Abstract:-

Human emotions play an essential role in what's going on in one's life. Humans oftenly use their emotions to express their feelings to each other. We also have hand gestures and tone of voice as the representative to explain one's feelings. Some humans tend to change the mood of other people by using their unique ways. Humans are an endless advanced version of machines who have the best prediction level with highest power of accuracy.

Human expressions are a mandatory task in determining the current mood of an individual. They read the curve of the smile, eyes, cheeks or even forehead. When sad or angry, people tend to speak their heart out to their friends or close one or maybe listen to music. Music is a flair that calms human anatomy and brain. Taking these two ingredients and mixing them together, our project deals with predicting the emotion of an individual through facial expressions and playing music according to an individual mood that will eventually calm the mood of the person.

Our project also offers an individual to play songs according to his current mood or for the betterment of mood. The project aims to capture the emotion expressed by a person through facial expressions. A music player is designed to capture human emotion through the web camera interface available on computing systems.

Chapter 1 .Introduction

Human face is the most important organ of an individual body. And, through that people express and recognize emotions of each other in the communicating system. Human happiness completes the art of living and Music plays one of the vital roles in keeping an individual with a peace of mind and at ease.

Music always turns out to be a stress and tensity reliever. It has always been a boon to an individual and its mindset. With advancement of technology, manual work tends to be in last line and automation has gained a lot of attention. Our system aims to predict the emotion of an individual which will be along with giving input by the end user in which the end user will be asked whether they want to listen to songs according to their current mood or for the betterment of their mood which will be followed by playing songs over YouTube. Our pure system takes the consideration of the respective individual mood i.e. Happy, Sad, Angry, Neutral, or Surprise.

This intelligent system proposes an intelligent agent that categorizes all sorts of emotions as a different music collection which will eventually play songs and recommends an appropriate playlist to the user according to their mood or choice for betterment of the mood.

1.1 Background

People tend to express their emotions, mainly by their facial expressions. Music has always been known to alter the mood of an individual. Capturing and recognizing the emotion being voiced by a person and displaying appropriate songs matching the one's mood and can increasingly calm the mind of a user and overall end up giving a pleasing effect. The project aims to lighten the mood of the user, by playing songs that match the requirements of the user by capturing the image of the user. Since ancient times the best form of expression analysis known to humankind is facial expression recognition.

The best possible way in which people tend to analyse or conclude the emotion or the feeling or the thoughts that another person is trying to express is by facial expression. In some cases, mood alteration may also help in overcoming situations like depression and sadness. With the aid of expression analysis, many health risks can be avoided, and also there can be steps taken that help brings the mood of a user to a better stage.

1.2 Motivation

As a music listener, I've always felt that music players should do way more things than just playing songs and allowing users to make play-lists. A music playlist should be intelligent and act in keeping with user's preferences. Using face emotion recognition for music recommendation has numerous benefits. By analyzing facial expressions, music recommendation systems can accurately determine a user's emotional state and suggest songs that match their mood. This not only enhances the listening experience but also helps users discover new music that they may not have found otherwise.

Furthermore, studies have shown that music can have a significant impact on our emotions and well-being. For example listening to sad music can actually make us feel happier by eliciting feeling of empathy and catharsis. By utilizing face emotion recognition technology, music recommendation systems can take into account these nuances and provide more personalized recommendations based on the user's emotional state.

1.3 Problem Definition

Music is often considered to be voice of the soul as it makes people emote their feelings no matter what the situation is. An angry person tries to calm himself by listening to music which might calm his nerves. A sad person listens to motivating song which helps him to come out of the depression phase. Music and emotion coexist.

Face emotion recognition is a technology that uses artificial intelligence to analyze facial expressions and identify emotions such as happiness, sadness, anger, and surprise. This technology is based on the idea that certain facial expressions are universal across cultures and can be used to infer emotional states.
To detect emotions, face emotion recognition algorithms use machine learning techniques to analyze various features of the face, such as eyebrow position, mouth shape, and eye movement. These features are then compared to a database of known facial expressions to determine the most likely emotion being expressed.

1.4 Scope

A music player which plays songs according to the user's emotion has been designed.
Clearly define the primary goal to recognize user emotions through facial expressions and generate music recommendations that align with the detected emotions. Enable real-time processing of facial expressions to ensure immediate and dynamic responses to users' changing emotional states.

Design an intuitive and user-friendly interface that displays the detected emotions and the recommended music. Consider multi-platform compatibility, such as web, mobile, and desktop applications. Generate dynamic playlists that evolve with the user's changing emotional states, ensuring a continuous and harmonious listening experience.

