VPA – A Virtual Personal Assistant

Project Report submitted in partial fulfillment of the requirement
for the degree of

Bachelor of Technology.

in

Information Technology

under the Supervision of

Dr. Pooja Jain

By

Mudit Gupta (111422)

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled “VPA – A Virtual Personal Assistant”, submitted by Mudit Gupta in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: **Dr. Pooja Jain**

Associate Professor, CSE/IT Department

Acknowledgement

I would like to express my deep and sincere gratitude and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by my guide Dr. Pooja Jain, who was the staunch supporter and motivator of this project. Right from the inception of this project work, my supervisor has guided me till the very end in the true sense of the word.

I would also like to extend my gratitude to one and all who are directly or indirectly involved in the successful completion of this project report.

Date: **Mudit Gupta**

Table of Content

S. No.	Topic	Page No.
1.	**An Overview**	**2**
1.1	Introduction	2
1.2	Aim and Objectives	3
1.3	Project Vision	3
1.4	Motivation of the Project	4
1.5	Genesis of the Problem	4
2.	**Speech Synthesis**	**5**
2.1	What is Speech?	5
2.2	How to synthesize Speech	6
2.3	Java Sound API	7
2.4	Java Speech API Specifications	8
3.	**Application Designing**	**15**
3.1 JNOTEPAD		15
	3.1.1 Introduction	15
	3.1.2 Use Case Modelling	15
	3.1.3 Class Oriented Model	16
	3.1.4 Requirement Analysis Models	17
	3.1.4.1 Data Flow Diagram	17
	3.1.4.2 Behavior Modelling	20
	3.1.4.2.1 Sequence Diagram	20
	3.1.4.3 State Diagram	23

3.1.5 Usefulness and Innovation 24
3.1.6 Features of the Application 24
3.1.7 Application User Interface 25
3.1.8 Code for Email Box of Application 27
3.1.8.1 Description 38

3.2 Expression Calculator 40
3.2.1 Introduction 40
3.2.2 Designing of the Problem 40
3.2.2.1 Algorithm 40
3.2.2.2 Description of the Algorithm 41
3.2.3 Code of the Application 42

3.3 File Search 45
3.3.1 Introduction 45
3.3.2 Designing of the Problem 45
3.3.3 Code of the Application 46
3.3.4 Demonstration of the Application 49

3.4 Google Search 51
3.4.1 Introduction 51
3.4.2 Designing of the Problem 51
3.4.3 Code of the Application 53
3.4.4 Description of the Code 56
3.4.5 Demonstration 56

3.5 Application Launcher 58
3.5.1 Introduction 58
3.5.2 Designing of the Application 58
3.5.3 Code of the Application 58
3.5.4 Demonstration of the Application 60

3.6 Webcam Controller 62
3.6.1 Introduction 62
3.6.2 Designing of the Application 62

3.7 Voice Recognition 63
3.7.1 Voice Recognition Modelling 63
3.7.2 Application Design 66

3.8 Story Teller 68

4. **Conclusion and Future Work** **69**

References **70**

Useful Links **71**

List of Figures

S.No.	Title	Page No.
1.	Fig 1: Use Case of JNotepad	16
2.	Fig 2: Class Diagram of JNotepad	17
3.	Fig 3: Data Flow Diagram of JNotepad	18
4.	Fig 4: Sequence Diagram of JNotepad	20
5.	Fig 5: State Diagram of JNotepad	23
6.	Fig 6 & 7: JNotepad GUI	26
7.	Fig 8: Working of File Search	45
8.	Fig 9, 10 &11: Input Screen of File Search	49
9.	Fig 12: Application Design for Application Launcher	58

List of Tables

S.No.	Title	Page No.
1.	Table 1: Interface Description (JSAPI)	9
2.	Table 2: Class Description (JSAPI)	10
3.	Table 3: Email Box Functions	38

Abstract

The world today has become rapid and the precedence of people have reformed, now every single individual want that their toil to be completed even without moving from there fortress of harmony and leisureliness. With the evolution of technology and growth of technical peers across the globe the world has been divided into type of people; one who are technical friendly and still too lazy to do their toil and the one who are not lazy but find it much difficult to have hands on at a computer.

The VPA software is a personal assistance software which deals with both kind of people mentioned above at the same time by providing assistance in performing rudimentary tasks on the computer in a stress-free manner, making it suitable for both the classifications of people round the globe having fundamental knowledge about the English language.

The application can perform rudimentary tasks like sending an email, searching the files, calculating the mathematical expressions, launching the installed applications, searching google for data and much more which are the toils that are nothing to the current technical generation but much more complicated for the present middle aged peoples. It does not only launches the pre – installed applications but also provides the modified form of some existing applications to make the user perform certain tasks rapidly in a standalone application. The application takes text as an input and generate the corresponding response by synthesizing voice signal from the customer's machine. As the input is taken in form of text and the output is generated in form of speech signals the application can be used for specially abled too, as input can be taken by using voice recognition in case of blind people.

Chapter – 1: An Overview

1.1 Introduction

Taking down the castle one brick at a time, I would first like to discuss about the most important part of the project i.e. “speech”, what is speech? According to the expert’s opinion speech or more precisely human speech or more prevalently human voice is an expression or the ability of peoples to express thoughts and feelings by articulate sounds. The rudimentary purpose of speech for the living beings is communication, as it is the easiest and yet comprehensible. After laying the foundation of speech and the basic application of the former, I would redirect back to the project and the reason of the discussion of speech. Now it is clear that the easiest form of communication is using the speech signals so why the concept of speech is still far away from performing the toils based on computer? The answer is that it is easy for individuals to understand each other as the almighty have hardcoded our brain in such a form, but when it comes to encoding the concept for speech synthesis and recognition it becomes a million dollar problem as right now this is one of the most explored matter by goliaths like Google, Microsoft, Apple and more enterprises which are vexing to expertise the concept and mint loads of money by hook or by crook. It’s not that anyone hasn’t made any success in the field, there have been numerous mobile grounded applications which accomplishes the toil based on the user voice by taking human voice as an input signal and performing some of the rudimentary tasks.

Speech synthesis or spoken language consists of three parts which requires three programming levels, namely conceptualization, formulation and articulation. But the question is what speech is of major use to synthesize the speech audible to human ears, the audible range for human’s lies from 20 Hz- 20 KHz, and majority of the part of human speech lies within the range of 1kHz to 4kHz so we will be using this frequency range only instead of the whole range to generate the speech.

My application deals with the voice synthesis part as it takes input in the form of text from the user and perform the deed elegantly while responding by the speech synthesized from the machine. The application is capable to perform the rudimentary computer toils including sending an email, launching an application, searching a file in the machine, calculating the result of a basic mathematical expressions and much more. The performing of every rudimentary task includes an implementation of different algorithms, generation of different graphic user interfaces, which once again a topic for discussion and will be discussed in the later report one at a time.

1.2 Aim and Objectives

The project aims at developing an interactive application which can take input in form of text and can generate the output in the speech synthesized form to perform the rudimentary tasks around the computer within an easy manner. The ‘MG Wiz’ is a very easy to use so its user friendly. It can be used for various purposes depending upon the situations to suit requirements of a programmer as well as a non-programmer human. It is developed in a manner so it can be used by anyone from a world famous IT Company to an ordinary man who barely knows how to operate the computer.

1.3 Project Vision

The vision of the project is to develop an application which comes hands on for all the people round the globe, it will help people to do their tasks with an ease and in a stress free manner. It will also provide support to differently abled people which face difficulty to have their hands – on the computer.

1.4 Motivation of the Project

The best type of motivation is the one which comes from within, the motivation of this project comes from my inner self, when I see my father struggling to do basic tasks on the computer, and frankly we can see almost every individual of a an age group struggling to have hands on the computer. The motivation also comes from external sources when I saw differently abled person which have ability to do great things but have to stay put due to unavailability of proper resources and guidance.

1.5 Genesis of the Problem

Previously, person has to open multiple applications to do the tasks; for example if they have to send an email they have to do the following tasks:

i) Open a web browser.
ii) Login to the mail server.
iii) Compose the mail and finally they can send the mail.

But with this application they can just write the mail in the JNOTEPAD application and send the mail with one click. Also this standalone application can launch various applications which are existing in the system but most users don’t know how to open them. This is an ultimate alternative to the computer assistant to whom you shell out hefty sum to do your tech work. So this application can be used by anyone from a world famous IT firm to an ordinary man who barely knows how to operate the computer.

Chapter – 2: Speech Synthesis

2.1 What is Speech?

Speech is the basic way of communication between two individuals, it is defined as an expression or the ability of persons to express opinions and moods by articulate noises. It can also be defined as the most easy, yet most complex way of communication between the persons; Easy in the way that it is understandable by the persons communicating with each other belonging to the same species as they have same neural network and have same logics for understanding and complex in the manner that speech of one species is an enigma for an individual belonging to other species, for example, it is easy to understand two humans talking in front of you, but will you be able to understand what two dogs are saying if they are barking on each other, the answer is NO, for you the dogs are making nothing but noise but for them they are having a conversation. Why we can't understand what the other species were talking about, the reason is simple it is because they have a different neural network then us and we can't process the information they are sharing; for that we would require a neural network of dogs. So in my project I will be focusing upon human speech and will be trying to synthesize the human speech through the user machine, which is a difficult tasks as the neural network of computer, or the basic language which computer understands is the machine language or binary language consisting of only 0's and 1's. For this task we would be requiring to generate a program which will be able to transfer the computer basic language to form a speech understandable by human neural network. The biggest difficulty in the task is that around the globe we don't have any standard language, in fact the most spoken and most common language i.e. '*Chinese'*, is the language unknown to most of us. So in my application I will be synthesizing the second most common language across the globe, i.e., *'English',* which is understood by most of us in an easeful manner. After laying the reason for using English as the communication language we move to superior problematic situation of speech synthesis.

2.2 How to synthesize Speech?

To generate speech through computer machine in Java we have two options as follows:

i) Creating a grammar file consisting of all the alphabets, words of the English dictionary; storing them in a file with extension *".grammar"* and then generating the *".aac"* or *".wav"* file for each of the alphabet and word in the dictionary and when we want the machine to synthesize the acoustic signals which are understandable to us, we have to search every .wav file and append them one after another to generate the sound.

ii) Using a pre – existing application programming interface to synthesize the speech which already contains all the information of the characters and mechanism to generate the speech signals within them.

It is quite easy to pick the option, we would go with the second option and will be using one of the Application Programming Interface (API) to ease up the process of developing as there is no use of *'re – inventing the wheel'.* Now bigger question is which API will be used as Java has many API's compatible with it for the purpose of generating speech from the machine, here after researching a lot I came up with the conclusion that JSAPI, i.e., Java Sound Application Programming Interface is most suited to generate the speech from the computer.

The Java Speech API (JSAPI) is an API or Application Programming interface for cross – platform maintenance of command and control recognizers, dictation systems, and speech generators. There had been some third party applications which have implemented the API, for example FreeTTS. The other alternatives for speech recognition and synthesizing includes sphinx, pocket sphinx, Google API for speech recognition and synthesis, Apple API for speech recognition and synthesis, Microsoft Hidden Markov Toolkit (HTK) for speech recognition and synthesis, pocket Sphinx application for speech recognition and synthesis, FreeTTS application for speech recognition and synthesis and much more. But all faces trouble while generating speech signals and the only API which was able to generate the speech among all in an easy manner was the Java Sound API.

2.3 Java Sound API

JSAPI is the most popular API used to do speech operations in the java based applications. It is used for the following two core speech technologies:

i) *To synthesize the speech signal.*

ii) *To recognize the speech signal.*

In my application I will be concentrating majorly on the first path of speech synthesis, for which we require a basic understanding about what speech synthesis actually is, according to Wikipedia:

"*Speech synthesis provides the reverse process of producing synthetic speech from text generated by an application, an applet, or a user. It is often referred to as text-to-speech technology*."

The two major steps in producing the speech from text are:

i) Process the text, entered by the user and analyze the structure of the text, i.e., to find where the paragraphs, sentences begins and terminates which can be done easily by considering the punctuations in the language like following:

"." is used to terminate the sentence.

";" is used when we have to terminate a sentence but didn't terminate it there.

ii) Analyzing the input text for special language constructs. In English, special treatment is required for abbreviations, acronyms, dates, times, numbers, currency amounts, e-mail addresses, and many other forms. Other languages need special processing for these forms, and most languages have other specialized requirements.

A few examples of special language constructs:

a) Hello Mr. Mudit Gupta how are you
   Will be spoken as “*Hello mister Mudit Gupta how are you*”.
b) You owe me $20
   Will be spoken as “*You owe me twenty dollars*”.

