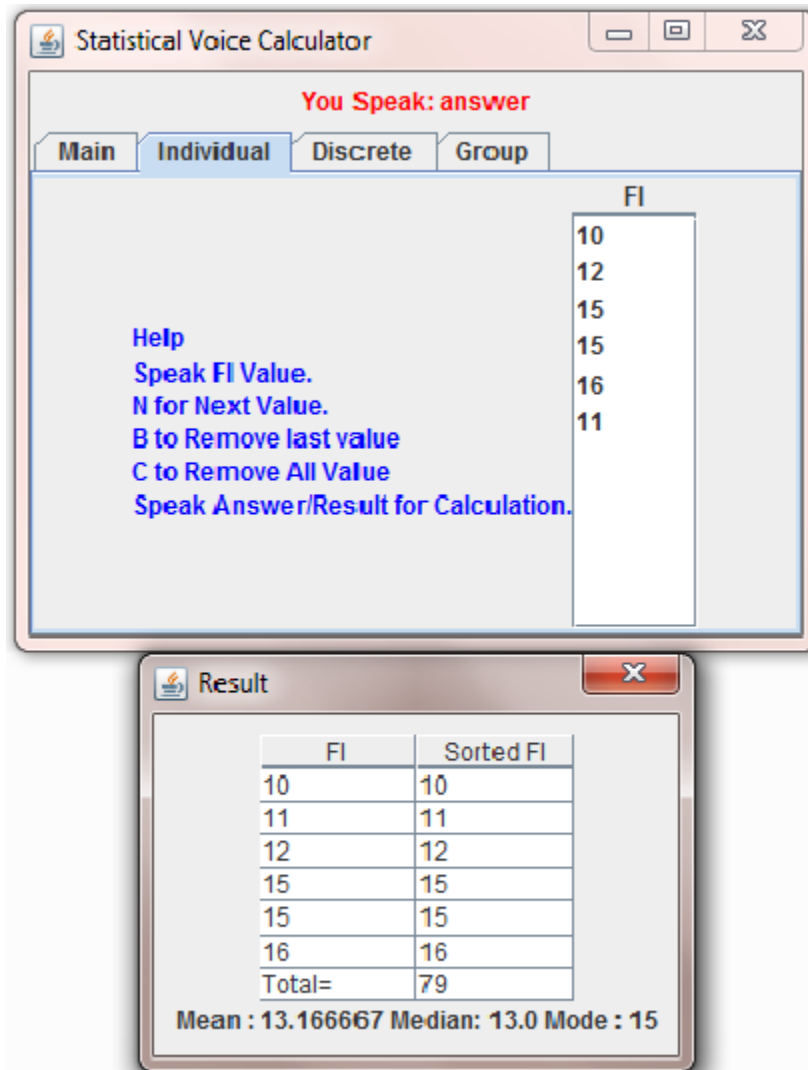
- After User speaks "Discrete", the control is shifted to "Discrete Tab".
- Here for calculation user need to specify " x_i " and " f_i " values for calculation by voice commands.
- All values will be specified in the list box. User will get the "Answer" after he has inputted the required values and speaks "Result/answer" for getting Answer.

4. Group Tab Screen:



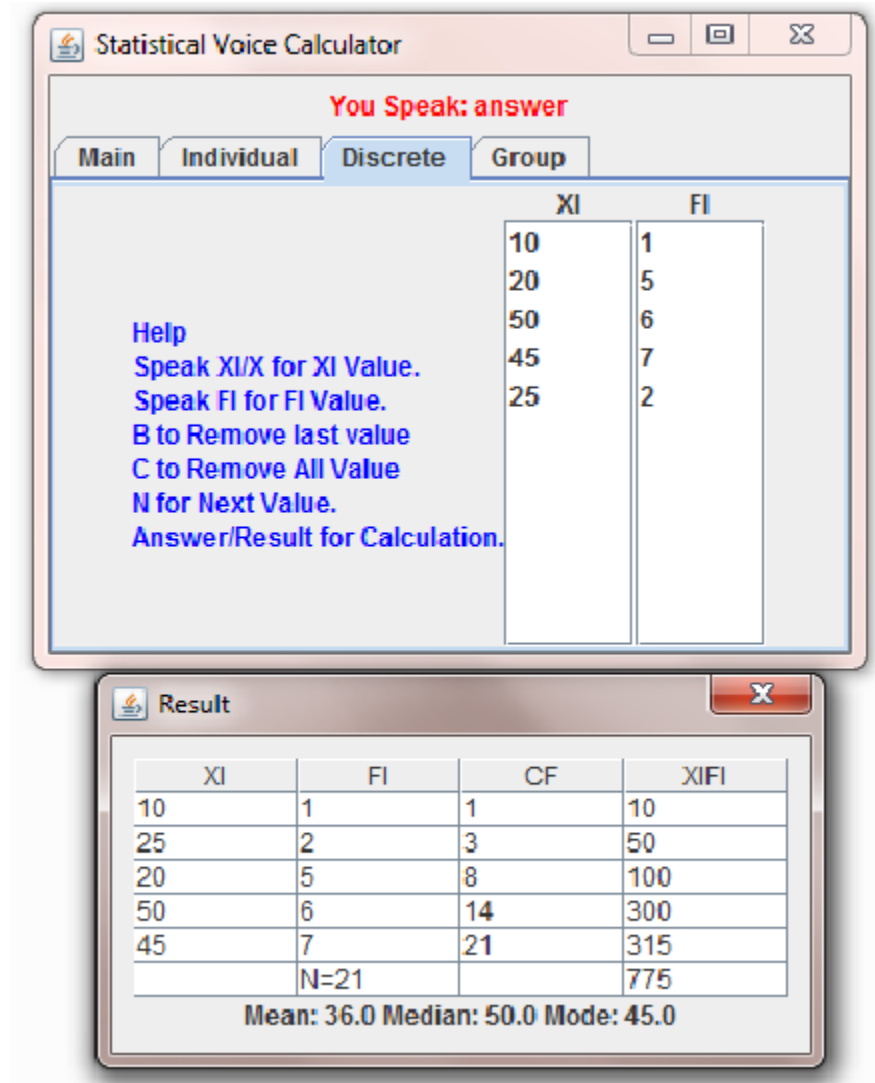
- After User speaks "Group", the control is shifted to "Group Tab".
- Here User need to "Lower Limits", "Upper Limits" and "Frequency values" by voice.
- All values will be entered in specified list boxes. User will get the "Answer" after he has inputted the required values and speaks "Result/answer" for getting Answer.

5. Calculation for Individual data



This tab is for calculation of Individual Data, Here user enters certain data and gets the answer in above format.

6. Calculation of Discrete Data



This tab is for calculation of Discrete Data, Here user enters certain data and gets the answer in above format.

7. Calculation of Grouped Data

Statistical Voice Calculator

You Speak: answer

Main
Individual
Discrete
Group

Lower Limit

Upper Limit

FI

0	10	11
10	20	12
20	30	14
30	40	13

Help

Speak Lower for Lower Limit.

UP/Upper for Upper Limit.

FI for FI Value.

B to Remove last value

C to Remove All Value

N for Next Value.

Answer/Result for Calculation.

