Project Report On

MorseTalk

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Dhiraj Bobby (U2103076)

Jerin Vincent (U2103112)

John Kurian (U2103116)

Kalidas Jayakumar (U2103122)

Under the guidance of

Ms. Janani K

Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report entitled ***"MorseTalk"*** *is a bonafide record of the work done by* ***Dhiraj Bobby(U2103076)***,***Jerin Vincent (U2103112)***, ***John Kurian (U2103116)***, ***Kalidas Jayakumar (U2103122)***, *submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Engineering" during the academic year 2021-2025.*

Ms. Janani K
Project Guide
Assistant Professor
Dept.of.CSE
RSET

Ms. Anu Maria Joykutty
Project Co-ordinator
Assistant Professor
Dept.of.CSE
RSET

Dr. Preetha K G
Professor & HOD
Dept.of.CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **"Dr Preetha K G"**, Head of the Department of "Computer Science and Engineering" for providing us with the opportunity to undertake our project, "MorseTalk".

We are highly indebted to our project coordinators, **Ms. Anu Maria Joykutty**, Assistant Professor, Department of CSE,**Dr. Sminu Izudheen**,Associate Professor, Department of CSE,**Dr. Jisha G**, Associate Professor, Department of CSE, for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Janani K** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Dhiraj Bobby
Jerin Vincent
John Kurian
Kalidas Jayakumar

Abstract

MorseTalk streamlines visual line-of-sight communication using Morse code for users in emergencies when conventionally used communication means such as mobile networks are not operating well or completely off. The mobile app allows users to input text into its user-friendly interface, which is then converted into Morse code using a predefined mapping of characters to Morse signals. Once generated, the Morse code is transferred to the flashlight of the smartphone. Each code, represented by dots (short flashes) and dashes (long flashes), consists of corresponding light pulses emitted from the flashlight. These pulses must be timed appropriately to accurately reflect the structure of the Morse code so that it is clear to read by the recipient device or individual. At the receiver end, the mobile app utilizes image processing techniques to analyze the incoming light signals by counting the number of continuous frames in which the flashlight is detected. Based on this frame count, the system classifies each flash as a dot or dash and decodes the message back into English text. Real-time analysis confirms that even in challenging conditions, decoding remains accurate, thereby enhancing the robustness and reliability of communication without requiring specialized equipment or network connectivity. The application also features an integral interface that educates users about the basics of Morse code, simplifying the initial learning process for new users.

Contents

List of Abbreviations

ROI - Regions of Interest

DFD - Data Flow Diagram

OpenCV - Open Source Computer Vision Library

LiFi - Light Fidelity

RF - Radio Frequency

wpm - Words per Minute

OOK - On-Off Keying

API - Application Programming Interface

UI - User Interface

XML - eXtensible Markup Language

JSON - JavaScript Object Notation

IDE - Integrated Development Environment

List of Figures

List of Tables

Chapter 1

Introduction

MorseTalk project aims to develop a mobile application that facilitates communication during disasters via flashlight signals. The application allows users to input their messages via a textfield. This text message would then be converted to a Morse code which would be ready for transmission. The application also has the capability to receive the flashlight signals and convert it back to text message which would be displayed in the user's screen. Since the application is designed in a way to operate efficiently in offline situations, it can be used in times of disaster.

1.1 Background

Morse code, which was once the most effective means of communication, is not really being used today as it should have been, despite its remarkable effectiveness in emergencies. Most of the modern means of communication fail at very critical times during network failures or due to noises in the communication channel. In such situations, Morse code could still be a simple way of communication. For example, the use of a flashlight to send Morse code signals can be a very effective method of communication. But lack of any modern tools for learning or using Morse code pockets has led to its downfall. In order to bring Morse code back as a practical communication method, the MorseTalk project intends to make use of modern mobile technology to narrow the gap between historical means of communication and contemporary needs. By such means, the project would attain its goal in making Morse code more accessible, relevant, and usable in today's world-all through translation of text into Morse code with real-time flashlight signaling.

