From Thinking To Text –Alphabet Recognition Through Brainwaves

By

RijaFidaUllahtipu

Hashim Ehsaan Bandesha

SaqlainHaider

Submitted to the Faculty of Computing Software Engineering

National University of Sciences and Technology, Islamabad

ABSTRACT

From Thinking To text-Alphabet Recognition through Brainwaves

Our brain is made up of billions of brain cells called neurons, which use electricity to communicate with each other. The combination of millions of neurons sending signals at once produces an enormous amount of electrical activity in the brain, which can be detected using sensitive medical equipment (such as an EEG), measuring electricity levels over areas of the scalp. The combination of electrical activity of the brain is commonly called a Brainwave pattern, because of its cyclic, *'wave-like'* nature. Our mind regulates its activities by means of electric waves which are registered in the brain, emitting tiny electrochemical impulses of varied frequencies, which can be registered by an electroencephalogram.

Paralysis patients are those which loss or impairment of voluntary movement in a body part, caused by injury or disease of the nerves, brain, or spinal cord. Due to this disease they are not able to communicate with other people. To make them able to communicate we have developed this application.

Now the patient wears the EEG headset, the EEG headset send the application the values of different types of brain waves and on the basis of these values our application will guess about the alphabet which the patient was thinking. This program will be developed using Java. The program will use Artificial Neural Networks for machine learning.

The target population of our project is patient with paralysis, and the also anyone can use this application for their own ease. The scope of work is limited to getting brain waves values via EEG headset and the decision support structure will be based upon it.Through this achieved recognition of alphabets, this project can serve to become a basis for further development to speech recognition which will serve the paralyzed and otherwise completely handicapped people to convey their inner most thoughts.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis –From Thinking To text- Alphabet Recognition through Brainwaves carried out by RijaFidaUllahTipu, Hashim Ehsaan Bandesha, SaqlainHaiderunder supervision of Dr. Seemab Latif for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Dr. Seemab Latif

CSE DEPARTMENT

MCS

DATED:

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation,

a work of this magnitude would not have been possible

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support and their critical reviews. We would like to thank our supervisor. Dr. Seemab Latif, for her continuous guidance and motivation throughout the course of our project. Without their help we would have not been able to accomplish anything.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1.1 Overview

Brainwaves are electrical impulses in the brain. There are 8 different types of brainwaves which includes High Alpha, Low Alpha, High Beta, Low Beta, High Gamma, Low Gamma, Delta, Theta. Now the values of these 8 different types of brainwaves can be get by using and EEG headset.We studied the experiment and found that when a person thinks about a certain thought then there is a pattern in the values of these brainwaves. So the idea of this project is to develop the program which should be able to recognize the patterns of 5 alphabets and on the basis of these patterns the program should tell the user what alphabet the user was thinking.

1.2 Problem Statement

We are concerned with alphabet recognition through the use of brainwaves. The alphabets may be any chosen by us for accurate differentiation. In this project we aim at recognizing any 5 alphabets.

1.3 Approach

We are working with Neuroskymindwave. It is a headset that differentiates brainwaves into 8 different types' waves. We aim to receive those wave values and differentiate them for each alphabet using neural Networks

1.4 Scope

The scope of the project will be to achieve alphabet recognition and have it written down by a text editor in computer through the use of a special headset that interprets brainwaves into electrical signals. These electrical signals would then be interpreted to code in order to achieve this alphabet recognition. We will be recognizing 5 alphabets which are A O N Z T

1.5 Objectives

The Objectives of this project are following:

- To learn about brainwaves
- To learn about the EEG(electroencephalogram)
- To learn Java Development Process/Cycle
- To learn Java Programming
- To learn about Artificial Neural Networks
- To learn about Weka

Through this achieved recognition of alphabets the project can serve to become a basis for further development to speech recognition which will serve the paralyzed and otherwise| completely handicapped people to convey their inner most thoughts.

1.6 Deliverables

Sr.	Tasks	Deliverables
1	Literature Review	Literature Survey
2	Requirements Gathering	SRS Document
3	Application Design	Design Document (SDS)
4	Implementation	Implementation on computer with a live test to show the accuracy and ability of the project
5	Testing	Evaluation plan and test document
6	Training	Deployment Plan
7	Deployment	Complete application along with necessary documentation

1.7 Overview of the document

This document shows the working of our application FTT(From Thinking to text). It starts of with the system architecture which highlights the modules of the software and represents the system in the form of Block Diagram, component diagram, Use Case Diagram, ER Diagram , Sequence Diagram and general design of the system. Then we move on to discuss the detailed Description of all the components involved. Further we discuss the dependencies of the system and its relationship with other products and the capacity of it to be reused. Then towards the end we shall discuss the Design Tradeoffs and the Pseudocode.

1.8 Purpose of the document:

This document aims to Elaborate the idea and design of the project that is From thinking to text using Brainwaves. This document will highlight all the specifications of our project i.e. how it will be used, what will be the scenario in which the project will be useful. This project is basically a baseline work for further research that is ultimately capturing the thoughts of human brain.

Chapter 2: Literature Review

There were a few projects that were based on the idea of thought recognition following is a detailed description of projects previously carried out in this context.

Brain Writing Research

BOSTON, Mass. -- Neuroscientists at the Mayo Clinic campus in Jacksonville, Fla., have demonstrated how brain waves can be used to type alphanumerical characters on a computer screen. By merely focusing on the "q" in a matrix of letters, for example, that "q" appears on the monitor.

Researchers say these findings, presented at the 2009 annual meeting of the American Epilepsy Society, represent concrete progress toward a mind-machine interface that may, one day, help people with a variety of disorders control devices, such as prosthetic arms and legs. These disorders include Lou Gehrig's disease and spinal cord injuries, among many others.

From thinking to Text -Alphabet Recognition Through Brainwaves

"Over 2 million people in the United States may benefit from assistive devices controlled by a brain-computer interface," says the study's lead investigator, neurologist Jerry Shih, M.D. "This study constitutes a baby step on the road toward that future, but it represents tangible progress in using brain waves to do certain tasks."

Dr. Shih and other Mayo Clinic researchers worked with Dean Krusienski, Ph.D., from the University of North Florida on this study, which was conducted in two patients with epilepsy. These patients were already being monitored for seizure activity using electrocorticography (ECoG), in which electrodes are placed directly on the surface of the brain to record electrical activity produced by the firing of nerve cells. This kind of procedure requires a craniotomy, a surgical incision into the skull.

Dr. Shih wanted to study a mind-machine interface in these patients because he hypothesized that feedback from electrodes placed directly on the brain would be much more specific than data collected from electroencephalography (EEG), in which electrodes are placed on the scalp. Most studies of mind-machine interaction have occurred with EEG, Dr. Shih says.

"There is a big difference in the quality of information you get from ECoG compared to EEG. The scalp and bony skull diffuses and distorts the signal, rather like how the Earth's atmosphere blurs the light from stars," he says. "That's why progress to date on developing these kind of mind interfaces has been slow."

Because these patients already had ECoG electrodes implanted in their brains to find the area where seizures originated, the researchers could test their fledgling brain-computer interface.

In the study, the two patients sat in front of a monitor that was hooked to a computer running the researchers' software, which was designed to interpret electrical signals coming from the electrodes.

The patients were asked to look at the screen, which contained a 6-by-6 matrix with a single alphanumeric character inside each square. Every time the square with a certain letter flashed, and the patient focused on it, the computer recorded the brain's response to the flashing letter. The patients were then asked to focus on specific letters, and the computer software recorded the information. The computer then calibrated the system with the individual patient's specific brain wave, and when the patient then focused on a letter, the letter appeared on the screen.

"We were able to consistently predict the desired letters for our patients at or near 100 percent accuracy," Dr. Shih says. "While this is comparable to other researchers' results with EEGs, this approach is more localized and can potentially provide a faster communication rate. Our goal is to find a way to effectively and consistently use a patient's brain waves to perform certain tasks."

Once the technique is perfected, its use will require patients to have a craniotomy, although it isn't yet known how many electrodes would have to be implanted. And software would have to calibrate each person's brain waves to the action that is desired, such as movement of a prosthetic arm, Dr. Shih says. "These patients would have to use a computer to interpret their brain waves, but these devices are getting so small, there is a possibility that they could be implanted at some point," he says.

"We find our progress so far to be very encouraging," he says.

Brain Keyboard

People who have severe movement disorders like Amyotrophic Lateral Sclerosis (ALS) or Locked in Syndrome, is a condition where a patient is awake and aware of its surrounding but unable to communicate or perform any action due to paralysis of almost every voluntary muscles in the body (with the exception of eye movements and blinking). The people who are suffering from severe Cerebral Palsy disorder are not able to speak or not understandable enough so they could not communicate and interact with others.

These people also have severe movement disorder. Imagine these patients having a fully functional brain trapped within a non-functioning body. The brain of the patient would be fully conscious and aware of its surroundings, it could think and process stimuli, but unable to translate thought into action. This is caused by a gradual degeneration of the nerve cells in the central nervous system that controls voluntary muscle movement.

Application

Abhinav has developed a Brain Keyboard Application. He used the processing platform and Think gear library to develop a Brain Keyboard application. This is a low cost solution which can serve as a big help in communication and interaction for people who have severe movement disorders like Amyotrophic Lateral Sclerosis (ALS) or Locked in Syndromes, cerebral palsy etc.

Technology

He uses NeuroskyMindwave EEG sensor. It detects

- Attention
- Meditation
- Intentional Eye Blink

Working

The application has 26 alphabetical keys with a space and a delete key. Here, a square selection box scans all the keys one by one in a sequence. There is a time delay of about 100ms to move the square selection box from one key to another key. The application takes the input when user blinks his eye. The eye blinks triggers the current selected key (as shown in the picture) and the letter gets print on the text box

Brain Controlled Drone

Another technology in the world of brainwaves are the brain controlled Drones that work on the principle of direction and the Drone takes off when the command “lift off “ is given and then the drone flies In different directions based on brain commands highly reliant on concentration levels of the user wearing the headset that is ending brain signals to the Drone.

Chapter 3: Software Requirement Specification

3.1 Introduction

The purpose of this part is to describe theproject titled “From Thinking to Text”. This part contains the functional and non-functional requirements of the project. It contains the guidelines for developers and examiners of the project.

Our project is to develop an application for translating thinking to text using brainwaves. We will preprocess the brain waves to reduce noise by enough to get a good signal into the computer. After pre-processing, brainwaves will be analyzed and finally interpret these signals into code and get meaningful result out of them.

The scope of this project is limited to recognizing 5 alphabets.

3.2 Overall Description

3.2.1 Product Perspective

The product is a new project that will help in future for more extensive work in this field. The aim of this project is to develop an application for translating thinking to text using brainwaves. Brain waves will be captured using the headset. Then we will preprocess the brain waves to reduce noise by enough so that we categorize the signals in a closer bracket and feed it to the computer, where we will process the data a bit more. As we will be using our computer's sound card to get the data in, we have to cut noise enough that the signal with noise does not spike above an amount of certain negative and positive voltage which would be the point where the sound card clips the data off. After pre-processing, brainwaves will be analyzed and then they will be calibrated so as to achieve frequency levels of desired alphabets and finally after matching frequencies the output will be displayed using a text editor. Through this recognition of alphabets, the project can serve to become a basis for further development to speech recognition which will serve the otherwise speech impaired or paralyzed patients.

