AUTOMATED THERAPY USING SOUND, MUSIC AND VIRTUAL THERAPIST

A PROJECT REPORT

Submitted by

ASHIQUE J	**(311617104012)**
ABISHAI S	**(311617104003)**
ARUN RAVISANKKAR	**(311617104010)**

in partial fulfilment for the award of degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

THORAIPAKKAM, CHENNAI-600097

ANNA UNIVERSITY: CHENNAI 600025

APRIL 2021

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**AUTOMATED THERAPY USING SOUNDS, MUSIC AND VIRTUAL THERAPIST**" is the bonafide work of "**ASHIQUE J, ABISHAI S & ARUN RAVISANKKAR "** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. N. SARAVANAN M.E., Ph.D	Ms. X.ANITHA SARAFIN, M.E., (Ph.D)
PROFESSOR	**SUPERVISOR**
HEAD OF THE DEPARTMENT	**ASSOCIATE PROFESSOR**
Department of Computer Science	Department of Computer Science
and Engineering	and Engineering
Misrimal Navajee Munoth Jain	Misrimal Navajee Munoth Jain
Engineering College,	Engineering College,
Thoraipakkam,	Thoraipakkam,
Chennai-600097	Chennai-600097

Submitted for the Project Viva-Voce Examination held on **09.08.2021**______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere gratitude to our honourable Secretary (Administration)
Dr. Harish L Metha and Secretary (Academic) **Shri L. Jaswant S Munoth** for providing the infrastructure facilities to do this project during our course period.

We thank our **Principal**, **Dr. C. C. Christopher ,** for his support and motivation for the development and completion of this project .

We express profound sense of gratitude and thanks to our **HEAD OF THE DEPARTMENT, Dr. N. Saravanan, Professor,** for his kind words and enthusiastic motivation which inspired us a lot in completing this project .

We thank our project coordinator, **Mrs. X. Anitha Sarafin, Associate Professor,** for her valuable suggestions and constant encouragement that led us to the successful completion of our project .

We thank our supervisor, **Mrs. X. Anitha Sarafin, Associate Professor,** who has always been our motivating mentor, for carrying out this project. An assemblage of this nature could never have been attempted without his help. We acknowledge our indebtedness too.

Finally, we thank all the **Teaching and Non -Teaching Staff members of our Department who helped us to complete our project.**

Above all we thank the Almighty, our parents and Siblings for their constant support and encouragement for completing this project.

ABSTRACT

Stress has always been a common thing that everyone encounters, especially young people. Many people across the world suffer from various health conditions due to stress. Sounds and Music have always been a part of our lives, used to aid in physical discomfort by improving respiration, lowering blood pressure, improved cardiac output, reduced heart rate and relaxed muscle tension.

Our aim is to develop an application which can provide sound therapy, music therapy and a virtual therapist. With the help of deep learning, we also provide a virtual therapist for the user to chat and interact with them. The virtual therapist will help us in providing the end-user with a therapist replacement to provide care and comfort.

Sound therapy consists of the user listening to various sounds such as binaural beats which are specifically designed to treat certain problems. Music therapy is an interactive session where the user can listen to various musical instruments to relieve their stress. The user can even play the instrument as the session goes on to provide even more interaction and stress relief.

Although we are fully aware that this is not an actual replacement for therapy, we aim to close the gap between actual therapy and our solution. We genuinely believe that our system is a stepping stone to provide people with proper help and treatment through sound and music therapy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO	ABBREVIATIONS	EXPANSION
1	AI	ARTIFICIAL INTELLIGENCE
2	ML	MACHINE LEARNING
3	DL	DEEP LEARNING
4	LSTM	LONG SHORT TERM MEMORY
5	NLTK	NATURAL LANGUAGE TOOLKIT
6	NLP	NATURAL LANGUAGE PROCESSING
7	JSON	JAVASCRIPT OBJECT NOTATION
8	Tk	Tkinter

CHAPTER 1

INTRODUCTION

1.1 TKINTER:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

1.2 DATA ANALYSIS:

Data analysis is a process of inspecting, cleansing, transforming and modeling data with the goal of discovering useful information, informing conclusions and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping and helping businesses operate more effectively.

1.3 MACHINE LEARNING (ML):

Machine learning (ML) is the scientific study of algorithms and statistical models that computers use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning (ML) algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine Learning (ML) is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning.

1.4 DEEP LEARNING:

Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

CHAPTER 2

LITERATURE SURVEY

Rubijesmin Abdul Latif et. al. [5] discusses comparing different sound types used in stress therapy and recognizing preferences among users. The aim of the stress therapy is to reduce stress among users, by replacing traditional guided therapy sessions with virtual reality environments. Besides the element of virtual environment, it is believed that sound has an impact towards reducing stress. Experiments were conducted (i.e. therapy sessions) using three sound types; natural beach environment sound, instrumental (piano) sound and zikr with instrumental (piano) sound. By the end of experiments, comparisons of stress level were made (measuring heart rate and users' expressions) before and after each session. Natural background sound has the highest impact followed closely by the zikr with instrumental music on reducing stress level among participants as compared to just instrumental sound. The result of the experiments displayed the different impact of sound towards reducing stress.

The Australian Psychological Society et. al. [1] (APS), in conjunction with an online research company, conducted the Stress and Wellbeing in Australia Survey with a representative sample of Australians. The national sample (n=1521) comprised approximately equal numbers of men and women and was representative of the Australian adult population (18 and above) for age, gender, geographical location and work status (matched on Australian Bureau of Statistics [ABS]) as shown in Appendix A: Sample Statistics). In addition to conducting the main survey, just under half the sample (n=740) completed an additional survey relating specifically to social media usage and FOMO. Further, a group of Australian teenagers aged 13 to 17 years (n=210) were recruited through parent panelists of the online research company to allow us to understand Australian teens’ experience of social media and how they compared with those of the adult population. The teen sample comprised equal numbers of boys (n=103) and girls (n=103), with four (n=4) individuals not nominating a gender. In total, 1,731 people completed the online survey, which was conducted over a two-and-a-half week period from 14 August to 31 August 2015

Kuhr C et. al. [3] studies user experience is about how a user interacts with, and experiences, a particular product, system or service. It includes a person's perceptions of utility, ease of use, and efficiency**.** User experience is the umbrella term for the research area that concerns itself with the investigation of people’s experience with interactive technologies. In academia, it is with the research about the elements and characteristics of this experience and the psychological constructs it incorporates. In industry, UX methods are applied in research and product development with the goal of making products better in one way or another**.** To reduce errors and make usage of interfaces more satisfying, research about the interaction between a human being and a technological system needs to be conducted. If there is knowledge about the benefits and shortcomings concerning the usage of an interface, designers and engineers can use this knowledge to improve and optimize the means of interaction. To make sure that in future the UX can be applied widely, further validation studies need to be conducted. It is necessary to investigate if the questionnaire produces consistent results independently from the to-be-evaluated object, researcher and situation.

Kent D [4] states that patients who listened to music while waiting for surgery subjectively reported lower anxiety and also displayed lower blood pressure and pulse rates than those who did not. A survey studying the difference in GPA between students who listen to music while studying and those who do not finds no overall significant difference, but does find that students who listen to hiphop and rap while studying score significantly lower while students who listen to easy listening and classical are likely to have higher GPAs.

Mental health statistics [2] states the following points.

Age Difference - 30% of older people reported never feeling overwhelmed or unable to cope compared to 7% of young adults.

Behavioural effects - 46% reported that they ate too much or ate unhealthily due to stress. 29% reported that they started drinking or increased their drinking, and 16% reported that they started smoking or increased their smoking.

Psychological effects - 51% of adults who felt stressed reported feeling depressed, and 61% reported feeling anxious. 16% had self-harmed and 32% said they had had suicidal thoughts and feelings.37% of adults who reported feeling stressed reported feeling lonely as a result.

Causes of Stress - Housing worries and pressure to succeed are the key source of stress.

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

3.1.1 SYSTEM OVERVIEW

Our main goal of the automated therapy program is to understand the reason for stress and the need for a solution that is easily accessible to as many people as we can.

In the existing system, there are various music players such as Spotify, Amazon Prime Music, JioSaavn, Gaana etc. Their main purpose is to provide users music and albums created by artists for commercial purposes.

Chatbots are the main thing when it comes to users talking with machines. Other bots such as voice-bots, text based bots, factual bots, etc. But none of them are really focused on therapy and helping the users with mental problems and closure.

Sound therapy in the existing system has always been a foreign and niche concept, both to consumers and innovators.There has been applications which focus only on meditation. In simple terms, there exist applications which focus on meditation but there hasn't been a proven product which focuses on sound and music therapy.

In short, there isn't an existing system that provides therapy using sound, music and virtual therapists.

3.1.2 DRAWBACK

Some applications dips their toes on sound therapy but nothing has been enough to use as a viable product.