1.2 Problem Definition

Morse code has proven its effectiveness in emergency communication throughout history, but it faces significant challenges. People are unable to learn Morse code and practice it fully in their day-to-day lives due to the absence of most of the tools and systems for learning and using Morse code. The imported traditional tools have been proven ineffective on instances such as power outages or poor network conditions, while sound-based signals may fail in environments of high-noise levels. With MorseTalk, this problem is avoided because it uses visual signals to deliver communications so that a simple and effective means of communicating can be available even when mobile networks or voice communication are not available. The project also has an objective of overcoming the problem of teaching Morse code by having a mobile platform that encourages both learning and practical observations in real-world scenarios.

1.3 Scope and Motivation

1.3.1 Scope

- The MorseTalk app converts voice input into Morse code, which is then transmitted using the smartphone's flashlight, and decodes incoming flashlight signals back into readable text.

- This application can be used in real-time communication through mobile devices and, therefore, is handy in learning and emergency situations.

- It offers a user interface towards learning the Morse code by allowing people to practice, visualize, and monitor progress.

- Current scope of the project covers:

  - Control of flashlight for sending Morse code.

  - Real time decoding of flashlight signals into English text.

  - Offline use for remote or network inaccessible areas.

1.3.2 Motivation

- Communication during such emergency situations greatly relies on visible signals as other important points are mostly not available due to the high environment noise or outage in the channel.
- Furthermore, Morse code has faced challenges due to advancements in modern technology that offer alternative communication methods.
- The impetus for MorseTalk is to come up with an application of visual signals that is at once practical and accessible, especially where traditional methods fail.
- Most goals of the application circled around making it much easier and accessible when learning Morse code as well as applicable to today's society particularly individuals in emergency situations where everything else fails.

1.4 Objectives

- Develop an app that lets you convert your text input to Morse code and send it via flashlight.
- Real-time decoding of the Morse code signals transmitted from a flashlight would save them back as English text.
- Provide a learning module for helping users learn and practice Morse code.
- Have offline availability for usage in emergencies totally devoid of internet access.
- Design a simple yet intuitive user interface that makes Morse code easily learnable and usable.

1.5 Challenges

- **Real-time decoding**: Real-time and accurate decoding of flashlight signals has now become a problem quite difficult to tackle under different varying light conditions.

- **Signal Interferences**: Problems with ambient light and multi-light sources in transmission of signals.
- **A user learning curve**: An important factor is to make engaging and user-friendly the learning of Morse code for varied levels of technical knowledge.

1.6 Assumptions

- Users are assumed to have a basic understanding of Morse code or a willingness to learn.
- The platform is compatible with most Android devices that ensure access to a wide variety of smartphones.
- Users should have access to a smartphone with a functional flashlight for transmitting and receiving Morse code signals.
- The platform assumes users are in a line of sight for effective Morse code communication via flashlight signals.

1.7 Societal / Industrial Relevance

The MorseTalk project holds great relevance to society and industry's needs. Morse Codes comes into play for communication in emergency cases when all the other systems fail to work. This mobile modernization introduced by MorseTalk can potentially bridge simple modes of communication using Morse codes in remote areas or disaster-struck zones. In industry, the application is for educating the new generation in the art of Morse code usage for relevance from practice and educational considerations. At the same time, it includes real-time decoding and learning features, thus making it available for a wider variety of user groups, promoting the wider society engagement of the app.

1.8 Organization of the Report

- **Chapter 1**: Introduction concerning the project background, problem definition, scope, and motivation, objectives, challenges, assumptions, and societal/industrial relevance.

- **Chapter 2**: System Design, illustrating how the architecture of the app looks like: from the text to Morse code translation, up to the flashlight signaling parts.
- **Chapter 3**: Methodology, showing approaches such as those used in implementing image processing based on detection of the signals, and real-time Morse code translation.
- **Chapter 4**: Testing and Evaluation, discussing aspects like the testing methods for the accuracy of the Morse code translation and flashlight detection used along with the results and what improvements have been made.