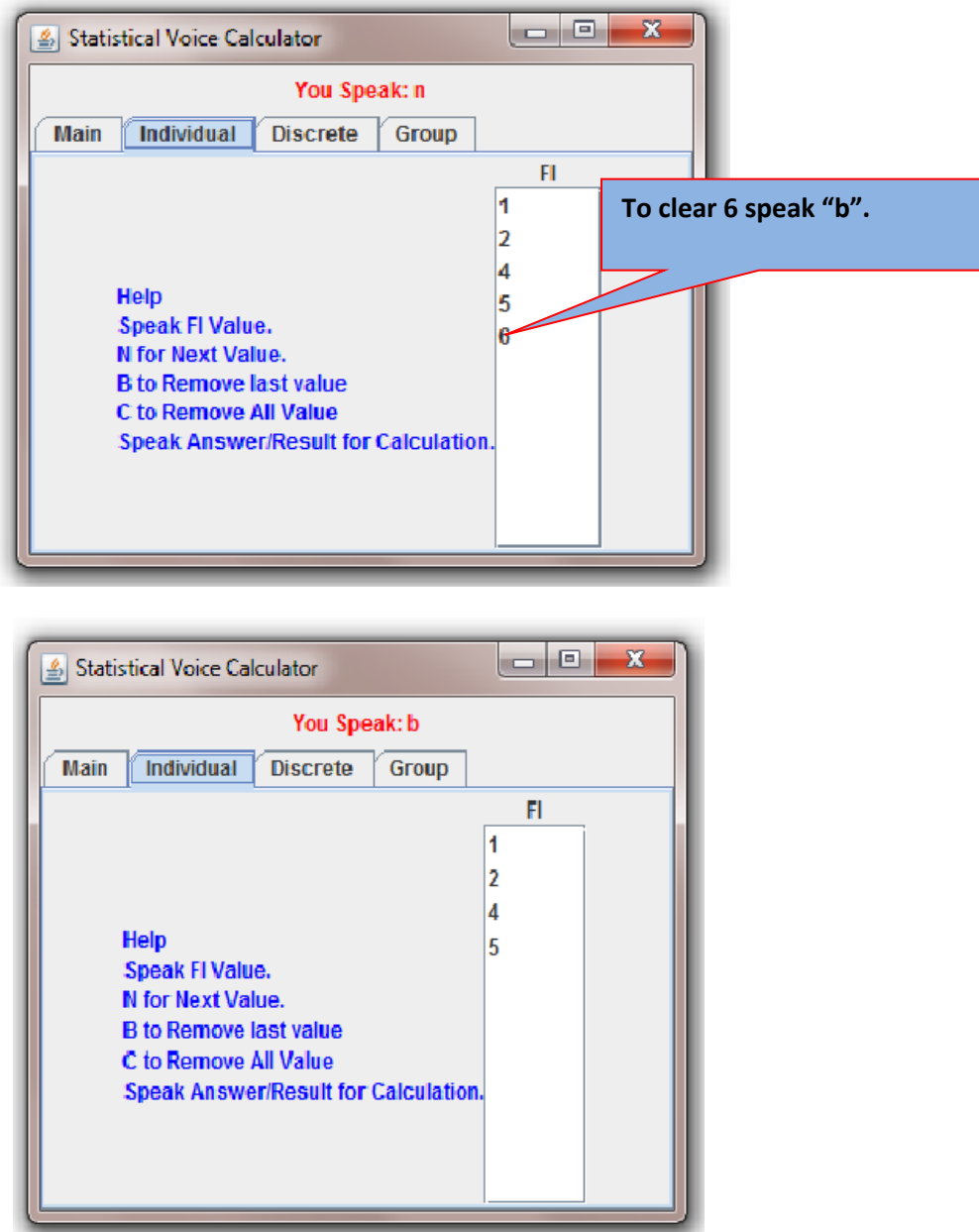
Result

Class	FI	Mid(XI)	XIFI	CF
0.0-10.0	11	5.0	55.0	11
10.0-20.0	12	15.0	180.0	23
20.0-30.0	14	25.0	350.0	37
30.0-40.0	13	35.0	455.0	50
	N=50		1040.0	

Mean : 20.8 Median : 21.428572 Mode : 26.666668

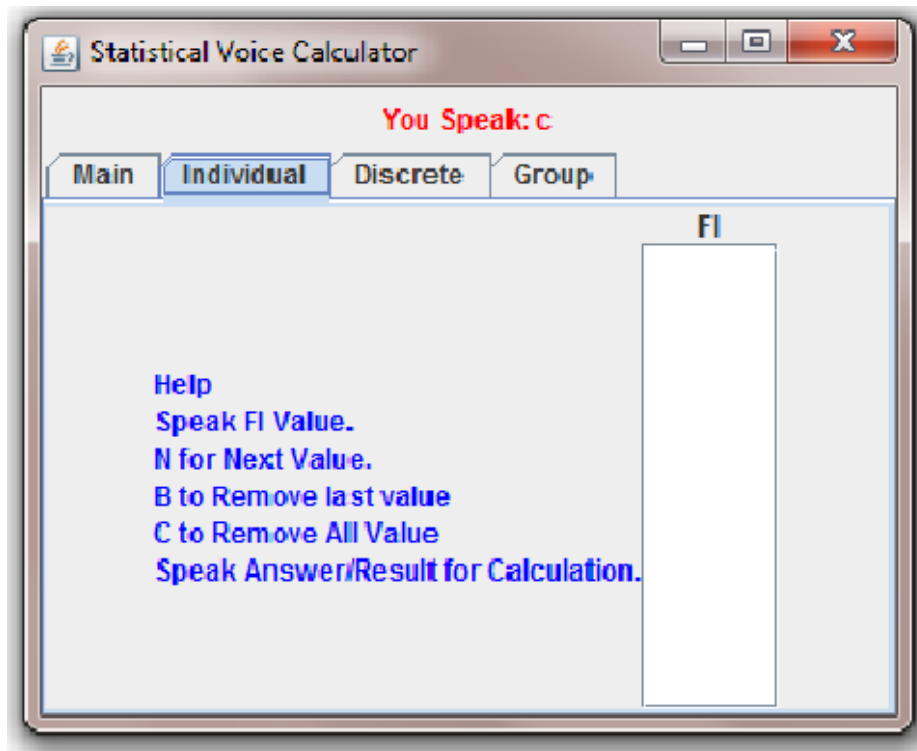
This tab is for calculation of Grouped Data, Here user enters certain data and gets the answer in above format. User need to specify lower and upper limits and frequencies of associated class, and hence after user speaks answer displays the following result format.

8. Correcting the input value



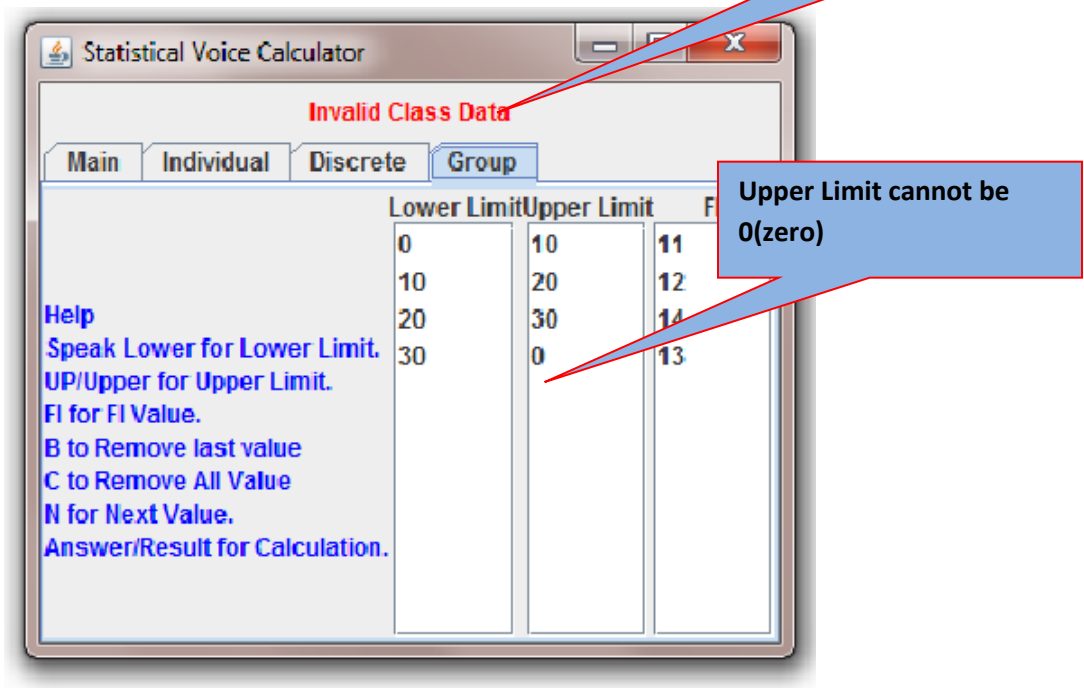
It may happen that User specifies a wrong value, he/she can correct the last input value by speaking “b”. The last spoken value will be erased. The illustration above shows the use of “b” voice command. In upper screen shot, User has specified “6”, then he has removed “6” by “b” voice command (see the lower screen shot).

9. Clear all values

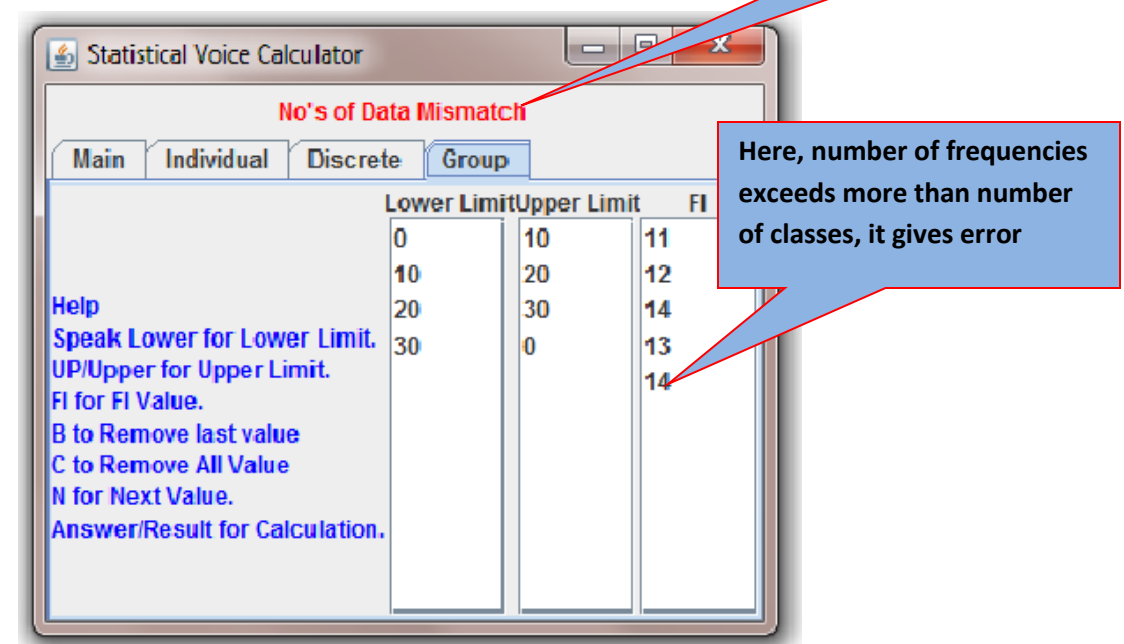


User can clear all Fields by speaking “C” voice command.

