

5-2015

Voice Driven Email Client

Channa J. Kumarage

Saint Cloud State University, cjkumarage@gmail.com

Follow this and additional works at: https://repository.stcloudstate.edu/csit_etds

Recommended Citation

Kumarage, Channa J., "Voice Driven Email Client" (2015). *Culminating Projects in Computer Science and Information Technology*. 4.
https://repository.stcloudstate.edu/csit_etds/4

This Thesis is brought to you for free and open access by the Department of Computer Science and Information Technology at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Computer Science and Information Technology by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rswexelbaum@stcloudstate.edu.

Voice Driven Email Client

by

Channa Jayanath Kumarage

A Thesis

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Computer Science

May 2015

Thesis Committee:

Dr. Bryant Julstrom, Chairperson

Dr. Jayantha Herath

Dr. Ettien Koffi

Abstract

With the advancements of technology, today the needs and expectations of humans keep growing seeking for more convenience. People prefer to communicate with technology using natural interactive approaches rather than with a key-board or mouse. Voice recognition is one such mostly preferred natural interactive approach.

Though voice recognition was originally introduced in applications for differently abled individuals, today we see with the complexity in work environments, a majority of people prefer to use more natural interactive approaches such as voice commands due to its' convenience.

The main objective of this project is to develop an Email client which allows users to perform tasks within the application by eliminating the use of the key-board or mouse. The system allows users to perform all the tasks within the application using voice commands. At present widely used web email service providers such as Gmail, yahoo etc do not support voice commands, the Voice Driven Email Client makes it possible to connect to a web based email client and carry out tasks using special voice commands. It will allow the users to dictate as well as navigate purely on voice commands.

This voice enabled email client introduces a better approach to accessing web-based email clients instead of remaining at the traditional text-based, typing and clicking approach. Further it will reach a wider target audience including differently abled individuals such as people without hands, people with difficulties in hand movements, dyslexic people who find it difficult to write and spell etc.

This document outlines the motivation, background, problem description, scope, research and analysis, design, implementation and testing carried out to develop the Voice Driven Email Client.

Acknowledgement

A word of appreciation is necessary for the many individuals who helped in different ways during the period of the project life cycle.

I extend a sincere note of gratitude to my student Harshani Perera for her input throughout the development and implementation of this project, for conducting informative progress meetings and for keeping me in check with my project deadlines.

I would like to thank my lecturers at the University of Colombo School of Computing (UCSC), Sri Lanka and Saint Cloud State University (SCSU), Minnesota, USA for not only laying the foundation of knowledge that I used, but also for imparting practical knowledge that I was able to use while developing the Voice Driven Email Client system.

My colleagues at UCSC and SCSU deserve my gratefulness as they mentored and advised me in times of doubt and encouraged me to think outside the box with regards to the development of the system.

I am thankful to my parents and family for supporting me throughout the project period and for their strong words of encouragement at all times.

I thank the staff, supporting staff, my seniors and friends at the UCSC and SCSU for all their support and the Saint Cloud State University for giving me the opportunity to develop such a challenging project.

Contents

Abstract	1
Acknowledgement	3
Table of Figures	6
Chapter 1 : Introduction.....	9
1.1 Chapter Overview.....	9
1.2 Motivation.....	9
1.2.1 Ease of use.....	9
1.2.2 Wider target audience - Differently abled individuals & busy users.....	9
1.2.3 Advancements in Speech Technology	9
1.3 Problem Description	10
1.4 Project Scope.....	11
1.5 Project Aims and Objectives.....	12
1.6 Approach	14
1.6.1 Currently Practiced Solution	14
1.2.2Ultimate Solution.....	14
Chapter 2 Literature Review	15
2.1 Chapter Overview.....	15
2.2 System Investigation.....	15
2.3 Similar Applications Analysis.....	16
2.3.1 Dragon Naturally Speaking.....	16
2.3.2 Chrome 11 Speech Recognition	17
2.3.3 Windows Speech Recognition	18
2.3.4 Siri.....	19
2.3 Technical Investigation.....	20
2.3.1 Speech Recognition.....	20
2.3.2 Speech Recognition APIs.....	24
2.3.3 Programming Language	27
2.3.4 POP vs. IMAP	28
2.3.5 Gmail Plug-in	29
Chapter 3 Requirement Specification	30
3.1 Chapter Overview.....	30
3.2 Functional Requirements.....	30

3.2.1 Functional Requirements for version 1 (own Interface)	30
3.2.1 Functional Requirements for version 2 (Gmail Interface).....	33
3.3 Non-Functional Requirements.....	35
3.4 Resource Requirements for Development.....	36
3.4.1 Software Requirement	36
3.4.2 Hardware Requirement.....	36
3.4.3 Other Requirements.....	36
Chapter 4 System Design	37
4.1 Chapter Overview	37
4.2 System Architecture.....	37
4.2.1 High-level Architecture Diagram	37
4.2.2 Three Tier Architecture	39
4.3 Use Case Diagram	43
4.4 Use Cases.....	44
4.5 Command and Diction Functions Design.....	62
4.5.1 Designing Commands.....	62
4.5.2 Differentiating Commands and Dictation	64
4.6 Screen Design	68
4.7 Property Matrix	70
4.7.1 Inbox.....	70
4.7.2 Login Form.....	74
4.7.3 Email Client Form	75
4.7.4 New Email.....	76
Chapter 5 System Implementation.....	81
5.1 Chapter Overview	81
5.2 Implementation Environment	81
5.2 Source Codes.....	81
5.2.1 Speech Recognition	81
5.2.2 Sending Email	84
5.2.3 Retrieving Email	85
5.2.4 Get Contacts.....	86
5.2.5 Converting English word to numbers	87
5.3 Source code overview	88

Chapter 6 Testing.....	89
6.1 Chapter Overview.....	89
6.2 Unit Testing.....	89
6.2.1 Speech Recognition.....	89
6.2.2 Sending Emails.....	91
6.2.3 Receiving Emails.....	91
6.3 Integration Testing.....	92
6.4 Overall system testing.....	94
6.4.1 Separate interface.....	94
6.4.1 Gmail interface.....	124
6.5 Performance Testing.....	146
Chapter 7 Deployment.....	148
7.1 Chapter overview.....	148
7.2 Setting up the system.....	148
7.2.1 Voice Driven Email Client Version – 1(Separate Interface).....	148
7.2.2 Voice Driven Email Client Version – 2 (Gmail Interface).....	149
Chapter 8 Evaluation.....	150
8.1 Limitations.....	150
8.2 Future Enhancements.....	150
8.3 Conclusion.....	150
Chapter 9 Appendix.....	152
9.1 References.....	152
9.2 Bibliography.....	153
9.3 List of Abbreviations.....	156
9.4 User manual.....	157
9.4.1 Voice Driven Email Client Version – 1 (Separate interface).....	157
9.4.2 Voice Driven Email Client Version – 2 (Gmail interface).....	173

Table of Figures

Figure 1 : Worldwide Email Account growth prediction – The Radicati Group, 2011.....	10
Figure 2: Chrome 11 speech input – Google Chrome, n.d.....	17

Figure 3: Dictate Email using Siri – media.wiley.com, n.d.....	19
Figure 4: SAPI Architecture – microsoft.com, n.d.....	24
Figure 5: Sphinx 4 Architecture – sourceforge.net, n.d.....	26
Figure 6 : High –Level System Architecture Diagram.....	38
Figure 7 : Presentation Tier of the System.....	39
Figure 8 : Application Tier of the System.....	40
Figure 9 : Data Tier of the System.....	42
Figure 10 : System Use Case Diagram.....	43
Figure 11 : Handling Commands.....	65
Figure 12 : Start up Screen Design.....	68
Figure 13 : Login Screen Design.....	68
Figure 14 : Alignment of Buttons Screen Design.....	69
Figure 15 : Displaying Emails Screen Design.....	69
Figure 16 : Source Code Overview.....	88
Figure 17 : startup splash screen.....	157
Figure 18: System Login Form.....	157
Figure 19: Login Form with Completed Fields.....	158
Figure 20: User Login Form with Invalid Credentials.....	158
Figure 21: Successful login.....	159
Figure 22: Create New Email.....	159
Figure 23: To Field - Search contact.....	160
Figure 24 : To Field - Add contact.....	160
Figure 25: CC Field – Focus.....	161
Figure 26: CC Field - Add contact.....	161
Figure 27: BCC Field - Focus.....	162
Figure 28: BCC Field – Add contact.....	162
Figure 29 : Select Subject.....	163
Figure 30 : Dictate to ‘subject’ field.....	163
Figure 31 : Mistype on Subject.....	164
Figure 32 : Undo Mistyping on Subject.....	164
Figure 33 : Select Attachments.....	165
Figure 34 : Attach File.....	165
Figure 35 : Select Body.....	166
Figure 36 : Dictate Body.....	166
Figure 37 : Mistyping on Body.....	167
Figure 38 : Undo Mistyping on Body.....	167
Figure 39 : Send Message.....	168
Figure 40 : Close Confirmation Popup Form.....	168
Figure 41 : Retrieve Inbox Message.....	169
Figure 42 : Open Email.....	169
Figure 43 : Reply to Email.....	170
Figure 44 : Reply to All - 1.....	170
Figure 45 : Reply to All - 2.....	171

Figure 46 : Reply to All - 3	171
Figure 47 : Forward Email - 1	172
Figure 48 : Forward Email - 2	172
Figure 49 : Startup Splash Screen	173
Figure 50: System Login Form	173
Figure 51: Login Form with Completed Fields	174
Figure 52 : User Login Form with Invalid Credentials.....	174
Figure 53: Successful Login.....	175
Figure 54: Create New Email	175
Figure 55: To Field – Focus	176
Figure 56: To Field - Add Contact.....	176
Figure 57: CC Field - Focus.....	177
Figure 58: CC Field - Add Contact.....	177
Figure 59: BCC Field - Focus.....	178
Figure 60: BCC Field – Add Contact.....	178
Figure 61 : Select Subject.....	179
Figure 62 : Dictate to ‘Subject’ Field.....	179
Figure 63 : Mistype on Subject	180
Figure 64 : Undo Mistyping on Subject	180
Figure 65 : Select Body.....	181
Figure 66 : Dictate Body	181
Figure 67 : Mistyping on Message.....	182
Figure 68 : Undo Mistyping on Message.....	182
Figure 69 : Send Message	183
Figure 70 : Message Sent	183
Figure 71 : Retrieve Inbox Messages	184
Figure 72 : Open Email.....	184
Figure 73 : Reply to Email.....	185
Figure 74 : Reply/Reply all/Forward Dictating Body.....	185
Figure 75 : Reply to All	186
Figure 76 : Forward Email.....	186
Figure 77 : Delete Email - 1	187
Figure 78 : Delete Email - 2	187

Chapter 1 : Introduction

1.1 Chapter Overview

This chapter outlines the motivation behind the research and implementation, the main objectives and aims of the project, project scope and the approach followed. This chapter shows the need for a Voice Driven Email Client in today's context and enables one to understand the proposed system comprehensibly.

1.2 Motivation

1.2.1 Ease of use

Checking Email has become an everyday task of the modern generation. It doesn't simply finish by reading the email; it will require replying to the email or creating new emails to be sent. When creating a new email or replying to an email there are many fields to be filled, generally the main fields would be 'to', 'subject' and 'body'. Some additional fields such as 'cc', 'bcc' fields might also have to be filled in certain cases. To carry out these tasks, a considerable amount of typing and clicking is required. Thus, it creates a need for a more convenient and innovative way to access an 'Email client'.

Convenience for humans can be achieved by following a natural approach. Humans' mode of communication is speech; therefore the most suitable approach to be followed in this case would be to offer a solution that allows the user to interact with the application using voice commands. This would allow users to do their everyday tasks which in this case is checking email in a more easy, convenient and efficient manner.

1.2.2 Wider target audience - Differently abled individuals & busy users

As discussed in section 1.2.1, it is clear that navigating through an Email Client involves a significant amount of typing and clicking. This manner of interaction limits certain users from accessing the system, for instance differently abled individuals such as users without the ability to use their hands, users with limited hand movement, dyslexic users who find it difficult to write and spell etc or in some cases even extremely busy users from. Therefore an Email Client which is driven by Voice Commands would be an ideal solution. The Voice Driven Email Client not only offers convenience but also allows expanding the target audience.

1.2.3 Advancements in Speech Technology

Voice recognition enables users to use their voice to perform tasks instead of typing or clicking. Voice recognition technology is mainly used in two ways in systems:

- To control the software
- To dictate

With the developments in speech technology, discrete speech recognition has been replaced by continuous speech recognition. User can now fluently dictate text without having to pause between words.

Further the availability of Speech Recognition APIs makes it possible for developers to create voice driven software with less effort without having to reinvent the wheel.

1.3 Problem Description

Today 'Checking Email' has become an everyday task which consumes a considerable amount of time. According to the Email Statics Report 2011-2015 carried out by the The Radicati Group.Inc (2011) the number of email accounts worldwide was 3.1 billion in 2011. It is predicted that the email account base will increase to nearly 4.1 billion by the end of 2015. These statics prove the large growth in email users in the years to come.

	2011	2012	2013	2014	2015
Worldwide Email Accounts (M)	3,146	3,375	3,606	3,843	4,087
Corporate Email Accounts	788	850	918	991	1,070
<i>% Corporate Email Accounts</i>	25%	25%	25%	26%	26%
Consumer Email Accounts	2,358	2,525	2,688	2,852	3,017
<i>% Consumer Email Accounts</i>	75%	75%	75%	74%	74%

Corporate vs. Consumer Email Accounts, 2011–2015

Figure 1 : Worldwide Email Account growth prediction – The Radicati Group, 2011

Although the email account base keeps increasing rapidly, the practice of accessing web-based clients seems to remain the same. It is still the traditional text-based, typing and clicking approach used. The evolving, convenient approach is voice driven systems. Users prefer working with voice since it is closer to human's method of communication which is speech.

It is not only busy users who will benefit from such systems. A voice driven email client will allow differently abled individuals such as users without hands, users with limited hand movement, dyslexic users who find it difficult to write and spell etc to access the system. Therefore the motive to implement a voice driven email client is to increase the quality of the practice accessing an email client by a large amount of users.

1.4 Project Scope

The proposed solution is a speech-enabled Email Client that allows the user to perform email-specific functions purely on voice commands. It is a separate interface that connects to an existing Web-based Email Client and enables the user to navigate and dictate using voice, thus eliminating the use of a key-board or mouse.

The Scope of the Voice Driven Email Client will be limited to the following aspects.

- Continuous Speech Recognition – the system will allow the user to fluently dictate sentences without having to pause between words.
- Speaker Dependent Speech Recognition – the system will recognize voice input from a single user, provided necessary training is done.
- English Language – the system will perform speech recognition in English, therefore the user can only speak in English.
- Homophone Detection – they system will handle Homophones only for navigation commands (not for Dictations).
- Large Vocabulary – the system will have a comparatively large vocabulary that will allow user to dictate without a problem.
- Gmail – the system will allow the user to connect only to Gmail.
- Navigation and Dictation – the system will allow the user to both navigate and dictate using voice input.
- Usage – the system will work only on Desktop and Laptop Computers.

1.5 Project Aims and Objectives

Described below are the main Aims and Objectives the system requires to accomplish at the end of the project.

- Development of a separate interface that allows the user to connect to a web based Email Client.

The proposed system will be a separate interface that allows connecting to an existing web-based email client for retrieving and sending emails. The user should be able to perform all email specific functions using voice. For this project purpose the scope is being limited to Gmail web client only, due to the time constraints

- Speech Recognition

The system should be speech enabled. It should accurately recognize the phrase spoken by the user and perform the required function accordingly. The system must make it possible for the user to speak in a natural way without having to pause much between words (Continuous Speech Recognition). It should allow the user to Control the software as well as Dictate.

- Control of software

The system will allow the user to Navigate through the system using specific voice commands. The navigations include simulating button clicks and moving to different fields. Listed below are few of the Navigations supported and the respective voice commands.

- to open a new email - **select new**
- to view inbox emails - **select inbox**
- to add a contact from address book to 'To' field - **include 'Contact ID'**
Example: **include one**

- Dictation

The system will also allow the user to dictate words, phrases or sentences. It should enable the user to dictate to the allowed fields. For example,

- dictate body of email
- dictate subject field of email

- Homophone Detection

The system should detect Homophones in navigation commands and handle them accordingly. For example when Navigating to the 'To' Field the system may recognize the command as one of the following;

- Original command - select to
 - Homophones 1 - select two
 - Homophone 2 - select too

Therefore the system should take such instances into account.

- Ease of use to user

Though the system implementation is complex, the system should be designed in such a way that the implementation complexities are hidden from the user facilitating seamless navigation.

1.6 Approach

1.6.1 Currently Practiced Solution

There are many software applications currently available that offer speech recognition. But none of these applications offer full support for 'Web-based Email Clients' to be driven on voice commands. Out of these, most applications offers support for dictation but does not offer support for navigation. In section 2.3 such similar applications and their capabilities are discussed in detail.

Though dictation support offered to an application helps to improve ease of use it does not contribute to convenience and does not help to increase the target audience. For an application to be fully driven on voice commands it should provide the ability to control the software as well the ability for the user to fluently dictate texts.

Most speech recognition software are targeted to be used on multiple applications in a computer. This often works well for dictation but lacks the capability of full software control unless these applications are similar. For example the Microsoft Office Package applications contain similar navigation patterns since the basic layout of the menus are similar. But such speech recognition software often cannot support navigation support to an application that contains different layouts and menus. One such application is the web-based email client. This creates a need to be able to drive a Web-based Email Client fully on voice commands.

1.2.2Ultimate Solution

The ultimate solution would be to have a Voice Driven Email Client that allows connecting to web-based email clients for both Navigation and Dictation. Having a Separate interface for this purpose would allow connecting to different web-based email clients such as Gmail, yahoo, hotmail etc. which in turn expands target audience. On the other hand a Plug-in will be restricted to specific web-based email client or browser; thus reduces target audience.

Chapter 2 Literature Review

2.1 Chapter Overview

This chapter outlines the literature review of the project which includes system investigations, similar application analysis and technical investigations carried out.

2.2 System Investigation

System Investigation, often known as System Analysis plays a vital role in implementing a new system. In order for a system to cater to the exact needs of the user, it is important to clearly identify the requirements. There are many requirement gathering techniques available; however the appropriate technique will depend on the nature of the project.

Requirement gathering techniques include Interviews, Questionnaires, Document Analysis, Brainstorming, Focus Groups, Observation, and Similar System Analysis etc.

The main source used for gathering requirements for this system was through Document Analysis. Documents relating to Speech Recognition and Email Clients were reviewed. The areas of research included Speech Recognition factors, Speech APIs, Emails, Email Servers, and Email Protocols etc. In addition to Document Analysis more requirements were gathered through Observation and Similar Systems.

Similar systems included Speech Driven Systems, Email Systems and few systems that had speech recognition to a certain extent in email systems. Analyzing Similar Systems helped to clearly understand areas that need to be improved as well as best practices of existing systems.

2.3 Similar Applications Analysis

It is important to identify the existing applications available before proceeding with the project. It is not only the limitations and flaws in similar existing systems that have to be discovered, it also important to identify the best practices and features used in such systems. Thereby limitations and flaws can be eliminated and the best features and practices could be incorporated to the proposed system. Thus creating a more effective and efficient product.

2.3.1 Dragon Naturally Speaking

“Dragon Naturally Speaking” is a Speech recognition software for all the applications in the computer. It basically allows the users to launch and navigate through applications using voice commands. It is not a software created specifically for Emails, but it somewhat has support for Emails.

- Dragon Naturally Speaking support for Emails

This software allows the users to dictate and sometimes even offers a few email-specific navigations using voice for some Email Clients. However Dragon Naturally Speaking is not capable of providing Navigation support for Web-based Email Clients such as Gmail, Hotmail, and Yahoo etc. It allows the user to only dictate. It cannot perform email-specific navigations such as invoking button clicks (select new, select inbox etc.) and navigating (select subject, select body etc.) to different fields using voice commands. Therefore if the user wants to navigate to different fields or buttons, the user needs to keep voicing-in the command ‘tab’. Hence it is not a complete solution offered to users.

- Evaluation

Dragon Naturally Speaking is not capable for providing full navigation support for Web-based Email Clients. The approach followed for navigation by this system is not satisfactory. Users would rather prefer having a way to directly navigate to a specific button or field without having to navigate using the ‘tab’ keyword. It would be user friendly, convenient and help users to save time.

2.3.2 Chrome 11 Speech Recognition

The chrome 11 beta version Browser is added with a Speech Recognition feature that allows the user to voice-in text into input textboxes. Software Engineer ‘Satish Sampath’ at Google Inc. explains about the experimental speech API on the chrome blog (2011). Users have the ability to download the Browser to test it and Developers have the opportunity to download and test the speech API.

To use this feature users need to click on the ‘microphone icon’ which appears at the end of the input textbox and speak input the computer’s microphone. The recorded audio is then sent to speech servers for transcription, after which the text is displayed in the textbox. A demo of this is available at the HTML5 Rocks site (<http://slides.html5rocks.com/#speech-input>) which is hosted by Google.

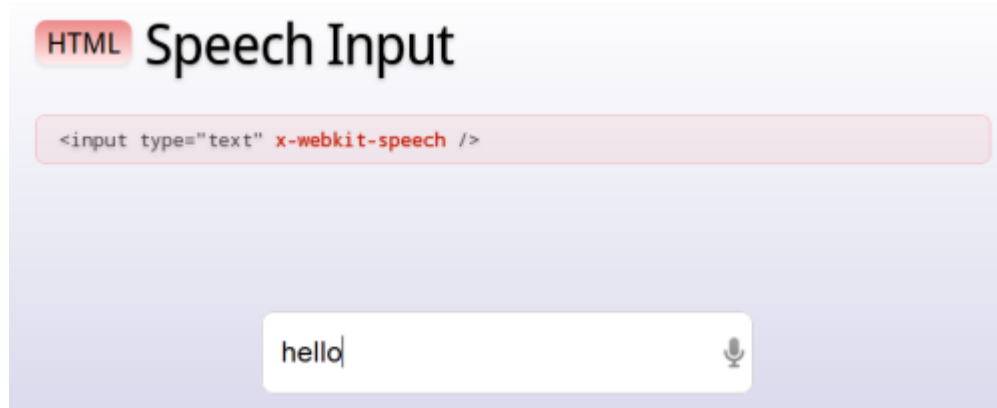


Figure 2: Chrome 11 speech input – Google Chrome, n.d

As per now the chrome 11 beta version Browser supports only dictation. To activate Speech Recognition for dictation the user needs to click on the microphone icon which displays at the end of field/textbox.

- Evaluation

Chrome 11 Speech Recognition requires the user to click on the microphone icon to activate Speech Recognition. Having to click on the microphone icon each time is a hassle. Also it does not eliminate the use of key board and mouse.

In addition, Chrome 11 Speech Recognition supports only dictation; therefore there is no way for the user to navigate throughout the application using voice commands. In

conclusion it is clear that Chrome 11 Speech Recognition fails to provide fully voice driven support and since this is still in the experimental stage the reliability and accuracy also cannot be guaranteed as yet.

2.3.3 Windows Speech Recognition

Windows Speech Recognition is an application which is shipped-in with Windows Vista and Windows 7. Windows Speech Recognition is a software that offers Speech Recognition support for all applications on a computer. It offers a common set of commands that could be used across all available applications. It offers both dictation and navigation support, but the extent to which it supports navigation varies from application to application.

- Windows Speech Recognition support for Web-based Emails Clients

Windows Speech Recognition supports dictation for Web-based email clients such as Gmail, Yahoo, Hotmail etc, but does not support direct navigation support. The users cannot direct navigate to required fields and buttons.

- Evaluation

Windows Speech Recognition is not capable of providing full voice command driven navigation support for Web-based Email Clients. Therefore for those users who wish to access web-based email clients on voice commands, Windows Speech Recognition is certainly not the answer.

2.3.4 Siri

Siri is an intelligent personal assistant and knowledge navigator which works as an application for Apple's iOS (Wikipedia, 2011). It is an application that comes with iPhone 4S, iPhone 5, iPad (3rd generation), and iPod touch (5th generation). To access Siri, users need to have Internet access but the service may not be available in all areas. Siri supports both dictation and navigation, but the extent to which how much support varies from application to application.

- Windows Speech Recognition support Emails
Siri allows users to dictate emails as well as to navigate. It can add contacts, navigate to different fields and then allows the user to dictate in those fields too.



Figure 3: Dictate Email using Siri – media.wiley.com, n.d

- Evaluation – however Siri is limited to the iOS with the above mentioned devices. It does not support desktop and laptop users.

2.3 Technical Investigation

2.3.1 Speech Recognition

There are many decisions to be made in terms of Speech Recognition before implementing a system based on Speech Recognition. Below described are the factors considered and evaluation for each factor.

2.3.1.1 Discrete Speech Recognition VS Continuous Speech Recognition

- Discrete Speech Recognition

Discrete Speech Recognition is ‘Recognition based on words spoken slowly and interrupted by a few seconds of silence between each word (library.ahima.org, n.d). Implementation of such a system is comparatively easy because it recognizes each word at a time. But this in-turn requires the user to pause between words when dictating since it recognizes word by word. From the users point of view this seems to be less user friendly since the user cannot fluently dictate sentences without having to pause after each word.

- Continuous Speech Recognition

Continuous Speech Recognition is Recognition based on natural flow of words normally used in spoken language (library.ahima.org, n.d) .Continuous Speech Recognition System allows users to fluently dictate sentences without having to pause between sentences. But from implementation point of view this a more complex approach since the user will speaker in much faster rates and there are multiple words to be recognized.

- Evaluation

Though Discrete Speech Recognition systems are easier to implement compared to Continuous Speech Recognition Systems, from the user’s point of view the more preferred system would be a Continuous Speech Recognition system.

The Voice Driven Email Client needs to perform a lot of speech recognition since it supports navigation support as well as dictation support. It is especially important to have Continuous Speech Recognition in this system given that the users need to dictate emails. Users cannot afford to pause between each word when dictating long emails. Discrete Speech Recognition consumes a large amount of time, therefore is not an appropriate approach for implementing the Voice Driven Email Client.

2.3.1.2 Speaker-Dependent Speech Recognition VS Speaker-Independent Speech Recognition

- Speaker-Dependent Speech Recognition

Speaker-Dependent Speech Recognition Systems are systems that are trained to be used by a single person. The system analyzes and learns the unique characteristics of the person's voice. It will not work for other users with high accuracy.

- Speaker-Independent Speech Recognition

Speaker-Independent Speech Recognition Systems are systems that recognize the user's voice regardless of the person speaking. Speaker-Independent systems do not involve any training. Generally Speaker-Independent Speech Recognitions Systems are less accurate than Speaker-Dependent Speech Recognition Systems.

- Evaluation

It is clearly better to have systems that are Speaker-Independent. It will allow more users to use the system. But a downside of Speaker-Independent Recognition systems are as discussed above is the reduced accuracy. When compensating between the level of accuracy of a system and the number of users who can use it, the level of accuracy is undoubtedly the choice.

The Voice Driven Email Client is locally installed on a computer. Therefore the user has the ability to train the system before using it. Speaker-Independent Recognition is usually chosen on systems which will be used by large number of users and that cannot be trained before using the system. Example: telephone applications. Therefore the best choice for the Voice Driven Email Client would be Speaker-Dependent Speech Recognition.

2.3.1.3 Limited Vocabulary VS Large Vocabulary

- **Limited Vocabulary**
Maintaining Limited Vocabularies in Speech Recognition Systems helps to achieve higher rates of accuracy in recognition. But on the other hand this would limit the number of words the user can speak.
- **Large Vocabulary**
Large Vocabularies are essentially important in Speech Recognition Systems that allow dictation. If a system only supports navigation, it will most probably have a set of key words. But in systems that support dictation the system vocabulary cannot be limited to a few words. The drawback of Large Vocabularies is the increment in error rates.
- **Evaluation**
The 'Speech Recognition' article in the 'learnartificialneuralnetworks.com' website states that 10 digits "zero" to "nine" can be recognized essentially perfectly, but vocabulary sizes of 200, 5000, or 100000 may have error rates of 3%, 7%, or 45%. Evidently large vocabularies increase error rates, but for the Voice Driven Email Client having a large vocabulary is important since it supports dictation. It is impractical for users to dictate with a small vocabulary.

2.3.1.4 Commands, Dictation and Natural Language Processing

- **Commands**

Commands are the keywords used for navigating within the system. A separate command dictionary in xml is maintained for this purpose. Further homophones are taken into consideration in commands.

Example:

```
<command>  
  <original>select to</original>  
  <homophone1>select too</homophone1>  
  <homophone2>select two</homophone2>  
</command>
```

- **Dictation**

Dictation vocabulary needs to be large. For this purpose the default dictionary of the Microsoft Speech API is used. Homophones have not been taken into consideration and processed due to the limited time.

- **Natural Language Processing**

Natural Language Processing is not incorporated in this system. It is suggested as a future enhancement.

2.3.2 Speech Recognition APIs

There are many Speech Recognition Application Programming interfaces available. Therefore to select the most appropriate Speech API for developing the system, it is important to compare the available APIs. Below are few of the APIs that were considered before selecting the right speech API for developing the Speech Driven Email Client.

2.3.2.1 Microsoft Speech API (SAPI 5.4)

SAPI is an API developed by Microsoft. There are two basic types of SAPI engines made available by Microsoft for developers.

- Speech Synthesis - converts text-to-speech (TTS). It takes in text strings and converts it to spoken audio using synthetic voices.
- Speech Recognition – converts speech-to-text. It takes in human voice and converts it to text strings.

SAPI is a high-level interface that lies between the Application and Speech engines (Speech Synthesis and Speech Recognition). An overview of the SAPI Architecture is shown in the diagram below.

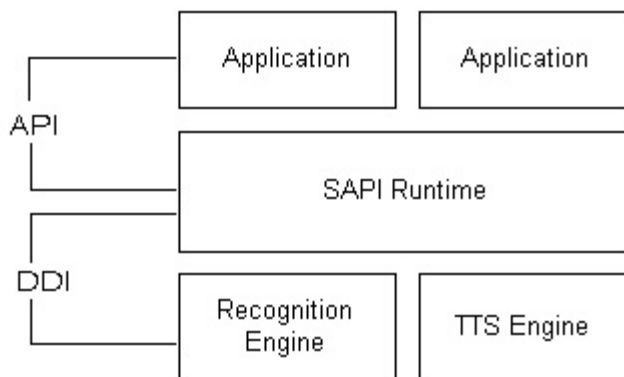


Figure 4: SAPI Architecture – microsoft.com, n.d

SAPI is middleware that provides an Application Programming Interface (API) and Device Driver Interface (DDI). Developers access this API using the *System.Speech** namespace which calls the Sapi.dll in turn. It is an API targeted mainly at C/C++/C# developers.

2.3.2.2 Nuance Dragon API (Dragon Naturally Speaking 11)

Dragon API, provided by Nuance makes it possible to integrate Speech Recognition to applications. It can be used when developing from scratch or when adding Speech Recognition to existing applications.

According to Dragon SDK datasheet 11,

- Accuracy - it can achieve accuracy rates up to 99% in Speech Recognition.
- Dictation - it allows user to dictate up to 160 words per minute.
- Vocabulary - it contains a language model and files of the words users can speak. Further it allows developers to create custom vocabularies.
- Text to Speech – it can convert text to speech using the Dragon Naturally Speaking Speech Synthesis Engine or any SAPI 4 compliant TTS engine.
- Languages –it supports English, French, German, Italian, Spanish and Dutch

In addition Dragon API provides Continuous Speech Recognition, and Speaker Independency.

2.3.2.3 Sphinx 4

Sphinx is a Speech Recognition system written purely using the Java programming language. Below are few features of Sphinx 4.

- Capable of recognizing of both Discrete and Continuous Speech Recognition
- Works on Solaris, Linux, Mac OS X and Win32 Operating Systems.
- Requires additional software to be installed;
 - Java SE 6 development Kit or higher
 - Ant 1.6.0 or higher
 - Subversion (svn)
- Comprises of complex installation process. Below are the steps in a very high level
 - Requires downloading two packages which include the binary and source sphinx4 files.
 - Before building Sphinx JSAPI 1.0 needs to be installed
 - Next run Ant
 - Set up IDE (Eclipse, Netbeans etc)
- The architecture of Sphinx 4 is such that it works on 3 main modes, FrontEnd, Decoder and Linguist. The concepts are show in detail in the diagram below.

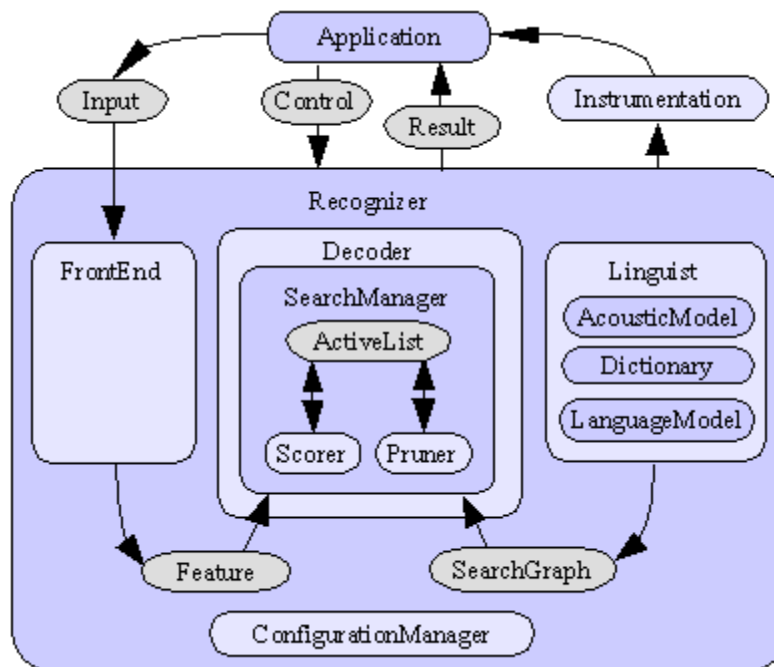


Figure 5: Sphinx 4 Architecture – sourceforge.net, n.d

2.3.2.4 Evaluation of Speech APIs

Microsoft Speech API and Nuance Dragon SDK are limited to Windows Operating Systems, where as Sphinx4 works on multiple Operating Systems which include Solaris, Linux, Mac OS X and Windows.

Microsoft Speech API and Nuance Dragon SDK comprises of its own large vocabularies where as Sphinx4 requires creating its' own dictionaries.

Installation and implement process of Sphinx4 is very complex where as Microsoft Speech API and Nuance Dragon SDK are comparatively easier to install and implement.

According to the above evaluation the Microsoft Speech API and Nuance Dragon SDK clearly are better choice compared to Sphinx4. However Nuance Dragon SDK comes with a purchase cost, therefore the Microsoft Speech API was selected as the Speech API to be used for developing the Voice Driven Email Client. SAPI is a very powerful, reliable and widely used Speech API, thus the undoubted choice.

2.3.3 Programming Language

Selection of the most appropriate Programming Language is one of main areas considered prior to developing a system. The programming Languages that were taken into consideration in this case were C++, C# and Java. The main areas that were judge are,

- Availability of speech libraries
 - Support for SAPI
- Availability of email libraries
- Support for Graphical User Interface (GUI)
- Object oriented
- Platform dependency

	C#	C++	Java
Availability of speech libraries	✓	✓	✓
Support for SAPI	✓	✓	
Availability of email libraries	✓	✓	✓
Support for GUI	✓		✓
Object Oriented	✓	✓	✓
Platform Independent			✓

The obvious choice for initially picking C++, C# and Java were availability of Speech APIs (Speech Libraries) and the fact that they are Object Oriented Languages.

Java is a very powerful language that is platform independent. However the Speech API (Sphinx4) to be used with Java is not the best choice of Speech API to be used in this case.

C++ supports the Microsoft SAPI but does not support Graphical user interfaces.

Therefore the selected Programming Language is C#. It is object oriented, support Microsoft SAPI and supports implementing Graphical User Interfaces.

2.3.4 POP vs. IMAP

These are two Protocols used to connect to Web-based Email Clients

POP3:

- Emails are downloaded into the local machine before displaying
- One way communication – changes made will not affect the server
Example: after reading an email from client, when you log into Gmail it will remain the same as unread

IMAP4:

- Two way communication – syncs with Gmail, made changes from client reflects in Gmail

However due to difficulties in implementing IMAP4, POP3 was used for the system implementation.

2.3.5 Gmail Plug-in

The possibility of implementing a Gmail Plug-in was considered. However due to technical limitations and lack of technical knowledge regarding Plug-in made it unable to develop a Gmail Plug-in for Chrome. The research carried out is explained below briefly.

- Gmail contextual gadgets:
Contextual gadgets are applications which are displayed at the bottom of each email. These are triggered by contextual clues from the email subject, recipient, sender etc. For example: The ‘YouTube Contextual Gadget’ provided by Gmail. If the email contains a YouTube link, a clickable thumbnail appears at the bottom of the email.

However this feature is insufficient for the tasks needed to be performed in the voice driven email client. This gadget has control only over emails, it cannot manage navigation controls. (Buttons etc)

- Gmail Services
Gmail Services provide access to the Gmail Inbox and Labels. However it again doesn’t allow managing navigation controls. Further it uses its own scripting language and cannot be linked with SAPI.