1.9 Conclusion

The MorseTalk project is a mobile application aimed at revitalizing Morse code as a medium of communication using modern technology to make it relevant and practical for today's actual users. Above all, communication is often rendered impossible during critical emergencies when people can not rely on their traditional methods, which is where learning Morse code would be appealing. Within the scope of this development, such integration mainly involves real-time decoding using a flashlight signaling facility bringing about an interactive and accessible platform for actual and learning purposes. Thus, the project aims to actually take the historical means of communication and bring them up to today's technological needs ensuring that Morse code still remains relevant in the modern world and usable by people today.

Chapter 2

Literature Survey

The chapter aims at advances in applications for image processing, optimization for mobile platforms, visible light communication, and modulation techniques using Morse code. An analysis of such research will form the basis of methodology and objectives with strengths, weaknesses, and gaps of research in the various disciplines mentioned.

2.1 Implementation of Mobile Platform Using QT and OpenCV for Image Processing Applications [1]

The research paper titled *"Implementation of Mobile Platform Using QT and OpenCV for Image Processing Applications"* proposes a rather complete system for efficient real-time image processing implementation on mobile platforms with limited resources. This system uses the Symbian OS with Qt libraries and OpenCV in order to manipulate image processing complex algorithms such as edge detection, histogram equalization, and Gaussian smoothing. The important achievement is integrating the cross-compilation tools to make mobile devices adapt image processing libraries. This is significant since, in many cases, it ensures compatibility as well as performance. In consideration of issues such as the limited memory and lack of arithmetic hardware dedicated, the system allows smartphones to perform imaging-intensive computations such as slicing pictures and noise removal. Further, experimental results substantiate the proposed solution in these cases without the actual quality degradation: edge detection and image manipulation on mobile devices. The use of Gaussian filters for noise reduction, histogram equalization for contrast enhancement, are highly emphasized as possibly the most crucial innovations in obtaining better quality of outputs. The demonstration of the system's real-time capabilities for computationally intensive algorithms is further illustrated by using it for edge detection using the Canny method. The discussion also emphasizes trade-offs between processing speed and memory usage with a balance for the resource-constrained device model. This work opens up avenues for other embedded system applications of such techniques: IoT devices may be considered with this technology. The study wraps up with a vision for an extension of the approach to real-time video processing that opens up for further research towards embedding computationally intensive applications in resource-constrained environments.

2.2 LiFi-Based Visible Light Communication [2]

According to the paper *"LiFi-Based Visible Light Communication"*, LiFi has emerged as a strong candidate to supplement and even act as an alternative to RF for short-distance, high-speed data transfers. The paper presented a novel modulation scheme combining the Morse and Manchester coding schemes to encode and transmit data while minimizing the bandwidth it occupied. Noteworthy features include segmentation of transmitted images, morphological operations for enhancing clarity, and usage of visible light for communication secured and interference-free. This ability to transmit data using regular LEDs and photodiodes makes the system most cost-effective and good to be used in versatile environments. Experimental results demonstrate the capacity of the system to achieve transmission speeds of around 1200 wpm (words per minute), keeping minimum error rates. The research is oriented towards the application of LiFi in indoor settings, such as local intranet deployments, a much more secure, high-speed alternative to Wi-Fi. The researchers deal with the problem of alignment dependency and external disturbances and propose solutions according to adaptive algorithms, maintaining system performance. Further, LiFi is advanced to enable smart home technologies in the future for better seamless access and control. The paper also mentions issues of scalability, especially in multi-user scenarios, and proposes improvements in modulation schemes. The authors also mention future applications for LiFi in crucial areas such as healthcare and transport, where secure and efficient communication is paramount.

2.3 A Video Processing-Based LiFi System Using Morse Code-Based Modulation [3]

The paper entitled *"A Video Processing-Based LiFi System Using Morse Code-Based Modulation"* introduces a new technique in Li-Fi communication, that is, data reception through videos rather than photodiode transmission. The method greatly increases the transmission ranges and hence reduces line-of-sight dependency to redefine its entire practicality. The proposed modulation comprises On-Off Keying (OOK) with an added component of Manchester encoding and Morse code to speed up transmission and optimize bandwidth. This inexpensive camera captures light signals using a standard LED bulb, thus showing the feasibility of the system. Experimental configurations lead to a 48-fold gain in decoding speed over existing methods while still keeping transmission errors very low. The aforementioned study exemplifies benefits expected from using cameras as receivers, like catching much larger areas of transmitted light or enabling systems to be more adaptable in dynamic countries. It also speaks to how inexpensive the setup costs since it is built with commodities commercially available. It focuses more on the wider audiences as it's very corruptible. Furthermore, the researchers looked at challenges such as image processing latency and recommended a number of optimization techniques that could speed up the response time of the system. Some of the potential applications discussed include retail environments where Li-Fi will help in inventory management and conducting business transaction activities with customers. The paper concludes with future recommendations of optimizing both hardware and software to achieve higher data rates and longer distances.