2.4 Java Speech API Specifications

The API has been broadly categorized into three packages which covers all the classes and interfaces:

i) *javax.speech:* Contains classes and interfaces for a generic speech engine.
ii) *javax.speech.synthesis*: Contains classes and interfaces for speech synthesis.
iii) *javax.speech.recognition*: Contains classes and interfaces for speech recognition.

For the recognition part only javax.speech.synthesis is required so I will be using that only which have various classes and interfaces, in which I will be explaining the use of each class and interface, one by one beginning from the interfaces of the package followed by the classes in the package. I will also be explaining the engine, i.e., the java sound engine which is used to synthesize the voice or speech from the user machine and also comes with the mechanism which uses all the classes and interfaces of the javax.speech.synthesis package to get the desired results.

The table below contains the interfaces in the javax.speech.synthesis package of the Java Sound Application Programming Interface:

Interface Name	**Description of the Interface**
Speakable	An object implementing the Speakable interface can be provided to the speak method of a Synthesizer to be spoken. The text is accessed through the getJSMLText method making it the spoken equivalent of the toString method of a Java object.
SpeakableListener	The listener interface for receiving notification of events during spoken output of a Speakable. Events are requested by either: ➢ Providing a SpeakableListener object when calling one of the speak or speakPlainText methods of a Synthesizer. ➢ Attaching a SpeakableListener to a Synthesizer with its addSpeakableListener method. The speakable events and the sequencing of events is defined in the documentation for SpeakableEvent.
Synthesizer	The Synthesizer interface provides primary access to speech synthesis capabilities. The Synthesizer interface extends the Engine interface. Thus, any Synthesizer implements basic speech engine capabilities plus the specialized capabilities required for speech synthesis.The primary functions provided by the Synthesizer interface are the ability to speak text, speak Java Speech Markup Language text, and control an output queue of objects to be spoken.
SynthesizerListener	An extension to the EngineListener interface for receiving notification of events associated with a Synthesizer. SynthesizerListener objects are attached to and removed from a Synthesizer.

SynthesizerProperties	Provides control of the run-time properties of a Synthesizer. The SynthesizerProperties object is obtained by calling the getEnginePropertiesmethod of the Synthesizer (inherited from the Engine interface). Because SynthesizerProperties extends the EngineProperties interface to provide synthesizer-specific properties. It also inherits the following properties and conventions from the EngineProperties interface.

Table 1: Interface Description (JSAPI)

After descripting the interfaces in the javax.speech.synthesis package of the Java Sound Application Programming Interface I would redirect to the classes in the package, as the following table describe the classes in the package:

Class Name	**Description**
SpeakableAdapter	Adapter that receives events associated with spoken output of a Speakable object. The methods in this class are empty; this class is provided as a convenience for easily creating listeners by extending this class and overriding only the methods of interest.
SpeakableEvent	SpeakableEvents are issued to SpeakableListeners. A single SpeakableListener can be provided to any of the speak and speakPlainText methods of a Synthesizer to monitor progress of a single item on the speech output queue. Any number of SpeakableListener objects can be attached to aSynthesizer with the addSpeakableListener method. These listeners receive every SpeakableEvent for every item on the speech output queue of theSynthesizer. The SpeakableListener attached to an individual item on the

	speech output queue is notified before the SpeakableListeners attached to the Synthesizer.
SynthesizerAdapter	Adapter that receives events associated with a Synthesizer. The methods in this class are empty; this class is provided as a convenience for easily creating listeners by extending this class and overriding only the methods of interest.
SynthesizerEvent	Event issued by Synthesizer to indicate a change in state or other activity. A SynthesizerEvent is issued to each SynthesizerListener attached to aSynthesizer using the addEngineListener method it inherits from the Engine interface.The SynthesizerEvent class extends the EngineEvent class Similarly, the SynthesizerListener interface extends the EngineListener interface.
SythesizerModeDesc	SynthesizerModeDesc extends the EngineModeDesc with properties that are specific to speech synthesizers. The SynthesizerModeDesc inherits engine name, mode name, locale alongwith all the event running properties from EngineModeDesc. SynthesizerModeDesc adds two properties: ➢ List of voices provided by the synthesizer ➢ Voice to be loaded when the synthesizer is started (not used in selection)
SythesizerQueueItem	Represents an object on the speech output queue of a Synthesizer. The item is described by the source object, the speakable text, a boolean value indicating whether it is a plain text object, and the SpeakableListener for the object.The source object is the object provided to a speak method (a Speakable objects, a URL, or a String). The text is the Java Speech Markup Language string or plain text obtained from

	the source object. The listener is the SpeakableListener object passed to the speak metho.ssd, or null.

Table 2: Class Description (JSAPI)

After explaining the interfaces and classes of the java.speech.synthesizer package of Java Speech API I would now discuss the engine of Java Sound API, the engine is the interface common to java.speech.recognizer and java.speech.synthesizer package. The engine interface is the parent of all the interfaces for all the speech engines included in both Recognizer and Synthesizer. The engine implementation has all the functionalities including the following:

- ➢ Allocate and deallocate methods, here allocate method is used to Allocate the resources required for the Engine and put it into the ALLOCATED state. When this method returns successfully the ALLOCATED bit of engine state is set, and the testEngineState(Engine.ALLOCATED) method returns true. During the processing of the method, the Engine is temporarily in the ALLOCATING_RESOURCES state, whereas the deallocate method is used to Free the resources of the engine that were acquired during allocation and during operation and return the engine to the DEALLOCATED. When this method returns the DEALLOCATED bit of engine state is set so the testEngineState (Engine.DEALLOCATED) method returns true. During the processing of the method, the Engine is temporarily in the DEALLOCATING_RESOURCES state.

Pause and resume methods where pause() is used to pause the audio stream for the engine and put the Engine into the PAUSED state. Pausing an engine pauses the underlying engine for all applications that are connected to that engine. Engines are typically paused and resumed by request from a user. Applications may pause an engine indefinately. When an engine moves from the RESUMED state to the PAUSED state, an ENGINE_PAUSED event is issued to each EngineListener attached to the Engine. The PAUSED bit of the engine state is set to true when paused, and can be tested by thegetEngineState method and other engine state methods and resume() method is used to put the Engine in the RESUMED state to resume audio streaming to or from a paused engine. Resuming an engine resuming the underlying engine for *all applications* that are connected to that engine. Engines are typically paused and resumed by request from a user. The specific pause/resume behavior of recognizers and synthesizers is defined in the documentation for the pause method. When an engine moves from the PAUSED state to the RESUMED state, an ENGINE_RESUMED event is issued to each EngineListener attached to theEngine. The RESUMED bit of the engine state is set to true when resumed, and can be tested by the getEngineState method and other engine state methods.

- Access to an AudioManager which provides management of the audio input or output of the engine and VocabManager which provides management of the vocabulary for the Engine.
- Access to EngineProperties to fetch the properties of the engine.
- getEngineState to fetch an ordered set of flags indicating the current state of the engine.
- waitEngineState to block the calling thread until the engine is in the state as specified according to the user requirement.
- testEngineState is to return true if the current engine state matches the specified state. The format of the state value is described above. The test performed is not an exact match to the current state. Only the specified states are tested. For example the following returns true only if the Synthesizer queue is empty, irrespective of the pause/resume and allocation states.
- getEngineModeDesc returns a mode descriptor that defines the operating properties of the engine. For a Recognizer the return value is aRecognizerModeDesc. For a Synthesizer the return value is aSynthesizerModeDesc.
- addEngineListener requests a notification of event related to the engine. An application can attach multiple listeners to an Engine. A single listener can be attached to multiple engines. The EngineListener is extended for both

recognition and synthesis. Typically, a RecognizerListener is attached to a Recognizer and a SynthesizerListener is attached to a Synthesizer. An EngineListener can be attached or removed in any state of an Engine.

- removeEngineListener, removes a listener from this engine. An EngineListener can be attached or remove in any state of an Engine.

Being clear about each and every aspect of the application programming interface it seems to be a piece of cake to synthesize the speech from the computer, but this is not the case the process of speech synthesis is still a tricky twisted task filled with lots of errors and exception which will be faced while implementing the code to synthesize the code.

Chapter 3: Applications Designing

3.1 JNOTEPAD

3.1.1 Introduction

This application aims at developing a highly – efficient notepad name ‘JNOTEPAD’ which would cater to the needs of the people of the programming sector. The JNOTEPAD is a very easy to use so its user friendly. It can be used for various purposes depending upon the situation to suit requirements of a programmer as well as a non-programmer human. It is developed in a manner so it can be used by anyone from a world famous IT Company to an ordinary man who barely knows how to operate the computer.

3.1.2 Use Case Modelling

It helps us to describe the functional requirements of the application developed in the overall system.

Use Case: JNOTEPAD

Primary Actor: Customer

Goal: To fulfill all the requirements of the customer

Use Case Description:

Actor: User

Pre-Condition: The user must know how to open and operate a text editor.

If I select an option say open the system it must open the jnotepad where I can choose the file related features by opening an existing file or by creating a new file. If I choose create a new file I can insert the data in it may print it and then save it or save it directly and then close it., similarly if I choose open an existing file I can modify that file and save it or I can print the file or access the reading options of the system and then close the file. After closing the file I can close my system too or can start over again.

Post-Condition: The user must know how to save a file and must be able to access the file in future for further use or modification.

Use Case Diagram:

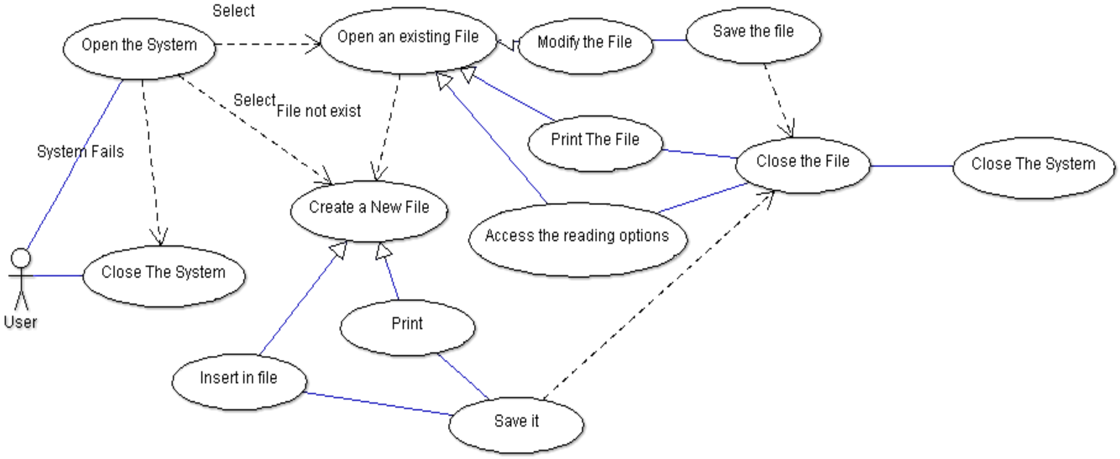


Fig 1: Use Case of JNotepad

3.1.3 Class Oriented Model

This helps developer to represent a system in form of object oriented manner in which classes collaborate to achieve the system requirements.

Classes: User

Class Diagram:

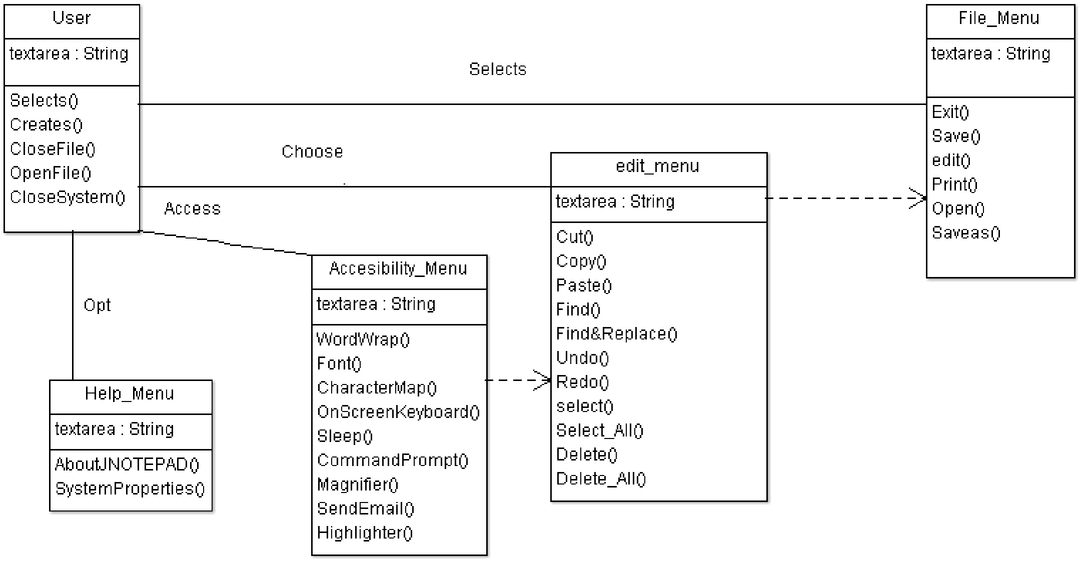


Fig 2: Class Diagram of JNotepad

3.1.4 Requirement Analysis Models

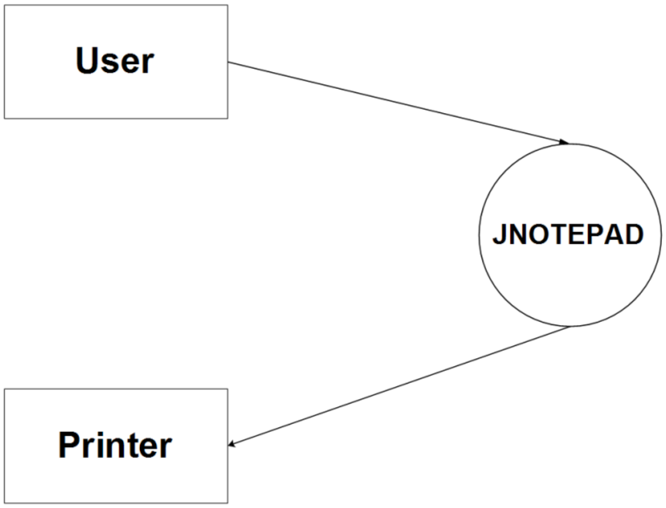
Through this phase we find the requirements more in detailed manner by examining their detailed boundary conditions & exceptional cases.