Given below is an example of its scripting language. This function returns the unread email count.

```
function getUnreadMsgCount()
{
    var emailAddress = Session.getActiveUser().getEmail();

    GmailApp.sendEmail(emailAddress,'subject: unread EmailCount', 'body:
the number of unread emails are = ' + GmailApp.getInboxUnreadCount());
}
```

- Google Data API
Google data API can be downloaded and used to access certain elements of Gmail. For example: the ‘Gmail Inbox Feed’. Yet again this doesn’t allow managing navigation controls.

Chapter 3 Requirement Specification

3.1 Chapter Overview

Identifying requirements is a very important aspect of a system. The entire system will depend on the requirements recognized at this stage. This section captures the main requirements of the system which include Functional, Non-Functional and Resource Requirements.

3.2 Functional Requirements

Two versions of the system will be available.

- Version 1 (My interface) – this version of the system will have its own interface
- Version 2 (Gmail interface) – this version of the system will have the Gmail interface. The Gmail interface will be loaded inside the voice driven email client in a web browser.

3.2.1 Functional Requirements for version 1 (own Interface)



Note: Following Gmail Settings should be set before proceeding to use the system

- * To stop Gmail messages with the same subject from grouping together following the below step;
 - Go to ‘General’ tab of the Gmail settings
 - Select radio button ‘Conversation view off’
- * To move Gmail’s copy of an email from the inbox when an email is deleted via the voice driven email client
 - Go to ‘Forwarding and POP/IMAP ’ tab of the Gmail settings
 - For ‘When messages are accessed with POP’ select archive Gmail’s copy / delete Gmail’s copy (default - keep Gmail’s copy in the inbox)

- System should behave as a separate interface

The voice driven email client should behave as a separate interface that allows the user to connect to an existing web email account (in this case a Gmail account since the scope is limited to Gmail) and perform all actions using voice commands. (Not directly through the browser).

- Login using voice commands

The system should start up with a splash screen which will stay for 5 seconds. Once the system is loaded, Login page/form should be displayed. If Speech Recognition is active the system should display  (green microphone icon) at the top of the page, else if it is inactive the system should display  (red microphone icon) at the top of the page. The login screen should include the following fields;

- Username – field to enter valid username of a web client



- Password – field to enter the corresponding valid password for the web client
- Web Email client – field to enter the web client name (Gmail)

The user should be capable of using voice commands to fill each field. The system should capture the login credentials from a predefined xml document. If credentials are accurate the system should allow the user to login, else error message should be displayed.

Reason for having credentials in an xml document without allowing user to dictate:

An email address contains special characters and numbers which make it ambiguous to capture accurately.

For example: test_one@abc.com

- When the user dictates ‘underscore’ the system wouldn’t know whether the user meant ‘underscore’ or ‘_’.
 - When the user dictates ‘one’ the system wouldn’t know whether the user meant ‘one or ‘1’.
- Display own interface with Speak Recognition icon
On successful login the system should display the main page/form of the system along with the Speak Recognition icon. If Speech Recognition is active it should display  else it should display 
 - Capable of connecting to web client (Gmail) and receiving inbox emails using voice commands.
When the user commands to open the Inbox, the system should display the inbox emails with a corresponding index in-order for the user to be able to uniquely address a specific email to be opened or deleted.

Once an email is opened the user should be able use voice commands to perform the following actions:

- Reply
 - Reply All
 - Forward
- Provide interface to create new email, fill the required fields and send email using voice commands.
User should be able to navigate to the New Email Form using voice commands. Once the Form/page is launched the following actions should be able to be performed.

- Filling To Field – navigate to ‘To Field’ on command, open address book on command and add contact by voicing-in the contact index (which should be allocated at run time).
- Filling CC and BCC Fields – same approach as filling To Field
- Filling Subject Field - navigate to ‘Subject Field’ on command and dictate Subject
- Filling Body Field - navigate to ‘Body Field’ on command and dictate Body
- Adding Attachments - navigate to ‘Attachment Field’ on command, browse files and attach file using voice commands

3.2.1 Functional Requirements for version 2 (Gmail Interface)



Note: Following Gmail Settings should be set before proceeding to use the system

- * Gmail should be switched to html view from standard view
- * To stop Gmail messages with the same subject from grouping together following the below step;
 - Go to 'General' tab of the Gmail settings
 - Select radio button 'Conversation view off'

- System should behave as a separate interface

The voice driven email client should behave as a separate interface that allows the user to connect to an existing Gmail account and perform all actions using voice commands. (Not directly through the browser).

- Login using voice commands

The system should start up with a splash screen which will stay for 5 seconds. Once the system is loaded, Login page/form of Gmail should be displayed in the web browser control. If Speak Recognition is active the system should display  (green microphone icon) at the top of the page, else if it is inactive the system should display  (red microphone icon) at the top of the page. The login screen will include the following fields;

- Username – field to enter valid username of a web client
- Password – field to enter the corresponding valid password for the web client

The user should be capable of using voice commands to fill each field. The system should capture the login credentials from a predefined xml document. If credentials are accurate the system should allow the user to login, else error message should be displayed.



Reason for having credentials in an xml document without allowing user to dictate:

An email address contains special characters and numbers which make it ambiguous to capture accurately.

For example: test_one@abc.com

- When the user dictates 'underscore' the system wouldn't know whether the user meant 'underscore' or '_'.
- When the user dictates 'one' the system wouldn't know whether the user meant 'one' or '1'.

- Display Gmail interface with Speak Recognition icon

On successful login the system should display the main page of the Gmail interface along with the Speak Recognition icon. If Speak Recognition is active it should display  else it should display 

- Capable of connecting to Gmail and receiving inbox emails using voice commands. When the user commands to open the Inbox, the system should display the inbox emails with a corresponding index in-order for the user to be able to uniquely address a specific email to be opened.

Once an email is opened the user should be able use voice commands to perform the following actions:

- Reply
 - Reply All
 - Forward
 - Delete
- Provide an interface to create new email, fill the required fields and send email using voice commands. User should be able to navigate to the New Email Form using voice commands. Once the Form/page is launched the following actions should be able to be performed.
 - Filling To Field – navigate to ‘To Field’ on command, open address book on command and add contact by voicing-in the contact index (which should be allocated at run time).
 - Filling CC and BCC Fields – same approach as filling To Field
 - Filling Subject Field - navigate to ‘Subject Field’ on command and dictate Subject
 - Filling Body Field - navigate to ‘Body Field’ on command and dictate Body

3.3 Non-Functional Requirements

- **Improved Accuracy and Reliability**
The system should provide correct results in terms of Speech Recognition. The speech engine should provide accurate results and thereafter the navigations and dictations should function properly.
- **Improve Availability**
The system downtime should be almost equal to zero.
- **User friendly**
The system usage should be user friendly. The user should be able to easily locate functions to be performed and a reduced learning curve.
- **Consistent**
Consistency should be maintained throughout the system in terms of user interface design and commands used.
- **Response time**
System should respond to input with minimum delay.

3.4 Resource Requirements for Development

3.4.1 Software Requirement

- Integrated Development Environment (IDE): Microsoft Visual C# 2010 Express
- .Net framework: CLR version 4.0.30319.0
- Microsoft Speech SDK – SAPI 5.4

3.4.2 Hardware Requirement

- Desktop Machine or Laptop
- Microphone

3.4.3 Other Requirements

- Gmail Account

Chapter 4 System Design

4.1 Chapter Overview

This chapter outlines the designing phase of the system. It includes designing the system architecture using flow charts, use case diagrams and use cases.

Note – this chapter includes the system design for version 1 of the Voice Driven Email Client. In version 2 of the system, the presentation tier of the system is replaced with the Gmail interface (minor changes are made to the Application and Data Tiers).

4.2 System Architecture

The architecture used to develop the Voice driven Email Client is the ‘3-Tier Architecture’.

4.2.1 High-level Architecture Diagram

The system architecture is briefly represented in the high-level diagram shown below. The Voice Driven Email Client is a separate interface that connects to an existing web-based email client (e.g. Gmail) using voice commands. The main input for the system will be the user’s voice. The application will capture the user’s voice and convert it to text using a speech API.

The converted text will go through a series of logical processes which includes cleansing (removing of background noise etc), differentiating between commands and dictation, handling homophones, converting words to integers etc.

The two main instances the system communicates with the outside world are, to send emails and to receive emails. Listed below are the main functions performed using voice commands.

Send Emails

- Create new email
- Add contacts from address book to ‘To’ field (‘CC’/ ‘BCC’ fields if required)
- Navigate to ‘Subject’ field and dictate subject
- Navigate to ‘Body’ field and dictate subject
- Add attachments

Receive Emails

- Connect to web client and retrieve Inbox emails
- Open emails
- Delete emails
- Reply/ Reply All/ Forward emails

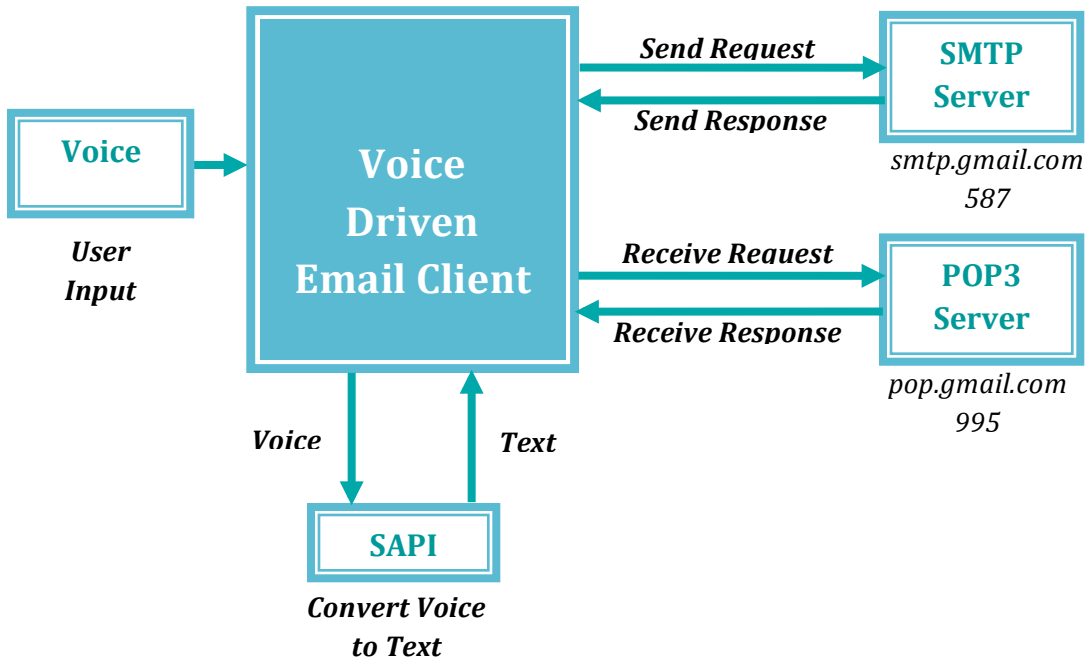


Figure 6 : High –Level System Architecture Diagram

4.2.2 Three Tier Architecture

Shown below is a high level representation of the 3-teirs.

4.2.2.1 Presentation Tier

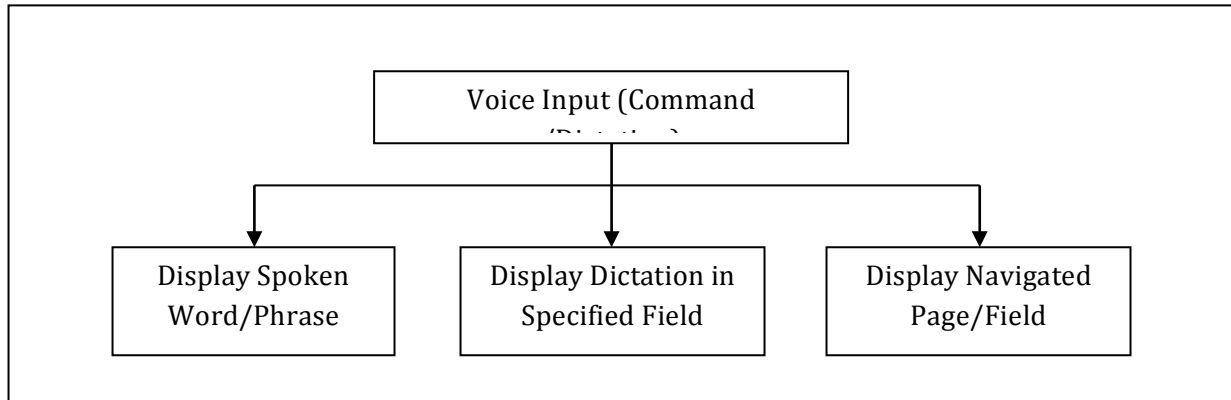


Figure 7 : Presentation Tier of the System

The presentation tier, being the topmost layer is responsible for displaying data. It is the layer that interacts with the user. It obtains input from the user and displays the output to the user.

The input from the user in this system is the user's voice. It allows the user to start speech recognition and displays functions of the recognized word/phrase. The voice input could be either a command or dictation. This layer is responsible for only acquiring the input and not processing the acquired input.

This layer is also responsible for displaying outputs. It will display all word/phrases recognized. Further if the spoken phrase is a command it will display the navigated page or field, else if the spoken phrase is a dictation, it will display the dictated phrase in the specific field.

4.2.2.2 Application Tier (Logic Tier/ Business Tier/ Data Access Tier)

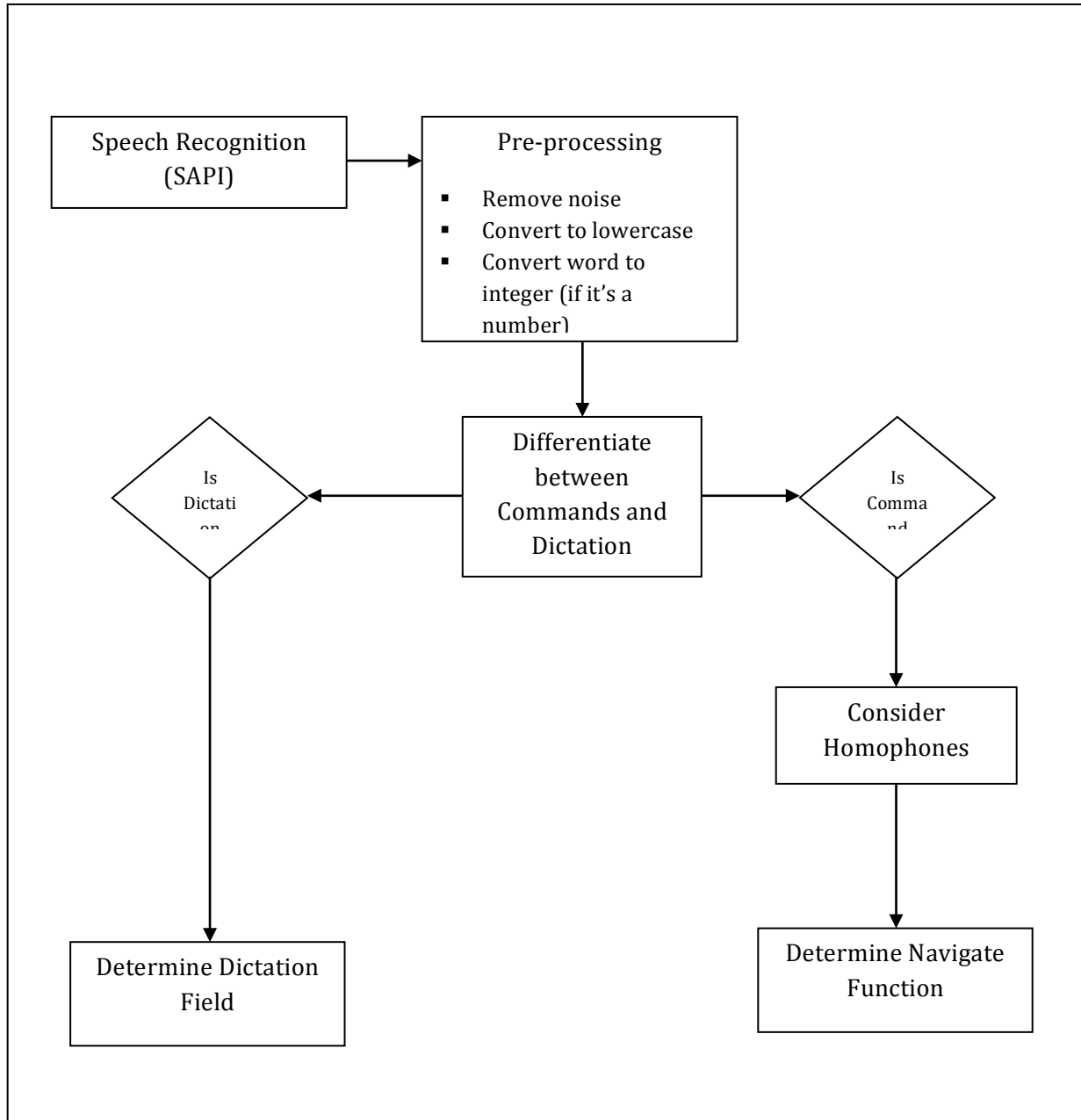


Figure 8 : Application Tier of the System

The presentation tier is the middle tier of the 3 tier architecture. It is the tier responsible for processing the data captured by the presentation tier and accessing the data in the data tier.

Below are the main processing functionalities of the application tier that pertain to the voice driven email client.

- **Speech Recognition & Pre-processing**

This layer will recognize the voice input of the user captured by the presentation tier using the speech API (SAPI). It will further process the input in-order to prepare it for further logical processing by removing noise, converting to lower case, converting words to integers (one to 1)etc.

- **Differentiate between Commands and Dictation**

Once the recognized phrase is pre-processed this tier will then identify whether the input is a command or dictation by mapping the phrase against the command set.

When mapping it against the command set, homophones are taken into consideration. For example the command 'select to' could be recognized by the system as either 'select two' or 'select too'. The command set keeps track of homophones.

- **Determining the Dictation Field and Navigation Function**

Finally this tier will determine what needs to be done with the recognized phrase. It will go through the logical processing and determine the specific function and pass it to the presentation tier to be displayed.

4.2.2.3 Data Tier

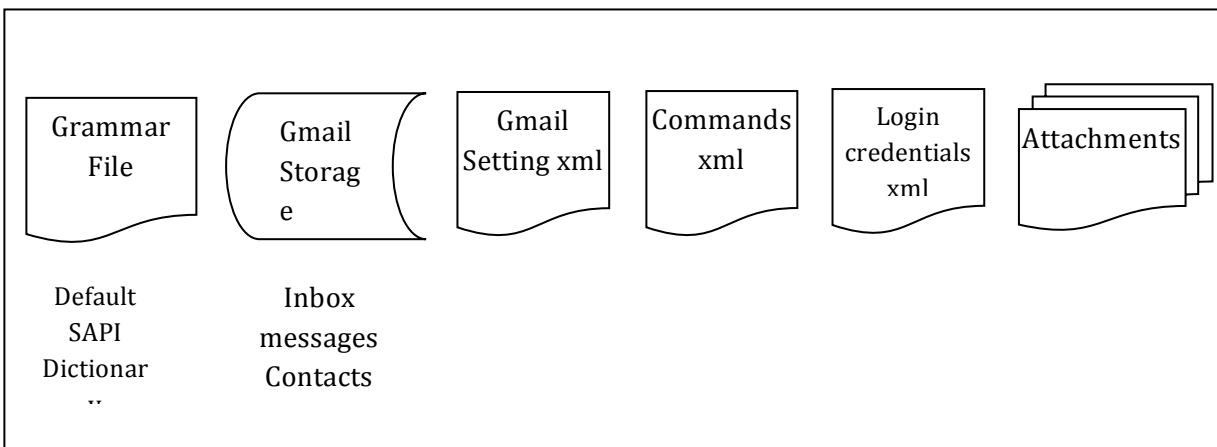


Figure 9 : Data Tier of the System

The Data Tier is the lowest level of the 3-tier architecture. This tier is responsible for holding and retrieving data of a system. The Data tier of the voice driven email client holds and retrieves the following data,

- Grammar File – the default grammar file of the Speech API (API). In the speech recognition process, this is the dictionary used for mapping the grammar.
- Login Credential XML – the Login Credential XML file holds the username and password of the user’s web email.
- Gmail settings XML – contains the Gmail SMTP Host name, SMTP port number, POP3 Host name and POP3 port number
- Commands XML – the Commands XML file holds the entire command set used in the voice driven email client along with the homophones.
- Attachments – the attachments folder holds the attachments the user requires to attach to emails
- Gmail Storage – the storage to which the system connects to retrieve data from the web mail.

4.3 Use Case Diagram

Below shown is the use case diagram of the Voice Driven Email Client

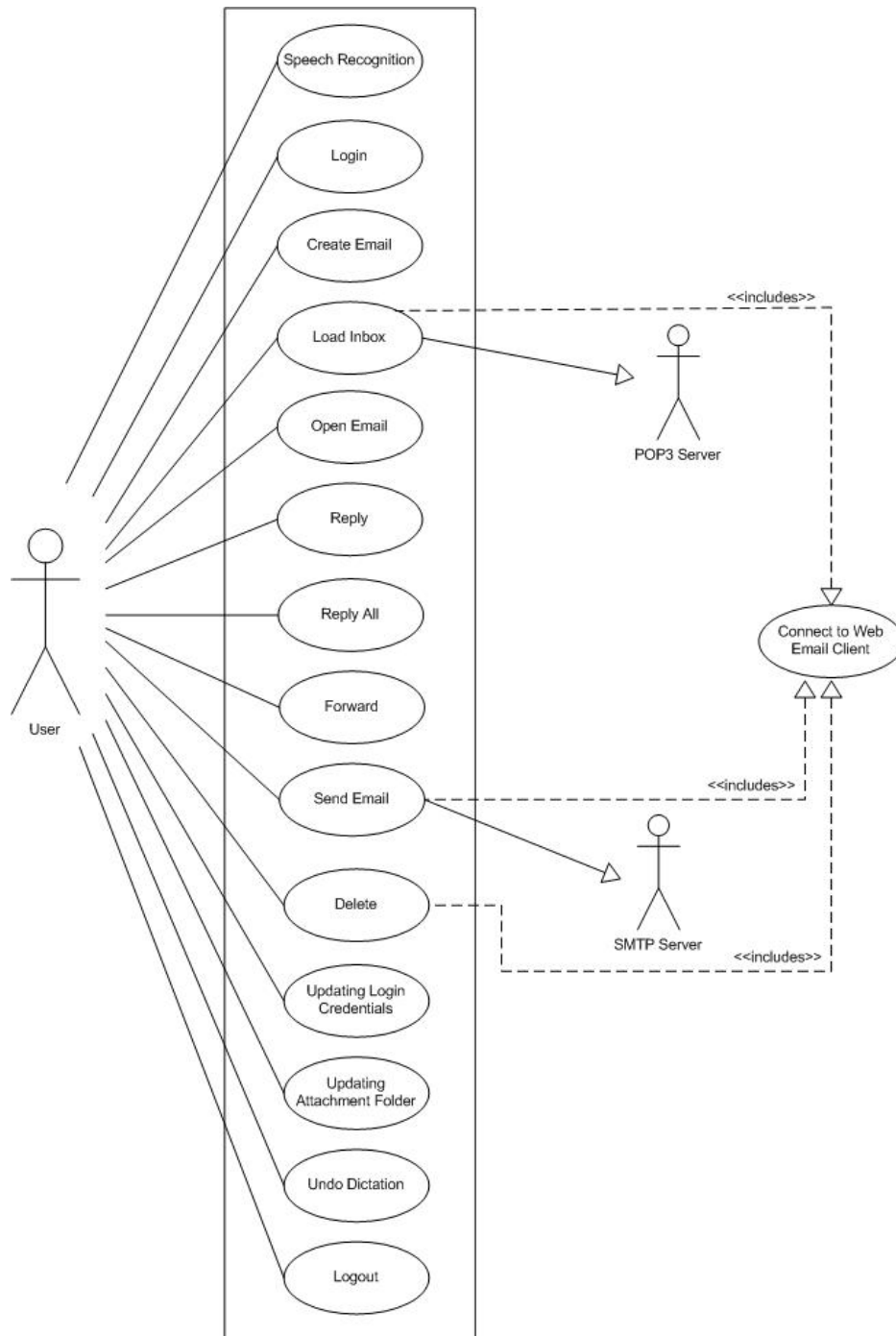


Figure 10 : System Use Case Diagram

4.4 Use Cases

Use Case ID	001
Use Case Name	Voice Driven Email Client Login
Use Case Description	Allows user to successfully log into the voice driven email system.
Actors	User
Pre-Conditions	<ol style="list-style-type: none"> 1. User should have a web-email account (Gmail) with valid login credentials. 2. User will have to update their username, password and web-client name which are saved on the 'LoginCredentials.xml' file prior to logging on to the system. (Refer to Use Case ID 019) 3. POP3 host and POP3 port need to be saved on the 'GmailConnectionDetails.xml' file. 4. User should have access to the Internet. 5. User should have a microphone connected to their computer.
Flow	<ol style="list-style-type: none"> 1. User launches system (Voice Driven Email Client) by clicking icon on desktop. 2. System starts up with a Splash Screen which displays for 5 seconds introducing the user to the system. 3. System then navigates to Login Form/Page. <ol style="list-style-type: none"> a. On the top left corner of the screen, the speech recognition icon (red icon) will be visible to the user. The system will display the message 'Start Speaking' next to the icon when the system activates Speech Recognition and is ready to receive input. (Refer Use Case ID 002). b. The Login page will have three text boxes on the bottom right hand corner: username, password and web-email client. 4. The user can fill the username, password and web-email client text boxes by delivering the following voice commands respectively: <ul style="list-style-type: none"> 'Include username' 'Include password' 'Include Client' 5. The text boxes will be filled by system using the 'LoginCredentials.xml' file (Refer to pre-condition 2) 6. To sign in to the system, the user will deliver the command ' Select Sign in'. The user will be directed to the 'EmailClientForm'.
Alternative Flow	<ol style="list-style-type: none"> ALT 01. The user will minimize the system (Refer Use Case ID 013) 02. The user will maximize the system (Refer Use Case ID 014) 03. the username is different to what is displayed:

	<p>i. a. User will change credentials in xml file (Refer Use Case ID 019)</p> <p>b. Repeat flow from step 1-6.</p> <p>ii. The user will type new username, password and web email client in login screen.</p>
Exception Flow	<p>EX 01.</p> <p>a. Systems prompts error message ‘Invalid Login Details Please Try Again’ when user signs in.</p> <p>b. User minimizes system (Refer Use Case ID 013)</p> <p>c. User stores valid credentials in xml file (Refer Use Case ID 019)</p> <p>d. User restores system by using Clicking on system from task bar</p> <p>e. Repeat from step 3 of main flow.</p> <p>02. The user Exits the system using the browser navigation buttons (Refer Use Case ID 015)</p> <p>03. The user Exits the system using the button provided by stating ‘Select Close’.</p>
Post Conditions	System navigates to main page (EmailClientForm)

Use Case ID	002
Use Case Name	Behavior of Speech Recognition Icon
Use Case Description	Details the behavior of the Speech Recognition Icon in the system that can be seen on every screen.
Actors	Voice Driven Email System
Pre-Conditions	<p>The Speech Recognition icon is found on the top left hand corner of every screen along with the last dictated command/ word/phrase that has been input through the microphone.</p> <p>User has opened the system.</p> <p>User should have a microphone connected to their computer.</p>
Flow	<ol style="list-style-type: none"> 1. The speech recognition icon will initially be red and the text displayed will be the last voice command, word or phrase input by the user. 2. As the user starts to speak, the icon will turn to Green to signify voice entry. 3. The voice command will be displayed in text by the side of the icon. 4. The icon will turn to red and the input command will be displayed.
Alternative Flow	ALT 01. The text on the login screen will read ‘Start Speaking’ next to the red speech recognition icon when the system activates speech recognition.

	02. The icon will turn green for two seconds when an audible sound is detected from a microphone. The icon will turn red after the two seconds are over.
Exception Flow	-
Post Conditions	The icon will turn to Red after the required voice input has been given.

Use Case ID	003
Use Case Name	Navigating the Email Client Form
Use Case Description	This use case describes the options available to the user from the Email Client Form
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Email Client Form. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.
Flow	1. a. The user states the command ‘Select New’ b. User will be directed to the new email page within the Email Client Form browser (Refer Use case 004). 2. a. User states the command ‘ Select Inbox’ b. User will be directed to the Inbox page within the Email Client Form browser (Refer Use case 007).
Alternative Flow	ALT 01. The user will minimize the system (Refer Use Case ID 013) 02. The user will maximize the system (Refer Use Case ID 014)
Exception Flow	The user Exits the system using the browser navigation buttons (Refer Use Case ID 015)
Post Conditions	The user will be able to create a new email. The user will be able to browse through their email inbox.

Use Case ID	004
--------------------	------------

Use Case Name	Creating and Sending a New Email
Use Case Description	This use case describes the process of creating a new email and sending it after the user selects 'new' using the voice command.
Actors	User
Pre-Conditions	User is logged into the system and navigated to the Email Client Form. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet User should have a microphone connected to their computer. User has stated the command ' Select New'
Flow	<ol style="list-style-type: none"> 1. System loads NewEmailForm inside the panel in the main page 2. User fills fields by stating the following commands <ol style="list-style-type: none"> a. User states 'Select To' (Refer use case ID 005) b. User states ' Select CC' (Optional) (Refer Use Case ID 005) c. User states ' Select BCC' (Optional) (Refer Use Case ID 005) d. User states 'Select Subject' (Optional) <ol style="list-style-type: none"> i. System sets focus to subject field ii. User dictates subject (example: this is the subject) iii. System displays the dictated phrase in the subject field iv. The Voice Recognition Icon will display the dictated text (Refer use case ID 002) e. User states 'Select Attachment' (Optional) (Refer use case ID 006) f. User states 'Select Body' <ol style="list-style-type: none"> i. System sets focus to body field ii. User dictates text for body (example: this is the body) iii. System displays the dictated phrase in the body field iv. The Voice Recognition Icon will display the dictated text (Refer use case ID 002) 3. User states command 'select send' <ol style="list-style-type: none"> 3.5 The email will be sent 3.6 A pop up message stating 'Email Sent' will be displayed to the user. <ol style="list-style-type: none"> i. minimize the message (Refer Use Case ID 016) ii. Maximize the message (Refer Use Case ID 017) iii. Close the message (Refer Use Case ID 018)
Alternative Flow	<p>ALT 01</p> <ol style="list-style-type: none"> i. The user can navigate away from the new email page by stating ' Select Inbox'

	<ul style="list-style-type: none"> ii. The user will be directed to the Inbox panel inside the main page. (Refer Use Case ID 007) <p>ALT 02</p> <p>A</p> <ul style="list-style-type: none"> i. The user will delete the last dictated word/phrase in the ‘Subject’ field (Refer Use Case ID 025) ii. Flow returns to 2.d <p>B</p> <ul style="list-style-type: none"> i. The user will delete the last dictated word/phrase in the ‘Body’ field (Refer Use Case ID 025) ii. Flow returns to 2.f <p>ALT 03</p> <ul style="list-style-type: none"> i. The user will minimize the system (Refer Use Case ID 013) ii. The user will maximize the system (Refer Use Case ID 014)
Exception Flow	<p>EX 01. The user Closes the system using the relevant voice command (Refer Use Case ID 015)</p> <p>02.</p> <ul style="list-style-type: none"> a. The user does not fill in the ‘To’ Field in the NewEmailForm. b. An error message is displayed to the user - ‘The Parameter ‘addresses’ cannot be an empty string’. c. The user can <ul style="list-style-type: none"> i. minimize the error message (Refer Use Case ID 016) ii. Maximize the error message (Refer Use Case ID 017) iii. Close the error message (Refer Use Case ID 018)
Post Conditions	User Successfully sends an email.

Use Case ID	005
Use Case Name	Select Contacts for the ‘To’, ‘CC’ and ‘BCC’ Fields
Use Case Description	This use case describes how the user is able to select contacts from the contact form to be included in the To, CC and BCC fields.
Actors	User
Pre-Conditions	<p>User is logged into the system and navigated to the new email page.</p> <p>Speech Recognition Icon on the screen is active (Refer Use Case ID 002).</p> <p>User should have access to the Internet</p> <p>User should have a microphone connected to their computer.</p>

	<p>User should have email contacts added in Gmail account. Each Contact (even with multiple email addresses) will have a unique Email ID. User should have stated 'Select To' User should have stated 'Select CC' User should have stated 'Select BCC'</p>
Flow	<ol style="list-style-type: none"> 1. User will state 'Select Contact(s)' 2. A contact Form will be displayed to the user with the contacts retrieved from their Gmail account. The Contact form will display the email ID, Name and Address. 3. The user will state 'Include (email ID)' stating the required email ID. 4. The user can exit the contacts form by stating 'Exit Form' 5. User will be returned to the Create New Email Page (Refer Use Case ID 004) where the 'To' field will be updated with the Contact.
Alternative Flow	ALT 01. The contact form can be maximized and minimized (Refer Use Case ID 016 and 017 respectively).
Exception Flow	Ex 01. The user may exit the form by stating 'Exit Form' and return to the New Email Page (Refer Use Case ID 018)
Post Conditions	Contact(s) have been successfully added to the To, CC and BCC fields.

Use Case ID	006
Use Case Name	Select Attachment for Attachment Field
Use Case Description	This use case describes the process of adding an attachment to an email to be sent
Actors	User
Pre-Conditions	<p>User is logged into the system and navigated to the new email page Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet User should have a microphone connected to their computer. User should have updated Attachment Folder with required attachment. (Refer Use Case ID 024) Each attachment will have a unique Attachment ID. User should have stated 'Select Attachment(s)'.</p>
Flow	<ol style="list-style-type: none"> 1. User will state 'Select Attachment(s)'

	<ol style="list-style-type: none"> 2. The Attachment Form will be displayed to the user with the attachments retrieved from the pre-defined folder. The Attachment form will display the Attachment ID, Attachment Name and Attachment Path. 3. The user will state 'Attach (Attachment ID)' stating the required Attachment ID. (E.g.:- 'Attach Two') 4. The user can exit the contacts form by stating 'Exit Form' 5. User will be returned to the Create New Email Page (Refer Use Case ID 004) where the attachment will be displayed in the Attachment Field.
Alternative Flow	ALT 01. The Attachment form can be maximized and minimized (Refer Use Case ID 016 and 017 respectively).
Exception Flow	Ex 01. The user may exit the form by stating 'Exit Form' and return to the New Email Page (Refer Use Case ID 018)
Post Conditions	An Attachment has been successfully added to the 'Attachment' Field.

Use Case ID	007
Use Case Name	Inbox page functionality
Use Case Description	This use case describes what functionality is available to the user in the Inbox page
Actors	User
Pre-Conditions	<p>User is logged into the system. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet User should have a microphone connected to their computer. User should have stated 'Select Inbox' to be navigated to this page</p>
Flow	<ol style="list-style-type: none"> 1. The inbox page is displayed inside the panel in the main page. 2. The user is shown a table (on to the left) with a list of emails stating the Email ID, Sender, Subject, Body and Date arranged in order of the date. On the right side of the screen is a preview box showing a preview of the email (Refer Use Case ID 008). 3. The user is able to <ol style="list-style-type: none"> a. Open an email (Refer Use Case ID 008) b. Delete an email (Refer Use Case ID 009)

	<p>c. Reply an email (Refer Use Case ID 010)</p> <p>d. Reply All on the email (Refer Use Case ID 011)</p> <p>e. Forward an email (Refer Use Case ID 012)</p> <p>4. The user will be able to use the scroll bars to navigate up and down the email panel.</p>
Alternative Flow	<p>ALT 01</p> <p>j. The user can navigate away from the Inbox page by stating ‘ Select New’</p> <p>iii. The user will be directed to the New email panel inside the main page. (Refer Use Case ID 004)</p> <p>ALT 02</p> <p>i. The user will minimize the system (Refer Use Case ID 013)</p> <p>ii. The user will maximize the system (Refer Use Case ID 014)</p>
Exception Flow	EX 01. The user Closes the system using the relevant voice command (Refer Use Case ID 015)
Post Conditions	The user will be able to complete a range of tasks such as opening, deleting, replying and forwarding an email in the inbox page.

