# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018, Karnataka, India

**PROJECT REPORT**
**on**

## "BCI-Operated Media Player"

Submitted in partial fulfillment of the requirements for the VIII Semester

**Bachelor of Engineering**
**IN**
**COMPUTER SCIENCE AND ENGINEERING**

**For the Academic year**
**2015-2016**
**BY**


| | |
|---|---|
| **Bhoomika Agarwal** | **1PE12CS033** |
| **Sreekumar T.** | **1PE12CS159** |
| **Sowmya R.** | **1PE13CS426** |
| **Ramesh M.** | **1PE13CS416** |

**Under the Guidance of**
**Sarawathi R. Punagin**
**Assistant Professor, Dept. of CSE**
**PESIT-BSC, Bengaluru-560100**




**Department of Computer Science and Engineering**
**PESIT BANGALORE SOUTH CAMPUS**
**Hosur Road, Bengaluru -560100**

# PESIT BANGALORE SOUTH CAMPUS

**Hosur Road, Bengaluru -560100**

## Department of Computer Science and Engineering


# *CERTIFICATE*

*Certified that the project work entitled* ***"BCI-Operated Media Player"*** *is a bonafide work carried out by* ***Bhoomika Agarwal, Sreekumar T., Sowmya R.*** *and* ***Ramesh M.*** *bearing USN* ***1PE12CS033***, ***1PE12CS159, 1PE13CS426*** *and* ***1PE13CS416*** *respectively, students of* ***PESIT Bangalore South Campus*** *in partial fulfillment for the award of* ***Bachelor of Engineering*** *in* ***Computer Science and Engineering*** *of the* ***Visvesvaraya Technological University****, Belagavi during the year 2015-2016. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated and the project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.*

***Signatures:***

| | | |
|---|---|---|
| Project Guide<br>**Ms. Saraswathi R. Punagin**<br>Asst. Professor, Dept. of CSE<br>PESIT-BSC, Bengaluru | Head of Department<br>**Mr. Sandesh B J**<br>Associate Professor,Dept.of CSE<br>PESIT-BSC, Bengaluru | Director/Principal<br>**Dr. Suryaprasad J**<br>PESIT-BSC,Bengaluru |

***External Viva***

| **Name of the Examiners** | **Signature with date** |
|---|---|
| *1.* _____ | _____ |
| *2.* _____ | _____ |

# ACKNOWLEDGEMENT

We would like to acknowledge all those that contributed to make this project a success.

We are indebted to our Guide **Ms. Saraswathi R. Punagin**, Assistant Professor, Department of Computer Science and Engineering, PESIT - Bangalore South Campus, who has not only coordinated our work but also given suggestions from time to time.

We would also like to thank **Dr. Snehanshu Saha**, Professor, of Computer Science and Engineering, PESIT - Bangalore South Campus, for his invaluable help and guidance throughout the course of this project.

We are also extremely grateful to our Project Co-ordinators, **Dr. Snehanshu Saha, Mrs. Sudha Y and Ms. Vandana M L**, Professor and Assistant Professors, Department of Computer Science and Engineering, PESIT - Bangalore South Campus, for their constant support and advice throughout the course of preparation of this document.

We are greatly thankful to **Mr. Sandesh B J**, Associate Prof and Head, Department of Computer Science and Engineering, PESIT - Bangalore South Campus, for his able guidance, regular source of encouragement and assistance throughout this project.

We would like to express our immense gratitude to **Dr. Suryaprasad J,** Director and Principal, PESIT - Bangalore South Campus, for providing us with excellent infrastructure to complete our project work.

We gratefully acknowledge the help lent out to us by all faculty members of the Department of Computer Science and Engineering, PESIT Bangalore South Campus, at all difficult times. We would also take this opportunity to thank our college management for the facilities provided during the course of the project. Furthermore, we acknowledge the support and feedback of our parents and friends.

**Bhoomika Agarwal**
**Sreekumar T.**
**Sowmya R.**
**Ramesh M.**

# ABSTRACT

BCI-Operated Media Player is a mp3 player that can be controlled through a brain-computer interface (BCI). It has been designed for patients who are severely impaired and partially or completely locked in. They have lost all motor ability but have active brains. In our media player, we have created four controls that have been encoded as a combination of left-hand and right-hand movements. The OpenBCI device reads the input in the form of EEG signals from the patient's brain and then uses the same to control the media player. It is easily scalable to multiple controls through a binary-like combination.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER - 1

# INTRODUCTION

# CHAPTER 1

# INTRODUCTION

A Brain Computer Interface (BCI) system is a communication system where a person has the ability to communicate with a computer through his or her brain signals rather than using the peripheral nerves and muscles. A BCI system effectively allows for the conversion of patterns of electrical brain activity into commands to control specific equipment.

BCI technology relies on the acquisition of electrical signals generated by billions of neurons inside the brain. The electrical fluctuations that arise from these neurons reach the scalp where they can be detected and recorded by means of non-invasive metal electrodes through a process known as electroencephalography (EEG).

In a BCI system EEG data is recorded from the human subject and this is then processed to extract reliable features which can then be mapped into computer based commands such as moving a cursor on a screen or selecting from sets of letters.

.

## 1.1 Purpose

Historically, assistive technologies(AT) have relied on the person being able to manoeuvre at least one part of their body. For example, an Augmented Communication Device may require them to press buttons on a keyboard that has pre-designated questions, statements or responses. These devices can be adapted in order for the buttons to be pressed with a finger, a toe, or a metal-pointer attached to their head.

The issue is with people with severe disabilities such as locked-in syndrome who aren't capable of motor function other than eye movements.

Technology in the form of the brain computer interface (BCI) provides hope for these and many other people because we no longer have to imagine being able to use our thoughts to control a wheelchair or a communication device. The BCI Operated Media Player designed by us promises to enhance life immensely for those with severe disabilities such as locked-in syndrome.

## 1.2 Scope

The BCI Operated Media Player has been designed keeping in mind the needs of people who are partially or completely locked in. Assistive technology fails in the case of such severe disability due to lack of or limited motor function.