3.1.4.1 Data Flow Diagram:

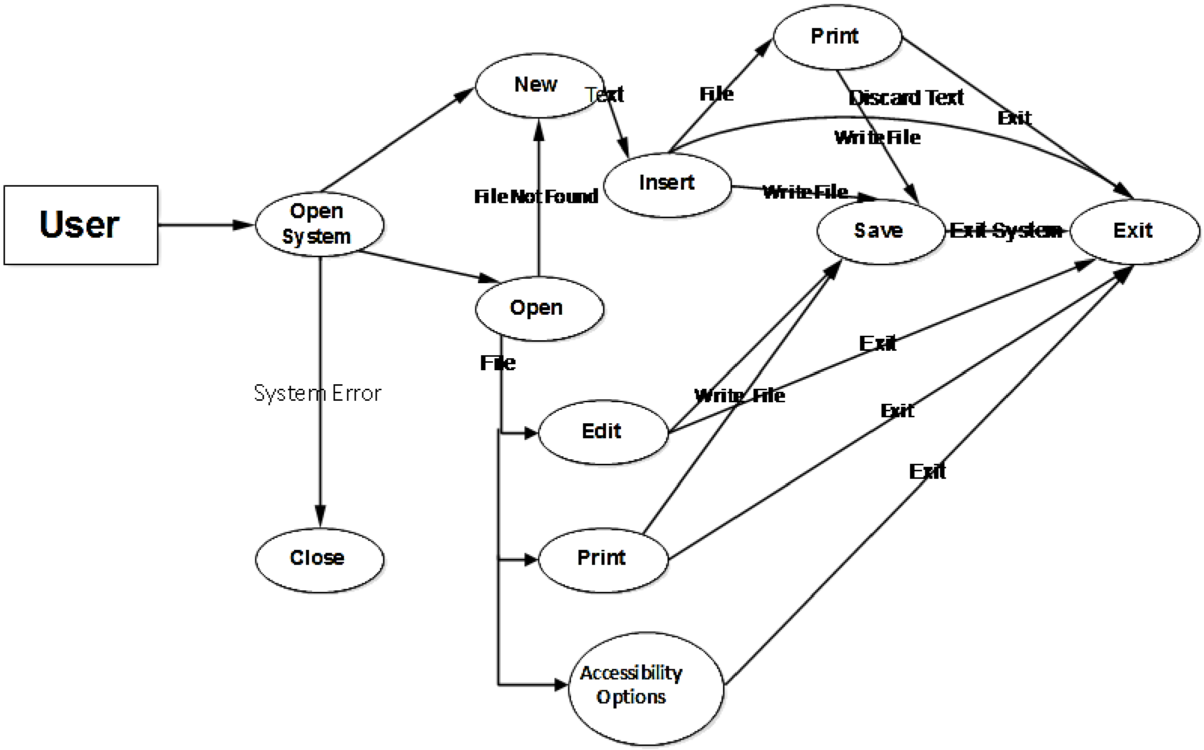
These diagrams depict the flow of data from one point of the system to another point. It mainly consists of three parts divided on the basis of its level:

a) *Level '0' Data Flow Diagram (Context Diagram).*
b) *Level '1' DFD.*
c) *Level '2' DFD.*

Level ‘0’ DFD or Context Diagram:



Level ‘1’ DFD:



Level ‘2’ DFD:

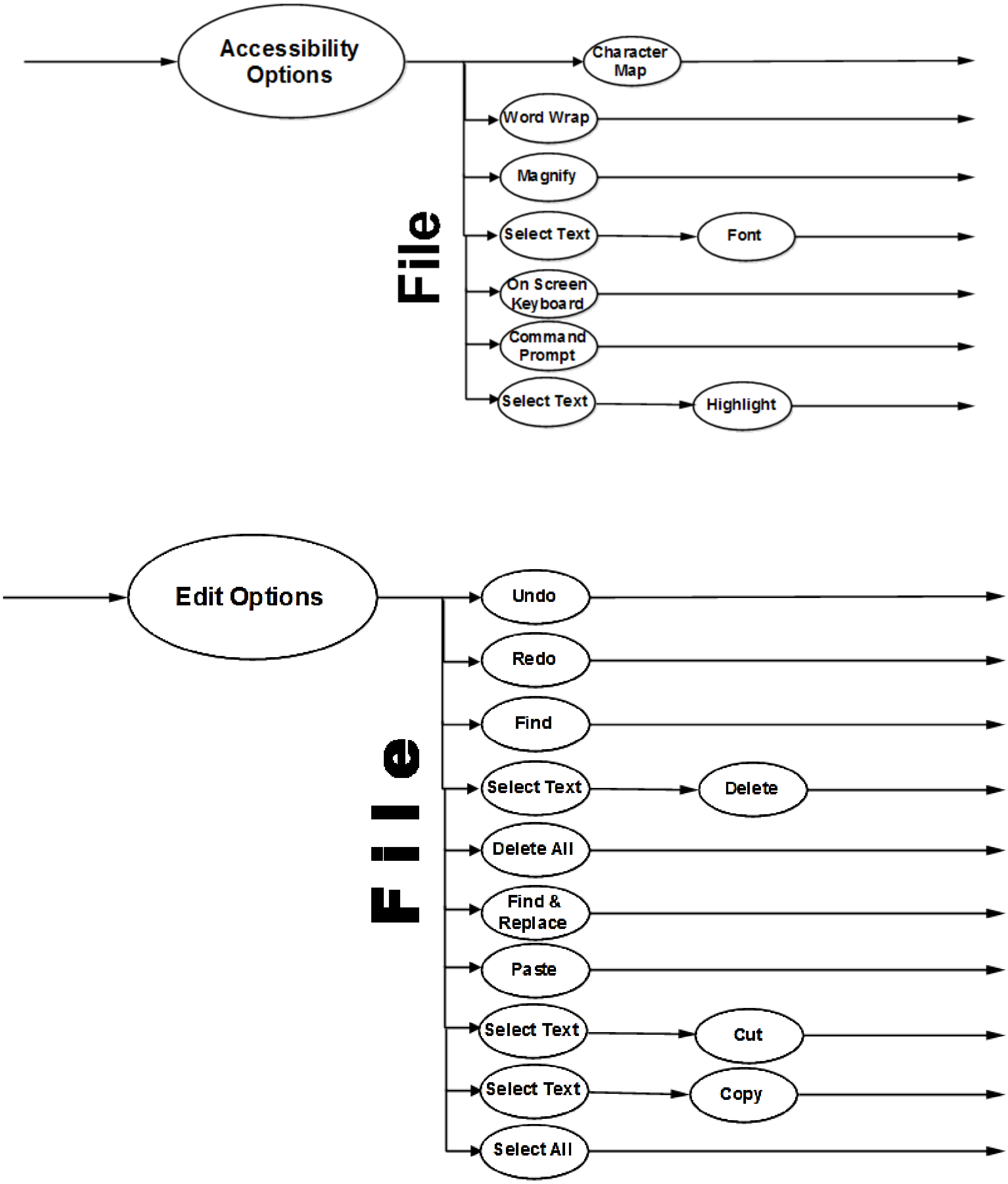
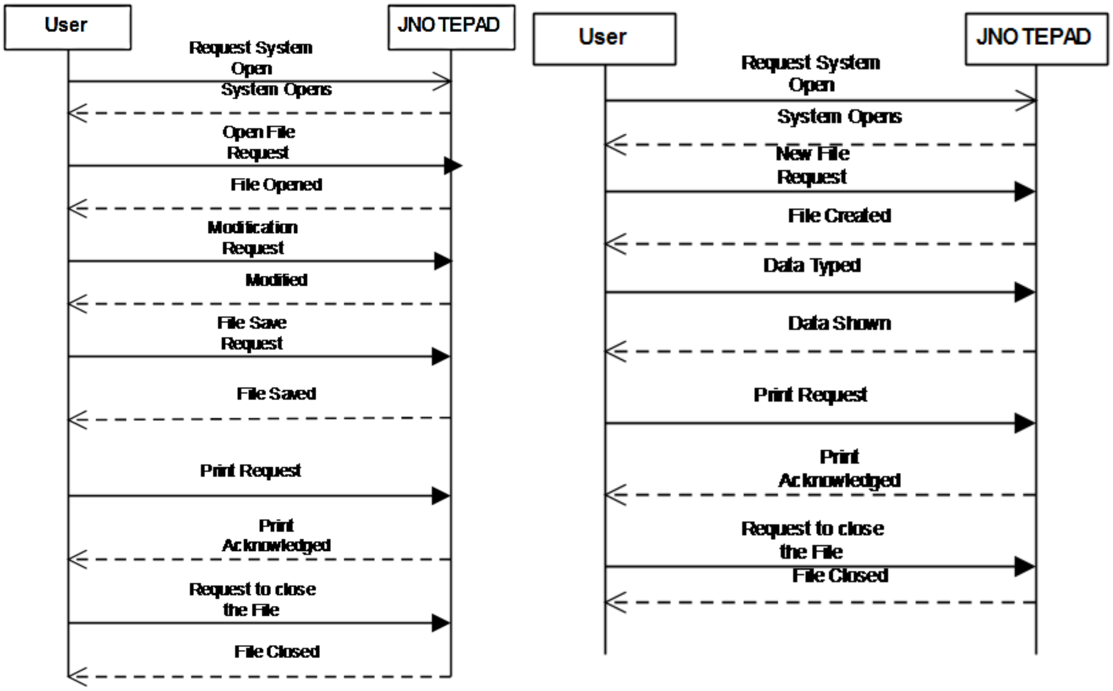


Fig 3: Data Flow Diagram of JNotepad

3.1.4.2 Behavior Modeling:

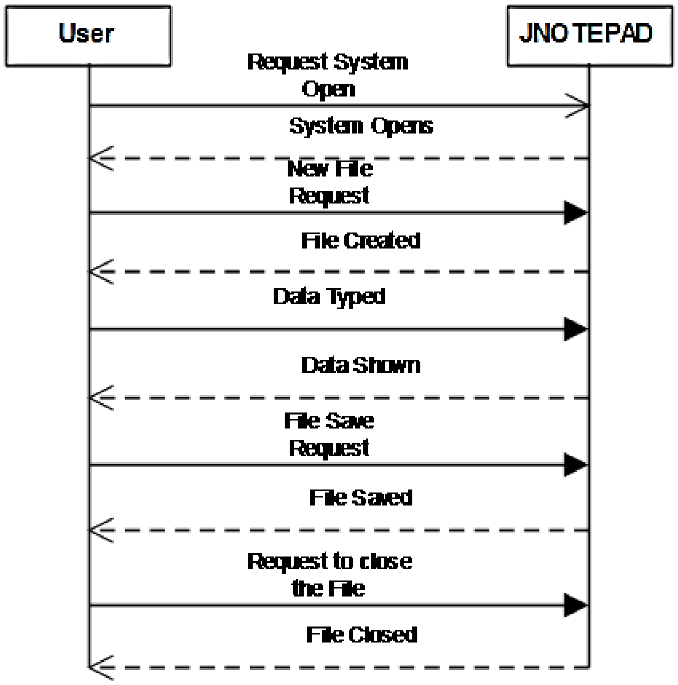
It tells us about how system behaves & how users use it. In this we have to use ‘Use-Cases’ & ‘Use-Case Description’ for which you have to develop steps. It includes two types of Diagrams