1.5 Objectives

The Objectives of this project is as follows;

- Accurately detect the mood of the person
- To create a playlist according to the identified emotion by using a real time dataset.
- Real time dataset allows us to capture the person’s image at the particular instant based on which songs can be suggested which complies with his mood.
- Generate dynamic playlists that evolve with the user's changing emotional states, creating a continuous and harmonious listening experience.

The overall objective is to create a user-centric, adaptive system that leverages facial emotion detection to enhance the music listening experience, providing a personalized and emotionally resonant journey for users.

1.6 Selection Of Life Cycle Model

1.7 Organization of Report

The Organization of the Report is structured in the following way:

CHAPTER 1 is entitled as Introduction, describes Background of project, Motivation, Problem definition , Scope , Objective, Selection of Life Cycle Model for Development and Organization of Report.

CHAPTER 2 is entitled as Project Planning and Management, describes Feasibility Study, Risk analysis, Project Scheduling , Effort Allocation, Cost Estimation.

CHAPTER 3 entitled as Analysis, describes Requirement Collection and Identification, Software Requirement Specification(SRS) includes Product Features, Operating Environment, Assumptions, Functional Requirements, Non-Functional Requirements, External Interfaces(User, Hardware, Software, Communication)

CHAPTER 4 entitled as Design, describes System Architecture, Data Flow Diagram, UML Diagrams(Use Case Diagram, Sequence Diagram, Collaboration Diagram, Class Diagram, State Chart Diagram, Component Diagram, Deployment Diagram)

CHAPTER 5 is entitled as Conclusion.

1.8 Summary

As mentioned in the above sections this project aims to design and implement a real time music player which plays music according to the emotion of the user.The Background, Motivation, Problem Definition, Scope, Objective, for Development are as mentioned above. In the next chapter Project Planning and Management will be discuss.

Chapter 2

Project Planning and Management

Project management is defined as a collection of proven techniques for proposing, planning, implementing, managing, and evaluating projects, combined with the art of managing people. It is the application of knowledge, skills, tools, and techniques to a broad range of activities to meet the specified requirements of a particular project. The life cycle of a project has five stages:

Stage 1: Visualizing, selling, and initiating the project.

Stage 2: Planning the project

Stage 3: Designing the processes and outputs

Stage 4: Implementing and tracking the project

Stage 5: Evaluating and closing out the project.

Project planning is part of project management, which relates to the use of schedules such as gantt charts to plan and subsequently report progress within the project environment. Project planning can be done manually or by the use of project management software.

Proposed System is described in Section 2.1. Section 2.2 describes the Feasibility Study, Summary is described in the last section.

2.1 Feasibility Study

The feasibility study is carried out to test whether the proposed system is worth being implemented. Feasibility study is a test of system proposed regarding its work ability, its impact on the organization ability to meet user needs and effective use of resources. The key consideration involve in the feasibility study are :-

- Economical Feasibility
- Operational Feasibility
- Technical Feasibility

2.1.1 Economical Feasibility

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success.

The proposed solution relies on browser-based interface for end user inputs. Since the interface is completely browser based it doesn't require bandwidth allocation and thus reduces the financial aspect of project implementation. The proposed solution requires some software and tools which follow freeware software standards.

2.1.2 Operational feasibility

No doubt the proposed system is GIU based that is very user-friendly and all inputs to be taken all self-explainatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with the new system. As far our study concerned the clients are comfortable and happy as the system has cut down their loads and doing.

2.1.3 Technical feasibility

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. Technical feasibility involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

All the required hardware and software are easily available in the market. So it's possible to implement project.

2.2 Risk Analysis

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Many problems can plague a software project. A risk is a potential problem which may happen or may not. But, regardless of the outcome, it is really a good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur.

Resume builder solely depends on dataset provided by the Candidates. If somehow data gets corrupted then the output shown to the end user will also be ambiguous may not produce desired results.

2.3 Project Scheduling

Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. It is important to note, however, that the schedule evolves over time. During early stages of project planning, a macroscopic schedule is developed. The schedule identities all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into a detailed schedule.

Table 2.1: Schedule Representation Table…..WE HAVE TO CREATE

Figure 2.1: Schedule Representation Chart….WE HAVE TO CREATE

2.4 Effort Allocation

Effort management refers to the effective and efficient allocation of time and resources to perform activities. These activities are generally performed in line with a strategy and a project. Effective effort management requires self-discipline, communication, motivation, energy and focus.