## 1.3 Definitions, Acronyms and Abbreviations

**EEG** : electroencephalography

**LIS** : Locked-in-Syndrome

**BCI** : Brain Computer Interface

**AT** : Assistive Technology

## 1.4 Literature Survey

In their paper titled **"Mu and Beta Rhythm Topographies During Motor Imagery and Actual Movements",** Dennis J. McFarland, Laurie A. Miner, Theresa M. Vaughan, and Jonathan R. Wolpaw compared the effects of motor imagery on mu and beta rhythms, and compared it with that of actual movement. They found that the the **mu and beta rhythm topographies are similar for movement and imagery**.

For both movement and imagery, mu desynchronization is greatest at CP3 and beta desynchronization is greatest at CZ.

Dandan Huang, Peter Lin, Ding-Yu Fei, Xuedong Chen and Ou Bai found that the reason for the fact that motor imagery has less robust performance than physical movement might be that **motor imagery is not a natural behaviour** and thus requires more effort than performing physical movement. Besides, compared with physical movement, there is no neural feedback in motor imagery which may exhibit less activity (ERS) in the motor cortex and result in a lower signal to noise ratio. Considering that motor imagery is more meaningful for BCI application, training may be needed to enhance the involvement of subjects with motor imagery. These findings have been published in **"Decoding human motor activity from EEG single trials for a discrete two-dimensional cursor control"**

In their research paper titled **"Patients with ALS can use sensorimotor rhythms to operate a brain-computer interface"**, Kubler et. al showed that using motor imagery based BCI can improve the quality of life in patients with severe motor disabilities. This is done by establishing a means of communication that is independent on muscle control via. In their experiment, our **people severely disabled by ALS learned to operate a BCI with EEG rhythms** recorded over sensorimotor cortex.

Ho¨hne J, Holz E, Staiger-Sa¨lzer P, Mu¨ ller K-R, Ku¨bler A, et al. showed that **non-invasive BCI using motor imagery can be used to overcome severe motor impairments in patients that are locked-in and almost completely locked-in** in their paper titled **"Motor Imagery for Severely Motor-Impaired Patients: Evidence for Brain-Computer Interfacing as Superior Control Solution".** The researchers used state of the art machine learning methods to enable these patients to use BCI in six or less sessions. The BCI was used to operate a gaming application. 3 out of 4 severely disabled patients were able to gain significant control via motor imagery.

In the paper , **"An EEG-Based BCI System for 2-D Cursor Control by Combining Mu/Beta Rhythm and P300 Potential"**, Yuanqing Li, Jinyi Long, Tianyou Yu, Zhuliang Yu, Chuanchu Wang, Haihong Zhang, and Cuntai Guan have presented a new BCI and its implementation for 2-D cursor-control by **combining the P300 potential and motor imagery**. Two almost independent signals have been obtained for controlling two degrees of movements of a cursor simultaneously. In particular, a motor imagery detection mechanism and a P300 potential detection mechanism were devised and integrated with a specially designed GUI.

In the study titled **"Quadcopter control in three-dimensional space using a non-invasive motor imagery based brain-computer interface"**, Karl LaFleur, Kaitlin Cassady, Alexander Doud, Kaleb Shades, Eitan Rogin, and Bin He performed an experimental investigation where 5 human subjects were trained to demonstrate their ability to control a robotic AR Drone quadcopter in a three-dimensional physical space by means of a motor imagery EEG brain-computer interface. Through the use of the BCI system, **subjects were able to quickly, accurately, and continuously pursue a series of foam ring targets and pass through them in a real-world environment using only their "thoughts".**

Jason Sleight, Preeti Pillai, Shiwali Mohan performed experiments to show that the using EEG signals to perform classification of executed and imagined motor movements can be performed effectively through the use of **sophisticated machine learning techniques such as ICA and SVM**, despite the close similarities between the two signals. However, they found that results differ significantly across patients, suggesting that **BMI is highly specialized and must be uniquely determined for each individual patient**. Published in **"Classification of Executed and Imagined Motor Movement EEG Signals"**

In **"An ERD/ERS Analysis of the Relation between Human Arm and Robot Manipulator Movements"**, results show that in the case of movement execution and imagination, the ERD in the **contralateral motor area (C3) is more**

**intense than in the ipsilateral motor area (C4)** as expected. Ernesto Pablo Lana, Bruno Vilhena Adorno, Carlos Julio Tierra-Criollo state that this feature could be used for left and right arm movement discrimination, in the case of an implementation of a BMI using two robotic manipulators.

In their paper- "**Virtual Keyboard Controlled by Spontaneous EEG Activity**",B. Obermaier, G. R. Mueller and G. Pfurtscheller demonstrate another application of motor imagery. They test a virtual keyboard on severely impaired patients. The operation of the VK is based on an application of Graz BCI. The spelling consists of the selection of a letter using successive steps of isolation. Three patients were tested and an error-free letter selection is done in six trials. The experiments showed that **this device can be used as an alternate spelling device because it has a higher spelling rate**.

In their paper titled "**Motor Imagery Based On Wavelet Power Spectrum for a Brain Computer Interface"**, Javier Castillo-Garcia, Eduardo Caicedo , Berthil Borges Longo, Alan Floriano ,and Teodiano Bastos-Filho performed a study where **a BCI based on wavelet power spectrum using motor imagery** is implemented. Four motor tasks were used and the method proposed achieved success rate of 89% and ITR of 11.40 bit/min. The proposed method had statistical significance (p-value<0.05) and the size of effect was very high (>2.0) for sensitivity, specificity, accuracy and Kappa coefficient.

## 1.5 Existing System

One of the major issues faced by people with LIS is that they cannot use modern technology, especially to communicate with computers. They cannot access the computers through any means at present. While there are a few applications like Graz BCI and Virtual Gaming that allow interactions with computers, these are extremely application specific. They do not focus on the general use of computer systems.

The current systems use multiple stimuli and modes to scale up from one dimension to multiple dimensions. That is, different methods are used in parallel to measure signals in each dimension. This can be resource extensive and expensive.

.

## 1.6 Proposed System

The proposed system aims to help the people in locked-in-state to operate a media player by imagined movement, also known as motor imagery.

Research has shown that the electrical signals produced in the brain upon actual motor function are the same as those produced when the same motor function is imagined. This is the basis of motor imagery.

We use a combination of binary digits, wherein left hand movement maps to a '0' and right hand movement maps to a '1', thereby creating a language. These strings are then mapped to inputs to the computer system. This allows us to scale up to multiple dimensions without the requirement for additional hardware or techniques.