2.4 Image Morse Code Text Input System Using Real-Time Mouth Image Recognition[4]

This paper *"Image Morse Code Text Input System Using Real-Time Mouth Image Recognition"*, describes a novel assistive communication interface based on mouth movement detection to be used with Morse code for text input which is focused on people who have very severe disabilities. This system captures real-time dynamic mouth images having a common webcam and it processes them via various image processing techniques such as color space transformation, skin color detection, and morphological operations to isolate and track mouth movements. The mouth movements are used to obtain Morse code signals through fuzzy recognition algorithm which gives the final converted code into ASCII characters to achieve the desired word processing.

Some of the advantages of the proposed system are: it is well adaptable to illumination and background variations; it is very cost effective since it uses commercially available components. The experimental data provided show that the system achieves about 95% accuracy after preliminary training, which implies proved strength and application. The main stress of this research is the use of fuzzy logic to time-based signals, which enhances differentiation between short signals (for dots) and long signals (for dashes). Other features that prove the system suitable for use by such patients for long hours is its non-invasive nature. The ability of the system to work in the midst of slight movements of the subject is accentuated in the paper. Most importantly, the authors give an elaborate lookup table to show how Morse is converted to ASCII, making implementation easily understood by developers. The paper also presents challenges involving latency during rapid inputs, as well as suggestions toward future optimization in hardware and software. The conclusion is that the system has a potential contribution to better accessibility for persons with disabilities.

2.5 Robust Real-Time Automatic Recognition for Maritime Optical Morse Communications[5]

This automated recognition system for optical Morse signals proposed in the paper, *"Robust Real-Time Automatic Recognition for Maritime Optical Morse Communications"* is widely used in maritime communication and presents itself as a remedy for the existing problems related to manual signaling. Signal binarization is achieved using a modified-k means clustering algorithm to dynamically adapt decision thresholds for robust recognition of dots, dashes, and spaces under even the most noise-bearing conditions. The full hardware setup makes use of LEDs, photodetectors, and an MCU microcontroller unit, enabling real-time acquisition of the signals and their decoding.

Validation yielding better than 99% recognition at more than 5dB signal-to-noise ratio shows that the system is extremely robust under most conditions of ambient light interferences. It adopts an easy, flexible free-space optical communication framework, which can be adapted to various maritime scenarios. Very advanced error correcting algorithms guarantee reliable performance in the presence of dynamics and noise. This work advances towards more computationally efficient recognition processes through selection-sort-assisted modifications to traditional k-means clustering. It is designed to allow easy integration into existing shipboard systems with modular hardware design. Future developments will include applications extension for longer-distance communications with much wider bandwidth. Authors will explore deep learning-based approaches to improving decoding accuracy and system robustness. Overall, this paper brings a new dawn into automated optical Morse communications for maritime operations.

2.6 Summary and Gaps Identified

2.6.1 Summary

The table below provides a summary of each paper discussed in the literature review, focusing on their main contributions, advantages, and limitations.