Simple features that are already available can be embedded to the applications in a more appealing and useful manner but aren't used due to the accessibility level for the youth. This can be solved by making the applications follow universal design principles.

Personalized Evaluation of therapy intention in future can be made based on available healthcare data set so that the user can evaluate or change its behavior based on various predictions.

In short, there isn't an existing system that provides therapy using sound, music and virtual therapists.

3.2 PROPOSED SYSTEM

3.2.1 OVERVIEW

We aim to provide a system that consists of sound therapy, music therapy and a virtual therapist.

- First, we provide the end-user with a personality survey to understand the user's problems and evaluate if the user requires any form of treatment. Based on the results, proper treatment and therapy sessions will be given to the user.
- We also provide a virtual therapist for the user to chat and interact with them using NLTK and CNN. We use a dataset that consists of various interactions with real people stored in a .json file. The virtual therapist will help us in providing the end-user with a therapist replacement to provide care and comfort.
- At the end of each session, we provide the user with another personality survey to find out if there's an improvement from our session, by comparing our initial analysis data with the current one.

3.2.2 ADVANTAGES

This personality survey will allow clients to know whether they have the need to undergo therapy or not.

This program provides an user interface that is easily accessible and easy to navigate for the clients.

Feature selection is done based on the mutual information classification which will largely help us to assign weightage to each parameter which is used in the model.

3.3 REQUIREMENT ANALYSIS

The requirement specification is a technical specification of requirements for the software products. The purpose of the software requirement specification is to provide a detailed overview of the software project, its parameter and goal. It describes the project target audience and its user interface, hardware and software requirements.

3.3.1 SOFTWARE REQUIREMENT

The software requirements give a detailed description of the system and all its features.

- Jupyter Notebook
- Keras
- NLTK

3.3.2 HARDWARE REQUIREMENT

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete engineer as the starting point for the system design.

Ram	:	8GB Ram and more
Processor	:	Any Intel i5 Processor
Hard Disk	:	50GB and more
Speed	:	2GHZ and more

3.4 TECHNOLOGIES USED

3.4.1 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

Jupyter notebook also provides us with a cell which can be used as a code editor for python based editors.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell.

3.4.2 KERAS

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

Up until version 2.3 Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML. As of version 2.4, only TensorFlow is supported. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System),[5] and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep-learning models on clusters of Graphics processing units (GPU) and tensor processing units (TPU).

Keras claims over 375,000 individual users as of early-2020. Keras has been adopted for use in scientific research due to Python (programming language) and its own ease of use and installation. Keras was the 10th most cited tool in the KDnuggets 2018 software poll and registered a 22% usage.

3.4.3 NLTK

The **Natural Language Toolkit**, or more commonly **NLTK**, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook.

NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems. There are 32 universities in the US and 25 countries using NLTK in their courses. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK.

CHAPTER 4

SYSTEM DESIGN

4.1 OVERVIEW OF ARCHITECTURE

The diagram shows the overall architecture of the project which consists of the personality survey, virtual therapist and music player. Each block shows their structural and functional components of the project.

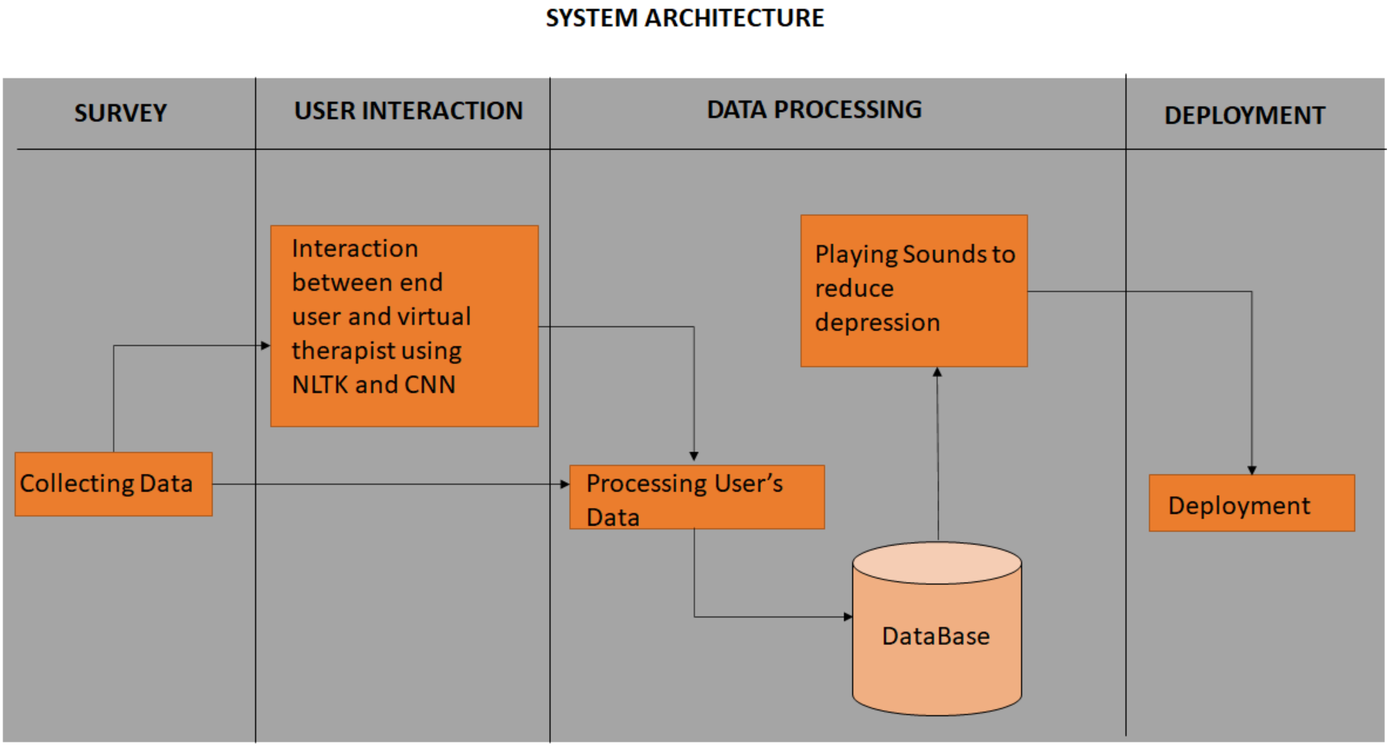


Figure 4.1 OVERVIEW OF ARCHITECTURE

4.2 UML USE CASE DIAGRAM

This diagram shows the user such as actor, editors and the role of developer in this project. This behaviour diagram models the functionality of the system using use cases.