Existing systems use different paradigms such as motor imagery, P300, SSVEP for each dimension when the system is required to scale to more than one dimension. This results in additional cost in terms of hardware and processing. We reduce this cost by allowing scaling up to multiple dimensions using just a single paradigm such as motor imagery.

## 1.7 Statement of the Problem

Design a system that allows people with severe motor disability to communicate with computers in real time. This system should be able to scale up to multiple dimensions effectively.

## 1.8 Summary

This chapter gives the details about the existing problem and the purpose of our project. It also contains details about the system and the proposed system. It makes us aware about the different terminologies, definition and abbreviations used in the project.

# CHAPTER – 2

# SOFTWARE REQUIREMENTS SPECIFICATION

# CHAPTER – 2

# SOFTWARE REQUIREMENTS SPECIFICATION

The Software requirement specification is the official statement of what the system developers should implement. It should include both the user requirements for a system and a detailed specification of the system requirements. In some cases, the user and the system requirements may be integrated into single description. In other cases, the user requirements are defined in an introduction to the system requirement specification.

The requirements may document external interfaces, describe system functionality and performance, specify logical database requirements, design constraints, emergent system properties and quality characteristics.

## 2.1 Operating Environment

This project has been implemented in Python using the OpenBCI toolkit. It runs on the Windows environment.

### 2.1.1 Hardware Requirements

- Tools : OpenBCI toolkit
- System : Pentium IV 2.4 GHz.
- Hard Disk : 80 GB
- RAM : 1 GB

### 2.1.2 Software Requirements

- Operating system : Windows.
- Implementation : Python, Wyrm module
- Front End : Python, TK module
- Tool : Python compiler

## 2.2 Functional Requirements

Functional requirements are associated with specific functions, tasks or behaviors the system must support. The functional requirements address the quality characteristics of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements. Because non-functional requirements tend to be tested in terms of constraints on the results of tasks which are given as functional requirements (e.g., constraints on the speed or efficiency of a given task), a task-based functional requirements statement is a useful skeleton upon which to construct a complete requirements statement. That is the approach taken in this work. It can be helpful to think of non-functional requirements as adverbially related to tasks or functional requirements.

The client-side of the system will be an application with a user interface. The user can perform one of the following operations:

1. Play: plays the current music file

2. Stop: stops the current music file

3. Play Next: plays the next music file in the sequence.

4. Play Previous: plays the previous music file in the sequence.

The server-side system will hold the entire music data in a directory and must include all functionality to perform operations on this data, receive requests from the clients, evaluate and execute the specified command.

## 2.3 Non Functional Requirements

- Safety: The system ensures safety by reducing the time for controlling the media player.
- Correctness: Ensures the correctness in the system by triggering the correct function as per the requirement of the user
- Reliability- The system, including all hardware and software, should satisfactorily perform the task for which it was designed or intended, for a specified time and in a specified environment.
- Performance-The system must be able to mine the data, cluster the data and give the best result for the give input in very less time.
- Availability-The system should be available to the users all time possible with as less downtime possible.

## 2.4 User Characteristics

The main purpose of the system is to provide a means of communication with the computer system to the severely impaired. The users are patients who are partially or completely locked- in due to various kinds of disability. Assistive technology or devices fail in the case

of such patients and they have no method to communicate with the world around them- especially the computer system.

## 2.5 Applications of System

- Can be used as a proof of concept for software applications with multiple dimensions and functions
- Provides a tool for users to control a media player through their brain signals

## 2.6 Advantages of System

- Easily scalable
- High performance and reliability
- Simplified hardware and software
- Efficient means of communication with the computer system
- Does not require much training of the user before usage

## 2.7 Summary

This chapter gives the detail about the various hardware and software requirements i.e., details about the various hardware and software equipments used in this project. It also gives the details about the functional and non-functional requirements of our project. It provides information about the various characteristics of the user too.

# CHAPTER – 3

# SYSTEM DESIGN

# CHAPTER – 3

# SYSTEM DESIGN

## 3.1 Introduction

The purpose of this system design chapter is to add the necessary detail to the current project description to represent a suitable model for coding. The system design documentation presents the structure of the system such as the Architecture and Data Flow Diagram.

## 3.2 Development Strategy

The system is designed using 'The Waterfall model'. The waterfall model was the first structured approach to systems development. The waterfall model is just a time ordered list of activities to be performed to obtain an IT system.

The activities in waterfall model are:

1. System Analysis
2. System Design
3. Coding
4. Testing
5. Implementation
6. Maintenance

## 3.3 System Architecture

System architecture is a conceptual model that defines the structure, behavior and more views of the system.

An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure of the system which comprises system components, the externally visible properties of those components, the relationship between them, and provides a plan from which products can be produced, and systems can be developed, that will work together to implement the overall system.

The system is divided into 3 modules:

- Hardware module
- Processing module
- User interface and media player

**Hardware module**: Here, the EEG signals from the user's brain are collected and passed to the software by the OpenBCI toolkit

**Processing Module:** Here, EEG signals are processed and classified into left-hand or right-hand movements.

**User interface and media player:** Here, the user can see the output generated by the controls. The music is played from the playlist as per the controls input by the user.

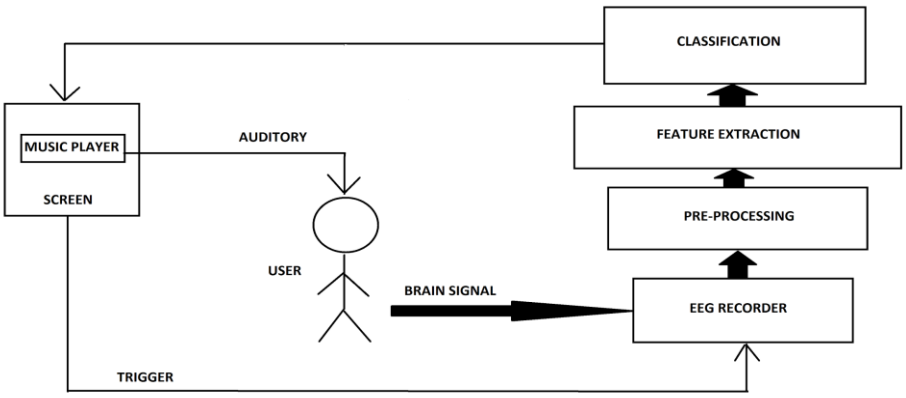


**Fig 3.3: System Architecture**

## 3.4 Data Flow Diagram

Data flow diagram (DFD) is a graphical representation of the flow of data that provides an overview of data flow in a system, transformations done on the data, files used and flow of results**.** It is also called as bubble chart which is used to represent a system in terms of the input data to the system. DFD is a good documentation aid which is understood by both programmer and nonprogrammers.