Paper Title	Advantages	Disadvantages
Implementation of Mobile Platform Using QT and OpenCV for Image Processing Applications	Enables real-time image processing on mobile devices; efficient use of limited resources with Symbian OS, Qt, and OpenCV integration	Limited to older mobile platforms; lacks real-time video processing capabilities
LiFi-Based Visible Light Communication	High-speed, secure data transmission using visible light; innovative use of Morse code and Manchester coding for efficient bandwidth usage	Limited transmission range; requires further optimization for higher speeds and broader applications
A Video Processing-Based LiFi System Using Morse Code-Based Modulation	Increases transmission range without line-of-sight dependency; achieves 48x speed improvement in decoding compared to prior methods	Requires further hardware optimization for scalability and reliability

Paper Title	**Advantages**	**Disadvantages**
Image Morse Code Text Input System Using Real-Time Mouth Image Recognition	High accuracy (95%) in real-time mouth image recognition; cost-effective, non-invasive, and adaptable to various lighting conditions	Latency during rapid input scenarios; accuracy influenced by factors like tinted facial features
Robust Real-Time Automatic Recognition for Maritime Optical Morse Communications	Over 99% accuracy in noisy environments; robust clustering algorithm for threshold optimization and error correction	Limited to short-range applications; relies on specific hardware for optimal performance

Table 2.1: Comparison of Papers with Key Advantages and Disadvantages

2.6.2 Gaps Identified

The following gaps were identified in the reviewed literature, indicating areas for further research and development:

1. There needs to be more advancement in terms of memory management and computational efficiencies since mobile platforms are still limited to a small number of optimizations with modern real-time video processing.

2. The necessitate application and set-up robustness with limited transmission range and accuracy for eye-alignment, which will be found in modern visible light communication systems.

3. Hybrid modulation techniques such as the use of Morse code and Manchester coding are highly computational, thus requiring improvements in hardware and software scalability.

4. It suggests investigating adaptive positioning algorithms to improve Li-Fi setups that depend on alignment in terms of usability as applied in dynamic environments.

5. Integrating context-aware and user-personalized capabilities in communication and automation platforms, which would go a long way in enhancing the user experience, is very limited.

2.6.3 Chapter Conclusion

In this chapter, we considered new developments in the areas of mobility platforms, visible light communication, and modulation techniques based on Morse code. These must-have incorporated applications improve productivity, accessibility, and reduce costs. However, there are still opening gaps concerning scalability, real-time communication, and personalization to individual needs. Refill these gaps and move towards creating stronger and more adaptable solutions in the future research and development process.

Chapter 3

Requirements

3.1 Hardware and Software Requirements

3.1.1 Hardware Requirements

The Morse Talk application requires the following hardware specifications to ensure optimal performance on Android devices:

- **Operating System:** Android 7.0 (Nougat) or higher to support modern APIs and features.
- **Memory (RAM):** At least 4 GB to handle real-time image processing and smooth app operation.
- **Internal Storage:** Minimum 8 GB of available space for app installation, cached data, and user files.
- **Processor:** Qualcomm Snapdragon 600 series or equivalent for efficient computation and multitasking.
- **Display Resolution:** At least 720 x 1280 pixels to ensure clear visibility of UI elements and text.
- **Camera and Flashlight:** Required for Morse code detection (via camera) and transmission (via flashlight).

3.1.2 Software Requirements

The development and operation of Morse Talk depend on the following software tools and libraries:

- **Development Framework:** Kotlin and Android Studio for mobile app development, providing a robust environment for building Android applications.
- **Programming Languages:**
  - Python for image processing tasks, particularly for analyzing flashlight signals.
  - Java or Kotlin to implement flashlight control functionality and core app logic.
- **Image Processing Library:** OpenCV for real-time detection of Morse code signals from camera input.
- **Integrated Development Environment (IDE):** Android Studio for coding, debugging, and testing the application.

3.2 Functional Requirements

The Morse Talk application must meet the following functional requirements to fulfill its purpose as a Morse code communication tool:

- **Morse Code Transmission** – Users can send Morse code messages using the device's flashlight with adjustable timing.
- **Morse Code Detection** – The app detects Morse code signals via the camera and decodes them into readable text in real-time.
- **User Interface** – Provides an intuitive interface with a side menu for navigation to features like "Learn," "Quiz," and "Translate."
- **Learn Mode** – Displays a reference table of Morse code mappings for users to study.
- **Quiz Mode** – Offers an interactive quiz to test users' Morse code knowledge with immediate feedback.
- **Translate Mode** – Converts text to Morse code and vice versa, supporting real-time input and output.
- **Persistent Display** – Maintains a history of decoded messages on the home screen for user reference.