As part of the effort management process, effective scheduling and recording of the performed activities is essential. Depending on the results of the activity, adjustments can be made to further benefit the project and these adjustments will generally be made in the areas of quantity, quality and direction. The main goal of effort management within organizations is to increase viable and beneficial opportunities.

Table 2.2 describes the efforts estimated of following activities.

Figure 2.2: Effort Allocation Table….WE HAVE TO CREATE

2.5 Cost Estimation

Effective software project estimation is one of the challenging and important activities in software development. Proper project planning and control is not possible without sound and reliable estimate. The four basic steps in software project estimation are:

- 1. Estimate the size of development product. The units of measure are lines of code (LOC) function point (FP).

- 2. Estimate the effort in person-months or person-hours.

- 3. Estimate the schedule in calendar months.

- Total number of persons working on project 4 persons.

- Time taken in months 4 months.

- Total time allotted per day in terms of hour 2 hrs.

- Actual working hours (120*2hours)=240hrs

C=aLb

C=cost of project

a=1.4(constant)

b=0.93 (constant)

t= size of code.

For our project considering the number of lines of code to be 18000 based on average numuber of lines code for similar projects we can calculate the cost as follows: Cost of project = 1.4 * (2000)*0.93-Rs 2,604

2.6 Summary

The Project is henceforth, feasible to use as its being developed for building music playlist which would lead and will secure the system, also be beneficial for Our Society.

In this chapter we saw Feasibility Study and it’s following types :- Economical Feasibility, Operational Feasibility, Technical Feasibility, Risk Analysis, Project Scheduling. And finally we come in short summary part. Hence this Chapter-2 is all about Project Planning and Management.In the next Chapter we will see about Analysis.

Chapter 3

Analysis

Project analysis lets you see the present problems and prepare for and avoid future problems. This ensures smooth project execution and timely project delivery. Project analysis is the process of determining the aspects of a project in project management. This analysis helps in identifying whether the project is executing as expected and uses the specified budget. By conducting a project analysis, the current or future problems that occur during the project can be identified.

The Section 3.1 Describe about Requirement collection and Identification. Section 3.2 Describe Software Requirements Specification (SRS). Section 3.3 Describe Functional, NonFunctional Requirements. Section 3.4 Contains Summary of the chapter. 3.1 Requirement Collection and Identification.

3.1 Requirement Collection and Identification

In system engineering and software engineering, requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

3.1.1 Requirements Collection

Requirement Collection is the process of determining, documenting, and managing stakeholder needs requirements to meet project objectives.

Hence, in Collection requirement process, the first step is to identify stakeholders' needs. Second step is to document the needs requirements. And then, manage them throughout the project to meet project goals. This process forms the basis for project scope definition. This process contributes to the success or failure of a project.

3.1.2 Requirement Identification

- Functional requirements:

1. Real-Time Capture: In this module, the system is to capture the face of the user correctly
2. Face Recognition: Here it will take the user's face as input. The convolutional neural network is programmed to evaluate the features of the user image.
3. Emotion Detection: In this section extraction of the features of the user image is done to detect the emotion and depending on the user's emotions, the system will generate captions.
4. Music Recommendation: Song is suggested by the recommendation module to the user by mapping their emotions to the mood type of the song.

3.2 Software Requirement Specification (SRS)

It provides requirements, needs of project and those things which help to complete project. System requirement describe a system from a technical perspective, which describe the essential characteristics of the hardware and software that will meet those needs. It should specify the capabilities, capacities and characteristics of the system in both qualitative and quantitative terms.

3.2.1 Product Features

Product features describe what characteristics a product has. Product features are a product’s discrete areas of new and upgraded functionality that deliver value to your customers. Product features can refer to capabilities, components, user interface (UI) design, and performance upgrades. The Product features of the web portal are as follows:

- Able to suggest songs based on the user's emotions that is sad, happy, neutral, surprise and angry.
- Manual selection of Songs.
- User able to browse through his playlist and select songs that would soothe his mood and emotional experience.
- Able to detect our emotions at real time using webcam feed and smarty classifies.
- The output will be displayed both audio and video in youtube website.

3.2.2 Operating Environment

The software will operate within the following environment:-

- Operating System Later/Linux/MacOS/Windows.
- Minimum Ram: 4GB and a good Internet connection.
- System with processor Intel i3 or above.
- Programming Language: Python
- Software: JetBrains PyCharm Community Edition 2017.1.4 x64

3.2.3 Assumptions

- It is assumed that the input data should contain appropriate information.