**3.4.1 LEVEL 0:** This diagram gives the outline of the system function when a user inputs a control. The control is in the form of imagined hand movement that is then mapped to the corresponding function in the media player. It shows the various steps that the data undergoes as part of the process before it is translated to the control.

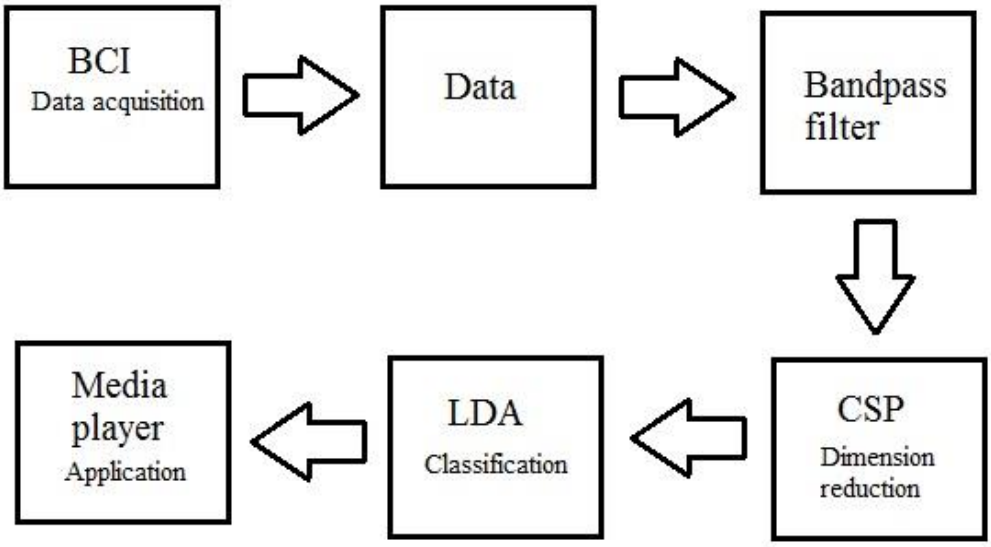


**Fig 3.4.1: Data flow diagram of system – Level 0**

**3.4.2 LEVEL 1:** This diagram gives the detailed dataflow of the system process that is happening when a user provides an input. This diagram shows the detailed processes and data flow that takes place in the BCI based media player. All the activities that take place in the system are shown below.

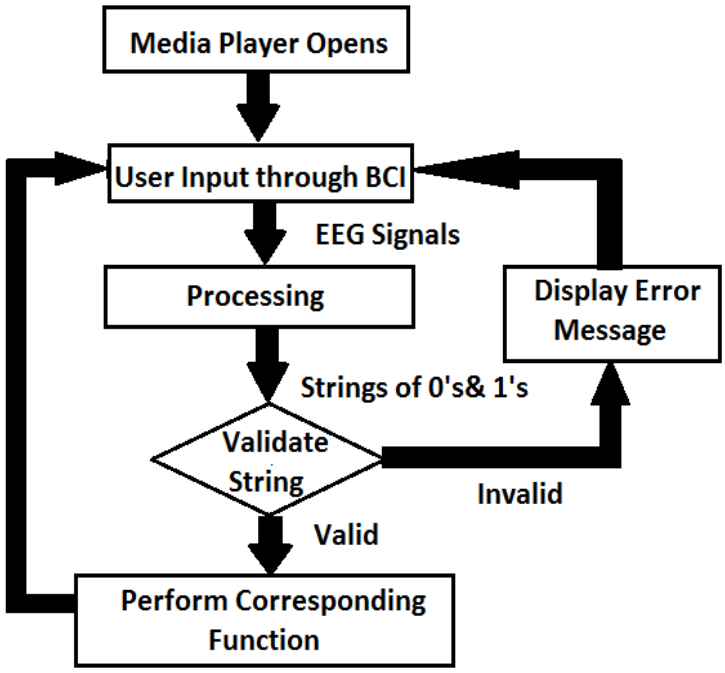


**Fig 3.4.2: Data flow diagram of system – Level 1**

## 3.5 Summary

This chapter provides details about the various functionality of our system. The system architecture and the dataflow diagram at different levels are also mentioned here. It provides a brief description about the architecture of our system. It also provides details about the goals and constraints of our system. The assumptions and dependencies of this system are also mentioned here.

# CHAPTER – 4

# DETAILED DESIGN

# CHAPTER - 4

# DETAILED DESIGN

The system design documentation presents the structure of the system such as the Activity diagram, Use Case diagram and Sequence diagram.

## 4.1 Activity Diagram

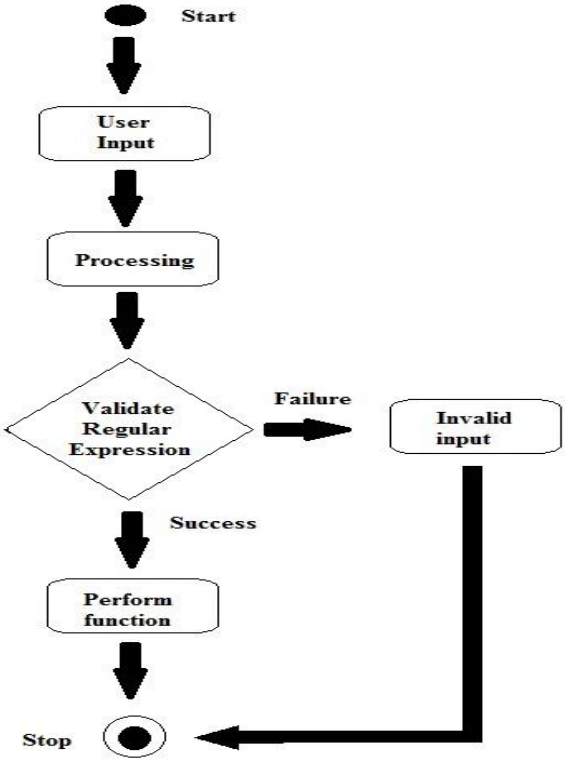


**Fig 4.1.1: Activity diagram of the system**

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

The system starts with the initialization of the media player and loading of a pre-loaded playlist (if any). Following that, the user enters input in the form of imagined hand movement. The input is acquired by the OpenBCI device and then processed and classified. A regular expression is used to validate the string of 0s and 1s. If the string matches the regular expression, it is then used to control the media player. Else, an error message is displayed to the user. The system then stops.

