A

MINI PROJECT REPORT

on

MORSE CODE CONVERTOR

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

(MIP-A27)

P SAITEJA (207Y5A0504)

S ANIL KUMAR (207Y5A0505)

Under the Guidance

of

Mr. CH.V Krishna Mohan

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MARRI LAXMAN REDDY

INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AUTONOMOUS)

(Affiliated to JNTU-H, Approved by AICTE New Delhi and Accredited by NBA & NAAC With 'A' Grade)

September 2022

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

CERTIFICATE

This is to certify that the project report titled "**Morse Code Convertor"** is being submitted by **P Saiteja (207Y5A0504) & S Anil Kumar (207Y5A0505)** in IV B.Tech I Semester **Computer Science & Engineering** is a record bonafide work carried out by him. The results embodied in this report have not been submitted to any other University for the award of any degree.

Internal Guide **HOD**

Principal **External Examiner**

DECLARATION

I hereby declare that the Mini Project Report entitled, "**Morse Code Convertor"** submitted for the B.Tech degree is entirely my work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree.

Date:

By:

P Saiteja	**207Y5A0504**
S Anil Kumar	**207Y5A0505**

ACKNOWLEDGEMENT

I am happy to express my deep sense of gratitude to the principal of the college **Dr. K. Venkateswara Reddy,** Professor, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided me with adequate facilities to pursue my project.

I would like to thank **Mr. Abdul Basith Khateeb,** Assoc. Professor and Head, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.

I am very grateful to my project guide **Mr. CH.V Krishna Mohan,** Assoc. Prof**.**, Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for his extensive patience and guidance throughout my project work.

I sincerely thank my seniors and all the teaching and non-teaching staff of the Department of Computer Science for their timely suggestions, healthy criticism and motivation during the course of this work.

I would also like to thank my classmates for always being there whenever I needed help or moral support. With great respect and obedience, I thank my parents and brother who were the backbone behind my deeds.

Finally, I express my immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to my need at right time for the development and success of this work.

CONTENTS

ABSTRACT

Morse code is a process of transmitting text information as a series of on-off tones and lights or clicks. If they use a tapping device recipient can understand the message without additional decoding equipment. Morse code is represented by the form of dots and dahs. Here dots refer to dots and dahs refer to dash. Morse code used to transmit only numerals at first. After that, Alfred Vail included letters and characters. Morse code can be transmitted by using electric telegraph wire, light, and sound, through a different medium in different ways. Morse code is a character encoding and decoding scheme.

Our main view of the project is to create an application that can converts the given text in to morse code and given morse code to text by using python programming.

In this report, we will go through the whole process related to our interesting project. A table of content will guide us to each individual part of this report. At the beginning, Introduction part gives a rough idea about why we do this project and what our objectives are. We did some research on the Morse code and the main technology python used in this project. Then we present our design and implementation of the application. Some special functions usage and realization process accompanied with some pictures helping explain in a detail way. Testing is always necessary in any project. In the following Experiment and Testing part, we present how we find problems, solve problems and improve our application's performance. Finally, in the conclusion part, we show how we implement our application step by step.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background

Morse code is a method of transmitting text information as a series of on-off tones, lights, or clicks that can be directly understood by a skilled listener or observer without special equipment. It is initially invented by Samuel Finley Breese Morse in 1937. And it has been in use for more than 160 years—longer than any other electrical coding system. What is called Morse code today is actually somewhat different from what was originally developed by Vail and Morse. After some changes, International Morse Code was standardized at the International Telegraphy Congress in 1865 in Paris, and was later made the standard by the International Telecommunication Union (ITU). International Morse code today is most popular among amateur radio operators . Morse code is useful in many fields, such as radio navigation, amateur radio, warship, the signal lamp included in a submarine periscope and so on. Besides, Morse code has been employed as an assistive technology, helping people with different native languages or people with a variety of disabilities to communicate. For the general public, an important application is signaling for help through SOS, "···— — — ···". This can be sent by many ways: keying a radio on and off, flashing a mirror, toggling a flashlight and similar methods, this will be very useful especially when you are in wild and your phone is out of power or no signal.

1.2 Motivation

Nowadays, there are many kinds of Android Apps of Morse Code in the android market. Here we are creating an application that converts given morse code in to text and given text to morse code.

some of the examples:

This is an app helping users to learn Morse Code. One can choose either transmitting or receiving mode to practice corresponding kill and receive the performing feedback immediately. It includes he functions of letter training (both transmitting and receiving), training (only transmitting), free mode, speed adjusting (WPM), sound effects adjusting and electronic handbook.

Morse Code Translator: This is an app allowing users to send short flashlight text messages using the International Morse Code. The apps Includes the features that the flashlight can transmits short messages of lights (Morse Code); there are some Template

stored in the database for emergencies. For example, SOS; allowing users to save new messages; changing frequency of the transmitted
signal.

1.3 Objectives

In the project, we are going to study the Morse code, python programming, tinker, visual studio, real-time processing and develop a Morse code translator for windows. The keyboard is used to give the text and decode the text to get the Morse code.
We want achieve the following objectives:

- To detect if a string is Morse code or normal text.
- Encoding text and display Morse code.
- Decoding Morse code and display text.
- Prepare a dictionary that maps all ‘normal text’ symbols to their respective Morse code translations.

1.4 Expected Outcome:

Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different signal durations, called dots and dashes, or dits and dahs. Morse code is named after Samuel Morse, one of the inventors of the telegraph. By using this application, we get Morse code for given text accurately within less time. we can also get text by giving the input as Morse code.

1.5 Morse code

1.5.1 Background

International Morse Code, also known as Continental Morse Code, encodes the 26 Latin letters a through z, one non-Latin letter, the Arabic numerals, and a small set of punctuation and procedural signals (prosigns). There is no distinction between upper and lower case letters. Each Morse code symbol is formed by a sequence of dits and dahs. The dit duration is the basic unit of time measurement in Morse code transmission. The duration of a dah is three times the duration of a dit. Each dit or dah within an encoded character is followed by a period of signal absence, called a space, equal to the dit duration. The letters of a word are separated by a space of duration equal to three dits, and words are separated by a space equal to seven dits.

1.5.2 Symbol Representation

1.5.2.1 Characters

Character	Code	Character	Code	Character	Code
A	._	J	.___	S	...
B	_...	K	_._	T	_
C	_._.	L	._..	U	.._
D	_..	M	__	V	..._
E	.	N	_.	W	.__
F	.._.	O	___	X	_.._
G	__.	P	.__.	Y	_.__
H	....	Q	__._	Z	__..
I	..	R	._.		