Use Case ID	008
Use Case Name	Opening an Email
Use Case Description	This use case describes how the user is able to open an email to view it in the preview box on the Inbox Page.
Actors	User
Pre-Conditions	<p>User is successfully logged into the system and navigated to Inbox page.</p> <p>User should have stated ‘Select Inbox’ to be navigated to this page</p> <p>Speech Recognition Icon on the screen is active (Refer Use Case ID 002).</p> <p>User should have access to the Internet.</p> <p>User should have a microphone connected to their computer.</p> <p>The users email inbox needs to have emails.</p> <p>Each Email has a unique ID</p>
Flow	<ol style="list-style-type: none"> 1. The user states the command ‘Open (Email ID)’ (e.g.:- ‘Open Three’) 2. The system displays the Sender, Subject, Date and Time and the whole email in the preview box on the right side of the screen. 3. The user can use the scroll bars to read the email.
Alternative Flow	ALT 01. The user will minimize the system (Refer Use Case

	ID 013) 02. The user will maximize the system (Refer Use Case ID 014)
Exception Flow	EX 01. The user Closes the system using the relevant voice command (Refer Use Case ID 015)
Post Conditions	The user will be able to open and browse through their emails

Use Case ID	009
Use Case Name	Deleting an Email
Use Case Description	This use case describes how the user is able to delete a specific Email.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to Inbox page. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer. User should have stated 'Select Inbox' to be navigated to this page Each Email has a unique ID
Flow	<ol style="list-style-type: none"> 1. The user states the command 'Delete (Email ID)' (e.g.:- 'Delete One') 2. The system displays a message to the user stating 'Message deleted Successfully, Re-loading Inbox' 3. The user has to close the form (Refer Use Case ID 018) 4. The user is directed back to the inbox page where the unique ID's for the emails are regenerated and the deleted email is taken off from the table.
Alternative Flow	ALT 01. The user will minimize the system (Refer Use Case ID 013) 02. The user will maximize the system (Refer Use Case ID 014)
Exception Flow	EX 01. The user Closes the system using the relevant voice command (Refer Use Case ID 015)
Post Conditions	The user will be to delete the selected email.

Use Case ID	010
--------------------	------------

Use Case Name	Replying an Email
Use Case Description	This use case describes the method in which a user can reply to an email
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Inbox Page. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer. User has already opened an email (Refer Use Case ID 008)
Flow	<ol style="list-style-type: none"> 1. The user states the command 'Email Reply' 2. User will be directed to the new email page within the Email Client Form browser (Refer Use Case ID 004). <ol style="list-style-type: none"> i. The 'To' Field will already be filled with the Sender's address ii. The Subject Field will automatically be filled with the subject of the opened email and a prefix of 'Re:'. iii. The text of the email (along with its' details) being replied to will appear in the body with allowance made for the user to input the new text above this text.
Alternative Flow	ALT 01. The user will minimize the system (Refer Use Case ID 013) 02. The user will maximize the system (Refer Use Case ID 014)
Exception Flow	EX 01. The user Closes the system using the relevant voice command (Refer Use Case ID 015)
Post Conditions	The user will be able to reply to a sender of the open email.

Use Case ID	011
Use Case Name	Reply All on email
Use Case Description	This use case describes how a user may reply all on the email with a simple voice command.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Inbox Page. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.

	User has already opened an email (Refer Use Case ID 008)
Flow	<ol style="list-style-type: none"> 1. The user states the command 'Email Reply All' 2. User will be directed to the new email page within the Email Client Form browser (Refer Use Case ID 004). <ol style="list-style-type: none"> i. The 'To' Field will already be filled with all the recipients addresses. ii. The Subject Field will automatically be filled with the subject of the opened email and a prefix of 'Re:'. iii. The text of the email (along with its' details) being replied to will appear in the body with allowance made for the user to input the new text above this text.
Alternative Flow	<p>ALT 01. The user will minimize the system (Refer Use Case ID 013)</p> <p>02. The user will maximize the system (Refer Use Case ID 014)</p>
Exception Flow	EX 01. The user Closes the system using the relevant voice command (Refer Use Case ID 015)
Post Conditions	The user will be able to reply to all on the opened email.

Use Case ID	012
Use Case Name	Forwarding an email
Use Case Description	This use case describes how a user can forward a selected email.
Actors	User
Pre-Conditions	<p>User is successfully logged into the system and navigated to the Inbox Page.</p> <p>Speech Recognition Icon on the screen is active (Refer Use Case ID 002).</p> <p>User should have access to the Internet.</p> <p>User should have a microphone connected to their computer.</p> <p>User has already opened an email (Refer Use Case ID 008)</p>
Flow	<ol style="list-style-type: none"> 1. The user states the command 'Email Forward'. 2. User will be directed to the new email page within the Email Client Form browser (Refer Use Case ID 004). <ol style="list-style-type: none"> i. The 'To' Field will already be filled with the Sender's address ii. The Subject Field will automatically be filled with the subject of the opened email and a prefix of 'Fw:'. iii. The text of the email (along with its' details) being forwarded will appear in the body with allowance made for the user to input the new text above this text.
Alternative Flow	ALT 01. The user will minimize the system (Refer Use Case

	ID 013) 02. The user will maximize the system (Refer Use Case ID 014)
Exception Flow	EX 01. The user Closes the system using the relevant voice command (Refer Use Case ID 015)
Post Conditions	The user will be able to forward the selected email to the required recipients.

Use Case ID	013
Use Case Name	Minimizing the system
Use Case Description	This use case describes how a user can minimize the system.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Email Client Form. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.
Flow	<ol style="list-style-type: none"> 1. The user will be able to minimize the system by stating the command “select minimize”. 2. The window will be minimized to the users task bar.
Alternative Flow	-
Exception Flow	-
Post Conditions	The user will be able to minimize the system to their taskbar.

Use Case ID	014
Use Case Name	Maximizing the system
Use Case Description	This use case describes how a user can maximize the system.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Email Client Form. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.
Flow	<ol style="list-style-type: none"> 1. The user will able to maximize the system stating the command “select maximize”. 2. The window will be maximized to a larger size.
Alternative Flow	-

Exception Flow	-
Post Conditions	The user will be able to maximize the system to a larger size than displayed.

Use Case ID	015
Use Case Name	Closing the system
Use Case Description	This use case describes how a user can close or escape the system.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Email Client Form. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.
Flow	<ol style="list-style-type: none"> 1. The user will state the command 'Select close'. 2. The system will close and the user will be directed to their desktop.
Alternative Flow	-
Exception Flow	-
Post Conditions	The user will be able to close the system and exit it.

Use Case ID	016
Use Case Name	Minimizing a form
Use Case Description	This use case describes how a user can minimize the form the user is currently viewing.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Email Client Form. User has opened a form within the Email Client Form page. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.
Flow	<ol style="list-style-type: none"> 1. The user will be able to minimize the form focused stating the command "select minimize" 2. The form will be minimized to the taskbar so the user can view the Email Client Form.
Alternative Flow	-

Exception Flow	-
Post Conditions	The user will be able to minimize the form to view the Email Client Form page.

Use Case ID	017
Use Case Name	Maximizing a form
Use Case Description	This use case describes how a user can maximize a form the user is currently viewing.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Email Client Form. User has opened a form within the Email Client Form page. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.
Flow	<ol style="list-style-type: none"> 1. The user will be able to maximize the form focused stating the command “select maximize” 2. The form will be maximized to a larger size.
Alternative Flow	-
Exception Flow	-
Post Conditions	The user will be able to maximize the form to a larger size than displayed.

Use Case ID	018
Use Case Name	Closing the Form
Use Case Description	This use case describes how a user can close or escape the form currently being viewed.
Actors	User
Pre-Conditions	User is successfully logged into the system and navigated to the Email Client Form. User has opened a form within the Email Client Form page. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet. User should have a microphone connected to their computer.
Flow	<ol style="list-style-type: none"> 1. The user will state the command ‘select close’ when a form has opened up. 2. The form will close and the user will be directed back to the Email Client Form.

Alternative Flow	-
Exception Flow	-
Post Conditions	The user will be able to close the form and return to the Email Client Form.

Use Case ID	019
Use Case Name	Updating XML File with Login Details
Use Case Description	This use case describes how the user will update his/her login credentials to the xml file before logging on to the system.
Actors	User
Pre-Conditions	User should have a web-email account (Gmail) with valid login credentials User should have access to the Internet
Flow	<ol style="list-style-type: none"> 1. User will open XML file. 2. User will then input the username (Email address), e.g.:- voicedrivenemailclient@gmail.com 3. User will input password for relevant account 4. User will input the Web Mail Client. E.g.:- Gmail 5. Data will be saved by using the save function Ctrl + S
Alternative Flow	ALT 01. i. Different user wants to input details ii. User performs steps 1-5
Exception Flow	Ex 01. i. User enters incorrect details ii. User will repeat steps 2-5
Post Conditions	Xml file has been successfully updated with user's credentials.

Use Case ID	020
Use Case Name	Communication with the Speech API
Use Case Description	This use case describes how the system will communicate with the Speech API to convert Voice into Text.
Actors	Voice driven Email Client
Pre-Conditions	The Speech Recognition Engine and the Grammar is loaded when system is launched. Voice needs to be input into the system
Flow	<ol style="list-style-type: none"> a. System will detect users voice over microphone. b. The email client will send the voice phrase to the Speech API. c. The Speech API will convert the voice phrase into text and send it back to the Voice Driven Email Client for displaying.
Alternative Flow	-

Exception Flow	-
Post Conditions	The voice phrase input by the user will be converted to text and displayed on the screen in the relevant areas in the system.

Use Case ID	021
Use Case Name	Communication with SMTP Server
Use Case Description	This use case describes how the system will communicate with the SMTP Server
Actors	Voice Driven Email Client
Pre-Conditions	System will capture the values for the following parameters from the saved data in the xml file: <ol style="list-style-type: none"> 1. Username 2. Password 3. SMTP Host name (Ex: smtp.gmail.com) 4. SMTP port number (Ex: 587) <p>User should be connected to the Internet.</p>
Flow	<ol style="list-style-type: none"> 1. System captures the email message fields (To, CC, BCC Subject, Attachment and Body) that need to be sent. 2. The system then sends the email via the SMTP server by using the client's SMTP host name and port number.
Alternative Flow	-
Exception Flow	Ex 01.a. The Email cannot be sent due to a loss in connection b. The Connection will be re-established and the email sent
Post Conditions	The Email will be sent through the SMTP Server.

Use Case ID	022
Use Case Name	Communication with POP3 Server to retrieve the inbox
Use Case Description	This use case describes how the system will communicate with the POP3 Server to retrieve the users' inbox.
Actors	Voice Driven Email Client
Pre-Conditions	System will capture the values for the following parameters from the saved data in the xml files when the system launches: <ol style="list-style-type: none"> 1. Username 2. Password 3. POP3 Host name (Ex: pop.gmail.com) 4. POP3 port number (Ex: 995) <p>User should be connected to the Internet.</p>

Flow	<ol style="list-style-type: none"> 1. User requests for inbox messages 2. System connects to the web email client's POP3 server (POP3 host name) using the POP3 port. 3. The POP3 server requires an account name and password, the system will provide the web email client credentials to the server. 4. The system retrieves the inbox.
Alternative Flow	-
Exception Flow	-
Post Conditions	The users' Inbox will be successfully retrieved and displayed on screen.

Use Case ID	023
Use Case Name	Communication with POP3 Server to delete an email
Use Case Description	This use case describes how the system will communicate with the POP3 Server to delete an email and update the inbox.
Actors	Voice Driven Email Client
Pre-Conditions	<p>System will capture the values for the following parameters from the saved data in the xml files when the system launches:</p> <ol style="list-style-type: none"> 1. Username 2. Password 3. POP3 Host name (Ex: pop.gmail.com) 4. POP3 port number (Ex: 995) <p>User should be connected to the Internet.</p>
Flow	<ol style="list-style-type: none"> 1. User requests to delete an email. 2. System connects to the web email client's POP3 server (POP3 host name) using the POP3 port. 3. The POP3 server requires an account name and password. The system will provide the web email client credentials to the server. 4. The system deletes the selected email. 5. The updated inbox is then displayed to the user.
Alternative Flow	-
Exception Flow	-
Post Conditions	The required email will be deleted and the updated Inbox will be displayed on screen.

Use Case ID	024
Use Case Name	Updating the Attachment Folder.

Use Case Description	This use case describes how the user can update the attachment folder so that it can be attached to an email.
Actors	User
Pre-Conditions	User should be logged in to the system. User should be connected to the internet.
Flow	<ol style="list-style-type: none"> 1. User opens Attachment Folder 2. User adds required attachment 3. User closes folder 4. Folder contents will be saved.
Alternative Flow	-
Exception Flow	-
Post Conditions	The Attachment folder will be successfully updated and the user will be able to attach the file to an email.

Use Case ID	025
Use Case Name	Deleting text that has been input in the ‘Subject’ and ‘Body’ field.
Use Case Description	This use case describes how the user can delete the last dictated phrase or word they have already input in the ‘Subject’ and ‘Body’ field.
Actors	User
Pre-Conditions	User is logged into the system and navigated to the Email Client Form. Speech Recognition Icon on the screen is active (Refer Use Case ID 002). User should have access to the Internet User should have a microphone connected to their computer. User has stated the command ‘ Select New’ / User has stated the command ‘ Reply’ / User has stated the command ‘ Reply All’ / User has stated the command ‘Forward’.
Flow	<ol style="list-style-type: none"> 1. The user will state the command ‘Undo’ 2. The last dictated word/phrase will be removed from the specific field. 3. The user will be allowed to continue with the dictation.
Alternative Flow	-
Exception Flow	-
Post Conditions	The system will allow the user to undo the last dictated phrase in the ‘Subject’ and ‘Body’ fields.

4.5 Command and Diction Functions Design

4.5.1 Designing Commands

One of the most important decisions to be made when implementing a speech based system is the format of the commands. The commands chosen need to be consistent throughout the system.

- Keywords

The keywords chosen for this system are;

- Select

The 'select' keyword is used for selecting fields that allow the user to directly dictate, close forms and for selecting buttons. Examples;

- New Button – select new
- Subject Field – select subject
- Close form – select close

There are 3 buttons excluded from this which are 'reply', 'reply all' and 'forward' buttons.

- Include

The 'include' keyword is used in instances where the fields need to be filled using data from a xml file or another form (without directly dictating to the field). This is true for all such fields except for the 'Attachment' Field. Examples;

- Getting credentials from XML file – include username, include password
- Getting contacts from address book form – include 'contact id'

- Open

The 'open' keyword is used for opening emails in the inbox. Example;

- Open 'email id'

- Delete

The 'delete' keyword is used for deleting emails in the inbox. Example;

- Delete 'email id'

- Email

The 'email' keyword is used in only 3 instances. This is used for the 3 functions that are allowed to be performed once an email is open. These functions are 'reply', 'reply all' and 'forward'. Examples;

- Email reply
- Email reply all
- Email forward

- Attach
 - The 'attach' keyword is used for adding attachments to the email. Example;
 - Attach 'attachment id'
- Command List

Given below is the entire command list used in the system. Every command will be the length of only 2 or 3 words. Further in commands, homophones are taken into consideration. Below shown are the commands along with the homophones (a few words that sound the same have also been added).
- Select
 - Select sign in (select signed in/select signing)
 - Select exit
 - Select new
 - Select inbox
 - Select to(select two/select too)
 - Select cc
 - Select bcc
 - Select contact (select contacts)
 - Select subject
 - Select attachment (select attachments)
 - Select body
 - Select send
 - Select close
 - Select enter
 - Select up
 - Select down
 - Select tab
 - Select minimize
 - Select maximize
- Include
 - Include username
 - Include password
 - Include client
 - Include 'contact id'
- Open
 - Open 'email id'

- Delete
 - Delete 'email id'

- Email
 - Email reply (e mail reply)
 - Email reply all
 - Email forward (email for word, email four word)

4.5.2 Differentiating Commands and Dictation

Once the speech API recognizes word/phrases, the system has to determine the function to be performed. The initial step is to identify whether it is a command or a phrase dictated to be printed on a field.

4.5.2.1 Identifying Commands

- Command rules

A command starts with the keywords **select**, **include**, **open**, **delete** or **email**

The length of a command will be 2 or 3 words. The diagram below depicts how the commands of the system are handled. If the conditions are true, they will be considered as commands, and the corresponding action will be performed.

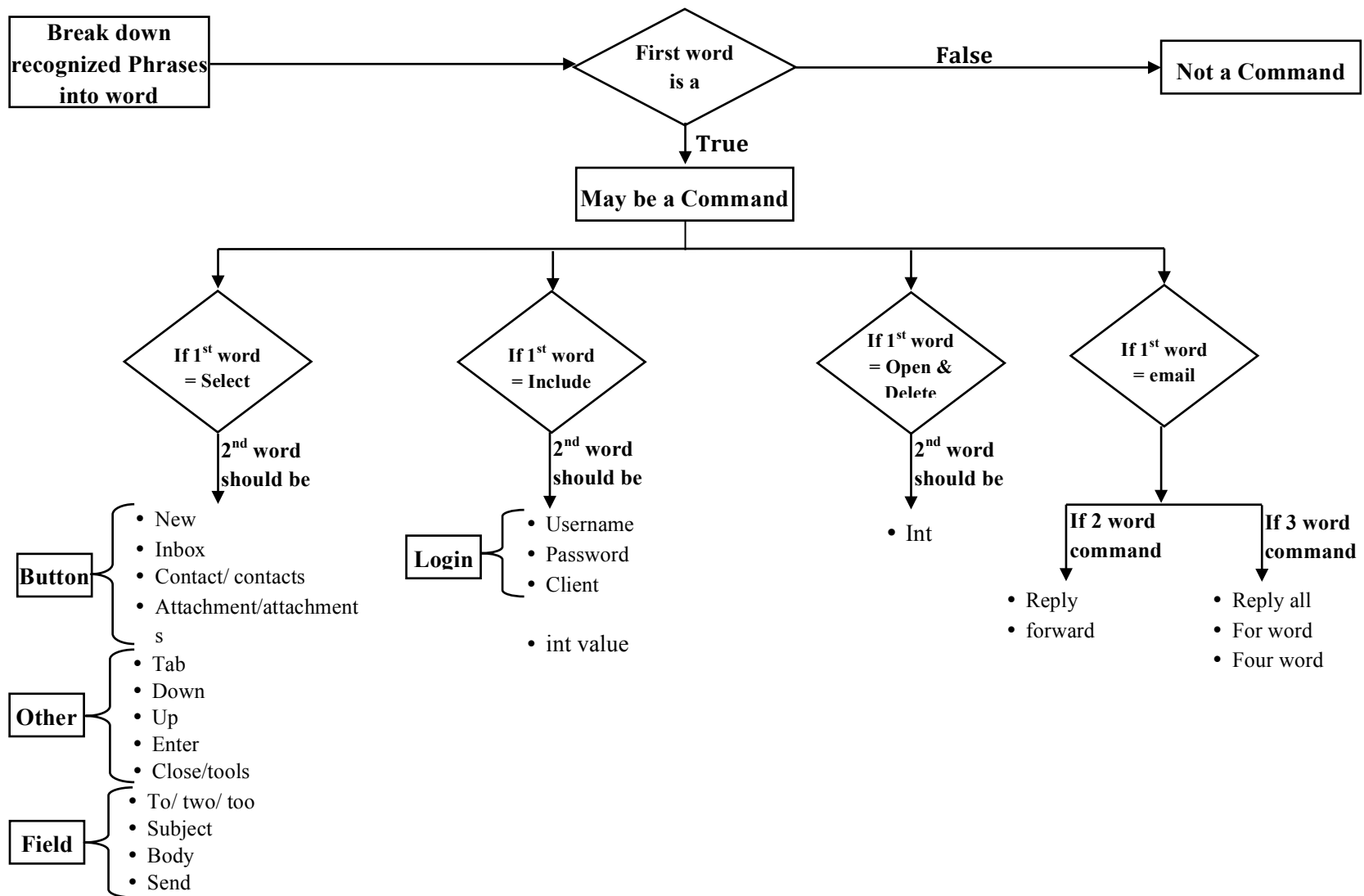


Figure 11 : Handling Commands

4.5.2.1 Identifying Dictation

There are currently only two fields in the system that allows the user to dictate. The two fields are the 'subject' field and the 'body' field.

Before dictating to a field, the user needs to select that particular field using voice commands. Once the focus is set, every word the user dictates gets typed into the field until the user commands the system to move to another function.

This can be achieved by maintaining flags for each function. Example of dictating to the subject field

The user will select the subject by voicing in the command 'select subject'. The system will then turn the 'subjectFieldFlag' to 'true'. As long as the flag is 'true', all words dictated will appear in the subject field.

Once the user moves to another function, the 'subjectFieldFlag' will be set to 'false', and the flag of the particular function selected will be set to 'true'.

4.5.2.1 Dictating Commands

The user might come across instances where commands need to be dictated. The user would have to achieve this by dictate such words separately.

For example the sentence 'she had to select new shoes' could be dictated as;

Option one:

- She had to select
- New shoes

Option two:

- She had to select new shoes

But not as:

- she had to
- Select new
- Shoes

4.5.2.2 Dictating Symbols

The following symbols are supported by the system for dictating.

Symbol	How to Dictate
.	Full stop
,	comma
:	colon
;	Semi colon
?	Question mark
-	hyphen

4.5.2.3 Undo Dictated word/phrase

The system should allow the user the undo dictated phrases in the 'Subject' and 'Body' fields. The undo option will be available only for one level of undoing.

The command for undoing dictated phrases is 'undo'. The user needs to simple voice in the command 'undo' and it will remove the last dictated word/phrase from the specific field.

4.6 Screen Design

A good screen design is an important aspect of a system. Given below are few designing related decisions taken when implementing the Voice Driven Email Client.

- A Good start up screen – eye catching, includes rights



Figure 12 : Start up Screen Design

- A good login screen – fields are clear, most important aspect ‘Speech Recognition’ placed right at the top.

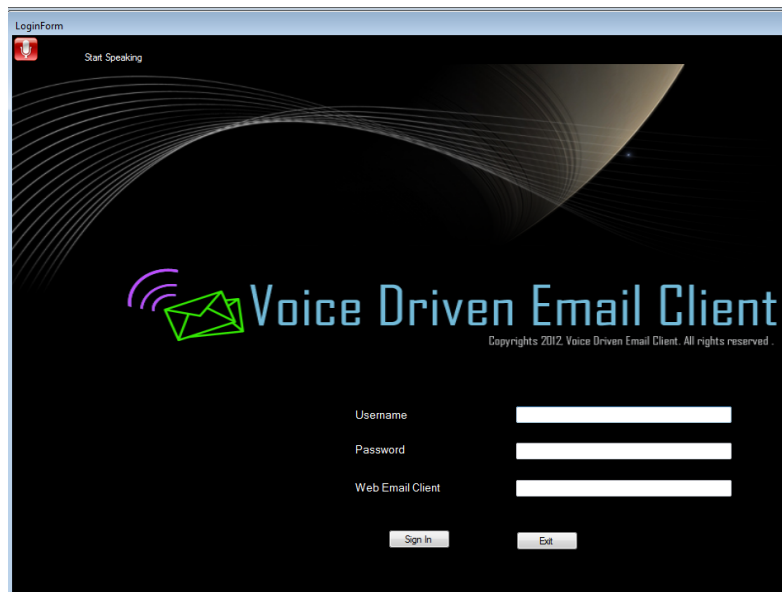


Figure 13 : Login Screen Design

- **Aligning Button** – though web email clients have their buttons aligned at the left vertically, the Voice Driven Email Client has the buttons aligned on the top horizontally. Since this system is a desktop application it follows the approach of most desktop applications.

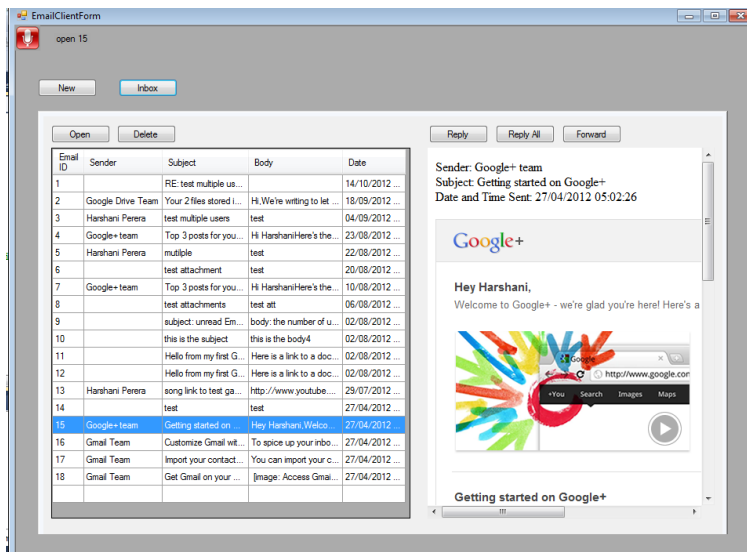


Figure 14 : Alignment of Buttons Screen Design

- **Displaying Emails** – the emails are displayed in a web browser control to allow user to view html content.

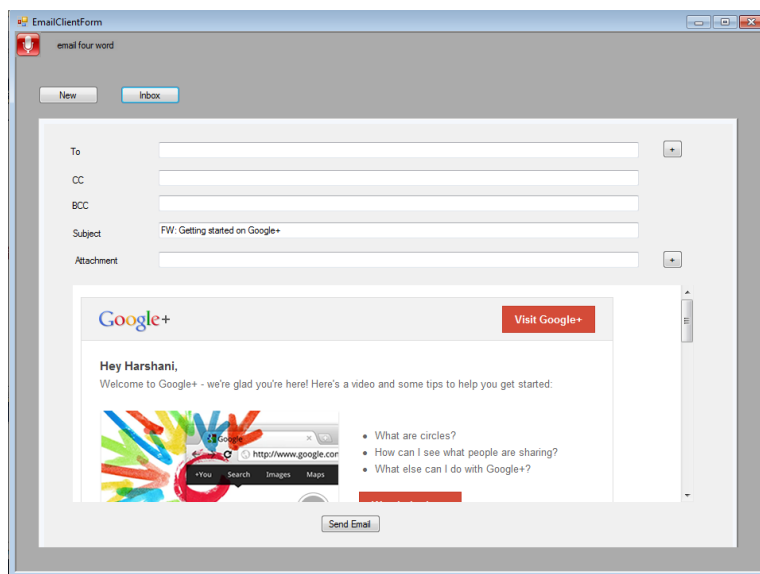


Figure 15 : Displaying Emails Screen Design

4.7 Property Matrix

4.7.1 Inbox

No.	Field Name	Field Type	Display Text	Voice Command	Required	Validations	Conditions	Remarks
	Open	Button	Open	Open (Email ID)				The opened email will be shown in the email preview section.
	Delete	Button	Delete	Delete (Email ID)			If selected email server is updated.	Message Deleted message displayed to user.
	Reply	Button	Reply	Email Reply				Once Clicked the New Email Screen will be shown. The To field will have the address already input as well as the subject and a 'Re' will be added before the subject.

	Reply All	Button	Reply All	Email Reply All				Once Clicked the New Email Screen will be shown. The To field will have the address already input as well as the subject and a 'Re' will be added before the subject.
	Forward	Button	Forward	Email Forward				Once Clicked the New Email Screen will be shown. The To field will be left blank. The subject field will already be pre-filled with a prefix of 'Fw' added before the subject.
	Email ID	Table Header	Unique Email ID given to all emails	Select (Email ID)				User can select Email using this ID e.g:- Select 'Two' will select email two.
	Sender	Table Header	Sender of email obtained from email					

			account on server					
	Subject	Table Header	Subject of email obtained from email account on server					
	Body	Table Header	Body of email obtained from email account on server					
	Date and Time	Table Header	Date and Time of email obtained from email account on server					
	Scroll bars-Horizontal/Vertical							User can use the scroll bars for viewing purposes
	Email Preview	Obtained from Web Mail Server(Preview						

		of Selected Email)						
	Scroll bars- Horizontal/Vertical							User can use the scroll bars for viewing purposes
	Message Deleted Message	Text	Message deleted Successfully. Re-loading Inbox.				Displayed when Delete option is Selected	
	Ok button	Button	Ok					
	Close	Button	Icon	Select Close				User can click on button or use Voice Command

4.7.2 Login Form

No.	Field Name	Field Type	Display Text	Voice Command	Required	Validations	Conditions	Remarks
1	Splash screen	Screen	Voice Driven Email Client		Yes		Display for 5 seconds	
2	User Name	Text Box	User Name	Include User Name	Yes	Input UserName = Gmail UserName	When voice command given, data will be fetched from xml file	User can input their own data even if it is not on xml file
3	Password	Text Box	Password	Include Password	Yes	Input Password = Gmail Password	When voice command given, data will be fetched from xml file	User can input their own data even if it is not on xml file
4	Web Email Client	Text Box	Web Email Client	Include Client	Yes	Valid account should exist	When voice command given, data will be fetched from xml file	User can input their own data even if it is not on xml file
5	Sign In	Button	Sign In	Select Sign In	Yes			
6	Exit	Button	Exit	Select Exit	Yes			
7	Speech Recognition Icon	Icon			Yes			'Start Speaking' will be the initial set of text that is displayed and this will be replaced with the voice input
8	Speech Recognition Text	Text	Word/phrase input by voice		Yes			

4.7.3 Email Client Form

No.	Field Name	Field Type	Display Text	Voice Command	Required	Validations	Conditions	Remarks
1	Speech Recognition Icon	Icon			Yes			Start Speaking' will be the initial set of text that is displayed and this will be replaced with the voice input.
2	Speech Recognition Text	Text	Word/phrase input by voice		Yes			
3	New	Button	New	Select New	Yes		Select = New Email Screen	
4	Inbox	Button	Inbox	Select Inbox				
5	Minimize	Button						Both the shortcut and clicking on the button will complete the function.
6	Maximize	Button						Both the shortcut and clicking on the button will complete the function.
7	Close	Button		Select Close				User can click on button or use Voice Command

4.7.4 New Email

No.	Field Name	Field Type	Display Text	Voice Command	Required	Validations	Conditions	Remarks
1	To	Text Box	To	Select To,Select Contact(s),	Yes	If To Field= Null, Display Error Message		Select Contact(s) will open Contact Form and added address will be displayed in the field.
2	CC	Text Box	CC	Select CC,Select Contact(s),	No			Select Contact(s) will open Contact Form and added address will be displayed in the field.
3	BCC	Text Box	BCC	Select BCC,Select Contact(s),	No			Select Contact(s) will open Contact Form and added address will be displayed in the field.
4	Subject	Text Box	Subject	Select Subject	No			

5	Attachment	Text Box	Attachment	Select Attachment(s)	No			On processing of Voice Command, The Attachment Form will be opened for the user to select an Attachment and the path will be displayed in this text box
6	Body	Text Box	Body	Select Body				User is able to dictate text
7	Send Email	Button	Send Email	Select Send			If Clicked= Email sent and Status form displayed	
8	Email Sent	Text	Email Sent				Displayed if email is sent	
9	Minimize	Button						Both the shortcut and clicking on the button will complete the function.
10	Maximize	Button						Both the shortcut and clicking on the button will complete the function.

11	Close	Button		Select Close				User can click on button or use Voice Command
12	Text	Text	The parameter 'addresses' cannot be an empty string.					
13	Minimize	Button						Both the shortcut and clicking on the button will complete the function.
14	Maximize	Button						Both the shortcut and clicking on the button will complete the function.
15	Close	Button		Select Close				User can click on button or use Voice Command
16	ID	Table Column Header	Unique ID	Select 'ID'				Voice command is given e.g.:- Select Three. The third Email address with ID 3 will be input into the field

17	Name	Table Column Header	Email contacts' Name generated from Web Mail Client					
18	Address	Table Column Header	Email Address generated from Web Mail Client					
19	Minimize	Button			Yes			Both the shortcut and clicking on the button will complete the function.
20	Maximize	Button			Yes			Both the shortcut and clicking on the button will complete the function.
21	Close	Button		Select Close	Yes			User can click on button or use Voice Command
22	Attachment ID	Table Column Header	Unique attachment ID generated from Attachment folder	Select (Attachment ID)				E.g.- Select Attachment Two. The attachment with the ID 2 will be appended.

23	Attachment Name	Table Column Header	Data obtained from Attachment Folder					
24	Attachment Path	Table Column Header	Data obtained from Attachment Folder					
25	Minimize	Button			Yes			Both the shortcut and clicking on the button will complete the function.
26	Maximize	Button			Yes			Both the shortcut and clicking on the button will complete the function.
27	Close	Button		Select Close	Yes			User can click on button or use Voice Command

Chapter 5 System Implementation

5.1 Chapter Overview

This chapter outlines the implementation of the system. It includes the development environment, and important source codes used in the system.

5.2 Implementation Environment

- The Language used for system development – C#
- The Integrated Development Environment IDE used for development – Microsoft Visual C# 2010 Express
- Speech Application Programming Interface (API) used – Microsoft Speech SDK 5.4 (SAPI 5.4)
- OpenPop .Net 2.0.4
- Google Data API setup 2.0
- Other – XML was used for storing information (commands, credentials etc)
- Web mail email account – in this case a Gmail account

5.2 Source Codes

This section includes often used important source codes of the system

5.2.1 Speech Recognition

The additional reference added to the system for Speech recognition is,

`using System.Speech.Recognition;`

- The Speech Recognition Engine and the Grammar are instantiated when the main form of the system is loaded.

```
SpeechRecognitionEngine recognizer = new SpeechRecognitionEngine();
Grammar dictationGrammar = new DictationGrammar();

recognizer.LoadGrammar(dictationGrammar);
```

- The user voice is Detected and Recognized using the ‘SpeechDetected’ and ‘SpeechRecognized’ Events respectively. The process will take place as long as the main form is active.

```
private void EmailClientForm_Activated(object sender, EventArgs e)
{
    recognizer.SetInputToDefaultAudioDevice();
    recognizer.SpeechDetected += new
    EventHandler<SpeechDetectedEventArgs>(recognizer_SpeechDetected);
    recognizer.SpeechRecognized += new
    EventHandler<SpeechRecognizedEventArgs>(recognizer_SpeechRecognized);
    recognizer.RecognizeAsync(RecognizeMode.Multiple);
}
```

- ‘SpeechRecognized’ Event recognizes the captured voice-input. It further performs the string cleansing and handling (determining the function to be performed) by passing the values to the application tier.

```
public void recognizer_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
{
    GlobalVariables.RecognizedString = e.Result.Text;

    SpeechRecognizedLbl.Visible = true;
    SpeechRecognizedLbl.Text = GlobalVariables.RecognizedString;

    cleansedString =
    commomFunctionsObj.cleansString(GlobalVariables.RecognizedString); //cleans
    string : remove noise, remove spaces at the ends, covert to lowercase

    if (string.IsNullOrEmpty(cleansedString) == false)
    {
        Speech speechObj = new Speech(this);
        speechObj.handleRecognizedString(cleansedString);
    }
}
```

When the recognized speech is handled, it determines whether the recognized phrase is a command or dictation. In this process when number are captured “one”, “two” they need to be converted to integers “1”, “2” etc. the algorithm to perform this task was obtained from the web site ‘stackoverflow’

- The 'SpeechDetected' Event in this case is used to indicate each time the microphone captures a sound. Once a sound is detected it is shown by the green microphone icon. (it stays for 2 seconds and turns back to red)

```
public void recognizer_SpeechDetected(object sender, SpeechDetectedEventArgs e)
{
    ActivepictureBox.Visible = true;
    InactivepictureBox.Visible = false;
    SpeechRecognitionIntimer.Interval = 2000;
    SpeechRecognitionIntimer.Start();
}
```


5.2.2 Sending Email

Emails are sent using the SMTP (Simple Mail Transfer Protocol) protocol. In this case;

- The Credentials used are the Gmail account Username and Password
- the SMTP Host used - smtp.gmail.com
- The SMTP Port used – 587
- The additional reference added to the system for Sending Email is, `using System.Net.Mail;`
- Sending Emails method

```
public void SendEmail(string from, string to, string subject, string body, string username,
string password, string attachment)
{
    try
    {
        var message = new MailMessage();
        message.From = new MailAddress(from);
        message.To.Add(to);
        message.Subject = subject;
        message.Body = body;

        if (String.IsNullOrEmpty(attachment) == false)
        {
            System.Net.Mail.Attachment data = new
            System.Net.Mail.Attachment(@attachment);
            message.Attachments.Add(data);
        }

        SmtplibClient mailer = new SmtplibClient(GlobalVariables.Smtphost,
        GlobalVariables.Smtpport);
        mailer.Credentials = new NetworkCredential(username, password);
        mailer.EnableSsl = true;
        mailer.Send(message);
    }

    catch (Exception ex)
    {
        throw (ex);
    }
}
```

5.2.3 Retrieving Email

Emails are retrieved using the POP3 (Post Office Protocol) protocol. In this case;

- The Credentials used are the Gmail account Username and Password
- the POP3 Host used - pop.gmail.com
- The POP3 Port used – 995

The open source POP3 client ‘OpenPop’ was for this purpose. The binary file ‘OpenPop.dll’ is copied to the solution binary folder. The additional references added for retrieving email were;

```
using OpenPop.Common.Logging;
using OpenPop;
using OpenPop.Mime;
using OpenPop.Mime.Decode;
using OpenPop.Mime.Header;
using OpenPop.Pop3;
using OpenPop.Pop3.Exceptions;
```

- Retrieving Emails method

```
public static List<OpenPop.Mime.Message> RetrieveInboxMsgs(Pop3Client client,
string hostname, int port, bool useSsl, string username, string password)
{
    client.Connect(hostname, port, useSsl); // Connect to the server
    client.Authenticate(username, password); //Authenticate

    int messageCount = client.GetMessageCount();
    List<OpenPop.Mime.Message> allMessages = new
    List<OpenPop.Mime.Message>(messageCount);

    for (int i = messageCount; i > 0; i--)
    {
        allMessages.Add(client.GetMessage(i));
    }
    return allMessages;
}
```

5.2.4 Get Contacts

Contacts are retrieved from Gmail using the Google Data API. The binary files added to the system are;

- Google.GData.Client.dll
- Google.GData.Contacts.dll
- Google.GData.Extensions.dll

The additional references added were;

```
using Google.GData.Contacts;
using Google.GData.Extensions;
using Google.Contacts;
using Google.GData.Client;
```

- Get contacts method

```
public void GetContacts(System.Windows.Forms.DataGridView dataGridView)
{
    RequestSettings requestSettings = new RequestSettings("",
    GlobalVariables.Username, GlobalVariables.Password);
    requestSettings.AutoPaging = true;
    ContactsRequest contactsRequest = new ContactsRequest(requestSettings);
    Feed<Google.Contacts.Contact> feedContacts = contactsRequest.GetContacts();
    int contactId = 1;

    foreach (Google.Contacts.Contact gmailAddresses in feedContacts.Entries)
    {
        string contactName = gmailAddresses.Title;

        foreach (EMail emailId in gmailAddresses.Emails)
        {
            string contactAddress = emailId.Address;
            dataGridView.Rows.Add(contactId, contactName, contactAddress);
            contactId++;
        }
    }
}
```

5.2.5 Converting English word to numbers

This algorithm for converting English words to numbers was obtained from the Stack Overflow website.

```
private static Dictionary<string, long> numberTable = new Dictionary<string, long>
    {{"zero",0},{ "one",1},{ "two",2},{ "three",3},{ "four",4},
    {"five",5},{ "six",6},{ "seven",7},{ "eight",8},{ "nine",9},
    {"ten",10},{ "eleven",11},{ "twelve",12},{ "thirteen",13},
    {"fourteen",14},{ "fifteen",15},{ "sixteen",16},
    {"seventeen",17},{ "eighteen",18},{ "nineteen",19},{ "twenty",20},
    {"thirty",30},{ "forty",40},{ "fifty",50},{ "sixty",60},
    {"seventy",70},{ "eighty",80},{ "ninety",90},{ "hundred",100},
    {"thousand",1000},{ "million",1000000},{ "billion",1000000000},
    {"trillion",1000000000000},{ "quadrillion",1000000000000000},
    {"quintillion",1000000000000000000}};

public static long ToLong(string numberString)
{
    var numbers = Regex.Matches(numberString, @"\w+").Cast<Match>()
        .Select(m => m.Value.ToLowerInvariant())
        .Where(v => numberTable.ContainsKey(v))
        .Select(v => numberTable[v]);
    long acc = 0, total = 0L;
    foreach (var n in numbers)
    {
        if (n >= 1000)
        {
            total += (acc * n);
            acc = 0;
        }
        else if (n >= 100)
        {
            acc *= n;
        }
        else acc += n;
    }
    return (total + acc) * (numberString.StartsWith("minus",
        StringComparison.InvariantCultureIgnoreCase) ? -1 : 1);
}
```

5.3 Source code overview

The diagram given below was generated by Visual Studio at the end of the implementation to show the overview of the source code. It shows the classes of the system.