10. Invalid Values



11. Data mismatch



6. Implementation

1. GUI.java

The entire GUI required in the application is set in gui.java file.

```
import javax.swing.*;
import javax.swing.JOptionPane;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.util.Vector;
import java.awt.Color;
import java.awt.GridBagLayout;
import java.awt.GridBagConstraints;
import java.awt.Dimension;
import java.util.Enumuration;
import javax.swing.JLabel;
import javax.swing.JPanel;
import edu.cmu.sphinx.frontend.util.Microphone;
import edu.cmu.sphinx.recognizer.Recognizer;
import edu.cmu.sphinx.result.Result;
import edu.cmu.sphinx.util.props.ConfigurationManager;
class GUI extends JFrame
{
    JButton start,stop;
    Vector data;
    JLabel msg,result;
    JLabel gdata_label;
    VoiceListener vl;
    MainVoiceListener mvl;
    ConfigurationManager cm;
    Recognizer recognizer;
    JTabbedPane tabbedPane;
    DefaultListModel listModel;
    JList list,dis_xi,dis_fi,group_ll,group_fi,group_ul;
    boolean st=false;
    VoiceListenerDis vld;
    VoiceListenerGroup vlg;

    GUI f;
    public GUI()
    {
```

Statistical Voice Calculator

```
super("Statistical Voice Calculator");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

setLocationRelativeTo(null);
initComponent();
createPane();

f=this;

pack();
setVisible(true);
callVoiceListener();

/*cm = new
ConfigurationManager(HelloWorld.class.getResource("main.config.xml"));
recognizer = (Recognizer) cm.lookup("recognizer");

recognizer.allocate();
// start the microphone or exit if the programmm if this is not
possible
Microphone microphone = (Microphone)
cm.lookup("microphone");
if (!microphone.startRecording()) {
System.out.println("Cannot start microphone.");

recognizer.deallocate();
JOptionPane.showMessageDialog(null,"Cannot start
microphone");

}
mvl=new MainVoiceListener(f,recognizer);
*/

}
void changeData(int i)
{

try{
System.out.println("change TAB");
tabbedPane.setSelectedIndex(i);
if(i==0)
mvl=new MainVoiceListener(f,recognizer,listModel);
if(i==2)
vld=new
VoiceListenerDis(f,recognizer,(DefaultListModel)dis_xi.getModel(),(DefaultListModel)dis_fi.g
etModel());
if(i==1)
vl=new VoiceListener(f,recognizer,listModel);
if(i==3)
```

Statistical Voice Calculator

```
        vlg=new
VoiceListenerGroup(f,recognizer,(DefaultListModel)group_ll.getModel(),(DefaultListModel)gro
up_ul.getModel(),(DefaultListModel)group_fi.getModel());

        }catch(Exception e){System.out.println("change..." +e);}
    }
    void initvoiceComponent()
    {

        cm = new
ConfigurationManager(HelloWorld.class.getResource("helloworld.config.xml"));

        recognizer = (Recognizer) cm.lookup("recognizer");

        recognizer.allocate();
        // start the microphone or exit if the programm if this is not
possible
        Microphone microphone = (Microphone)
cm.lookup("microphone");
        if (!microphone.startRecording()) {
            System.out.println("Cannot start microphone.");

            recognizer.deallocate();
            JOptionPane.showMessageDialog(null,"Cannot start
microphone");

            return;
        }
        st=true;

    }
    void createPane()
    {
        tabbedPane = new JTabbedPane();
        JPanel panel0 = makeMainPanel();
        tabbedPane.addTab("Main",panel0);
        JPanel panel1 = makeIndPanel();
        tabbedPane.addTab("Individual",panel1);
        JPanel panel2 = makeDisPanel() ;
        tabbedPane.addTab("Discrete",panel2);
        JPanel panel3 = makeGroupPanel() ;
        tabbedPane.addTab("Group",panel3);
        add(tabbedPane,BorderLayout.CENTER);
    }
    JPanel makeMainPanel()
    {
        JPanel panel = new JPanel(new GridBagLayout());
```

Statistical Voice Calculator

```
GridBagConstraints c = new GridBagConstraints();
JLabel main=new JLabel("Statistical Voice Calculator");
main.setForeground(Color.green);
JLabel main1=new JLabel("It Calculate Mean Median and
Mode.");
main1.setForeground(Color.green);
JLabel data_label = new JLabel("<html>Help<br>Speak
Individual For Individual Data .<br>Discrete For Discrete Data<br>" +
" Group For Group Data.<br>Main to Open Main TAB"
+"</html>");
data_label.setForeground(Color.blue);

c.gridx = 0;
c.gridy = 0;
panel.add(main,c);

c.gridx = 0;
c.gridy = 1;
panel.add(main1,c);

c.gridx = 0;
c.gridy = 2;
panel.add(data_label,c);

return panel;
}
void setgdata_label(String s)
{
    gdata_label.setText(s);
}
JPanel makeGroupPanel()
{
    JPanel panel = new JPanel(new GridBagLayout());

    GridBagConstraints c = new GridBagConstraints();

    gdata_label = new JLabel("<html>Help <br>Speak Lower for
Lower Limit.<br>UP/Upper for Upper Limit.<br>FI for FI Value.<br>"
+"B to Remove last value<br>C to Remove All Value<br>N for
Next Value.<br> Answer/Result for Calculation."
+"</html>");
    gdata_label.setForeground(Color.blue);
    group_ul = new JList(new DefaultListModel());
    group_ll = new JList(new DefaultListModel());
    group_fi = new JList(new DefaultListModel());
```

Statistical Voice Calculator

```
        JScrollPane listul = new JScrollPane(group_ul);
        JScrollPane listll = new JScrollPane(group_ll);
        JScrollPane listfi = new JScrollPane(group_fi);
        listul.setPreferredSize(new Dimension(60, 200));
        listll.setPreferredSize(new Dimension(60, 200));
        listfi.setPreferredSize(new Dimension(60, 200));
        c.gridx = 1;
c.gridy = 0;
        panel.add(new JLabel("Lower Limit"),c);
        c.gridx = 2;
c.gridy = 0;
        panel.add(new JLabel("Upper Limit"),c);
        c.gridx = 3;
c.gridy = 0;
        panel.add(new JLabel("FI"),c);
        c.gridx = 0;
c.gridy = 1;
        panel.add(gdata_label,c);
        c.gridx = 1;
c.gridy = 1;
        panel.add(listll,c);
        c.gridx = 2;
c.gridy = 1;
        panel.add(listul,c);
        c.gridx = 3;
c.gridy = 1;
        panel.add(listfi,c);
        return panel;
    }
    JPanel makeDisPanel()
    {
        JPanel panel = new JPanel(new GridBagLayout());

        GridBagConstraints c = new GridBagConstraints();
        JLabel data_label = new JLabel("<html>Help <br>Speak XI/X for
XI Value.<br> Speak FI for FI Value.<br>"
        +"B to Remove last value<br>C to Remove All Value<br>N for
Next Value.<br> Answer/Result for Calculation."
        +"</html>");
        data_label.setForeground(Color.blue);
        dis_xi = new JList(new DefaultListModel());
        dis_fi = new JList(new DefaultListModel());
        JScrollPane listxi = new JScrollPane(dis_xi);
        JScrollPane listfi = new JScrollPane(dis_fi);
        listxi.setPreferredSize(new Dimension(60, 200));
        listfi.setPreferredSize(new Dimension(60, 200));
```

Statistical Voice Calculator

```
c.gridx = 1;
c.gridy = 0;
panel.add(new JLabel("XI"),c);
c.gridx = 2;
c.gridy = 0;
panel.add(new JLabel("FI"),c);
c.gridx = 0;
c.gridy = 1;
panel.add(data_label,c);
c.gridx = 1;
c.gridy = 1;
panel.add(listxi,c);
c.gridx = 2;
c.gridy = 1;
panel.add(listfi,c);
return panel;
}
JPanel makeIndPanel()
{
    JPanel panel = new JPanel(new GridBagLayout());

    GridBagConstraints c = new GridBagConstraints();
    JLabel data_label = new JLabel("<html>Help<br>Speak FI
Value.<br>N for Next Value.<br>B to Remove last value<br>"+
"C to Remove All Value<br>Speak Answer/Result for
Calculation."
    + "</html>");
    data_label.setForeground(Color.blue);
    listModel = new DefaultListModel();
    list = new JList(listModel);
    JScrollPane listScrollPane = new JScrollPane(list);
    listScrollPane.setPreferredSize(new Dimension(60, 200));
    c.gridx = 1;
c.gridy = 0;
panel.add(new JLabel("FI"),c);
c.gridx = 0;
c.gridy = 1;
panel.add(data_label,c);
c.gridx = 1;
c.gridy = 1;
panel.add(listScrollPane,c);
return panel;
}
void callVoiceListener()
{
```


Statistical Voice Calculator

```
initvoiceComponent();

    if(st){
        mvl=new MainVoiceListener(f,recognizer,listModel);
        msg.setText("Speak ");
    }
}
void initComponents()
{
    JPanel panel=new JPanel();

    msg=new JLabel("Wait Loading audio devices .....");
    msg.setForeground(Color.red);
    result=new JLabel("Mean:  Median:  Mode: ");
    panel.add(msg);
    add(panel,BorderLayout.PAGE_START);
    // add(result,BorderLayout.PAGE_END);

}
void setMessage(String s)
{
    msg.setText(s);
}
void setResult(String s)
{
    result.setText(s);
}

public static void main(String args[])
{
    GUI g=new GUI();

}
}
```

