

## 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 scientific. 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 Wyrms

**Wyrms**, 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.

Wyrms tries to cover both toolbox-aspects: Wyrms can be used as a toolbox for offline analysis and visualization of neurophysiological data, and in real-time settings, like an online BCI experiment. Wyrms 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 wyrms 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. Wyrms 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.