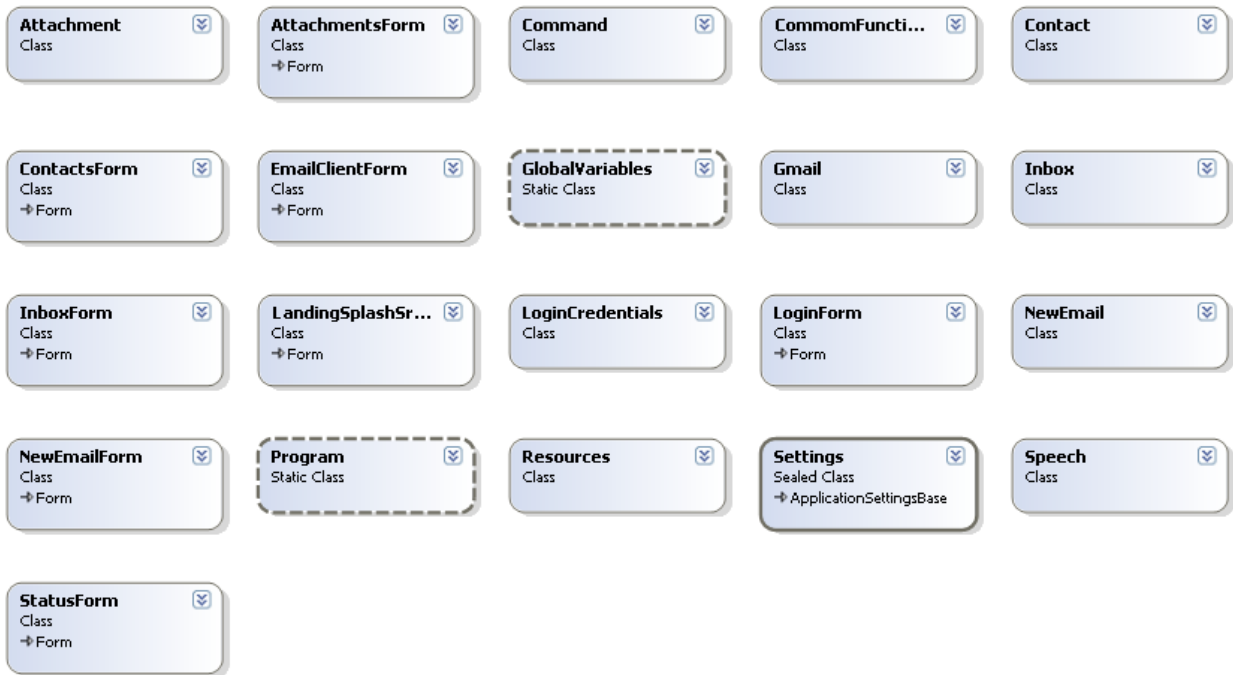


Figure 16 : Source Code Overview

Chapter 6 Testing

6.1 Chapter Overview

This chapter outlines the testing of the system that was carried out. The types of testing approaches carried out were unit testing, integration testing, overall system testing and performance testing.

6.2 Unit Testing

Unit testing was carried out for the main function components of the system. The main functions of the system are;

- Speech Recognition – to ensure recognition happens accurately
- Sending Emails – to ensure emails are sent successfully
- Receiving Emails – to ensure emails are received successfully

6.2.1 Speech Recognition

Unit Test Case Name	Speech Recognition test 1
Feature Description	Validate the accuracy of speech recognition for a single word
Preconditions	Have a microphone
Test Description	Speak one word at a time and record the accuracy.
Input Data	Voice input – one word
Expected Result	100% accurate
Actual Result	73.33% accurate

Word Spoken	Word Recognized	Pass/Fail
new	new	pass
inbox	inbox	pass
to	too	fail
contact	contact	pass
attachment	attachment	pass
send	send	pass

close	tools	fail
email	email	pass
username	username	pass
client	client	pass
forward	Four word	fail
Reply	To play	fail
open	open	pass
include	include	pass
select	select	pass

Unit Test Case Name	Speech Recognition test 2
Feature Description	Validate accuracy of speech recognition of a phrase
Preconditions	Have a microphone
Test Description	Speak few words at a time and record the accuracy.
Input Data	Voice input – one word
Expected Result	100% accurate
Actual Result	60% accurate

Phrase Spoken	Phrase Recognized	Pass/Fail
This is the subject	This is the subject	pass
This is the body	This is the body	pass
This is a email client	Receives email client	fail
Final year project	Final year project	pass

Testing speech recognition	Testing speech recognition	pass
Reply to the email	Or replying to the email	fail
Recognize the dictated text	Recognize the dictated text	pass
Environment has to be silent	Environment has to be silent	pass
Try dictating a lot of words	Try dictating after that words	fail
This is the to field	This is the tall field	fail

6.2.2 Sending Emails

Unit Test Case Name	Sending Emails
Feature Description	Verify whether emails are sent
Preconditions	Following values are correct and not empty; Valid username Valid password SMTP Host name SMTP port number Email to field Email subject field Email body
Test Description	Try for different contacts, subjects and body
Input Data	N/A
Expected Result	100% accurate
Actual Result	100% accurate

6.2.3 Receiving Emails

Unit Test Case Name	Receive Emails
---------------------	----------------

Feature Description	Verify whether emails are received every time
Preconditions	Following values are correct and not empty; Valid username Valid password POP3 Host name POP3 port number
Test Description	Try for several times
Input Data	N/A
Expected Result	100% accurate
Actual Result	100% accurate

6.3 Integration Testing

Integration testing was carried out for the main three components of the system. ‘Sending’ and ‘Receiving’ emails was integrated with ‘Speech Recognition’

Test Case Name	Integration Testing
Feature Description	Verify whether emails can be sent and received using voice commands (only navigation)
Preconditions	Following values are correct and not empty; Valid username Valid password POP3 Host name POP3 port number SMTP Host name

	SMTP port number Email to field Email subject field Email body
Test Description	1.Voice-in command “select send” to send email 2.Voice-in command “select inbox” to receive emails
Input Data	Voice inputs - “select send” and “select inbox”
Expected Result	100% accurate
Actual Result	100% accurate

6.4 Overall system testing

The overall system testing allows verifying the functionality and accuracy of the entire system. Shown below are the test cases for each function of the system. Overall system testing was carried out for both versions (Separate interface and Gmail interface).

All the test cases passed except for the sign out test cases. The sign out function works, but the system doesn't recognize the phrase "sign out" at the moment. This could be resolved by training the system.

6.4.1 Separate interface

1.0 Launch the system (Voice Driven Email Client)

Test Case ID	1.0.1
Category	Functional
Feature Description	Validate whether the user can launch the system (Voice Driven Email Client)
Preconditions	<ol style="list-style-type: none"> 1. User must have web-email account (Gmail) with valid login credentials 2. The username, password and web-client name are saved on the 'LoginCredentials.xml' file 3. The pop3 host and pop3 port must saved on the 'GmailConnectionDetails.xml' file 4. User must have access to the Internet 5. User must have a microphone
Test Description	1. Double click on the system
Input Data	N/A
Expected Result	1. User should be able to launch the system
Actual Result	User is able to successfully launch the system
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	1.0.2
Category	UI
Feature Description	Verify the look and feel of the Login Form/Page
Preconditions	1. User must be in startup screen
Test Description	1. Verify the start up screen
Input Data	N/A
Expected Result	1. System should starts up with Splash Screen which stays for 5 seconds 2. System should navigate to the Login Form/Page after 5 seconds
Actual Result	System starts up with Splash screen which stays for 5 seconds are navigates to the login form
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	1.0.3
Category	UI
Feature Description	Verify the look and feel of the Login Form/Page
Preconditions	1. User must be in Login Form/Page
Test Description	1. Verify following fields under Login Form/Page - Red color microphone icon - "Start Speaking" message (Speech Recognition status message) - Username <Input Text Field> - Password<Input Text field>

	<ul style="list-style-type: none"> - Web Email Client <Input Text Field> - "Sign In" <Button> - "Exit" <Button>
Input Data	N/A
Expected Result	<p>1. User Should be able to see following fields under Login Form/Page</p> <ul style="list-style-type: none"> - Red color microphone icon should display left top side of the Login Form Screen Under LoginForm Title bar - "Start Speaking" message should display at right side of the microphone icon - Username <Input Text Field> - Password<Input Text field> - Web Email Client <Input Text Field> - "Sign In" <Button> - "Exit" <Button>
Actual Result	<ul style="list-style-type: none"> - Red color microphone icon is displayed on left top side of the Login and the login form consists of the following fields and buttons - User Name <Input Text Field> - Password<Input Text field> - Web email Client <Input Text Field> - "Sign In" <Button> - "Exit" <Button>
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	1.0.4
Category	Functional
Feature Description	Validate whether the system can recognize the user's speech
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color 4. "Start Speaking" Message must be display
Test Description	<ol style="list-style-type: none"> 1. Speak something using microphone <p>EX:" hello world "</p>
Input Data	Voice input – ‘hello world’
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. "Start speaking" message (Speech Recognition status message) should be change in to "hello world" 3. after 2 seconds microphone should change back to red color
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon is changed to green and the message changes to “hello world” 2. after 2 seconds microphone should change back to red color
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

1.1 Login to the Main Page (Email Client Page)

Test Case ID	1.1.1
Category	Functional
Feature Description	Validate whether the user can input username by giving a valid voice command
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <p>- Include Username</p>
Input Data	Refer xml, voice input – “include username”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "include username" 3. User name which is include in xml file should be display in username input text field
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "include username" 3. username input text field displays “voicedrivenemailclient@gmail.com”
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	1.1.2
Category	Functional

Feature Description	Validate whether the user can input password by giving a valid voice command
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <p>- Include password</p>
Input Data	Refer xml, voice input – “include password”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "include password" 3. password which is included in xml file should be display in password input text field
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "include password" 3. username input text field displays “qetu6421”
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	1.1.3
Category	Functional
Feature Description	Validate whether the user can input client by giving a valid voice command

Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <p>- Include client</p>
Input Data	Refer xml, voice input – “include client”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "include password" 3. Client which is included in xml file should be display in web email client input text field
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "include client" 3. username input text field displays “Gmail”
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	1.1.4
Test Case Name	
Category	Functional
Feature Description	Validate whether the user can navigate to the main page by giving the valid command for sign in
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone

	<ul style="list-style-type: none"> 3. Microphone must be red color 4. User must have given the valid login credentials
Test Description	<ul style="list-style-type: none"> 1. Speak Up Following Commands - Select Sign in
Input Data	voice input - Select Sign in
Expected Result	<ul style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "Select sign in" 3. User should be navigate to the main page
Actual Result	<ul style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "select sign in" 3. User navigates to the main page
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	1.1.5
Test Case Name	
Category	Functional
Feature Description	Validate whether the user can navigate to the main page by giving the valid voice command for exit
Preconditions	<ul style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color 4. User must have given the valid login credentials

Test Description	1. Speak Up Following Commands - Select exit
Input Data	voice input - Select exit
Expected Result	1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "select exit" 3. User should be able to exit from the system
Actual Result	1. Microphone icon changes in to Green color 2. Speech Recognition status message should be change in to "select exit" 3. User exits from the system
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

- **Negative test cases**

Test Case ID	1.1.6
Category	Functional
Feature Description	Validate whether the user can navigate to the system using invalid credentials
Preconditions	1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color 4. User must have given the in valid login credentials in the xml file
Test Description	1. Speak Up Following Command

	- Select Sign in
Input Data	Refer xml, voice input “select sign in”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "Select sign in" 3. User should not be navigated to the main page 4. Following error message should be displayed "Invalid login details. Please try again"
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "Select sign in" 3. "Invalid login details. Please try again" error message is displayed, and user remains in the same page without being navigated to the main page.
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

2.0 Email Client Form

Test Case ID	2.0.1
Category	UI
Feature Description	Verify the look and feel of the Email Client form
Preconditions	User must be in Email Client form after a successful login
Test Description	<ol style="list-style-type: none"> 1. Verify the following fields under Email Client Form <ul style="list-style-type: none"> - Red color microphone icon - New<Button>

	<ul style="list-style-type: none"> - Inbox<Button> - Blank White Grid (Panel)
Input Data	N/A
Expected Result	<p>1. Following fields should be display under Email Client Form</p> <ul style="list-style-type: none"> -Red color microphone icon - New<Button> - Inbox<Button> - Blank White Grid (Panel)
Actual Result	The following fields are displayed - New<Button>, Inbox<Button> and Blank White Grid (Panel)
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	2.0.2
Category	Functional
Feature Description	Validate whether the user can navigate to the New email Section using voice commands
Preconditions	User must be in Email Client form after a successful login
Test Description	<p>1. Speak Up Following Command</p> <ul style="list-style-type: none"> - Select New
Input Data	Voice input "select new"
Expected Result	<p>1. Microphone icon should change in to Green color</p> <p>2. Speech Recognition status message should change in to "Select New"</p> <p>3. System should load NewEmailForm inside the panel in the main</p>

	page
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes to Green color 2. Speech Recognition status message changes in to "Select New" 3. System loads NewEmailForm inside the panel in the main page
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	2.0.3
Category	Functional
Feature Description	Validate whether the user can navigate to the Inbox Section using voice commands
Preconditions	User must be in Email Client form after a successful login
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <p>- Select Inbox</p>
Input Data	Voice input = "select inbox"
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be changed in to Green color 2. Speech Recognition status message should be changed in to "Select inbox" 3. System should load Inbox form inside the panel in the main page
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes to "Select inbox" 3. System loads Inbox form inside the panel in the main page
Speech Recognition Accuracy (Good/Fair/Low)	Good

Status (Pass/Fail)	Pass
--------------------	------

3.0 New Email

Test Case ID	3.0.1
Category	UI
Feature Description	Verify the look and feel of the new email
Preconditions	NewEmailForm is loaded inside the panel in the main page
Test Description	<p>1. verify following fields under new email</p> <ul style="list-style-type: none"> - Red Color Microphone - "Select New" Message - New<Button> - Inbox<Button> -To<Input Text Field> & <Add Button> (contacts button) -CC<Input Text Field> -BCC<Input Text Field> - Subject<Input Text Field> - Attachment<Input Text Field>& <Add Button> - Body - Send Email<Button>
Input Data	N/A
Expected Result	<p>1. User should be able to see following fields in the following conditions</p> <ul style="list-style-type: none"> - Red Color Microphone - "Select New" Message - New<Button> should be highlighted (focused)

	<ul style="list-style-type: none"> - Inbox<Button> 2. Following fields should be visible in the panel -To<Input Text Field>& <Add Button> -CC<Input Text Field> -BCC<Input Text Field> - Subject<Input Text Field> - Attachment<Input Text Field>& <Add Button> - Body - Send Email<Button>
Actual Result	<ul style="list-style-type: none"> 1. User is able to see following fields in the following conditions - Red Color Microphone - "Select New" Message - New<Button> should be highlighted (focused) - Inbox<Button> 2. Following fields are visible in the panel -To<Input Text Field>& <Add Button> -CC<Input Text Field> -BCC<Input Text Field> - Subject<Input Text Field> - Attachment<Input Text Field>& <Add Button> - Body - Send Email<Button>
Speech Recognition Accuracy (Good/Fair/Low)	N/A

Status (Pass/Fail)	Pass
--------------------	------

Test Case ID	3.0.2
Category	Functional
Feature Description	Validate whether the user can add contacts in to "TO" field using voice commands
Preconditions	NewEmailForm is loaded inside the panel in the main page
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <ul style="list-style-type: none"> - Select to 2. Speak Up Following Command <ul style="list-style-type: none"> - Select contact(s) 3. Speak Up Following Command <ul style="list-style-type: none"> - Include one 4. Speak Up Following Command <ul style="list-style-type: none"> -Select close
Input Data	The following voice inputs – “select contact(s)”, “select to”, “include one”, “select close”
Expected Result	<ol style="list-style-type: none"> 1 Contacts form should be open 2. It should contain the contacts in the web clients address book (Gmail contacts) 2. The row of the selected contact id should get highlighted. 3. Selected contact's email id should be display in "To field" 4. Contacts form closes.
Actual Result	<p>It gives the Expected results if the commands are given in the order of the test description.</p> <p>If the commands are given in ““select to”, select contact(s)”, “include one”, “select close” order it closes the main form instead</p>

	of the contacts form since the focus is in the main form.
Speech Recognition Accuracy (Good/Fair/Low)	“select contact(s)”, “select to”, “include one” - Good “select close” – Low (close often recognized as tools)
Status (Pass/Fail)	Pass

Test Case ID	3.0.3
Category	Functional
Feature Description	Validate whether the user can add contacts in to "CC" field
Preconditions	NewEmailForm is loaded inside the panel in the main page
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <ul style="list-style-type: none"> - Select cc 2. Speak Up Following Command <ul style="list-style-type: none"> - Select contact(s) 3. Speak Up Following Command <ul style="list-style-type: none"> - Include one 4. Speak Up Following Command <ul style="list-style-type: none"> -Select close
Input Data	The following voice inputs – “select contact(s)”, “select cc”, “include one”, “select close”
Expected Result	<ol style="list-style-type: none"> 1 Contacts form should be open 2. It should contain the contacts in the web clients address book (Gmail contacts) 2. The row of the selected contact id should get highlighted. 3. Selected contact's email id should be display in "CC field" 4. contacts form closes.

Actual Result	It gives the Expected results if the commands are given in the order of the test description. If the commands are given in ““select cc”, select contact(s)”, “include one”, “select close” order it closes the main form instead of the contacts form since the focus is in the main form.
Speech Recognition Accuracy (Good/Fair/Low)	“select contact(s)”, “select cc”, “include one” - Fair “select close” – Low (close often recognized as tools)
Status (Pass/Fail)	Pass

Test Case ID	3.0.4
Category	Functional
Feature Description	Validate whether the user can add contacts in to "BCC" field using voice commands
Preconditions	NewEmailForm is loaded inside the panel in the main page
Test Description	1. Speak Up Following Command - Select bcc 2. Speak Up Following Command - Select contact(s) 3. Speak Up Following Command - Include one 4. Speak Up Following Command -Select close
Input Data	The following voice inputs – “select contact(s)”, “select bcc”, “include one”, “select close”
Expected Result	1 Contacts form should be open 2. It should contain the contacts in the web clients address book

	<p>(Gmail contacts)</p> <ol style="list-style-type: none"> 2. The row of the selected contact id should get highlighted. 3. Selected contact's email id should be display in "BCC field" 4. contacts form closes.
Actual Result	<p>It gives the Expected results if the commands are given in the order of the test description.</p> <p>If the commands are given in ““select cc”, select contact(s)”, “include one”, “select close” order it closes the main form instead of the contacts form since the focus is in the main form.</p>
Speech Recognition Accuracy (Good/Fair/Low)	<p>“select contact(s)”, “select bcc”, “include one” - Fair</p> <p>“select close” – Low (close often recognized as tools)</p>
Status (Pass/Fail)	Pass

Test Case ID	3.0.5
Category	Functional
Feature Description	Validate whether the user can dictate to the "Subject" field
Preconditions	NewEmailForm is loaded inside the panel in the main page
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command - Select Subject 2. Say anything what you want to put as subject
Input Data	<p>Voice inputs</p> <p>-“select subject”</p> <p>-“this is the subject” (can be anything)</p>
Expected Result	<ol style="list-style-type: none"> 1. Curser should be focus to "Subject" field 2. Following Message should be display in Subject field

	"this is the Subject"
Actual Result	1. Curser focuses to "Subject" field 2. "this is the subject" displays in Subject field
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	3.0.6
Category	Functional
Feature Description	Validate whether the user can add an attachment in to "Attachment" field
Preconditions	NewEmailForm is loaded inside the panel in the main page
Test Description	1. Speak Up Following Command - Select Attachment 1. Speak Up Following Command - Attach one 1. Speak Up Following Command - Select close
Input Data	Voice input – “select attachment(s)”, “attach one”, “select close”
Expected Result	1.Attachments form should be open 2. Is should contain the files of the ‘Attachment’ folder 3.The row of the selected contact id should get highlighted.(row one) 4. the path of the attachment id should display on the Attachment Field (attachment id one)

Actual Result	<ol style="list-style-type: none"> 1. Attachments form opens 2. It contains the files of the 'Attachment' folder 3. The row one gets highlighted. 4. the path of attachment one displays on the Attachment Fields
Speech Recognition Accuracy (Good/Fair/Low)	<p>“select attachment(s)”, “attach one” - Good</p> <p>“select close” – Low (close often recognized as tools)”</p>
Status (Pass/Fail)	Pass

Test Case ID	3.0.7
Category	Functional
Feature Description	Validate whether the user can dictate to the body field
Preconditions	NewEmailForm is loaded inside the panel in the main page
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command - Select body 2. Say anything what you want to put as the body
Input Data	<p>Voice inputs</p> <p>-“select body”</p> <p>-“this is the body” (can be anything)</p>
Expected Result	<ol style="list-style-type: none"> 1. Curser should be focus to "Body" field 2. Following message should be display in Body Section <p>This is the Body</p>
Actual Result	<ol style="list-style-type: none"> 1. Curser focuses to "Body" field 2. "this is the body" displays in Body field
Speech Recognition Accuracy	Fair

(Good/Fair/Low)	
Status (Pass/Fail)	Pass

Test Case ID	3.0.8
Category	
Feature Description	Validate whether the user can send an email successfully using voice commands
Preconditions	1.NewEmailForm is loaded inside the panel in the main page 2. 'To' Field must be filled
Test Description	1. Speak Up Following Command - Select send 2. Speak Up Following Command - Select close (to close sent confirmation message)
Input Data	Voice input – “select send”, “select close”
Expected Result	1.Email should be sent and a new window should pop up with message "Email Sent" (If 'To' field is not filled, system throws error “ the parameter address cannot be an empty string”)
Actual Result	When email is sent system shows confirmation message in new window displaying “Email Sent”
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

4.0 Inbox

Test Case ID	4.0.1
Category	
Feature Description	Verify the look and feel of the inbox
Preconditions	System must loads Inbox inside the panel in the main page
Test Description	<p>1. verify following fields</p> <ul style="list-style-type: none"> - Red Color Microphone - "Select Inbox" Message - New<Button> - Inbox<Button -Open <Button> - Delete<Button> - Undder Inbox Grid <ul style="list-style-type: none"> - Email ID - Sender - Subject - Body - Date - Reply<Button> -Reply all <Button> - Forward<Button> - Empty Body
Input Data	N/A
Expected Result	<p>1. User should be able to display following fields</p> <ul style="list-style-type: none"> - Red Color Microphone

	<ul style="list-style-type: none"> - "Select New" Message - New<Button> - Inbox<Button>should be highlighted <p>2. Following fields should be display under inbox</p> <ul style="list-style-type: none"> -Open <Button> - Delete<Button> - Undder Inbox Grid <ul style="list-style-type: none"> - Email ID - Sender - Subject - Body - Date - Reply<Button> -Reply all <Button> - Forward<Button> - Empty Body
Actual Result	All buttons and fields display as required
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	4.0.2
Category	Functional
Feature Description	Validate whether the user can open an email successfully using voice commands

Preconditions	1.Gmail inbox page is loaded in the system main page
Test Description	1. Speak Up Following Command to third email - Open three
Input Data	Voice input – “open three”
Expected Result	1.Third email should be open and email body should be display under body section 2. Above body the follwing field should display; -sender - subject - date and time sent
Actual Result	The email opens and the fields sender, subject, date and time are displayed on the top while the body is displayed at the bottom
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.3
Category	Functional
Feature Description	Validate whether the user can reply to an email successfully using voice commands
Preconditions	1.Gmail inbox page is loaded in the system main page 2.Selected email must be open
Test Description	1. Speak Up Following Command to third email - email reply 2. Speak Up Following Command

	<ul style="list-style-type: none"> - Select body 3. Say anything what you want to put as subject -“This is the reply ” 4. Speak Up Following Command to third email - select send
Input Data	<ul style="list-style-type: none"> 1.Voice input – “email reply” 2. Voice input – “This is the reply ”
Expected Result	<ul style="list-style-type: none"> 1.Reply email should be open and “To” field should include the sender and “Subject” field include the “RE:” followed by “Subject” 2..Following message should be display under body section “This is the reply” above the original body. 3..once email is sent display messages “Email Sent”
Actual Result	The “To” and “Subject” fields are filled properly. Dictation occurs correctly above the original body and once email is sent the messages “Email Sent” is displayed.
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.4
Category	Functional
Feature Description	Validate whether the user can forward an email successfully using voice commands
Preconditions	<ul style="list-style-type: none"> 1.Gmail inbox page is loaded in the system main page 2.Selected email must be open

Test Description	<p>1. Speak Up Following Command to third email</p> <ul style="list-style-type: none"> - email forward <p>2. Speak Up Following Command</p> <ul style="list-style-type: none"> - Select to <p>3. Speak Up Following Command</p> <ul style="list-style-type: none"> - Select contact(s) <p>4. Speak Up Following Command</p> <ul style="list-style-type: none"> - Include one <p>5. Speak Up Following Command</p> <ul style="list-style-type: none"> -Select close <p>5. Speak Up Following Command to third email</p> <ul style="list-style-type: none"> - select send
Input Data	The following voice inputs – “select contact(s)”, “select to”, “include one”, “select close”
Expected Result	<p>1.Email should be open and only “Subject” and “Body” fields should be filled</p> <p>2 Separate Contacts form should be open with Contact ID allocated for each Contact</p> <p>3. The row of the selected contact id should get highlighted.</p> <p>4. Selected contact's email id should be display in "To field"</p> <p>5. Close contacts form.</p> <p>6.once email is sent, system should navigate to inbox</p> <p>7. Show message ”Email Sent”</p>
Actual Result	Subject is filled as “FW:” followed by the original subject and original body is displayed in the body field. Contacts get added correctly and once the email is sent the message “Email Sent is

	displayed”
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.5
Category	Functional
Feature Description	Validate whether the user can do reply all for an email successfully using voice commands
Preconditions	1.Gmail inbox page is loaded in the system main page 2.Selected email must be open
Test Description	1. Speak Up Following Command to third email - email reply all 2. Speak Up Following Command - Select body 3. Say anything what you want to put as subject -“This is the reply ” 4. Speak Up Following Command to third email - select send
Input Data	1.Voice input – “email reply all” 2. Voice input – “This is reply all ”
Expected Result	1.Reply email should be open and “To” field should include the sender and “Subject” field include the “RE:” followed by “Subject” 2..Following message should be display under body section

	<p>“This is the reply” above the original body.</p> <p>3..once email is sent display messages “Email Sent”t”</p>
Actual Result	The “To” (includes all replying addresses) and “Subject” fields are filled properly. Dictation occurs correctly above the original body and once email is sent the messages “Email Sent” is displayed.
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.6
Category	Functional
Feature Description	Validate whether the user can delete an email successfully using voice commands
Preconditions	1.Gmail inbox page is loaded in the system main page
Test Description	<p>1. Speak Up Following Command to third email</p> <p>- Delete three</p> <p>2. Speak Up Following Command to third email</p> <p>- Select Ok</p>
Input Data	<p>Voice input – “Delete three”</p> <p>Voice input- “Select ok</p>
Expected Result	<p>1.Third email should be Deleted and following pop up message should be displayed</p> <p>-“Message deleted sucessfully.reloading inbox””</p> <p>2. Message should be deleted from the inbox</p>
Actual Result	The third email get deleted and is confirmed by the “Message

	deleted successfully.reloading inbox”
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	4.0.7
Test Case Name	
Category	Functional
Feature Description	Validate whether the user can sign out from the system by giving the voice command
Preconditions	<ol style="list-style-type: none"> 1. User must be in Main Form/Page 2. User must have a microphone 3. Microphone must be red color 4. User must have given the valid login credentials
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Commands <p>- Select Sign out</p>
Input Data	voice input – “Select Sign out”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "Select sign out" 3. Application should be closed
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. System doesn't recognize the command “select sign out”
Speech Recognition Accuracy (Good/Fair/Low)	Low (very low)

Status (Pass/Fail)	Fail
--------------------	------

6.4.1 Gmail interface

1.0 Launch the system (Voice Driven Email Client)

Test Case ID	1.0.1
Category	Functional
Feature Description	Validate whether the user can launch the system (Voice Driven Email Client)
Preconditions	<ol style="list-style-type: none"> 1. User must have web-email account (Gmail) with valid login credentials 2. The username, password saved on the 'LoginCredentials.xml' file 4. User must have access to the Internet 5. User must have a microphone
Test Description	1. Double click on the system
Input Data	N/A
Expected Result	1. User should be able to launch the system
Actual Result	User is able to successfully launch the system
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	1.0.2
Category	UI
Feature Description	Verify the look and feel of the Login Form/Page
Preconditions	1. User must be in startup screen
Test Description	1. Verify the start up screen
Input Data	N/A

Expected Result	<ol style="list-style-type: none"> 1. System should starts up with Splash Screen which stays for 5 seconds 2. System should navigate to the Gmail Login Form/Page after 5 seconds
Actual Result	System starts up with Splash screen which stays for 5 seconds are navigates to the login form
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	1.0.3
Category	UI
Feature Description	Verify the look and feel of the Login Form/Page
Preconditions	1. User must be in Login Form/Page
Test Description	<ol style="list-style-type: none"> 1. Verify following fields under Login Form/Page <ul style="list-style-type: none"> - Red color microphone icon - "Start Speaking" message (Speech Recognition status message) - Gmail Login interface below it
Input Data	N/A
Expected Result	<ol style="list-style-type: none"> 1. User Should be able to see following fields under Gmail Login Form/Page <ul style="list-style-type: none"> - Red color microphone icon should display left top side of the Login Form --"Start Speaking" message should display at right side of the microphone icon - Gmail Login interface below it
Actual Result	- Red color microphone icon is displayed on left top side

	- Gmail Login interface is displayed below it
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	1.0.4
Category	Functional
Feature Description	Validate whether the system can recognize the user's speech
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color 4. "Start Speaking" Message must be display
Test Description	<ol style="list-style-type: none"> 1. Speak something using microphone EX:" hello world "
Input Data	Voice input – ‘hello world’
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. "Start speaking" message (Speech Recognition status message) should be change in to "hello world" 3. after 2 seconds microphone should change back to red color
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon is changed to green and the message changes to “hello world” 2. after 2 seconds microphone should change back to red color
Speech Recognition Accuracy (Good/Fair/Low)	Good

Status (Pass/Fail)	Pass
--------------------	------

1.1 Login to the Main Page (Email Client Page)

Test Case ID	1.1.1
Category	Functional
Feature Description	Validate whether the user can input username by giving a valid voice command
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <p>- Include Username</p>
Input Data	Refer xml, voice input – “include username”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "include username" 3. User name which is include in xml file should be display in username input text field
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "include username" 3. username input text field displays “voicedrivenemailclient@gmail.com”
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	1.1.2
Category	Functional
Feature Description	Validate whether the user can input password by giving a valid voice command
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <p>- Include password</p>
Input Data	Refer xml, voice input – “include password”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "include password" 3. password which is included in xml file should be display in password input text field
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "include password" 3. username input text field displays username in dots
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	1.1.3
Test Case Name	
Category	Functional

Feature Description	Validate whether the user can navigate to the main page by giving the valid command for sign in
Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color 4. User must have given the valid login credentials
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Commands <p>- Select Sign in</p>
Input Data	voice input - Select Sign in
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "Select sign in" 3. User should be navigate to the main page
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "select sign in" 3. User navigates to the Gmail main page
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Negative test cases

Test Case ID	1.1.6
Category	Functional
Feature Description	Validate whether the user can navigate to the system using invalid credentials

Preconditions	<ol style="list-style-type: none"> 1. User must be in Login Form/Page 2. User must have a microphone 3. Microphone must be red color 4. User must have given the in valid login credentials in the xml file
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <p>- Select Sign in</p>
Input Data	Refer xml, voice input “select sign in”
Expected Result	<ol style="list-style-type: none"> 1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "Select sign in" 3. User should not be navigated to the main page 4. Gmail error message should be displayed
Actual Result	<ol style="list-style-type: none"> 1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "Select sign in" 3. "The username or password you entered is incorrect" error message is displayed, and user remains in the same page without being navigated to the main page.
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

2.0 Email Client Form

Test Case ID	2.0.1
Category	UI
Feature Description	Verify the look and feel of the Email Client form

Preconditions	User must be in Email Client form after a successful login
Test Description	1. Verify the following fields under Email Client Form <ul style="list-style-type: none"> - Red color microphone icon, speech recognition status message - Gmail main page below it
Input Data	N/A
Expected Result	1. Following fields should be display under Email Client Form <ul style="list-style-type: none"> - Red color microphone icon, speech recognition status message - Gmail main page below it
Actual Result	The Red color microphone icon, and speech recognition status message are displayed on the top and the Gmail main page below it
Speech Recognition Accuracy (Good/Fair/Low)	N/A
Status (Pass/Fail)	Pass

Test Case ID	2.0.2
Category	Functional
Feature Description	Validate whether the user can navigate to the New email Section using voice commands
Preconditions	User must be in Email Client form after a successful login
Test Description	1. Speak Up Following Command <ul style="list-style-type: none"> - Select New
Input Data	Voice input "select new"
Expected Result	1. Microphone icon should change in to Green color 2. Speech Recognition status message should change in to "Select New"

	3.System should load Gmail new email form
Actual Result	1. Microphone icon changes to Green color 2. Speech Recognition status message changes in to "Select New" 3.System loads Gmail new email form
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	2.0.3
Category	Functional
Feature Description	Validate whether the user can navigate to the Inbox Section using voice commands
Preconditions	User must be in Email Client form after a successful login
Test Description	1. Speak Up Following Command - Select Inbox
Input Data	Voice input = “select inbox”
Expected Result	1. Microphone icon should be changed in to Green color 2. Speech Recognition status message should be changed in to "Select inbox" 3.System should load separate Inbox form with a message ID allocated to each email
Actual Result	1. Microphone icon changes in to Green color 2. Speech Recognition status message changes to "Select inbox" 3.System loads separate Inbox form with a Email ID allocated to each email
Speech Recognition	Good

Accuracy (Good/Fair/Low)	
Status (Pass/Fail)	Pass

3.0 New Email

Test Case ID	3.0.1
Category	Functional
Feature Description	Validate whether the user can add contacts in to "TO" field using voice commands
Preconditions	Gmail New email page is loaded in the system main page
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <ul style="list-style-type: none"> - Select to 2. Speak Up Following Command <ul style="list-style-type: none"> - Select contact(s) 3. Speak Up Following Command <ul style="list-style-type: none"> - Include one 4. Speak Up Following Command <ul style="list-style-type: none"> -Select close
Input Data	The following voice inputs – “select contact(s)”, “select to”, “include one”, “select close”
Expected Result	<ol style="list-style-type: none"> 1 Separate Contacts form should be open with Contact ID allocated for each Contact 2. The row of the selected contact id should get highlighted. 3. Selected contact's email id should be display in "To field" 4. Close contacts form.