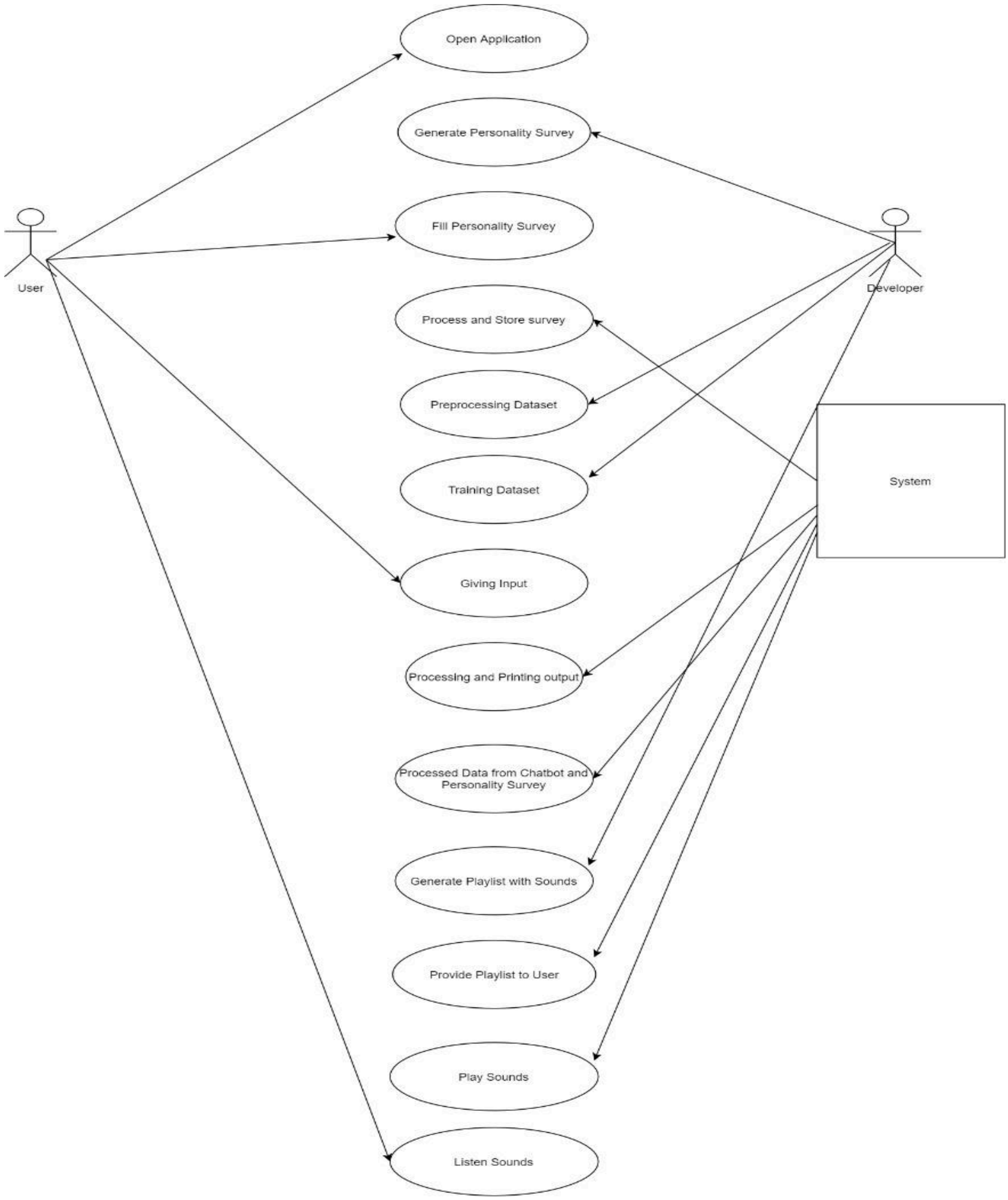


Figure 4.2 USE CASE DIAGRAM

4.3 UML CLASS DIAGRAM

The structure of the application id described in the class diagram by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Each class has their attributes and the operations defined with the relationship between the classes.

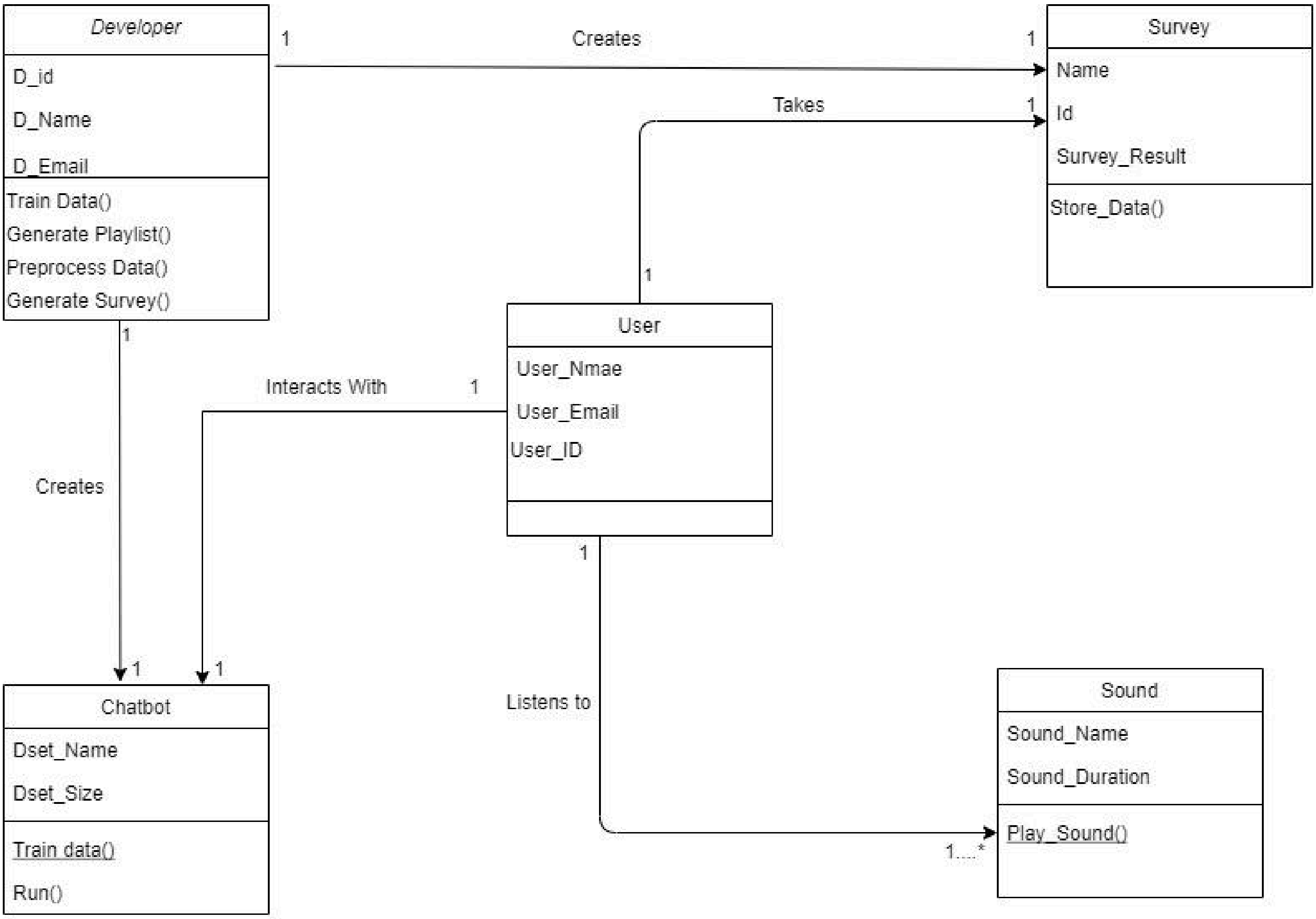


Figure 4.3 CLASS DIAGRAM

4.4.1 MODULE 1 – SYSTEM SEQUENCE DIAGRAM - VIRTUAL THERAPIST

This module diagram represents the flow from one sequence to another sequence. The activity can be described as an operation of the system. Some activities are based on conditions satisfied by the actor/object.

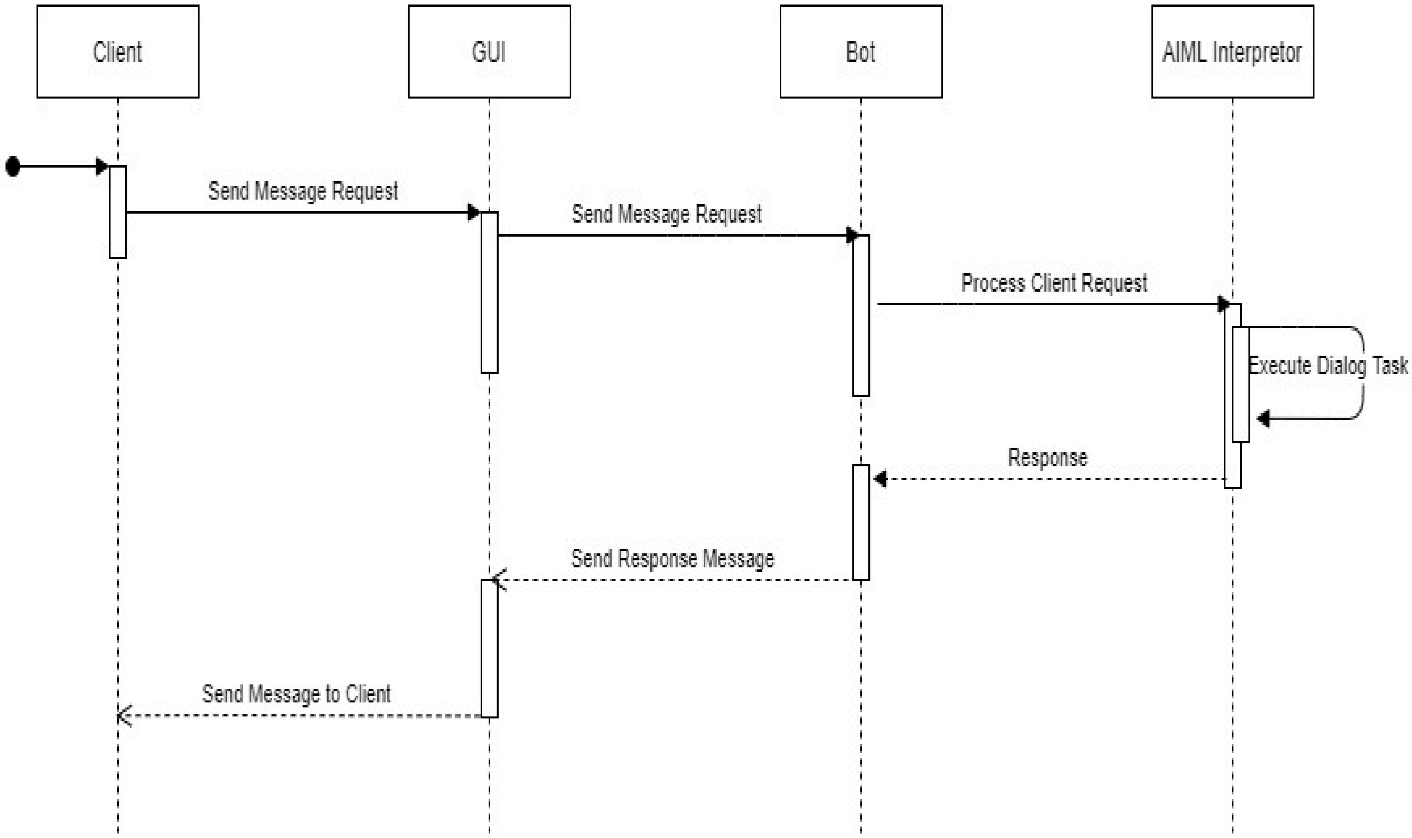


Figure 4.4.1 MODULE: Sequence diagram for Virtual Therapist

4.4.2 MODULE 2 – SYSTEM SEQUENCE DIAGRAM - MUSIC PLAYER

This module diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. The objects in this module are the client, database and application .

