EASY COMPUTER ACCESS FOR THE DISABLED

PROJECT REPORT

Submitted by

14027075: Jishnu Jeevan

To the Mahatma Gandhi University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ALBERTIAN INSTITUTE OF SCIENCE AND TECHNOLOGY

KOCHI - 682 022

2017-2018

ALBERTIAN INSTITUTE OF SCIENCE AND TECHNOLOGY
KOCHI-682 022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **Easy Computer Access For The Disabled** is submitted by **Jishnu Jeevan – 14027075** in partial fulfillment of the requirements for the award of the Degree of *Bachelor of Technology in Computer Science and Engineering* to the Mahatma Gandhi University as a record of bonafide work done by him under our supervision and guidance.

Project Guide	Project Coordinator	Head of Department
Mr. Jeswin Roy Dcouth	**Mr. Jeswin Roy Dcouth**	**Ms. Jessy George**
Assistant Professor,	Assistant Professor,	Assistant Professor,
Dept. of CSE, AISAT	Dept. of CSE, AISAT	Dept. of CSE, AISAT

Acknowledgment

I express our deep sense of gratitude and indebtedness to our institution Albertian Institute of Science and Technology, Kalamassery which provided us the opportunity to fulfill our goals. We extend our sincere thanks and regards to Prof. Dr. Philip Kurian, Principal, AISAT for giving us this opportunity to carry out this work in the college. We sincerely acknowledge the encouragement and support given to us by our guide and our project coordinator Mr. Jeswin Roy Dcouth., Assistant professor, Dept. of CSE, AISAT in completion of our project. We would thank the god almighty, our family members, friends and faculty’s of the CSE department without whom this project would be impossible.

Abstract

Easy Computer Access for the Disabled (ECAD) is software that can be used to help paralyzed people for easily manipulating their computer system. People who are paralyzed will be having difficulty to write text using the keyboard as they will have difficulty going over to the entire keys in the keyboard. Our software can help them to use their computer system by providing keyboard and mouse simulation that is designed for people with paralysis. The mouse need not be dragged to the required key or to the required function. They can use the left and right mouse button to type the text. The right mouse button is used to scan through the list of keys and functions while the left mouse button is used to select the scanned key or function. A database is used to store a set of words that the user can choose from it. Here suggestions are provided for the most frequently used words, so that they don't have to write the entire word. They can just select the required word from the database. New words written by the user are added into the database. The user interface is a 6X7 matrix which contains the following buttons list, letter from a–z, number, space, backspace, open/close talk window, speak, mouse actions, shortcuts, navigations, other buttons, symbol set 1, symbol set 2, symbol set 3, open application, switch tabs, open/close settings.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DB – Database

TTS – Text to speech convertor

IDE – Integrated development environment

JDBC – Java database connectivity

API – Application programming interface

Chapter 1

Introduction

It is stand alone software that can be made to run in any laptop. This software is to help paralyzed people to convey message to others. Text to speech converter (TTS) is used, which is a technology that convert text to speech. Here the users can use the mouse to type text. Right mouse button is used to scan the alphabets and left mouse button is used to select the required letter. This is much less fatiguing that typing texts using the entire keyboard. Patients with movements in their fingers but not in their shoulders and wrist can also use this software to type texts. A DB is included to show suggestion of frequently used words, so that users don't have to write the entire words again and again. They can just select it from the DB which is much faster that typing the words again and again. If a new word is written by the user then it will be added to the DB. The predefined set of words is shown as a list based on the frequency of their occurrence. If a word has a large frequency of occurrence then it will show at the top of the list. The list shows words based on the decreasing order of frequency. The software can also be used to simulate mouse and keyboard operations so that they can control the computer using this software. This software facilitates the provision of mouse simulation using left and right buttons, single mouse button, and using eye blink. In case of step access scanning, the right mouse button is used to scan through the list of keys and functions while the left mouse button is used to select the scanned key or function. In switch access scanning, highlighting happens in a time slice where as selection of keys is done using left mouse button. In eye blink scanning, selection of keys are done using eye blink detection

Chapter 2

System Study

2.1 Existing Systems

There are text to speech converters available in our computer and there are on screen keyboards available also.

Disadvantages:

But in those systems we need to make use of whole keyboard for typing text. The mouse needs to be places near each letter in order to be clicked which can be fatiguing for people with paralysis. Text will remain there even after speak operation is performed. Database is not used to store words in the TTS convertor. Two application, the on screen keyboard and the text to speech convertor, needs to opened at the same time to convert text to speech.

2.2 Proposed System

This software uses the mouse or the eye blink for typing. So that it can be used by paralyzed people who are not able to move their whole hands for typing but has some minute muscular movement.

Advantages:

The letters can be typed using either the buttons of the mouse or using their eye blinks. Three types of scanning methods are used. The first one is step scanning. In this method the highlighting is done using the right mouse button while the selection is done using the left mouse button. In switch access scanning the highlighting occurs after a few seconds which can be adjusted by the user. When the row is highlighted for a particular amount of time the user needs to click the mouse button to select the row or the column being highlighted. If the user does not click the mouse button in the specified time then the time limit expires and the next row or column gets highlighted. In eye blink we have implemented switch access scanning using eye blink in which is same as switch access scanning using mouse but it uses eye blink for selection.

Chapter 3

Literature Survey

3.1. Adaptable Virtual Keyboard and Mouse for People with Special Needs [1]

Assistive technologies are helping people without speech or functional writing to communicate through alternative methods and devices. The scientific area of assistive technology which specifically collaborates to the expansion of communication skills is named Alternative and Augmentative Communication (AAC). AAC helps persons without speech or writing ability to communicate with others. The **application developed constitutes a keyboard and mouse emulator, configurable and adaptable.The** keyboard and a mouse's emulator software associated to low-cost sensors, adequately to assist a large portion of the population with some kind of physical disability that have impaired their communication abilities. Originally, an electronic circuit was connected to the mouse and keyboard inputs of the computer. To generate a more affordable solution, the hardware was replaced by sensors (and associated software) directly connected to computer device inputs such as USB (Universal Serial Board) and the Serial Port. The program consists internally of a layout’s manager, responsible for the visual part of the virtual keyboards, a script processor for processing chained commands, one device manager to acquire signals from the sensors and the command processor that is the core part, responsible for emulation of keyboard and mouse. Emulation means to declare to the operating system to type characters as if received from the keyboard and to click or move the cursor as if received from the mouse.

It is very easy to use. It has high text entry speed and reduce the risk of keystroke logging i.e. noise is less than conventional keyboard.

Costly due to the extra hardware. Here in our system we only intent to use the mouse and a laptop. Multiple shortcuts and mouse action cannot be executed. Another application cannot be opened from this software. Very slow opening, closing and switching of different application.

3.2. On Eye Gaze System to Operate Virtual Keyboard [2]

For many people who have disability in communication, giving some information to other people is a hard thing to do. To make the communication easier, some people who is not able to speak or has disability in motoric function especially hand, need a system that can help them to communicate. Based on that condition, this paper built a system that can help them to communicate. The system uses eye gaze to write some information with virtual keyboard.

The process includes capturing user's face by a camera , then the camera do some feature detection of face to get position of eye gaze, and this position will become a reference of pointer position to choose some word in virtual keyboard.

HaarCascade method is used to detect some feature of face, and Integral Projection method is used to get position of the eye movement.