Table 1.5.2.1 Morse code representation for characters

1.5.2.2 Numbers

Numeric	Code	Numeric	code
1	.____	**6**	_....
2	..___	**7**	__...
3	...__	**8**	___..
4	...._	**9**	____.
5	.....	**0**	_____

Table 1.5.2.2 Morse code representation for Numbers

1.5.2.3 Punctuation

Character	Code	Character	Code
Period [.]	·–·–·–	Colon [:]	–––···
Comma [,]	––··––	Semicolon [;]	–·–·–·
Question mark [?]	··––··	Double dash [=]	–···–
Apostrophe [‘]	·––––·	Plus [+]	·–·–·
Exclamation mark [!]	–·–·––	Hyphen, Minus [-]	–····–
Slash [/]	–··–·	Underscore [_]	··––·–
Parenthesis open [(]	–·––·	Quotation mark [“]	·–··–·
Parenthesis close [)]	–·––·–	Dollar sign [$]	···–··–
Ampersand [&]	·–···	At sign [@]	·––·–·

Table 1.5.2.3 Morse code representation for Punctuations

1.6 Instance

Here is an example of phrase “F Y P” in Morse Code format, each letter is separated by a space:

··–· –·–– ·––·
F Y P

Morse Code is often spoken or written with “dah” for dashes, “dit” for dots located at the end of a character, and “di” for dots located at the beginning or internally within the character.
Thus, “F Y P” is orally: Di-di-dah-dit Dah-di-dah-dah Di-dah-dah- dit

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

This project is mainly developed for windows operating system. Internet is not required for this application. There are some of the other Morse code converter apps in android and windows many of them use taping sounds and flash light to recognize the Morse code. In this chapter we discuss about related work, comparative studies, scope of problem and the challenges.

2.2"MORSE CODE DATASETS FOR MACHINE LEARNING"

Authors:

Sourya Dey, Keith M. Chugg and Peter A. Beerel

Abstract:

They have presented an algorithm to generate synthetic datasets of tenable difficulty on the classification of morse code symbols for supervised machine learning problems. They completely used machine learning for this. They have done this process in four steps frame portioning, assigning values for intensity levels, noising, mass generation

2.3 "MORSE CODE GENERATOR USING MICROCONTROLLER USING ALPHANUMERIC KEYBOARD"

Authors:

Paparao Nalajala Bhavana Godavarth, M Lakshmi Raviteja, Deepthi Simhadri

Abstract:

They presented the paper using radiotelegraphy .they used keypads etc., the clock signal is generated by matrix keypad.it uses an LCD, led, buzzer, most importantly transmitter.it uses a microcontroller for processing the morse code to display on the LCD screen.

2.4 "CLOUD STORAGE SECURITY SCHEME USING DNA COMPUTING WITH MORSE CODE AND ZIGZAG PATTERN"

Authors:

Dr A. Murugan, R. Thilagavathy

Abstract:

Cloud Storage Security Scheme using DNA Computing with Morse code and Zigzag Pattern .storage and exchange of data are done in cloud computing, DNA computing, zigzag chippers. It uses a DNA of an organism to process an input and by computing this they take initially a stream of charactersand converted to binary data, to DNA sequence and then to morse code and then to zigzag pattern the resultant is the cipher text.

2.5 "C-MORSE":

Authors:

Zhimeng Yin, Wenchao Jiang,Song Min Kim, Tian

Abstract:

Cross technology Communication with Transparent Morse. Coding it uses a Wi-Fi to zig bee and viceversa it uses a transport protocol like TCP, UDP. Degradations are very low nearly 4%. The Wi-Fi 33 sender modulates its symbols by only perturbing the transmission timing of through data packets and control frames within a negligible delay. In this way, C-Morse constructs the statistically recognizable energy patterns in the ISM band in a transparent way. On top of the modulator, C-Morse has its multiple access control, and a packet buffer to opportunistically utilize the existing data packets, avoiding the need for generating dummy packets. In addition to transparency, and provide the CTC throughput while maintaining transparency. With 100 packets/s, C-morse achieves a throughput of 12bps and manages to boost its CTC throughput to 137bps at 800 packets/s. For a fair comparison, the CTC sender avoids the generation of dedicated packets (e.g beacons or data packets) and does not require synchronization. In the literature, only Free Bee (asynchronous version) meets these conditions with a 14.6bps throughput. The 9× throughput gain is achieved by utilizing all kinds of existing Wi-Fi traffic, instead of the limited beacons. Note that the CTC throughput discrepancy between UDP and TCP is due to the different tolerable delay bounds (10ms for TCP and 15ms for UDP).

2.6 "MORSE CODE TRANSLATOR USING THE ARDUINO PLATFORM CRAFTING THE FUTURE OF MICROCONTROLLERS"

Authors:

Sérgio Silva, António Valente, Salviano Soares, M.J.C.S. Reis, Jean Paiva, Paulo Bartolomeu

Abstract:

This revolutionary idea allows long distance communication between people connecting humanity and building bridges between nations. By the end of the 19th century, however, new technologies began to emerge, many of them based on the same principles first developed for the telegraph system. In time, these new technologies would overshadow the telegraph, which would fall out of regular widespread usage, but as in the past, today the same principles developed by the telegraph can craft again the future so researchers think with the rise of Li-Fi and VLC. The electric equivalent circuit is shown. There are two buttons, one for the dit signal and one for the dash. S1 represents the dit and S2 the dash. The LED will blink at the same frequency of the dash or dit, making Morse visible besides audible. The electric circuit was drawn usingthe IDE Fritzing. of the morse string. Context stores morse code of a single character. It keeps count ofthe spaces between morse characters.

CHAPTER 3
ANALYSIS

3.1 Definition:

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

3.2 Overview:

At the beginning of this report, we've already had a rough idea about what Morse code is and what Morse code can do. Here we introduce the mechanism of Morse code in detail, like what Morse code is composed of and how it works. International Morse code will be introduced and used here.

3.3 Algorithm:

The algorithm is very simple. Every character in the English language is substituted by a series of 'dots' and 'dashes' or sometimes just singular 'dot' or 'dash' and vice versa.

3.4 Coding rule:

International Morse code is composed of five elements:

1) Short mark, dot or "dit" (●) which is one time unit long.
2) Longer mark, dash or "dah" ( ) which is three times units long.
3) Inter-element gap between the dots and dashes within a character which is one dot's duration (one unit long).
4) Short gap between letters which is three times units long.
5) Medium gap between words which is seven times units long.

3.5 Feasibility study

A feasibility analysis usually involves through assessment of the operational (need), financial and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints. A feasibility study should be relatively cheap and done at the earliest possible time. Depending on the study, the decision is made whether to go head with a more detailed analysis.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.

Facts considered in the feasibility analysis were:

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

3.5.1 Technical Feasibility

Technical feasibility includes whether the technology is available in the market for development and its availability. The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc., in order to give an introduction of technical system. Considering our project it is technical feasible. Training and Placement System, with its emphasis on a more strategic decision-making process is fast gaining ground as a popular outsourced function.

3.5.2 Economic Feasibility

This feasibility study present tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service.

Thus feasibility study should center along the following points:

- Improvement resulting over the existing method in terms of accuracy, timeliness.
- Cost comparison

- Estimate on the life expectancy of the hardware.
- Overall objective.

Our project is economically feasible. It does not require much cost to be involved in the overall process. The overall objective is in easing out the recruitment processes.

3.5.3 Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept broad audience from around the world.

3.6 Functional Requirements

This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. A user can open application by clicking on the button and can choose from which language to which he has to convert the code.

3.7 Non-Functional Requirements

- **3.7.1 Usability**

This section includes all of those requirements that effect usability.