Actual Result	<p>It gives the Expected results if the commands are given in the order of the test description.</p> <p>If the commands are given in ““select to”, select contact(s)”, “include one”, “select close” order it closes the main form instead of the contacts form since the focus is in the main form.</p>
Speech Recognition Accuracy (Good/Fair/Low)	<p>“select contact(s)”, “select to”, “include one” - Good</p> <p>“select close” – Low (close often recognized as tools)</p>
Status (Pass/Fail)	Pass

Test Case ID	3.0.2
Category	Functional
Feature Description	Validate whether the user can add contacts in to "CC" field
Preconditions	Gmail New email page is loaded in the system main page
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command <ul style="list-style-type: none"> - Select cc 2. Speak Up Following Command <ul style="list-style-type: none"> - Select contact(s) 3. Speak Up Following Command <ul style="list-style-type: none"> - Include one 4. Speak Up Following Command <ul style="list-style-type: none"> -Select close
Input Data	The following voice inputs – “select contact(s)”, “select cc”, “include one”, “select close”
Expected Result	<ol style="list-style-type: none"> 1 Separate Contacts form should be open with Contact ID allocated for each Contact 2. The row of the selected contact id should get highlighted.

	<p>3. Selected contact's email id should be display in "CC field"</p> <p>4. Close contacts form</p>
Actual Result	<p>It gives the Expected results if the commands are given in the order of the test description.</p> <p>If the commands are given in ““select cc”, select contact(s)”, “include one”, “select close” order it closes the main form instead of the contacts form since the focus is in the main form.</p>
Speech Recognition Accuracy (Good/Fair/Low)	<p>“select contact(s)”, “select cc”, “include one” - Fair</p> <p>“select close” – Low (close often recognized as tools)</p>
Status (Pass/Fail)	Pass

Test Case ID	3.0.3
Category	Functional
Feature Description	Validate whether the user can add contacts in to "BCC" field using voice commands
Preconditions	Gmail New email page is loaded in the system main page
Test Description	<p>1. Speak Up Following Command</p> <p>- Select bcc</p> <p>2. Speak Up Following Command</p> <p>- Select contact(s)</p> <p>3. Speak Up Following Command</p> <p>- Include one</p> <p>4. Speak Up Following Command</p> <p>-Select close</p>
Input Data	The following voice inputs – “select contact(s)”, “select bcc”, “include one”, “select close”

Expected Result	<ol style="list-style-type: none"> 1 Separate Contacts form should be open with Contact ID allocated for each Contact 2. The row of the selected contact id should get highlighted. 3. Selected contact's email id should be display in "BCC field" 4. Close contacts form.
Actual Result	<p>It gives the Expected results if the commands are given in the order of the test description.</p> <p>If the commands are given in ““select cc”, select contact(s)”, “include one”, “select close” order it closes the main form instead of the contacts form since the focus is in the main form.</p>
Speech Recognition Accuracy (Good/Fair/Low)	<p>“select contact(s)”, “select bcc”, “include one” - Fair</p> <p>“select close” – Low (close often recognized as tools)</p>
Status (Pass/Fail)	Pass

Test Case ID	3.0.4
Category	Functional
Feature Description	Validate whether the user can dictate to the "Subject" field
Preconditions	Gmail New email page is loaded in the system main page
Test Description	<ol style="list-style-type: none"> 1. Speak Up Following Command - Select Subject 2. Say any thing what you want to put as subject
Input Data	<p>Voice inputs</p> <p>-“select subject”</p> <p>-“this is the subject” (can be anything)</p>
Expected Result	<ol style="list-style-type: none"> 1. Curser should be focus to "Subject" field 2. Following Message should be display in Subject field

	"this is the Subject"
Actual Result	1. Curser focuses to "Subject" field 2. "this is the subject" displays in Subject field
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	3.0.5
Category	Functional
Feature Description	Validate whether the user can dictate to the body field
Preconditions	Gmail New email page is loaded in the system main page
Test Description	1. Speak Up Following Command - Select body 2.Say anything what you want to put as the body
Input Data	Voice inputs -“select body” -“this is the body” (can be anything)
Expected Result	1. Curser should be focus to "Body" field 2. Following message should be display in Body Section This is the Body
Actual Result	1. Curser focuses to "Body" field 2. "this is the body" displays in Body field
Speech Recognition Accuracy (Good/Fair/Low)	Fair

Status (Pass/Fail)	Pass
--------------------	------

Test Case ID	3.0.6
Category	
Feature Description	Validate whether the user can send an email successfully using voice commands
Preconditions	1.Gmail New email page is loaded in the system main page 2. 'To' Field must be filled
Test Description	1. Speak Up Following Command - Select send
Input Data	Voice input – “select send”
Expected Result	1.once email is sent, system should navigate to inbox 2. Show message ”Your message has been sent”
Actual Result	When email is sent system navigates and displays confirmation message
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

4.0 Inbox

Test Case ID	4.0.1
Category	Functional
Feature Description	Validate whether the user can open an email successfully using voice commands
Preconditions	1.Gmail inbox page is loaded in the system main page

Test Description	1. Speak Up Following Command to third email - Open three
Input Data	Voice input – “open three”
Expected Result	1.Third email should be open and page should navigate to opened email page of Gmail
Actual Result	Email with id 3 opens in the Gmail ‘opened email page’
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.2
Category	Functional
Feature Description	Validate whether the user can reply to an email successfully using voice commands
Preconditions	1.Gmail ‘email opened page’ is loaded displaying the selected email
Test Description	1. Speak Up Following Command to third email - email reply 2. Speak Up Following Command - Select body 3. Say anything what you want to put as subject -“This is the reply ” 4. Speak Up Following Command to third email - select send
Input Data	1.Voice input – “email reply” 2. Voice input – “This is the reply ”

Expected Result	<p>1.Reply email should be open and “To” field should include the sender and “Subject” field include the “RE:” followed by “Subject”</p> <p>2..Following message should be display under body section “This is the reply” above the original body.</p> <p>3..once email is sent system should navigate to inbox</p>
Actual Result	The “To” and “Subject” fields are filled properly. Dictation occurs correctly above the original body and once email is system navigates to Gmail inbox
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.3
Category	Functional
Feature Description	Validate whether the user can forward an email successfully using voice commands
Preconditions	1.Gmail ‘email opened page’ is loaded displaying the selected email
Test Description	<p>1. Speak Up Following Command to third email</p> <ul style="list-style-type: none"> - email forward <p>2. Speak Up Following Command</p> <ul style="list-style-type: none"> - Select to <p>3. Speak Up Following Command</p> <ul style="list-style-type: none"> - Select contact(s) <p>4. Speak Up Following Command</p> <ul style="list-style-type: none"> - Include one <p>5. Speak Up Following Command</p>

	<p>-Select close</p> <p>5. Speak Up Following Command to third email</p> <p>- select send</p>
Input Data	The following voice inputs – “select contact(s)”, “select to”, “include one”, “select close”
Expected Result	<p>1.Email should be open and only “Subject” and “Body” fields should be filled</p> <p>2 Separate Contacts form should be open with Contact ID allocated for each Contact</p> <p>3. The row of the selected contact id should get highlighted.</p> <p>4. Selected contact's email id should be display in "To field"</p> <p>5. Close contacts form.</p> <p>6.once email is sent, system should navigate to inbox</p>
Actual Result	Subject is filled as “FW:” followed by the original subject and original body is displayed in the body field. Contacts get added correctly and once the email is sent the message system navigates to inbox
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.4
Category	Functional
Feature Description	Validate whether the user can do reply all for an email successfully using voice commands
Preconditions	1.Gmail ‘email opened page’ is loaded displaying the selected email

Test Description	<p>1. Speak Up Following Command to third email</p> <p>- email reply all</p> <p>2. Speak Up Following Command</p> <p>- Select body</p> <p>3. Say anything what you want to put as subject</p> <p>-“This is the reply ”</p> <p>4. Speak Up Following Command to third email</p> <p>- select send</p>
Input Data	<p>1.Voice input – “email reply all”</p> <p>2. Voice loaded in the system main page input – “This is reply all ”</p>
Expected Result	<p>1.Reply email should be open and “To” field should include the sender and “Subject” field include the “RE:” followed by “Subject”</p> <p>2..Following message should be display under body section</p> <p>“This is the reply” above the original body.</p> <p>3..once email is sent system should navigate to inbox</p>
Actual Result	The “To” (includes all replying addresses) and “Subject” fields are filled properly. Dictation occurs correctly above the original body and once email is sent system navigates to inbox
Speech Recognition Accuracy (Good/Fair/Low)	Fair
Status (Pass/Fail)	Pass

Test Case ID	4.0.5
Category	Functional

Feature Description	Validate whether the user can delete an email successfully using voice commands
Preconditions	1.Gmail 'email opened page' is loaded displaying the selected email
Test Description	1. Speak Up Following Command to delete opened email - email delete
Input Data	Voice input – “email delete”
Expected Result	1.Third email should be Deleted and the system should naviagte to the inbox 2. the messages “The message has been moved to the Trash”should disaply on the top of the inbox 2. Message should be deleted from the inbox
Actual Result	The third email get deleted and is confirmed by the “The message has been moved to the Trash”
Speech Recognition Accuracy (Good/Fair/Low)	Good
Status (Pass/Fail)	Pass

Test Case ID	4.0.6
Test Case Name	
Category	Functional
Feature Description	Validate whether the user can sign out from the system by giving the voice command
Preconditions	1. User must be in Main Form/Page 2. User must have a microphone 3. Microphone must be red color 4. User must have given the valid login credentials

Test Description	1. Speak Up Following Commands - Select Sign out
Input Data	voice input – “Select Sign out”
Expected Result	1. Microphone icon should be change in to Green color 2. Speech Recognition status message should be change in to "Select sign out" 3. Application should be closed
Actual Result	1. Microphone icon changes in to Green color 2. Speech Recognition status message changes in to "select sign out" 3. Application closed
Speech Recognition Accuracy (Good/Fair/Low)	Low (very low)
Status (Pass/Fail)	Pass

6.5 Performance Testing

Performance Testing was carried out for the following functions of the system;

- Time taken for login
 - Obtained an average of 3.7s

Attempt Number	Time taken for login (in seconds)
1	4.8
2	3.3
3	3.7
4	3.6
5	3.1

- Time taken for retrieving inbox messages
 - Obtained an average of 24.5s

Attempt Number	Time taken for retrieving inbox messages (in seconds)
1	23.2
2	25.8
3	22.9
4	26.7
5	23.7

- Dictating long sentences
 - It can on average capture up to maximum of 15 words
 - The sentence tested: “I would like to express my interest in a position as editorial assistant your publishing company”
 - 1st try: I would like to express my interest in a position as they did audio the system
 - 5nd try: I would like to express my interest in a position as editorial assistant fire your publishing company

- More the training, better the recognition

Chapter 7 Deployment

7.1 Chapter overview

This chapter outlines the deployment process of the system. It includes setting up the system (both versions).

7.2 Setting up the system

7.2.1 Voice Driven Email Client Version – 1(Separate Interface)

- The following changes should be made to Gmail before logging in to the system
 - Login in to Gmail
 - To stop Gmail messages with the same subject from grouping together following the below step;
 - Go to ‘General’ tab of the Gmail settings
 - Select radio button ‘Conversation view off’
 - Select POP for retrieving Emails
 - Go to settings
 - Select tab “Forwarding and POP/IMAP”
 - Under “POP Download”:
 1. “Status”: select “Enable POP for all mail (even mail that's already been downloaded)”
 2. “When messages are accessed with POP” select “archive Gmail’s copy / delete Gmail’s copy (Default setting – keep Gmail’s copy in the inbox

- The login credentials have to saved on to the LoginCredentials.xml


```
<?xml version="1.0"?>
<credentials>
  <username>voicedrivenemailclient@gmail.com</username>
  <password>password</password>
  <webemailclient>Gmail</webemailclient>
</credentials>
```

7.2.2 Voice Driven Email Client Version – 2 (Gmail Interface)

- The following changes should be made to Gmail before logging in to the system
 - Login in to Gmail
 - To stop Gmail messages with the same subject from grouping together following the below step;
 - Go to ‘General’ tab of the Gmail settings
 - Select radio button ‘Conversation view off’
 - Select POP for retrieving Emails
 - Go to settings
 - Select tab “Forwarding and POP/IMAP”
 - Under “POP Download”:
 1. “Status”: select “Enable POP for all mail (even mail that's already been downloaded)”
 2. “When messages are accessed with POP” select “archive Gmail’s copy / delete Gmail’s copy (Default setting – keep Gmail’s copy in the inbox
 - Set html view as default
 - Go to url: <https://mail.google.com/mail/h/>
 - Click on “Set basic HTML as default view” displayed on the top of the page
- The login credentials have to saved on to the LoginCredentials.xml


```
<?xml version="1.0"?>
<credentials>
  <username>voicedrivenemailclient@gmail.com</username>
  <password>password</password>
</credentials>
```

Chapter 8 Evaluation

8.1 Limitations

- The system currently allows the user to connect to only Gmail
- System displays all words in lower-case English
- The system is unable to allow users to dictate email addresses
 - Email addresses may contain special characters and numbers.
- User can add only one attachment to an email
- Allows undoing while dictating only up-to one level back
- Cannot download attachments
- Cannot type in the body of the email, only dictate
- Only files that are contained in the 'Attachments' Folder can be attached to an email

8.2 Future Enhancements

- Support connecting to any web-based email client such as yahoo, hotmail etc.
- Support speaker independency
- Have system as a plug-in
- Support dictating email addresses
- Support natural language processing
- Switch from POP3 to IMAP4 for retrieving emails from the web client(Gmail)
- Add all the remaining functions that are included in web email clients into the system

8.3 Conclusion

The proposed system was successfully implementing during the specified time period meeting its requirements. Though the initial scope was to develop a system with a separate user interface design, at the end of the project two versions of the system was developed.

- Version 1 – A Voice Driven Email Client with a separate user interface design
- Version 2 – A Voice Driven Email Client which communicates with the Gmail Interface. Gmail is loaded inside a 'web browser control' in the system and is driven using the user's voice.

The system investigations, technical investigations, relevant background analysis carried out made it possible to identify the exact requirements of the systems and best practices to be used. The background research further helped to identify drawbacks of existing applications and eliminate them in order to create a more useful system.

The system was designed according to the recognized architecture (3-tier architecture) which made the design phase as well as the development easy. UML modeling techniques were used to

document the design. All of this helped to create a manageable code. Speech Recognition was implemented using a Speech API instead of re-inventing the wheel.

Using all of the above mentioned techniques it was possible to implement a Voice Driven Email Client that enables the user to connect to Gmail and perform actions included in the scope successfully.

Chapter 9 Appendix

9.1 References

- Google Chrome. n.d. *Chrome speech input 2011*. [image online] Available at: <http://www.thechromesource.com/wp-content/uploads/2011/03/chromespeechinput.png> [Accessed 10th August, 2012]
- Media Wiley. n.d. *Dictate Emails*. [image online] Available at: <http://media.wiley.com/Lux/01/317901.image1.jpg> [Accessed 12th April, 2012]
- Microsoft. n.d. *SAPI Architecture*. [image online] Available at: <http://i.msdn.microsoft.com/dynimg/IC27245.jpg> [Accessed 7th July, 2012]
- Source Forge. n.d. *Sphinx Architecture*. [image online] Available at: <http://cmusphinx.sourceforge.net/sphinx4/doc-files/architecture.gif> [Accessed 11th July, 2012]
- The Radicate Group, Inc. 2011. *Email-Statistics-Report-2011-2015-Executive-Summary*. Available at: <http://www.radicati.com/wp/wp-content/uploads/2011/05/Email-Statistics-Report-2011-2015-Executive-Summary.pdf> [online] Available at: <http://www.radicati.com/wp/wp-content/uploads/2011/05/Email-Statistics-Report-2011-2015-Executive-Summary.pdf> [Accessed 5th July, 2012]

9.2 Bibliography

- Anon. n.d. *Dragon - Continuous vs. Discrete Speech Recognition* [online] Available at: <http://www.synapseadaptive.com/naturallyspeaking/define.html> [Accessed 4th July, 2012]
- Anon. n.d. *How to Use Siri for Writing and Sending E-Mail - For Dummies* [online] Available at: <http://www.dummies.com/how-to/content/how-to-use-siri-for-writing-and-sending-email.html> [Accessed 12th April, 2012]
- Anon. n.d. *Speech Recognition* [online] Available at: <http://www.learnartificialneuralnetworks.com/speechrecognition.html> [Accessed 18th May, 2012]
- Anon. n.d. *Speech Tips, Types of Speech Recognition* [online] Available at: <http://www.lumenvox.com/resources/tips/types-of-speech-recognition.aspx> [Accessed 16th May, 2012]
- Anon. n.d. *Voice Recognition* [online] Available at: <http://www.answers.com/topic/voice-recognition> [Accessed 3rd June, 2012]
- Apple Inc. n.d. *Apple - iOS 6 - Use your voice to do even more with Siri.* [online] Available at: <http://www.apple.com/ios/siri/> [Accessed 11th April, 2012]
- Google Chrome. n.d. *Chrome.experimental.speechInput - Google Chrome* [online] Available at: <http://developer.chrome.com/extensions/experimental.speechInput.html> [Accessed 10th August, 2012]
- Google Chrome. n.d. *Speech-input* [online] Available at: <http://slides.html5rocks.com/#speech-input> [Accessed 10th August, 2012]
- Hasnat Md. Abul. n.d. *Isolated and Continuous Bangla Speech Recognition: Implementation, Performance and Application Perspective* | [online] Available at: http://www.academia.edu/670292/Isolated_and_continuous_bangla_speech_recognition_implementation_performance_and_application_perspective [Accessed 17th July, 2012]

- Josh Write. 2010 *Sending & Receiving Email in CSharp* [online] Available at: <http://bendytree.com/tips/Sending-Receiving-Email-in-CSharp> [Accessed 12th April, 2012]
- Microsoft. n.d. *User Interface Technologies* [online] Available at: [http://msdn.microsoft.com/en-us/library/windows/desktop/ff728830\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ff728830(v=vs.85).aspx) [Accessed 29th May, 2012]
- Microsoft.com. n.d. *Developing for Speech* [online] Available at: <http://msdn.microsoft.com/en-us/library/windows/desktop/bb756992.aspx> [Accessed 29th May, 2012]
- Microsoft.com. n.d. *Multitier Architecture - Wikipedia, the free encyclopedia* [online] Available at: http://en.wikipedia.org/wiki/Multitier_architecture [Accessed 23rd August, 2012]
- Microsoft. n.d. *SmtpClient Class (System.Net.Mail)* [online] Available at: <http://msdn.microsoft.com/en-us/library/system.net.mail.smtpclient.aspx> [Accessed 11th April, 2012]
- Microsoft. n.d. *Speech API Overview (SAPI 5.4)* [online] Available at: [http://msdn.microsoft.com/en-us/library/ee125077\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ee125077(v=vs.85).aspx) [Accessed 29th May, 2012]
- Nuance. n.d. *Client SDK - Nuance* [online] Available at: <http://www.nuance.com/for-developers/dragon/client-sdk/index.htm> [Accessed June 3, 2012]
Nuance. n.d. *Dragon Naturally Speaking 11 Datasheet*. [online] Available at: http://www.nuance.com/ucmprod/groups/dragon/@web/documents/webasset/nc_008811.pdf [Accessed 8th September, 2012]
- Nuance.com.. n.d. *Dragon NaturallySpeaking 10 SDK Frequently Asked Questions* [online] Available at: <http://www.nuance.com/naturallyspeaking/products/sdk/SDK-frequently-asked-questions.asp> [Accessed 8th, 2012]
- Satish Sampath. 2011. *Google Chrome Blog: Talking to your computer (with HTML!)* [online] Available at: <http://chrome.blogspot.com/2011/03/talking-to-your-computer-with-html5.html> [Accessed 19th Septemeber, 2012]

- Scott Sehlhorst. 2006. *Ten Requirements Gathering Techniques* [online] Available at: <http://tynerblain.com/blog/2006/11/21/ten-requirements-gathering-techniques/> [Accessed 10th July, 2012]
- Source Forge. n.d. *Sphinx-4 - A speech recognizer written entirely in the Java(TM) programming language* [online] Available at: <http://cmusphinx.sourceforge.net/sphinx4/> [Accessed 29th June, 2012]
- Stack Overflow, 2010. *C# Convert Words(string) to Int - Stack Overflow* [online] Available at: <http://stackoverflow.com/questions/11278081/c-sharp-convert-wordsstring-to-int> [Accessed 21st April, 2012]
- Stack Overflow, 2010. *Speech Recognition - SAPI and Windows 7 Problem - Stack Overflow* [online] Available at: <http://stackoverflow.com/questions/4213035/sapi-and-windows-7-problem> [Accessed 21st April, 2012]
- Source Forge. n.d. *OpenPop.NET - Robust POP3 client and MIME parser written in C#* [online] Available at: <http://hpop.sourceforge.net/> [Accessed 23rd May, 2012]
- The Radicate Group,Inc. 2011. *Email-Statistics-Report-2011-2015-Executive-Summary*. Available at: <http://www.radicati.com/wp/wp-content/uploads/2011/05/Email-Statistics-Report-2011-2015-Executive-Summary.pdf> [online] Available at: <http://www.radicati.com/wp/wp-content/uploads/2011/05/Email-Statistics-Report-2011-2015-Executive-Summary.pdf> [Accessed 5th July, 2012]
- Webopedia Computer Dictionary. n.d. *What is voice recognition?* [online] Available at: http://www.webopedia.com/TERM/V/voice_recognition.html [Accessed 5th June, 2012]
- Wikipedia. 2012. *Microsoft Speech API - Wikipedia, the free encyclopedia* [online] Available at: http://en.wikipedia.org/wiki/Microsoft_Speech_API [Accessed 28th May, 2012]
- Wikipedia. 2011. *Siri (software) - Wikipedia, the free encyclopedia* [online] Available at: [http://en.wikipedia.org/wiki/Siri_\(software\)](http://en.wikipedia.org/wiki/Siri_(software)) [Accessed 11th April, 2012]

9.3 List of Abbreviations

API	- Application Programming Interface
GUI	- Graphical User Interface
IDE	- Integrated Development Environment
POP	- Post Office Protocol
SAPI	- Speech Application Programming Interface
SDK	- Software Development Kit
SMTP	- Simple Mail Transfer Protocol
XML	- Extensible Markup Language

9.4 User manual

9.4.1 Voice Driven Email Client Version – 1 (Separate interface)

Launch system



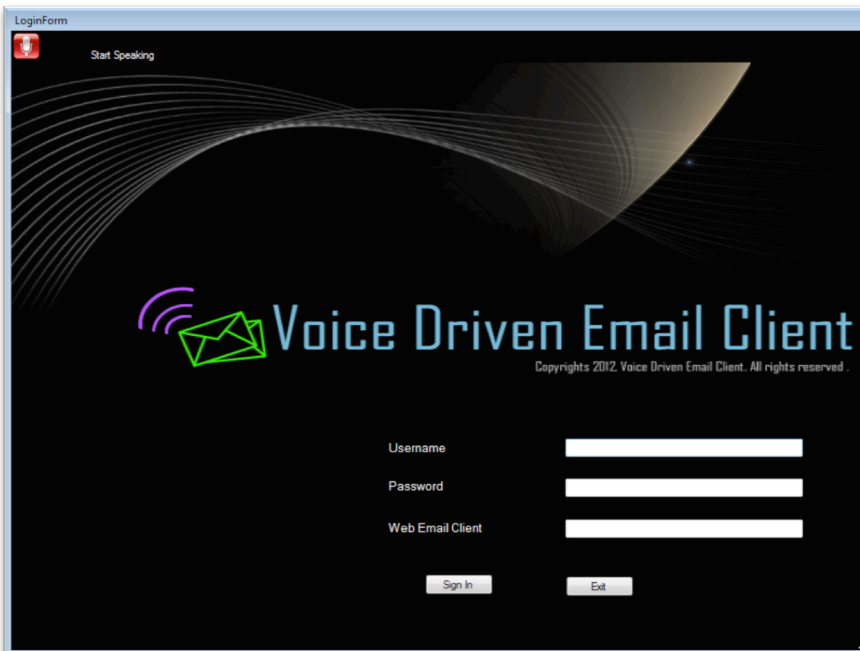
Commands:

- N/A

Comments:

Double click on “VoiceDrivenEmailClientVersion1” icon
System starts up with a Splash Screen which displays for 5 seconds introducing the user to the system

Figure 17 : startup splash screen



Commands:

- Include Username
- include Password
- Include Client

Comments:

User will have to update their username, password and web-client name which are saved on the ‘LoginCredentials.xml’

Figure 18: System Login Form



The screenshot shows the login form for the Voice Driven Email Client. The form is titled "Voice Driven Email Client" and includes the text "Copyrights 2012, Voice Driven Email Client. All rights reserved.". The form fields are: Username (voicedrivemailclient@gmail.com), Password (getu6421), and Web Email Client (Gmail). There are two buttons: "Sign In" and "Exit".


Commands:

- Select sign in or select exit

Comments:

N/A

Figure 19: Login Form with Completed Fields



The screenshot shows the login form for the Voice Driven Email Client. The form is titled "Voice Driven Email Client" and includes the text "Copyrights 2012, Voice Driven Email Client. All rights reserved.". The form fields are: Username (voicedrivemailclient@gmail.com), Password (getu6421111), and Web Email Client (Gmail). There are two buttons: "Sign In" and "Exit". An error message "Invalid Login Details Please Try Again" is displayed in red text above the form fields.

Commands:

- N/A

Comments:

Error Message

Figure 20: User Login Form with Invalid Credentials

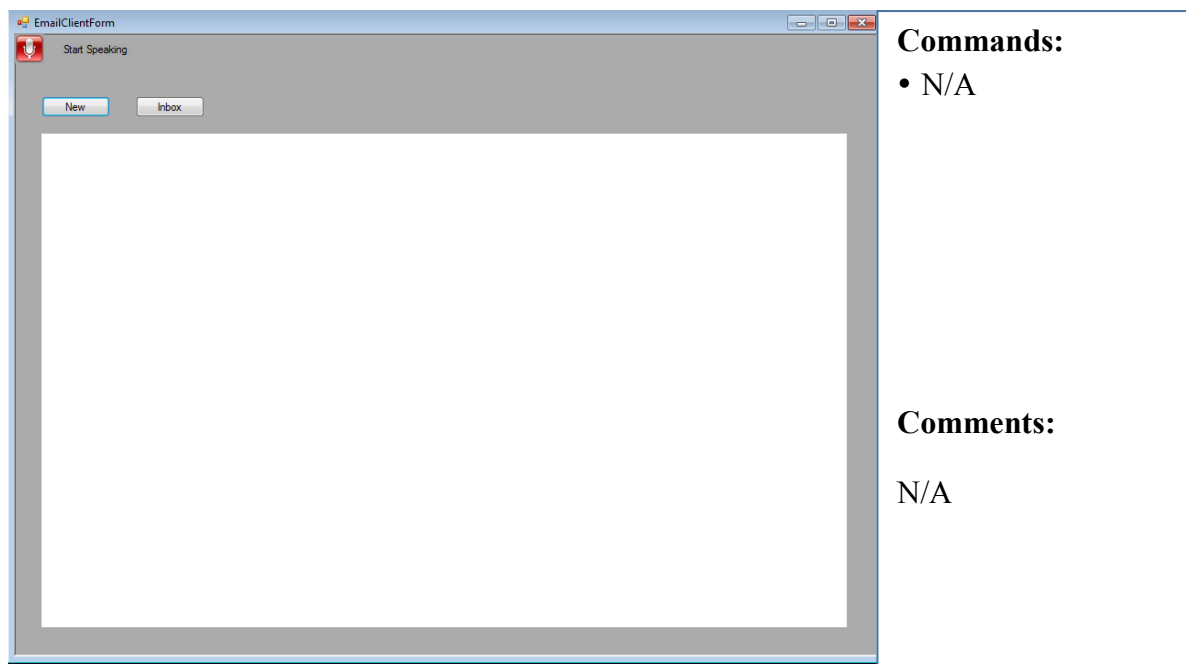


Figure 21: Successful login

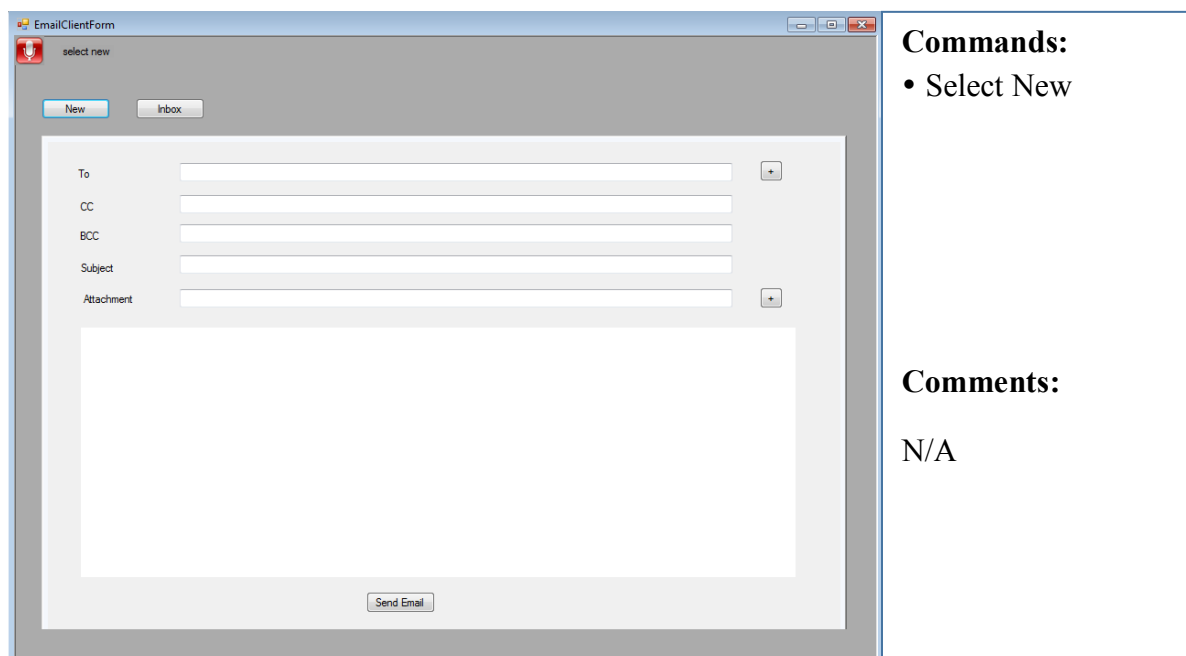
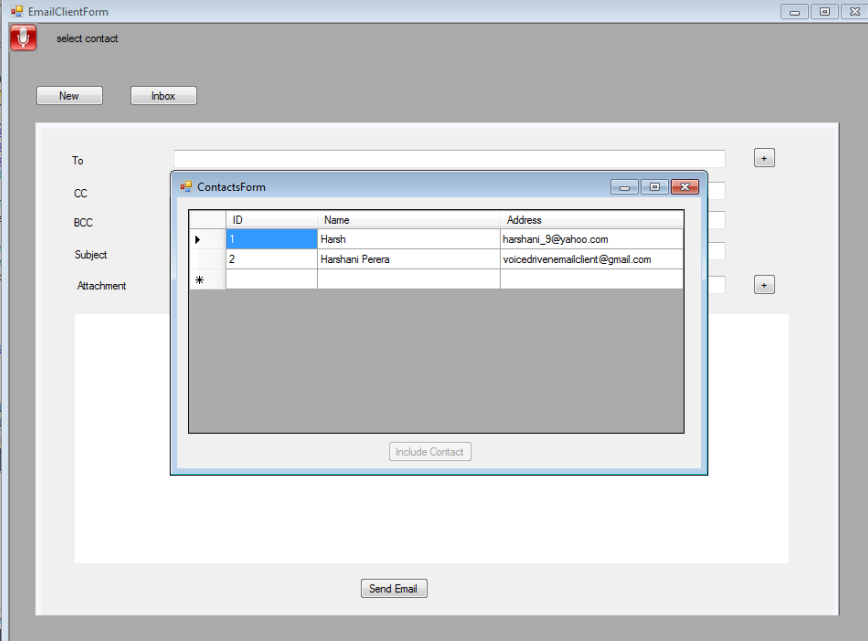


Figure 22: Create New Email



The screenshot shows the EmailClientForm interface. The 'To' field contains the text 'select contact'. A 'ContactsForm' dialog box is open, displaying a table with the following data:

ID	Name	Address
1	Harsh	harshani_9@yahoo.com
2	Harshani Perera	voicedrivemalclient@gmail.com

The 'Include Contact' button is visible at the bottom of the dialog. The main window also shows 'New' and 'Inbox' buttons, and a 'Send Email' button at the bottom.

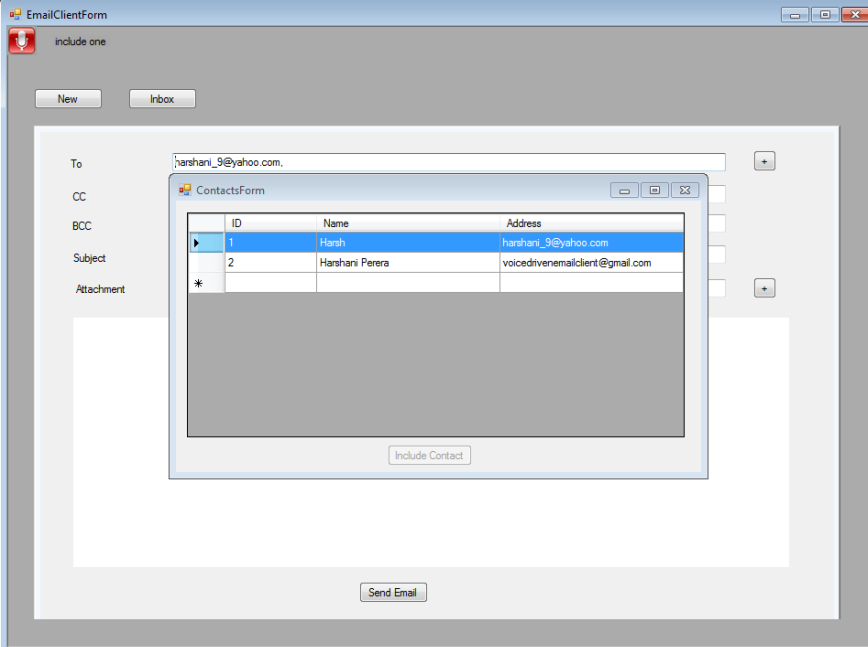
Commands:

- Select to
- Select contact

Comments:

Opens Gmail contact list

Figure 23: To Field - Search contact



The screenshot shows the EmailClientForm interface. The 'To' field now contains the email address 'harshani_9@yahoo.com'. The 'ContactsForm' dialog box is still open, displaying the same table as in Figure 23:

ID	Name	Address
1	Harsh	harshani_9@yahoo.com
2	Harshani Perera	voicedrivemalclient@gmail.com

The 'Include Contact' button is visible at the bottom of the dialog. The main window also shows 'New' and 'Inbox' buttons, and a 'Send Email' button at the bottom.