Highlighting features:

- The eye gaze system helps the differently abled to communicate
- Hands free keyboard
- Here feature detection is done efficiently and quickly with minimized errors
- User can give information quite clearly enough.

Limitations:

- Based on the results the comparison ratio between duration of normal writing and typing using system for two words is 1:13 second. It takes one minute to type one word
- Minimized user movements

Our proposed system aims to resolve the above limitations by improving the eye gaze tracking more efficiently and precisely. The proposed system primarily focuses on quick eye blink detection, which gradually increases the typing speed of the virtual keyboard. And our system aims to provide a better simulation of keyboard using eye gaze. This system can only type letters but cannot control the mouse. Our system will have the ability to control the mouse. This system doesn’t have the ability to convert text to speech and doesn’t have database to store words. Also cannot control the computer and other applications.

3.3. Assistive Context-Aware Toolkit (ACAT) [3]

Assistive Context-Aware Toolkit (ACAT) is an open source platform developed to help people with limited mobility to communicate and interact with their computer. It is a useful tool for a wide range of disabilities. It enables users to communicate using keyboard simulation and text to speech capability. It also enables users to perform common tasks such as editing, managing documents, navigating the internet and accessing emails. It enables the creation of customized interactive solutions that leverage the movement’s people with disabilities might be limited to such as touch, eye blinks and eyebrow movements.

The ACAT User Interface is designed to be used without the need for a keyboard or a mouse. The User Interface is made up of ‘scanners’. The User Interface elements are iteratively highlighted one after another allowing the user to make a selection. The user selects an element by activating a switch when the element is highlighted. ACAT will then execute the action associated with the element. Because user needs vary greatly there are many ways a user can activate a switch and ACAT supports a range of input mechanisms.

ACAT supports keyboard and ACAT vision as input switch mechanisms. The software provides facility to Converse with others, Edit text, Interact with applications like email, Browse the web, Manage files and Deliver lectures.

Highlighting features:

The software serves different applications like vision tryout, ACAT tryout.

Vision Tryout is a standalone utility which uses a webcam to detect facial gestures .ACAT Tryout familiarizes the functioning of scanners. The Talk application works for both QWERTY keyboard and ABC keyboard. It allows customization of the software according to the end-user needs. It provides various hot key utilities for easier access. The scanning speed of the letters is adjustable. The software employs efficient text entry.

Limitations:

ACAT is software designed specifically for Stephen hawking. So it has software like lecture manager which allows him to give lectures. Since we are designing the software for a common man this feature is not needed. The mouse scanning occurs in three steps, we plan to overcome the time complexity by doing it in two steps. It only has switch access scanning but does not have step scanning. Step scanning is useful for some patients as it has less cognitive strain.

3.4. Low cost eye based human computer interface system [4]

This paper presents a low cost eye based Human Computer Interface (HCI) developed for paralyzed persons. The system is a computer interface that provides the functionality of an input device like mouse based on eye actions such as eye blink, eye-gaze and gaze control. It consists of a very simple automatic initialization phase during which the eyes of the user are precisely detected. The detected eye pair is tracked during the movement of the user and the system is automatically re-initialized if the tracker loses the eye pair. The creation of the cursor movement includes the initializing process of the current user's iris. Iris detection is done in the following ways; firstly, after acquiring the image from the webcam, face detection of the user is carried out. After that, the forehead is detected using the canny edges' detection theorem followed by eye detection using cross shaped kernel as the algorithm. The morphological filters are applied to erase the jitters and speckles around the object of interest. Thus, the eye is being detected.

The clicking option of the mouse is being implemented by gaze determination of the iris over the icon of interest. In the initial phase of the proposed method, an approach to the clicking action of the mouse by winking action was selected, where as the wink of the left eye constitutes the left click and the right wink constituting the right click. Another concept used here is that when the x co-ordinate specifications (eye movement to the right side) is more 35 units, then it considered to be a right click, if it is beyond -45 units (eye movement to the left side), then it is considered to be a left click. The evolution of clicking in the present work started with winking system and ended with gaze being replaced for wink. Single blink is a blink when the user closes and opens the eyes once. Double blinks are two rapid successive single blinks. The blink detection begins with motion analysis. As an alternative method, determining the cursor movement by gaze determination technique has been chosen. The threshold time for holding the gaze on the particular icon is set. The time could be set from a variation of one second to five seconds, depending on the usability factor of the subject.

The proposed method uses implementing the movement of the Iris, instead of using mouse. The HCI proposed in this paper is a custom developed low cost and efficient system to achieve fast and accurate detection of eyes, eye blinks and tracking of eyes and iris with unconstrained background.

Cursor movement of up and down was feasible, but right-left movement got confused with right-left clicking action. Sync between clicking options and cursor movement was lost. For disabled people it will become difficult to control the entire system just by using the movement of mouse using eye. They need to operate the virtual keyboard application for text typing and may need to open the text to speech converter to convey message. So our system can overcome this because we will integrate mouse, keyboard, and text to speech applications in one.

Chapter 4

Software Requirements Specification

4.1. Introduction

4.1.1. Purpose

The purpose of this document is to provide the software requirement specification report for the software easy computer access for the disabled. This software can be used by people suffering from paralysis for accessing and controlling their personal computer easily. This software can be used for augmentative and alternative communication (AAC), which means they can use this software to talk with people if they are not able to speak. This is a single system. By using this software user can communicate freely with anyone and also use their computer with ease.

4.1.2. Document Conventions

DB – Database
TTS - Text to speech
AAC - Augmentative and alternative communication
ALS - Amyotrophic Lateral Sclerosis

4.1.3. Intended Audience and Reading Suggestions

This project is a college level project and is implementing under the guidance of our college professors. This project will be useful to those people who are paralyzed and cannot speak but have some minute muscular movement. Paralysis can be caused due to various reasons like Amyotrophic Lateral Sclerosis (ALS), stroke, Parkinson's etc.

4.1.4. Product Scope

The purpose of this software is to help paralyzed people. By using this software they can convey messages to others without any translators and they can also use their computer systems with ease. They can control this software by using any of the two scanning methods. The scanning methods are step scanning (that requires the use of two mouse buttons) and switch access scanning (that only requires a single mouse button). If the patients are having severe paralysis such that they are not able to use the mouse buttons, then we have provided a way for them to control this software using their eye blinks.

4.1.5. References

1. Switch Access Scanning - https://en.wikipedia.org/wiki/Switch_access_scanning
2. Adaptable virtual keyboard and mouse for people with special needs - http://ieeexplore.ieee.org/document/7821782/
3. Assistive context aware toolkit (ACAT) - https://01.org/acat

4.2. Overall Description

4.2.1. Product Perspective

Paralyzed people communicate through various methods like letters on a spelling card, but in those methods user requires more help from a nurse or a caretaker. We implement our project same as letters on a spelling card. The spell card or the layout will not only have the letters from a-z but will have every keys on the keyboard and also buttons that perform special functions like moving the mouse pointer from one location to another. Using these buttons they will be able to control the computer with ease and perform any task they require with their computer.

4.2.2. Product Functions

ECAD can be controlled in three ways i.e. by using a single mouse button, using both the mouse button or by using their eye blinks. It also has a TTS converter that will convert text to speech so that the user can communicate with others. The software can simulate the keyboard and mouse using which they can control the computer to perform the tasks that they require like editing documents, web browsing etc.

4.2.3. User Classes and Characteristics

This software can be used by people who are having paralysis and cannot type text using keyboard. This method can be quite useful, as the equipment are readily available whereas the training needed for them are easy. This method becomes a last resort for many patients.