- We get the response within seconds.
- The software must have a simple, user-friendly interface so customers can save time and confusion.
- As the project is made using python, it has fast loading time then the application made using any other language

- **3.7.2 Reliability**

- The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

➢ 3.7.3 Interface

• The user interface is based on the application. The application is developed using python and tinker.
• The Interface design is aimed at a flexible environment to provide the user with clear information in navigating a user-friendly interface is planned.

3.8 Performance Requirements

- High Speed: System should process requested task in parallel for various action to give quick response.
- The response to the user is within seconds, providing all the information at a glance.
- Accuracy: System should correctly execute process, display the result accurately. System output should be in user required format.

CHAPTER 4
DESIGN

4.1 Definition

The most creative and challenging face of the system development is System Design. It provides the understanding and procedural details necessary for the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfill. The first step is to determine how the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing.

Design of the system can be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus system design is a solution to ―how to approach to the creation of a new system. This important phase provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The design step provides a data design, architectural design, and a procedural design.

4.2 Architecture Diagram

- When a character is given as input by translating it we get a morse code from the hashmaps as output.
- When Morse code is given as input then by translating it we get a character as output.
- The use of the hashmaps is most predominant in python for accessing the data structure.
- Every key of the character has its value in the database we need to access through a function called itertools.
- Cipher stores the morse translated form of the English string.
- Decipher stores the English translated form of the morse string.
- Context stores morse code of a single character. It keeps count of the spaces between morse characters.

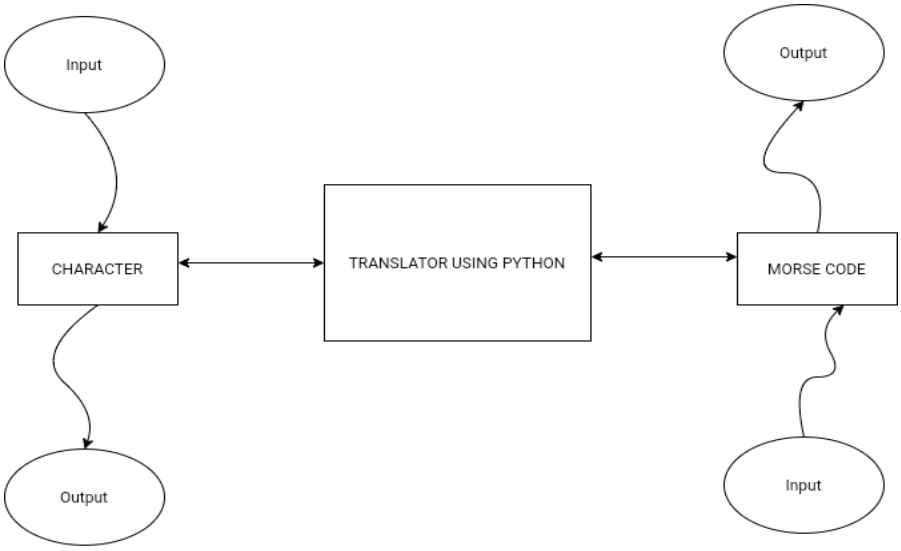


Fig.4.2.1 System Architecture

4.3 UML Diagram

A Morse code translator program from the UML design given below for translating English message files into Morse code, and Morse code files into English. The requirements for this program are as follows:

- English message files will be stored as text files, one sentence per line.
- Morse code message files will be stored as text files, one encoded character per line, where end of words are indicated by a blank line, and end of sentences by two blank lines.
- English message files will contain file extension .eng, and Morse code files file extension .mor. The assumptions for this program are as follows:
- Messages will only contain lowercase letters, the digits 0–9, periods, commas, and question marks. Following is the UML diagram for the program.

The Translator class coordinates the translation of messages (English to Morse code, and Morse code to English). It contains two class members through composition: Input-Buffer and Output-Buffer. Thus the Translator class is responsible for their construction. The Input-Buffer contains the current message line read from the file. If the file contains an English message, then an English-Input-Buffer is used; if the files

contains a Morse-coded message, then a Morse-Input-Buffer is used. As the Translator reads the current Msg Char from the Input-Buffer, it requests the Msg Char to translate itself, sending it to the Output-Buffer. When is End Of Word or is End Of Sentence is found true, it calls the corresponding Mark End-Of Word or Mark End Of Sentence method of the output-buffer. If the output is in English, a space character for each end of word and a period for each end of sentence is added to the buffer; if the output is in Morse code, then a blank line for each end of word, and two blank lines for each end of sentence is added to the buffer. This is repeated until the end of file has been reached. Note that what a given Msg Char is depends on the type of message fi le reading. If the message being read is an English message, then a Msg Char is a single character (letter, digit, etc.). If, however, the message being read is a Morse-code message, then a Msg Char is a Morse Char (i.e., a string containing up to six dot and dash characters). Thus, when getting and putting Msg Chars of a given Input-Buffer and Output-Buffer, the number of characters actually retrieved or placed in the buffer depends on what type of buffer is involved.