3.1.4.2.1 Sequence Diagrams:

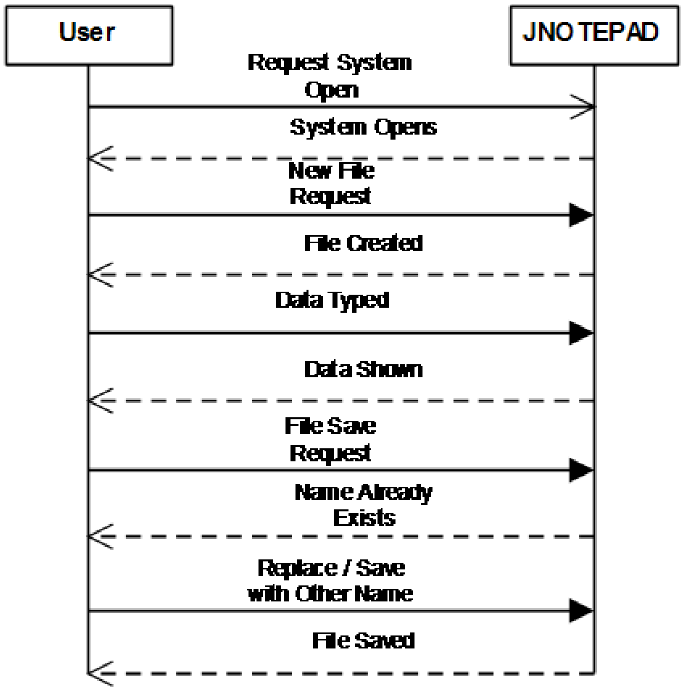


Open a File Sequence Diagram **Printing a New File Sequence Diagram**

Fig 4: Sequence Diagram of JNotepad

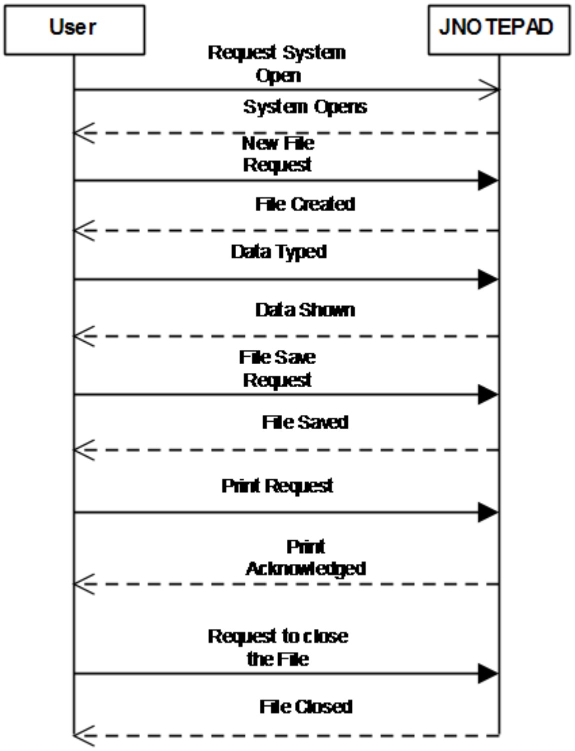


Creating and Saving a New File Sequence Diagram

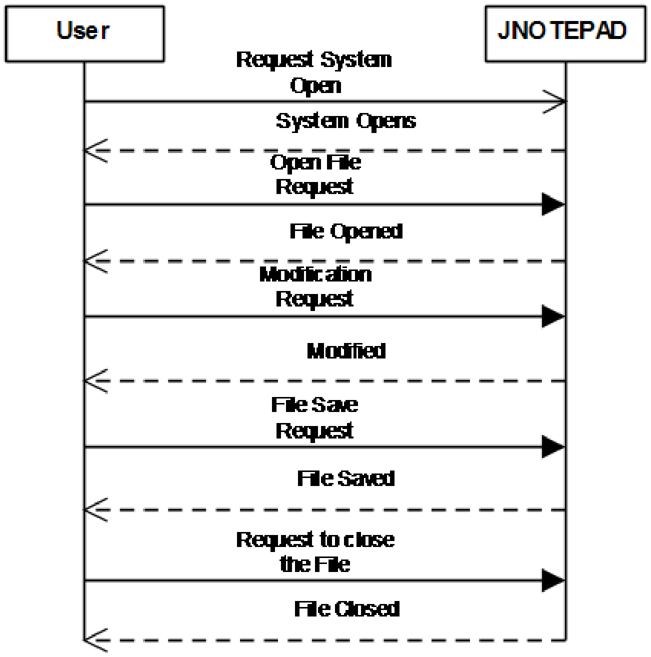


Creating and Saving a New File Sequence Diagram

(Extended Scenario)



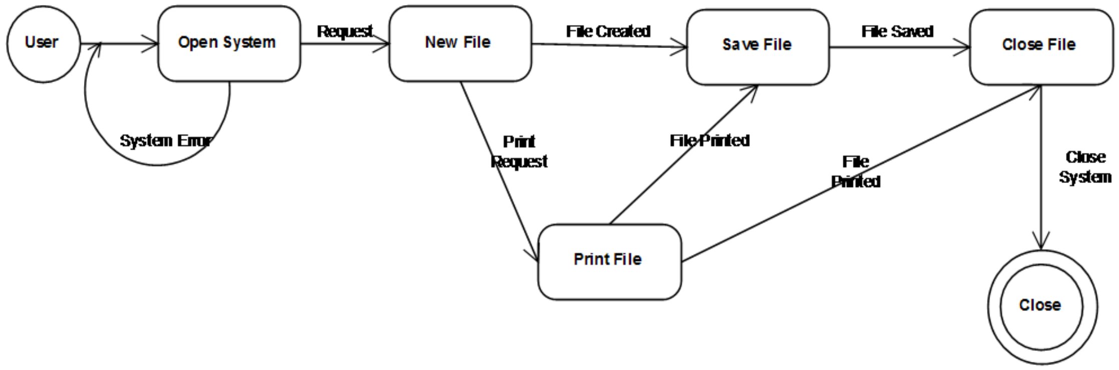
Printing & Saving a New File Sequence Diagram



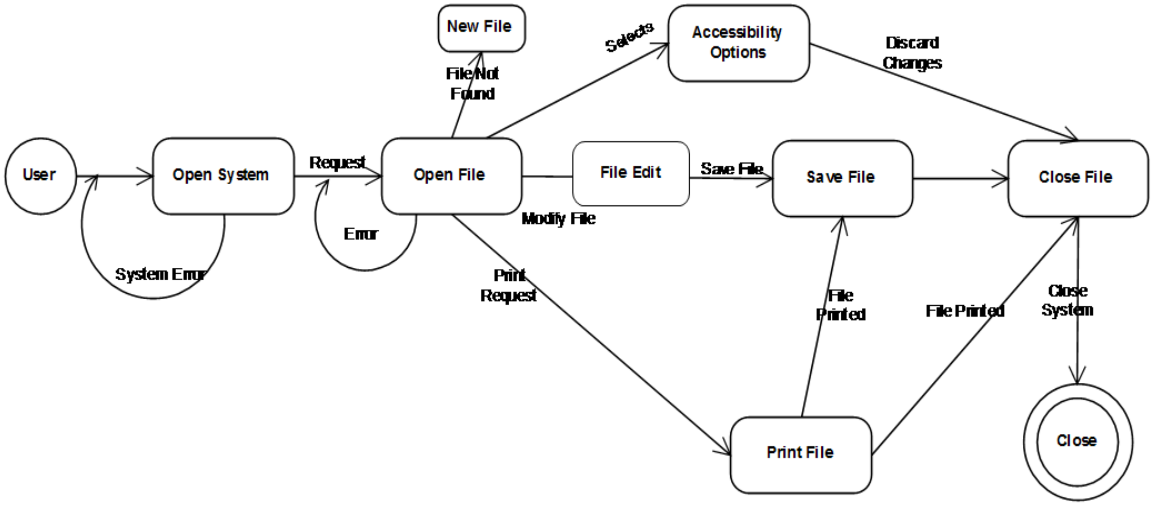
Modifying an Existing File Sequence Diagram

3.1.4.3 State Diagram:

It is useful when an object has a dynamic behavior. It specifies the sequence of states in which an object can exist along with the events and conditions that causes transition between the states.



State Diagram: Creating a New File



State Diagram: Opening an Existing File

Fig 5: State Diagram of JNotepad

3.1.5 Usefulness and Innovation

This software developed totally according to programmers & Web Developers perspective which include powerful tools for them to use. It practically allows us to save file in any format and to make them readable in the software used to read them and even modification can be done through it. It includes very nice accessibility functions to help every age-group, specially it has an magnifier for elders if they want to read a text without changing the size of it and to all those new generation programmers they have to explore all of its features but it even includes features from making their computers sleep at that time to checking its properties.

3.1.6 Features of the Application

JNOTEPAD is an advanced tool that allows you to view and edit HTML, CSS, JavaScript, PHP, C, C++, PERL, SQL and many other languages source codes. It supports all the features of a windows notepad and also a ton of extra functions to make it way better than the windows notepad.While it can serve as a perfect replacement for windows Notepad, it also offers many powerful features for webpage authors and programmers. It also have some additional features than any of the existing notepad in the market at current state, following are some of the features of the JNOTEPAD application present in our system.

i) *Unlimited text & line size:* It supports literally unlimited text and can still work like at light speed.

ii) *Inbuilt Character Map:* It has an inbuilt character map so the webpage authors may have no problem inserting special characters and symbols into it.

iii) *Printing:* Both black and white and colored printout can be achieved through it.

iv) *Drag & Drop Support:* It supports all the edit options of Cut, Copy, Paste and Move.

v) *One Click Erase:* It can be cleared all at one instance without selecting and then deleting.

vi) *Status Bar:* It includes status bar to tell you your current column number and line number on which you are working.

vii) *Command Prompt:* It supports direct link to command prompt so you don't have to find it in start menu and can directly use it to run the java codes through jdk.

viii) *On Screen Keyboard:* The application supports On Screen Keyboard so that your work or your programming continues even if keyboard isn't working.

ix) *Magnifier:* To magnify the text in the application.

x) *Sleep*: To put the machine into sleep mode

xi) *Highlight*: To highlight the text in the file.

xii) *Send Email:* To send the file as an email to any of your friend or coulleagues.

xiii) *Alignment:* To set the alignment of the text.

3.1.7 Application User Interface

As the application has got its interface created the following descripts the run through of the application, the Figure 6 depicts the basic user view of the application, Figure 7 depicts the file open preview of the application, demonstarion shows the various menu of the application including the accesinility menu which includes the options of On Screen Keyboard, Magnifier, Sleep, Highlight, Send Email, Character Map and a few more. The application also has file menu including the basic items like new, open, save, save as and more. The Edit menu of the application includes the cut, copy, paste, delete all features to make it easy to use for the user. The user interface of the application is pretty easy and user will find easy to use the application both for regular work and for industrial work if required.

