## SYNOPSIS ON

**Morse Code Translator**

**IN PARTIAL FULFILLMENT OF**

**MASTER OF COMPUTER APPLICATION BY**

### Kulkarni Soham Milind

**MCA – I SEM – I, Div~~-~~C**

**Roll No- 23332**

**Under the guidance of**

### Mrs. Deepali Patil

**SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY**

**SINHGAD INSTITUTE OF MANAGEMENT PUNE-411 041**

**2023-2024**

**INDEX**

### 1. Introduction

## 2. Existing System and Need for System

### Existing System

### Need for System

## 3. Scope and Objectives of System

## 4. List of Modules/ Functionalities with description:

## 5 System requirement specification (SRS) (Operating Environment Software and Hardware)

5.1 Server side requirement

### Hardware Requirements

Processor

RAM

HDD

### Software Requirements:

Operating System

Database

Front End

Server Side Script

Software Development Tool

5.2 Client side requirement

### Hardware Requirements:

Processor

RAM

HDD

### Software Requirements:

Operating System

Browser

6. Proposed System

7. Feasibility Study

### 7.1 Technical Feasibility

### Economic Feasibility

### Operational Feasibility

1. Introduction

The "Morse Code Translator" project is a Java-based application aimed at bridging the gap between the rich history of Morse code communication and modern technology.

Morse code is originally developed in the 1830s by Samuel Morse and Alfred Vail.

It had played a pivotal role in telecommunications for decades.

While its practical use has diminished with the advent of digital communication, Morse code remains a fascinating and educational topic for enthusiasts and learners.

This project seeks to create a user-friendly and interactive tool that allows users to encode text into Morse code and decode Morse code back into text. By doing so, it provides an engaging platform for individuals to explore the world of Morse code, fostering both an appreciation for its historical significance and its relevance in specific contexts.

Therefore in simple words, a Morse Code is the collection of the series of dots (.) and dashes (-)

which depicts the text.

In this synopsis, we will look into the key features, technologies used, potential enhancements, and the project's educational value, offering a detailed overview of the "Morse Code Translator" project in Java

2. Existing and Updated System

2.1 Existing System:

The existing system for Morse code translation may consist of traditional methods where Morse code is manually deciphered or encoded by referring to printed reference charts.

It lacks user-friendliness and automation, making it time-consuming and prone to errors. Users need to have a deep understanding of Morse code patterns and symbols to use it effectively.

Challenges in the Existing System:

* Lack of Automation:

The manual nature of the existing system makes it cumbersome and slow, particularly for lengthy messages.

* Error-Prone:

Deciphering Morse code manually can lead to errors, especially for users who are not well-versed in Morse code.

* Limited Accessibility:

Printed Morse code charts are not easily accessible to everyone, limiting their use.

* Learning Curve:

Users need to invest time in learning Morse code and its symbols.

2.2 Updated System:

The updated system, the "Morse Code Translator" Java project, addresses the limitations of the existing system by providing a user-friendly and automated platform.

Key Features of the Updated System:

* User-Friendly Interface:

The updated system features a graphical user interface (GUI) that simplifies Morse code translation for users, eliminating the need for manual decoding.

* Text Input:

Users can input plain text messages directly into the application, making it accessible to a wide audience.

* Real-Time Translation:

The system translates text to Morse code and vice versa in real-time, offering immediate feedback.

* Error Handling:

It includes error handling mechanisms to detect and inform users of incorrect input.

Benefits of the Updated System:

* Improved Efficiency:

Automation reduces the time and effort required for Morse code translation.

* Error Reduction:

Automated translation minimizes the risk of errors in Morse code communication.

* Accessibility:

The user-friendly GUI makes Morse code accessible to a broader audience, including those without prior Morse code knowledge.

* Learning Aid:

The system can aid in learning Morse code, offering an interactive platform for practice.

* Modernization:

The project brings Morse code into the digital age, making it relevant in contemporary contexts.

3. Scope and Objectives

Scope:

The Morse Code Project in Java aims to create a versatile and user-friendly application for translating text messages into Morse code and vice versa. The project's scope encompasses various aspects, including:

* Morse Code Translation:

The primary focus is on accurately translating text to Morse code and Morse code back to text in real-time.

* User Interface (UI):

Developing an intuitive graphical user interface (GUI) to facilitate user interaction, input, and feedback.