4.2.4. Operating Environment

ECAD can be used in computer system that supports Java. The computer that the user wishes to control/use must have a mouse and a webcam in order to control the software. ECAD is completely written in Java using the Eclipse IDE and we have used OpenCV with haarcascade classifier for eye blink detection. We have used SQLite 3 as the database and MaryTTS for the TTS converter.

4.2.5. Design and Implementation Constraints

A few implementation constrains that can occur with ECAD are

1. The scanning time for switch access may vary from computer to computer even though the scanning time provided is the same. In a system with Intel core i3 processor and another system with Intel core i7 processer the same scanning time may occur faster in one and slower in another.
2. ECAD will only work properly for systems with keyboard of the QWERTY standard.

4.2.6. User Documentation

A few things that the user need to know about ECAD are

1. Step scanning - Highlighting the rows and columns of the user interface using right mouse button and selecting the required row or column using the left mouse button.
2. Switch access scanning - Only a single button is needed to select the required row or column. The highlighting of the row or column is time based i.e. the row or the column will be highlighted for a particular amount of time and the user will have to click the mouse button within that specified time. The time can be manually set by the user according to their comfort.
3. The same principle of switch access scanning can be done using their eye blinks.

4.3. External Interface Requirements

4.3.1. User Interfaces

Here when the software is opened the spell card is displayed.

For step scanning when the user presses the right mouse button the first row will get highlighted. The right mouse button is used for highlighting of the row and column. If the user wants to select a highlighted letter they will have to press the left mouse button. The left mouse button is used for selection of row or column.

For switch access scanning the highlighting is done for a particular interval of time and the user must select the required row or column when it is highlighted. After the time interval has expired the next row or column will be highlighted.

They can also control the software using their eye blinks if they are not able to use the mouse. The same principle of step scanning or switch access scanning is used but here it is done with the eyes.

4.3.2. Hardware Interfaces

The only hardware that is being used in this software is a laptop and the mouse. No other external hardware devices have to be used. The laptop should contain a web cam for eye blink detection.

4.3.3. Software Interfaces

The software’s that will be used along with our project are:

A TTS convertor that will take the text written by the patient in our software and pass it as input to the TTS convertor and the TTS convertor will speak out the written text. The TTS convertor used here is the java MaryTTS.

Also we use a DB with four tables to store words, shortcuts, application, and system configuration. The DB used here is SQLite.

We have also used OpenCV with haarcascade classifier for eye blink detection.

4.3.4. System features

4.3.4.1. Description and Priority

1. Software will be able to take text from paralyzed patients.
2. TTS will take the text written by the patient and convert it to speech.
3. A DB is used.
4. Keyboard and mouse simulation is provided so that they can use the keyboard and mouse just like any other person would use the computer.

4.3.4.2. Stimulus/Response Sequences

1. The patient can start the software to using the right mouse button.
2. The right mouse button is used for highlighting each row or column of the spell card.
3. The left mouse button is used for selection of the highlighted row or column.
4. Eye blink is used as an interface to operate the software.

4.3.4.3. Functional Requirements

1. The TTS won’t accept any symbols or special characters.

4.3.5. Other Nonfunctional Requirements

4.3.5.1. Performance Requirements

Here the rate at which the patient can write the text onto the screen depends on how strong the patient’s fingers are to click the mouse button, how agile their finger is and also depends on the accuracy of eye blink detection.

4.3.5.2. Software Quality Attributes

This software can be used to operate the entire computer system. User can surf the internet. Database is used to add words into it for suggestion and also used to add shortcuts to easily simulate the keyboard.

Chapter 5

Project Design

5.1 Class Diagram

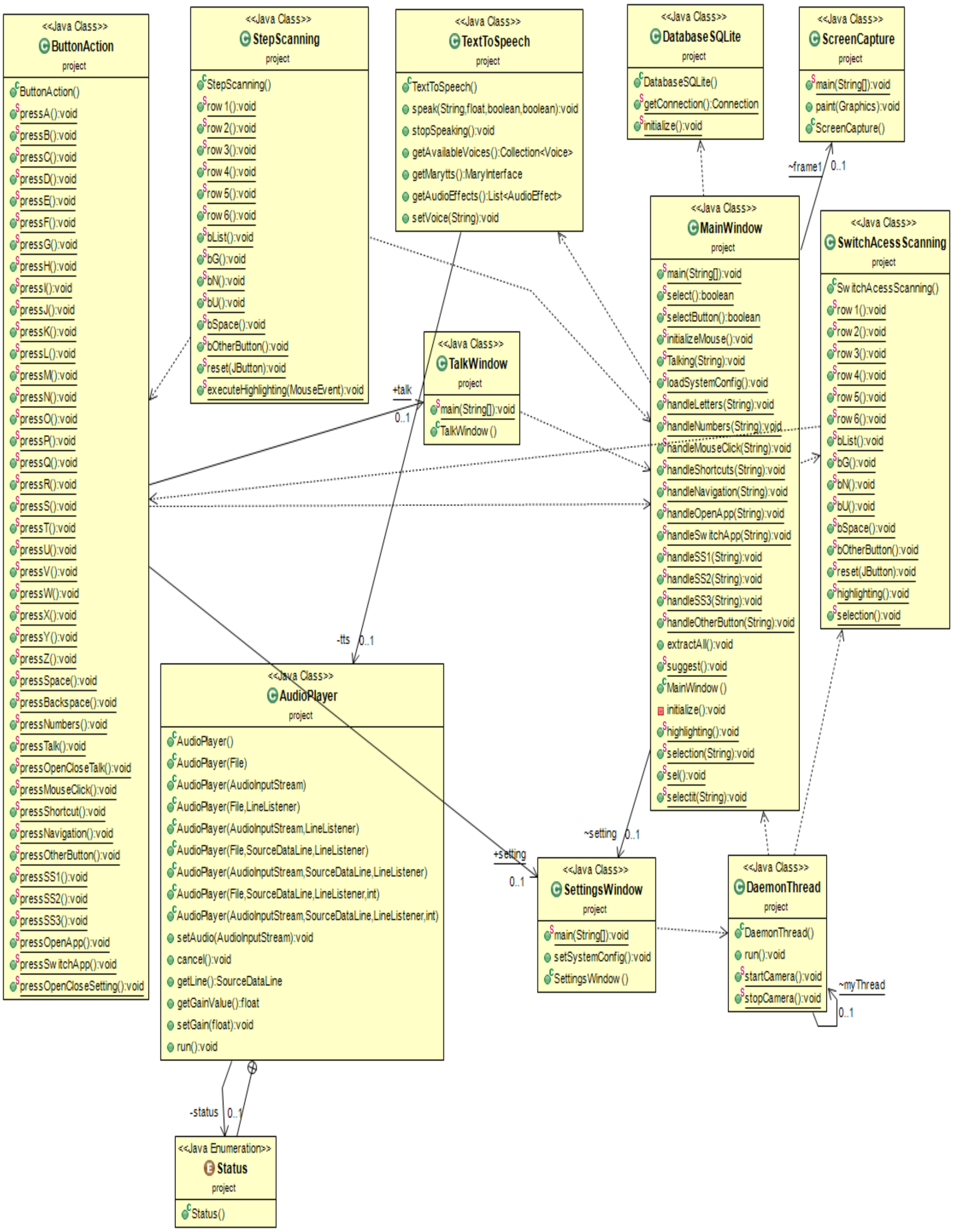


Fig 5.1: Class diagram

The classes for the project are as follows:

1. Main Window – This class was used to create the user interface of the project.
2. Step scanning class – Contains the code for highlighting and selecting of the buttons using the two mouse buttons. If step scanning is enabled this class is called and the code is executed.
3. Switch access scanning class – Contains the code for highlighting and selecting of the buttons using only a single mouse button.
4. Deamon Thread class – This class contains the code that is used for capturing the eye when eye blink detection is done.
5. DatabaseSqlite – This class is used for creation of the database and also for linking the application with the database if it already has been created.
6. Talk Window – This class has the code for a talk window that the user can type text to communicate with other.
7. Settings Window – This class was used to create the settings window in which the user can change the system setting according to their convenience.
8. Button Action – Contains the code that performs the action for each buttons.
9. Screen Capture – Used in mouse simulation
10. TextToSpeech, AudioPlayer – This is the Mary TTS converter that is created entirely in java. These two class are interdependent on each other.