3.2.2 Product Functions

Following are the functions which will be performed:

1. Brain Wave Reading
2. Noise Cancellation
3. Signal Enhancement
4. Categorizing and Matching of Brain Waves (Machine Learning Algorithm will be used)
5. Output in form of Alphabets

3.2.3 User Classes and Characteristics

Following are the user classes and their brief description.

3.2.3.1 Researchers

Researchers will use this Project as a guide to understand the “Brain waves”. They will use this as a base for upgrading and adding new features. They can also use this for developing a new project by using this a reference material.

3.2.3.2 Tester

Tester will also use this project to check for bug finding. They will also use this to check it is in accordance to the srs document.

3.2.3.3 Project Evaluator/Supervisor

Project supervisor/Evaluator will also use the product to evaluate. They will use this product to find the accuracy and error in the output.

3.2.4 Operating Environment

The operating environment required for this project is:

3.2.4.1 Software requirements:

Visualizer: To visualize the brain waves on computer.

OS: Windows 7, 8, 8.1

3.2.5 Design and Implementation Constraints

Following are the constraints of design and implementation in our project

Noise:

As we will be testing the project in a room so there will be many electrical appliances due to which *electrical noise* will be there. Apart from electrical noise brain waves will also have noise which can affect the accuracy or output of the project.

Signal spikes:

Brain Signals are very small so you need signal enhancement and stuff like that. This can also affect the output of the project.

Signal Strength:

As the headset will communicate with the “Brain Wave Visualizer” through Bluetooth so the signal strength will also have a great impact on the working of the project. Signal strength will be used to detect poor contact and whether the device is off the head

3.2.6 User Documentation

For the user documentation, a user manual will be provided with the system. It will include the details of the system's working. Help documents will also be a part of the system. The project report will also be available for the users which will highlight the system features, working and procedures.

3.2.7 Assumptions and Dependencies

1. Overall performance of this project will depend on the accuracy of the headset used in gathering brainwaves.
2. Accuracy depends on noise removal and signal enhancement.
3. It will also depend on the alertness of the person whose brainwave activity will be recorded and converted.

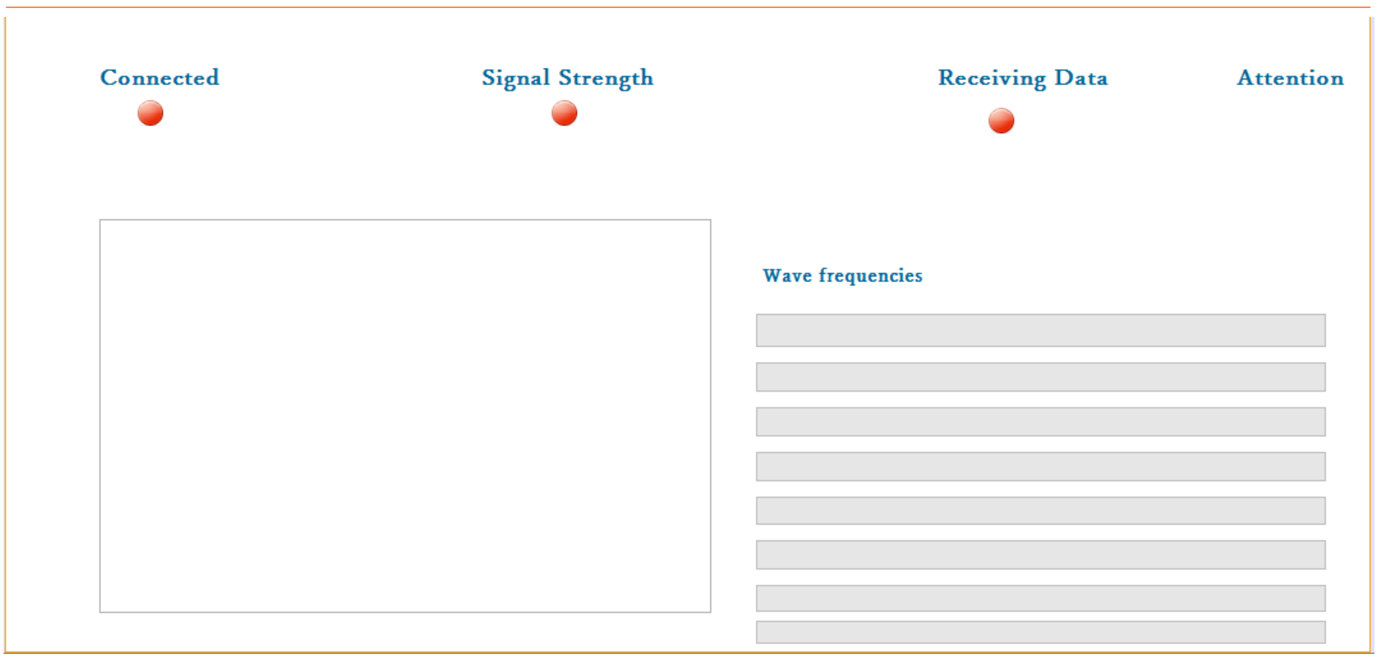
3.3 External Interface Requirements

3.3.1 User Interfaces

The first UI screen will be gathering brainwaves and calibrating the Headset. Next UI screen will be where the brainwaves will be converted to text and the output will be show in the right side of the screen.

1. The communication between the software and the headset will be using Bluetooth.
2. The brainwaves activity will be gathered by the NeuroSky headset.

User Interface:



3.3.2 Hardware Interfaces

1. The communication between the software and the headset will be using Bluetooth.
2. The brainwaves activity will be gathered by the NeuroSky headset.

3.3.3 Software Interfaces

To visualize the brainwaves on the monitor Brainwave Visualizer software will be used.

3.3.4 Communication Interfaces

The Bluetooth in the headset has following features:

1. BT Version: 2.1
2. BT Output Power: Class 2
3. BT Minimum Voltage: 1.0V
4. BT Range: 10m range
5. BT Power Consumption: 80mA (when connected and transmitting)
6. Low Battery Indicator 1.1V

3.4 System Features

3.4.1 Brain waves reading

3.4.1.1 Description and Priority

This feature allows the user to collect the Brain Waves of the "*User".* This is the most important feature as all other functions are dependent upon this feature.

3.4.1.2 Stimulus/Response Sequences

1. Once the headset is connected to Software a message will be show "Headset Connected ".

2. Then User will Press Button “Gather brain waves/Calibrate”. The Headset will gather brainwaves and will display brains in the side window in the software.
3. Once the Brain wave is gathered for specific time it will be saved for further processing.

3.4.1.3 Functional Requirements

REQ-1: The software will check whether it is connected and communicating with headset for gathering waves.

3.4.2 Noise Cancellation:

3.4.2.1 Description and Priority

This feature will help user to cancel the noise and make required waves like alpha, beta, gamma more pure to use and achieve desired result. This feature is high priority as noise will have a great impact on the final output as lesser the noise better the output.

3.4.2.2 Stimulus/Response Sequences

1. Once brain waves gathered a dialog message will be show for the user “Press continue to reduce noise/cancel noise”.
2. User presses continue and the noise cancellation process will start resulting in wave with reduced noise.

3.4.2.3 Functional Requirements

REQ-1: For noise cancellation Brain wave will be required which will be stored after Brain Waves reading.

3.4.3 Signal Enhancement:

3.4.3.1 Description and Priority

This feature will help user to Enhance the recorded brain waves signals spike more understandable .This feature is high priority as well as the more enhanced wave will result in most accurate output.

3.4.3.2 Stimulus/Response Sequences

1. Once brain waves has passed the noise cancellation step a dialog message will be show for the user “Press continue to Perform Signal Enhancement”.
2. User presses continue and the Signal cancellation process will start resulting in wave with Enhanced wave.

3.4.3.3 Functional Requirements

REQ-1: For signal Enhancement Brain waves with reduced noise will be required which will be stored after Noise Cancellation.

3.4.4Signal Categorizing and Wave Matching:

3.4.4.1 Description and Priority

This feature will help user to categorize and match brain waves signals into alpha, beta, gamma etc. This feature is high priority as well if the waves are not categorized correctly it will result in false output.

3.4.4.2 Stimulus/Response Sequences

1. After signal Enhancement user will press continue to categorize the brain waves and the values of waves will be stored.
2. Waves will also be show in a side window for user with values for easy understanding.

3.4.4.3 Functional Requirements

REQ-1: For categorizing and matching Enhancement will be required which will be stored after step three.

3.4.5 Output

3.4.5.1 Description and Priority

This will be the last step. This feature will show the output after the processing of the wave in the features mentioned above. This is also important because it will show what user was thinking.

3.4.5.2 Stimulus/Response Sequences

1. After categorize and matching of the waves output will be show in a new window.
2. After output is show user will be asked to give feedback/ mark how accurate was the result for future improvements.

3.4.5.3 Functional Requirements

REQ-1: Categorized and correct matching of brain waves for correct output will be required.

3.5 Other Nonfunctional Requirements

3.5.1 Performance Requirements

1. The project should be able to calibrate in minimum time.
2. The project should be accurate enough so that it can be used in future as well for other developments.
3. This project should use the minimum resources.

3.5.2 Safety Requirements

As the headset will communicate via Bluetooth so it should not be used for long time.

3.5.3 Software Quality Attributes

1. **Maintainability:** The maintainability cost for this project should be less.
2. **Accuracy:** This accuracy should be there in the reading and converting from thinking to text.
3. **Usability** - System should be simple and user friendly interfaces need to be provided.

Chapter 4: Design and Development

4.1 INTRODUCTION

Currently a lot of study is being conducted world over on The Human Brain and its activity. The brain is the most complex tool in the human body and is responsible for the whole body's functionality. We are particularly concentrating on the linguistic abilities of the Brain and circled our project around the Alphabet recognition capacity of the Brain.

Today a very huge research is concentrated on capturing the thoughts of Human Mind. This could revolutionize the medical field specifically as it would give the ability to communicate to people who are paralyzed or in comma this would also have extensive applications around all areas of Life. But currently no such research or work has been done in Pakistan and we aim to provide baseline for this endeavor or a starting point into this area of research.

4.1.1 Over View of Modules/Components.

FTT(From Thinking To Text) comprises of following components:

1. Application GUI
2. Headset(Hardware)
3. Process data
4. Data Comparison

Application UI acquires all its services from the Process Data module. The user interacts directly with this component and provides an input to the application which is different for a already familiar user and a different interface Is displayed for an unfamiliar or new user.

Headset Is the hardware that captures the brainwaves and creates a packet and sends data per second to the software via Bluetooth connection and the data is displayed in the form of waves.

Input Datais the first data that is given by the user as he/she places the headset on head and connect the headset with the software application via Bluetooth.