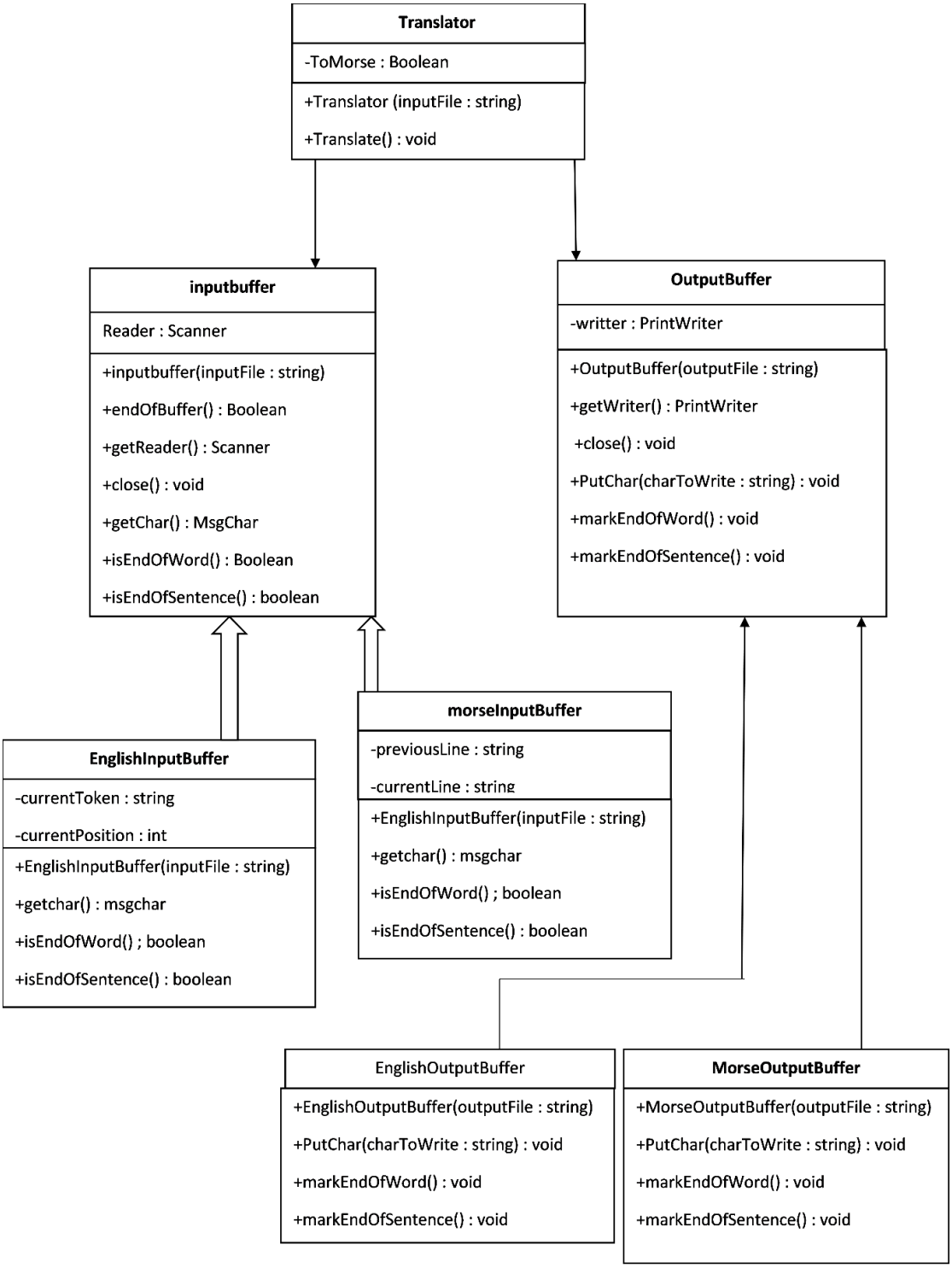


Fig.4.3.1ClassDiagram

4.4 Msg Chars:

The converted value is determined by the Msg Char itself. Whether it is an English Char or a Morse Char, each type of Msg Char has two arrays for converting from its type to the opposite. If an English character appears at a specific index in the English Chars array, then the same index in the morse Chars array will be the conversion, and vice versa. When we get to collections later on in the semester, you will see an alternative to this two-array approach that will make these types of lookups more efficient, however for now, we will use arrays.

You must implement the UML design as given for Msg Chars.

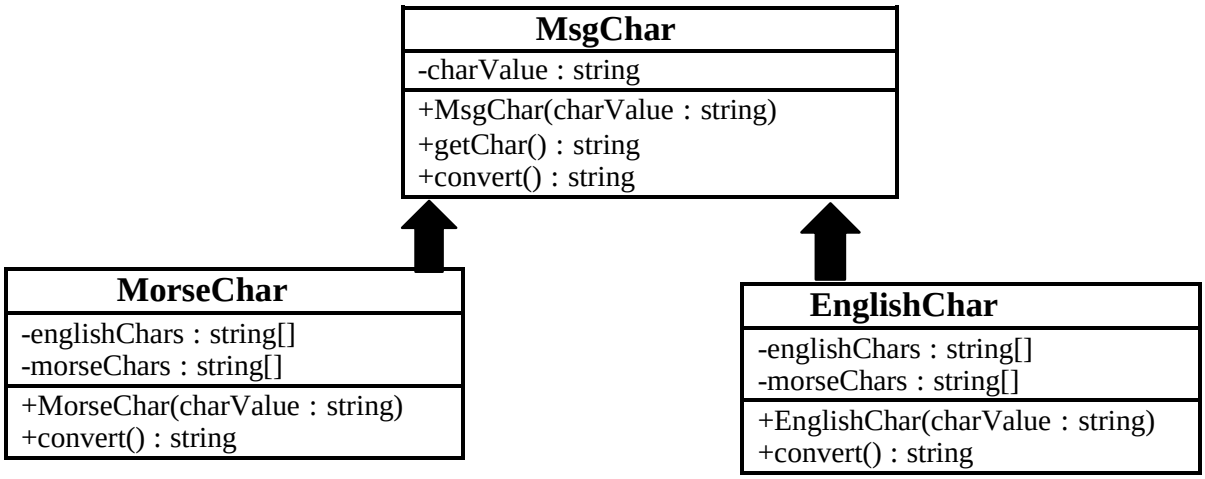


Fig.4.4.1 MsgChars

4.5 Use case Diagram :

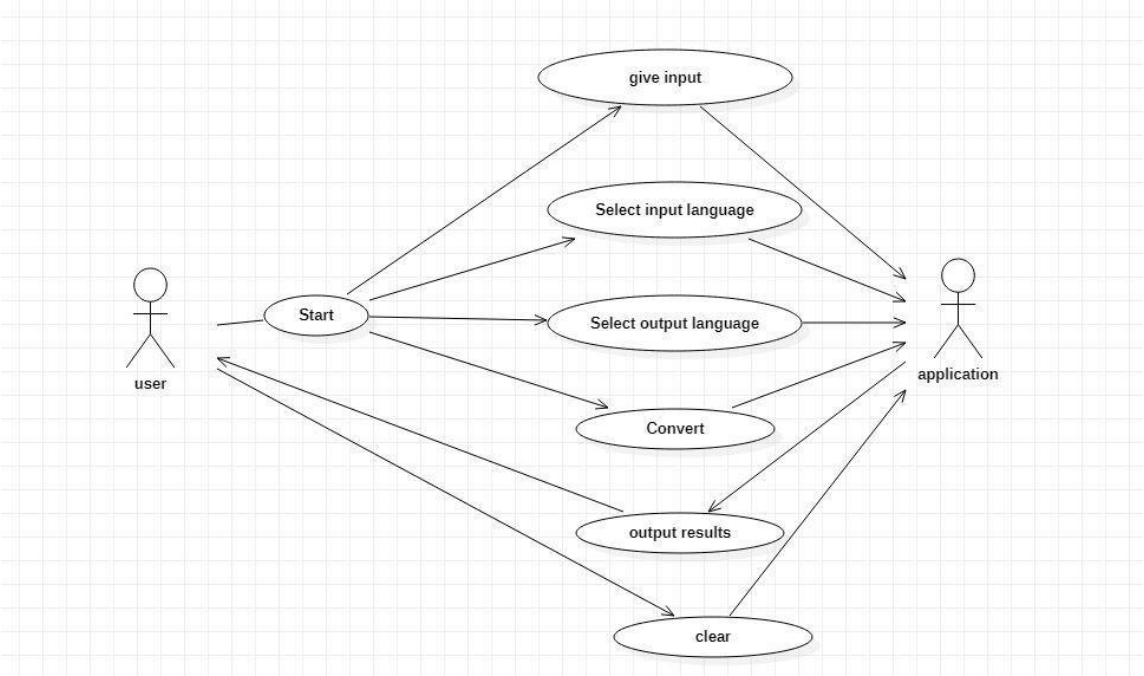


Fig 4.5 use case diagram

4.6 Sequence Diagram

A **Sequence Diagram** simply depicts interaction between objects in a sequential order. The purpose of a sequence diagram in UML is to visualize the sequence of a message flow in the system. The sequence diagram shows the interaction between two lifelines as a time-ordered sequence of events.

- A sequence diagram shows an implementation of a scenario in the system. Lifelines in the system take part during the execution of a system.
- In a sequence diagram, a lifeline is represented by a vertical bar.
- A message flow between two or more objects is represented using a vertical dotted line which extends across the bottom of the page.
- In a sequence diagram, different types of messages and operators are used which are described above.
- In a sequence diagram, iteration and branching are also used.