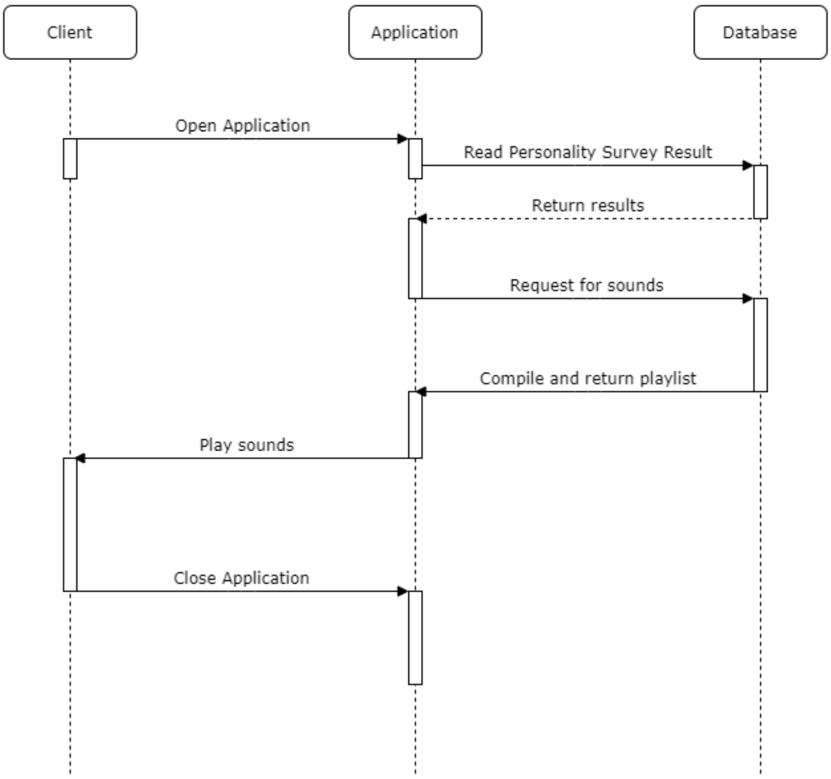


Figure 4.4.2 MODULE 2 – Sequence Diagram for Music Player

CHAPTER 5

IMPLEMENTATION

5.1 MODULES

- Understanding the environment
- Tools required
- Personality Survey
- Virtual therapist
- Music Player
- Integration and Testing
- Deployment

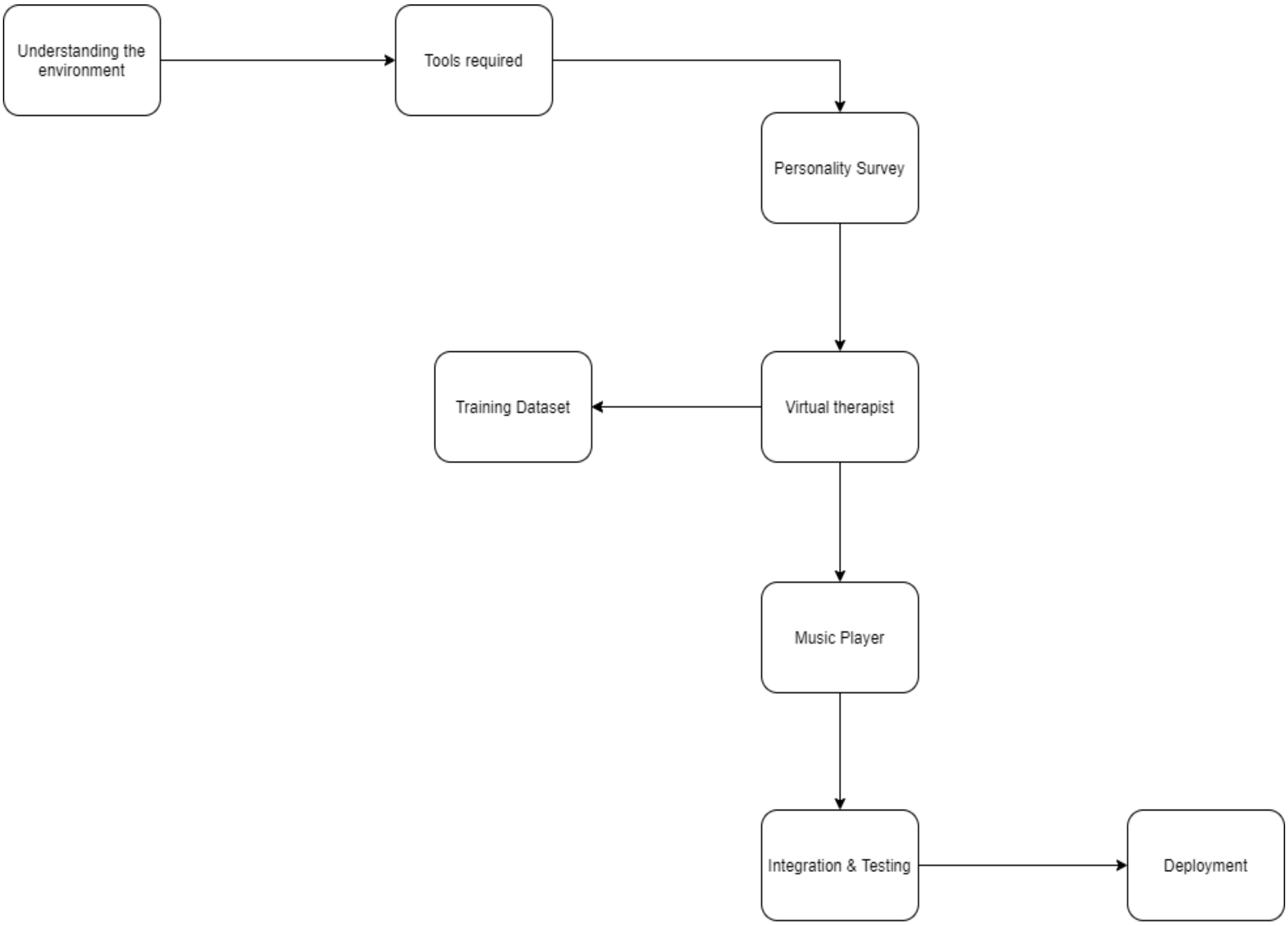


Figure 5.1 phases of implementation

5.1.1 UNDERSTANDING THE ENVIRONMENT

In this project, our goal has three folds of objectives.

- The first goal is to understand the need for therapy. This includes finding different types of stress, reasons for stress and how to overcome them.
- The second is to create a tool that can easily identify the stress levels of the user.
- The third is to use that tool to provide suitable treatment using sounds, music and virtual therapists to overcome their stress.

We utilize the data from our literature surveys and base paper to understand the reason for stress and the need for a tool to combat them. We analyze the tool and its modules, construct them, analyze and test and verify that there is an improvement in one's well-being after using our constructed tool.

5.1.2 TOOLS REQUIRED

5.1.2.1 SOFTWARE REQUIREMENT

The software requirements give a detailed description of the system and all its features.

- MS Excel
- Python 3
- Jupyter Notebook
- Keras
- NLTK

5.1.2.2 HARDWARE REQUIREMENT

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete engineer as the starting point for the system design.

Ram	:	8GB Ram and more
Processor	:	Any Intel i5 Processor
Hard Disk	:	50GB and more
Speed	:	2GHZ and more

5.1.3 PERSONALITY SURVEY

The Personality Survey includes finding personal information about the client such as their personality. It consists of various statements and the client will consist of how much they agree/disagree to that statement. We compile and store the observations from the data, in order to create a hypothesis. The data survey given by the client must be reliable and accurate.

Question Number	Question	Answers
Q1	I have a greater dependency on alcohol, caffeine, nicotine or drugs	Strongly agree
Q2	You almost always feel that life is very much not worth living.	Disagree
Q3	You are prone to worrying that things will take a turn for the worse.	Agree
…	…	…
Q..	…	…

5.1.4 VIRTUAL THERAPIST

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete engineer as the starting point for the system design.