Processing Data is a module that handles the whole input data and extracts the necessary information out of it that is for every user it stores a Data set of values and familiarizes itself via a Neural Network. Following are further 3 modules of processing data

- **User Data** is that data which is stored when for an already familiar user so that when a familiar user comes non need to train the user or calibrate the user on different Alphabets
- **Calibration data** is a data set that is stored incase there is a new user for this the data of the new user is first checked against the user data if no match then another data set is created for the new user for all the alphabets
- **Test data** Is that data when we have verified that the user is not new we simply access the test data Files and access the already stored values for all the alphabets.

Data Comparison is that component of the system which compares the fresh data values generated by the headset against the stored values of waves for a particular Alphabet.

4.1.2 Scope of the Development Project

The project aims at developing a baseline application for Thinking an Alphabet and converting it to text. This application also helps to provide a starting point for further research in this field that may someday further lead to complete thought recognition.

4.2 SYSTEM ARCHITECTURE DESCRIPTION

Detailed description of system architecture and design pattern which this system is going to use is discussed later in the document in section 5 'Design Decisions and Tradeoffs'.

This Section overview of application, its higher and lower levels details and user interfaces.

4.2.1STRUCTURE AND RELATIONSHIPS

This section covers the overall technical description of FTT(From thinking to text) (refer Section 1.3). It shows the working of application in perspective of different point-of-views and also shows relationships between different components.

4.2.1.1 System Block Diagram

This diagram shows the higher level description of the application. It shows generic working of the application and interaction with the user. In this application, User will interact with the UI of the application, UI will interact with the application logic which will then interact with Files. The ANN will then train the system on the specific data set. The comparison component will then compare the fresh data set of the user with the trained ANN and then the end display component will then output the appropriate Alphabet.

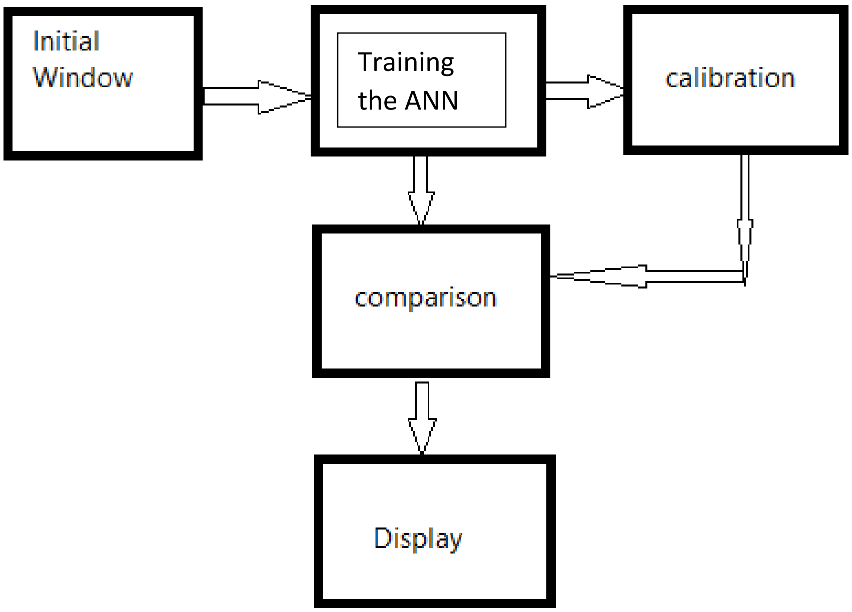


Fig. 2.2.1.1:*Block diagram for FTT*

4.2.1.2 File Handling

We have used File handling in our project as the need for a Files was not necessary and this added efficiency to the project as it speeded up the process of training and matching which in turn increased the speed of the whole software

Following files have been created to hold the relevant Data :

- AO.arff
- AO_Model
- AZ.arff
- AZ_Model
- ON.arff
- ON_Model

- OT.arff
- OT_Model
- TN.arff
- TN_Model
- ZO.arff
- ZO_Model

4.2.1.3 Use Case Diagram

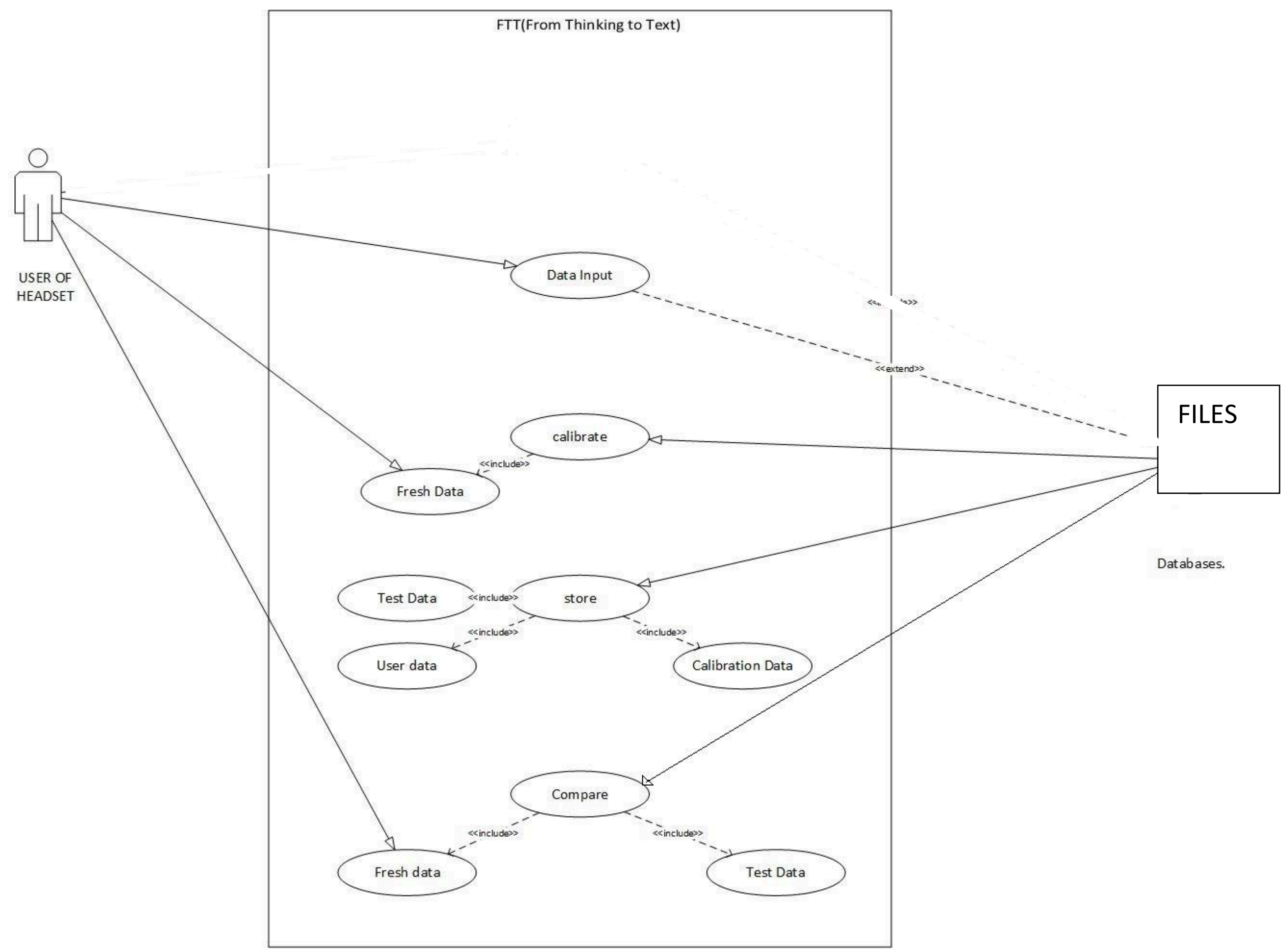


Fig. 2.2.3.1: *Use Case diagram for FTT*

Actors

Primary Actor(s): User of Headset *Secondary Actor(s):*Files

Use Cases

1. Data Input
2. Calibrate
3. Store
4. Compare

Use Case Description

UseCase 1

Use Case Name	Input Data
Primary Actor	User of headset
Secondary Actor	Files
Normal Course	• The user wears the headset and connects it to the application via Bluetooth • headset starts to send data to the application in form of waves.

Alternate Course	The headset is unable to connect to the application and the session fails
Pre-Condition	The headset must be connected to the application for any transfer of data.
Post Condition	The data is received by the application
Extends	*N/A*
Include	*N/A*
Assumptions	The data is received without hindrance.

UseCase 2

Use Case Name	Calibrate
Primary Actor	User
Secondary Actor	Files
Normal Course	The user wears the headset the signals generated are checked that whether the user already exists or is the user new. In this case user will be new so control will go to calibration and the new user will be calibrated

	against all the Alphabets
Alternate Course	The user is already there so no calibration required.
Pre-Condition	• The user must be a new one. • headset must be properly connected vis Bluetooth.
Post Condition	The user is calibrated successfully against all the Alphabets
Extends	*N/A*
Include	*N/A*
Assumptions	The user is a new one.

Use Case 3

Use Case Name	Store
Primary Actor	N/A
Secondary Actor	Files
Normal Course	The calibrated data or test data or user data is stored successfully in their respective Filess
Alternate Course	There is a Files error and the data is not stored
Pre-Condition	The headset must be connected properly to the

	application via Bluetooth.
Post Condition	The Files is maintained
Extends	*N/A*
Include	*N/A*
Assumptions	The data received is correct and

Use Case 4

Use Case Name	Compare
Primary Actor	N/A
Secondary Actor	Files
Normal Course	The fresh data received by the application is compared against the already stored information for the same Alphabets and if the comparison matches to the data set of a specific Alphabet that alphabet is displayed.
Alternate Course	There is a Files error in comparison and hence no output is produced.
Pre-Condition	• the headset must be connected properly. • There must be correct dataset of fresh values received from the user

Post Condition	The match is a success i.e the value were matched to the correct dataset and the alphabet printed out is a the same shown to the user wearing the headset.
Extends	*N/A*
Include	*N/A*
Assumptions	The data stored in test data is correct and the data received from the headset is also without interruption or error.