* Error Handling:

Implementing error-checking mechanisms to validate input and provide meaningful error messages for incorrect input.

* Educational Value:

Offering a tool for learning and practicing Morse code, making it accessible to users regardless of their prior knowledge.

* Extensibility:

Designing the project to be easily extensible, allowing for future enhancements and additional features.

Objectives:

* User-Friendly Interface:

Create a straightforward and visually appealing user interface to ensure that users can easily navigate the application.

* Text to Morse Code Conversion:

Enable users to input plain text messages, and translate them into Morse code with a single click.

* Morse Code to Text Decoding:

Allow users to input Morse code sequences, and decode them into plain text efficiently.

4. User Interface (UI)

Create an intuitive and user-friendly graphical interface that allows users to interact with the application comfortably.

* Text to Morse Code Conversion:

Provide the capability to input plain text messages, converting them into Morse code using the Morse code mapping rules.

* Morse Code to Text Decoding:

Allow users to input Morse code sequences using dots (.) and dashes (-) and decode them into plain text messages.

* Real-Time Translation Feedback:

Display the translated Morse code or text in real-time as users type or modify their input, providing immediate feedback.

* Audio Output :

Optionally, provide an audio output feature to transmit Morse code messages as sound signals, enhancing the user experience.

5. System Requirement Specifications (SRS)

5.1 Server-side Requirements:

1. Hardware Requirements:

* Processor: A multi-core processor (e.g., Intel Core i5 or equivalent)
* RAM: A minimum of 8 GB of RAM.
* HDD: An adequate storage space, with a recommended maximum of 50 GB for application files, database storage, and logs.

2. Software Requirements:

* Operating System: Windows
* Database: PostgreSQL
* Front End: No specific front-end requirements
* Software Development Tool: Java Development Kit (JDK) 11 or later. Integrated Development Environments (IDEs) like Visual Studio Code for code development, debugging, and deployment.

5.2 Client-side Requirements:

1. Hardware Requirements

* Processor: Modern multi-core processors for responsive client-side performance.
* RAM: A minimum of 4 GB of RAM to ensure smooth operation and responsiveness.

2. Software Requirements:

* Operating System: Windows.

6. Proposed System

The proposed system for the Morse Code Project in Java aims to create a feature-rich and user-friendly application for translating text messages into Morse code and vice versa. Below are the key components and features of the proposed system:

1. User Interface (UI):

- Develop a modern and intuitive graphical user interface (GUI) for the application to enhance user experience.

- The UI should include input fields for text and Morse code, translation buttons, and real-time feedback displays.

2. Text to Morse Code Conversion:

- Allow users to input plain text messages in a designated text input field.

- Implement a translation function that converts the entered text into Morse code based on standard Morse code mappings.

3. Morse Code to Text Decoding:

- Enable users to input Morse code sequences using dots (.) and dashes (-) in a dedicated Morse code input field.

- Implement a decoding function to translate the Morse code back into plain text.

4. Real-Time Translation Feedback:

- Provide real-time translation feedback by displaying the translated Morse code or text as users type or modify their input.

5. Audio Output:

- Optionally, provide an audio output feature to transmit Morse code messages as sound signals, enhancing the user experience.

6. Extensibility:

- Design the project codebase to be modular and easily extensible, allowing for future enhancements and additional features.

7. Feasibility Study

7.1 Technical Feasibility:\*

Technical feasibility assesses whether the proposed project can be successfully implemented from a technical perspective. For the Morse Code Project:

* Technical Expertise:

The project requires Java programming skills, knowledge of Java Database Connection. Ensuring that the development team possesses these skills is crucial for technical feasibility.

* Hardware and Software Requirements:

Ensuring that the specified hardware and software requirements for both the server-side and client-side are available and compatible with the project is essential.

7.2 Economic Feasibility:\*

Economic feasibility assesses whether the project is financially viable. Key considerations for the Morse Code Project include:

* Development Costs:

Estimate the costs associated with hardware, software, development tools, and human resources.

* Operational Costs:

Analyze ongoing operational expenses, including server hosting, maintenance, and potential support and updates after deployment.

7.3 Operational Feasibility:

Operational feasibility evaluates whether the proposed system can be effectively integrated into existing operations and whether it aligns with organizational goals. For project:

* User Adoption:

Assess whether potential users are willing to adopt and use the application.

* User Training:

Evaluate the ease of use of the application