These virtual therapy options allow people to seek treatment in the comfort of their own home, without having to travel to see a therapist in person. There are different types of virtual therapy, each of which has various benefits and limitations.

Virtual therapy is a type of telemedicine. It includes any treatment that a person seeks through an electronic device.

Some examples of virtual therapy include:

- talking to a practitioner via video conferencing software
- using an app to access therapy
- phone- and email-based therapies, such as when a physical therapist suggests specific exercises via email
- the use of online devices to assess clients or patients remotely — for instance, when a speech therapist uses online tools to measure progress

In theory, any treatment that does not require physical contact or laboratory testing can work on a virtual platform.

The most prevalent types of virtual therapy include:

Virtual psychotherapy

Virtual psychotherapy, sometimes called telemental health or telepsychology, treats people with mental health issues, relationship or sexual health problems, or significant stress via video chat, email, phone, text messaging, or email.

In most virtual psychotherapy sessions, a licensed therapist provides traditional therapy through a new platform. A client might talk about their emotions, seek insight on their relationships, and ask for help implementing lifestyle changes.

A newer form of virtual psychotherapy uses apps or coaching to improve mental health. This approach is not a form of traditional therapy because a person does not get care from a licensed practitioner. Instead, they might monitor their own symptoms over time, get virtual coaching from a bot, or receive daily mental health tips.

Virtual physical therapy

Virtual physical therapy offers traditional care but in an online or phone-based setting. A physical therapist might discuss recent symptoms, recommend exercises, or administer screenings.

In some cases, a therapist might ask a client to perform exercises and then use a camera to evaluate their form and progress.

Some physical therapy apps complement therapy by offering additional exercises or allowing a client to track their progress between sessions. A person can use these apps alongside virtual or in-person therapy.

Virtual speech therapy

Virtual speech therapy can treat a range of speech disorders, such as a stutter, aphasia from a stroke, or pronunciation difficulties.

In a virtual session, a therapist may evaluate a person's speech, offer them strategies for correcting speech issues, or help them practice new speech patterns. An emerging form of virtual speech therapy uses bots in place of real people to improve speech.

Virtual speech therapy apps are also available to help people work toward their speech goals between sessions or track speech changes over time.

Virtual occupational therapy

Occupational therapy helps people master specific life skills. People often use it in conjunction with other types of treatment. For example, a person with speech issues resulting from a stroke might choose speech therapy, then use occupational therapy to help them master the motor skills necessary to use a speech assistive device.

In virtual occupational therapy, a therapist offers coaching, tips, and feedback on techniques on a virtual platform, such as via video chat. Some forms of virtual occupational therapy may also use virtual reality to mimic real-world situations that the individual might face.

5.1.5 MUSIC PLAYER

The music player allows a user to play various media file formats. It can be used to play audio as well as video files. The music player is a software project supporting all known media files and has the ability to play them with ease.

The project features are as follows:

- Users may attach Folder to Play and add various media files within it.
- Users may see track lists and play desired ones accordingly.
- Supports various music formats including .mp3, WMA, WAV etc.
- Interactive GUI
- Consists of Pause/Play/Stop Features
- Consists of a Volume controller
- The system also consists of a sound Equalizer
- It Displays the media playing time with Track Bar so that user may drag the media play as needed.

5.1.6 INTEGRATION & TESTING

Once all the individual modules are created, we must integrate all these modules into a single module, so that users can seamlessly flow through our project without any hassle.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

Various testing will be done, which includes:

1. Integration testing.
2. Unit testing.
3. Functional testing
4. System testing.
5. Acceptance testing.
6. Black-Box testing.
7. White-Box testing.

5.1.7 DEPLOYMENT

Deployment is the process of organizing the information gained, such as the model, so that it is understandable for the model user. This step is carried out by the user rather than the analyst, so it is essential for the user to understand how to use the models. The result of this step is a final report.

CHAPTER 6

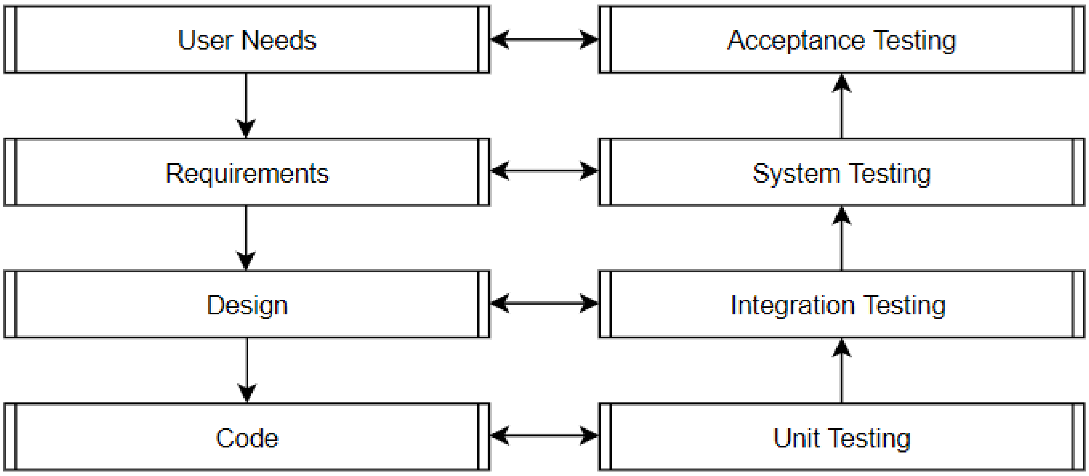
SYSTEM TESTING

6.1 TESTING OBJECTIVES

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTS

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are



6.2.1 UNIT TEST CASES

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

In this project the unit testing validates the program logic of all modules in the range of machine learning and big data.

The testing takes place as:

Input: Module select, Text, drop down boxes

Output: Adapt user interface based on module select. Show information on getting input required for each operation, to show required output.

6.2.2 INTEGRATION TEST CASES

Integration tests are designed to test integrated software and hardware components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing. The combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

In this project the integration takes place in hardware components and software testing is verified.

Input: Survey module, Player module and VT module

Output: System working well, instantaneous output, less memory consumption and Data usage.

Front End: Python GUI(Tkinter)

Back End: Deep Learning.

6.2.3 WHITE BOX TEST CASES

White box testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

In this project, white box testing takes place as follows

- Testing of Back end through the database provided.
- Testing of Front end through the input provided and relevant output produced for the inputs through the coding logic and verified.

6.2.4 BLACK BOX TEST CASES

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests must be written from a definitive source document, like specification or requirements documents, such as specification or requirements document. It is attesting in which the software under test is treated, as a black box. The test provides inputs and responds to outputs without considering how the software works.

In this project Black box testing is the final step which is the documentation process with the theoretical information and does not involve the practical step by step procedure, hence the testing is performed and verified.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

Sound therapy uses different aspects of sound to improve your emotional and physical well-being. How it works depends on the method being used. Most music therapy sessions are experienced one-on-one with a specially trained practitioner.

Though evidence may be limited on some methods, music therapy has been found to be effective for stress reduction and relaxation and has been shown to offer many health benefits.

There is little risk to listening to music.

7.2 FUTURE ENHANCEMENT

This project can be further improved by including a wider range of sounds and music, since our project is mainly focused on adolescents and young adults. In order to improve the accuracy of the prediction model, it is essential to increase the amount of data or choose better, more fitting, variables.