2. MainVoiceListener.java

This code is for moving to the Individual, Discrete or Group tabs.

```
import edu.cmu.sphinx.frontend.util.Microphone;
import edu.cmu.sphinx.recognizer.Recognizer;
import edu.cmu.sphinx.result.Result;
import edu.cmu.sphinx.util.props.ConfigurationManager;
import javax.swing.JFrame;
import javax.swing.DefaultListModel;
import java.util.TreeSet;
class MainVoiceListener implements Runnable
{
    Thread t;
    boolean done=true;
    GUI f;
    Recognizer recognizer;

    DefaultListModel list;

    //constructor
    MainVoiceListener(GUI f,Recognizer recognizer,DefaultListModel list)
    {
        t=new Thread(this,"Listener");
        this.f=f;
        this.recognizer=recognizer;
        this.list=list;
        t.start();
    }
    public void stop()
    {
        done=false;
    }
    public void run() {
        try{

            Result result=null;
            //Recognizer.State rs=recognizer.getState();

            while(recognizer.getState()!=Recognizer.State.READY){
                //System.out.println(rs.toString());
                Thread.sleep(500);
            }

            f.setMessage("Speak ");
            while (done)
```

Statistical Voice Calculator

```
{

result= recognizer.recognize();

if (result != null ) {
    String resultText = result.getBestFinalResultNoFiller();
    System.out.print(resultText );
    f.setMessage("You Speak: "+resultText);
    if(resultText.equals("discrete"))
    {
        f.changeData(2);f.setMessage("Wait.....");
        return;
    }
    if(resultText.equals("group"))
    {
        f.changeData(3);f.setMessage("Wait.....");
        return;
    }
    if(resultText.equals("individual"))
    {
        f.changeData(1);f.setMessage("Wait.....");
        return;
    }
} else {
    System.out.println("I can't hear what you said.\n");
}
}

} catch(Exception e){System.out.println(e.toString());}
} //run End
}
```

3. VoiceListener.java

This code is for calculation of individual data.

```
import edu.cmu.sphinx.frontend.util.Microphone;
import edu.cmu.sphinx.recognizer.Recognizer;
import edu.cmu.sphinx.result.Result;
import edu.cmu.sphinx.util.props.ConfigurationManager;
import javax.swing.JFrame;
import javax.swing.DefaultListModel;
import java.util.TreeSet;
class VoiceListener implements Runnable
{
    Thread t;
    boolean done=true;
    GUI f;
    Recognizer recognizer;
    DefaultListModel list;

    //constructor
    VoiceListener(GUI f,Recognizer recognizer,DefaultListModel list)
    {
        t=new Thread(this,"Listener");
        this.f=f;
        this.recognizer=recognizer;
        this.list=list;
        t.start();
    }
    public void stop()
    {
        done=false;
    }
    public void run() {
        try{

            Result result=null;
            Recognizer.State rs=recognizer.getState();
            while(recognizer.getState()!=Recognizer.State.READY){
                System.out.println(rs.toString());
                Thread.sleep(500);
            }
            String digit="";
            f.setMessage("Speak ");
            while (done)
            {
```

Statistical Voice Calculator

```
result= recognizer.recognize();

if (result != null ) {
    String resultText = result.getBestFinalResultNoFiller();
    System.out.print(resultText );
    f.setMessage("You Speak: "+resultText);
    if(resultText.equals("discrete"))
    {
        f.changeData(2);f.setMessage("Wait.....");
        return;
    }
    if(resultText.equals("main"))
    {
        f.changeData(0);f.setMessage("Wait.....");
        return;
    }
    if(resultText.equals("group"))
    {
        f.changeData(3);f.setMessage("Wait.....");
        return;
    }
    if(resultText.equals("b")||resultText.equals("back"))
    {
        if(list.getSize()<1)continue;
        list.removeElementAt(list.getSize()-1);digit="";continue;
    }
    if(resultText.equals("c"))
    {
        list.removeAllElements();digit="";continue;
    }
    if(resultText.equals("next")||resultText.equals("n"))
    {
        if(!digit.equals("")){
            list.addElement(digit);digit="";}
        continue;
    }
    if(resultText.equals("answer")||resultText.equals("result")){
        calculate();
    }
    else
    {
        if(resultText.equals("one")){ digit+="1 ";}
        else if(resultText.equals("two")){ digit+="2 ";}
        else if(resultText.equals("three")){ digit+="3 ";}
        else if(resultText.equals("four")){ digit+="4 ";}
```

Statistical Voice Calculator

```
        else if(resultText.equals("five")){ digit+="5";}
        else if(resultText.equals("six")){ digit+="6";}
        else if(resultText.equals("seven")){ digit+="7";}
        else if(resultText.equals("eight")){ digit+="8";}
        else if(resultText.equals("nine")){ digit+="9";}
        else if(resultText.equals("zero")){ digit+="0";}
    }
} else {
    System.out.println("I can't hear what you said.\n");
}
}

} catch(Exception e){System.out.println(e.toString());}
} //run End

void calculate()
{
    int elements=list.getSize();
    int fi=0;
    int datalist[]=new int[elements];

    for(int i=0;i<elements;i++)
    {
        int value=Integer.parseInt(list.getElementAt(i).toString());
        datalist[i]=value;
        fi+=value;
    }
    int list[]=datalist;

    for(int i=0;i<elements;i++)
    {
        for(int j=i+1;j<elements;j++)
        {
            if(datalist[i]>datalist[j])
            {
                int temp=datalist[i];
                datalist[i]=datalist[j];
                datalist[j]=temp;
            }
        }
    }
}
```

```
System.out.println("Elements="+elements);
System.out.println("fi="+fi);
if(fi==0||elements==0)return;
float mean=(float)fi/elements;
String result="Mean : "+mean;
//f.setResult("Mean: "+mean);
if(elements%2!=0)
{
    int m=(elements+1)/2;
    result+=" Median: "+datalist[m-1];
}else
{
    int m=(elements)/2;
    mean=(datalist[m-1]+datalist[m])/2;
    result+=" Median: "+mean;
}

int max=0, maxCount=0;
for (int i = 0; i <elements; ++i)
{
    int count = 0;
    for (int j = 0; j <elements; ++j)
    {
        if (datalist[j] == datalist[i])
            ++count;
    }
    if (count > maxCount)
    {
        maxCount = count;
        max = datalist[i];
    }
}
if(maxCount==1)    result+=" Mode : None";
else result+=" Mode : "+max;
//f.setResult(result);
new MyDialog(f,list,datalist,result);
}
}
```

4. MyDialog.java

This code is for displaying answer of Individual data.

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Container;
import javax.swing.JDialog;
import javax.swing.JScrollPane;
import java.awt.BorderLayout;
import java.awt.Dimension;
import javax.swing.JTable;
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.GridBagLayout;
import java.awt.GridBagConstraints;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Color;
import javax.swing.JOptionPane;
import java.util.Vector;
import java.util.Enumeraion;
import javax.swing.JTable;
import javax.swing.BoxLayout;
import javax.swing.BorderFactory;
import javax.swing.table.DefaultTableModel;
public class MyDialog extends JDialog
{

    public MyDialog(JFrame parent,int filist[],int sortfi[],String result){
        super(parent,"Result",true);
        setLocation(parent.getLocation().x+100,parent.getLocation().y+100);
        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints c = new GridBagConstraints();
        panel.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
        Vector col=new Vector(2);
        col.addElement("FI");col.addElement("Sorted FI");
        Vector data=new Vector(5);
        DefaultTableModel model =new DefaultTableModel(data,col);
        JTable table=new JTable(model);
```



```
        int fi=0;

        for(int i=0;i<filist.length;i++)
        {
            Vector v=new Vector(2);
            v.add(filist[i]);
            v.add(sortfi[i]);
            fi+=filist[i];
            model.addRow(v);
        }
        Vector v=new Vector(2);
        v.add("Total=");v.add(fi);
        model.addRow(v);
        // container.add(table.getTableHeader(), BorderLayout.PAGE_START);
//container.add(table, BorderLayout.CENTER);
        c.fill = GridBagConstraints.HORIZONTAL;
        c.fill = GridBagConstraints.VERTICAL;
        c.gridx = 0;
        c.gridy = 0;
        panel.add(table.getTableHeader(),c);
        c.gridx = 0;
        c.gridy = 1;
        panel.add(table,c);
        c.gridx = 0;
        c.gridy = 2;
        panel.add(new JLabel(result),c);
        add(panel);
        pack();
        setVisible(true);
    }

}
```

5. VoiceListenerDis.java

This code is for calculation of Discrete Data.