4.2.1.4 Sequence diagram

Sequence Diagrams show a series of steps followed by the user in order to operate the software developed successfully

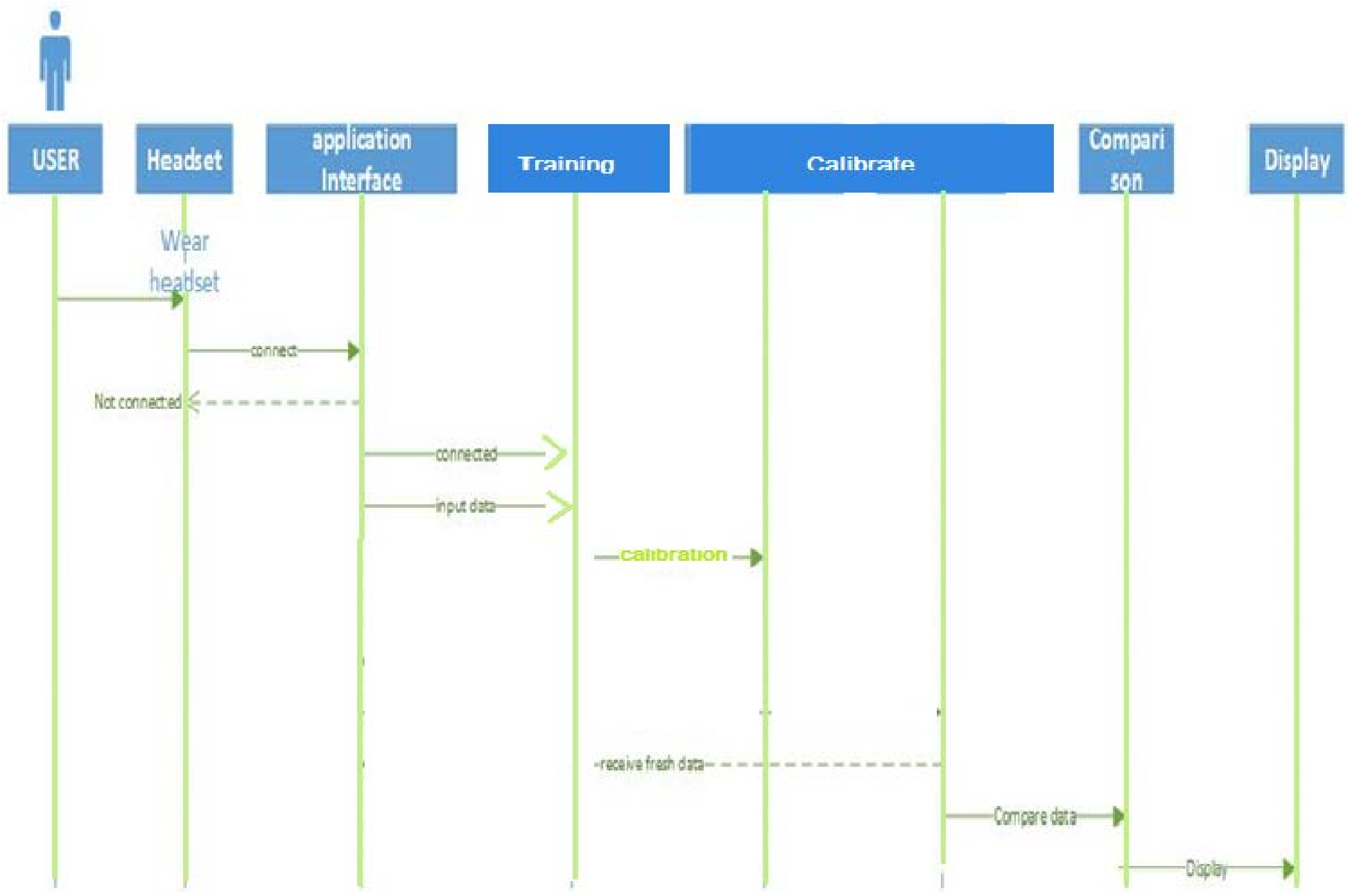


Fig. 2.2.4.2:*Sequence diagram for Unfamiliar User*

4.2.1.4 State Transition Diagram

The State Transitions occurring in the application are shown in fig. 2.2.5.1 below:

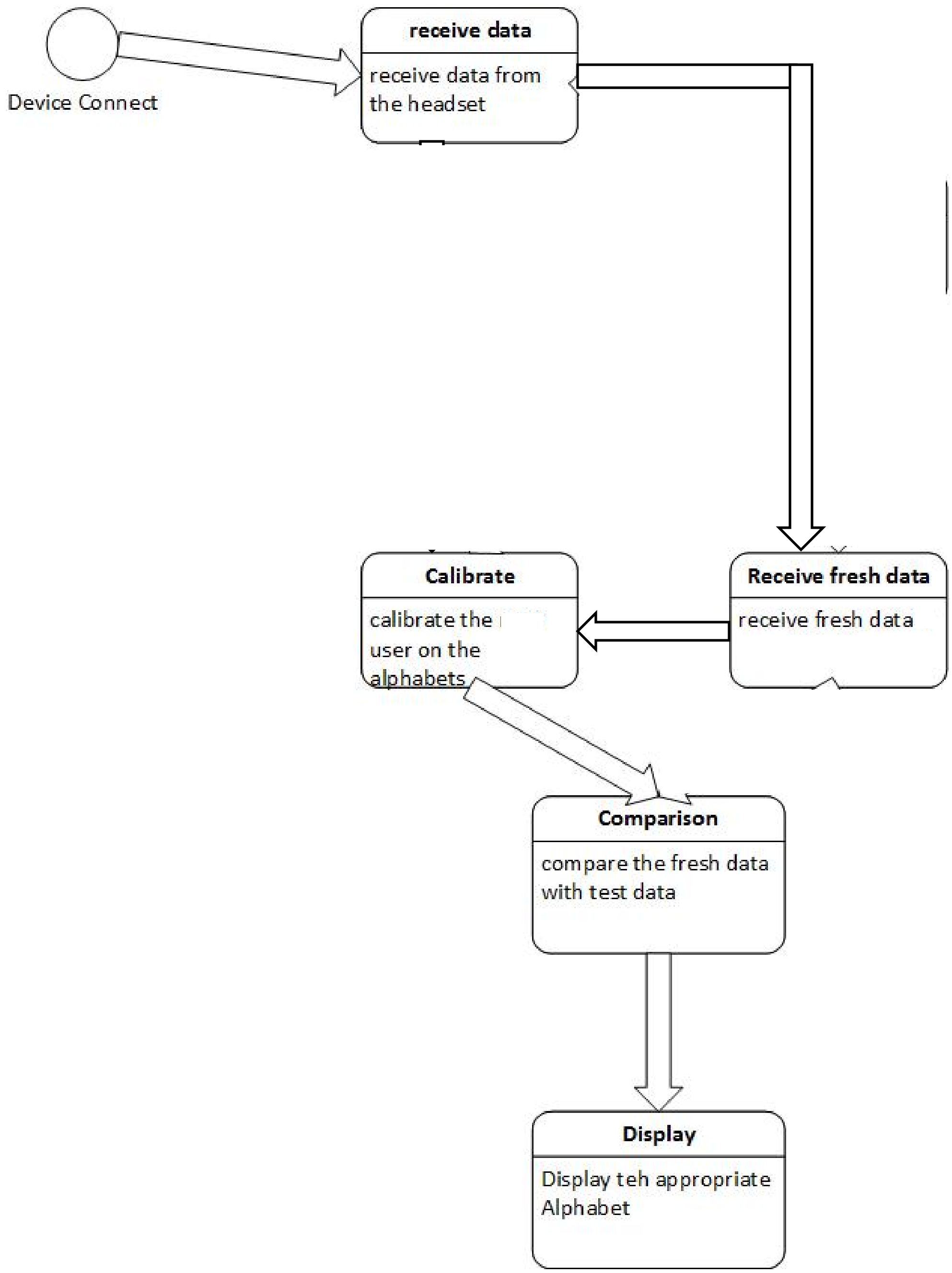


Fig. 2.2.5.1:*State Transition diagram for FTT*

4.2.1.5 Activity Diagram

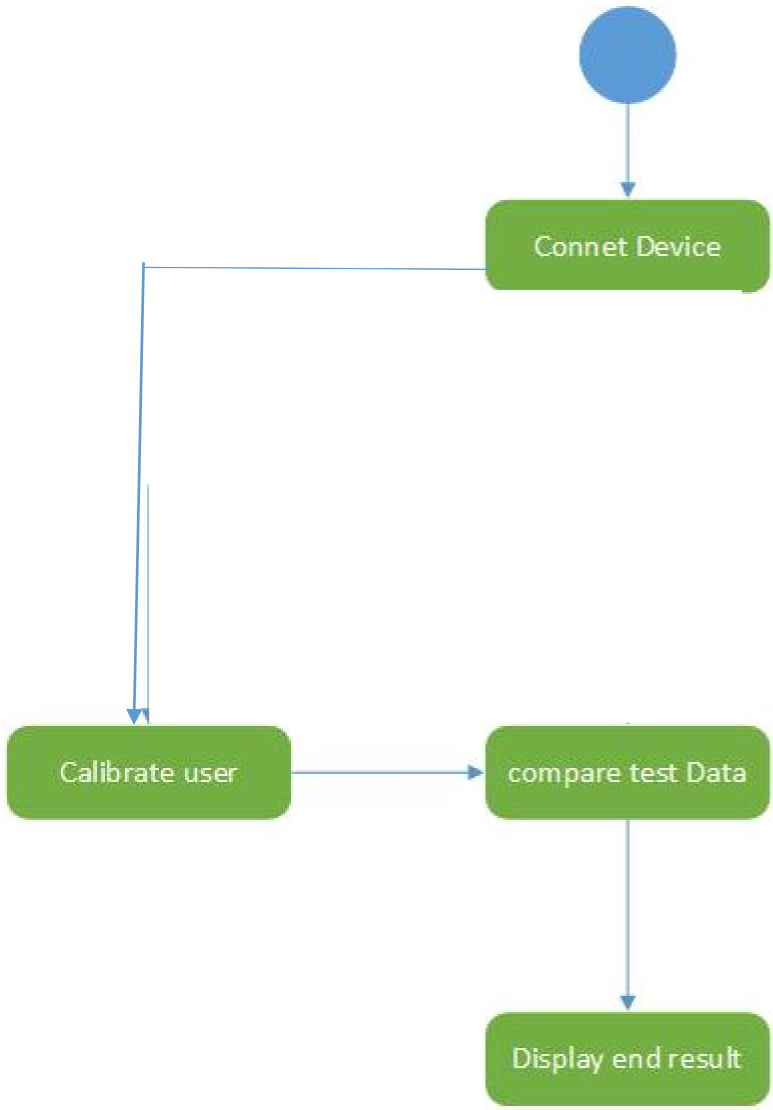


Fig. 2.2.6.1:*Activity Diagram for FTT*

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.