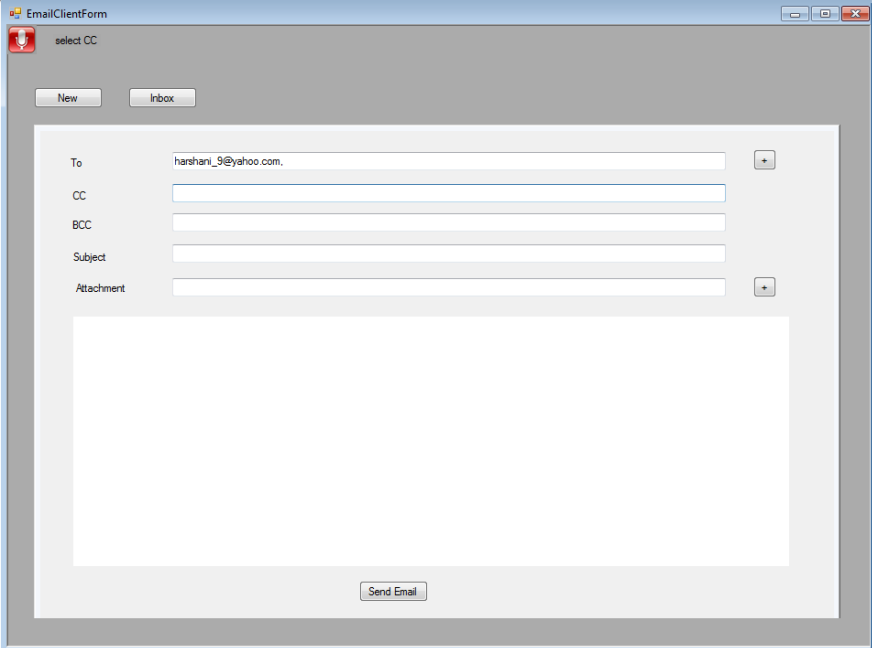
Commands:

- include 'contact id' (example- include one)

Comments:

N/A

Figure 24 : To Field - Add contact



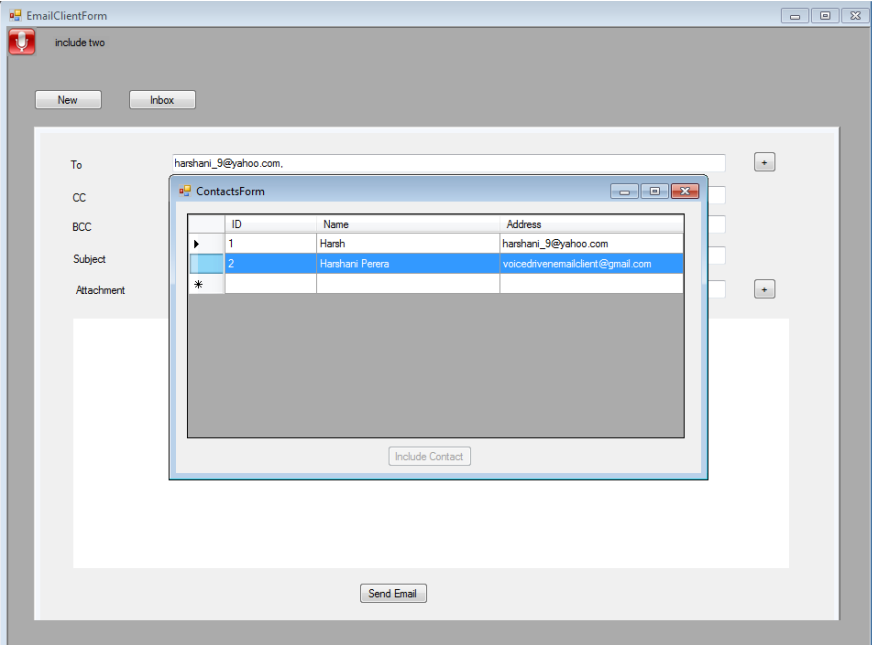
Commands:

- Select cc

Comments:

Sets focus to 'cc field'

Figure 25: CC Field – Focus



Commands:

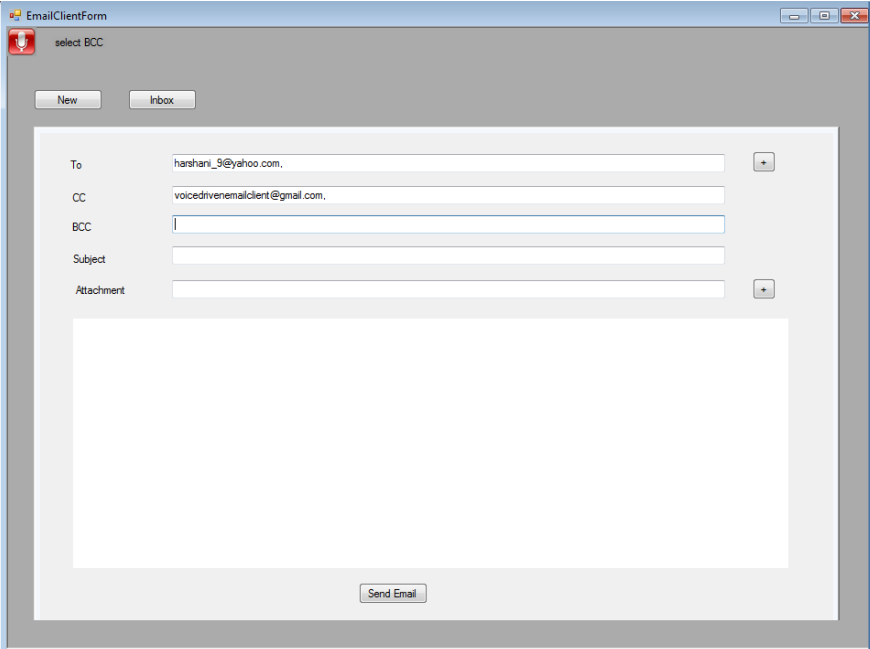
- include 'contact id' (example- include two)

Comments:

N/A

ID	Name	Address
1	Harsh	harshani_9@yahoo.com
2	Harshani Perera	voicediv@emal-client@gmail.com

Figure 26: CC Field - Add contact



The screenshot shows a window titled "EmailClientForm" with a subtitle "select BCC". The window contains a form with fields for "To", "CC", "BCC", "Subject", and "Attachment". The "BCC" field is currently selected and has a blue border. Below the form is a "Send Email" button.

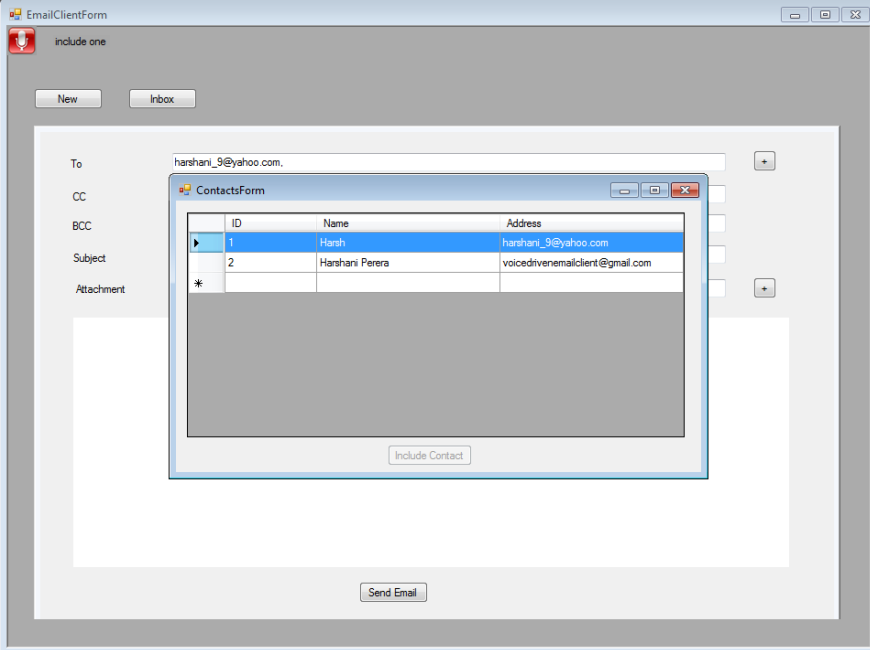
Commands:

- Select bcc

Comments:

Sets focus to 'bcc field

Figure 27: BCC Field - Focus



The screenshot shows the "EmailClientForm" window with the subtitle "include one". A "ContactsForm" dialog box is open over the form, displaying a table of contacts. The "To" field in the background is selected. The "ContactsForm" dialog has an "Include Contact" button at the bottom.

ID	Name	Address
1	Harsh	harshani_9@yahoo.com
2	Harshani Perera	voicedrivenemailclient@gmail.com
*		

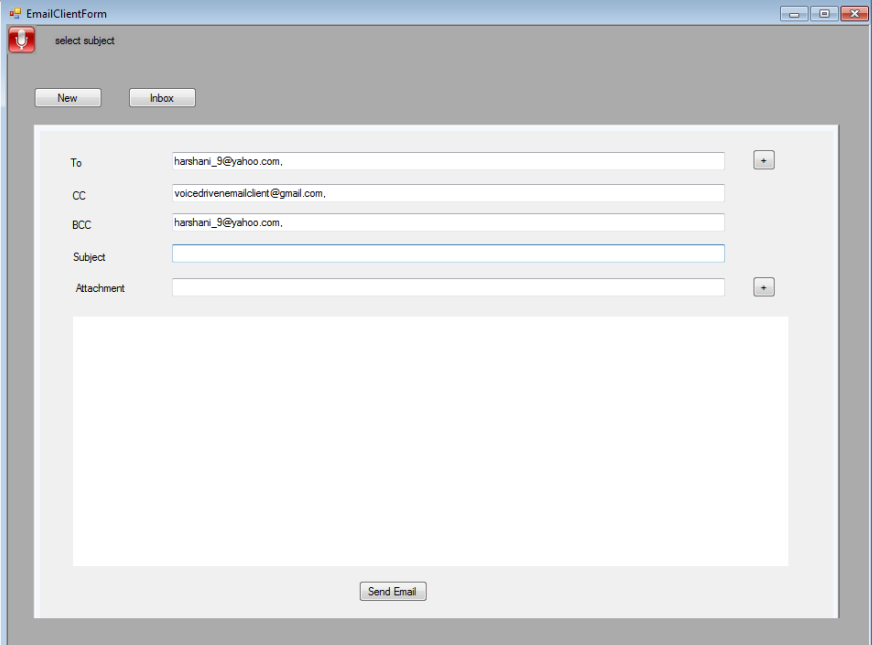
Commands:

- include 'contact id' (example- include one)

Comments:

N/A

Figure 28: BCC Field – Add contact



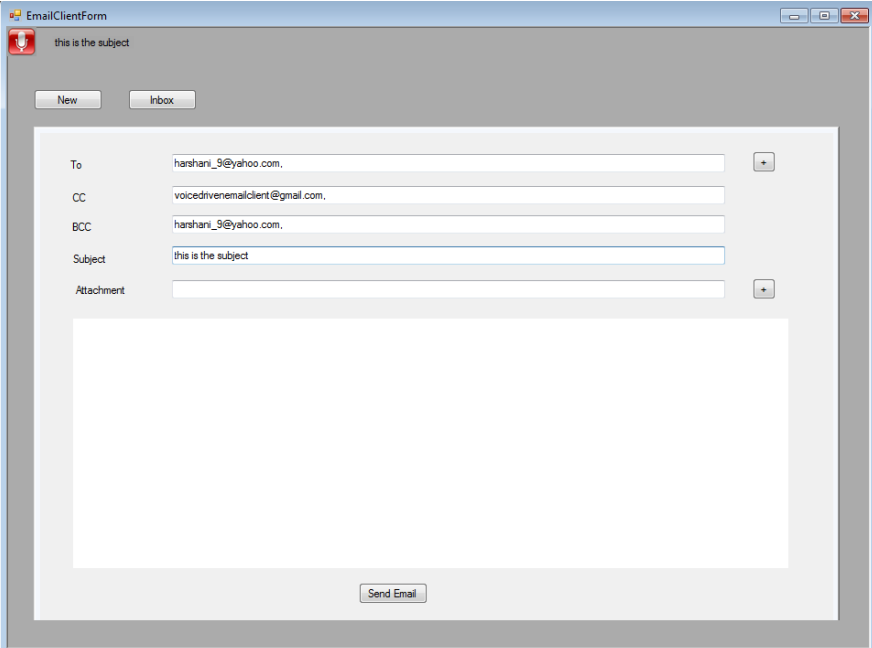
Commands:

- Select subject

Comments:

Sets focus to 'subject field'

Figure 29 : Select Subject



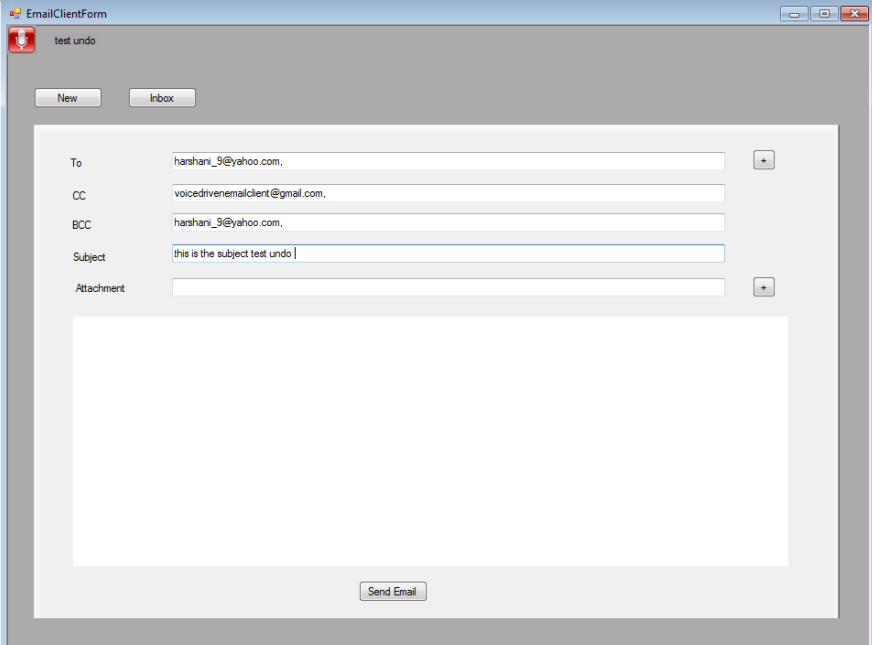
Commands:

- N/A

Comments:

Dictate subject

Figure 30 : Dictate to 'subject' field



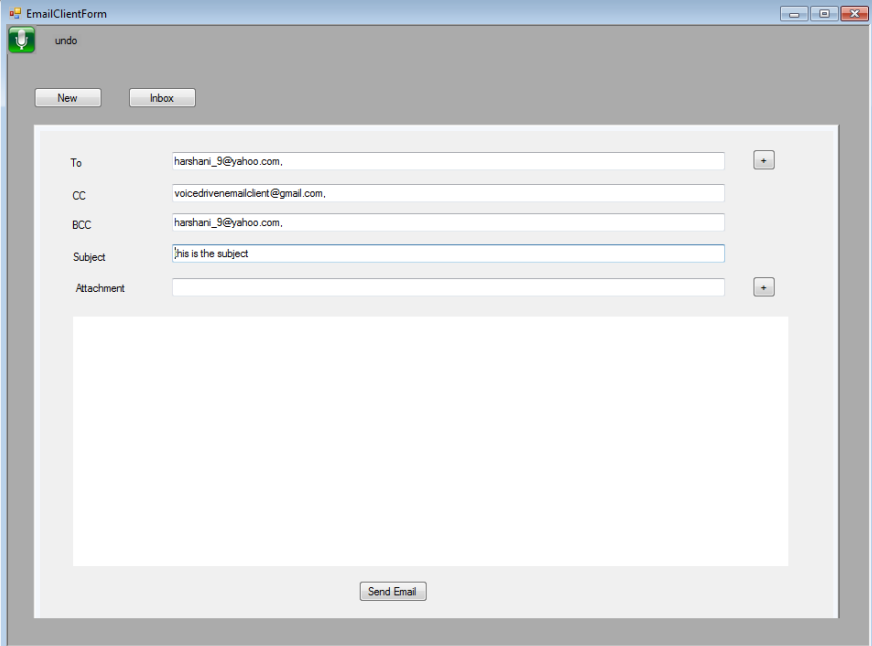
Commands:

- N/A

Comments:

Dictated subject is recognized wrong or dictates wrong phrase

Figure 31 : Mistype on Subject



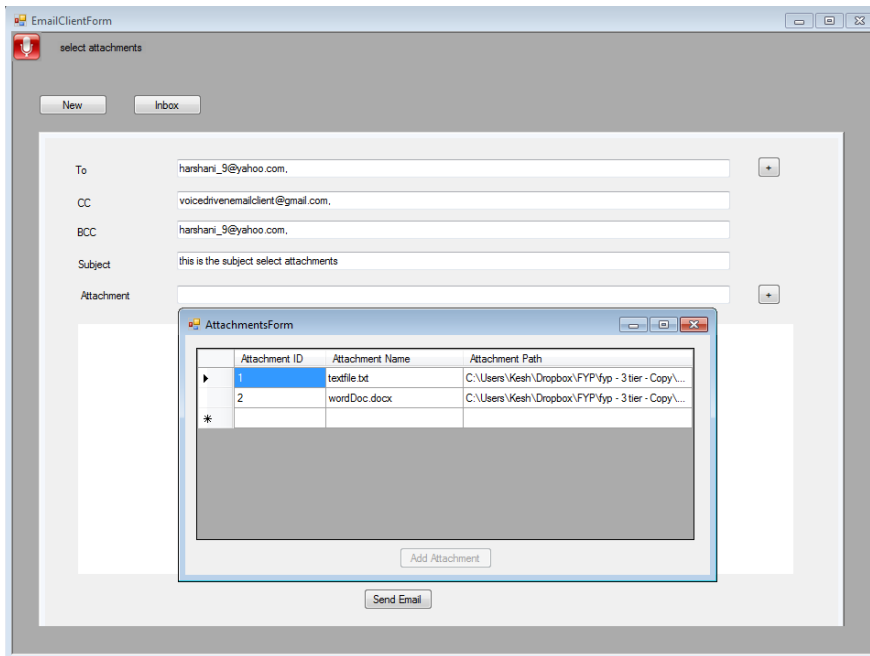
Commands:

- Undo

Comments:

Removes the last dictated text. Undo works only up to one level back

Figure 32 : Undo Mistyping on Subject



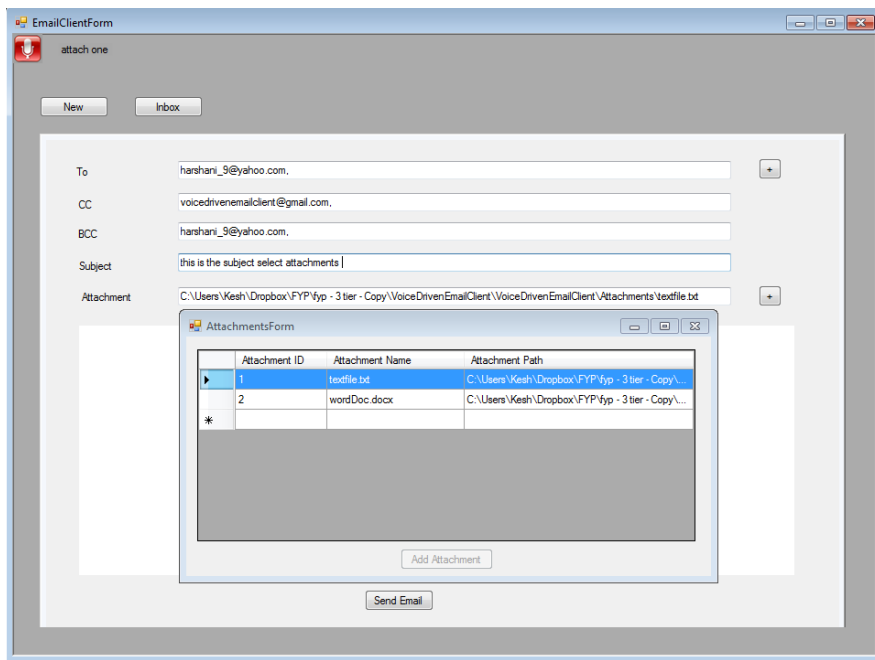
Commands:

- Select attachment(s)

Comments:

Displays all files in the attachments folder

Figure 33 : Select Attachments



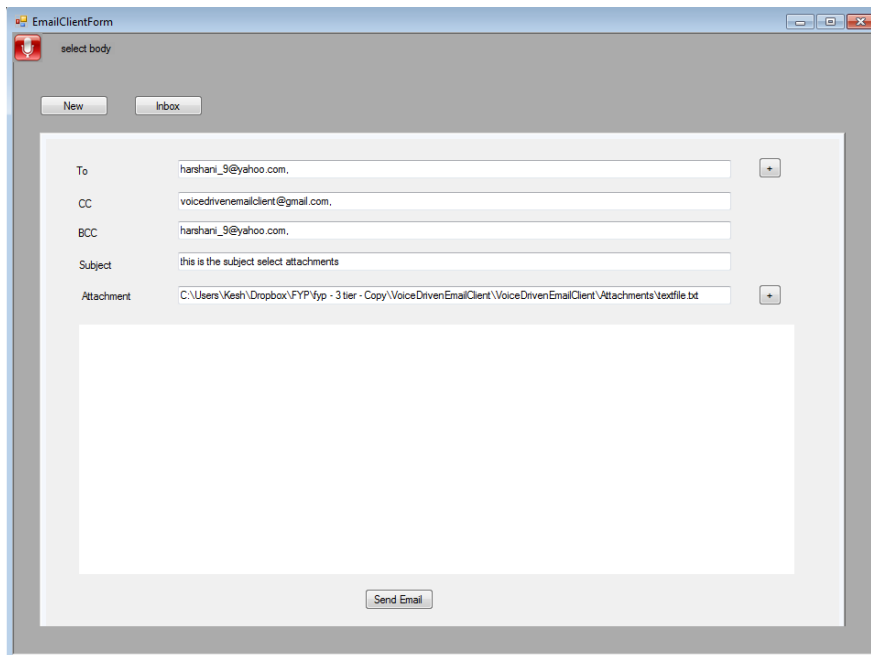
Commands:

- Attach 'attachment id'

Comments:

Adds path of the attachment selected to the attachment field

Figure 34 : Attach File

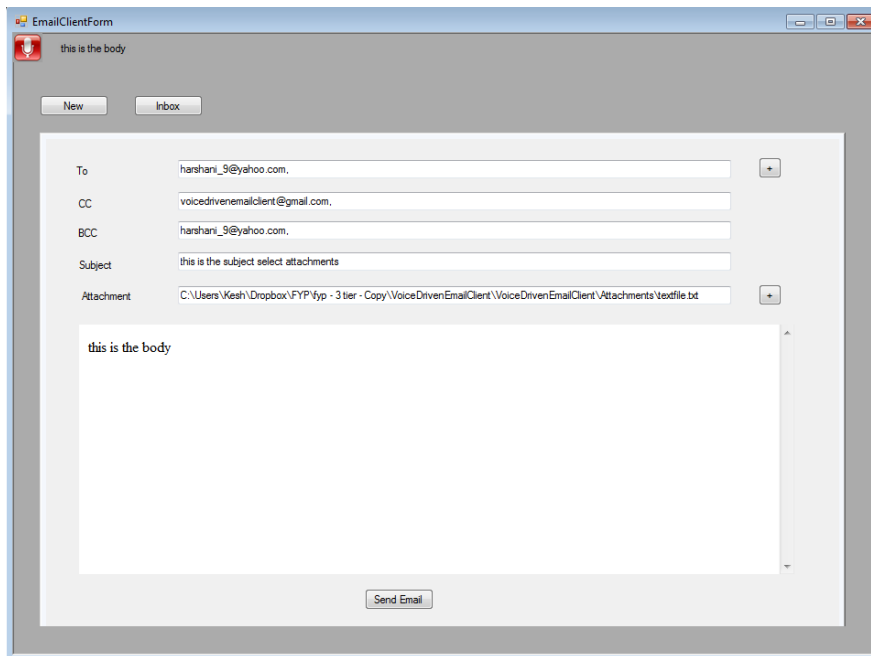
**Commands:**

- Select body

Comments:

Sets focus to 'body field'

Figure 35 : Select Body

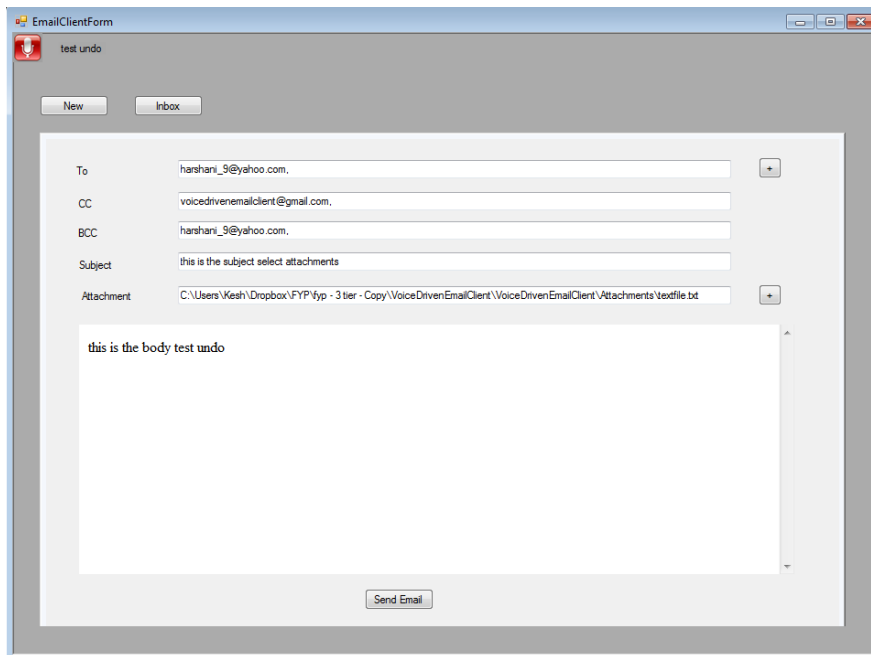
**Commands:**

- N/A

Comments:

Dictate body of email

Figure 36 : Dictate Body

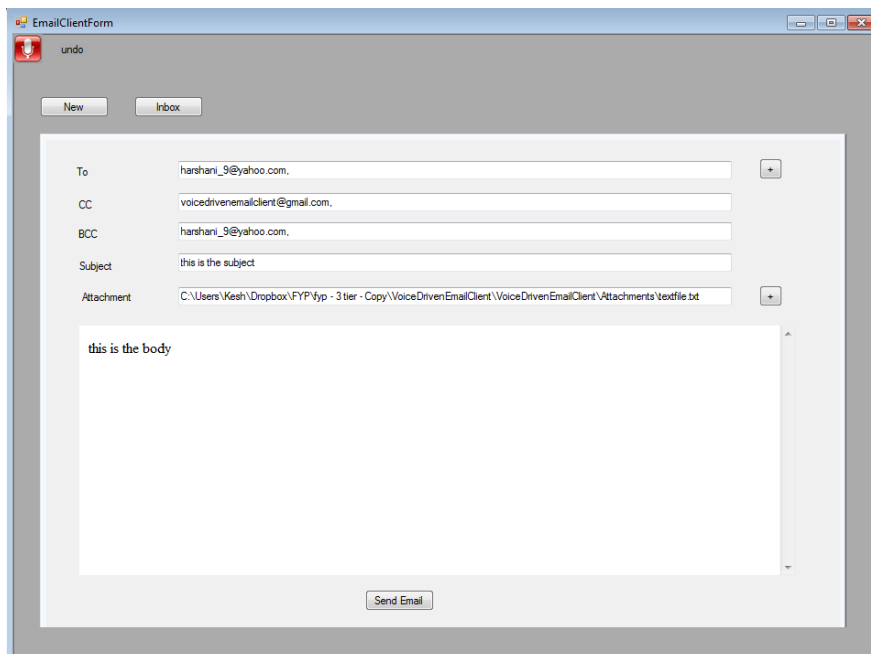
**Commands:**

- N/A

Comments:

Dictated body is recognized wrong or dictates wrong phrase

Figure 37 : Mistyping on Body

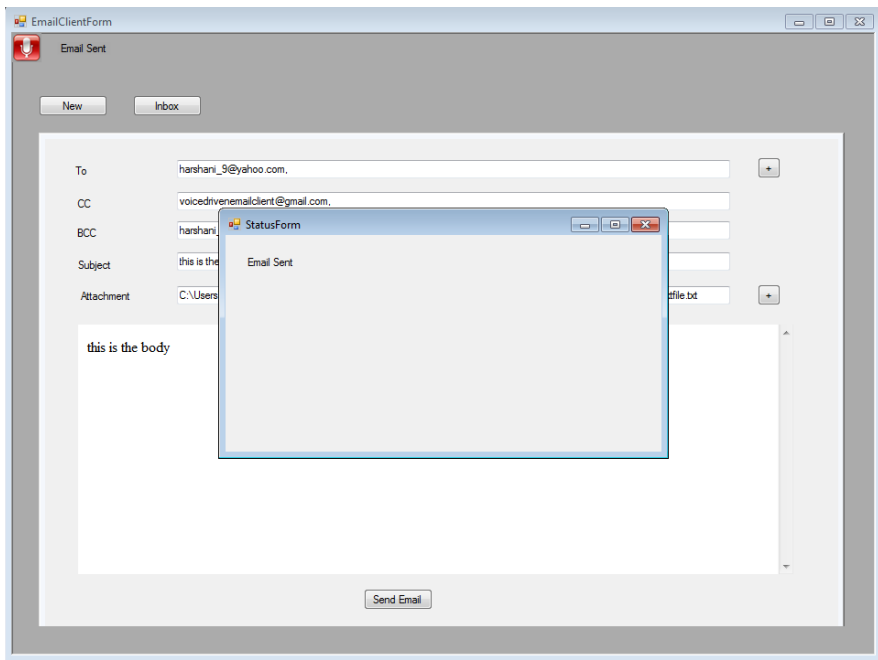
**Commands:**

- Undo

Comments:

Removes the last dictated text. Undo works only up to one level back

Figure 38 : Undo Mistyping on Body

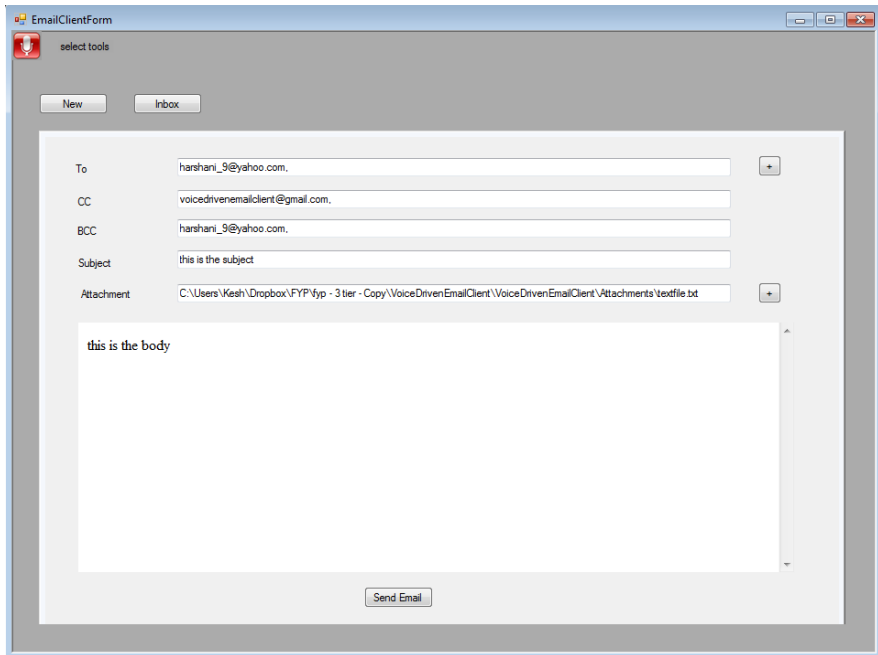


Commands:

- Select send

Comments:
Confirmation message pops up
,

Figure 39 : Send Message

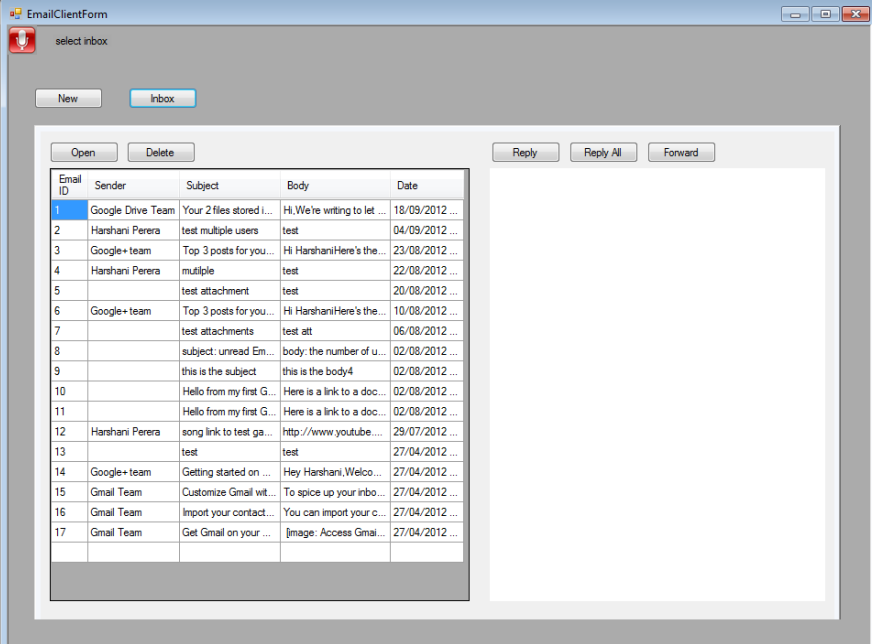


Commands:

- Select close

Comments:
Closes confirmation message form

Figure 40 : Close Confirmation Popup Form

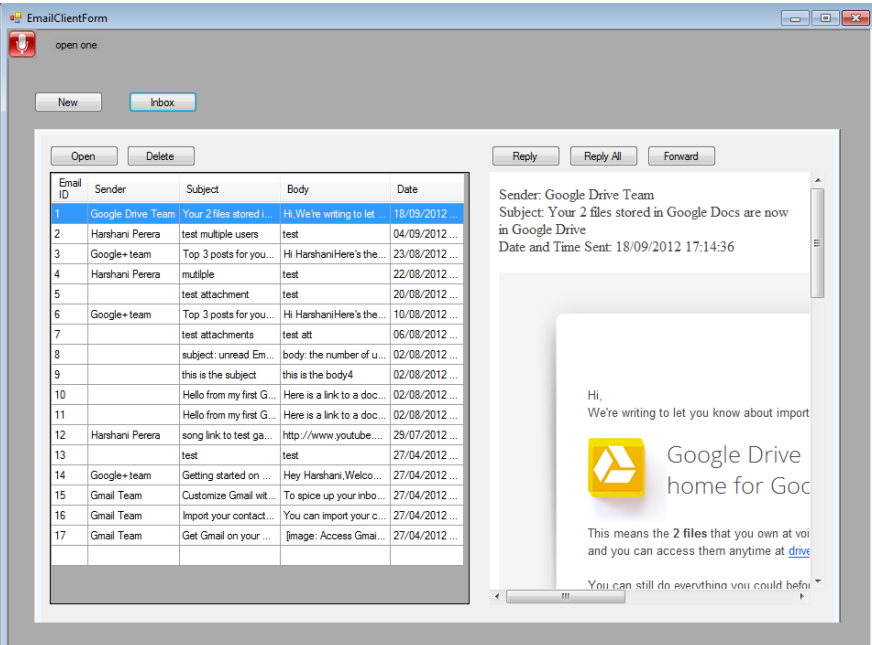


Commands:

- Select inbox

Comments:
Retrieves Inbox messages from web client (Gmail)

Figure 41 : Retrieve Inbox Message



Commands:

- Open 'email id'