5.2 State diagram

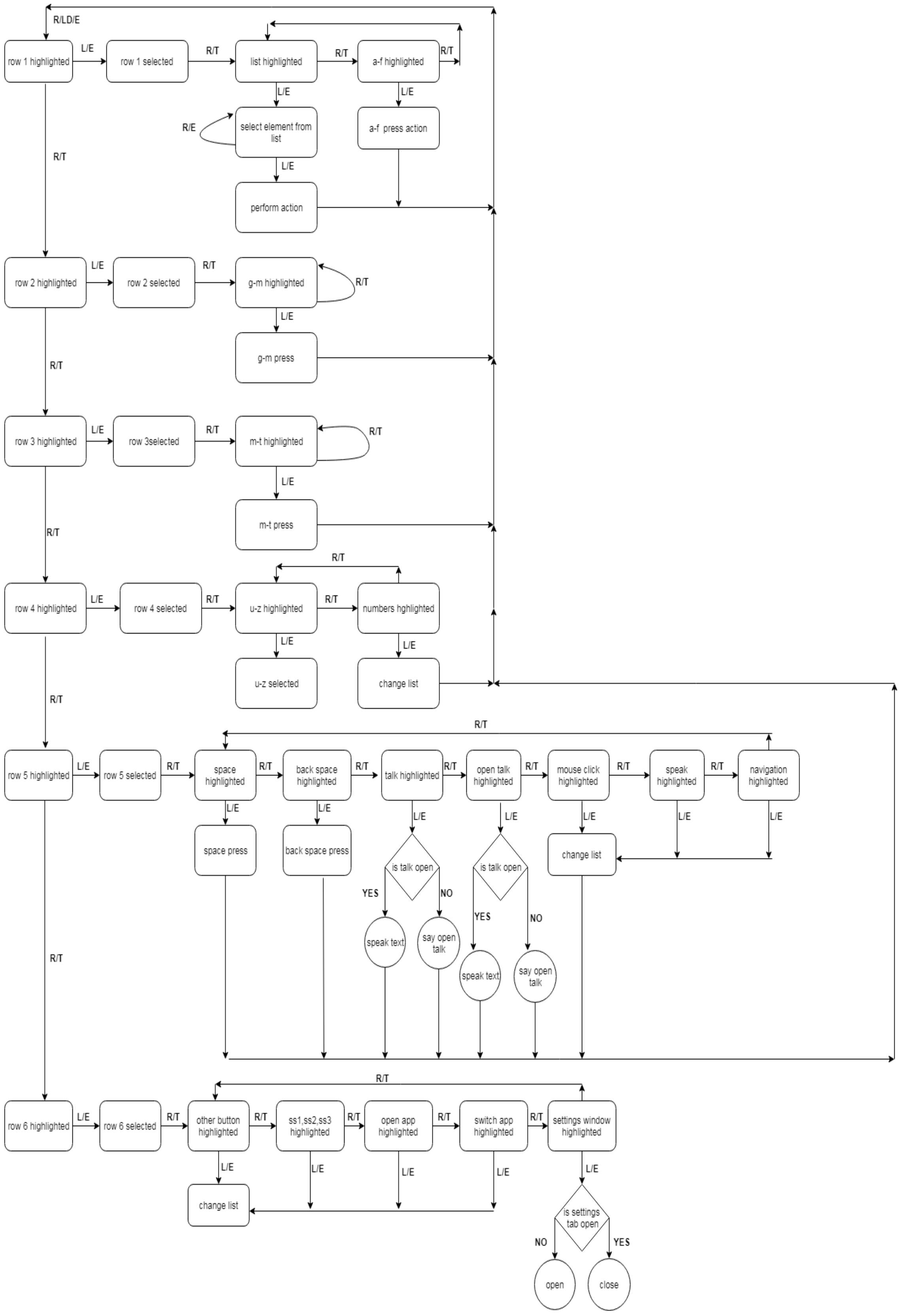


Fig 5.2: State diagram

R – Right mouse button

L – Left mouse button

LD – Left double click

E – Eye blink

T – Time slice expired

Here initially no rows or columns will be highlighted. When the user presses the right mouse button the first row will highlighted. If they press it again then the second row will be highlighted, again then the third row will be highlighted. Press it again and it will highlight the fourth row. If they press it again then the first row will be highlighted.

If the user wants to type A in a window, which is on row one and column two, the user will have to press the right mouse button two times so that the second row will be highlighted. Now the user must select this row when it is highlighted using the left mouse button. When the right mouse button is pressed the columns of the second row will be highlighted.

Now when the letter A is highlighted the user will have to press the left mouse button to select it. When it is selected it will perform its required action i.e. print A onto the text field. After the action of the button is performed then the system will go back to its initial state i.e. now row or column will be highlighted.

For switch access scanning the starting of the software happens when the left mouse button is pressed twice. For selection the user must press the left mouse button once.

For eye blink detection, works with the same principle of switch access scanning but using the eye blink.

Chapter 6

Detailed Design

Our project is divided into six modules which.

6.1. Module 1: Creation of layout and button action.

Here layout is created. The layout is a 6X7 matrix that consists of 26 letters in the alphabet , numbers, speak button, backspace, space, settings button, mouse click, shortcuts button, symbol set , navigation button and list which is a button that is used to select word from the list of saved words.

Alphabets : letters in the keyboard are pressed when these buttons are clicked.

Numbers : numbers in the keyboard are pressed when the number button is clicked.

Space : space button action is performed when this button is clicked

Backspace : deletion of characters occur when this button is performed

Speak : text to speech convertion occurs when this button is pressed.

Talk window : text area where all characters are converted to speech.

Mouse click : list area shows all mouse click actions that eases the mouse simulation.

Shortcuts : all shortcuts stored in database are displayed in the list area .

Navigation : all actions to navigate the web page is listed in the list area.

Other button : button provides all other button actions like enter key, caps lock etc.

Symbol set : list all special characters , arithmetic symbols etc.

Open app : list all frequently opened applications, files and folders.

Switch app : displays all files or folders that are currently opened.

Settings : a user interface to customize the ECAD software for easy simulation.

6.2. Module 2: Creation of database and TTS

6.2.1. Database:

The application will have the following databases

1. Open app database – Database that hold the applications that can be opened by the user easily

   Name: Application

Path	Name	Count

Table 6.1: Application

   Path – holds the path of the application

   Name – Holds the name of the application

   Count – Counts the number of times the application was opened.