```
import edu.cmu.sphinx.frontend.util.Microphone;
import edu.cmu.sphinx.recognizer.Recognizer;
import edu.cmu.sphinx.result.Result;
import edu.cmu.sphinx.util.props.ConfigurationManager;
import javax.swing.JFrame;
import javax.swing.DefaultListModel;
class VoiceListenerDis implements Runnable
{
```

Statistical Voice Calculator

```
Thread t;
boolean done=true;
GUI f;
Recognizer recognizer;
DefaultListModel xi_list,fi_list;
//constructor
VoiceListenerDis(GUI f,Recognizer recognizer,DefaultListModel xi_list,DefaultListModel
fi_list)
{
    t=new Thread(this,"Listener");
    this.f=f;
    this.recognizer=recognizer;
    this.xi_list=xi_list;
    this.fi_list=fi_list;
    t.start();
}
public void stop()
{
    done=false;
}
public void run() {
    try{

        Result result=null;
        Recognizer.State rs=recognizer.getState();
        while(recognizer.getState()!=Recognizer.State.READY){
            System.out.println(rs.toString());
            Thread.sleep(500);
        }
        int i=0;
        String digit="";
        f.setMessage("Speak ");
        while (done)
        {

            result= recognizer.recognize();

            if (result != null )
            {
                String resultText = result.getBestFinalResultNoFiller();
                System.out.print(resultText );
                f.setMessage("You Speak: "+resultText);
                if(resultText.equals("x")||resultText.equals("xi"))
                {
                    i=0;continue;
                }
            }
        }
    }
}
```

```
        }
        if(resultText.equals("fi")||resultText.equals("f"))
        {
            i=1;continue;
        }
        if(resultText.equals("b")||resultText.equals("back"))
        {
            if(i==0 && xi_list.getSize()!=0
){xi_list.removeElementAt(xi_list.getSize()-1);digit="";continue;}
            if(i==1 && fi_list.getSize()!=0
){fi_list.removeElementAt(fi_list.getSize()-1);digit="";continue;}
        }
        if(resultText.equals("main"))
        {
            f.changeData(0);f.setMessage("Wait.....");
            return;
        }
        if(resultText.equals("individual"))
        {
            f.changeData(1);f.setMessage("Wait.....");
            return;
        }
        if(resultText.equals("group"))
        {
            f.changeData(3);f.setMessage("Wait.....");
            return;
        }
        if(resultText.equals("c"))
        {
            fi_list.removeAllElements();xi_list.removeAllElements();digit="";continue;
        }
        if(resultText.equals("next")||resultText.equals("n"))
        {
            if(!digit.equals("")&&i==0){xi_list.addElement(digit);digit="";}

            if(!digit.equals("")&&i==1){fi_list.addElement(digit);digit="";}
            continue;
        }
        if(resultText.equals("answer")||resultText.equals("result"))
        {
            calculate();
        }
        else
        {
```

```
        if(resultText.equals("one")){digit+="1";}
        else if(resultText.equals("two")){digit+="2";}
        else if(resultText.equals("three")){digit+="3";}
        else if(resultText.equals("four")){digit+="4";}
        else if(resultText.equals("five")){digit+="5";}
        else if(resultText.equals("six")){digit+="6";}
        else if(resultText.equals("seven")){digit+="7";}
        else if(resultText.equals("eight")){digit+="8";}
        else if(resultText.equals("nine")){digit+="9";}
        else if(resultText.equals("zero")){digit+="0";}
    }
}

        else
        {
            System.out.println("I can't hear what you said.\n");
        }
}

        }catch(Exception e){System.out.println(e.toString());}
}

void calculate()
{

    int xi_elements=xi_list.getSize();
    int fi_elements=fi_list.getSize();
    if(xi_elements!=fi_elements)return;
    if(xi_elements==0 || fi_elements==0)return;
    System.out.println("xi="+xi_elements);
    System.out.println("fi="+fi_elements);
    int fi=0,fixi=0;
    int xi_datalist[]=new int[xi_elements];
    int fi_datalist[]=new int[fi_elements];
    for(int i=0;i<xi_elements;i++)
    {
        xi_datalist[i]=Integer.parseInt(xi_list.getElementAt(i).toString());
        fi_datalist[i]=Integer.parseInt(fi_list.getElementAt(i).toString());
        fi+=fi_datalist[i];
        fixi+=fi_datalist[i]*xi_datalist[i];
    }

    for(int i=0;i<xi_elements;i++)
    {
        for(int j=i+1;j<xi_elements;j++)
```

```
        {
            if(fi_datalist[i]>fi_datalist[j])
            {
                int temp=fi_datalist[i];
                fi_datalist[i]=fi_datalist[j];
                fi_datalist[j]=temp;
                temp=xi_datalist[i];
                xi_datalist[i]=xi_datalist[j];
                xi_datalist[j]=temp;
            }
        }
    }

    float mean=fixi/fi;
    String result="Mean: "+mean;
    mean=(fi+1)/2;
    int cum=0;
    for(int i=0;i<xi_elements;i++)
    {
        cum+=fi_datalist[i];
        if(cum>mean)
        {
            mean=xi_datalist[i];break;
        }
    }
    result+=" Median: "+mean;
    mean=xi_datalist[xi_elements-1];
    result+=" Mode: "+mean;
    // f.setResult(result);
    new ResultDialog(f,xi_datalist,fi_datalist,fi,result);
}

}
```

6. VoiceListenerGroup.java

This Code is for calculation of Grouped Data.

```
import edu.cmu.sphinx.frontend.util.Microphone;
import edu.cmu.sphinx.recognizer.Recognizer;
import edu.cmu.sphinx.result.Result;
import edu.cmu.sphinx.util.props.ConfigurationManager;
import javax.swing.JFrame;
import javax.swing.DefaultListModel;
class VoiceListenerGroup implements Runnable
```

```
{
    Thread t;
    boolean done=true;
    GUI f;
    Recognizer recognizer;
    DefaultListModel ll_list,fi_list,ul_list;
    //constructor
    VoiceListenerGroup(GUI f,Recognizer recognizer,DefaultListModel ll_list,DefaultListModel
ul_list,DefaultListModel fi_list)
    {
        t=new Thread(this,"Listener");
        this.f=f;
        this.recognizer=recognizer;
        this.ll_list=ll_list;
        this.ul_list=ul_list;
        this.fi_list=fi_list;
        t.start();
    }
    public void stop()
    {
        done=false;
    }
    public void run() {
        try{

            Result result=null;
            Recognizer.State rs=recognizer.getState();
            while(recognizer.getState()!=Recognizer.State.READY){
                System.out.println(rs.toString());
                Thread.sleep(500);
            }
            int i=0,j=0;
            String digit="";
            String lower,upper,inter;
            f.setMessage("Speak ");
            while (done)
            {

                result= recognizer.recognize();

                if (result != null )
                {
                    String resultText = result.getBestFinalResultNoFiller();
                    System.out.print(resultText );
                    f.setMessage("You Speak: "+resultText);
```

Statistical Voice Calculator

```
        if(resultText.equals("b")||resultText.equals("back"))
        {
            if(i==0 && ll_list.getSize() != 0
){ll_list.removeElementAt(ll_list.getSize()-1);digit="";continue;}
            if(i==1 && fi_list.getSize() != 0
){fi_list.removeElementAt(fi_list.getSize()-1);digit="";continue;}
            if(i==2 && ul_list.getSize() != 0
){ul_list.removeElementAt(ul_list.getSize()-1);digit="";continue;}
        }
        if(resultText.equals("lower"))
        {
            i=0;continue;
        }
        if(resultText.equals("upper")||resultText.equals("up"))
        {
            i=2;continue;
        }
        if(resultText.equals("fi")||resultText.equals("f"))
        {
            i=1;continue;
        }
        if(resultText.equals("main"))
        {
            f.changeData(0);f.setMessage("Wait.....");
            return;
        }
        if(resultText.equals("individual"))
        {
            f.changeData(1);f.setMessage("Wait.....");

            return;
        }
        if(resultText.equals("discrete"))
        {
            f.changeData(2);f.setMessage("Wait.....");
            return;
        }
        if(resultText.equals("c"))
        {
            ul_list.removeAllElements();fi_list.removeAllElements();ll_list.removeAllElements();digit="";c
ontinue;
        }
        if(resultText.equals("next")||resultText.equals("n"))
        {
```

Statistical Voice Calculator

```
        if(!digit.equals("")&&i==0){
ll_list.addElement(digit);digit="";continue;}
        if(!digit.equals("")&&i==2){
ul_list.addElement(digit);digit="";continue;}

        if(!digit.equals("")&&i==1){ fi_list.addElement(digit);digit="";continue;}
        continue;
    }
    if(resultText.equals("answer")||resultText.equals("result"))
    {
        calculate();
    }
    else
    {
        if(resultText.equals("one")){ digit+="1";}
        else if(resultText.equals("two")){ digit+="2";}
        else if(resultText.equals("three")){ digit+="3";}
        else if(resultText.equals("four")){ digit+="4";}
        else if(resultText.equals("five")){ digit+="5";}
        else if(resultText.equals("six")){ digit+="6";}
        else if(resultText.equals("seven")){ digit+="7";}
        else if(resultText.equals("eight")){ digit+="8";}
        else if(resultText.equals("nine")){ digit+="9";}
        else if(resultText.equals("zero")){ digit+="0";}
    }
}

else
{
    System.out.println("I can't hear what you said.\n");
}
}

} catch(Exception e){System.out.println(e.toString());}
}

void calculate()
{
    int ll_elements=ll_list.getSize();
    int fi_elements=fi_list.getSize();
    int ul_elements=ul_list.getSize();
    System.out.println("LL="+ll_elements);
    System.out.println("UL="+ul_elements);
    System.out.println("FI="+fi_elements);
    if(ll_elements==0 || fi_elements==0 || ul_elements==0){f.setMessage("Zero Data Not
Allowed ");return;}
}
```