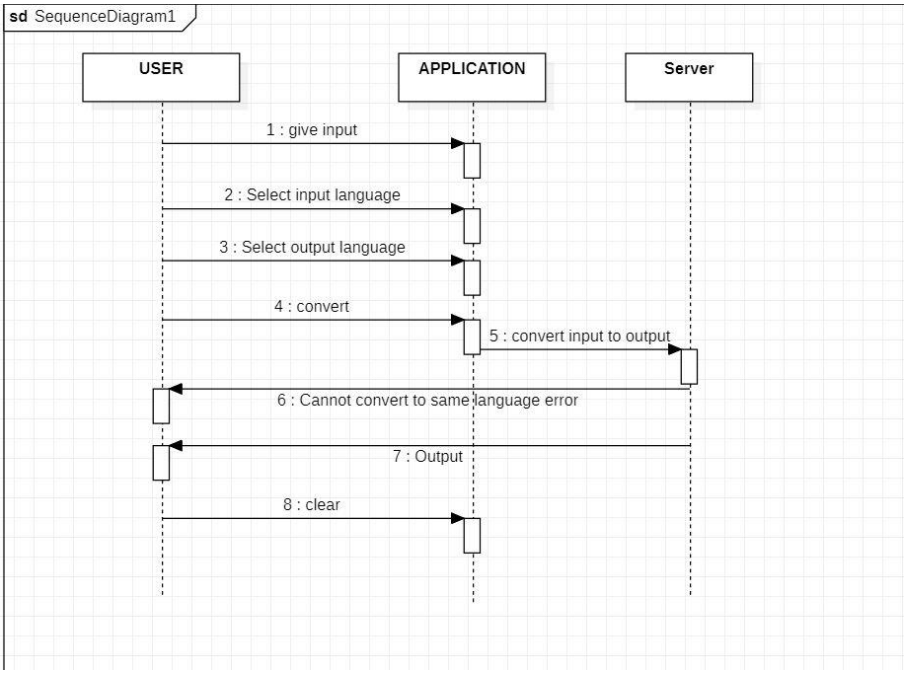


Fig 4.6 sequence diagram

4.7 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.

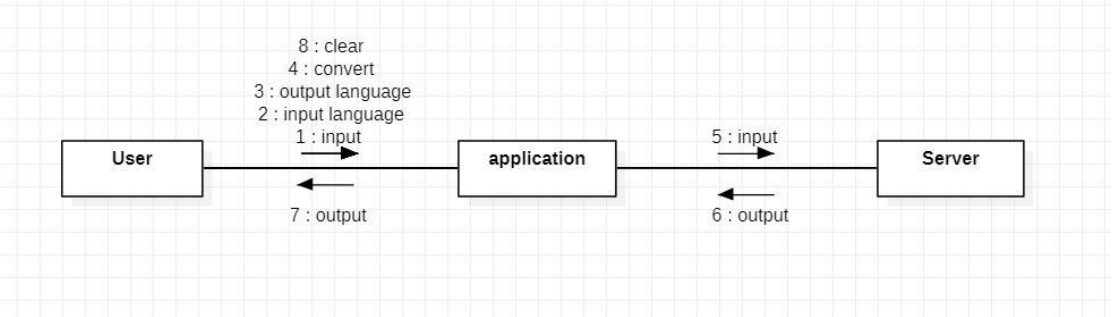


Fig 4.7 Collaboration Diagram

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Introduction

Python provides a data structure called dictionary which stores information in the form of key-value pairs which is very convenient for implementing a cipher such as the morse code. We can save the morse code chart in a dictionary where (key-value pairs)

=> (English Characters-Morse Code). The plain text (English characters) take the place of keys and the ciphertext (Morse code) form the values of the corresponding keys. The values of keys can be accessed from the dictionary in the same way we access the values of an array through their index and vice versa.

```
# Dictionary representing the morse code chart
MORSE_CODE_DICT = {'A': '.-', 'B': '-...', 'C': '-.-.', 'D': '-..', 'E': '.', 'F': '..-.', 'G': '--.', 'H': '....',
                   'I': '..', 'J': '.---', 'K': '-.-', 'L': '.-..', 'M': '--', 'N': '-.', 'O': '---', 'P': '.--.',
                   'Q': '--.-', 'R': '.-.', 'S': '...', 'T': '-', 'U': '..-', 'V': '...-', 'W': '.--', 'X': '-..-',
                   'Y': '-.--', 'Z': '--..', '1': '.----', '2': '..---', '3': '...--', '4': '....-', '5': '.....',
                   '6': '-....', '7': '--...', '8': '---..', '9': '----.', '0': '-----', ', ': '--..--', '.': '.-.-.-',
                   '?': '..--..', '/': '-..-.', '-': '-....-', '(': '-.--.', ')': '-.--.-', ' ': '/', '!': '-.-.--',
                   '@': '.--.-.', '&': '.-...'}

# Function to clear both the text areas
def clearAll() :
    # whole content of text area is deleted
    language1_field.delete(1.0, END)
    language2_field.delete(1.0, END)
```

Fig 5.1.1 shows code used to store Morse code

5.2 Encryption

In case of encryption we extract each character (if not a space) from a word one at a time and match it with its corresponding morse code stored in whichever data structure we have chosen (if you are coding in python, dictionaries can turn out to be very useful in this case) Store the morse code in a variable which will contain our encoded string and then we add a space to our string which will contain the result. While encoding in morse code we need to add 1 space between every character and 2 consecutive spaces between every word. If the character is a space then add another space to the variable containing the result. We repeat this process until we traverse the whole string.

```
# Function to encrypt the string
# according to the morse code chart
def encrypt(message):
    cipher = ''
    for letter in message:
        if letter != ' ':

            # Looks up the dictionary and adds the
            # corresponding morse code
            # along with a space to separate
            # morse codes for different characters
            cipher += MORSE_CODE_DICT[letter] + ' '
        else:
            # 1 space indicates different characters
            # and 2 indicates different words
            cipher += ' '

    return cipher
```

Fig 5.2.1 Encryption

5.2.2 Decryption

In the case of decryption, we start by adding a space at the end of the string to be decoded (this will be explained later). Now we keep extracting characters from the string till we are not getting any space. As soon as we get a space we look up the corresponding English language character to the extracted sequence of characters (or our morse code) and add it to a variable which will store the result. Remember keeping track of the space is the most important part of this decryption process. As soon as we get 2 consecutive spaces we will add another space to our variable containing the decoded string. The last space at the end of the string will help us identify the last sequence of Morse code characters (since space acts as a plaintext (English characters) take the place of keys and the ciphertext (Morse code) form the values of the corresponding keys. The values of keys can be accessed from the dictionary in the same way we access the values of an array through their index and vice versa.

```
# Function to decrypt the string
# from morse to english
def decrypt(message):

    # extra space added at the end to access the
    # last morse code
    message += ' '

    decipher = ''
    citext = ''
    for letter in message:

        # checks for space
        if (letter != ' '):

            # counter to keep track of space
            i = 0

            # storing morse code of a single character
            citext += letter

        # in case of space
        else:
            # if i = 1 that indicates a new character
            i += 1

            # if i = 2 that indicates a new word
            if i == 2 :

                # adding space to separate words
                decipher += ' '
            else:

                # accessing the keys using their values
                # (reverse of encryption)
                decipher += list(MORSE_CODE_DICT.keys())[
                            list(MORSE_CODE_DICT .values()).index(citext)]
                citext = ''

    return decipher
```