2. Short cut database – database that hold the short cuts that are to be used

   Name: Shortcuts

Shortcuts	Name	Count

Table 6.2: Shortcuts

   Shortcuts – Holds the shortcuts key combination

   Name – Holds the name of the short cut

   Count – Number of times the shortcut has been used

3. List short cut – Database that hold the list of words used

   Name: Words

Words	Count

Table 6.3: Words

   Words – Holds the words typed by the user

   Count – Number of times the words was used

4. System Configuration – Holds the details about the system configurations like what type of scanning method is to be used and the scanning time

Name: System Configuration

Scanning method	Scanning Time	Mouse Scanning time	Voice Type

Table 6.4: System Configuration

All of these can be created using the concept of embedded database. This will initially hold them with default values. All these tables will be held in a single database

6.2.2. Text to speech

The software uses MaryTTS , a TTS converter written in java.

6.2.2.1.Providing function to talk

- The talk button will only work if the talk window is opened. If it is closed then a message will be displayed saying open talk window to talk.
- It will talk the message from the talk window and feed it to the text to speech engine.

So the overall code for talk will be

```
public static void Talking(String sentence)
{
        TextToSpeech tts=new TextToSpeech();
        //Enable Voice
        tts.getAvailableVoices().stream().forEach(voice ->
System.out.println("Voice: " + voice));
        tts.setVoice(voiceType);
        tts.speak(sentence, 1.5f, false, false);
  }
```

6.2.2.2.Providing function to open/close talk

- Initially the button will only say open talk
- When the button is pressed the talk window gets opened. And the button will say close talk
- When the button is pressed again the talk window gets close. And the button will be closed.

So the overall code is as follows

```
public static void pressOpenCloseTalk()
    {
    if(MainWindow.isTalkOpen==false)
    {
            talk.setVisible(true);   MainWindow.isTalkOpen=true;
    }
    else
    {
            talk.setVisible(false);
            talk.dispose();
            MainWindow.isTalkOpen=false;
    }
    }
```

6.3. Module 3: Creation of list and list action

The buttons that require only the list are Mouse click, Numbers, Navigation, Buttons, Symbol set 1, Symbol set 2, Symbol set 3 and switch app. The reason why they require only a list is that they their quantity is limited. Users are not adding anything new to the existing set.

6.3.1. Where these contents need to be stored.

```
JButton btnA = new JButton("SS1");

        btnA.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent arg0) {
                        list.setModel(new AbstractListModel() {
                                String[] values = new String[] {" ( ", " ) ", " [ ", " ] ", " { ", " } ", " < ", " > ", " \" ", " \' "};
                                public int getSize() {
                                        return values.length;
                                }
                                public Object getElementAt(int index) {
                                        return values[index];
                                }

                });
        }});
```

So this is how and where the contents of the each button are going to be stored. When the button is pressed the first thing it will do is change the model of the list. The list shown above is for symbol set 1. The advantage of this method is that we don’t need to create an extra table. The space and time complexity is reduced. In older method we have to extract contents from the database and then insert them into the list. So you can imagine it takes a longer time. But here the same thing is done in ten to twelve lines of code. And the table just takes up a lot of space in database.

6.3.1.1.How the action is to be performed when the value is selected

Considering the fact that we have only a single list but different set of values each time a button is pressed it is necessary that instead of selecting a value from the list and giving it to a common function (like in mini project) it is necessary to – select the value from the list – find out which category the value belongs to – and send it to the appropriate function to be handled.

So we should have functions for handling list values. So we need to have functions for each button that uses the list. So every time a button is pressed a list model is changed but also a flag variable need to be changed indicating that the list is now being used by this button. So this flag variable is used to indicate which function needs to be called when the list value is selected.

So this is how it needs to be implemented algorithmically

1. The button is pressed.
2. The list model is changed and the flag for the corresponding button is set to true.
3. The value form the list is selected.
4. We check which flag is set to true.
5. We call the function for which the flag is set to true.

Now there is one thing about the flag variable. Instead of created multiple Boolean variable for each button that uses the list and assigning one to true and the others to false every time a button is pressed it will be easier to assign each button that uses the list with a number. They are as follows

1. Letters and list (since they have a common content which are just words)

2. Open app
3. Mouse click
4. Shortcuts
5. Numbers
6. Navigations
7. Buttons
8. Symbol set 1
9. Symbol set 2
10. Symbol set 3

An example for the working is as follows.

Consider the example of list and letters

1. The button list or a letter is pressed.
2. The model of the list is changed (either by extracting the words from DB or in code for buttons that do not use the DB).
3. whichList= ”Letters”;   // since we are using the example of list and letters
4. A value is selected from list and stored in a variable ch

```
String selectedValue=(String) list.getSelectedValue();
if(whichList.equals("Letters"))
{
  try
  {
          handleLetters(selectedValue);
  } catch (AWTException e1) {          e1.printStackTrace();  }
  }
  if(whichList.equals("OpenApp"))
      handleOpenApp(selectedValue);
  if(whichList.equals("MouseClick"))
      handleMouseClick(selectedValue);
  if(whichList.equals("Shortcuts"))
  {
      try
          {
                  handleShortcuts(selectedValue);
          }
      catch (AWTException e1) {e1.printStackTrace();          }
  }
  if(whichList.equals("Numbers"))
  {
      try
```

```
    {
                handleNumbers(selectedValue);
        }
        catch (AWTException e) {    e.printStackTrace();    }
}
if(whichList.equals("Navigation"))
{
    try
        {

                handleNavigation(selectedValue);
        }
        catch (AWTException e) {    e.printStackTrace();    }
}
if(whichList.equals("OtherButton"))
{
    try
        {

                handleOtherButton(selectedValue);
        } catch (AWTException e) {  e.printStackTrace();    }
}
if(whichList.equals("SS1"))
{
        try
        {
                handleSS1(selectedValue);
        } catch (AWTException e) {  e.printStackTrace();    }
}
if(whichList.equals("SS2"))
{
        try
        {
                handleSS2(selectedValue);
        } catch (AWTException e) {  e.printStackTrace();    }
}
if(whichList.equals("SS3"))
{
    try
        {
                handleSS3(selectedValue);
        } catch (AWTException e) {  e.printStackTrace();    }
}

if(whichList.equals("SwitchApp"))
{
        try
        {
                handleSwitchApp(selectedValue);
        } catch (AWTException e) {  e.printStackTrace();    }
```

```
}
```

Now considering that this chapter deals with the buttons that use only the list and not the database we are going to describe the functions that use only the list

The ones that use only the list are Mouse click, Numbers, Navigation, Buttons, Symbol set 1, and Symbol set 2, Symbol set 3.

Description of function that handles mouse click

```
Void handleMouseClick(char ch)

{       If(ch==”Right”)

        Execute the code that performs right click

        Else if(ch==”Left”)

        Execute the code that performs left click

        And this continues for other functions

}
```

Description for function that handles number.