## 4.2 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-Case analysis.

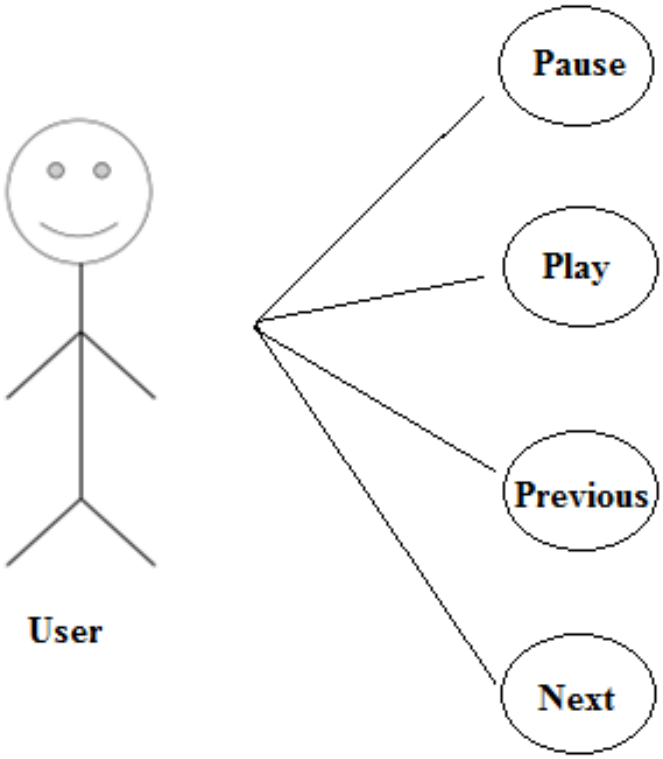


**Fig.4.2.1: Use Case diagram for the system**

The user can perform four functions:

1. Pause - Pause the song being currently played
2. Play – Play a song from the playlist
3. Previous – Move to the previous song in the playlist
4. Next – Move to the next song in the playlist

## 4.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. The sequence diagrams depicting the 3 modules of this system are shown below:

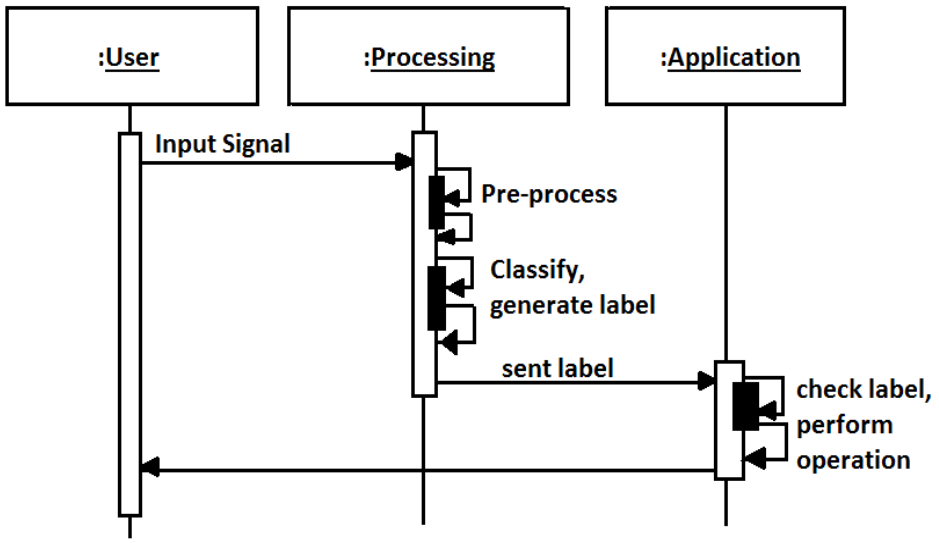


**Fig 4.3.1: Sequence diagram for the system**

The user first provides an input signal to the processing module. This module carries out pre-processing and classification to generate a label. This label is sent to the application. The application checks the label using a regular expression and performs the corresponding operation if it is valid.

## 4.4 Summary

This chapter summarizes the activity diagram, the use case diagram and the sequence diagram of the system. The 3 modules of the system and their communication are explained with the help of these diagrams here.

# CHAPTER – 5

# IMPLEMENTATION

# CHAPTER – 5

# IMPLEMENTATION

Implementation of any software is always preceded by important decisions regarding selection of the platform, the language used, etc. These decisions are often influenced by several factors such as the real environment in which the system works, the speed that is required, security concerns, other implementation specific details etc.

A software product implementation method is a blueprint to get users and/or organizations running with a specific software product. The method is a set of rules and views to cope with the most common issues that occur when implementing a software product business alignment from the organizational view and acceptance from the human view.

## 5.1 Programming Language Selection

### 5.1.1 Overview of Python:

**Python** is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard

library. Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems.

Since 2003, Python has consistently ranked in the top ten most popular programming languages as measured by the TIOBE Programming Community Index. As of January 2016, it is in the fifth position. It was ranked as Programming Language of the Year for the year 2007 and 2010. It is the third most popular language whose grammatical syntax is not predominantly based on C, e.g. C++, Objective-C.

Libraries like NumPy, SciPy and Matplotlib allow the effective use of Python in scientifc.Python has also been used in artificial intelligence tasks. As a scripting language with module architecture, simple syntax and rich text processing tools, Python is often used for natural language processing tasks. Many operating systems include Python as a standard component; the language ships with most Linux distributions.

## 5.2 Platform Selection

For our project, we have selected Windows platform which is discussed in the following section.