Fig 5.2.2 Decryption

5.3 Interface

By using tkinter we crate all interfaces in the application, where we give input and where we get output.

```
# import all functions from the tkinter
from tkinter import *

# import messagebox class from tkinter
from tkinter import messagebox

# Create a GUI window
root = Tk()

# create a global variables
variable1 = StringVar(root)
variable2 = StringVar(root)

# initialise the variables
variable1.set("select language")
variable2.set("select language")
```

Fig 5.3 Interface

5.4 Application implementation

Python provides a data structure called dictionary which stores information in the form of key-value pairs which is very convenient for implementing a cipher such as a Morse code. We can save the morse code chart in a dictionary where (key -value pairs) => (English character – morse code). The plaintext (English characters) take the place of keys and the ciphertext (Morse code) form the values of the corresponding keys. The values of keys can be accessed from the dictionary in the same way we access the values of an array through their index and vice versa. We know that we have hash maps in python which is a form of data used to find the frequency of characters in a string. () Basically a dictionary is like a list instead of integer index it can be of any type.to define a dictionary we use a key value pair with a colon between them Iter () method returns an iterator for the given object .in case of sentinel provided, it returns the iterator object that calls the callable object until the sentinel character isn't found. We need to import the dictionary by calling the function MORSE_CODE_DICT (). We can use any of the python compilers like python 3.7(32bit). or any other online python compilers .so for fast output we can use online python compilers. We also give some error identifications where the system shows us error when the code we given as input is not correct.

Chapter 6
TESTING

6.1 Introduction

In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met,one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended.

Software testing is the technical investigation of the product under test to provide stakeholders with quality related information. Software testing may be viewed as a sub-field of Software Quality Assurance but typically exists independently (and there may be no SQA areas in some companies). In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in the code or deliver faster.

Software Testing is the process used to help identify the Correctness, Completeness, Security, and Quality of developed computer software.

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

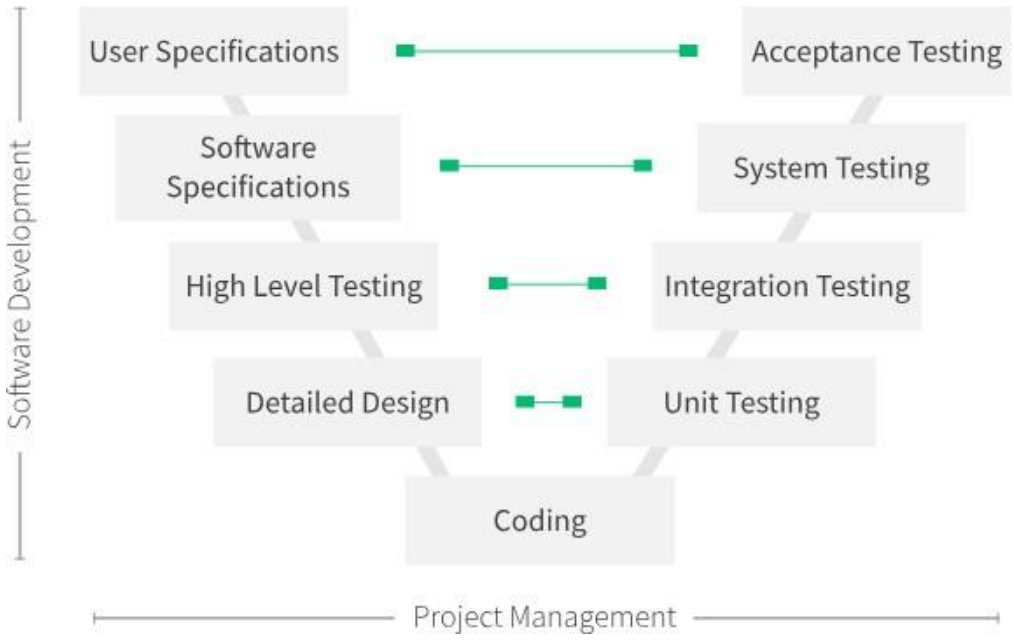


FIG.6 Levels of Testing

6.2 Testing Implementation

To ensure that final interaction part ie.,,Encode and Decode goes well, we do testing step-by-step. Once one basic part passed through the Test, we go to another complex next.

6.2.1 Unit test

It focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design & the process. Specification testing is done to uncover the error within the boundary of the module. All modules must be successful in the unit test before the start of integration testing begins.

6.2.2 Integration Testing

After the unit testing we have to perform the integration testing. The goal here is to see if the module can be integrated properly, the emphasis begins on the testing between module this testing activity can be considered as testing the design & hence the emphases on testing module interaction.

6.2.3 System Testing

Here the entire software system is tested. The reference document for this process is the requirement document, & the goal of operating system to see if software meet sits requirements.

6.2.4 Acceptance Testing

It is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system: the internal logic of the program is not emphasized.

6.2.5 White Box Testing

This is the unit testing method where a unit will be taken at a time & tested thoroughly at a statement level to find the maximum possible error.

6.2.6 Black Box Testing

This testing method considered a module as a single unit and checks the unit as interface and communication with other module rather getting into details at statement level. Here the module will be treated as black box that will take some input and generate output. Output for given set of input combination are forwarded to the other module.

6.3 Testing Plan

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting anomaly for the software engineer.

- **Testing Objective includes**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an undiscovered error.

- **Testing Principles**

  - ❖ All tests should be traceable to end user requirements.
  - ❖ Tests should be planned long before testing begins.
  - ❖ Testing should begin on a small scale and progress towards testing in large.
  - ❖ Exhaustive testing is not possible.
  - ❖ To be most effective testing should be conducted by an independent third party.

6.3.1 Test Cases

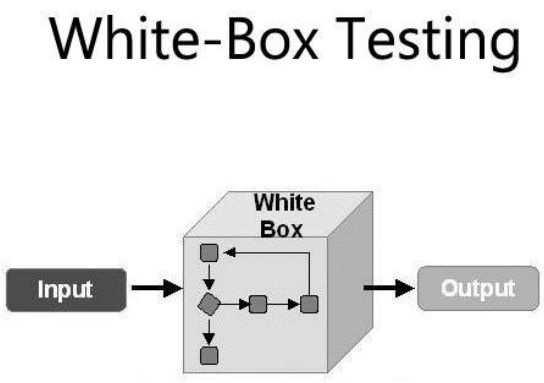
Test Case	Expected Result	Actual Result	Result Status
1	Interface	Interface to give input	Pass
2	Store Morsecode	Morse code for different characters stored	Pass
3	Encoding	Text to Morse code encoded	Pass
4	Decoding	Morse code to text decoded	Pass
5	Output	Show desired output	Pass

Table 6.3 Test Cases

6.4 Test Results

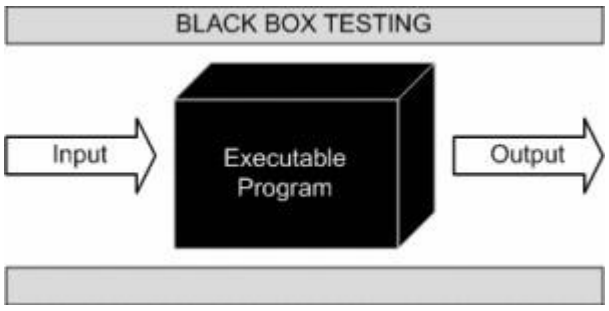
6.4.1 WHITE BOX TESTING (is also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a

system. This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees.