This same description can be applied to other functions like button, navigation, symbol set 1, 2 and 3 as all these are just key press and key release operation. The only complicated one is mouse click algorithm

```
public static void handleNumbers(String selectedValue) throws AWTException
{
                Robot r=new Robot();
                if(selectedValue.equals(" 0 "))
                {
                        r.keyPress(KeyEvent.VK_0);
                        r.keyRelease(KeyEvent.VK_0);
                }
                else if(selectedValue.equals(" 1 "))
                {
                        r.keyPress(KeyEvent.VK_1);
                        r.keyRelease(KeyEvent.VK_1);
```

```
}
else if(selectedValue.equals(" 2 "))
{
        r.keyPress(KeyEvent.VK_2);
        r.keyRelease(KeyEvent.VK_2);
}

else if(selectedValue.equals(" 3 "))
{
        r.keyPress(KeyEvent.VK_3);
        r.keyRelease(KeyEvent.VK_3);
}
else if(selectedValue.equals(" 4 "))
{
        r.keyPress(KeyEvent.VK_4);
        r.keyRelease(KeyEvent.VK_4);
}
else if(selectedValue.equals(" 5 "))
{
        r.keyPress(KeyEvent.VK_5);
        r.keyRelease(KeyEvent.VK_5);
}
else if(selectedValue.equals(" 6 "))
{
        r.keyPress(KeyEvent.VK_6);
        r.keyRelease(KeyEvent.VK_6);
}
else if(selectedValue.equals(" 7 "))
{
        r.keyPress(KeyEvent.VK_7);
        r.keyRelease(KeyEvent.VK_7);
}
else if(selectedValue.equals(" 8 "))
{
```

```
                    r.keyPress(KeyEvent.VK_8);
                    r.keyRelease(KeyEvent.VK_8);
                }
                else if(selectedValue.equals(" 9 "))
                {
                    r.keyPress(KeyEvent.VK_9);
                    r.keyRelease(KeyEvent.VK_9);
                }
 }
```

6.3.1.2.Providing action to switch app button

The switch app button uses a fixed size list. So no need for a database. The switch app button will be able to switch between 12 applications due to the fact that the list has 12 rows.

Same function as before

```
Void handleSwithchApp(char ch)
{ If(ch==”Alt + tab”)
  {
          KeyPress(alt)
          KeyPress(tab)
          KeyRelease(tab)
          KeyRelease(alt)
  }
  Else if(ch==”2”)
  {
          KeyPress(alt)
          KeyPress(tab)
          KeyRelease(tab)
```

```
            Delay(1000)

            KeyPress(tab)

            KeyRelease(tab)

            KeyRelease(alt)

    }

    And this continues for the rest of the alt tab also

    }
```

6.4. MODULE 4: Step Scanning

A scanning pattern refers to the way items in the selection set are presented to the user. It allows for easier item selection as the scanning is systematic and predictable. In linear scanning, items are usually arranged in a grid and the scanning indicator moves through each item in each row systematically.

In step scanning, the user controls each movement (or step) of the scanning indicator through its preset pattern by hitting a switch. To select an item, the AAC user hits a second switch once the indicator reaches his desired item. Because of the constant switch activation, this method might be too fatiguing for the certain scanners.

Methodology

```
1. Loop
        1.1.If right mouse button is pressed
                1.1.1.  If row 1 is not highlighted
                        1.1.1.1.Highlight first row
                        1.1.1.2.GOTO step 1
                1.1.2.  If row 1 is selected
                        1.1.2.1.If first column of row 1 is not highlighted
                                1.1.2.1.1.  Highlight first column
                                1.1.2.1.2.  GOTO step 1
        1.2.If left mouse button is pressed
                1.2.1.  If row 1 is highlighted
```

```
            1.2.1.1.Select row 1
            1.2.1.2.GOTO Step 1
      1.2.2.  If a column of row1 is highlighted
            1.2.2.1.Select that column of row 1.
            1.2.2.2.GOTO step 1
2.  End
```

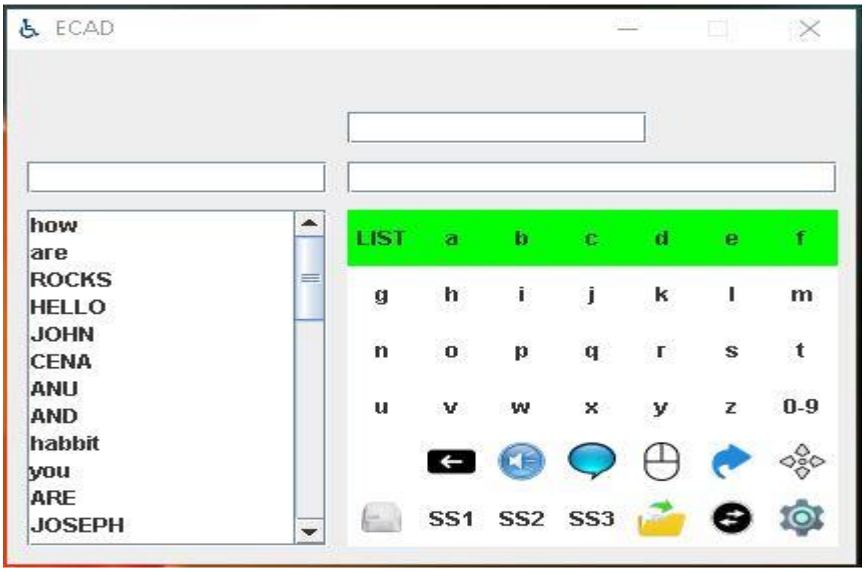


Fig 6.1: Row highlighting

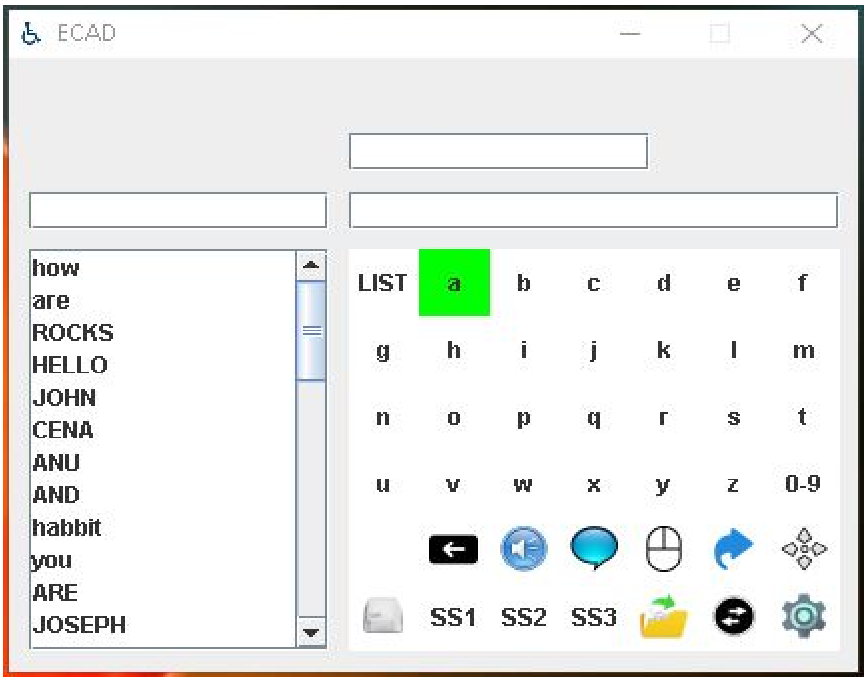


Fig 6.2: Column Highlighting

6.5. MODULE 5: Switch Scanning

Switch access scanning is an indirect selection technique, used by an assistive technology user, including those who use augmentative and alternative communication (AAC) to choose items from the selection set. Unlike direct selection (e.g., typing on a keyboard, touching a screen), a scanner can only make selections when the scanning indicator (or cursor) of the electronic device is on the desired choice. The scanning indicator moves through items by highlighting each item on the screen (i.e., visual scanning), and then the user activates a switch to select the item.