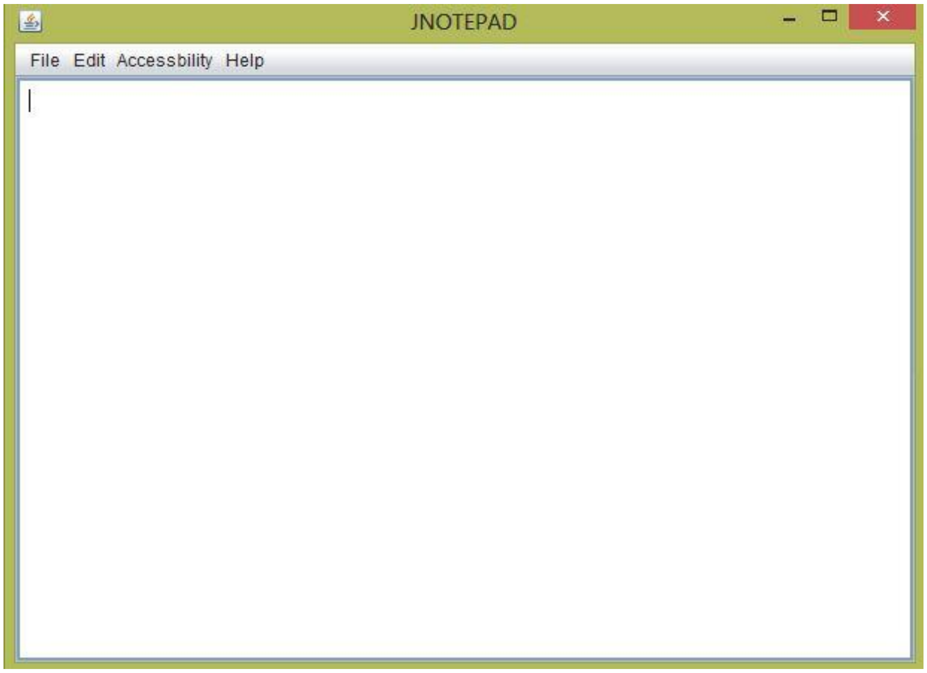


Figure 6: JNotepad GUI

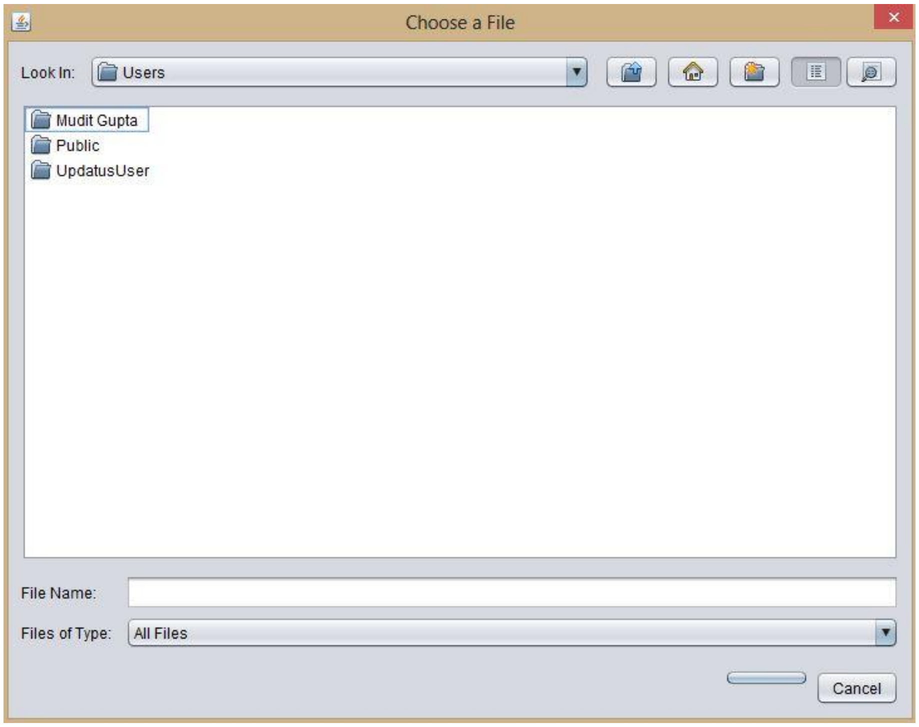


Figure 7: Jnotepad GUI

3.1.8 Code for Email box of Application

```
package notepad.notepad;

import java.util.Properties;

import javax.mail.Message;

import javax.mail.MessagingException;

import javax.mail.PasswordAuthentication;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage;

import notepad.notepad.NewJFrame;

public class EmailBox extends javax.swing.JFrame{

   String str = new String();

   public EmailBox() {

       initComponents();

   }

   public EmailBox(String str) {

     initComponents();

       this.str= str;

   }

   @SuppressWarnings("unchecked")

   //  <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
```

```
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();

    YourEmailId = new javax.swing.JTextField();

    jLabel2 = new javax.swing.JLabel();

    YourPassword = new javax.swing.JPasswordField();

    jLabel3 = new javax.swing.JLabel();

    ReciepientsEmail = new javax.swing.JTextField();

    jLabel4 = new javax.swing.JLabel();

    Subject = new javax.swing.JTextField();

    Send = new javax.swing.JButton();

    Cancel = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setText("Your Email Id");

    YourEmailId.addActionListener(new java.awt.event.ActionListener() {

        public void actionPerformed(java.awt.event.ActionEvent evt) {

            YourEmailIdActionPerformed(evt);

        }

    });

    jLabel2.setText("Password");

    jLabel3.setText("Receipient's Id");

    ReciepientsEmail.addActionListener(new java.awt.event.ActionListener() {

     public void actionPerformed(java.awt.event.ActionEvent evt) {

        ReciepientsEmailActionPerformed(evt);
```

```
            }
        });
        jLabel4.setText("Subject");
        Send.setText("Send");
        Send.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                SendActionPerformed(evt);
            }
        });
        Cancel.setText("Cancel");
        Cancel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                CancelActionPerformed(evt);
            }
        });
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(32, 32, 32)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(Send)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel1)
            .addComponent(jLabel2)
            .addComponent(jLabel3)
            .addComponent(jLabel4)))
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addGap(28, 28, 28)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
              .addComponent(YourEmailId)
              .addComponent(YourPassword)
              .addComponent(ReciepientsEmail)
              .addComponent(Subject,
javax.swing.GroupLayout.PREFERRED_SIZE,                              211,
javax.swing.GroupLayout.PREFERRED_SIZE)))
          .addGroup(layout.createSequentialGroup()
            .addGap(101, 101, 101)
            .addComponent(Cancel)))
        .addContainerGap(20, Short.MAX_VALUE))
    );
```

```
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
                    .addComponent(jLabel1)
                    .addComponent(YourEmailId,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
                    .addComponent(jLabel2)
                    .addComponent(YourPassword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
                    .addComponent(jLabel3)
                    .addComponent(ReciepientsEmail,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
                    .addComponent(jLabel4)
                    .addComponent(Subject,       javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
                    .addComponent(Send)
                    .addComponent(Cancel))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents
    private void YourEmailIdActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_YourEmailIdActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_YourEmailIdActionPerformed

    private void ReciepientsEmailActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ReciepientsEmailActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_ReciepientsEmailActionPerformed

    private void CancelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_CancelActionPerformed
this.dispose();        // TODO add your handling code here:
    }//GEN-LAST:event_CancelActionPerformed

    private void SendActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_SendActionPerformed
String semail=YourEmailId.getText();
String pass=YourPassword.getText();
String remail=ReciepientsEmail.getText();
String subject=Subject.getText();
NewJFrame em = new NewJFrame();
String msg=em.get();
System.out.println(str);
```

```
final String username = semail;

final String password = pass;

Properties props = new Properties();

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.starttls.enable", "true");

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

props.put("mail.smtp.port", "587");

props.put("mail.smtp.user", "username");

props.put("mail.smtp.password", "password");

Session session;

session = Session.getDefaultInstance(props,new javax.mail.Authenticator()

{

   @Override

   protected PasswordAuthentication getPasswordAuthentication()

   {

      return new PasswordAuthentication(username,password);


   }

});

session.setDebug(true);

try

{

         Message message=new MimeMessage(session);
```

```
                    message.setFrom(new InternetAddress(semail));
                    message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(remail));
                    message.setSubject(subject);
                    message.setText(str);
                    Transport.send(message);
            System.out.println("Email is sent successfully");
        }
        catch(MessagingException e)
        {
            throw new RuntimeException(e);
        }
        this.dispose();
    }//GEN-LAST:event_SendActionPerformed

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

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

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(EmailBox.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new EmailBox().setVisible(true);
            }
        });
    }
      private javax.swing.JButton Cancel;
    private javax.swing.JTextField ReciepientsEmail;
    private javax.swing.JButton Send;
    private javax.swing.JTextField Subject;
    private javax.swing.JTextField YourEmailId;
    private javax.swing.JPasswordField YourPassword;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
  }
```

3.1.8.1 Description:

The above code for email box is a basic code with a connection established only with the google mail server, here the user will prompt to enter their own email id, the password of the email id they are having followed with the recipient's email id and subject; The message delivered in this case will be the text written in the file of the JNOTEPAD application. The main Application Programming interface used in the module are the follows:

i) Javax.mail application programming interface version 1.5.0.
ii) Pop3 application programming interface version 1.5.0
iii) smtp application programming interface version 1.5.0

The following table describes the main functions used in the email box module, here message is the object of Message class and its of type Mime message.

Function Name	**Description of Function**
props.put( )	To set the properties of the mail server.
message.setReciepients( )	To set the reciepient of the message.
message.setSubject( )	To set the subject of the message to be sent by the user.
message.setText( )	To set the message body of the email to be sent by the user or to be received by the recipient.
message.setFrom( )	This helps to set the email id from which the email will be sent, it has two parameters namely: *Email:* to define the sender's email id. *Password:* to define the password of sender's email id.

message.setRecipients( )	This will help us to set the email address of the recipient, It can be a single address or multiple addresses as per the requirement of the user.

Table 3: Email Box Functions

The above table briefly explains the functions of the Application Programming Interface which we have to use in our program, but the most important thing in sending a mail is to generate a session, use a session and finally destroy the session is taken care using the session management algorithms in java which are very helpful for both the user as well as the application to avoid unwanted breaches, incomings and stuff like that or of similar order.

The session management has the following main steps:

i) Create Session: To create a session for the mail i.e., to establish link with mail server.

ii) Start Session: To start a session which will describe when the session starts and at what time it will be destroyed.

iii) Fetch Session Values: To fetch all the keys pressed, mouse clicked and all the user data generated by the user in a session.

iv) Destroy the Session: To remove all global session variables and to destroy the session.

With the basic knowledge of session it is easy to understand that how the session of mail in the email box is maintained in the above module, now I would move on to the next application included in the VPA software, which is an expression calculator.

3.2 Expression Calculator

3.2.1 Introduction

This application will assist in computing the rudimentary arithmetic expressions submitted by the client, this calculator has an ability to perform rudimentary toils of addition, multiplication, subtraction and division which are obligatory to be performed on a circadian regime basis.

3.2.2 Designing of the Problem

The application is designed in java using the basic Shunting Yard algorithm, which is method of parsing mathematical expression submitted by the user in an infix notation It can be used to produce output in Reverse Polish notation (RPN) or as an abstract syntax tree (AST). The algorithm was invented by Edsger Dijkstra and named the "shunting yard" algorithm because its operation resembles that of a railroad shunting yard.

3.2.2.1 Algorithm:

i) While there are tokens to be read:

ii) Read a token.

iii) If the token is a number, then add it to the output queue.

iv) If the token is a function token, then push it onto the stack.

v) If the token is a function argument separator (e.g., a comma):

vi) Until the token at the top of the stack is a left parenthesis, pop operators off the stack onto the output queue. If no left parentheses are encountered, either the separator was misplaced or parentheses were mismatched.

vii) If the token is an operator, o_1, then:

viii) While there is an operator token, o_2, at the top of the stack, and

a. either o_1 is left-associative and its precedence is *less than or equal* to that of o_2,

b. or o_1 if right associative, and has precedence *less than* that of o_2,

c. then pop o_2 off the stack, onto the output queue;

ix) Push o_1 onto the stack.

x) If the token is a left parenthesis, then push it onto the stack. If the token is a right parenthesis:

xi) Until the token at the top of the stack is a left parenthesis, pop operators off the stack onto the output queue. Pop the left parenthesis from the stack, but not onto the output queue.

xii) If the token at the top of the stack is a function token, pop it onto the output queue.

xiii) If the stack runs out without finding a left parenthesis, then there are mismatched parentheses.

xiv) When there are no more tokens to read: While there are still operator tokens in the stack:

xv) If the operator token on the top of the stack is a parenthesis, then there are mismatched parentheses.

xvi) Pop the operator onto the output queue.

xx) Exit.

3.2.2.2 Description of the Algorithm

To analyze the running time complexity of this algorithm, one has only to note that each token will be read once, each number, function, or operator will be printed once, and each function, operator, or parenthesis will be pushed onto the stack and popped off the stack once – therefore, there are at most a constant number of operations executed per token, and the running time is thus O(n) – linear in the size of the input.

The shunting yard algorithm can also be applied to produce prefix notation (also known as polish notation). To do this one would simply start from the end of a string of tokens to be parsed and work backwards, reverse the output queue (therefore making the output queue an output stack), and flip the left and right parenthesis behavior (remembering that the now-left parenthesis behavior should pop until it finds a now-right parenthesis).