APPENDICES

APPENDIX 1

SAMPLE CODING

MUSIC PLAYER

```
from tkinter import *

from ttk themes import themed_tk

from tkinter import filedialog

import tkinter.ttk as ttk

from tkinter import messagebox

from mutagen.mp3 import MP3

from mutagen.mp4 import MP4

import os

import vlc

import time

root = themed_tk.ThemedTk(theme='adapta')

root.title('AUTOMATED THERAPY USING SMV')

root.geometry('770x514')

root.maxsize(770,514)

root.minsize(770,514)

root.configure(background='#F3F3F3')
root.iconphoto(False, PhotoImage(file='images/logo.png').subsample(2,2))

play_image = PhotoImage(file='images/play final.png').subsample(32,32)

pause_image = PhotoImage(file='images/pause final.png').subsample(32,32)

play_prev = PhotoImage(file='images/play previous.png').subsample(7,7)

play_next = PhotoImage(file='images/play next.png').subsample(7,7)

-------------------------------------------------------------------------------------------

flag = {'val':0}

default_volume = {'vol':50}

time_dict = {'current':0, 'length':1}

prev_dir = {'val':''}

def play_pause():

  if p.is_playing():

      play_button.config(image=play_image)

   else:

      play_button.config(image=pause_image)

def play_time():

   global current_time

   current_time = int(p.get_time()/1000)

   time_dict['current'] = current_time

   real_time = time.strftime('%M:%S', time.gmtime(current_time))
```

```
    status_time.config(text=real_time)
    status_time.after(1000, play_time)
    slider.config(value=current_time)
    if time_dict['current'] == time_dict['length']:
        next()
def song_length(index):
    tag = lb.get(index)[-4:]
    global total_length
    if '.mp3' == tag:
        s = directory + '/' + lb.get(index)
        audio = MP3(s)
        total_length = int(audio.info.length)
        time_dict['length'] = total_length
        real_length = time.strftime('%M:%S', time.gmtime(total_length))
        total_time.config(text=real_length)
    elif '.m4a' == tag:
        s = directory + '/' + lb.get(index)
        audio = MP4(s)
        total_length = int(audio.info.length)
        time_dict['length'] = total_length
        real_length = time.strftime('%M:%S', time.gmtime(total_length))
        total_time.config(text=real_length)
    slider.config(to=total_length)
define manipulate(index):
    song_name = lb.get(index)[:-4]
    if len(song_name) < 30:
        lb_item.config(text=song_name)
    else:
        s_name = song_name[:27]+'...'
        lb_item.config(text=s_name)
    song = directory + '/' + lb.get(index)
    global p
    if flag['val'] == 0:
        p = vlc.MediaPlayer(song)
        p.play()
        flag['val'] += 1
    else:
        p.stop()
        p = vlc.MediaPlayer(song)
        p.play()
    p.audio_set_volume(default_volume['vol'])

    play_time()

    song_length(index)

def select_item(event):

    global index

    try:

        index = lb.curselection()

        manipulate(index)

        play_pause()

    except Exception:

        pass

def add_playlist():

    global directory

    global last_song_index

    directory = filedialog.askdirectory(title='Choose directory')

    try:

        if directory != '':

            lst = os.listdir(directory)

            lb.delete(0, END)

            lb_item.config(text='Choose song')
```

```
for i in list:
                if ('.mp3' in i) or ('.m4a' in i):
                    last_song = i
                    lb.insert(END, i)
            if lb.size() > 0:
                last_song_index = lb.get(0, END).index(last_song)
            if lb.size() == 0:
                messagebox.showinfo(title='Empty -(AUTOMATED THERAPY USING SMV)', message='This folder is empty!\nPlease select another folder.')
                add_playlist()
            prev_dir['val'] = directory
        else:
            path_error = 'Choose a valid path!'
            messagebox.showerror(title='Path error -(AUTOMATED THERAPY USING SMV)', message=path_error)
            directory = prev_dir['val']
    except Exception:
        pass
def pause():
        try:
```

```
        p.pause()

        play_pause()

    except Exception:

        pass

def volume_adjust(s):

    try:

        default_volume['vol'] = 100-int(volume_slider.get())

        p.audio_set_volume(default_volume['vol'])

    except Exception:

        pass

def next():

        try:

         index = lb.curselection()

        index = index[0]+1

        if (index <= last_song_index):

            manipulate(index)

            lb.select_clear(0, END)

            lb.activate(index)

            lb.selection_set(index, last=None)

            play_pause()
    except Exception:
        pass
def previous():
        try:
        index = lb.curselection()
        index = index[0]-1
        if index >= 0:
            manipulate(index)
            lb.select_clear(0, END)
            lb.activate(index)
            lb.selection_set(index, last=None)
            play_pause()
    except Exception:
        pass
def slide(x):
   try:
        slide_position = slider.get()
        to_set = slide_position / total_length
        p.set_position(to_set)
        slider.config(value=slide_position)
    except Exception:
        pass
img = PhotoImage(file='images/Bhoomi.png').subsample(2,2)
left_frame = Frame(root, width=400, height=500, background='#F3F3F3', bd=0)
left_frame.grid(row=0, column=0)
left_frame.config(highlightthickness=0, borderwidth=0)
right_frame = Frame(root, width=500, height=500, background='#F3F3F3', bd=0)
right_frame.grid(row=0, column=1)
Label(left_frame, image=img, bg='#F3F3F3').pack()
first_frame = Frame(right_frame, width=500, height=50, background='#F3F3F3')
first_frame.pack(anchor='nw', pady=8)
second_frame = Frame(right_frame, width=500, height=400,
background='#F3F3F3')
second_frame.pack(anchor='nw')
third_frame = Frame(right_frame, width=500, height=400, background='#F3F3F3',
pady=15)
third_frame.pack(anchor='nw')
Button(first_frame, text='+ Add folder', background='#5E17EB',
borderwidth=0,foreground='white', pady=8, font='ubuntu 10 bold',
command=add_playlist, activebackground='#5E17EB',
activeforeground='white').grid(row=0,column=0)
volume_slider = ttk.Scale(first_frame, from_=0, to=100, length=80,
command=volume_adjust,
                    value=default_volume['vol'], orient=VERTICAL)
volume_slider.grid(row=0, column=2, sticky='ne')
Label(first_frame, text='', bg='#F0F2F6', padx=100).grid(row=0, column=1)
Label(second_frame, text = 'Songs', bg='#F0F2F6', fg='#545454', pady=15,
font='ubuntu 10 bold').pack(anchor='nw', side=TOP)
sb = Scrollbar(second_frame, orient=VERTICAL, borderwidth=0, width=8, tru
color='white', bg='#ffafd7',
           activebackground='#ffafd7')
lb = Listbox(second_frame,height=12, font='ubuntu 10', fg='#545454', width=40,
          select background='#ffafd7', activstyle=NONE, yscrollcommand=sb.set)
lb.config(highlightthickness=0, highlight background='grey',
foreground='#545454', bd=0)
sb.config(command=lb.yview)
sb.pack(side=RIGHT, fill=Y)
lb.bind('<ButtonRelease-1>', select_item)
lb.pack()
lb_item = Label(third_frame, text='AUTOMATED THERAPY USING SMV',
font='ubuntu 15', bg='#F3F3F3', fg='#545454')
```

```
lb_item.grid(row=0, column=0, columnspan=4, sticky='nw', pady=10)

status_time = Label(third_frame, text='00:00', bg='#F3F3F3', font='ubuntu 10', fg='#545454')

status_time.grid(row=1, column=0, sticky='nw')

total_time = Label(third_frame, text='00:00', bg='#F3F3F3', font='ubuntu 10', fg='#545454')

total_time.grid(row=1, column=2, sticky='ne')

slider = ttk.Scale(third_frame, from_=0, to=100, length=325, command=slide)

slider.grid(row=2, column=0, columnspan=3)

play_button = Button(third_frame, command=pause, image=play_image, bg='#F3F3F3', borderwidth=0, activebackground='#F3F3F3', highlightthickness=0)

play_button.grid(row=3, column=1, sticky='news',pady=15)

Button(third_frame, image=play_prev, font='ubuntu 20 bold', bg='#F3F3F3', command=previous, borderwidth=0, activebackground='#F3F3F3', highlightthickness=0).grid(row=3, column=0, sticky='ne', pady=15, padx=15)

Button(third_frame, image=play_next, font='ubuntu 20 bold', bg='#F3F3F3', command=next, borderwidth=0, activebackground='#F3F3F3', highlightthickness=0).grid(row=3, column=2, sticky='nw', pady=15, padx=15)

root.mainloop()
```

VIRTUAL THERAPIST [TRAINING]

```
import nltk

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

import json

import pickle

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Activation, Dropout

from keras.optimizers import SGD

import random

words=[]

classes = []

documents = []

ignore_words = ['?', '!']

data_file = open('intents.json').read()

intents = json.loads(data_file)

for intent in intents['intents']:

    for pattern in intent['patterns']:

        w = nltk.word_tokenize(pattern)
```

```
      words.extend(w)

      documents.append((w, intent['tag']))

           if intent['tag'] not in classes:

         classes.append(intent['tag'])

words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore_words]

words = sorted(list(set(words)))

classes = sorted(list(set(classes)))

print (len(documents), "documents")

print (len(classes), "classes", classes)

print (len(words), "unique lemmatized words", words)

pickle.dump(words,open('words.pkl','wb'))

pickle.dump(classes,open('classes.pkl','wb'))

training = []

output_empty = [0] * len(classes)

for doc in documents:

    bag = []

    pattern_words = doc[0]

            pattern_words = [lemmatizer.lemmatize(word.lower()) for word in pattern_words]
```

```
       for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)
    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1
    training.append([bag, output_row])
random.shuffle(training)
training = np.array(training)
train_x = list(training[:,0])
train_y = list(training[:,1])
print("Training data created")
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd,
metrics=['accuracy'])
hist = model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=5,
verbose=1)
model.save('chatbot_model.h5', hist)
print("model created")
```