4.2.1.6 Implementation View (Class Diagram)

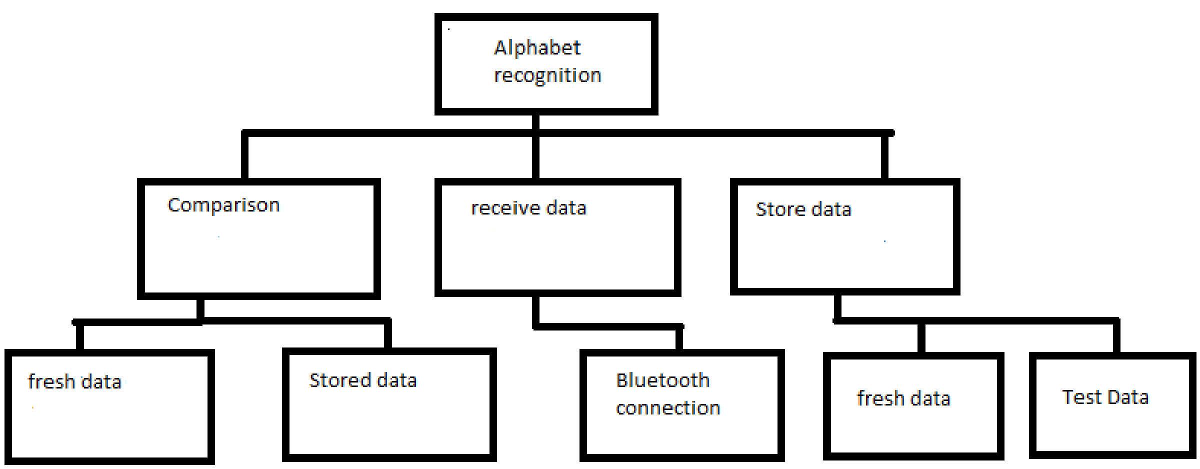
Fig. 2.2.7.1(a): *Class diagram for FTT*

Classes	Description
Main	The main class of the application, which'll be executed first in when the application is run.it check if the headset is connected properly and then starts to receive data .
Receive Data	The data is then received for all the alphabets and the fresh data is calibrated and then matched against the test data or the fresh data is directly matched against the test

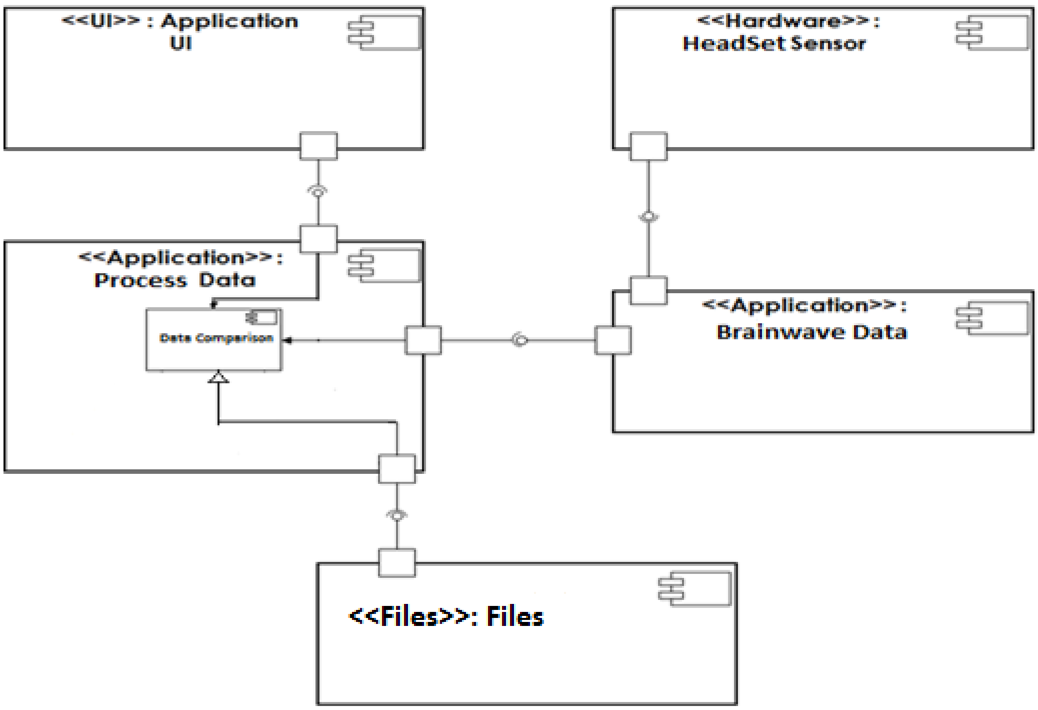
	data.
Match Data	Once the fresh data is received it is matched against the test data
Display	The appropriate Alphabet whose data set the fresh data matches is displayed.

4.2.1.7 Structure chart

This chart shows the breakdown of the application to its lowest manageable levels. It shows the modules and their corresponding functions which this application will implement. This chart basically shows the structure breakdown of the application starting from main modules to specific functions.

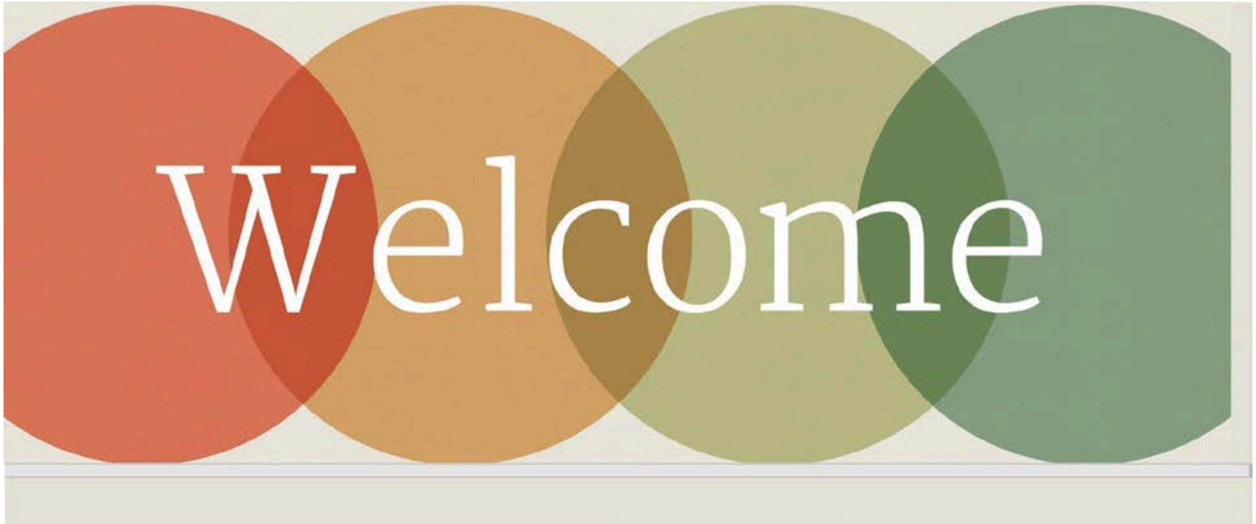


4.2.1.8 Component Diagram:

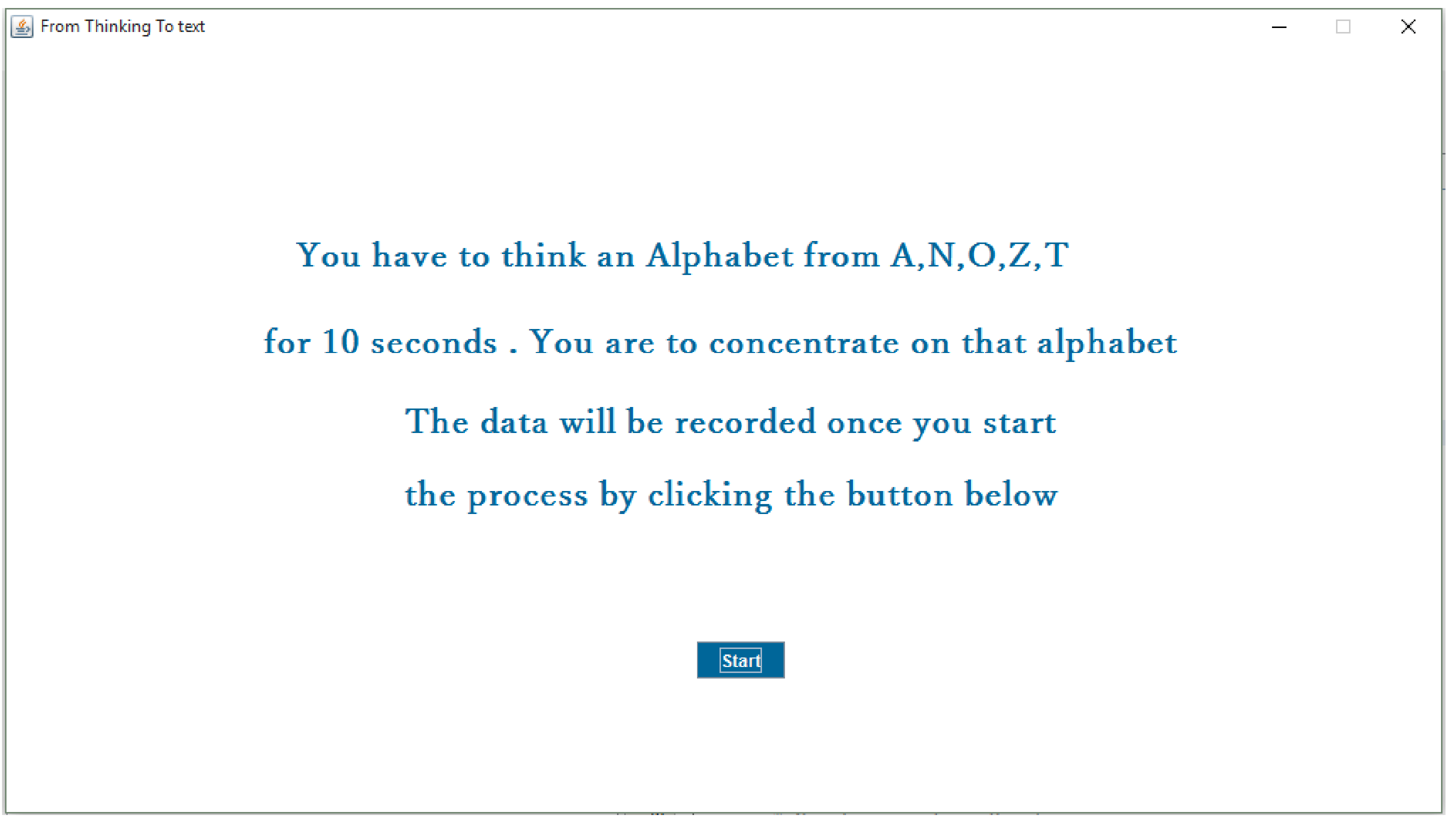


4.2.3 User interface

The user interface has different modules for different scenarios an initial idea about the interface has been displayed below with specifications for each interface.