6.4.2 BLACK BOX TESTING, is also known as Behavioral Testing is a software testing method in which the internal structure /design /implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This Method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see.

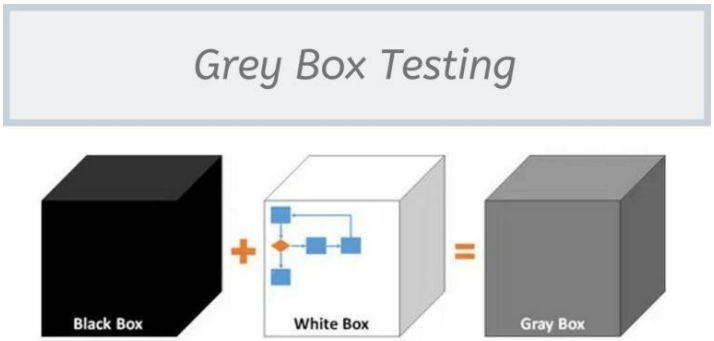


Black Box Testing method is applicable to the following levels of software testing:

- Integration Testing
- System Testing
- Acceptance Testing

6.4.3 GREY BOX TESTING

It is the process in which the combination of black box and white box tonics' are used.



6.5 System testing for present system

In this process of system testing we test whether the system is giving correct output or not.

First we check whether the interface is opening or not. Then we check whether the Morse code is stored correctly for each and every alphanumeric characters.

Later we test whether the system takes the input correctly or not. Then tests whether the application Encodes and Decodes the code correctly or not. Later we test whether we get the desired output or not.

CHAPTER 7

EXPERIMENTAL RESULTS

7.1 Interface Result

In this project first we create an interface where we can give the input and where we can get the output. For this we use tinker to create one of the best user-friendly interfaces.

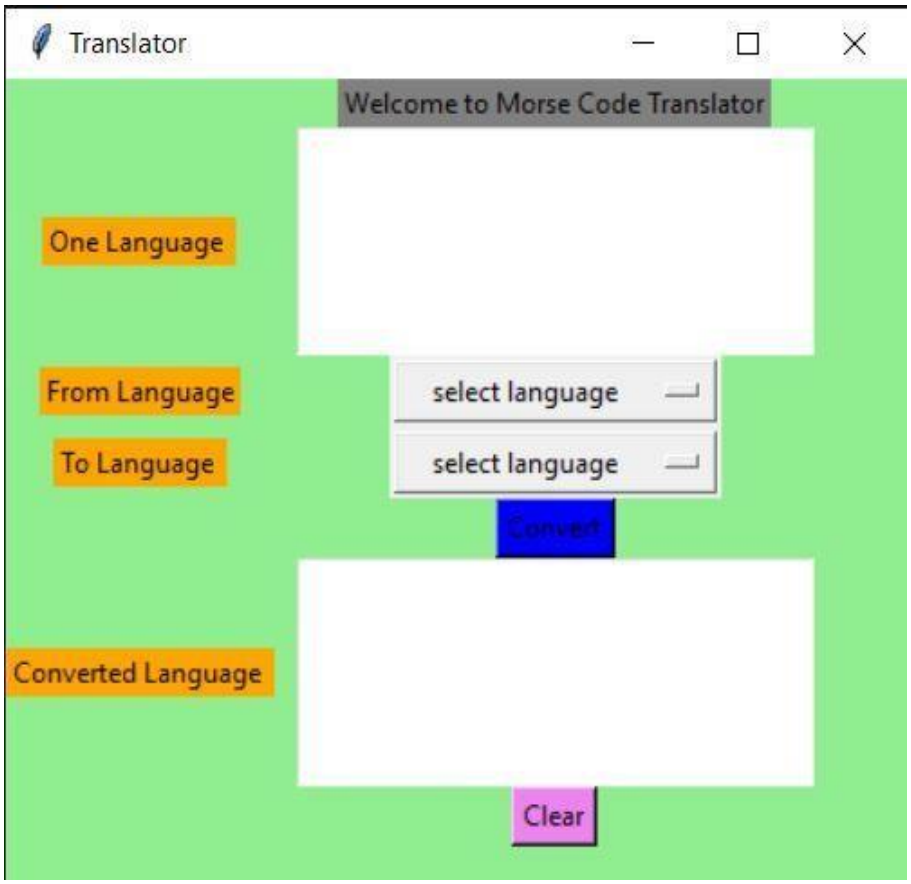


Fig 7.1 Interface Result

7.2 Input and Output Result

This application takes English or morse code as input. We have to specify what input we are going to give. Then we have to select in to which language we want convert the code and the output comes in that format.

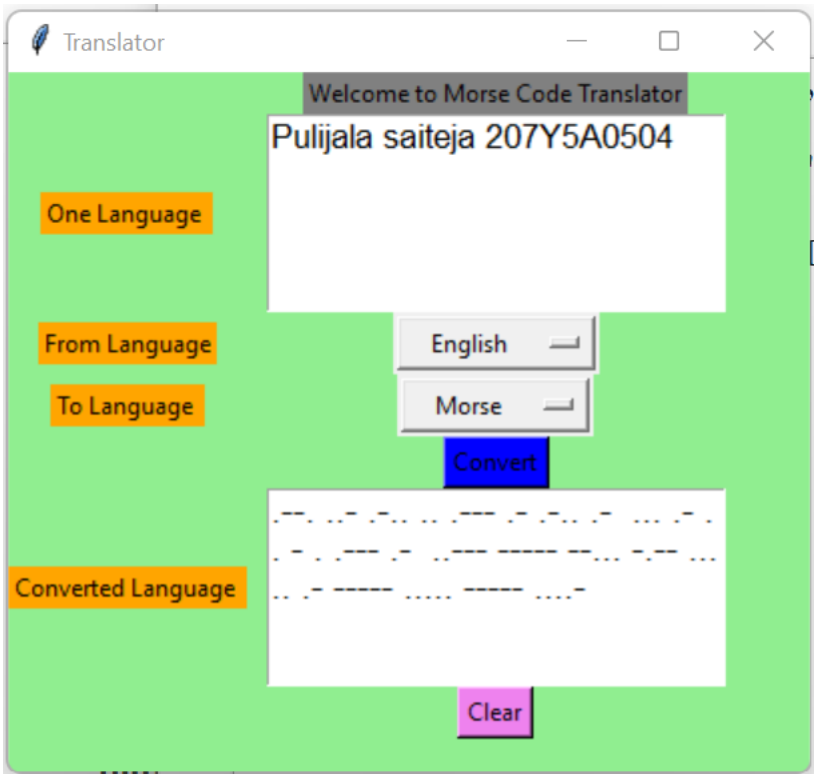


Fig 7.2 Input and Output Result

7.3 Morse Code as Input

We can also get English string as output when we give specified morse code as input. We also want to check whether the given morse code is correct or not. If it is not correct then the system will show an error.