Comments:
Displays Email

Figure 42 : Open Email

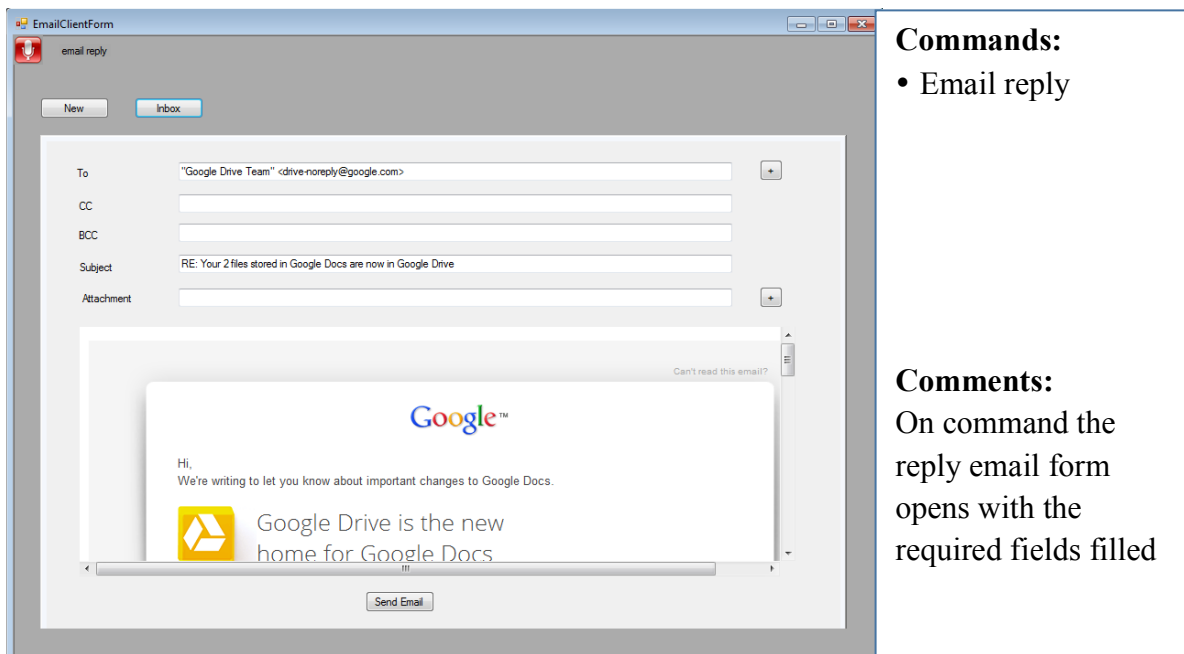


Figure 43 : Reply to Email

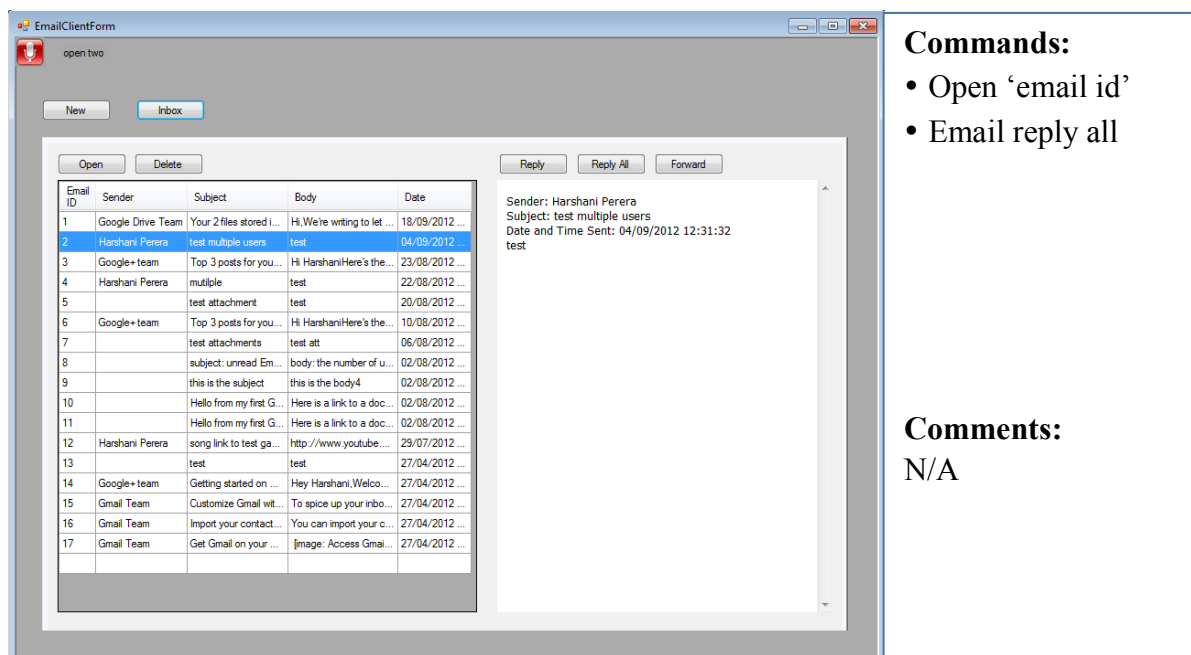
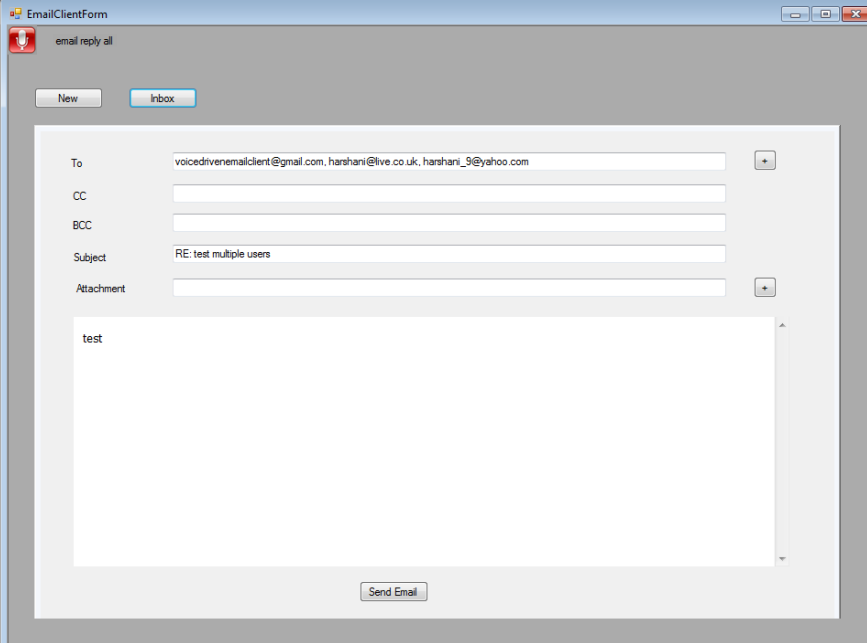


Figure 44 : Reply to All - 1



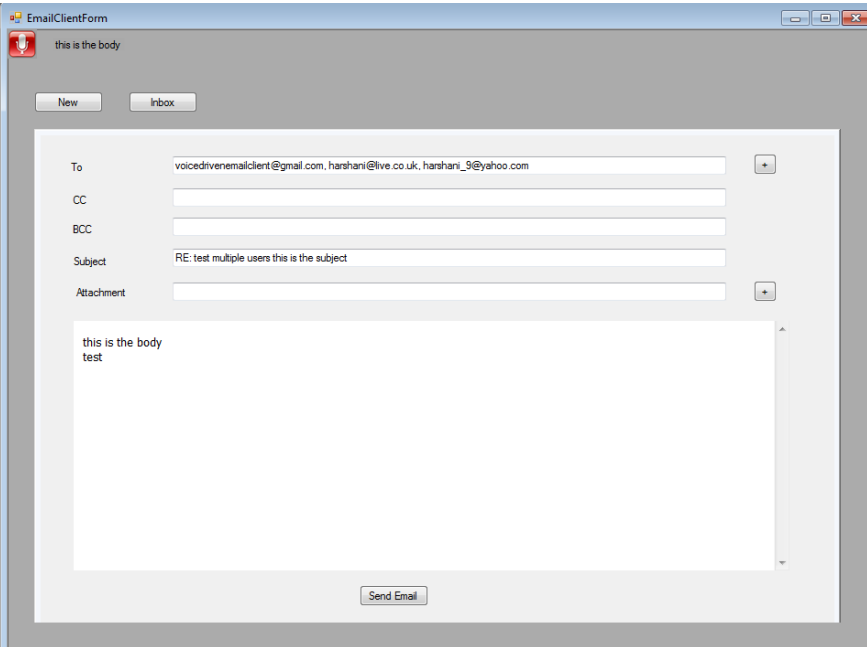
Commands:

- Email reply all

Comments:

On command the reply email form opens with the required fields filled

Figure 45 : Reply to All - 2



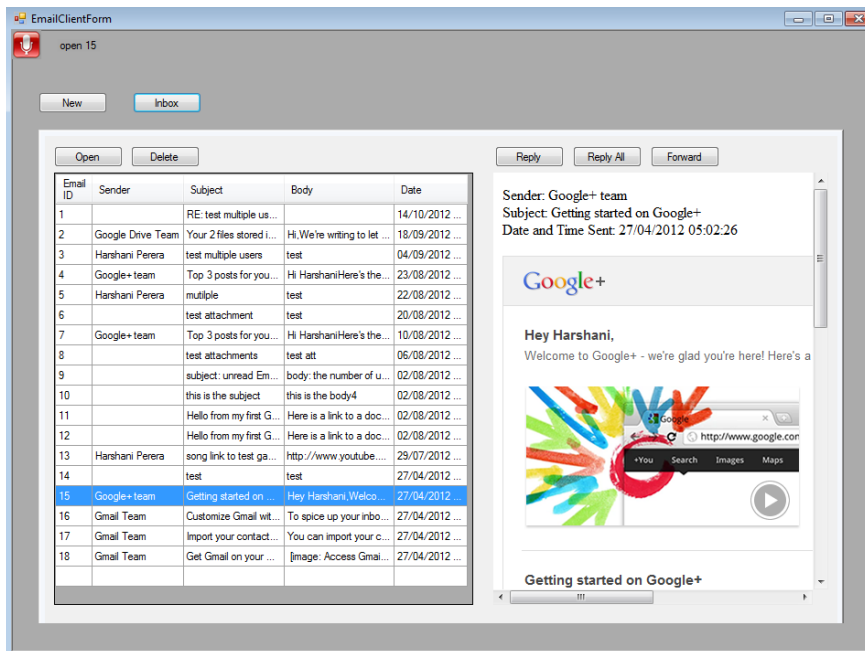
Commands:

- Same commands used for filling new email form applies for reply, reply all and forward forms

Comments:

Body gets dictated above the original body

Figure 46 : Reply to All - 3



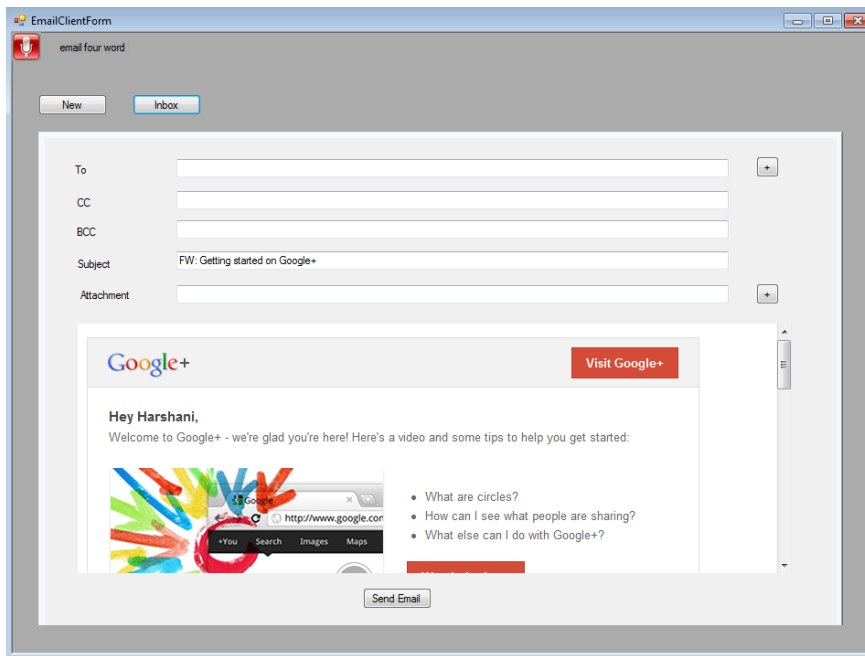
Commands:

- Open 'email id'

Comments:

N/A

Figure 47 : Forward Email - 1



Commands:

- Same commands used for filling new email form applies for reply, reply all and forward forms

Comments:

Body gets dictated above the original body

Figure 48 : Forward Email - 2

9.4.2 Voice Driven Email Client Version – 2 (Gmail interface)



Commands:

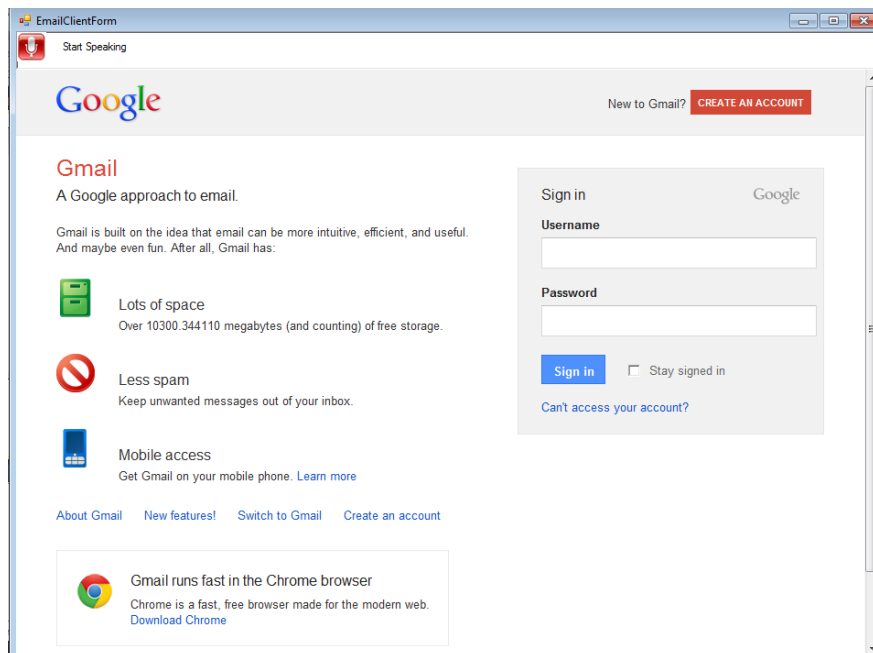
- N/A

Comments:

Double click on “VoiceDrivenEmailClientVersion2” icon.

System starts up with a Splash Screen which displays for 5 seconds introducing the user to the system

Figure 49 : Startup Splash Screen



Commands:

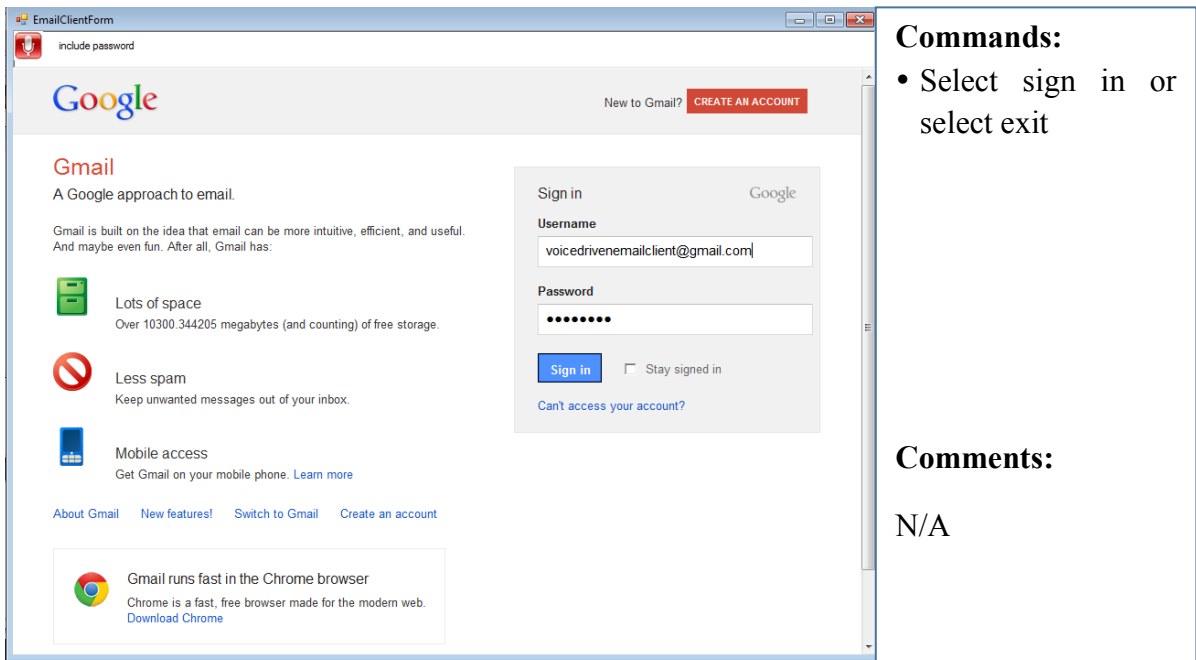
- Include Username
- include Password
- Include Client

Comments:

User will have to update their username, password and web-client name which are saved on the ‘LoginCredentials.xml’

Figure 50: System Login

Form



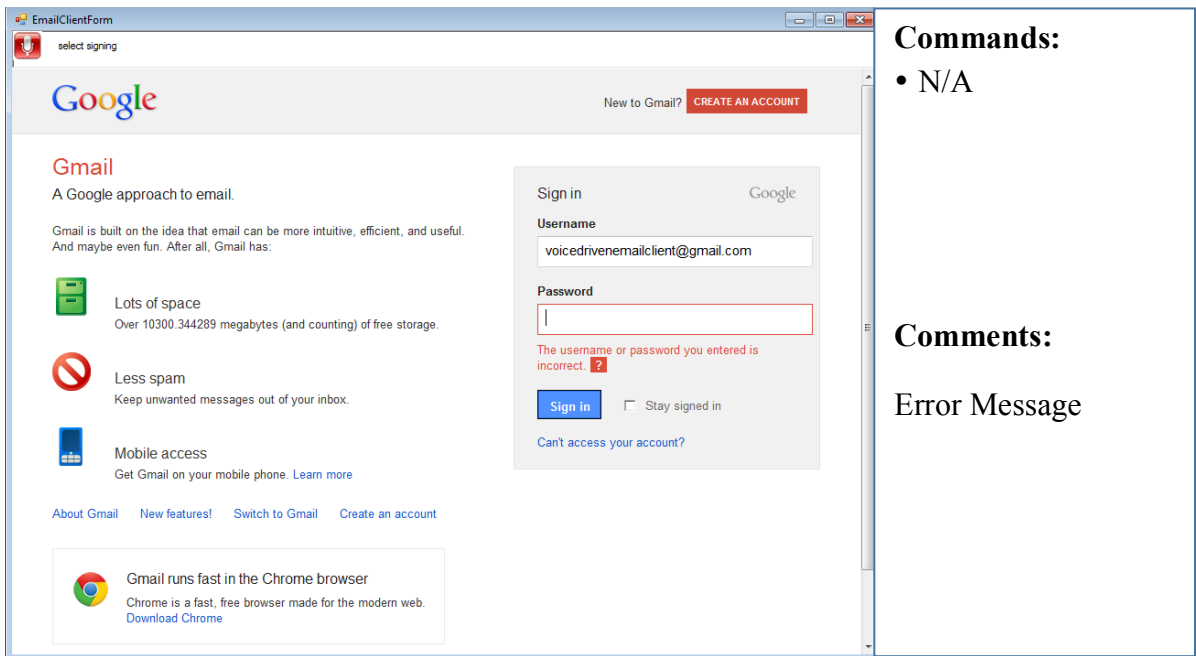
Commands:

- Select sign in or select exit

Comments:

N/A

Figure 51: Login Form with Completed Fields



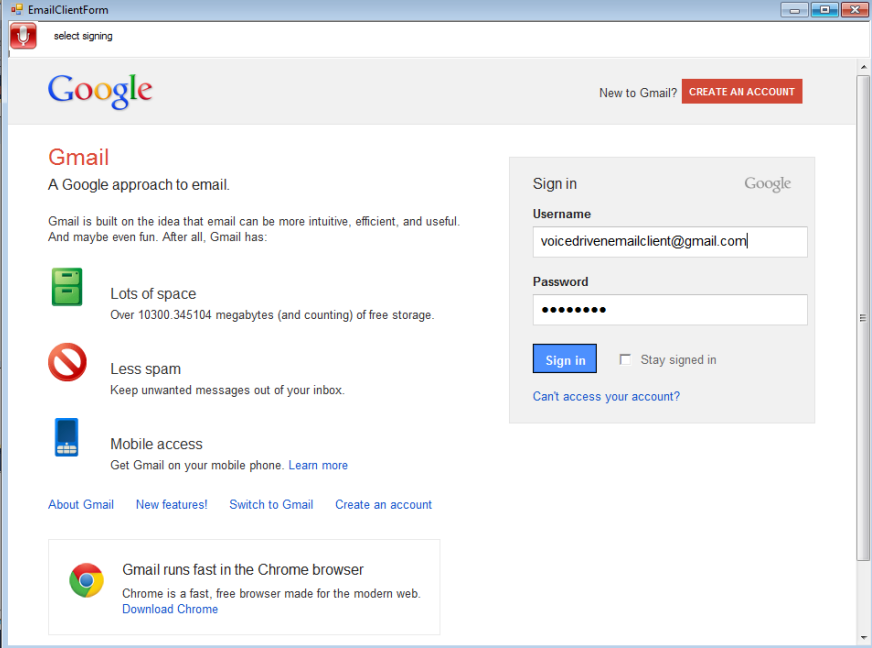
Commands:

- N/A

Comments:

Error Message

Figure 52 : User Login Form with Invalid Credentials



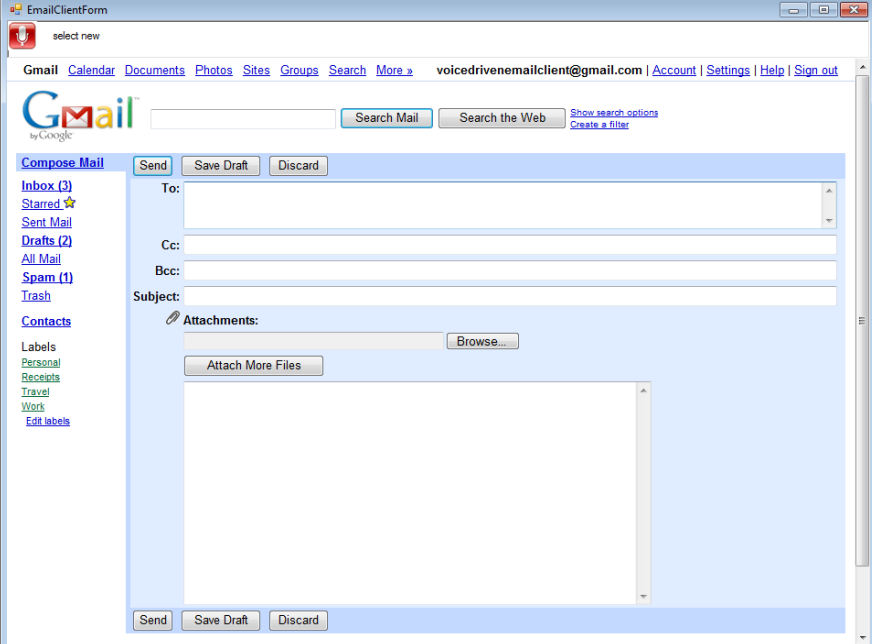
Commands:

- Select sign in

Comments:

N/A

Figure 53: Successful Login



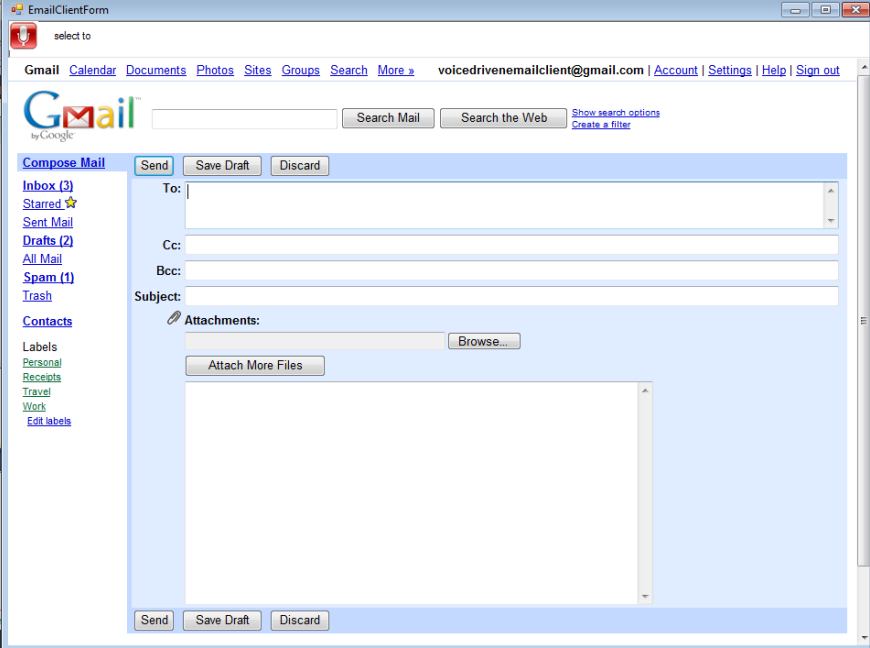
Commands:

- Select New

Comments:

N/A

Figure 54: Create New Email



The screenshot shows the Gmail 'Compose Mail' window. The 'To:' field is highlighted with a blue border, indicating it is the focus of the command. The interface includes a left sidebar with folders like 'Inbox (3)', 'Starred', 'Sent Mail', 'Drafts (2)', 'All Mail', and 'Spam (1)'. The main area has fields for 'To:', 'Cc:', 'Bcc:', and 'Subject:'. There are 'Send', 'Save Draft', and 'Discard' buttons at the top and bottom.

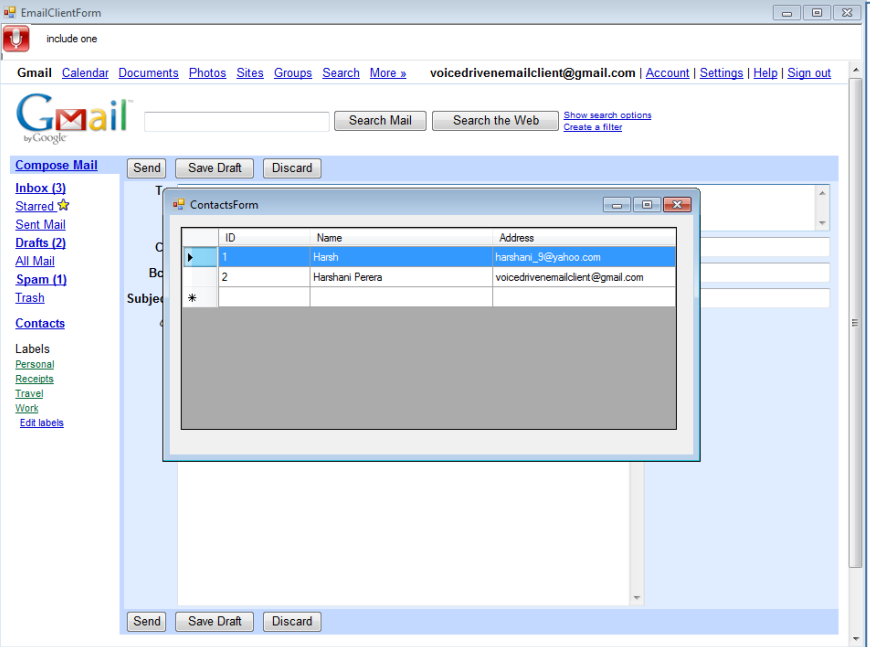
Commands:

- Select to

Comments:

Sets focus to 'to' field

Figure 55: To Field – Focus



The screenshot shows the Gmail 'Compose Mail' window with a 'ContactsForm' dialog box open. The dialog box contains a table with contact information. The 'To:' field in the background is partially visible and highlighted.

ID	Name	Address
1	Harsh	harshani_9@yahoo.com
2	Harshani Perera	voicedrivenemailclient@gmail.com
*		

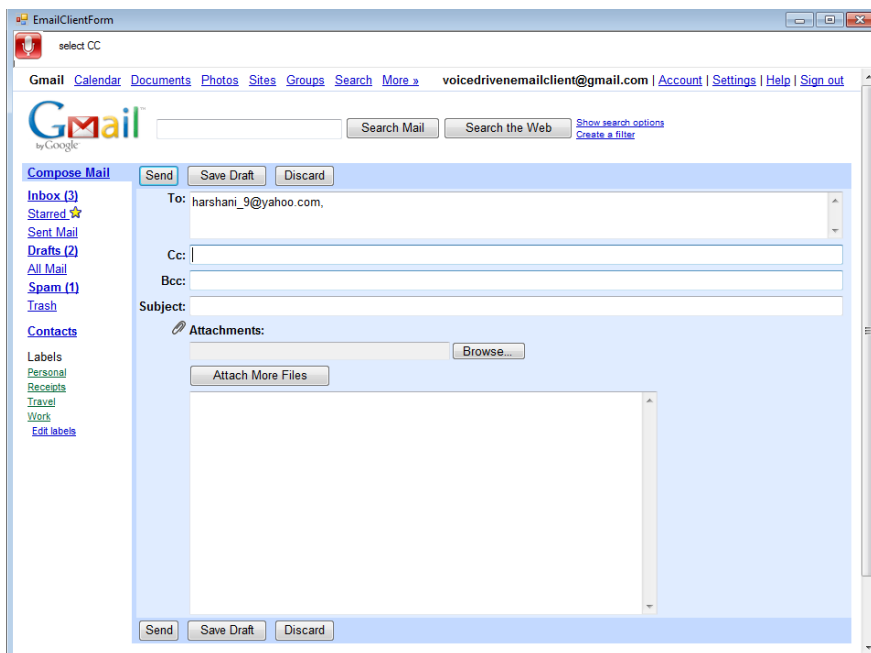
Commands:

- Select contact
- include 'contact id' (example- include one)

Comments:

Opens Gmail contact list in a new form

Figure 56: To Field - Add Contact



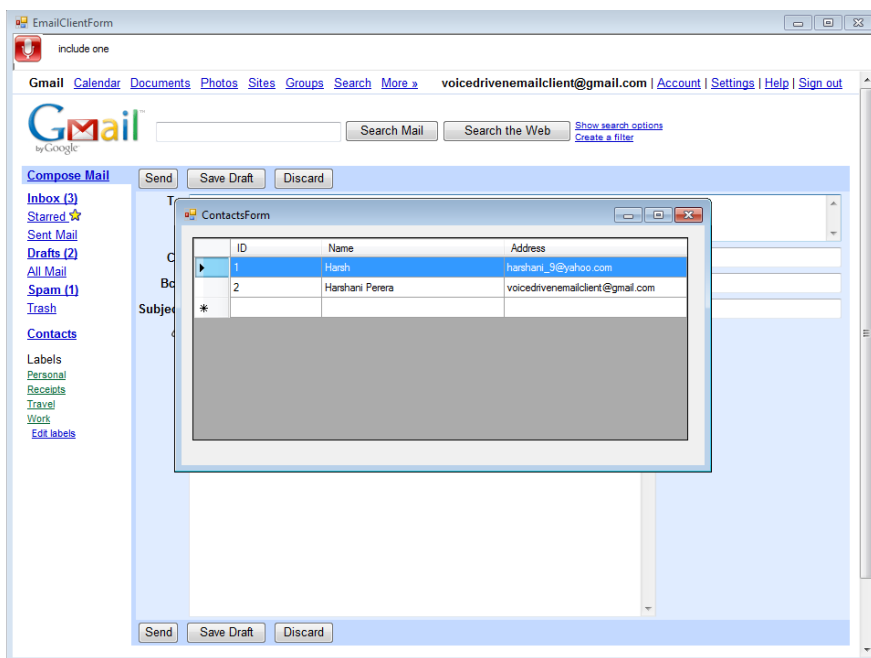
Commands:

- Select cc

Comments:

Sets focus to 'cc field'

Figure 57: CC Field - Focus



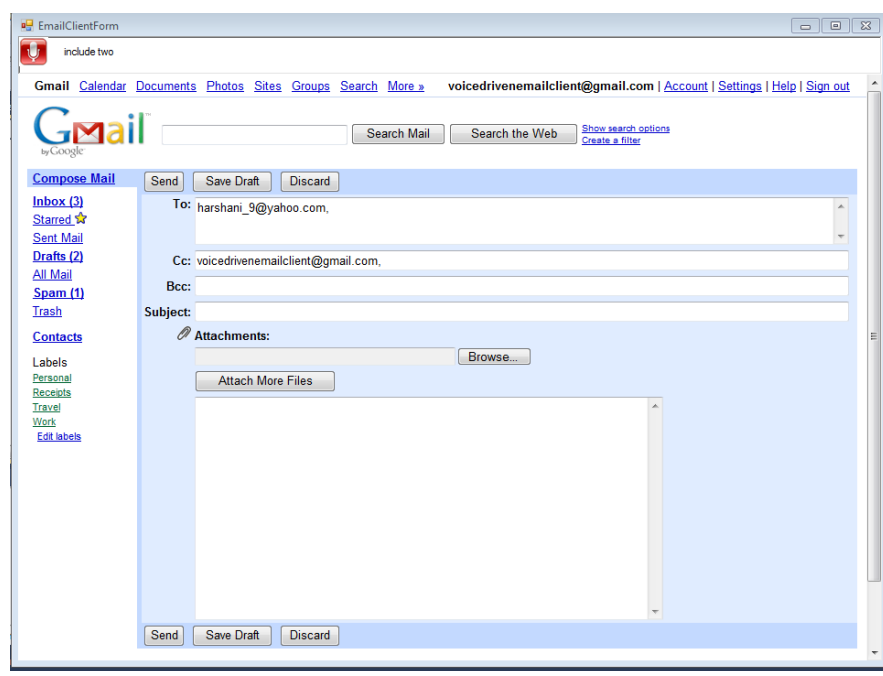
Commands:

- include 'contact id' (example- include two)

Comments:

N/A

Figure 58: CC Field - Add Contact



The screenshot shows the Gmail 'Compose Mail' interface. The 'To' field contains 'harshani_9@yahoo.com', the 'Cc' field contains 'voicedrivenemailclient@gmail.com', and the 'Bcc' field is currently empty and has the focus. The 'Subject' field is also empty. There are 'Send', 'Save Draft', and 'Discard' buttons at the top and bottom of the compose area. The left sidebar shows navigation links for 'Compose Mail', 'Inbox (3)', 'Starred', 'Sent Mail', 'Drafts (2)', 'All Mail', 'Spam (1)', 'Trash', 'Contacts', and 'Labels'.

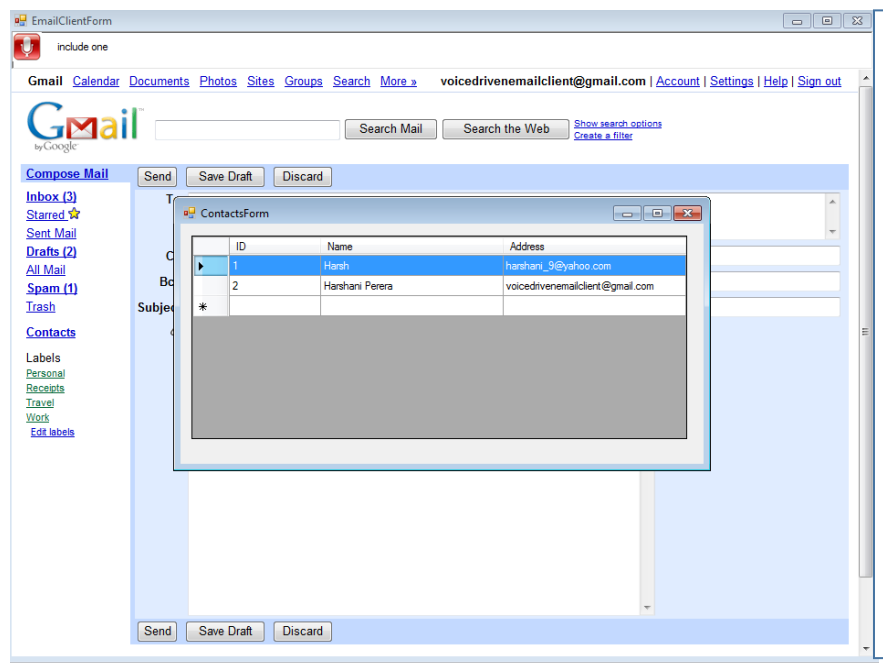
Commands:

- Select bcc

Comments:

Sets focus to 'bcc field'

Figure 59: BCC Field - Focus



The screenshot shows the Gmail 'Compose Mail' interface with a 'ContactsForm' dialog box open. The dialog box contains a table with contact information:

ID	Name	Address
1	Harsh	harshani_9@yahoo.com
2	Harshani Perera	voicedrivenemailclient@gmail.com
*		

The 'To' field in the background contains 'harshani_9@yahoo.com', 'Cc' contains 'voicedrivenemailclient@gmail.com', and 'Bcc' is empty. The 'Subject' field is also empty. The 'ContactsForm' dialog box has a search bar and a list of contacts. The 'Compose Mail' buttons ('Send', 'Save Draft', 'Discard') are visible at the top and bottom of the compose area.