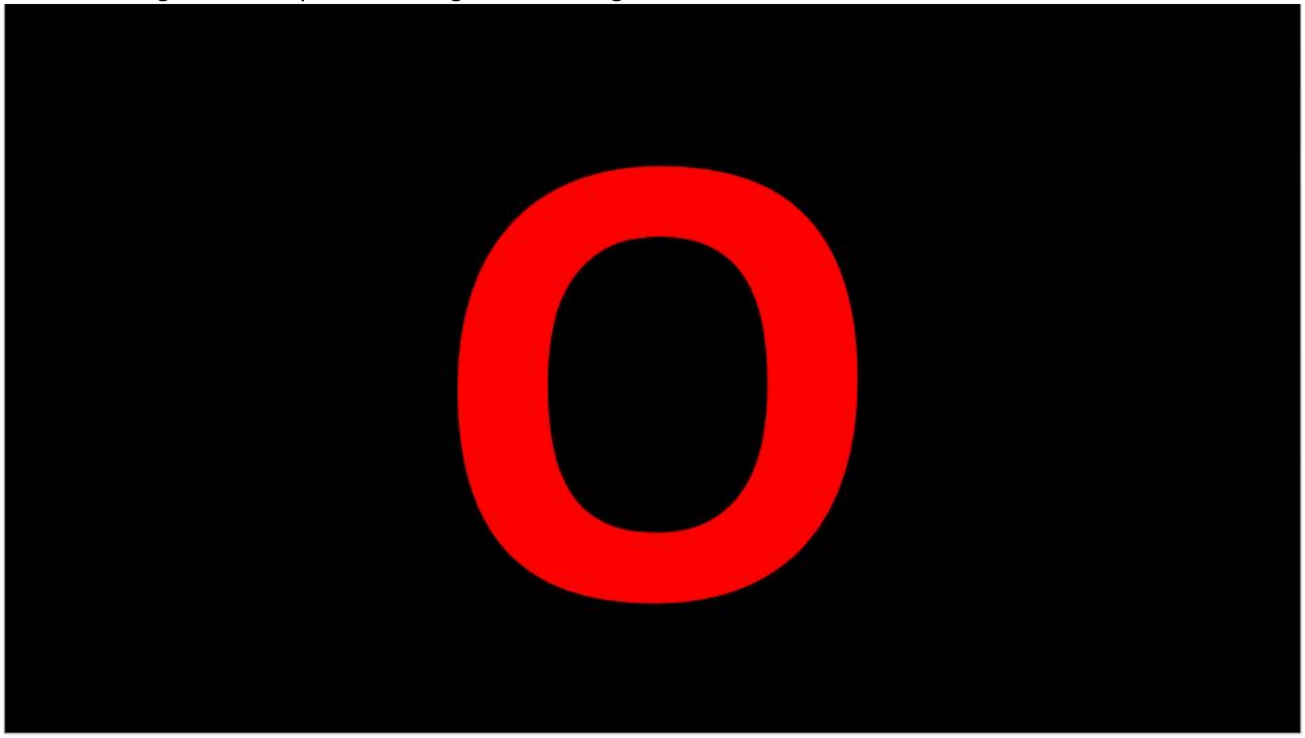


Main Screen



This above screen will instruct the user to concentrate on the letter being shown on a separate screen

Once you have started to concentrate on the letter press the start button it will star gathering your data for about 10 seconds. Below is the screen which will be shown to the user to gather the data



Above screen will be shown to calibrate the user against all alphabets we want to test and this screen will also be displayed once we want to test out our application whether it produces the same result as the alphabet shown.

Once this is done we will move to the final display screen that will display the frequency of the waves and the alphabet detected.

4.3 Detailed Description Of Components

4.3.1 Application UI

Identification	*Name:* Application GUI *Location:* Presentation layer of the system architecture
Type	GUI component
Purpose	The user directly interacts with this component. He/she provides an input for the required action (through this component) and it displays its output respectively. This component fulfills following functional requirements related to user interaction in the application: **REQ-1:** Application should be able to tell whether the headset is connected or not. **REQ-2:** If the invalid username or invalid password is entered, application should not load next screen and generate an error message. **REQ-3:** If both username and password are valid, application should load next screen. **REQ-4:** The application should show the calibration data to the registered user. **REQ-5:** If the user has not calibrated his data then the system

	should prompt to the user to calibrate the system according to the user. **REQ-6:** The user should be then shown the pictures of the alphabets that are to be guessed. **REQ-7:** The system should be able to compare the thinking alphabet with the systems Files and determine what alphabet the user was thinking. **REQ-8:** The system should be able to make amends if the system guessed the wrong alphabet and then train its self-according to it. **REQ-9:** The application should notify if headset is receiving bad signals and then notify the user to wear it properly. **REQ-10:** The application should give the option of logout on different screens so that users can logout of the application at any time.
Function	This component has two major functions; take input from the user and display all application screens. It takes input from user in form of brain waves or other mouse events, and provides a graphical output accordingly.
Subordinates	This component has two subordinates; one is responsible for input, the other for output. The input subordinate satisfies all functional requirements that

	require user input: REQ-2, REQ-3 and REQ-7. The output subordinate satisfies all functional requirements that provide output: REQ-1, REQ-2, REQ-3, REQ-4, REQ-5, REQ-6, REQ-7, REQ-8 , REQ-9 and REQ-10.
Dependencies	It interacts with *Process Data* (refer Section 3.3), whenever a user interacts with the application. This component is dependent on the *Application GUI* whereas no component depends upon it. It gets and stores values by using function calls like *getInfo()*, *setInfo()* etc. (refer Section 3.3)
Interfaces	All user interfaces defined in section 2.3 are part of it. The user input and output on screens will be shown using these interfaces. It will provide external interface to *Process Data* in form of inputs taken from the user. It will display error messages like: 1. Headset not Connected 2. Invalid Username or Password 3. No Calibration Record Exists 4. Unable to Save *(Files related)* 5. Unable to Retrieve *(Files related)*

Resources	**Hardware:** Headset and mouse enable user to interact with the application. It will require a user screen to display the application. The screens will be run by using internal memory i.e.; RAM, Processor of the computer. **Software:** NetBeans(java) will be used to design interface for display on screen.
Processing	Takes user input in form of headset input and other mouse and shapes the output according to user intent. (Refer Section 6.1)
Data	Headset sent packet, Name String, Option Selected Integer etc.

4.3.2 Headset

Identification	*Name:* Headset *Location:* Presentation layer of the system architecture
Type	Hardware component
Purpose	Following functional requirements are fulfilled by this component: **REQ-1:** The application should detect the users brain activity and transmit it for processing through Bluetooth.
Function	This component gets the brain activity of the user while using

	the application. It takes input from 5 sensors and record the 8 different brainwaves and then send the values of those brainwaves in a Bluetooth packet to the application for processing.
Subordinates	It has two subordinates; sensors and Bluetooth sender. Both these subordinates fulfill the REQ-1 mentioned in the purpose section above.
Dependencies	This component isn't dependent on any other component. but it does provide service to the component *Processing* (refer Section 3.4). *Processing* uses these functions of this component: *OnDeviceConnected(),OnDeviceValidation()* and *OnDataRecieved()*.
Interfaces	Headset, which is responsible for getting the brainwaves (refer Section 1.3) of the user. It provides different brainwaves values exhibits following kind of error messages: 1. Headset not Connected 2. Noise or Poor Signal
Resources	**Hardware:** Brainwaves Headset **Software:** Brain Wave Stimulator v1.0, Neurosky Brainwave SDK v1.0.
Processing	Headset receives 8 different brainwaves from 5 sensors spread

	all over the head of the user and transmits the data to the application via Bluetooth. (Refer Section 6.2).
Data	Theta value, Delta value, Alpha 1value, Alpha 2 value, Beta 1 value, Beta 2 value, Gamma 1 value, Gamma 2 value

4.3.3 Process DATA

This component has 3 sub-components:

4.3.3.4 Input Data

Identification	*Name:* Input Data *Location:* Application Logic layer of the system architecture
Type	Sub-component
Purpose	Following functional requirements are fulfilled by this sub-component: **REQ-1:** If the user's data does not exist in the Files, application should show the user should be prompted to enter his/her data for new entry in authentication Files. **REQ-2:** The application should assign a unique ID to every user. **REQ-3:** The system should be able to change the information of the selected user which was recorded in the Files at the time of addition of user.

	REQ-4: The system should be able to delete all the information of the user in the Files. **REQ-5:** The system should be able to guess the alphabet that the user is thinking and show it in the separate screen.
Function	The function of this sub-component is to manage user's data i.e. add, edit, delete, select or show user information. For showing user information, it takes input from *Data comparison* (refer Section 3.5) and shows output on the *Application UI* component (refer Section 3.1). For adding, editing and deleting user, it takes input from *Application UI* and transfers the data to the *Data Comparison*. It generates a query according to user input and sends it to *Data Comparison*.
Subordinates	It has two subordinates, *update data* and *retrieve data*. Update data performs functions like add, delete and edit and satisfy these functional requirements: REQ-1, REQ-2, REQ-3 and REQ-4. Whereas retrieve data will satisfy these requirements: REQ-2, REQ-3 and REQ-4 and REQ-5.
Dependencies	This dependent is dependent on *Data Comparison* (refer Section 3.5). Component 3.1 'Application UI' depends upon the

	successful execution of this component. Some of the functions of this component are: 'addInfo()', 'editInfo()', 'deleteInfo()' etc. and these functions interact with other components with the functions like 'OnDataRecieved()', 'DisplayOutput() ' , 'deleteUserInfo()' , 'editUserInfo()' etc.
Interfaces	User information is fetched from the User File and displayed onto screen. It provides external interface to component 3.1 'Application UI' in form of data which it gets from the Files Whereas it provides external interfaces to component 3.5 'Data Comparison'. Error Messages: 1. No record found. 2. Invalid Entry.
Resources	Hardware: RAM and Processor of the system will be utilized. Software: java core libraries
Processing	The component handles the organization of user data in the application. It takes input in form of brainwaves or other mouse events from other components (refer fig 2.1.1) and generate a query according to the user intent. (Refer

	section 6.5).
Data	Brainwave Packet

4.3.3.5 Processing Data

Identification	*Name:* Processing Data *Location:* Application Logic layer of the system architecture
Type	Sub-Component
Purpose	Following functional requirements are fulfilled by this sub-component: **REQ-1: T**he application should be able to recognize registered user correctly. **REQ-2:** The application should be able to store the calibration data. **REQ-3:** The application should be able to match the new data with calibration data and perform query accordingly.
Function	The function of this sub-component is to manage the user's brainwaves activity which includes the calibration data, fresh data and compare by using the Data Comparison module and show the results to the output screen by Application GUI.

Subordinates	It has two subordinates, UI interaction and data handling. UI interaction will display a message to think about a particular letter for successful calibration. Data handling will get the data of brainwaves of the user and store the data into Files .
Dependencies	This subcomponent provides service to the component 3.1 ‘Application UI’ i.e. these components depends on this component. The functions of this component which will interact with other components are: display_output(), perform_callibration() and store_data() etc.
Interfaces	It provides external interface to Application GUI component.
Resources	Hardware: RAM and Processor of the system will be utilized. Software: java core libraries
Processing	Take input from other components (refer Fig 2.1.1) in form of mouse events and brainwaves and then navigates user to selected option.
Data	Brainwave Packet

4.3.4 Data Comparison

Identification	*Name:* Data Comparison *Location:* Files layer of the system architecture
Type	Files component
Purpose	Following functional requirements are fulfilled by this sub-component: **REQ-1:** Application should be able to notify invalid username and password if it is not found in Files. **REQ-2:** If the invalid username or invalid password is entered, application should not load next screen and generate an error message. **REQ-3:** If both username and password are valid, application should load next screen. **REQ-4:** If the user has not calibrated his data then the system should prompt to the user to calibrate the system according to the user. **REQ-5:** The system should be able to change the information of the selected user which was recorded in the Files at the time of addition of user. **REQ-6:** The system should be able to delete all the information of the user in the Files.

	REQ-7: This component should compare data accurately and learn if the compared result is wrong
Function	This component handles Files transactions i.e.; add, update, delete and select statements. It provides a bridge between the application and the Files, i.e.; it can take input from the *Process Data* component (refer Section 3.3.1 and 3.3.2) and update the Files. Also, it can fetch data from Files and send it to the *Process Data* component.
Subordinates	It has 2 subordinates; *set* and *get*. *Set* does the modifying part of the Files and it will fulfill REQ-1, REQ-2, REQ-3, and REQ-4. *Get* does the fetching part from the Files and fulfills REQ-1, REQ-2, REQ-3, REQ-4, REQ-5, REQ-6 and REQ-7 of the functional requirements.
Dependencies	It depends upon data from the headset. *Process Data* component depends upon the successful execution of this component. All the functions related to saving, getting and displaying directly or indirectly interacts with this component. Some of these functions are: *getInfo()*, *setInfo()*, *DisplayOutput()*, *DeleteUserInfo(int)* etc.