VIRTUAL THERAPIST [APPLICATION]

```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
import pickle
import numpy as np
from keras.models import load_model
model = load_model('chatbot_model.h5')
import json
import random
intents = json.loads(open('intents.json').read())
words = pickle.load(open('words.pkl','rb'))
classes = pickle.load(open('classes.pkl','rb'))
def clean_up_sentence(sentence):
sentence_words = nltk.word_tokenize(sentence)
sentence_words = [lemmatizer.lemmatize(word.lower()) for word in sentence_words]
```

```
return sentence_words

def bow(sentence, words, show_details=True):

sentence_words = clean_up_sentence(sentence)

bag = [0]*len(words)

for s in sentence_words:

        for i,w in enumerate(words):

            if w == s:

                bag[i] = 1

                if show_details:

                    print ("found in bag: %s" % w)

    return(np.array(bag))

def predict_class(sentence, model):

    p = bow(sentence, words,show_details=False)

    res = model.predict(np.array([p]))[0]

    ERROR_THRESHOLD = 0.25

    results = [[i,r] for i,r in enumerate(res) if r>ERROR_THRESHOLD]

    results.sort(key=lambda x: x[1], reverse=True)

    return_list = []

    for r in results:

        return_list.append({"intent": classes[r[0]], "probability": str(r[1])})
    return return_list
def getResponse(ints, intents_json):
    tag = ints[0]['intent']
    list_of_intents = intents_json['intents']
    for i in list_of_intents:
        if(i['tag']== tag):
            result = random.choice(i['responses'])
            break
    return result
def chatbot_response(msg):
    ints = predict_class(msg, model)
    res = getResponse(ints, intents)
    return res
import tkinter
from tkinter import *
def send():
    msg = EntryBox.get("1.0",'end-1c').strip()
    EntryBox.delete("0.0",END)
    if msg != '':
        ChatLog.config(state=NORMAL)
    ChatLog.insert(END, "You: " + msg + '\n\n')

    ChatLog.config(foreground="#442265", font=("Verdana", 12 ))

    res = chatbot_response(msg)

    ChatLog.insert(END, "Bot: " + res + '\n\n')

    ChatLog.config(state=DISABLED)

    ChatLog.yview(END)

base = Tk()

base.title("Hello")

base.geometry("400x500")

base.resizable(width=FALSE, height=FALSE)

ChatLog = Text(base, bd=0, bg="white", height="8", width="50", font="Arial",)

ChatLog.config(state=DISABLED)

scrollbar = Scrollbar(base, command=ChatLog.yview, cursor="heart")

ChatLog['yscrollcommand'] = scrollbar.set

SendButton = Button(base, font=("Verdana",12,'bold'), text="Send", width="12",
height=5, bd=0, bg="#32de97", activebackground="#3c9d9b",fg='#ffffff',
command= send )

EntryBox = Text(base, bd=0, bg="white",width="29", height="5", font="Arial")

scrollbar.place(x=376,y=6, height=386)

ChatLog.place(x=6,y=6, height=386, width=370)
```

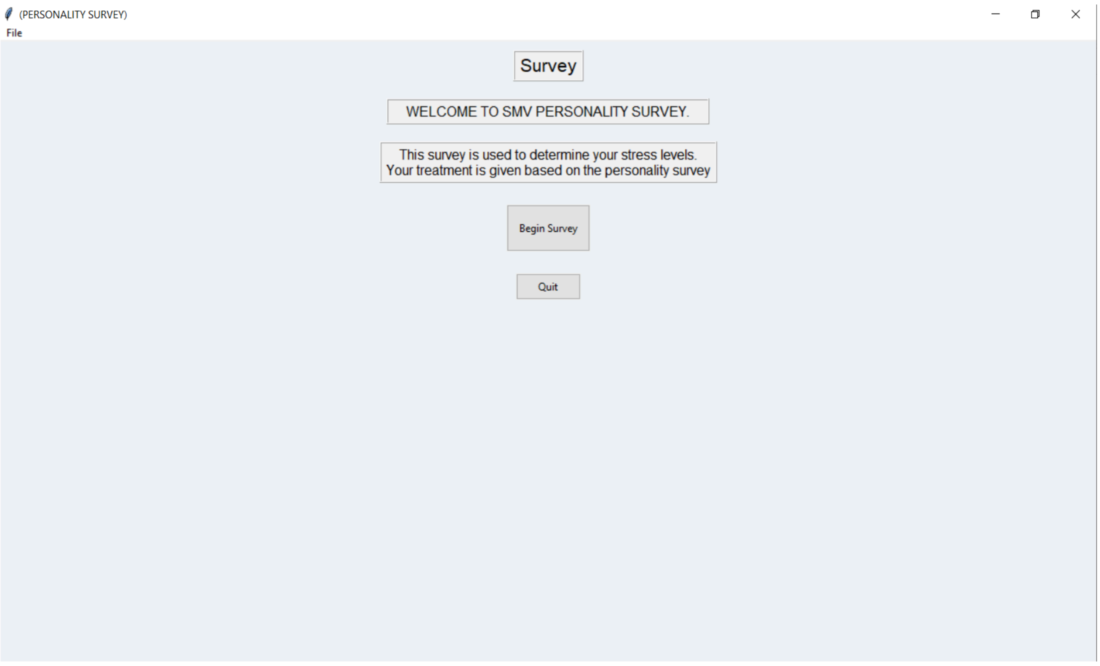
```
EntryBox.place(x=128, y=401, height=90, width=265)

SendButton.place(x=6, y=401, height=90)

base.mainloop()
```

APPENDIX 2

SCREENSHOTS

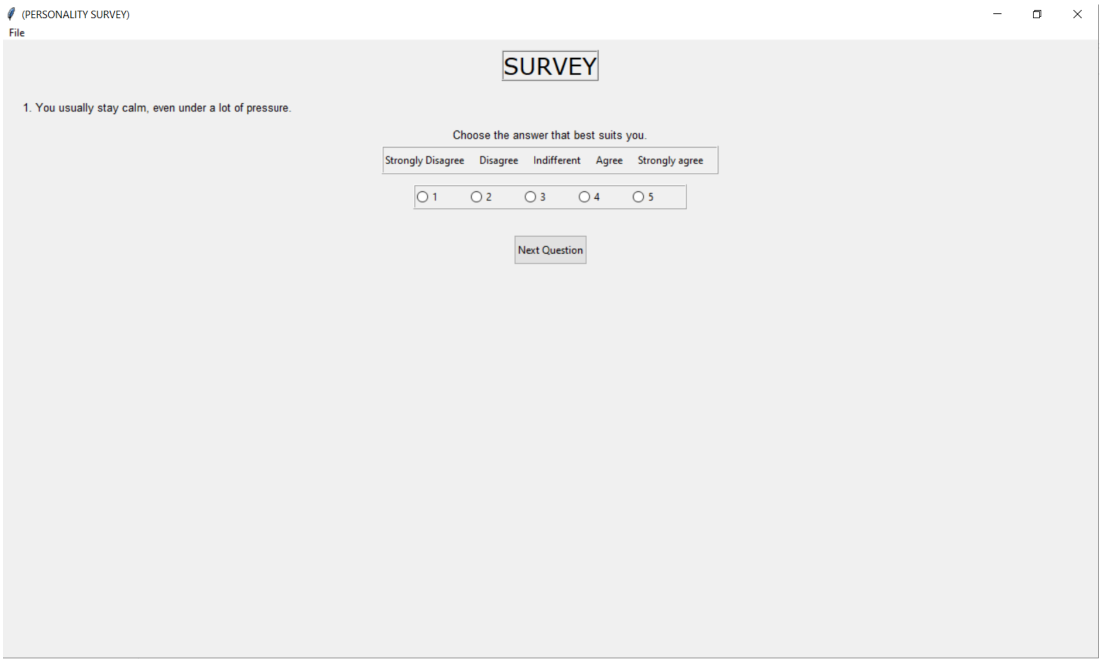
PERSONALITY SURVEY

The Personality Survey includes finding personal information about the client such as their personality. It consists of various statements and the client will consist of how much they agree/disagree to that statement. We compile and store the observations from the data, in order to create a hypothesis. The data survey given by the client must be reliable and accurate.



WELCOME TO SMV PERSONALITY SURVEY.