### 5.2.1 Windows

**Microsoft Windows** is a metafamily of graphical operating systems developed, marketed, and sold by Microsoft. Windows makes it easier to do things because of its graphical user interface (or GUI for short). It only needs a keyboard or a mouse to work. By clicking a few buttons on the screen, Windows helps keep your files safe, and easier to change and move. Almost 90% of desktop and laptop computers use Windows.

## 5.3 Libraries

For our project, we have used various python libraries which are discussed in the following section.

### 5.3.1 Wyrm

**Wyrm**, a Python toolbox for on-line BCI experiments and off-line BCI data analysis. It's a signal processing toolbox for BCI. The signal processing part of an online BCI system is responsible for translating the brain signals into actionable output signals by detecting certain patterns in the brain signals. In order to detect those patterns, the raw brain signals usually have to be preprocessed and specific features that represent those patterns best, have to be extracted and classified. The actual methods used to translate the raw brain signals to output signals differ highly from application to application, and a large part of BCI research is devoted to finding better methods or improving existing ones. Researchers are constantly looking for ways to improve the information transfer rate, classification accuracy, or the representation of the brain signals as feature vectors. Therefore, they spend a lot of time working with toolboxes that allow them to manipulate data, try out new methods, visualize different aspects of the data, etc.

Wyrm tries to cover both toolbox-aspects: Wyrm can be used as a toolbox for offline analysis and visualization of neurophysiological data, and in real-time settings, like an online BCI experiment. Wyrm implements dozens of different toolbox methods, which makes it applicable to a broad range of neuroscientific problems.

Since BCI researchers spend a lot of time with a toolbox, the wyrm toolbox was designed to encourage researchers to "dive" into their data and play with it: Its main data structure is very flexible yet easy to understand, and the toolbox methods follow a set of rules to keep the syntax and semantics consistent. Heavy use of unit testing for the toolbox methods gives users the confidence that the methods work as expected and the extensive documentation makes the toolbox easy learn and use. Wyrm also comes with example scripts for common BCI paradigms in online- and offline settings.

### 5.3.2 Numpy

**NumPy** is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high level mathematical functions to operate on these arrays.

The core functionality of NumPy is its "ndarray", for *n*-dimensional array, data structure. These arrays are strided views on memory. In contrast to Python's built-in list data structure, these arrays are homogeneously typed: all elements of a single array must be of the same type.

### 5.3.3 Scipy

**SciPy** is an open source Python library used by scientists, analysts, and engineers doing scientific computing and technical computing. Scipy contains modules for optimization, linear algebra, integration, interpolation, special functions, signal and image processing and other tasks common in science and engineering.

SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas and SymPy. There is an expanding set of scientific computing libraries that are being added to the NumPy stack every day. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Scilab. The NumPy stack is also sometimes referred to as the SciPy stack.

SciPy is also a family of conferences for users and developers of these tools: SciPy (in the United States), EuroSciPy (in Europe) and SciPy.in (in India). Enthought originated the SciPy conference in the United States and continues to sponsor many of the international conferences as well as host the SciPy website.

### 5.3.4 scikit-learn

**scikit-learn** is a free software machine learning library for the Python programming language. scikit-learn features various classification, regression and clustering algorithms including support for vector machines, random forests, gradient boosting, *k*-

means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

### 5.3.5 Tkinter

**Tkinter** is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI, and is included with the standard Microsoft Windows and Mac OS X install of Python.

The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

Python 2.7 and Python 3.1 incorporate the "themed Tk" ("ttk") functionality of Tk 8.5.This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of Tkinter).Tkinter is free software released under a Python license.

### 5.3.6 Mp3play

**Mp3play,** a simple interface for playing music from an MP3 file. The module is currently only supported for Windows python programs. It provides basic operations like load, play and stop the mp3 file.

## 5.4 Graphical User Interface

Graphical user interface is a type of user interface that allows users to interact with electronic devices using images rather than text commands.

For our project, the graphical user interface was required for the media player application. The graphical user interface has been fully developed using the Tkinter module in Python. The user interface was developed keeping in mind simplicity and ease of use.

There are two windows in our application. The graphical user interface main window is provided with four buttons which corresponds to the four operations; play, stop, play next, play previous. The media player also has the feature to display the current file being played. This is present in the main window. The second window displays all the .mp3 files loaded in the form of a playlist.

## 5.5 Summary

In this chapter we have discussed about the implementation. We have discussed each module individually. We have even discussed about the software environment and different technologies used.

# CHAPTER – 6

# TESTING

# CHAPTER - 6

# TESTING

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not.

## 6.1 Unit Testing

Unit testing involves design of test cases that validate that the internal program logic is functioning properly, and that the program inputs produce valid outputs. It is the testing of individual software units of the application. It is done after the completion of an individual unit.

Unit tests perform basic tests at component level. Unit tests ensure that each unique path of a business process forms accurately to the documented specifications and contains clearly defined inputs and expected results.

## 6.2 Test Cases

### 6.2.1 Unit Testing of Modules

**Unit test case 1: Test case for dataset formatting**

| | |
|---|---|
| Name of the test | Test Case for formatting the dataset. |
| Test Description | A test for checking the dataset is formatted as required. |
| Sample Input | A dataset file in matlab format. |
| Expected Output | Displays the dataset after converting to the required format. |
| Actual result/Remarks | As expected. |
| Passed (?) | Yes |

**Table 6.2.1**

**Unit test case 2: Test case for label generation**

| | |
|---|---|
| Name of the test | Test Case for label generation for the dataset. |
| Test Description | This test checks for successful generation of labels for the input dataset. |
| Sample Input | Training samples, testing samples, true label of test samples. |
| Expected Output | The LDA classifier is trained without much delay and the testing sample are classified with correct labels. |
| Actual result/Remarks | As expected. |
| Passed (?) | Yes. |

**Table 6.2.2**

**Unit test case 3: Test Case for validate the user input**

| | |
|---|---|
| Name of the test | Test Case to validate the user input |
| Test Description | This test validates the user input to the media player using a regular expression. |
| Sample Input | A string containing binary sequence. |
| Expected Output | The application validates the string using the regular expression and performs the operations in a sequence. |
| Actual result/Remarks | As expected. |
| Passed (?) | Yes. |