Interfaces	Files will be maintained through file Handling in which all the data will be saved. It provides external interface to *Process Data* (refer Section 3.3) in form of service of data management this component offers to it.
Resources	**Hardware:** RAM, Processor **Software:** java core Libraries
Processing	Transaction is performed in order to retrieve data from the Files. The component receives a query in form of an input brainwave from other components (Refer fig 2.1.1). The query is then executed and transaction is performed either to retrieve and compare data and guess the alphabet which user has thought from Files or to update the user calibration data.
Data	Brainwave Packet

4.4 Reuse and Relationship to other products

This is not based on any previous systems neither it's an extension of any other applications at any level. But it can be evolved into a bigger and more complex system with more features and functionality The practical usage of the system can be increased by monitoring and recording brain activity of the patient via Electroencephalography (EEG), during the duration he/she is using the system to provide a better and more accurate knowledge about person's mental state, so that further evaluation can be based upon that.

The application can also be enhanced to further include more activities to exercise more areas of the brain. It has the same difficulty level on all exercises for all patients with different mental levels. Introducing different difficulty levels for patients (depending on the extent of their mental retardness) may yield results that are more beneficial.

4.5 Design and tradeoffs

The Headset-based FTT (refer Section 1.3) is an interactive application which requires multiple types of user interface. Developing such systems require thorough consideration on the design factors as it might result in complexity problem. A poorly-designed application results in a system consuming more resources with very little efficiency and a slower response time which directly affects the experience of the target user (refer Section 1.3). Besides this, poor designs make testing and maintenance activities difficult.

The application is a context-aware pervasive system (refer Section 1.3). Interface of the system is distinct from the application logic. Layered architecture is used to isolate application logic from the user interface. It can be modeled using **Multitier Layered Architecture** consisting of three layers i.e.; presentation, application logic and Files. Presentation layer corresponds to elements of the user interface such as text, checkbox item etc., and application logic layer controls the communication of data between the presentation and the Files layer, and is the part where the main logic, user actions and working of the system is defined. In general, it controls the complete behavior of the system, while the Files layer is responsible for handling and storing of the processed data.

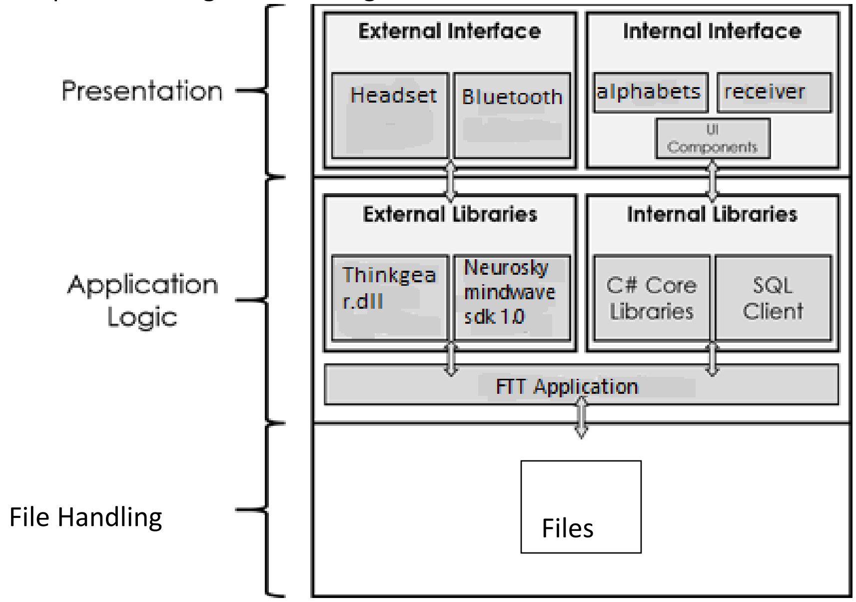


Fig. 5.1: *Three-tier Layered Architecture for FTT*

Presentation layer consists of hardware (Headset sensor and Bluetooth) and user interfaces of the application, with which the users will be interacting. All brain-activity feeds, objects, tables, buttons and form elements (textbox, radio buttons etc.) used in the application will come under view.

The interface of the application will be decorated in Windows Presentation Foundation (WPF) all the text, picture-boxes, buttons will be stylized in it.

The data from environment can be sensed into the system using the Headset sensor present in the presentation layer.

Application logic layer *(aka Business Logic)* includes all the algorithms that will be implemented, handles whatever action is done on the presentation layer. It will communicate with and process the information gathered from the presentation and the Files layer. It provides a communication channel between both these layers and also controls the data flow between them. It also controls the information that is to be displayed and the data that is to be stored.

This layer can be further categorized into two:

1. **External libraries will take input from the external interfaces:** External interfaces consist of the Headset sensor and it's Bluetooth, while the *NeuroskyMindwave SDK v1.0* external library is used. *NeuroskyMindwave* SDK provides capabilities (refer Section 1.3) to developers to build applications.

2. **Internal libraries will take input from the internal interfaces:** Internal interfaces constitute of WPF and other UI Components, while the internal libraries consist of java core libraries. WPF provides developer with a unified programming model for building rich Windows smart client user experiences that incorporate UI. The appearance of application's interface is designed in WPF, while the working and back-end functionality is defined in java

.

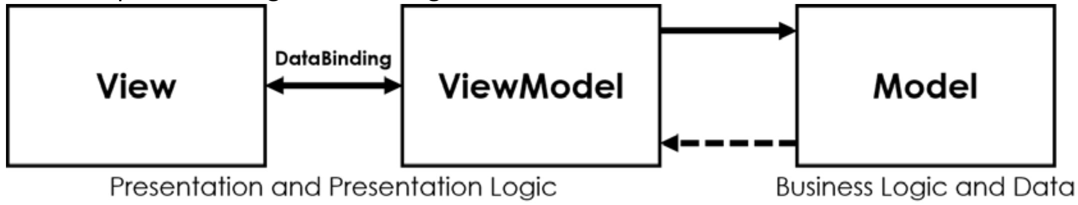


Fig. 5.2:*MVVM Pattern*

MVVM facilitates a clear separation of the development of graphical user interface (View) from the development of the application logic (Model). The ViewModel is a value converter i.e.; responsible for exposing the data objects from the model in such a way that those objects are easily managed and consumed. In this way, the ViewModel is more model than View, and handles most of it not all of the view's display logic. It also implements a mediator pattern (refer Section 4.3) organizing access to the back-end logic around set of use cases supported by the view (refer Section 2.2)

Chapter 5: Project Analysis and Evaluation

5.1Testing

Introduction

To ensure quality of the product, testing is conducted. Accuracy of functions performed by From Thinking to Texthas to be tested and maintained to improve quality of software. Software testing techniques and results obtained are discussed in the coming sections.

Testing Levels

Separate modules are developed to provide different functionalities of From Thinking to Text. All of these modules are tested at different levels in their development and post-integration process. Different levels at which From Thinking to Text has been tested, and results obtained are described in this section.

Unit Testing

Unit testing involves the testing of each module at the completion and at times, during the very course of development of the module. It is a testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. From Thinking to Text app has also been passed through unit testing process, wherein the various units have been tested in accordance with the anticipated output of each unit.

Integration Testing

From Thinking to Text different modules which were developed and tested independently were also tested during integration to ensure system stability. Integration testing helped in ensuring that different modules when combined give complete functionality and nothing is missed or some functionality doesn't give error when integrated with other modules. Integration testing gave us more than 90% results ensuring that most modules were integrated with others as well as compatible. This shows that errors were minimized during integration testing.

System Testing

System testing was performed at the end of development and integration of From Thinking to Text. Complete system was tested using sample data. Almost all of the test cases were successful ensuring that most of errors and bugs in the system were removed and system was stable enough to perform optimally.

Summary

Testing not only maintains the software quality but also improves overall usability of the project. At different stages of development suitable testing techniques were used to ensure product works accurately and efficiently. All errors detected during testing were removed and the test cases were prepared and made part of this document for the future compliance.

	Main Page Button
Test Case Number	6
Description	Detecting O
Preconditions	Application should be open
Input	Brain Waves values which are read from the file
Steps	After getting the Brain Waves values from file it directly goes to Detecting Alphabets
Expected output	Three Tests for Detecting O
Results	Three Tests in which O was detected

	History Button
Test Case Number	7
Description	Detecting A
Preconditions	Application should be open
Input	Brain Waves values which are read from the file
Steps	After getting the Brain Waves values from file it directly goes to Detecting Alphabets
Expected output	Three Tests for Detecting A
Results	Three Tests in which A was detected

	About Button
Test Case Number	8
Description	Detecting Z
Preconditions	Application should be open
Input	Brain Waves values which are read from the file
Steps	After getting the Brain Waves values from file it directly goes to Detecting Alphabets
Expected output	Three Tests for Detecting Z
Results	Three Tests in which Z was detected

	Home Button
Test Case Number	9
Description	Writing to File
Preconditions	Application must be open
Input	Brain Waves values which we get from headset
Steps	After getting the Brain Waves values from headset it directly goes to Writing those values in file
Expected output	It should move to next step of detecting the alphabet
Results	Test to check whether the program flow moves forward or not

Test Case Name	Enter Query
Test Case Number	10
Description	Reading the Brain Waves values from the file
Preconditions	Application must be open
Input	Written file
Steps	When file is written it moves directly to reading the values form the file.
Expected output	File is read and those values are put in classifier for classification so the program moves forward
Results	Test for checking that

Test Case Number	11
Description	Displaying the Output
Preconditions	Application must be open
Input	The result of the ANN classification
Steps	The program will itself show the Result once it gets the result from ANN
Expected output	The alphabet which the user was thinking
Results	Test for whether we are getting the Alphabet in Output

Performance Analysis:

Id	Name	Age	Expected Result	Actual Result	Attention Level
1	Malik Dilshad	23	A	A	70%
			N	N	65%
			T	O	35%
			Z	N	20%
			O	O	76%
2	ShahidNazir	18	A	N	40%
			N	N	66%
			T	T	68%
			Z	Z	72%
			O	O	71%
3	Muhammad Umer	22	Z	Z	65%
			A	T	5%
			O	O	60%
			N	N	72%
			T	T	50%
4	RijaFidaUllahTipu	23	N	N	73.5%
			A	A	78%
			Z	Z	53%
			O	O	88%
			T	N	90%
5	Hashim Ehsaan	22	T	T	69.9%
			A	A	68%
			N	O	20%
			Z	Z	80%
			O	O	705
6	SaqlainHaider	22	O	O	72%
			T	T	66%
			A	A	77%
			Z	N	39%
			N	T	33%
7	Aisha Saeed	21	A	A	65%
			Z	Z	78%

			O	N	10%
			N	N	89%
			T	A	17%
8	AsnaSohail Zaman	23	A	T	9%
			Z	Z	67%
			O	O	59%
			N	N	67%
			T	T	77%
9	Nasir Hussain	23	A	A	66%
			T	T	76%
			Z	A	33%
			N	N	69%
			O	O	76%
10	AzeemQamar	22	A	A	65%
			T	T	66%
			Z	Z	72%
			N	N	71%
			O	N	29%

Chapter 6: Future work

The project developed could then be used as a basis for further work in the field of medical science to help paralyzed patients communicate through brainwaves. One can increase the number of alphabets which are being recognized by the program to complete 26 alphabets and then we will have the ability to communicate just by thinking in our brain. If one could detect from a number of test cases the accurate letter which the person is thinking with the help of Neural networks, then who knows one day due to technological advancement one could figure out the whole thinking process of the human mind and that could revolutionize the medical fields.