- All of the equipment will be in working condition throughout project lifecycle.

- Poor quality of input data can affect the result.

- The project should be user friendly so that it is easy to use for the users.

3.2.4 Functional Requirements

Functional requirements are the functions which are expected from the software or platform. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior. Functional requirements are as follows:

Figure 3.1: Functional Requirements…we have to create.

3.2.5 Non-Functional requirements

Non-functional Requirement is mostly quality requirement. That stipulates how well the portal does, what it has to do. Other than functional requirements in practice, this would entail detail analysis of issues such as availability, security, usability and maintainability.

• Non- requirements are as follows:- functional

Response Time:- System will take approx 5 to 10 sec of response time depending on the hardware specification .

Workload :- System can process one file at a time on local host and may support 100 users at given time, when deployed.

Availability :- If the internet service gets disrupted while processing, input can be processed again.

Safety :- Information transmission should be securely transmitted to server without any changes in information.

Usability :- As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

3.2.6 External Interfaces Requirement :- User, Hardware, Software, Communication

1. User Interface :-

2. Software Interface/Requirements :-

- OS: Windows 7 and above /UBUNTU
- Programming Language: Python
- Software: JetBrains PyCharm Community Edition 2017.1.4 x64
- Backend: Keras
- Additional requirements: TensorFlow.

3.Communication

3.3 Summary

This application will work as remarkable advantage for candidates and job seekers. The Software Requirements Specification (SRS) document lists sufficient and necessary requirements, functional and non-functional requirements, their operating environment, product features, etc. for the project development. This chapter specifies Structure and Behavior of project.

Chapter 4

Design

System design provides the understanding and procedural details necessary for implementing the system. Design is an activity concerned with making major decisions, often of a structural nature.

Design builds coherent, well planned representations of programs that concentrate on the interrelationships of parts at the higher level and the logical operations involved at the lower levels. Software design is the first of the three technical activities designs, coding and test which are required to build and verify the software.

4.1 System Architecture

System architecture defines the structure of a software system. This is usually a series of diagrams that illustrate services, components, layers and interactions. A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. The architecture of a system reflects how the system is used and how it interacts with other systems and the outside world. It describes the interconnection of all the system's components and the data link between them. The architecture of a system reflects the way it is thought about in terms of its structure, functions, and relationships.

Figure 4.1: System Architecture Diagram

4.2 Data Flow Diagram(DFD)

A data flow diagram (DFD) is a graphical representation of the 'flow' of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

Figure 4.2: Data Flow Diagram(DFD)

4.3 UML Diagrams

UML is a graphical notation for expressing object-oriented designs. It is called a modeling language and not a design notation as it allows representing various aspects of the system, not just the design that has to be implemented. For an Object Oriented design, a specification of the classes that exist in the system might suffice. However, while modeling, during the design process, the designer also tries to understand how the different classes are related and how they interact to provide the desired functionality. This aspect of modeling helps build designs that are more likely to satisfy the requirements of the system. Due to the ability of UML to create different models, it has become an aid for understanding the system, designing the system, as well as a notation for representing design. Though UML has now evolved into a fairly comprehensive and large modeling notation, we will focus on a few central concepts and notations relating to classes and their relationships and interactions. The UML is a language for :-

- **Visualizing : -** The structures which are transient can be represented using the UML

- **Specifying: -** The UML addresses the specification of all the important analysis, design and implementation decisions that must be made in developing and deploying a software intensive system.

- **Constructing: -** The UML is not a visual programming language, but its models can be directly connected to a variety of programming languages.

- **Documenting: -** The UML addresses the documentation of a system's architecture and all of its details.

4.3.1 Use Case Diagram

Use case diagram shows the interaction between Use case which represents system functionality and actor which represent the people or system. The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Figure 4.3.1: Use Case Diagram

4.3.2 Sequence Diagram

The sequence diagram shows the flow of functionality through Use case. A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects work together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

Figure 4.3.2: Sequence Diagram

4.3.3 Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Figure 4.3.3: Collaboration Diagram

4.3.4 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

4.3.5 State Transition Diagram

A state chart diagram, also known as a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modelling Language (UML).

4.3.6 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

4.3.7 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

4.4 Summary

This chapter contains important aspects of Designing of Model like the System Architecture, Data Flow Diagrams (DFDs), UML diagrams which contains the Blueprints of the system. They highlight the architecture, and sequential execution of the events. As Designing is one of the most important process for building a project and which also helps in coding phase. UML diagrams are useful for understanding features of application.