**Table 6.2.3**

# CHAPTER – 7

# SNAPSHOTS

# CHAPTER - 7

# SNAPSHOTS

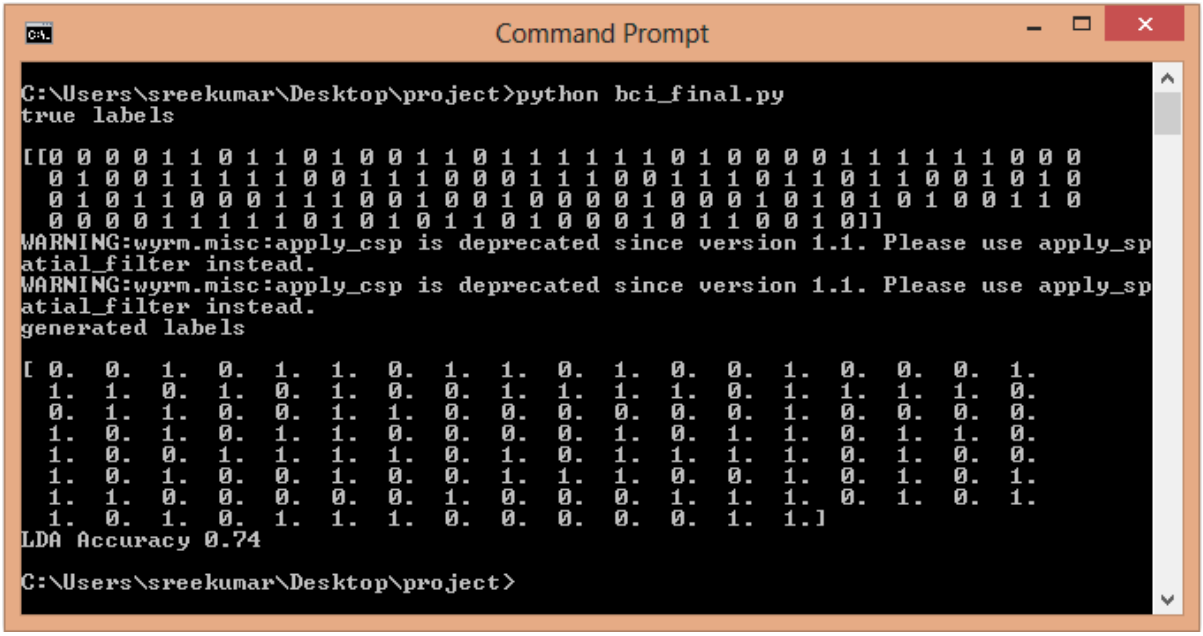


**Fig. 7.1 Label Generation**

The fig.7.1 shows the result obtained from the classifier for the input dataset.

As we see above, the true labels, for the test sample, which were already provided is displayed.

This is followed by the labels which were generated by our application for the same test samples.

These two arrays of labels are compared and the accuracy is determined to be 74%.

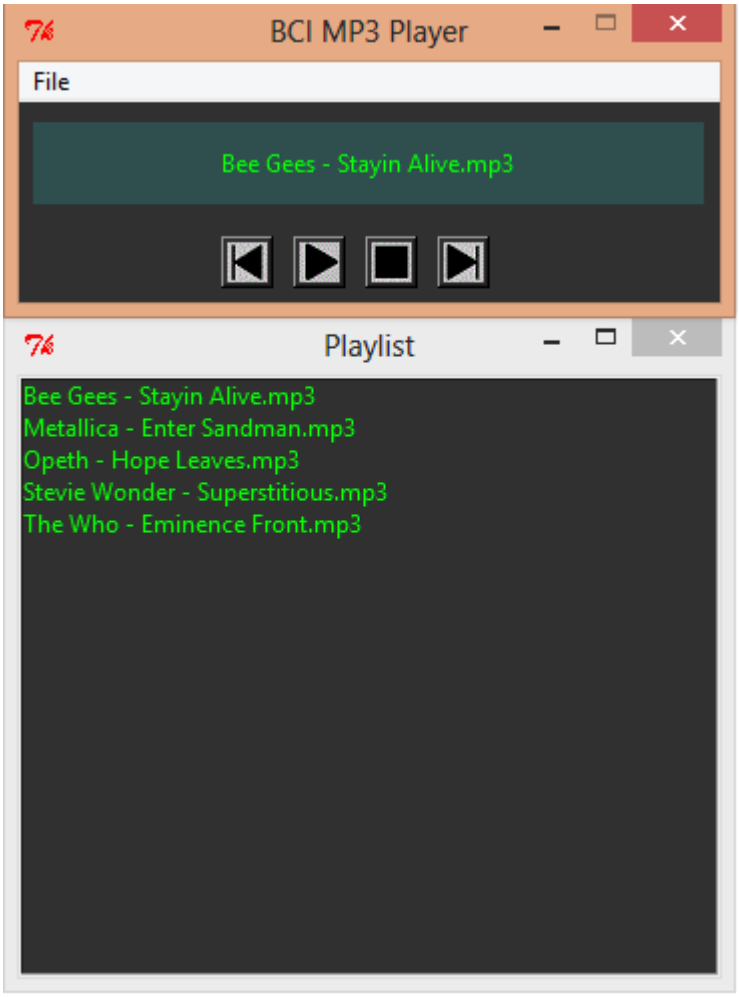


**Fig. 7.2 Application**

The fig.7.2 shows the media player application. The media player supports four operations; play, stop play next and play previous. The application is operable either through the command line or the GUI. Also there is a playlist window that displays the pre-loaded list of songs.