```
        if(ll_elements!=fi_elements || ll_elements!=ul_elements){f.setMessage("No's of Data
Mismatch ");return;}

        float interval=0;

        float m=Float.parseFloat(ul_list.getElementAt(0).toString())-
Float.parseFloat(ll_list.getElementAt(0).toString());

        for(int i=0;i<ll_elements;i++)
        {
            interval=Float.parseFloat(ul_list.getElementAt(i).toString())-
Float.parseFloat(ll_list.getElementAt(i).toString());
            if(m!=interval){f.setMessage("Invalid Class Data");return;}
        }

        float ll_datalist[]=new float[ll_elements];
        float ul_datalist[]=new float[ll_elements];
        int fi_datalist[]=new int[ll_elements];

        for(int i=0;i<ll_elements;i++)
        {
            ll_datalist[i]=Float.parseFloat(ll_list.getElementAt(i).toString());
            ul_datalist[i]=Float.parseFloat(ul_list.getElementAt(i).toString());
            fi_datalist[i]=Integer.parseInt(fi_list.getElementAt(i).toString());
        }

        new GroupResultDialog(f,ll_datalist,ul_datalist,fi_datalist,interval);

    }

}
```

7. GroupResultDialog.java

This code is for displaying answer of grouped data.

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Container;
import javax.swing.JDialog;
import javax.swing.JScrollPane;
import java.awt.BorderLayout;
import java.awt.Dimension;
import javax.swing.JTable;
```

Statistical Voice Calculator

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Color;
import javax.swing.JOptionPane;
import java.util.Vector;
import java.util.Enumeraion;
import javax.swing.JTable;
import java.awt.GridBagLayout;
import java.awt.GridBagConstraints;
import javax.swing.BoxLayout;
import javax.swing.BorderFactory;
import javax.swing.table.DefaultTableModel;
public class GroupResultDialog extends JDialog
{

    public GroupResultDialog(JFrame parent,float ll_list[],float ul_list[],int fi_list[],float
interval)
    {
        super(parent,"Result",true);
        setLocation(parent.getLocation().x+100,parent.getLocation().y+100);
        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints c = new GridBagConstraints();
        panel.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));

        Vector col=new Vector(5);

        col.addElement("Class");col.addElement("FI");col.addElement("Mid(XI)");col.addEleme
nt("XIFI");col.addElement("CF");
        Vector data=new Vector(5);
        DefaultTableModel model =new DefaultTableModel(data,col);
        JTable table=new JTable(model);
        //JScrollPane jsp=new JScrollPane(table);
        //table.setFillViewportHeight(true);
        //table.setPreferredScrollableViewportSize(new Dimension(150, 100));

        int cf=0;
        float mid,fixtotal=0;
        for(int i=0;i<ll_list.length;i++)
        {
            Vector v=new Vector(5);
            v.add(ll_list[i]+"-"+ul_list[i]);
            v.add(fi_list[i]);
```

Statistical Voice Calculator

```
        mid=(ll_list[i]+ul_list[i])/2;
        v.add(mid);
        v.add(mid*fi_list[i]);
        cf+=fi_list[i];
        v.add(cf);
        fixitotal+=mid*fi_list[i];
    model.addRow(v);
}
Vector v=new Vector(5);
    v.add("");v.add("N="+cf);v.add("");v.add(fixitotal);v.add("");
model.addRow(v);
    float mean=fixitotal/cf;

    float m=(float)cf/2;

    int f=0,c1=0;
    float l=0;
    System.out.println("ll---"+ll_list.length);
    for(int i=0;i<ll_list.length;i++)
    {
        int value=(Integer)model.getValueAt(i,4);
        if(value>=m )
        {
            System.out.println("i---"+i);
            l=ll_list[i];f=fi_list[i];
            if(i!=0)
                c1=(Integer)model.getValueAt(i-1,4);break;
        }
    }

    //^Median = L(i)/(f)(m-c)`
        float j=m-c1;
    float median=l+(interval/f)*j;

    //Mode = L + [(F - F1) / {(F - F1) + (F - F2)}] * h

    int max=fi_list[0];
    int cnt=0;
    for(int i=0;i<fi_list.length;i++)
    {
        if(max<fi_list[i])
        {
```

```
        max=fi_list[i];cnt=i;
    }
}
l=ll_list[cnt];
System.out.println("cnt"+cnt);
System.out.println("cnt size"+fi_list.length);
int f1=0;
if(cnt!=0)
f1=fi_list[cnt-1];
int f2=0;
if(cnt!=(fi_list.length-1))
f2=fi_list[cnt+1];

float med=((f-f1)+(f-f2));
float mode=l+((f-f1)/med)*interval;
String result="Mean : "+mean+" Median : "+median+" Mode : "+mode;
System.out.println("L..." +l);System.out.println("inter..." +interval);

System.out.println("F..." +f);System.out.println("f1..." +f1);System.out.println("f2..." +f2);
//System.out.println("m..." +mean);
System.out.println("mode..." +mode);
// container.add(table.getTableHeader(), BorderLayout.PAGE_START);
//container.add(table, BorderLayout.CENTER);
c.fill = GridBagConstraints.HORIZONTAL;
c.fill = GridBagConstraints.VERTICAL;
c.gridx = 0;
c.gridy = 0;
panel.add(table.getTableHeader(),c);
c.gridx = 0;
c.gridy = 1;
panel.add(table,c);
c.gridx = 0;
c.gridy = 2;
panel.add(new JLabel(result),c);
add(panel);
pack();
setVisible(true);
}
```

8. ResultDialog.java

This code is for displaying answer of discrete data.

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Container;
import javax.swing.JDialog;
import javax.swing.JScrollPane;
import java.awt.BorderLayout;
import java.awt.Dimension;
import javax.swing.JTable;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Color;
import javax.swing.JOptionPane;
import java.util.Vector;
import java.util.Enumuration;
import javax.swing.JTable;
import java.awt.GridBagLayout;
import java.awt.GridBagConstraints;
import javax.swing.BoxLayout;
import javax.swing.BorderFactory;
import javax.swing.table.DefaultTableModel;
public class ResultDialog extends JDialog
{

    public ResultDialog(JFrame parent,int xlist[],int filist[],int fitotal,String result)
    {
        super(parent,"Result",true);

        setLocation(parent.getLocation().x+100,parent.getLocation().y+100);
        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints c = new GridBagConstraints();
        //this.getContentPane().setBackground(Color.white);
        panel.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));

        Vector col=new Vector(4);

        col.addElement("XI");col.addElement("FI");col.addElement("CF");col.addElement("XIF
I");

        Vector data=new Vector(5);
        DefaultTableModel model =new DefaultTableModel(data,col);
```

Statistical Voice Calculator

```
JTable table=new JTable(model);
//JScrollPane jsp=new JScrollPane(table);
//table.setFillViewportHeight(true);
//table.setPreferredScrollableViewportSize(new Dimension(150, 100));

int cf=0,fixtotal=0;

for(int i=0;i<xilist.length;i++)
{
    Vector v=new Vector(4);
    v.add(xilist[i]);
    v.add(filist[i]);
    cf+=filist[i];
    v.add(cf);
    fixtotal+=filist[i]*xilist[i];
    v.add(filist[i]*xilist[i])    ;
    model.addRow(v);
}
Vector v=new Vector(4);
    v.add("");v.add("N="+fitotal);v.add("");v.add(fixtotal);
model.addRow(v);
    // container.add(table.getTableHeader(), BorderLayout.PAGE_START);
//container.add(table, BorderLayout.CENTER);
    c.fill = GridBagConstraints.HORIZONTAL;
    c.fill = GridBagConstraints.VERTICAL;
    c.gridx = 0;
c.gridy = 0;
    panel.add(table.getTableHeader(),c);
    c.gridx = 0;
c.gridy = 1;
    panel.add(table,c);
    c.gridx = 0;
c.gridy = 2;
    panel.add(new JLabel(result),c);
    add(panel);
    pack();
    setVisible(true);
}

}
```

9. VoiceListener2.java

```
import edu.cmu.sphinx.frontend.util.Microphone;
import edu.cmu.sphinx.recognizer.Recognizer;
import edu.cmu.sphinx.result.Result;
import edu.cmu.sphinx.util.props.ConfigurationManager;
import javax.swing.JFrame;
import javax.swing.DefaultListModel;
class VoiceListener2 implements Runnable
{
    Thread t;
    boolean done=true;
    JFrame f;
    Recognizer recognizer;

    //constructor
    VoiceListener2(JFrame f,Recognizer recognizer)
    {
        t=new Thread(this,"Listener");
        this.f=f;
        this.recognizer=recognizer;
        t.start();
    }
    public void stop()
    {
        done=false;
    }
    public void run() {
        try{

            Result result=null;
            Recognizer.State rs=recognizer.getState();
            while(recognizer.getState()!=Recognizer.State.READY){
                System.out.println(rs.toString());
                Thread.sleep(500);
            }

            while (done)
            {

                result= recognizer.recognize();

                if (result != null ) {
                    String resultText = result.getBestFinalResultNoFiller();
```

Statistical Voice Calculator

```
        System.out.print("R="+resultText);

        } else {
            System.out.println("I can't hear what you said.\n");
        }
    }

    }catch(Exception e){System.out.println(e.toString());}

}

}
```