Methodology

```
1.  Loop
        1.1.If time limit has expired (left mouse button is not pressed within the time
            limit)
                1.1.1.  If row 1 is not highlighted
                            1.1.1.1.Highlight first row
                            1.1.1.2.GOTO step 1
                1.1.2.  If row 1 is selected
                            1.1.2.1.If first column of row 1 is not highlighted
                                        1.1.2.1.1.  Highlight first column
                                        1.1.2.1.2.  GOTO step 1
        1.2.If left mouse button is pressed (within the time period)
                1.2.1.  If row 1 is highlighted
                            1.2.1.1.Select row 1
                            1.2.1.2.GOTO Step 1
                1.2.2.  If a column of row1 is highlighted
                            1.2.2.1.Select that column of row 1.
                            1.2.2.2.GOTO step 1
2.  End
```

6.6. MODULE 6: EYE BLINK DETECTION

The eye blink detection is same as that of switch access scanning but instead of using a mouse button for selection we have used eye blink for selection.

We have implemented eye blink detection using OpenCV and harrcascade classifier which is used for detecting eye and eye blinks.

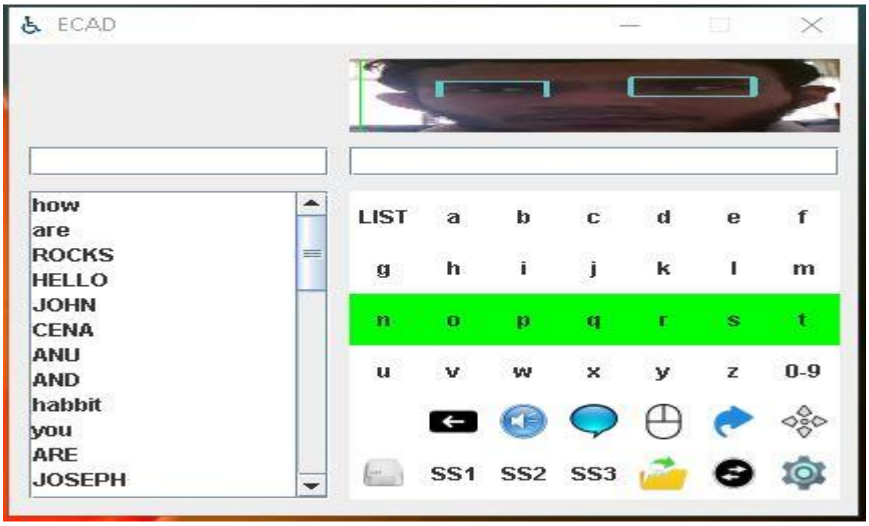


Figure 6.3: Eye blink detection

```
webSource.retrieve(frame);
Graphics g = MainWindow.canvas.getGraphics();
faceDetector.detectMultiScale(frame, faceDetections);
for (Rect rect : faceDetections.toArray())
{
Imgproc.rectangle(frame, new Point(rect.x, rect.y), new Point(rect.x + rect.width, rect.y + rect.height),new Scalar(0, 255,0));
rectCrop =new Rect(rect.x - 5,rect.y - 5,rect.width + 5,rect.height + 5);
}
MatOfRect eyes = new MatOfRect();
cascadeEyeClassifier.detectMultiScale(frame, eyes);
if(eyes.toArray().length==0)
{
count_blink++;
```

```
//counter++;
if(count_blink>=0)
{
//System.out.println("1. Blink Detected.");
if(SwitchAcessScanning.row==-1)
{
System.out.println("2. Blink Detected.");
Robot r=new Robot();
r.mousePress(InputEvent.BUTTON1_DOWN_MASK);
r.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);
r.delay(25);
r.mousePress(InputEvent.BUTTON1_DOWN_MASK);
r.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);
//counter=0;
}
else
{
System.out.println("3. Blink Detected.");
Robot r=new Robot();
r.mousePress(InputEvent.BUTTON1_DOWN_MASK);
r.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);
}
}
count_blink=0;
```

Here when the eye blink is detected the code for switch access scanning is executed. So the blink acts as an indicator for selection and highlighting just like the mouse acts as an indicator in switch access scanning.

Chapter 7

Implementation Details

7.1. User interface

Virtual keyboard comprising alphabets, numbers, speak button, shortcuts, navigation and facility to open various applications easily along with list showing most frequently used words.

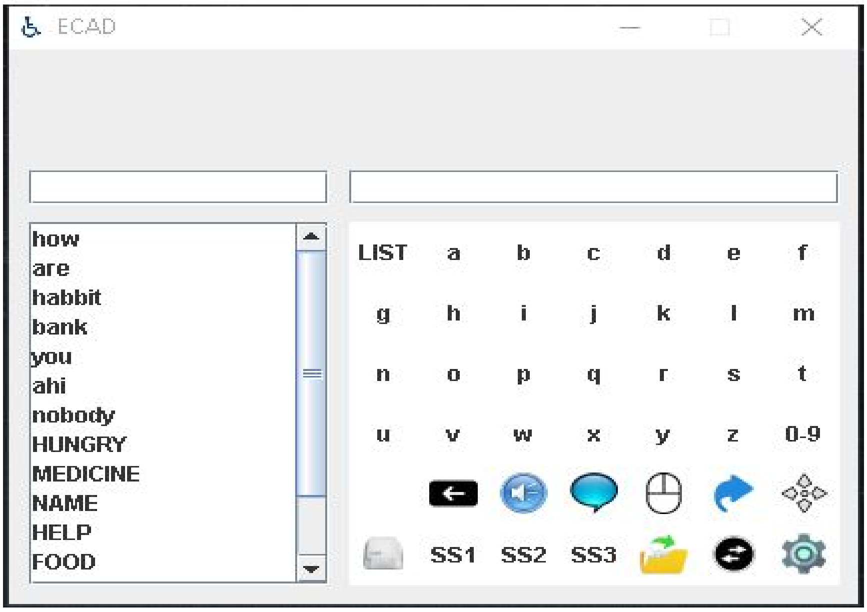


Fig 7.1: User interface

7.2. Database

Database consists of four tables:

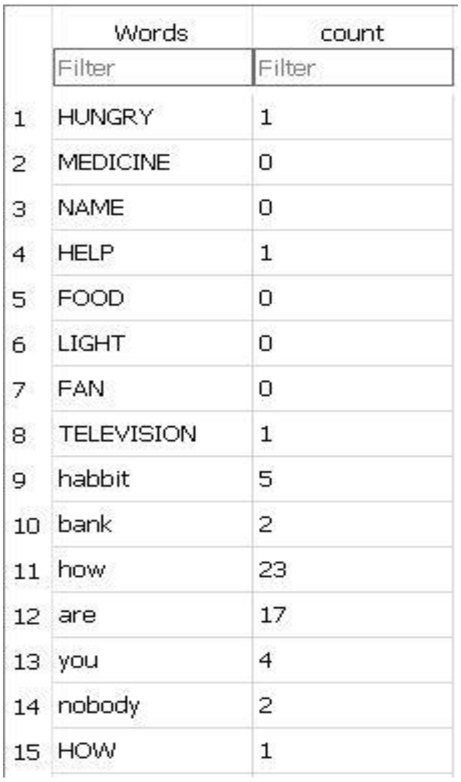
Words: To hold the most frequently typed words.

Shortcuts: To hold shortcuts i.e. key click combinations.