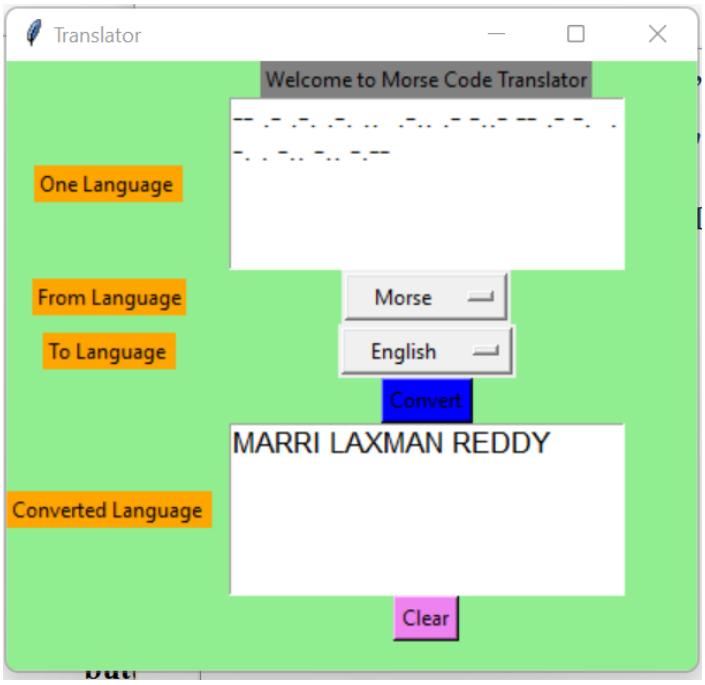


Fig 7.3 Morse Code as Input

7.4 Input and Output as One Language

If we select the input and output as one language then the system shows error. Otherwise, if we give unknown morse code then also it shows error.

Fig 7.4 Input and Output as One Language

Chapter 8

CONCLUSION AND FUTURE SCOPES

8.1 Discussion and Conclusion

Day by day technology makes our better and better. We developed this application as the purpose. This is the implementation of MORSE CODE translator using python. This system was designed to convert message securely within less time. The design makes use of a programming language python, tinker&.ico. This design can be used in different areas like military, external affairs etc…We can also use this application to learn Morse code. By using this system there is no need of tapping device for transmission .so this overcomes the security problems.

The test results demonstrate the functionality of the Morse code converter. The Morse code converter successfully outputs all alphanumeric characters, as shown in fig..,

This proposed system provides an excellent assist tool for Morse code translator.

Morse message for transmission of alphanumeric information with which each character can be translated into a predefined sequence of dots and dashes.

It is a simple & versatile technique to convert text to morse symbols.

We use python dictionary to store specified Morse symbols for each and every alphanumeric characters, including special characters.

We use tinker to create interface where we give input, to get the desired output.

8.2 Difference Between Our and previous Application

There are many applications that can translate light signals to morse code. They use different devices like Arduino, tapping devices. But now in this project we use python language. This is totally depends on software, no tapping devices were used which takes more time.

Features	Our Application	Previous Application
Translation	Yes	Yes, but not work properly
Python language	Yes	No
Security	Yes	Yes, but less secure
Time complexity	More	Less
Encoding and decoding	Yes	Yes, but minor applications

Table 8.2 Difference Between Our and Previous Application

8.3 Scope for Further Development

Developers developing their application for making their application more efficient more effective. We are also developed an application too and it has also scope for future development. Earlier we told that our application is limited. There is a scope for future development. we can combine the encoding and decoding code in to single one.

For the encoding part:

- Optimize the User Interface and make it user friendly.
- Increase the accuracy of sending the Morse code. For the decoding part:

We can optimize the time complexity.

We can optimize the code more efficiently where the spaces can be identified more accurately.

We can also further include the communication process also which provides more security and accuracy.

System requirements:

Software requirements:

Operating System- Windows7 or 10,

Linux: RHEL 6/7,64bit

python idle

Hardware requirements:

- x86 64-bit CPU (Intel / AMD architecture)
- 4GB RAM
- 250GB Hard disk
- MOUSE
- KEYBOARD

REFERENCES

[1] Wikipedia. “Morse code,” Wikipedia.org. [Online]. Available: http://en.wikipedia.org/wiki/Morse_code

[2] Google Play. “Morse Code Trainer,” Google.com. [Online]. Available: https://play.google.com/store/apps/details?id=hunt.morseDit

[3] Apps Zoom. “Morse Code Translator,“ AppsZoom.com. [Online]. Available: http://cn.appszoom.com/android_applications/tools/morse-code-translator_gnfkp.html

[4] Apps Zoom. “Simple Morse Code Translator,“ AppsZoom.com. [Online]. Available: http://cn.appszoom.com/android_applications/tools/simple-morse-code-translator_grsdw.html

[5] Apps Zoom. “SMS2CW – Convert to Morse Code,“ AppsZoom.com. [Online]. Available: http://cn.appszoom.com/android_applications/communication/sms2cw-convert-to-morse-code_ceze.html

[6] Wikipedia. “Morse code,” Wikipedia.org. [Online]. Available: http://en.wikipedia.org/wiki/Morse_code

[7] OpenCV. “OpenCV ABOUT,” opencv.org. [Online]. Available: http://opencv.org/about.html

[8] OpenCV. “OpenCV PLATFORM,” opencv.org. [Online]. Available: http://opencv.org/platforms.html

[9] OpenCV. “OpenC4Android Samples,” opencv.org. [Online]. Available: http://opencv.org/platforms/android/opencv4android-samples.html

[10] Android. “Developers,” android.com. [Online]. Available: http://developer.android.com/reference/android/hardware/Camera.html

[11] Sourya Dey, Keith M. Chugg and Peter A. Beerel presented a paper on Morse code datasets for machine learning university of south California Los Angeles.

[12] Paparao Nalajala, Bhavana Godavarth , M Lakshmi Raviteja, Deepthi Simhadri presented a paper on Morse code generator using Microcontroller using Alphanumeric keyboard, Department of electronics institute of Aeronautical, Hyderabad.

[13] Dr A.Murugan, R.Thilagavathy Associate Professor, PG and Research Department of Computer Science,Affiliated to University of Madras, Chennai, India.

[14] Zhimeng Yin, Wenchao Jiang, Song Min Kim, Tian, Department of Computer Science and Engineering, University of Minnesota, U.S.

[15] Sérgio Silva, António Valente, Salviano Soares, M.J.C.S. Reis ,Jean Paiva, Paulo Bartolomeu School of Sciences and Technology, Engineering Department UTAD, Vila Real, Portugal ,INESC TEC -INESC Technology and Science (formerly INESC Porto, UTAD pole), Porto, Portugal, IEETA -Institute of Electronics and Informatics Engineering of Aveiro, Aveiro, Portugal ,Globaltronic - Electronics and Telecommunications - Águeda, Portugal.