This survey is used to determine your stress levels.
Your treatment is given based on the personality survey

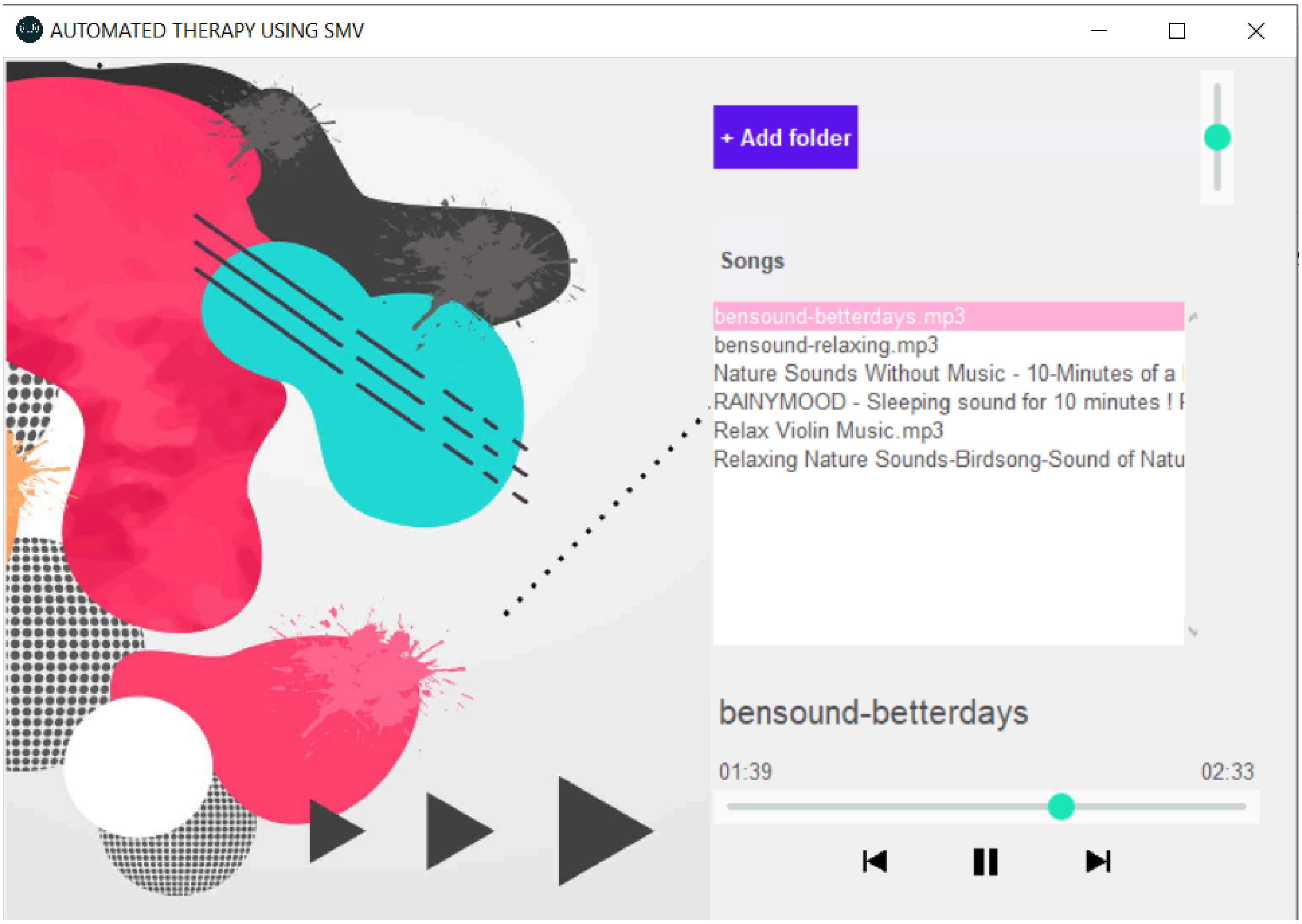


1. You usually stay calm, even under a lot of pressure.

Choose the answer that best suits you.

MUSIC PLAYER

The music player allows a user to play various media file formats. It can be used to play audio as well as video files. The music player is a software project supporting all known media files and has the ability to play them with ease.



AUTOMATED THERAPY USING SMV

+ Add folder

Songs

bensound-betterdays.mp3
bensound-relaxing.mp3
Nature Sounds Without Music - 10-Minutes of a
RAINYMOOD - Sleeping sound for 10 minutes ! F
Relax Violin Music.mp3
Relaxing Nature Sounds-Birdsong-Sound of Natu

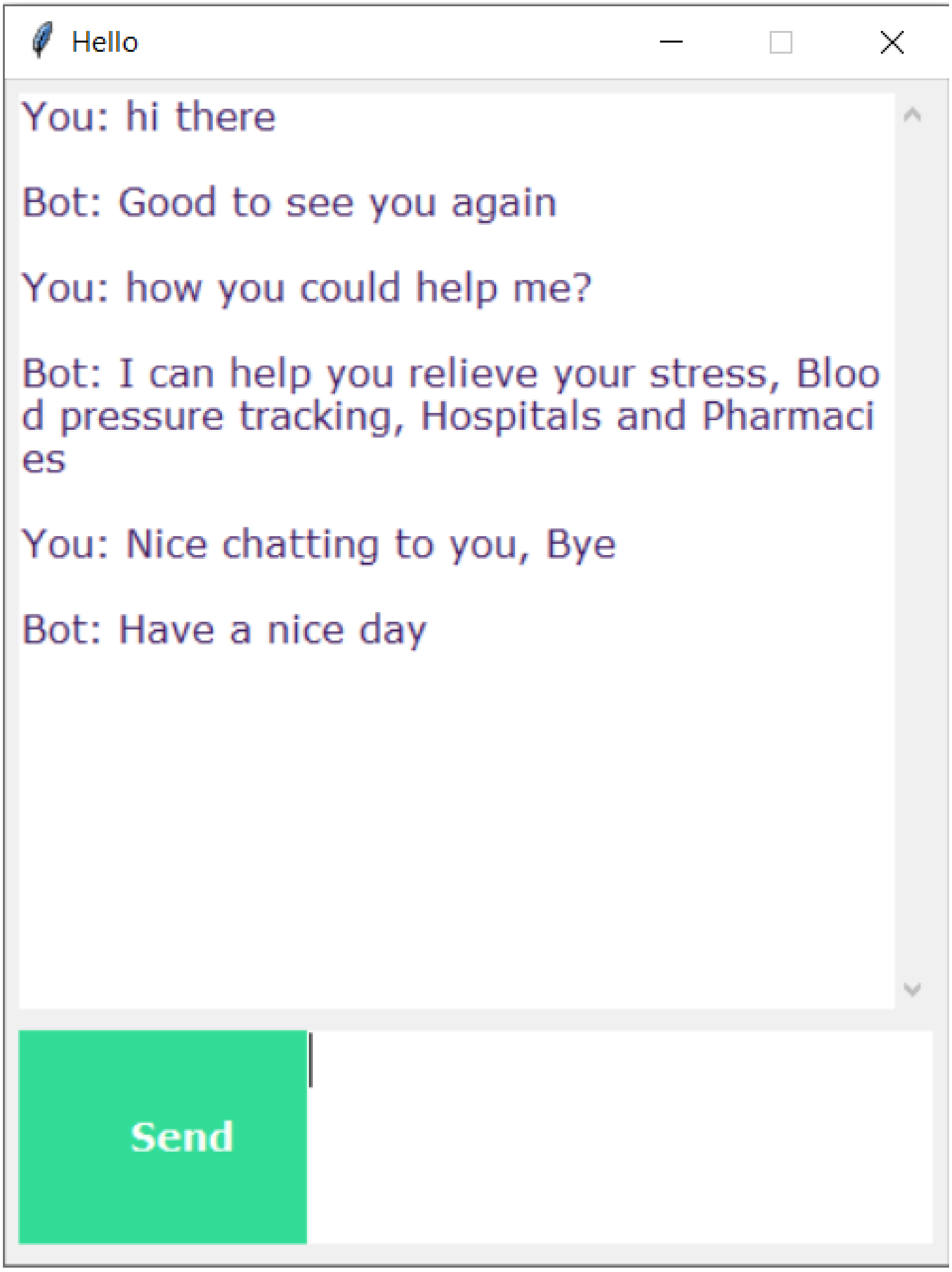
bensound-betterdays

01:39 02:33

VIRTUAL THERAPIST

- These virtual therapy options allow people to seek treatment in the comfort of their own home, without having to travel to see a therapist in person. There are different types of virtual therapy, each of which has various benefits and limitations.

- Virtual therapy is a type of telemedicine. It includes any treatment that a person seeks through an electronic device.



You: hi there

Bot: Good to see you again

You: how you could help me?

Bot: I can help you relieve your stress, Bloo
d pressure tracking, Hospitals and Pharmaci
es

You: Nice chatting to you, Bye

Bot: Have a nice day

REFERENCES

[1] Australian Psychological Society.: Understanding and man[1]aging stress. Melbourne, Australia. [Online]. Retrieved at (Jan 2016) from https://www.psychology.org.au/assets/Files/StressTipSheet.pdf.

[2] Mental Health Foundation Stress, retrieved at (Sep 2016) from https://www.mental[1]health.org.uk/a-to-z/s/stress (2015).

[3]Kuhr, C., Minge, M., Thüring, M.: Measuring the User Experience of mobile applications. An empirical validation of a quantitative method. IEEE Journal (2014).

[4] Kent, D: The Effect of Music on the Human Body and Mind. Retrieved from Senior Thesis submitted in partial fulfillment of the requirements for graduation in the Honors Program Liberty University Spring (Oct 2016)

[5] Rubijesmin Abdul Latif: Preferred Sound Type for Stress Therapy. IEEE Journal (Aug 2018).

[6] Chaehan So: Are You an Introvert or Extrovert? Accurate Classification With Only Ten Predictors. IEEE Journal (Feb 2020)