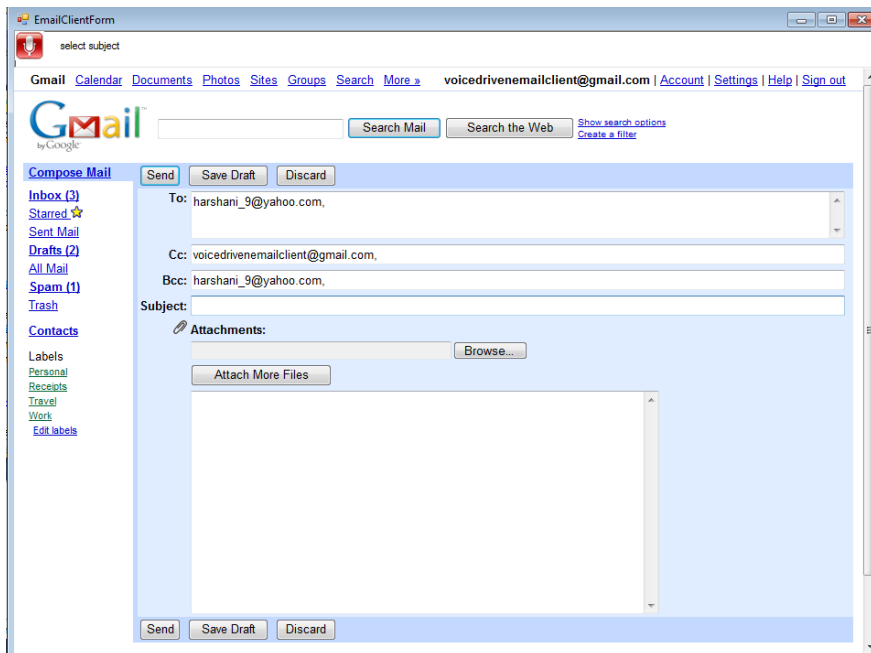
Commands:

- include 'contact id' (example- include one)

Comments:

Adds the contact address to the 'to field'

Figure 60: BCC Field – Add Contact

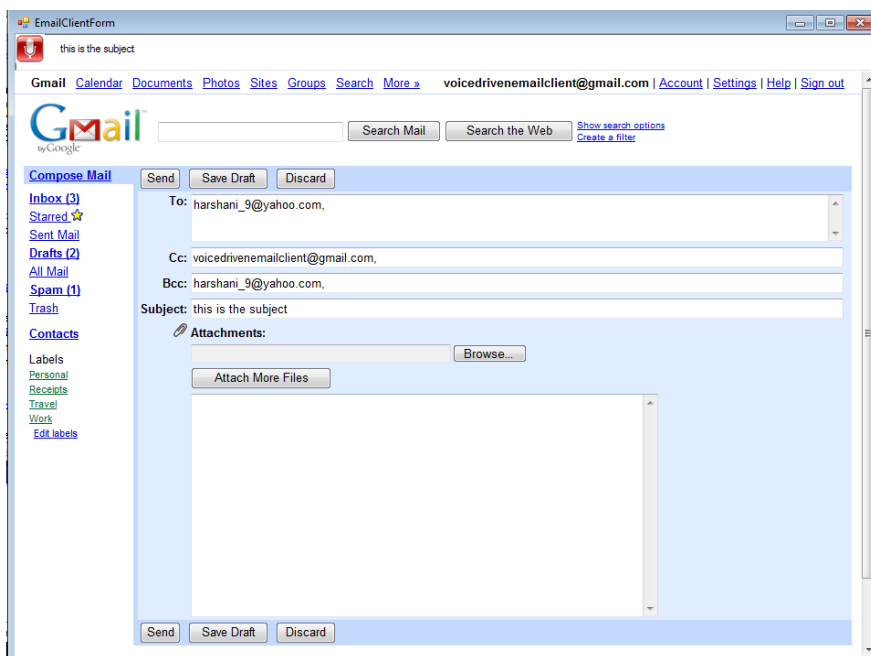
**Commands:**

- Select subject

Comments:

Sets focus to the 'subject field'

Figure 61 : Select Subject

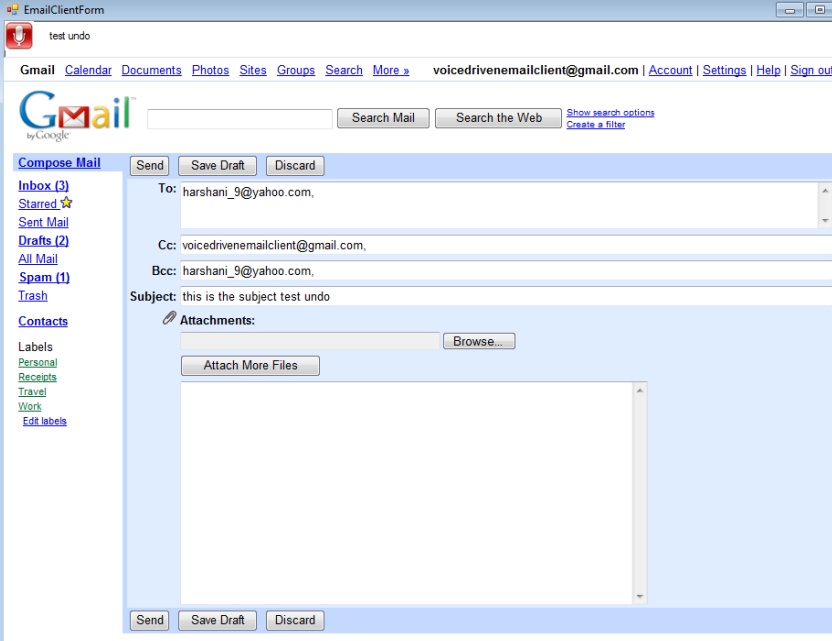
**Commands:**

- N/A

Comments:

Dictate subject

Figure 62 : Dictate to 'Subject' Field



The screenshot shows the Gmail 'Compose Mail' interface. The subject line is 'this is the subject test undo'. The 'To' field contains 'harshani_9@yahoo.com', 'Cc' contains 'voicedrivenemailclient@gmail.com', and 'Bcc' contains 'harshani_9@yahoo.com'. The interface includes standard Gmail navigation and compose controls.

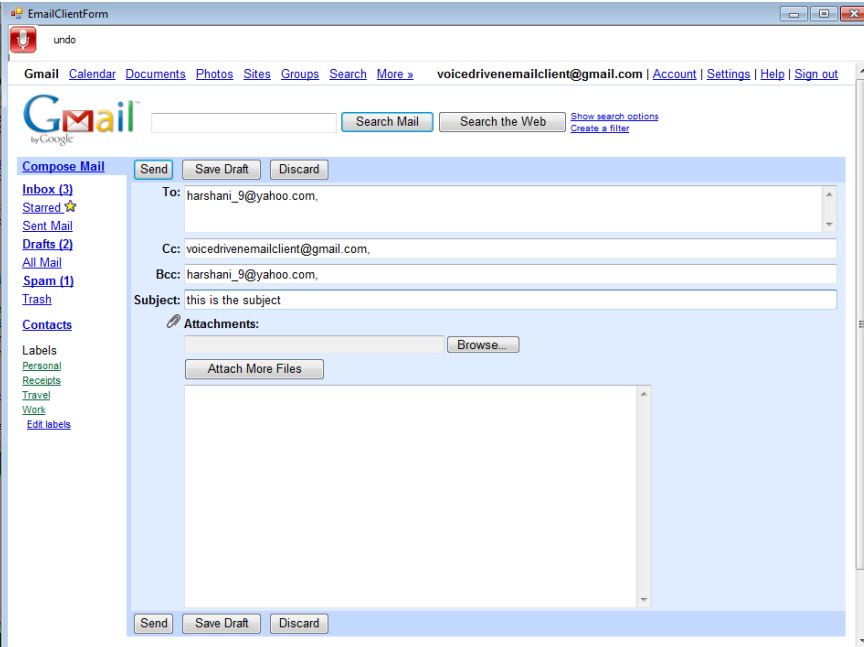
Commands:

- N/A

Comments:

Dictated subject is recognized wrong or dictates wrong phrase

Figure 63 : Mistype on Subject



The screenshot shows the Gmail 'Compose Mail' interface after an undo action. The subject line is now 'this is the subject'. The 'To', 'Cc', and 'Bcc' fields remain the same as in the previous screenshot.

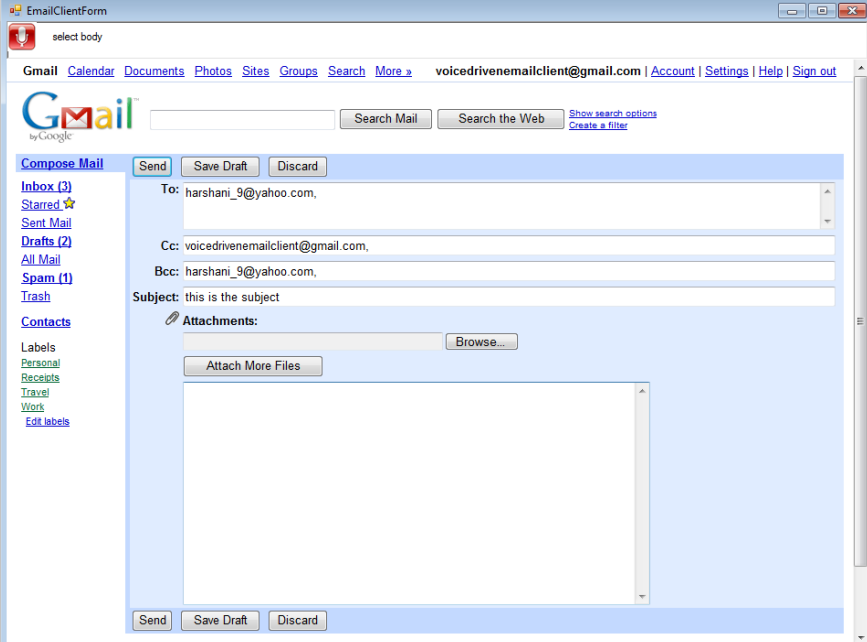
Commands:

- Undo

Comments:

Removes the last dictated text. Undo works only up to one level back

Figure 64 : Undo Mistyping on Subject



The screenshot shows the Gmail 'Compose Mail' interface. The window title is 'EmailClientForm' and the address bar shows 'select body'. The Gmail logo and navigation links are visible at the top. The 'Compose Mail' section has buttons for 'Send', 'Save Draft', and 'Discard'. The email fields are filled with: To: harshani_9@yahoo.com, Cc: voicedrivemailclient@gmail.com, Bcc: harshani_9@yahoo.com, and Subject: this is the subject. The 'Attachments' section has a 'Browse...' button and an 'Attach More Files' button. The main body text area is empty.

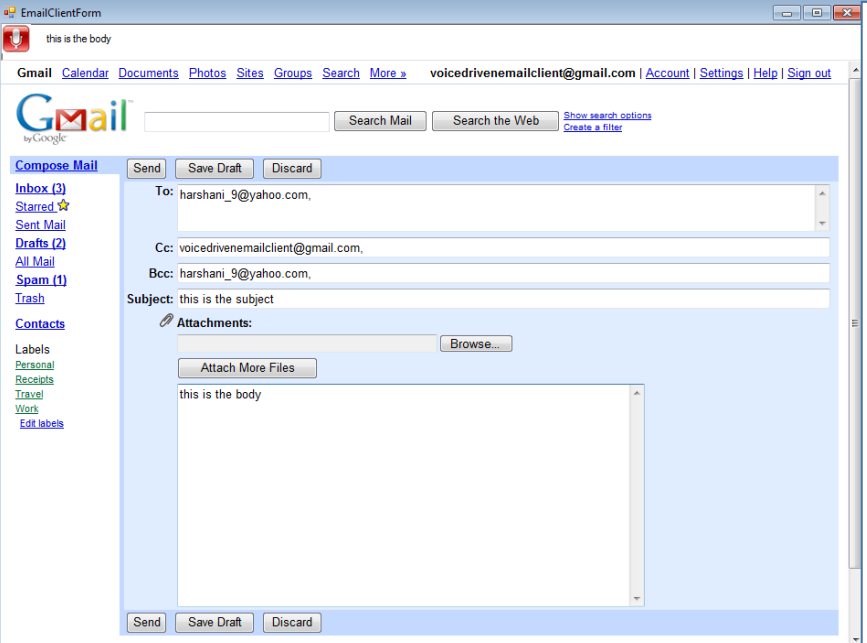
Commands:

- Select body

Comments:

Sets focus to 'body field'

Figure 65 : Select Body



The screenshot shows the Gmail 'Compose Mail' interface. The window title is 'EmailClientForm' and the address bar shows 'this is the body'. The Gmail logo and navigation links are visible at the top. The 'Compose Mail' section has buttons for 'Send', 'Save Draft', and 'Discard'. The email fields are filled with: To: harshani_9@yahoo.com, Cc: voicedrivemailclient@gmail.com, Bcc: harshani_9@yahoo.com, and Subject: this is the subject. The 'Attachments' section has a 'Browse...' button and an 'Attach More Files' button. The main body text area contains the text 'this is the body'.

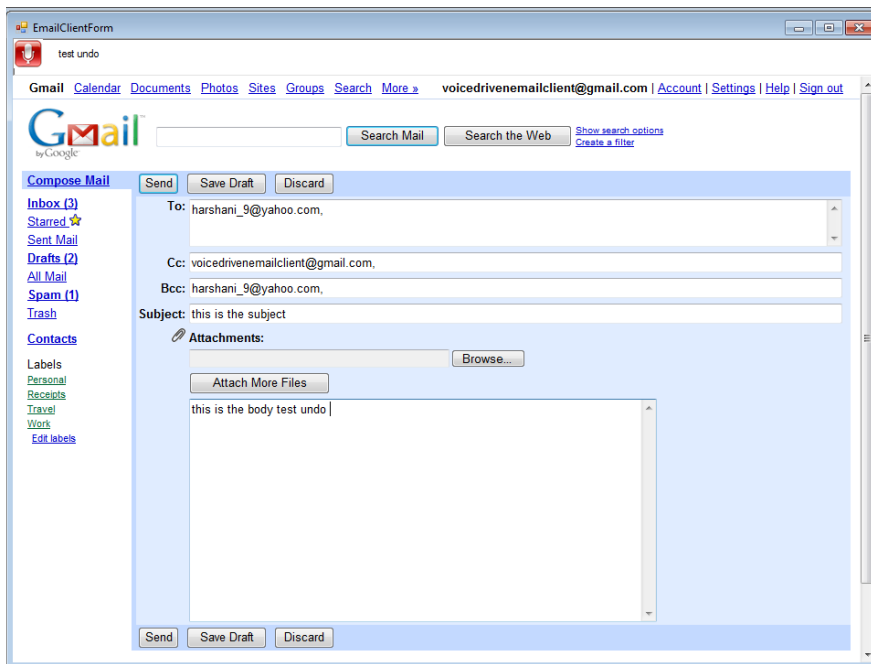
Commands:

- N/A

Comments:

Dictate body of email

Figure 66 : Dictate Body



The screenshot shows the Gmail 'Compose Mail' window. The subject line is 'this is the subject'. The body text is 'this is the body test undo |'. The interface includes a navigation menu on the left with options like 'Inbox (3)', 'Starred', 'Sent Mail', 'Drafts (2)', 'All Mail', 'Spam (1)', 'Trash', and 'Contacts'. The top navigation bar includes 'Gmail', 'Calendar', 'Documents', 'Photos', 'Sites', 'Groups', 'Search', and 'More'. The user is logged in as 'voicedrivenemailclient@gmail.com'.

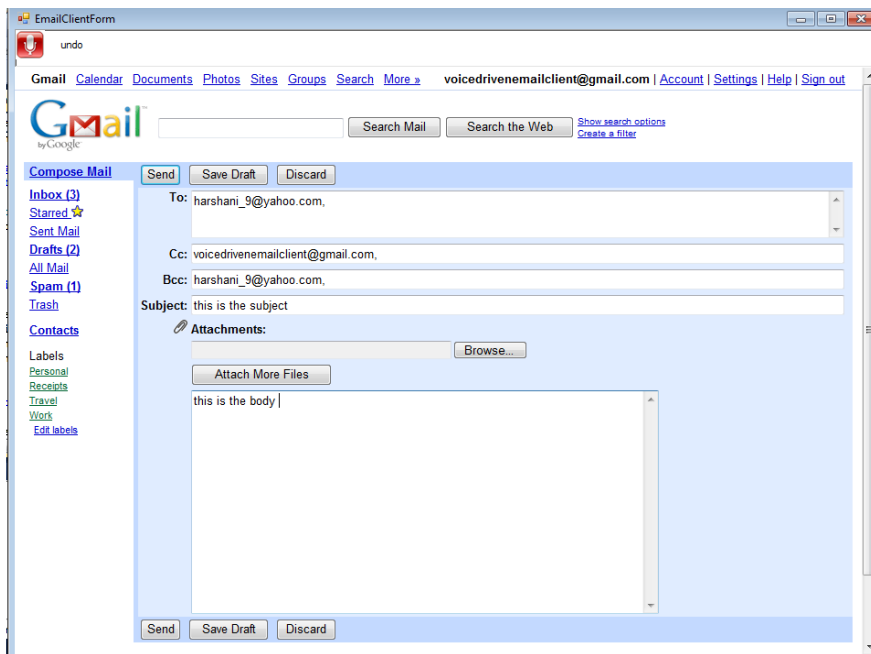
Commands:

- N/A

Comments:

Dictated subject is recognized wrong or dictates wrong phrase

Figure 67 : Mistyping on Message



The screenshot shows the same Gmail 'Compose Mail' window as Figure 67, but the subject line has been corrected to 'this is the subject' and the body text is 'this is the body |'. The 'undo' button is visible in the top left corner of the window. The rest of the interface is identical to the previous screenshot.

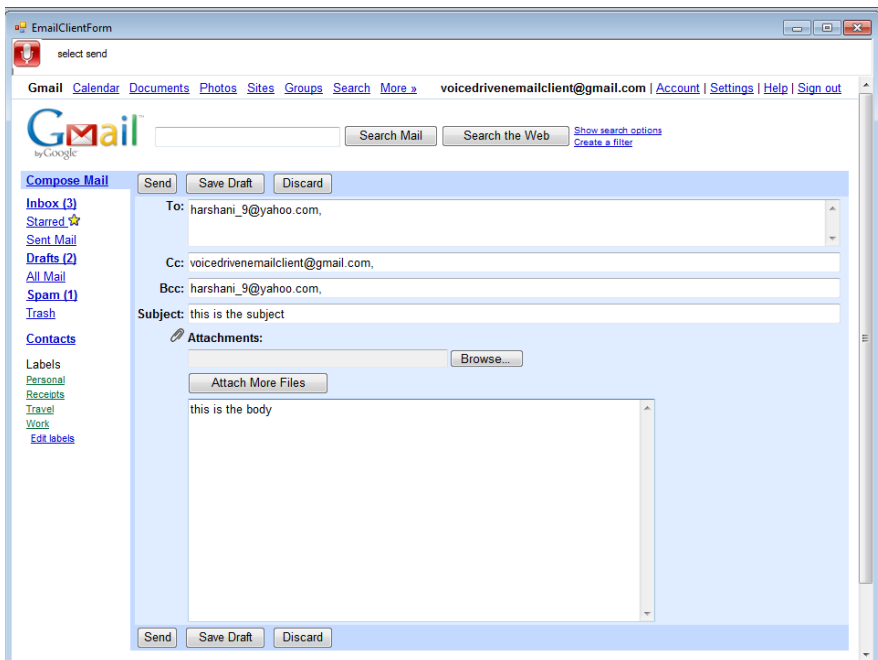
Commands:

- Undo

Comments:

Removes the last dictated text. Undo works only up to one level back

Figure 68 : Undo Mistyping on Message



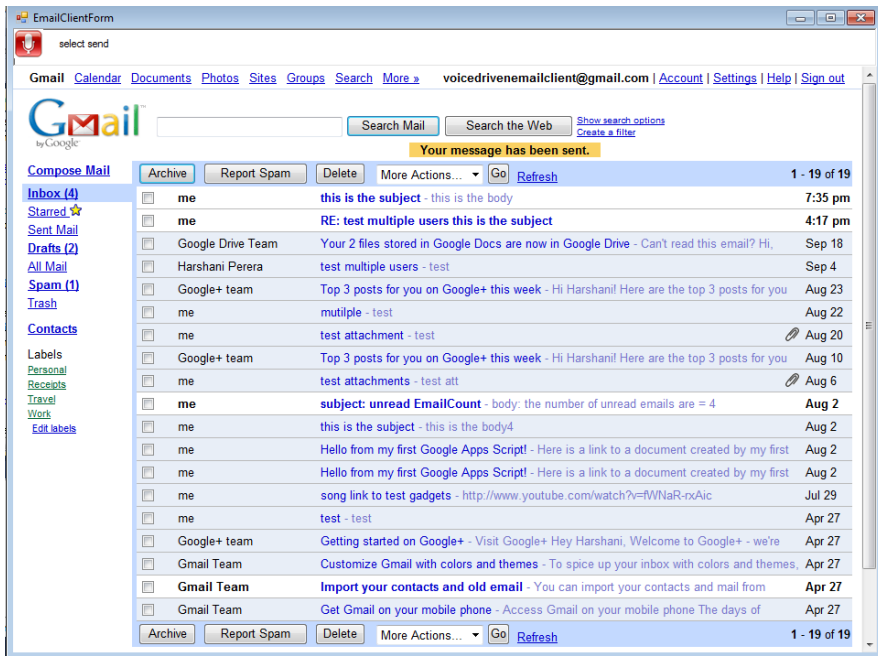
Commands:

- Select send

Comments:

N/A

Figure 69 : Send Message



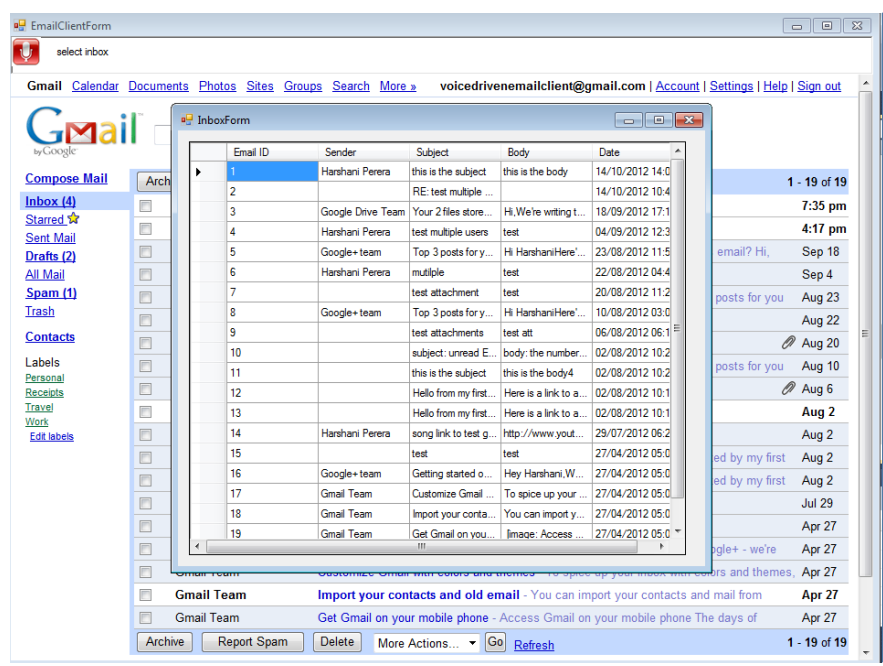
Commands:

- N/A

Comments:

Navigates to inbox and displays conformation message.

Figure 70 : Message Sent



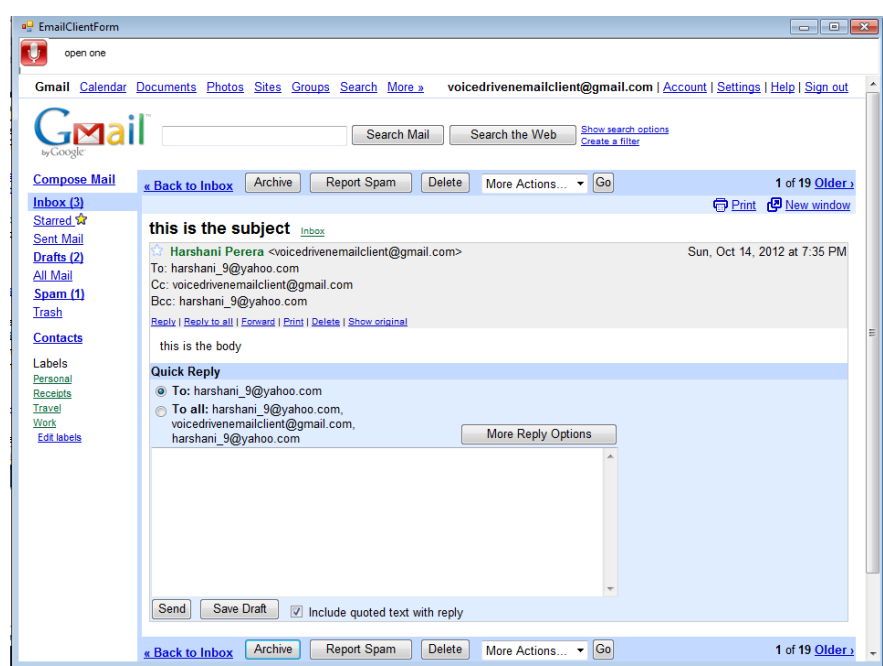
Commands:

- Select inbox

Comments:

Inbox is open in a separate form. Every email is given an id

Figure 71 : Retrieve Inbox Messages



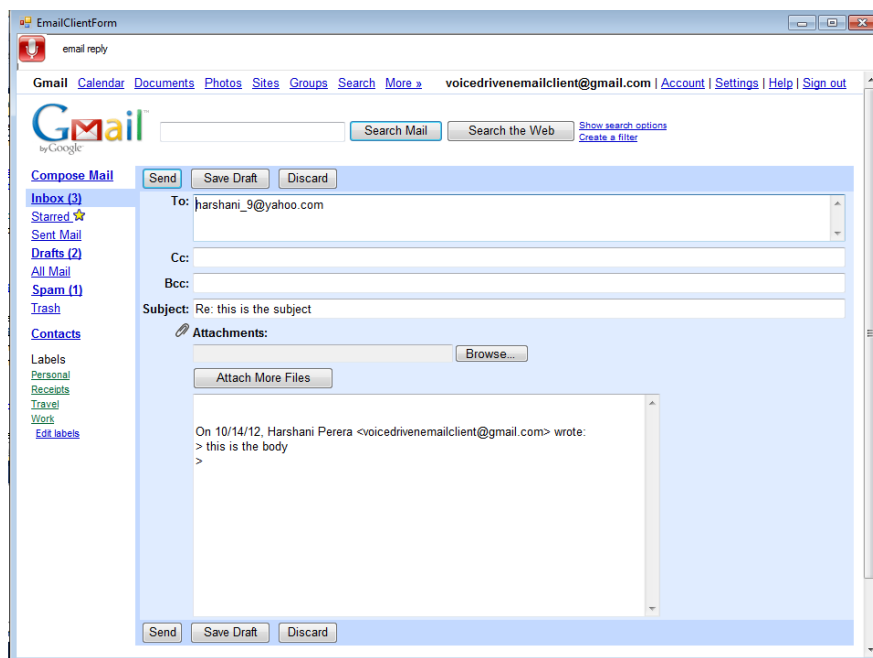
Commands:

- Open 'email id'

Comments:

Opens the voiced-in emails id

Figure 72 : Open Email



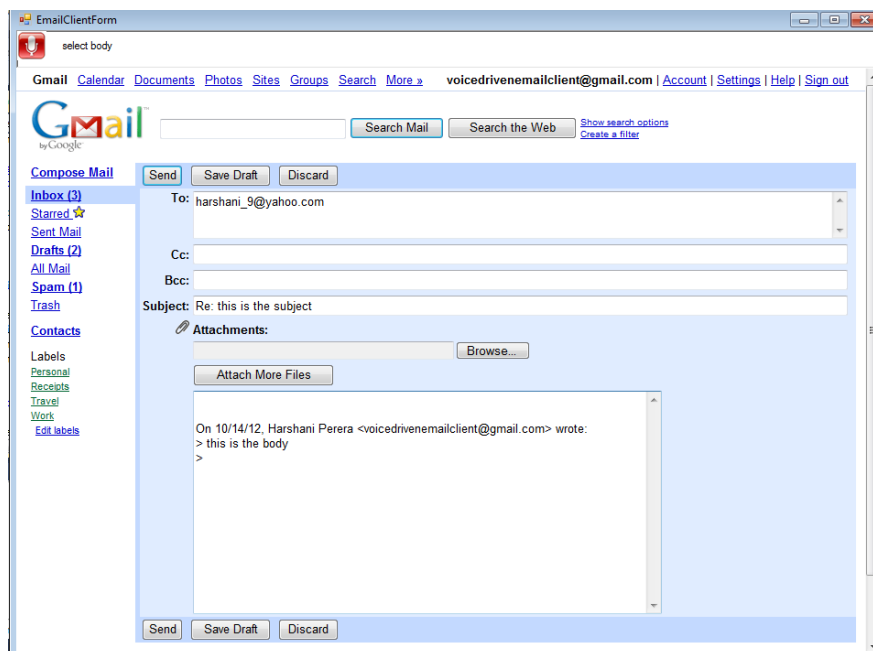
Commands:

- Email reply

Comments:

On command the reply email form opens with the required fields filled

Figure 73 : Reply to Email



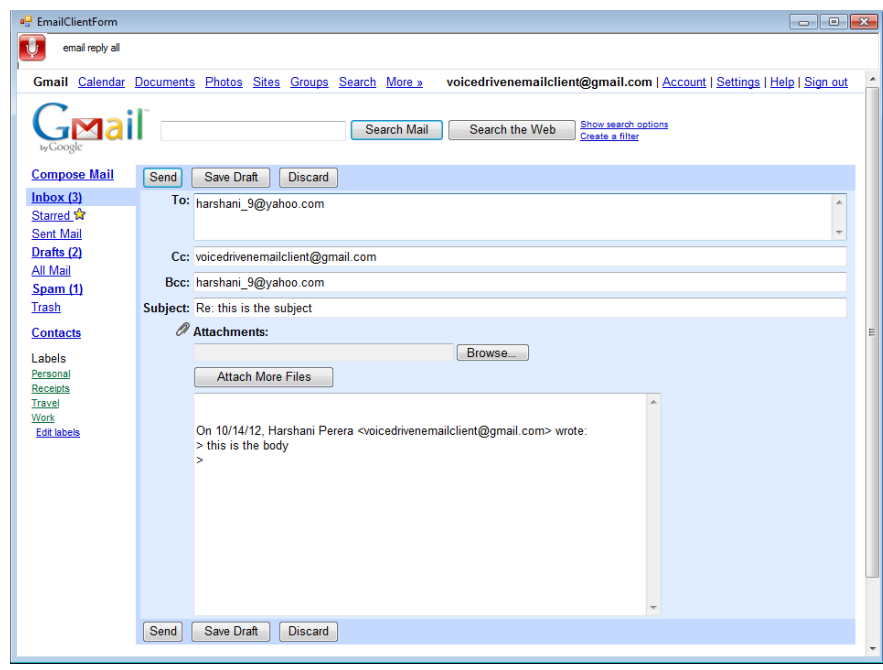
Commands:

- Same commands used for filling new email form applies for reply, reply all and forward forms

Comments:

Body gets dictated above the original body

Figure 74 : Reply/Reply all/Forward Dictating Body



The screenshot shows the Gmail 'Reply to All' form. The 'To' field is filled with 'harshani_9@yahoo.com', 'Cc' with 'voicedrivenemailclient@gmail.com', and 'Bcc' with 'harshani_9@yahoo.com'. The subject is 'Re: this is the subject'. The body contains a quoted message: 'On 10/14/12, Harshani Perera <voicedrivenemailclient@gmail.com> wrote: > this is the body >'. The 'Attachments' section is empty with a 'Browse...' button and an 'Attach More Files' button. The 'Compose Mail' header has 'Send', 'Save Draft', and 'Discard' buttons.

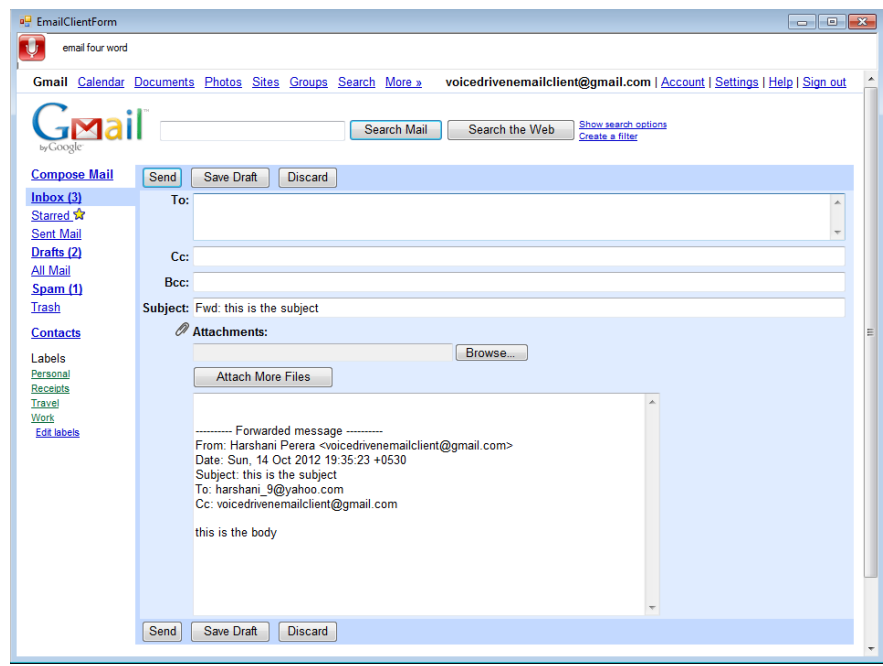
Commands:

- Email reply all

Comments:

On command the reply email all form opens with the required fields filled

Figure 75 : Reply to All



The screenshot shows the Gmail 'Forward Email' form. The 'Subject' field is filled with 'Fwd: this is the subject'. The body contains a forwarded message: '----- Forwarded message ----- From: Harshani Perera <voicedrivenemailclient@gmail.com> Date: Sun, 14 Oct 2012 19:35:23 +0530 Subject: this is the subject To: harshani_9@yahoo.com Cc: voicedrivenemailclient@gmail.com this is the body'. The 'Attachments' section is empty with a 'Browse...' button and an 'Attach More Files' button. The 'Compose Mail' header has 'Send', 'Save Draft', and 'Discard' buttons.

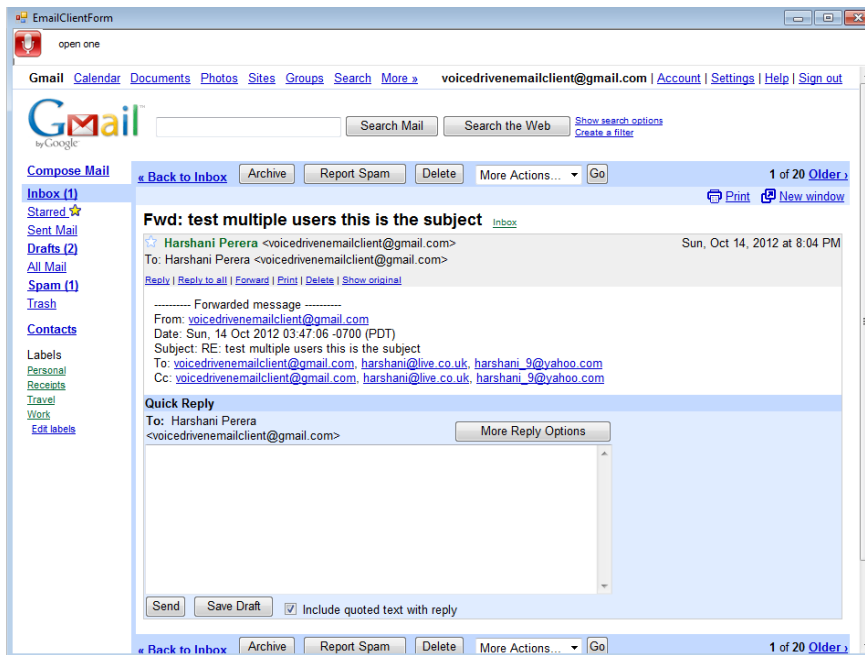
Commands:

- Forward Email

Comments:

On command the forward email form opens with the required fields filled

Figure 76 : Forward Email



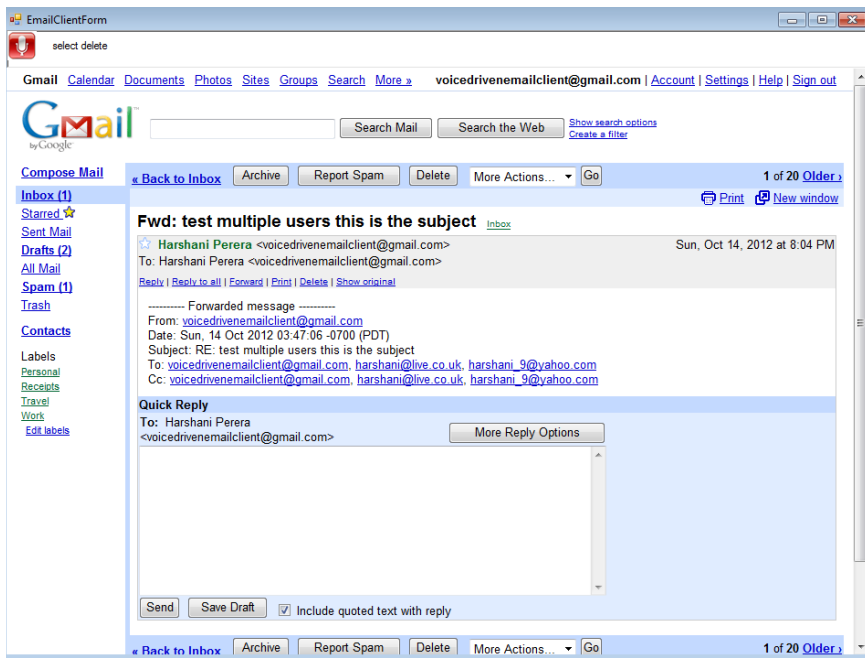
Commands:

- Open 'email id'

Comments:

Opens email intended to be deleted

Figure 77 : Delete Email - 1



Commands:

- Email delete

Comments:

Returns to inbox confirming the deletion

Figure 78 : Delete Email - 2