Chapter 7: Conclusion

Appendix A: Proposal

Brief Description of the Project:

The aim of theirs project is to develop an application for translating thinking to text using brainwaves. Brain waves will be captured using the headset. Then we will preprocess the brain waves to reduce noise by enough to get a good signal into the computer, where we will process the data a bit more. As we will be using our computer's sound card to get the data in, we have to cut noise enough that the signal with noise does not spike above an amount of certain negative and positive voltage which would be the point where the sound card clips the data off. After pre-processing, brainwaves will be analyzed and finally interpret these signals into code and get meaningful result out of them.

Scope of Work:

The scope of the project will be to achieve alphabet recognition and have it written down by a text editor in computer through the use of a special headset that interprets brainwaves into electrical signals. These electrical signals would then be interpreted to code in order to achieve this alphabet recognition. We will be recognizing 5 alphabets.

Application /End Goal Objective:

Through this achieved recognition of alphabets the project can serve to become a basis for further development to speech recognition which will serve the paralyzed and otherwise completely handicapped people to convey their inner most thoughts.

Pseudo code For components

For Application UI

```
if Device is Successfully Connected

begin

            Show Welcome_Message

            Show Validating User Message

            If new User found

              Begin

                Add New User to Files

                  Calibrating the user's data

                 end

             else

            begin

                  Gather User's Brainwaves Activity

                 Go to Comparison Function
```

```
                    Show Output

                       end

end

else

      print 'Device not Connected'
```

FOR Headset Sensor

```
begin

      Connect_Sensor

      Gather_Data

      Send Data to Application

end
```

For Comparison

```
if Headset_Sensor is Running

begin

      Get values of Brainwaves from headset

      Pass the values to training algorithm

      Store the fresh values into the Files
```

```
    Compare the fresh values with the old values

end

else

    print 'No Headset Found'
```

For User Management

```
begin

    Check the User in Files

    if user selects exists

        then gather new data

    if user does not exist

        then create new user and add it to Files

            gather data for user's callibration

             if user selects edit option then ask for user input

             if user selects delete option then delete user from Files

end
```

For Data comparison

```
begin

        if user_thinks

                execute user_comparison_query and return data

        if user_update_data

                execute user_query and store data into Files

end
```

Appendix B:

Hardware requirements:

Headset: NeuroSky Headset to extract brains activity

Computer: To install software to visualize waves.

Bluetooth Dongle: Bluetooth is required for the communication of headset and the Software install in the computer for visualizing waves.

Glossary

NeuroSky: **NeuroSky, Inc.** is a manufacturer of Brain-Computer Interface (BCI) technologies for consumer product applications, which was founded in 2004 in Silicon Valley, California .The company adapts electroencephalography (EEG) and electromyography(EMG) technology to fit a consumer market within a number of fields such as entertainment (toys and games),education and health.

Electroencephalogram: An **electroencephalogram** (EEG) is a test that detects electrical activity in your brain using small, flat metal discs (electrodes) attached to your scalp. Your brain cells communicate via electrical impulses and are active all the time, even when you're asleep. This activity shows up as wavy lines on an EEG recording.

Brain Waves: An electrical impulse in the brain.

Appendix C: User Manual

Intended Audience:

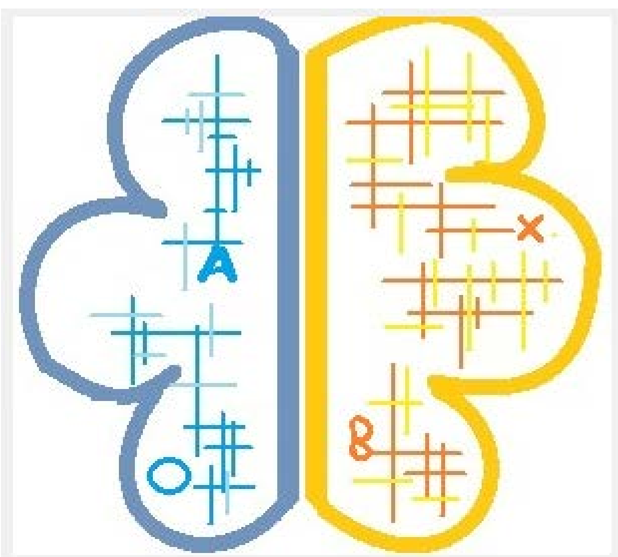
The audience of this document is everyone. This user manual is made to help and guide the users of this product for using the product.

Introduction:

From Thinking to Text(FTT) is a software program which uses Neurosky headset to gather the brainwaves of the user. Once the brainwaves data is collected by the headset and sent to the computer then FTT will tell the user which alphabet he/she was thinking. The program is capable to detect 5 alphabets which are A, N, O, Z, T, if the user thinks any one of the alphabets then the program will tell the user the alphabet which was thought by the user.

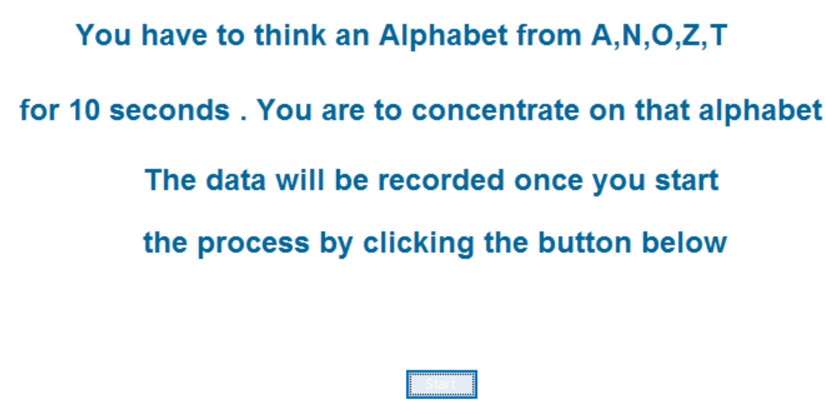
Functions:

1st Screen:



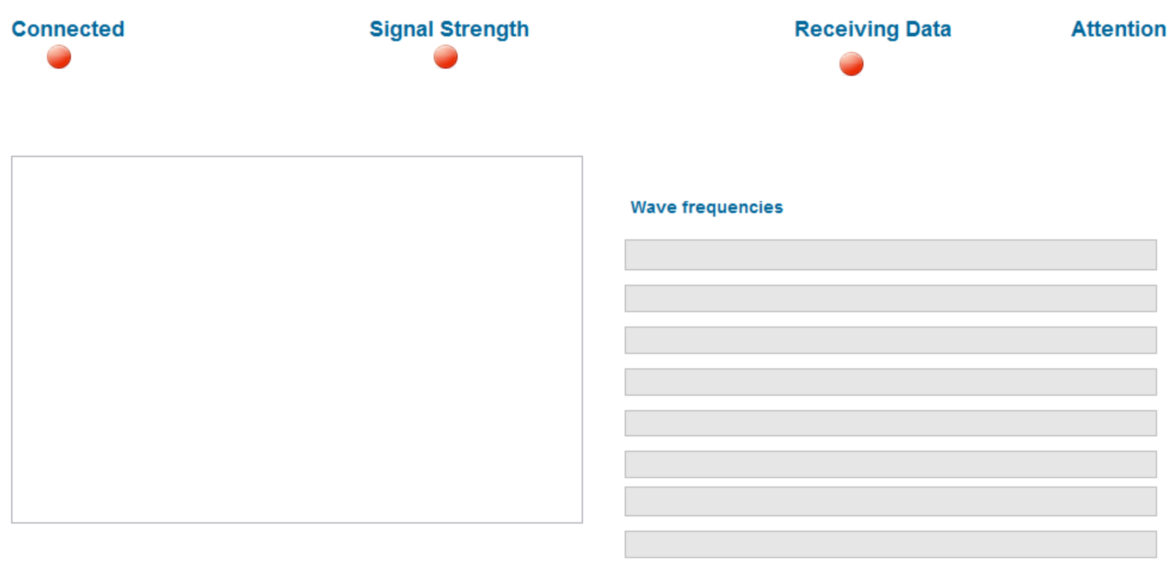
This is the starting screen of the product it comes for 3-4 seconds at most. The user needs not to do anything because after some time the program will move towards the next screen automatically.

2nd Screen:



This is the second screen known as the instruction screen. The user needs to read the instructions and whenever the user is ready he/she should click on start button on the bottom of this screen to move towards the next screen.

3rd Screen:



This is the last screen of the program. The Connected shows weather the headset is connected with the computer of the user, Once the headset connects the red circle turns into green circle. The signal strength shows that the signals received by the computer are of good quality or not. If the signals received by the computer are of good quality, then the red circle turns green. Similar is the case with the receiving data, once the program starts receiving the brainwaves data from the headset the red icon turns green. The attention shows the level of the attention of the user in a numeric value. The eight bars show the 8 different types of brainwaves and their intensities at that moment. The 8 different brainwaves are:

1- High Alpha
2- Low Alpha
3- High Beta
4- Low Beta
5- High Gamma
6- Low Gamma
7- Delta
8- Theta

The alphabet that the user thought will be shown in the box on the bottom left of the screen.

Bibliography

Mayo Clinic researchers show brain waves can 'write' on a computer in early tests

http://www.eurekalert.org/pub_releases/2009-12/mc-mcr120409.php

www.sciencedaily.com/releases/2009/12/091206181911.html.

1. **Interpreting Brain Waves**

http://well.blogs.nytimes.com/2013/01/28/interpreting-brain-waves/

2. **As scientists discover how to 'translate' brainwaves into words... Could a machine read your innermost thoughts?**

   http://www.dailymail.co.uk/sciencetech/article-2095214/As-scientists-discover-translate-brainwaves-words--Could-machine-read-innermost-thoughts.html#ixzz3awJN9w8B

3. **Code to read EEG data via a soundcard or Arduino.**

https://github.com/cah6/EEG

4. **DIY EEG (andECG) circuit processing**

http://www.instructables.com/id/DIY-EEG-and-ECG-Circuit/step10/Processing-the-Data/

5. WHAT ARE BRAINWAVES?

http://www.brainworksneurotherapy.com/what-are-brainwaves

http://www.eurekalert.org/pub_releases/2009-12/mc-mcr120409.php

https://www.pantechsolutions.net/blog/brain-keyboard-using-neurosky-mindwave-mobile-and-java/

From thinking to Text -Alphabet Recognition Through Brainwaves