3.2.3 Code of the Application

```
import java.util.Scanner;

public class Calc {

    public static void main(String args[]){

                    int j=0;

                    float op1=0,op2=0,result=0;

                    Scanner inp=new Scanner(System.in);

                     System.out.println("Sir,Please enter the expression");

                     String exp=inp.next();

                     //String exp = expq.replaceAll(""," ");

                     System.out.println(exp);

                      for(int i=0;i<exp.length();i++){

                                     if(exp.charAt(i)=='+'||exp.charAt(i)=='-
                        '||exp.charAt(i)=='*'||exp.charAt(i)=='/'){

                           if(exp.charAt(i)=='+'){

                                   i=j;
    op1=Float.parseFloat(exp.substring(0, j+1));
    op2=Float.parseFloat(exp.substring(j+2,exp.length()));
    result=op1+op2;
    System.out.println(result);
    break;
}
if(exp.charAt(i)=='*'){
    i=j;
    op1=Float.parseFloat(exp.substring(0, j+1));
    op2=Float.parseFloat(exp.substring(j+2,exp.length()));
    result=op1*op2;
    System.out.println(result);
    break;
}
if(exp.charAt(i)=='-'){
    i=j;
    op1=Float.parseFloat(exp.substring(0, j+1));
    op2=Float.parseFloat(exp.substring(j+2,exp.length()));
    result=op1-op2;
    System.out.println(result);
    break;
}
if(exp.charAt(i)=='/'){
```

```
                    i=j;

                    op1=Float.parseFloat(exp.substring(0, j+1));

                    op2=Float.parseFloat(exp.substring(j+2,exp.length()));

                    if(op2==0){

                        System.err.println("Division by zero is not possible");

                        break;

                    }

                    else{

                       result=op1/op2;

                       System.out.println(result);

                       break;

                    }

              }

            }

        }

    }

}
```

Output:

```
Sir, please enter the expression

2+3

5.0
```

3.3 File Search

3.3.1 Introduction

The file search application is used to search the files on the computer, all you have to specify that which folder might have the file in it and pass the filename with its extension as input to the application, if the directory provided contains the file it will search the file and show the final output as a string to user containing the location of the file which we want to search in our machine.

3.3.2 Designing of the problem

The design of the file search application is easy it takes two argument, i.e., the possible position of the folder in which you want to find the file and the filename along with the extension of the file. The possible path of the file is stored in the string which is passed to the search directory function as first argument and filename as the second argument. Then the file name is saved with the file type and passed to search file function which searches everywhere for the file in the subdirectory submitted by the user and it returns the file location if found in the directory, else returns that no such file exists.

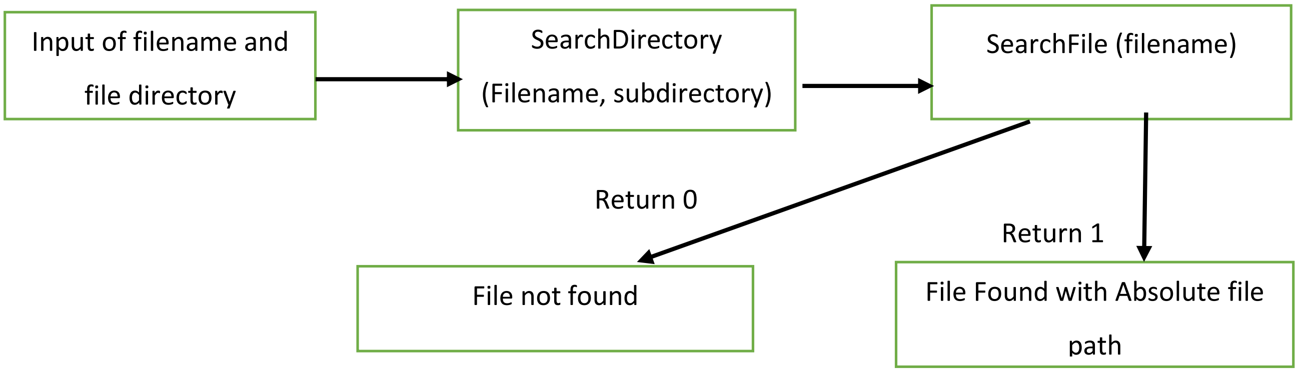

Input of filename and file directory
SearchDirectory (Filename, subdirectory)
SearchFile (filename)
Return 0
Return 1
File not found
File Found with Absolute file path

Figure 8: Working of File Search

3.3.3 Code of the Application

```
import java.io.File;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;


public class FileSearch {

   private String fileNameToSearch;

   private List<String> result = new ArrayList<String>();

   public String getFileNameToSearch() {

      return fileNameToSearch;

   }

   public void setFileNameToSearch(String fileNameToSearch) {

      this.fileNameToSearch = fileNameToSearch;

   }

   public List<String> getResult() {

      return result;

   }

   public static void main(String[] args) {

      FileSearch fileSearch = new FileSearch();

      String fname = new String();
```

```
Scanner bucky = new Scanner(System.in);

System.out.println("Enter the exact file name with extension");

fname = bucky.next();

System.out.println("Enter the exact folder");

String folder = bucky.next();

System.out.println("Which Drive?");

String d = bucky.next();

if (d.equalsIgnoreCase("c")) {

   fileSearch.searchDirectory(new File("/" + folder), fname);

}

if (d.equalsIgnoreCase("d")) {

   fileSearch.searchDirectory(new File("D:/" + folder), fname);

}

//try different directory and filename :)

//fileSearch.searchDirectory(new File("/"+folder),fname);

int count = fileSearch.getResult().size();

if (count == 0) {

   System.out.println("\nNo result found!");

} else {

   System.out.println("\nFound " + count + " result!\n");

   for (String matched : fileSearch.getResult()) {

      System.out.println("Found : " + matched);

   }
```

```
    }
}
public void searchDirectory(File directory, String fileNameToSearch) {
    setFileNameToSearch(fileNameToSearch);
    if (directory.isDirectory()) {
        search(directory);
    } else {
        System.out.println(directory.getAbsoluteFile() + " is not a directory!");
    }
}
private void search(File file) {
    if (file.isDirectory()) {
        System.out.println("Searching directory ... " + file.getAbsoluteFile());
        //do you have permission to read this directory?
        if (file.canRead()) {
            for (File temp : file.listFiles()) {
                if (temp.isDirectory()) {
                    search(temp);
                } else {
                    if (getFileNameToSearch().equals(temp.getName().toLowerCase())) {
                        result.add(temp.getAbsoluteFile().toString());
                    }
                }
                }
            } else {
                System.out.println(file.getAbsoluteFile() + "Permission Denied");
            }
        }
    }
}
```

3.3.4 Demonstration of the Application

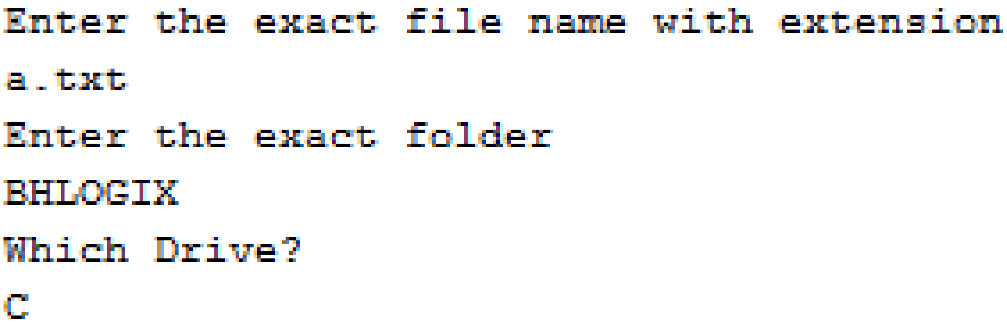
The application will search the file in the computer, let’s say you want to search a file say a.txt somewhere in your machine. It will take a total of 3 inputs:

i) Filename, a.txt

ii) Drive, C

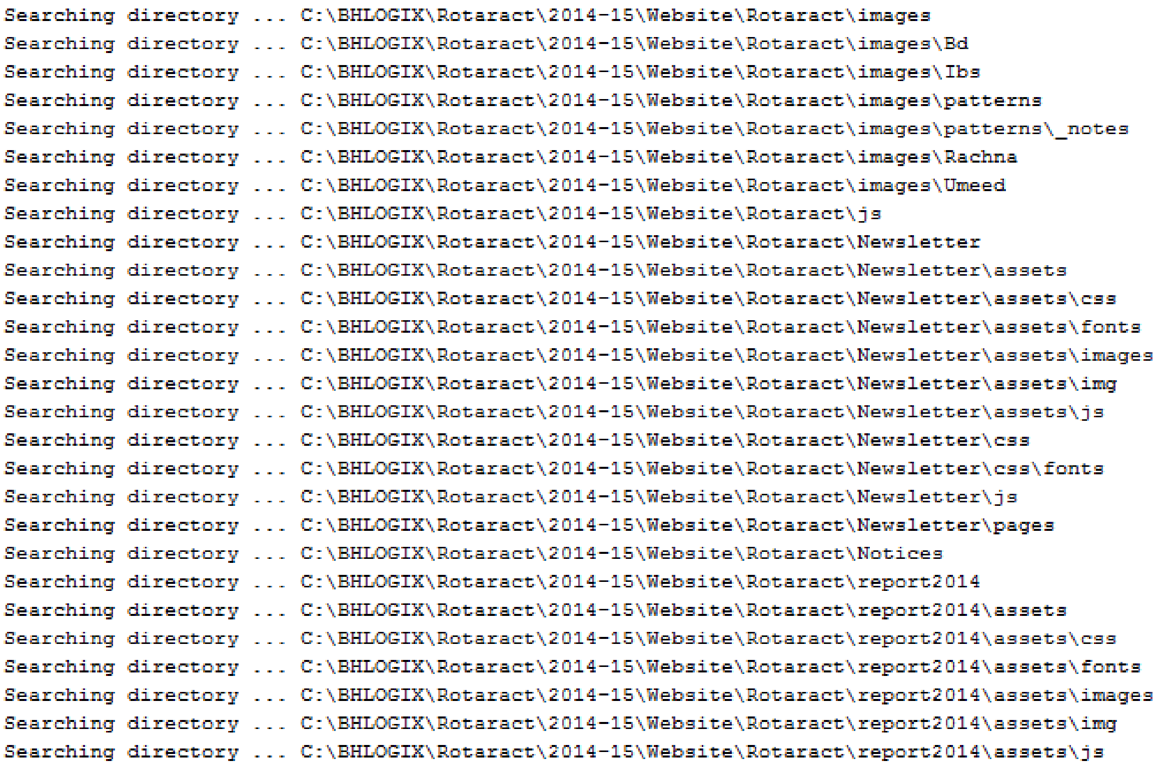
iii) Folder, BHLOGIX

And will show the following output:



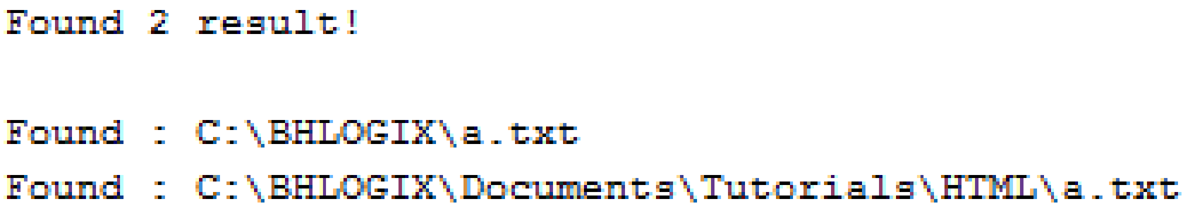
```
Enter the exact file name with extension
a.txt
Enter the exact folder
BHLOGIX
Which Drive?
C
```

Figure 9: Input Screen of File Search



```
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\images
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\images\Bd
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\images\Ibs
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\images\patterns
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\images\patterns\_notes
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\images\Rachna
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\images\Umeed
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\js
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\assets
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\assets\css
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\assets\fonts
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\assets\images
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\assets\img
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\assets\js
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\css
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\css\fonts
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\js
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Newsletter\pages
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\Notices
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\report2014
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\report2014\assets
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\report2014\assets\css
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\report2014\assets\fonts
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\report2014\assets\images
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\report2014\assets\img
Searching directory ... C:\BHLOGIX\Rotaract\2014-15\Website\Rotaract\report2014\assets\js
```

Figure 10: Input Screen of File Search



```
Found 2 result!

Found : C:\BHLOGIX\a.txt
Found : C:\BHLOGIX\Documents\Tutorials\HTML\a.txt
```

Figure 11: Input Screen of File Search

3.4 Google Search

3.4.1 Introduction

This application has the power to search and retrieve google results without opening the browser, it will be a useful application when we want a small data related to some topic and don’t want to open a new browser to do so, especially when you have been performing some performance exclusive task on your machine, which will reduce the performance of your system. This application comes handy, also when the machine we are operating can’t do much of multitasking.

3.4.2 Designing of the Application

The application involves use of the application programming interface named, GSON version 2.3.1 which is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Gson can work with arbitrary Java objects including pre-existing objects that you do not have source-code of. There are a few open-source projects that can convert Java objects to JSON. However, most of them require that you place Java annotations in your classes; something that you cannot do if you do not have access to the source-code. Most also do not fully support the use of Java Generics. Gson considers both of these as very important design goals.

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language

JSON is built on two structures:A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.

- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are worldwide data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

The GSON Application Programming Interface consists mainly of 4 packages and each package has many classes each having multiple methods and numerous implementations, because it's a very large API and discussing about the API alone will be requiring a significant amount of time, so I will keep the discussion short by discussing only those methods which have been used by me in the google search module. The discussion will be carried out in the later part where I will take down the entire code one by one.

Before going to coding section I would describe the goals of GSON, why it was launched by the Google Corporation.

i) Provide simple toJson() and fromJson() methods to convert Java objects to JSON and vice-versa.

ii) Allow pre-existing unmodifiable objects to be converted to and from JSON.

iii) Extensive support of Java Generics.

iv) Allow custom representations for objects.

v) Support arbitrarily complex objects (with deep inheritance hierarchies and extensive use of generic types)

3.4.3 Code of the Application

```
import java.io.IOException;

import java.io.InputStreamReader;

import java.io.Reader;

import java.net.URL;

import java.net.URLEncoder;

import java.util.List;

import com.google.gson.Gson;



public class GoogleSearch {

        public static void main(String[] args) throws IOException {

                String                    address                    =
                "http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=";

                String query = "facebook";

                String charset = "UTF-8";

                URL url = new URL(address + URLEncoder.encode(query, charset));

                Reader reader = new InputStreamReader(url.openStream(), charset);
```

```
            GoogleResults results = new Gson().fromJson(reader,
GoogleResults.class);

            int total = results.getResponseData().getResults().size();

            System.out.println("total: "+total);

            // Show title and URL of each results

            for(int i=0; i<=total-1; i++)

            {

            System.out.println("Title: " + results.getResponseData().getResults().get
            (i).getTitle());

            System.out.println("URL: " + results.getResponseData().getResults().get
(i).getUrl() + "\n");

            }

      }

}


class GoogleResults

{

  private GoogleResults.ResponseData responseData;

  public GoogleResults.ResponseData getResponseData() { return responseData; }
```

```
   public void setResponseData(GoogleResults.ResponseData responseData) {
this.responseData = responseData; }

   public String toString() { return "ResponseData[" + responseData + "]"; }

    static class ResponseData {

       private List<GoogleResults.Result> results;

       public List<GoogleResults.Result> getResults() { return results; }

       public void setResults(List<GoogleResults.Result> results) { this.results = results; }

       public String toString() { return "Results[" + results + "]"; }

   }

   static class Result

{

       private String url;

       private String title;

       public String getUrl() { return url; }

       public String getTitle() { return title; }

       public void setUrl(String url) { this.url = url; }

       public void setTitle(String title) { this.title = title; }

       public String toString() { return "Result[url:" + url +",title:" + title + "]"; }    }

}
```

3.4.4 Description of the Code

The code takes the keyboard from the user, let’s say user entered facebook as the string to search. The code will be executed following the below mentioned steps:

i) Then the string will be passed to the address where to search the keyword which is stored in the address keyword of the above code.

ii) The url is generated containing the query and address of the keyword.

iii) The generated URL is then passed to the input stream reader which is used to provide input to the google search query.

iv) The input is passed to the results object of the Google Results Class which then gathers the results and display it to the user

3.4.5 Demonstration

The code will display the following result when executed in Netbeans:

Output:

```
Title: Welcome to <b>Facebook</b> - Log In, Sign Up or Learn More

URL: https://www.facebook.com/
```

Title: <b>Facebook</b>

URL: https://www.facebook.com/login

Title: About - <b>Facebook</b>

URL: https://www.facebook.com/facebook

Title: <b>Facebook</b> - Android Apps on Google Play

URL:
https://play.google.com/store/apps/details%3Fid%3Dcom.facebook.katana%26hl%3Den

3.5 Application Launcher

3.5.1 Introduction

The application launcher is an application which permits user to launch the application in the computer in one click, i.e., the user can launch practically every application in the machine using an artless application. It is obliging exclusively when user find it problematic to locate the directory of application to be launched and also supportive in the cases where user is naïve of the applications which came pre – installed in the user machine, it is basically the second category of persons as discussed in the beginning of the report who faces the second problem, while the first problem is faced by both categories of persons.

3.5.2 Designing of the Application

The application designing involves use of drag and drop feature of java swings, in this all we have to do is to add the button for each application, and launch the applications accessing the predefined java execute function.

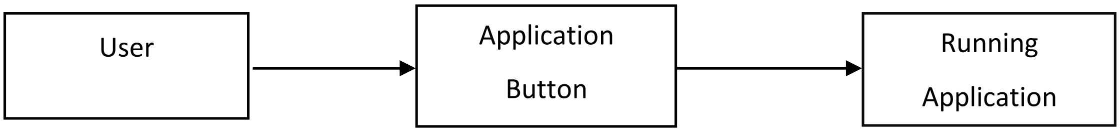


Figure 12: Application Design of Application Launcher

The functionality of this application is pretty much clear with the above diagram. After explaining the designing of this rudimentary application, I would describe the code snippet of the application.

3.5.3 Code of the Application

The following section of the report consists of the snippet of the code, to open an application say Matlab

```
private void matlabActionPerformed(java.awt.event.ActionEvent evt)
{
 try
 {
        String path = "C:\\Program Files\\MATLAB\\R2012a\\bin\\matlab.exe";
        Runtime runtime = Runtime.getRuntime();
        if (path=="C:\\Program Files\\MATLAB\\R2012a\\bin\\matlab.exe")
                 runtime.exec("explorer.exe "+path);
        else
                JOptionPane.showMessageDialog(a,   "Matlab   application   not
                installed\n"
                                + "or you have the software in some\n"
                                + " other directory than the default one");
     }
catch (Exception E)
{
        System.out.println("Error / Exception Occured");
   }
   }
```

Here this code opens up the Matlab application in the user machine, this version of code is for a 64 – bit machine which are more popular these days, and I also have designed a 32 – bit version of the same code, which will launch the set of applications in a 32 – bit machine without any trouble; In cases there exists no such application in the user machine it will generate a pop up window saying that the application you want to launch is not present in your machine and in case there is an existing application it will also show a prompt that it might be present in some other directory than the default directory, there are two major reasons why the application is not searching where the executable file is located in the user machine as follows:

i) Time take to search the application is going to be very huge, so the resulting wait time for the user who wants to perform a search operation will increase exponentially resulting in goliath latency.

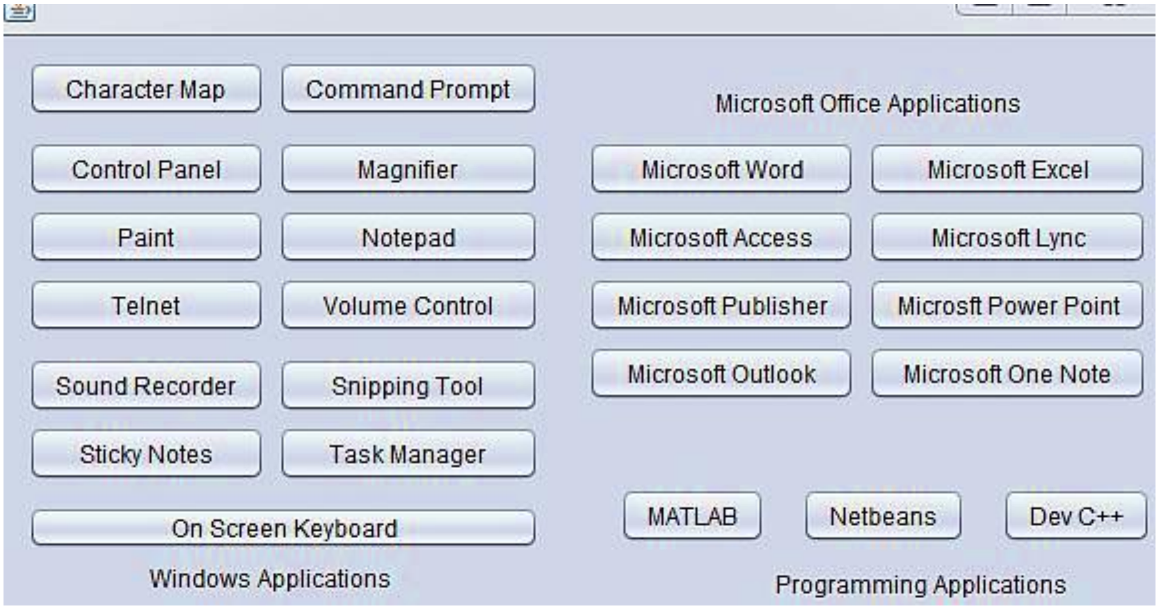
ii) If the user has knowledge to install the application other than the default directory, he must know where the directory is and where the file is located.

The above mentioned code was the snippet of the overall code which enable us to launch virtually every application available in our machine with a single click access making it a cake walk.

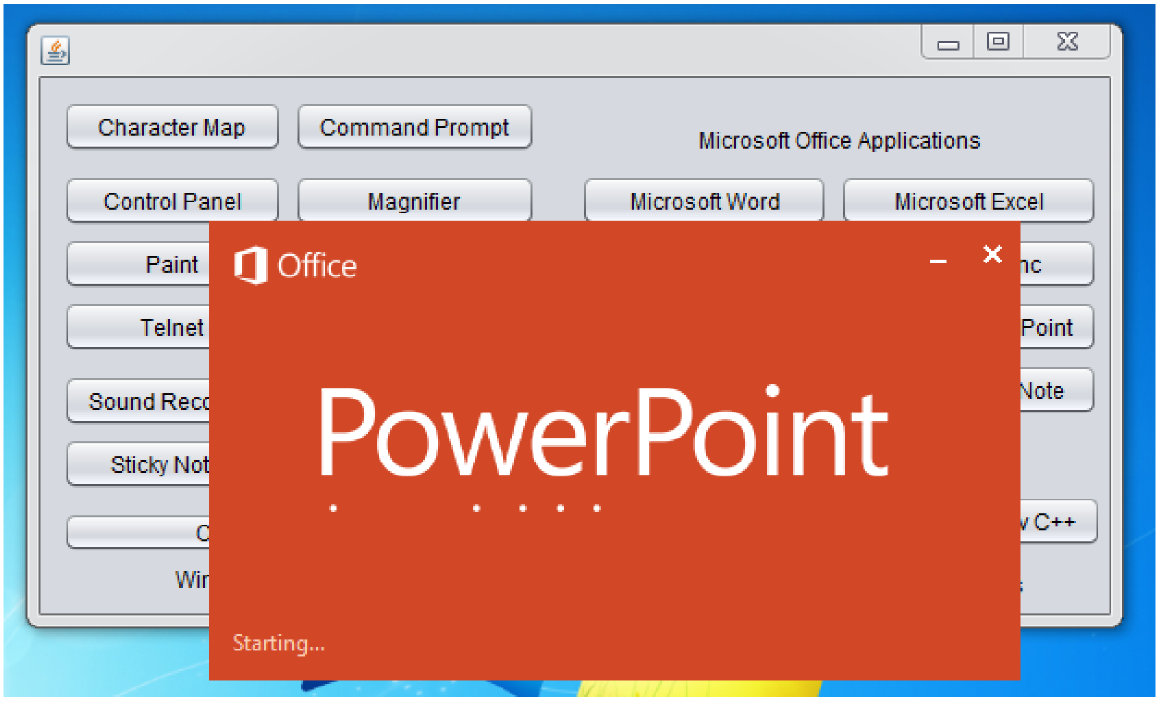
3.5.4 Demonstration of the Application

After laying the basic discussion of the application, I will demonstrate the steps involved in opening an application using the application launcher application, as it is going to take a total of two steps namely starting the application launcher application in the machine and then just click on the application you want to launch.

i) Start the application launcher application.



ii) Click any application you want to launch, say we clicked on the Power point application.



3.6 Webcam Controller

3.6.1 Introduction

This application enables user to control the webcam, in an easy way without installing any add – on application for using the webcam to take a picture, or to record a video using the webcam in the machine. For this application to run smoothly we require a webcam in the user machine, if there is no camera in the machine the application will generate error message saying, webcam not present or please check whether the webcam is configured properly or not.

3.6.2 Designing of the Application

The designing of the application involves using the Java media framework Application Programming Interface version 2.2.1 which enable the developer to access the webcam and various other media features of the machine using the classes and implementation methods in our application. The Java media framework has many classes and implementation, but in my application I will be using a specific set of classes and there methods to make the application and to run it in the user machine in a successful manner. The webcam can be launched by the voice recognition part i.e. the visual support of the program by using the keyword “Open Webcam” in the microphone of the system.

3.7 Voice Recognition

3.7.1 Voice Recognition Modelling

Speech recognition will play a very imperative role in future as a medium for an interactive human-computer conversation. As known to everyone this field has not been expertise till now and is still under the research so that computer can hear and understand what the computer says. The existing state of the art schemes have lower accuracy in the presence of ambient noise. This has been a problematic situation for the users as accuracy level of audio- visual recognition is expected to be much higher than the existing systems, especially this kind of facility faces huge problems in public places such as the railway station, airports, bus stands, and other outdoor places of similar context.