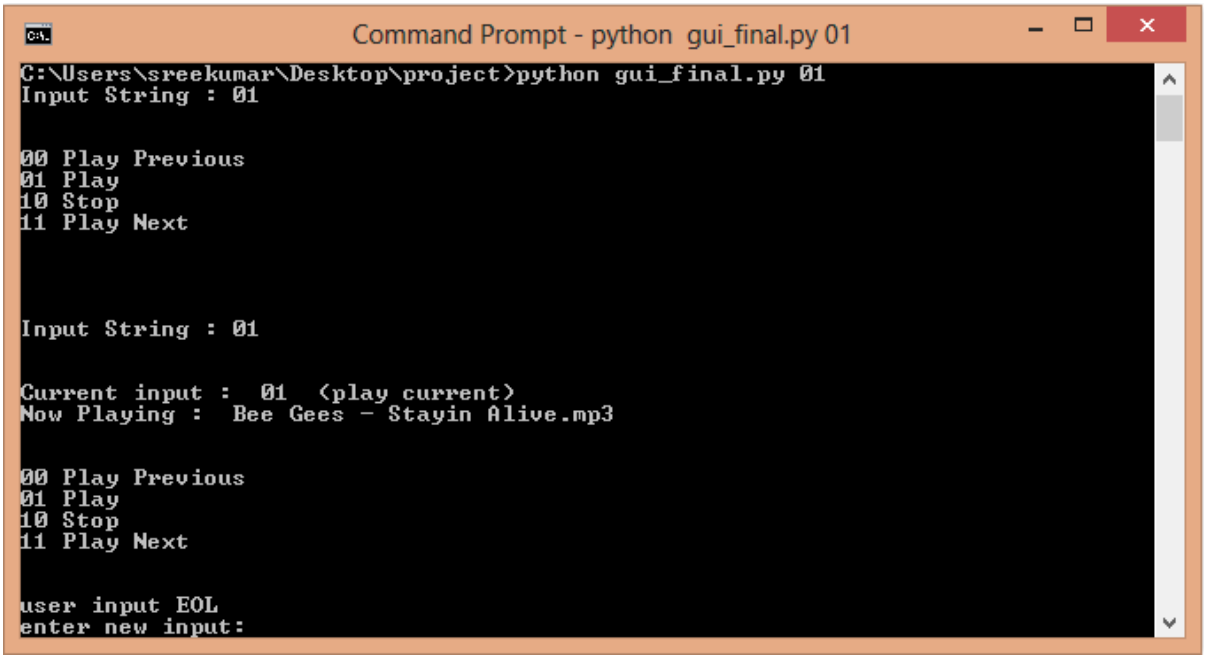


**Fig. 7.3 Command Line Input**

The fig.7.3 shows the command line input operation of the media player. Initially we pass input as argument along with the program to the interpreter. During the program execution, the input sequence is read one after the other and corresponding operation is executed. Once the user input End-Of-Line is encountered, the user is prompted for next sequence of input.

# CHAPTER – 8

# CONCLUSION

# CHAPTER - 8

# CONCLUSION

## 8.1 Conclusion

The main motive of this project is to provide a proof of concept that using the motor imagery paradigm and a combination of binary digits, it is possible to create software systems that can be used by physically disabled people.

An mp3player application was developed for this purpose provided with basic operations such as play, stop, play next and play previous.

The backend processing module was able to classify the dataset and label them with a good accuracy. These labels where fed to the application which then performed the appropriate operations.

## 8.2 Future Enhancements

As for implementation of additional features,

- To improve the accuracy of the classifier
- Incorporate the hardware for the signal acquisition from the user on a real time basis
- Scale the application to support more operations

# REFERENCES

# REFERENCES

[1] “Wyrm: A Brain-Computer Interface Toolbox in Python” by Bastian Venthur, Sven D¨ahne, Johannes H¨ohne, Hendrik Heller and Benjamin Blankertz

[2] “Mu and Beta Rhythm Topographies During Motor Imagery and Actual Movements” by Dennis J. McFarland, Laurie A. Miner, Theresa M. Vaughan, and Jonathan R. Wolpaw

[3] “Decoding human motor activity from EEG single trials for a discrete two-dimensional cursor control” by Dandan Huang, Peter Lin, Ding-Yu Fei, Xuedong Chen and OuBai

[4] “Patients with ALS can use sensorimotor rhythms to operate a brain-computer interface” by Kubler et. al

[5] “Motor Imagery for Severely Motor-Impaired Patients: Evidence for Brain-Computer Interfacing as Superior Control Solution” by Ho¨hne J, Holz E, Staiger-Sa¨lzer P, Mu¨ ller K-R, Ku¨bler A et al.

[6] “An EEG-Based BCI System for 2-D Cursor Control by Combining Mu/Beta Rhythm and P300 Potential” by Yuanqing Li, Jinyi Long, Tianyou Yu, Zhuliang Yu, Chuanchu Wang, Haihong Zhang, and Cuntai Guan

[7] “Quadcopter control in three-dimensional space using a noninvasive motor imagery based brain-computer interface” by Karl LaFleur, Kaitlin Cassady, Alexander Doud, Kaleb Shades, EitanRogin, and Bin He

[8] “Classification of Executed and Imagined Motor Movement EEG Signals” by Jason Sleight, PreetiPillai, Shiwali Mohan

[9] “An ERD/ERS Analysis of the Relation between Human Arm and Robot Manipulator Movements” by Ernesto Pablo Lana, Bruno VilhenaAdorno, Carlos Julio Tierra-Criollo

[10] “Virtual Keyboard Controlled by Spontaneous EEG Activity” by B. Obermaier, G. R. Mueller and G. Pfurtscheller

[11] “Motor Imagery Based On Wavelet Power Spectrum for a Brain Computer Interface” by Javier Castillo-Garcia, Eduardo Caicedo , Berthil Borges Longo, Alan Floriano and TeodianoBastos-Filho

# TEAM INFORMATION

| | |
|---|---|
| 1. NAME: Bhoomika Agarwal<br>USN: 1PE12CS033<br>CONTACT NUMBER: 9036187446<br>EMAIL ID: bhoomika10@gmail.com<br>ADDRESS: 902,2nd main,2nd cross,<br>Koramangala 8th block,<br>Bangalore-560095 | 2. NAME: Sreekumar T<br>USN: 1PE12CS159<br>CONTACT NUMBER: 8904679189<br>EMAIL ID: nair.sreekumar.t@gmail.com<br>ADDRESS: Souparnika house,<br>Periyatadukkam,(PO) Panayal,<br>Kasargod,Kerala-671318 |
| 3. NAME: Sowmya R.<br>USN: 1PE13CS426<br>CONTACT NUMBER: 8147292607<br>EMAIL ID: sowmyarg03@gmail.com<br>ADDRESS: #11/C 2nd main,2nd cross, Balaji<br>Nagar, DRC Post,<br>Bangalore-560029 | 4. NAME: Ramesh M.<br>USN: 1PE13CS416<br>CONTACT NUMBER: 9035230882<br>EMAIL ID: ramesh.m0000@gmail.com<br>ADDRESS: #66,Jangal palya,Jigani hobli,<br>Anekal taluku,Bannerghatta<br>Post,Bangalore-560083 |