10. HelloWorld.java

```
import edu.cmu.sphinx.frontend.util.Microphone;
import edu.cmu.sphinx.recognizer.Recognizer;
import edu.cmu.sphinx.result.Result;
import edu.cmu.sphinx.util.props.ConfigurationManager;

public class HelloWorld {

    public static void main(String[] args) {
        ConfigurationManager cm;

        if (args.length > 0) {
            cm = new ConfigurationManager(args[0]);
        } else {
            cm = new
ConfigurationManager(HelloWorld.class.getResource("helloworld.config.xml"));
        }

        Recognizer recognizer = (Recognizer) cm.lookup("recognizer");
        recognizer.allocate();

        // start the microphone or exit if the program fails if this is not possible
        Microphone microphone = (Microphone) cm.lookup("microphone");
        if (!microphone.startRecording()) {
            System.out.println("Cannot start microphone.");
            recognizer.deallocate();
            System.exit(1);
        }
        // loop the recognition until the program exits.
        while (true) {
```


Statistical Voice Calculator

```
Result result = recognizer.recognize();

if (result != null ) {
    String resultText = result.getBestFinalResultNoFiller();
    if(resultText.equals("next"))
        System.out.println("");
    else
        System.out.print(resultText );
} else {
    System.out.println("I can't hear what you said.\n");
}
}
}
```

GRAM Files

Grammar files contain the list of probable words that user can speak. Recognizer will identify only those words that will be specified in grammar file.

1. hello.gram

This file contains all possible words that user can speak for calculation.

```
#JSGF V1.0;

/**
 * JSGF Grammar
 */

grammar hello;

public <greet> = ( Main | Up | B |Back |Upper | Lower | Result | XI | X | F |
FI | Discrete | Group | Individual | C | Answer | N |Next |Zero | one | Two |
Three | Four | Five | Six | Seven | Eight | Nine );
```

2. maingram.gram

This file contains all possible words that user can speak for moving from one tab to another.

```
#JSGF V1.0;

/**
 * JSGF Grammar
 */

grammar hello;

public <greet> = ( Discrete | Home | Group | Individual | One | Answer |Clear
);
```

7. Testing

Software testing is an investigation conducted to provide with information about the quality of the product or service under test.

7.1 Test Plan

A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements.

Design Verification or Compliance test - to be performed during the development or approval stages of the product, typically on a small sample of units.

Service and Repair test - to be performed as required over the service life of the product.

7.2 Testing Strategies

We followed “Black Box Testing” technique for performing Testing of our system. We need to provide the Inputs to the system and see that Outputs are as per the specification or not. Inside process is not visible to us.

7.3 Test Cases, Test Data and Test Results

Test Case 1: Selecting method for calculation.

Test Case Name:	Selecting method for calculation	
Test Case Description:	This test case is to see that whether tab changes for respective methods when user speaks the name of method for calculation.	
Serial No.	Voice Input	Expected Output
1	Individual	Individual tab opens.
2	Discrete	Discrete tab opens.
3	Group	Group tab opens.

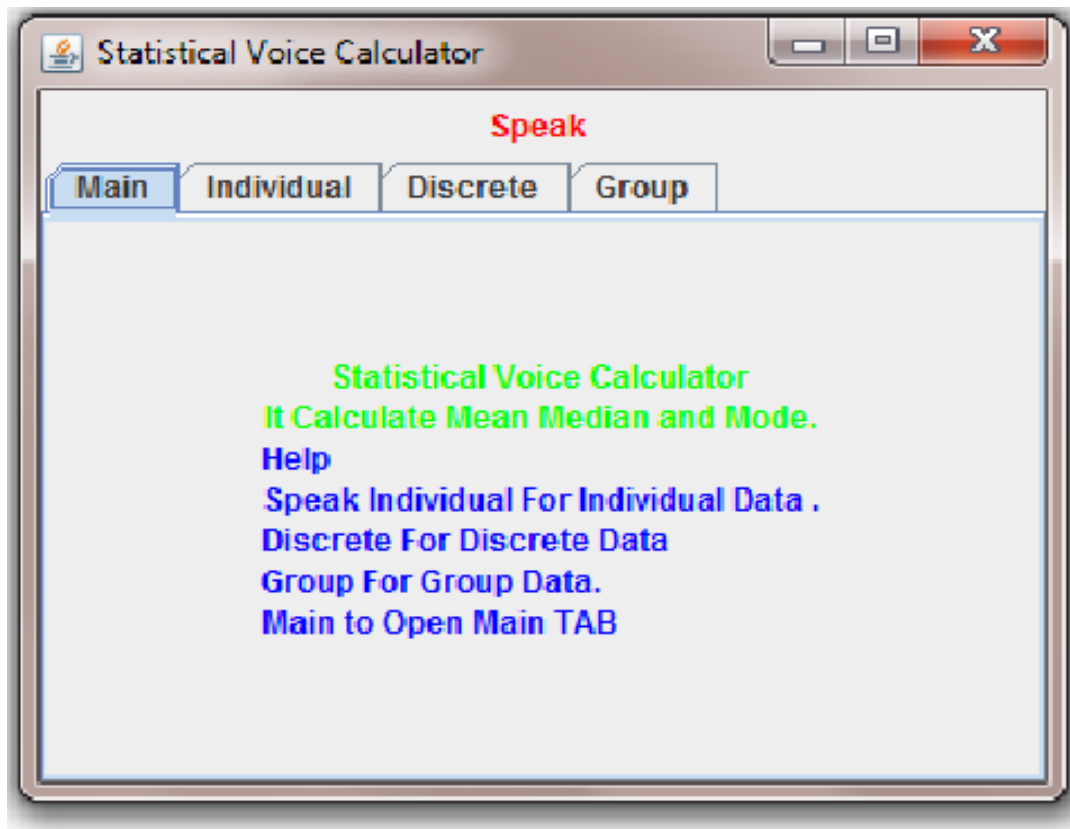
Statistical Voice Calculator

Test Case 2: Input Process.

Test Case Name:	Input Process	
Test Case Description:	This test case is to see which all values user can give to the system.	
For Individual Data:		
Serial no.	Voice Input	Expected Result
1	FI	Focus to the list box where data has to be entered.
2	N	User can now enter another value.
3	B	Erases the previous value.
4	C	Clears the list box.
5	Answer/Result	Displays answer.
For Discrete Data:		
1	XI/X	Focus to the xi list box.
2	N	User can now enter another value.
3	B	Erases the previous value.
4	C	Clears all fields.
5	FI	Focus to the Fi list box.
6	Answer/Result	Displays answer.
For Grouped Data:		
1	LOWER	Focus to lower limit list box.
2	UP/UPPER	Focus to upper limit list box.
3	N	User can now enter another value.
4	B	Erases the previous value.
5	C	Clears all fields.
6	FI	Focus to Fi list box.
7	Answer/Result	Displays answer.
General Inputs:		
1	0-9	Displays the number.
2	Rest of words	No change reflected in any list box.

8. User Manuals

For Main Page

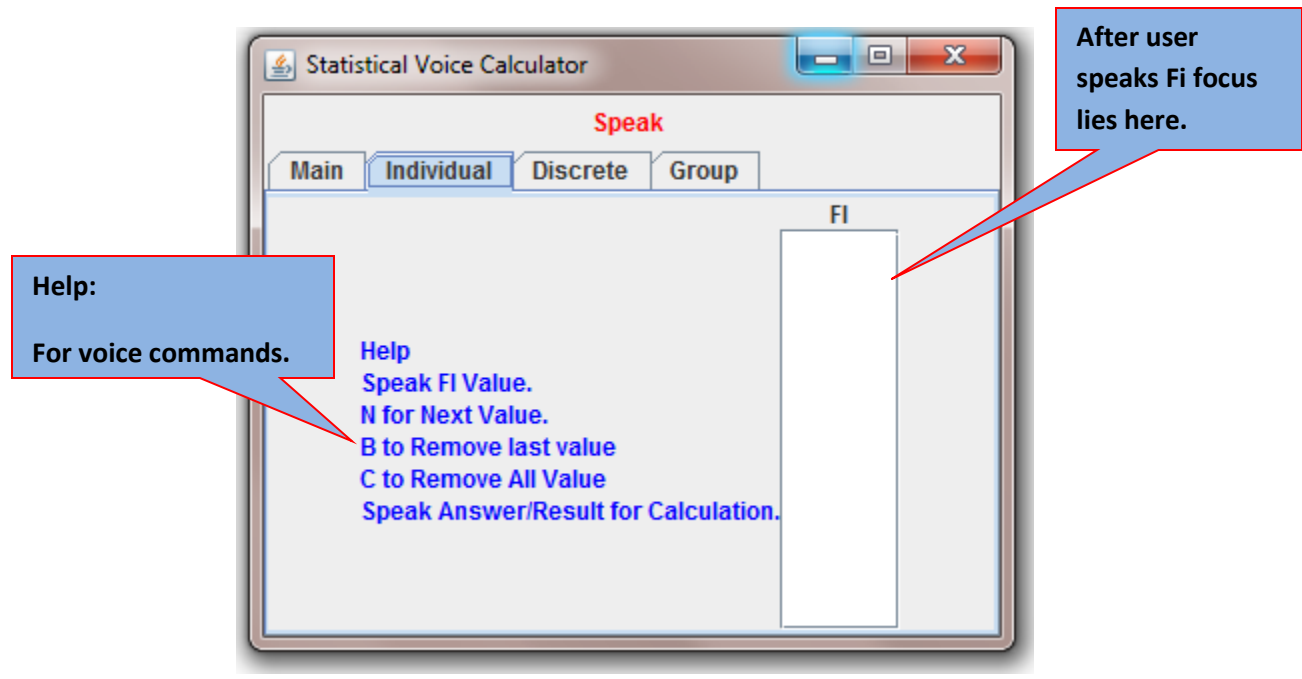


As such, help is given on each tab, At this point to select the method for calculation, User need to:

- Speak **“Individual”** to find Mean, Median and Mode of Individual Data.
- Speak **“Discrete”** to find Mean, Median and Mode of Discrete Data.
- Speak **“Group”** to find Mean, Median and Mode of Grouped Data.