Application: To hold the applications that needs to be frequently opened by the user.

System configuration: To hold system settings like the scanning method, scanning time, mouse scanning time and voice type.

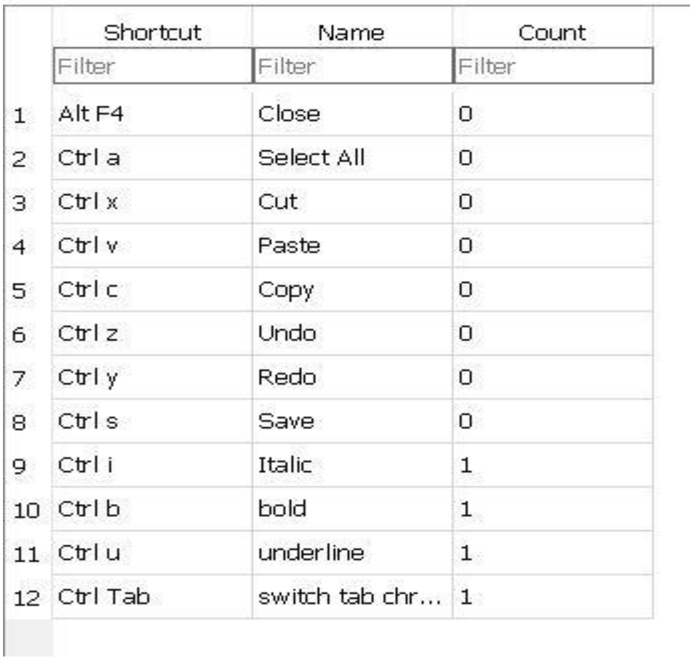
Table words which contains the set of words and its number of occurrence



	Words	count
	Filter	Filter
1	HUNGRY	1
2	MEDICINE	0
3	NAME	0
4	HELP	1
5	FOOD	0
6	LIGHT	0
7	FAN	0
8	TELEVISION	1
9	habbit	5
10	bank	2
11	how	23
12	are	17
13	you	4
14	nobody	2
15	HOW	1

Table 7.1: Words

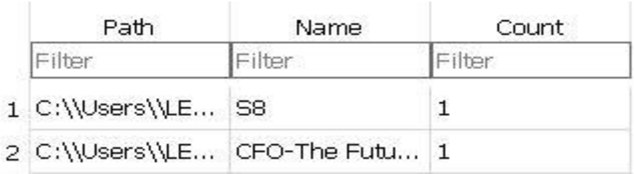
Table shortcuts which contains various shortcuts for easy access of computer system



	Shortcut	Name	Count
	Filter	Filter	Filter
1	Alt F4	Close	0
2	Ctrl a	Select All	0
3	Ctrl x	Cut	0
4	Ctrl v	Paste	0
5	Ctrl c	Copy	0
6	Ctrl z	Undo	0
7	Ctrl y	Redo	0
8	Ctrl s	Save	0
9	Ctrl i	Italic	1
10	Ctrl b	bold	1
11	Ctrl u	underline	1
12	Ctrl Tab	switch tab chr...	1

Table 7.2: Shortcuts

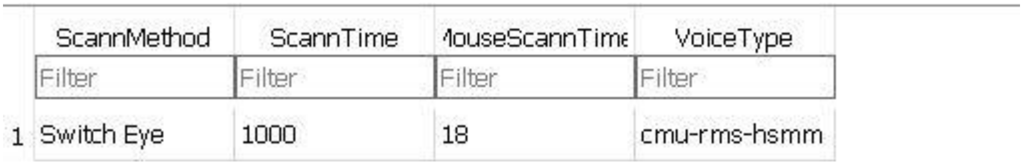
Table application contains the name and the path of the application that the user wants to open.



	Path	Name	Count
	Filter	Filter	Filter
1	C:\\Users\\LE...	S8	1
2	C:\\Users\\LE...	CFO-The Futu...	1

Table 7.3: Application

Table system configuration helps users to customize ECAD



	ScannMethod	ScannTime	/louseScannTim€	VoiceType
	Filter	Filter	Filter	Filter
1	Switch Eye	1000	18	cmu-rms-hsmm

Table 7.4: System configuration

7.3. List

Now other than the database we have a list which contains some other useful tools. We have not used the database to store this data as they are of fixed size and they don't require a database.

The list and its contents are as follows:

Number: 0,1,2,3,4,5,6,7,8,9

Mouse actions: Right, left, left double click, drag, hover, right + left, left + left

Navigations: Up, down, left, right, page up, page down, home, end, windows.

Other Buttons: Caps lock, enter, tab, ctrl, alt, delete, insert, esc, shift, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12.

Symbol Set 1: ( ) < > [ ] { } " '

Symbol Set 2: + - * / % ^ = \ ` ~

Symbol Set 3: : ; ! @ $ # & _ . , ? |

Switch Application: Alt + Tab, Alt + 2 Tab, Alt + 3 Tab, Alt + 4 Tab, Alt + 5 Tab, Alt + 6 Tab, Alt + 7 Tab, Alt + 8 Tab, Alt + 9 Tab, Alt + 10 Tab, Alt + 11 Tab, Alt + 12 Tab.

Chapter 8

Conclusion

ECAD is software that can be used by paralyzed people to uses their computer system with ease and also for communication purposes. With the help of the mouse or eye blink users can control the computer. Step scanning is used. Right mouse button is used for scanning the buttons and the left mouse button is used for the selection of the scanned button. Switch Scanning allows easy simulation of keyboard by using single mouse button. Eye blink detection is used with switch access scanning.

Text to speech converter (TTS) is used .The Voice is the central processing point for FreeTTS. The Voice takes as input a FreeTTSSpeakable, translates the text associated with the FreeTTSSpeakable into speech and generates audio output corresponding to that speech. MaryTTS provides both male and female voice.

Eye blink used as an interface for keyboard simulation.

The facility to control the mouse is also provided using mouse simulation.

Database is used to show suggestion of frequently used words. And also used for saving new words. It also contains the list of applications that needs to be opened, shortcuts etc.

We hope that our system will be helpful to some users for their communication purpose.

Chapter 9

Future Scope

- Other hardware can be implemented along with this system like sensors to monitor the heart rate and breathing. If any fluctuations happen in their heart rate then a notification or a help message can be sent.
- The blink detection can be improved to provide much better accuracy so that the selection and highlighting happens in synchronisation with the eye blink.
- Isolating the left and right eye so that step scanning is possible using both the eyes.

Chapter 10

References

1. Adaptable virtual keyboard and mouse for people with special needs: http://ieeexplore.ieee.org/document/7821782/
2. On Eye Gaze System to Operate Virtual Keyboard: https://ieeexplore.ieee.org/document/7860997/
3. Assistive context aware toolkit (ACAT): https://01.org/acat
4. Low cost eye based human computer interface system: https://ieeexplore.ieee.org/document/7838906/
5. Text-to-speech (TTS) Overview: www.voicerss.org/tts/
6. Switch access Scanning: en.wikipedia.org/wiki/Switch_access_scanning
7. Step Scanning: en.wikipedia.org/wiki/Switch_access_scanning
8. SQLite database: www.tutorialspoint.com/sqlite/
9. Mary TTS: http://mary.dfki.de/, https://github.com/goxr3plus/Java-Text-To-Speech-Tutorial
10. Open CV Java - http://opencv-java-tutorials.readthedocs.io/en/latest/