The reason for relying on the audio-visual aspects simultaneously comes from the human itself, as they can separate the audio signals produced by other human from the ambient noise, if any, present in the surrounding. The result is explained by the "Mc Gurk Effect" which states that if we give different signal of both audio and visual to the human, even the human will confuse that what the other human said, hence a failure of understanding was observed by humans so how the computer will be able to explain it. Mc Gurk also concluded that the user cannot force himself to hear correct vocal sounds when fed with wrong visual information.

Using the Bi – Modal speech recognition system, i.e., both audio and visual models for speech perception and production of speech is one of the possible approach to solve the speech recognition problem along with the solution of the existing models. Scientifically, it has already been proven that the visual signals are more significant as compare to audio signals when it comes to the speech recognition, there exists a real life example of hearing impaired people who relies only on the visual signals and models for the recognition and perception of speech.

The primary advantage of using visual signals is that the model based on video signal is less effected by the ambient noise and it also don't face the problem of cross – talk as faced in case of audio based signals. But the visual signals unaided are not ample as several of the verses produced are similar in case of video like 'bah' and 'gah'; along with the phonemes like ‘/b/’ and ‘/p/’ of English semantic, so we require the audio signals. Moreover audio signals alone faces the problem of cross talk and faces a substantial diminution in manifestation of ambient noise, even in the complete absence of the noise it faces complications due to similar acoustic signals produced by the phonemes like /b/ and /k/ which can be distinguished easily using the visual signal.

The above can be summarized like this, that we require both acoustic and visual signals simultaneously to recognize the speech with higher accuracy and precision. The effect of ambient noise will also be condensed using the Bi – Modal mechanism as discussed above. Furthermore, the studies spectates that the visual signal model can with stand a depreciation of an extra 4dB of noise in acoustic signals when used alone or used simultaneously.

One of the major difficulty in today’s time is to recognize what feature is the most suitable for the visual gesture recognition. There are various options highlighted by various technologies existing today. The first audio visual system was developed taking in consideration the mouth area, perimeter, and height along with the derivative width from the angles of binary image produced. The other alternatives include grayscale image parameters were considered for the visual signal analysis some of them took the intensity of each pixel in an image sequence and it was considered as a function of time and one-dimensional wavelet and Fourier transformations were applied to the generated intensity versus time function for modelling the lip gesture recognition. The researchers have been using the depth stream analysis of the acoustic and visual signals to recognize the speech synthesized but here we took a new hybrid approach. From the study above we came to a conclusion that most of the research and study is based upon the concept of image and geometric features for visual signal modelling which is very expensive to implement as the algorithm associated with such an analysis is very complex and we require state of the art machinery to run such an algorithm. So a better alternative will be to opt a method lesser complex and can be performed by a standard computer so that it can be used by individuals at a regular basis.

a) Acoustic Audio Signal Modelling

The acoustic audio signal is modeled [7] as the hidden Markov modelling of speech signals and a comparative study is carried out with HTK models. It also uses 13 dimensional Mel Frequency Cepstral Coefficients (MFCC) as the standard for audio features. Whereas [5], [6] believes in the use of first as well as the second derivative of the Mel frequency cepstral coefficients with respect to the raw speech signal. The author [8] also brings in the concept of SVM, and have used both HMM and SVM model and calculated the F-Score of both the models and proved that there was an increase of 9% in the accuracy of the model as compare to the existing state of the art models. Along with the HMM modelling [7] author also uses the concept of neural networks and fuzzy logic to detect the speech and to produce the high precision results.

It also suggested generation of stochastic models or random variables to take an average and generate results based on the models in a probabilistic manner to generate more precise results, as human audio waves are continuous waves therefore, it is difficult to implement the code for the entire continuous wave and hence the role of probabilistic approach comes into play.

The studies indicates that the Baum – Welch Algorithm is the enhanced route to implement the preparation procedure and must be castoff for teaching the signal models by taking into account the real-world implementations.

b) Visual Signal Modelling

The most important part of the speech recognition is the recognition and modelling of visual part more properly, as the visual signals are the signals which remain unaffected even in the presence of cross – talks and other ambient noise, but the question is which part of the face will give us a more precise output. There had been a lot of research going on this agenda, some [7] recommend using lower lip gesture along with the first order derivative of the modelling and take only 6 viewpoints on the human face, [8] recommends to use the whole face and model it using the HMM and SVM modelling using 66 viewpoints which will cover the entire human face and form a fully connected graph of the face to compute the visual based speech and gesture generated by the user, the research done also shows that calculating both HMM and SVM of the visual signal results in an overall increase in the accuracy of the model by 9% as compare to the existing state of the art models. The author have used the concept of F – Score to model the visual signals in the system. Visual recognition have also been done based on the depth information [6], here the concept of laser was used to find the depth of the human mouth along with the lip gesture and the visual is divided into two parts namely face and lip recognition as the ratio of face of every individual remain the same. The concept of region of interest is also used in the model, where the region of interest being the lip and the depth. There is also a possibility that we can use the concept of [9] Space time interest points for the mouth motion analysis, as it makes use of the ratio of jibber-jabber of diverse accents along with image and video processing tools to generate the keywords spoken by the user, as every word in every accent has different mouth opening, so it uses the ratio of the upper lips, lower lips and the difference between the lips to recognize the keywords spoken by the user. It also share the concept of region as [6] but more precise by enlightening the point that the teeth are useless in the study of ratios, so the teeth are removed from the picture using the image processing features and then the final image is generated. The final image is used in construction of a spatial - temporal scale-space representation of the image by applying a convolution with spatial temporal kernels, after the convolution at each point in space-time domain we compute a spatial - temporal second moment matrix which is then used to generate the alphabet.

3.7.2 Application Design

The application is designed using the MATLAB software which is popular among the various calculations and performing the various events in most of the field of technology known to mankind and scientists.

Code:

```
i1=wavread('Open Calculator.wav');
i2=wavread('Open Email Box.wav');
i3=wavread('Open Google Search.wav');
i4=wavread('Open Magnifier.wav');
```

The above code is used to read the various files located in the database of the .wav files created by taking samples of 10 different users 5 male and 5 females speaking in various accents and making a different set of outputs at a large.

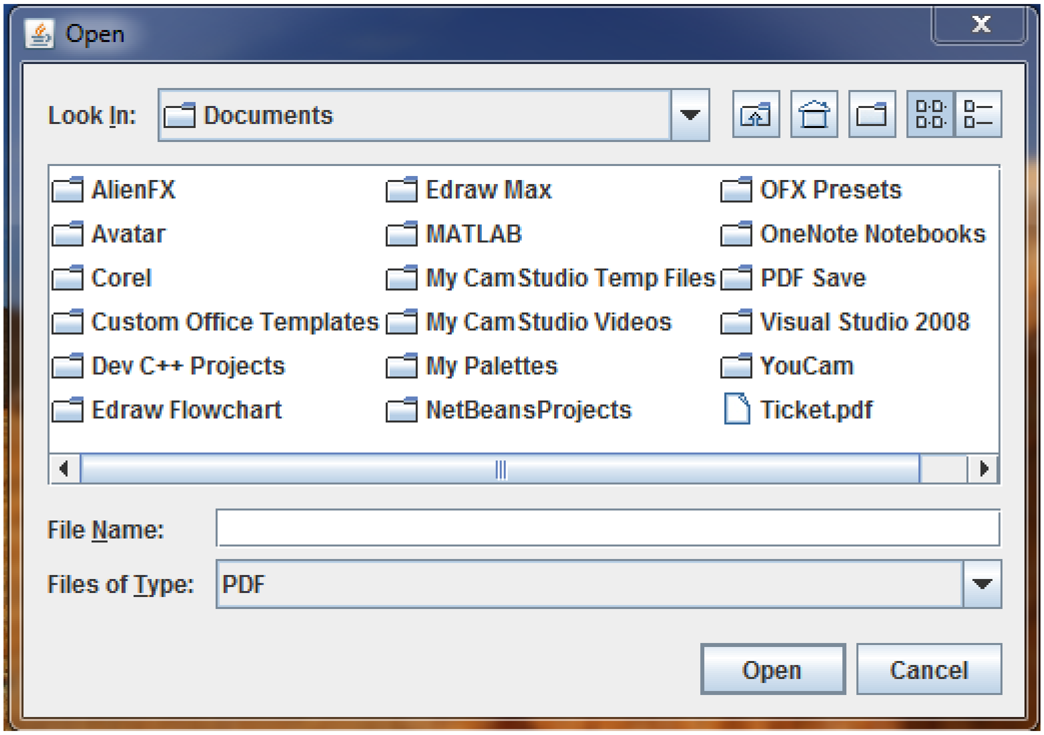
```
recObj = audiorecorder
disp('Start speaking.')
recordblocking(recObj, 5);
disp('End of Recording.');
```

The above code will take the input from the user from the computer's microphone and will process it, for processing I have used the correlation property of MATLAB in which two signals with majority of correlation will be considered as the command spoken by the user which is theoretically true, as the two files having the maximum correlation will be the most similar files in the database and in the input provided by the user.

After taking the correlation the index with maximum value is used as the final output and the corresponding output is generated.

3.8 Story Teller

The story teller application is used to read stories to the toddler, it can be used by the parents who are specially abled to read story to their children before bedtime which will help in making the children sleep as well calm their nerves.



Look In: Documents

AlienFX
Avatar
Corel
Custom Office Templates
Dev C++ Projects
Edraw Flowchart
Edraw Max
MATLAB
My CamStudio Temp Files
My CamStudio Videos
My Palettes
NetBeansProjects
OFX Presets
OneNote Notebooks
PDF Save
Visual Studio 2008
YouCam
Ticket.pdf

File Name:

Files of Type: PDF

This is the interface of the story teller application of my project, here the user have to select the pdf file he wants to be read for his child and click open, the application will start reading the file, however, the interface is not of much help as the application is disabled for the pause, stop or some additional features.

Chapter 4: Conclusion and Future Work

The application building part of the overall application has come to an end, where I have pointed out the major applications which is desired by the people along with the visual and audio support.

The final VPA application is an application where user will select what kind of support is desired and on the basis of the decision made by the user the program will give him/her the option either to take audio support, which is preferred for people having visual disability so that he/she can use the voice recognition based application; or the visual support, preferred for the people having audio based disabilities so that the normal one click interface can be used.

The future work of the application will be to make the application a speech based personal assistant application, which perform every user toils as they speak by increasing the database and giving more commands that are available at the user end.

References

[1] E. D. Petajan, “Automatic lipreading to enhance speech recognition,” in Proc. IEEE Global Telecommunication Conf. (GLOBECOM), vol. 1, Atlanta, GA, Nov. 26–29, 1984, pp. 265–272.

[2] E. D. Petajan, N. M. Brooke, B. J. Bischoff, and D. A. Bodoff, “An improved automatic lip reading system to enhance speech recognition, “in Proc. CHI’88, New York, 1988, pp. 19–25.

[3] S. Basu, C. Neti, N. Rajput, A. Senior, L. Subramaniam, and A. Verma, “Audio-visual large vocabulary continuous speech recognition in the broadcast domain,” in Proc. IEEE 3rd Workshop Multimedia Signal Process., Copenhagen, Denmark, Sept. 13–15, 1999, pp. 475–81.

[4] K. Yu, X. Jiang, and H. Bunke, “Lip - reading using signal analysis over time,” Signal Process. vol. 77, no. 2, pp. 195–208, 1999.

[5] Georgios Galatas , Gerasimos Potamianos , Fillia Makedon, Audio-Visual Speech Recognition using Depth Information from the Kinect in Noisy Video Conditions. Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments, June 06-08, 2012, Heraklion, Crete, Greece

[6] Georgios Galatas , Gerasimos Potamianos , Fillia Makedon, Audio-visual speech recognition using depth information from the Kinect in noisy video conditions, Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments, June 06-08, 2012, Heraklion, Crete, Greece

[7] Z. Zeng, J. Tu, B. Pianfetti, M. Liu, T. Zhang, Z. Zhang, T. S. Huang, and S. Levinson. Audio-visual affect recognition through multi-stream fused HMM for HCI. In CVPR, pages 967–972, 2005.

Useful Links

i) http://docs.oracle.com/cd/E17802_01/products/products/java-media/speech/forDevelopers/jsapi-doc/

ii) http://cmusphinx.sourceforge.net/wiki/tutorialsphinx4

iii) http://en.wikipedia.org/wiki/Shunting-yard_algorithm

iv) http://www.oracle.com/technetwork/java/javamail/index.html

v) http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm

vi) http://www.oracle.com/technetwork/java/index-140170.html

vii) http://www.oracle.com/technetwork/java/javase/documentation-138769.html