Statistical Voice Calculator

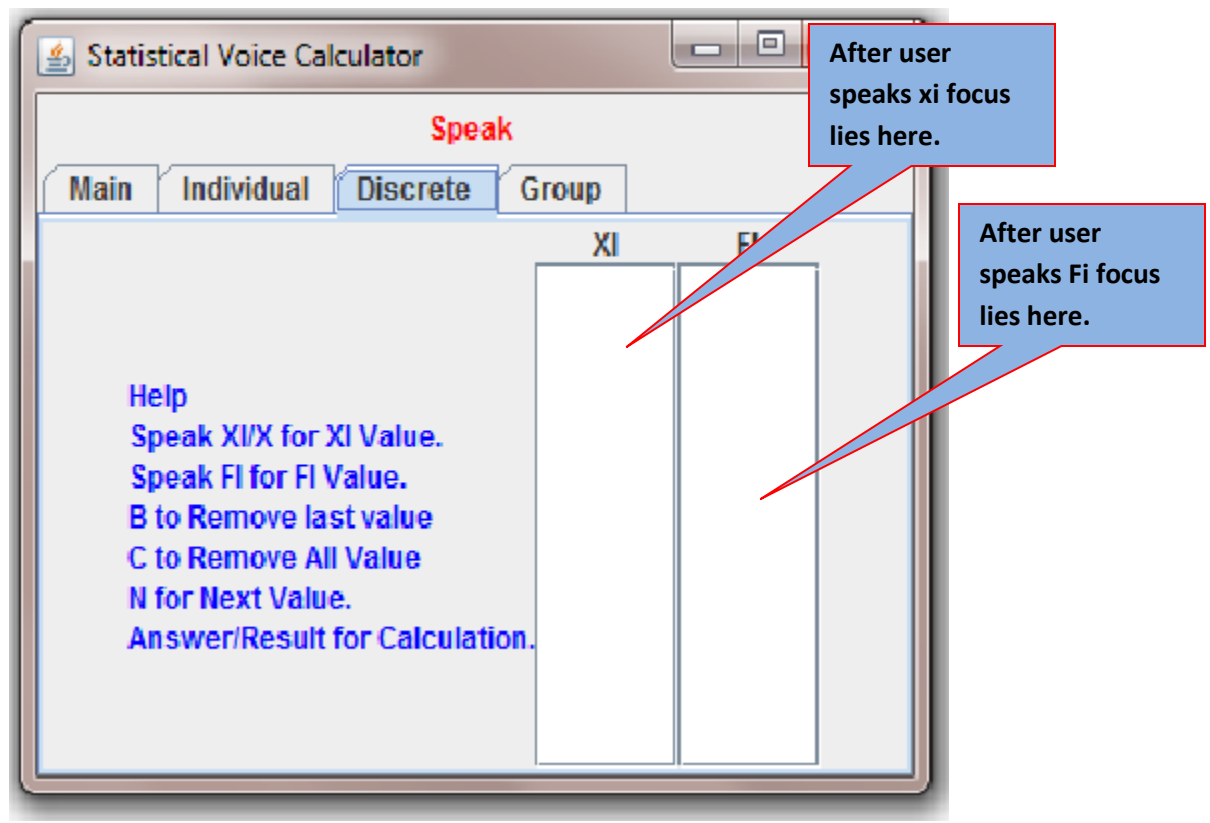
For Individual Data



- After User speaks “**Individual**”, this tab opens. When User speaks **FI** a focus will be shifted to list box for the user to enter the value.
- After specifying “10”, to insert another value, user need to Speak “**N**”
- If user by mistake specifies wrong value he can correct it by “**B**” voice command.
- To clear all fields, User can by “**C**” voice command.
- To get answer, User can by “**Answer/Result**” voice command.

Statistical Voice Calculator

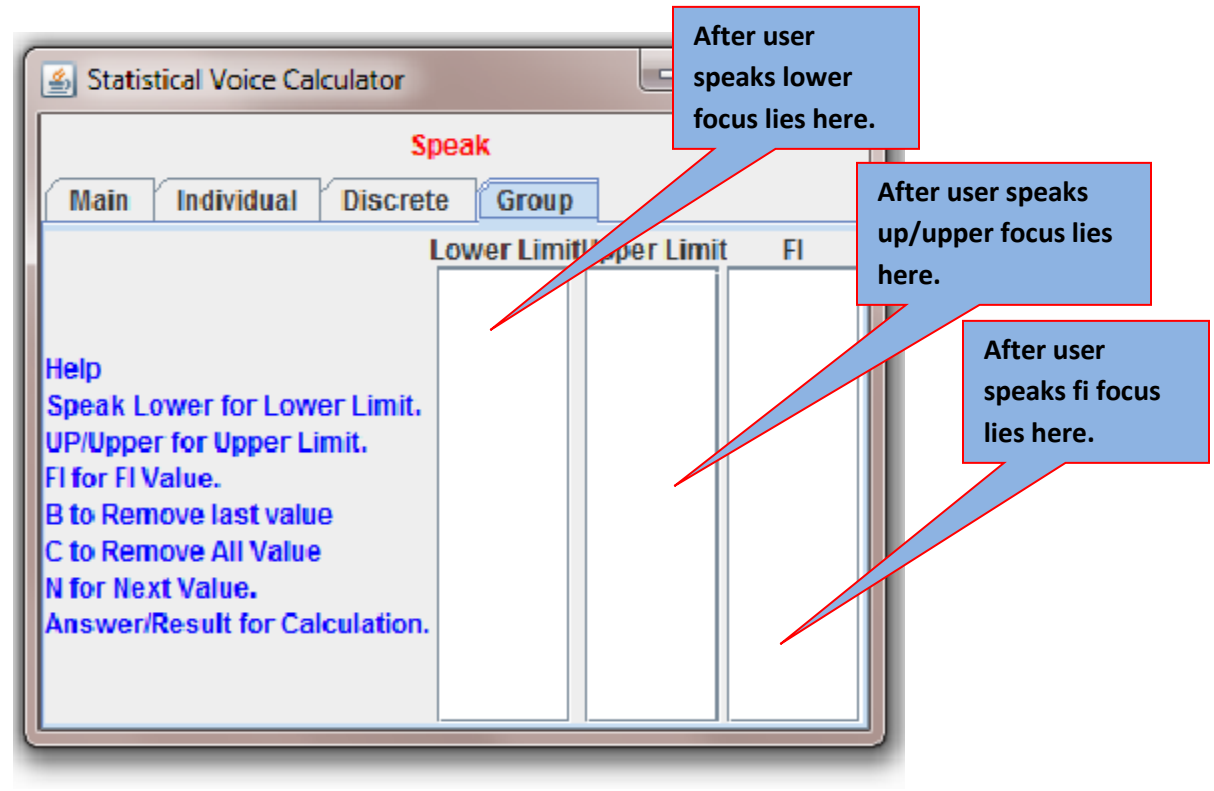
For Discrete Data



- User need to speak "**Xi**" to input xi values.
- User need to speak "**Fi**" to input frequency values.
- In both while inserting xi or fi, in order to insert another value, user need to speak "**N**".
- If user by mistake specifies wrong value he can correct it by "**B**" voice command.
- To clear all fields, User can by "**C**" voice command.
- To get answer, User can by "**Answer/Result**" voice command

Statistical Voice Calculator

For Grouped Data:



- User need to speak **“Lower”** to input lower limit values.
- User need to speak **“Upper”** to input upper limit values.
- User need to speak **“fi”** to input frequency values.
- In all above cases while inserting lower limits, upper limits and frequencies, in order to insert another value, user need to speak **“N”**.
- If user by mistake specifies wrong value he can correct it by **“B”** voice command.
- To clear all fields, User can by **“C”** voice command.
- To get answer, User can by **“Answer/Result”** voice command

9. Future Expansion

In “statistical voice calculator” we can find Mean, Median and Mode of only integer values, we cannot find the answer for the fractional data. In our current system, we have to specify numbers like 10, 11, 20, 25 but cannot specify any value like 10.5, 11.8, 20.5, 25.5 as Input.

So as a part of Future Expansion, we would like to overcome this problem by making appropriate changes to the system so that we can have fractional values as input for calculation of Mean, Median and Mode.

10. Bibliography & References

- www.cloudgarden.com
- www.cmusphinx.sourceforge.net/sphinx4
for speech API.
- www.google.com
- Java doc
- Analysis and Design of Information System
By: James A. Senn