Chapter 4

System Design

4.1 System Architecture

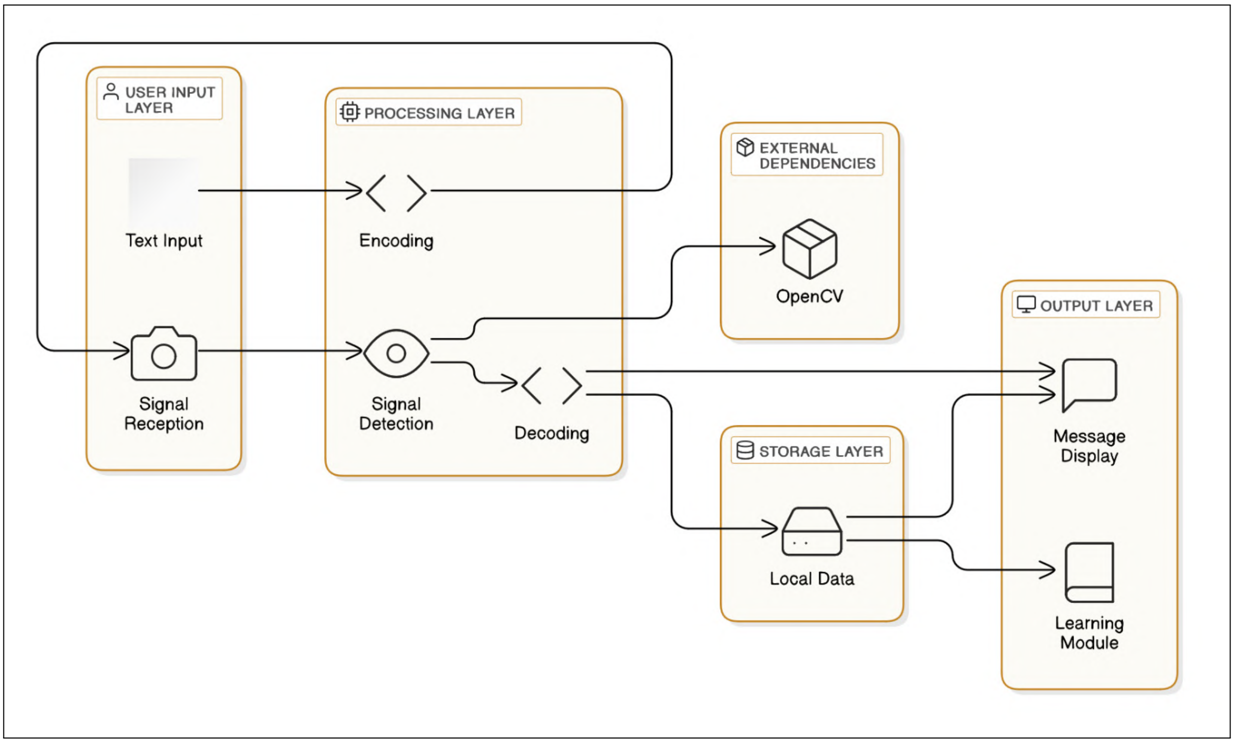


Figure 4.1: Architecture diagram.

4.2 Designing Components

1. **Input Module:** The input will carry the message through text. It will also include an active mechanism for maintaining input robustness through validation mechanisms, ensuring accurate text transmission.

2. **Encoder Module:** The module helps encode text-based messages into Morse code. The operations involved are text normalization; mapping each character as defined

in the dictionary, systematic formatting of Morse code sequences that guarantees the accuracy and consistency of outputs.

3. **Flashlight Control Module:** This is the module responsible for the emission of Morse code through flashlight signals. This module accurately controls the flashlight to produce dots, dashes, and pauses in the signals.

4. **Receiver Module:** It receives flashlight signals through the mobile camera. It then analyzes and processes video frames to detect and isolate light sources and analyze blink patterns. Techniques such as ROI detection and brightness analysis are involved for reliable Morse code decoding.

5. **Decoder Module:** This module converts Morse code into readable text. It processes dots, dashes, and spaces to form letters and words according to a predefined Morse code dictionary, completing the communication cycle.

4.3 Data Flow Diagram(DFD)

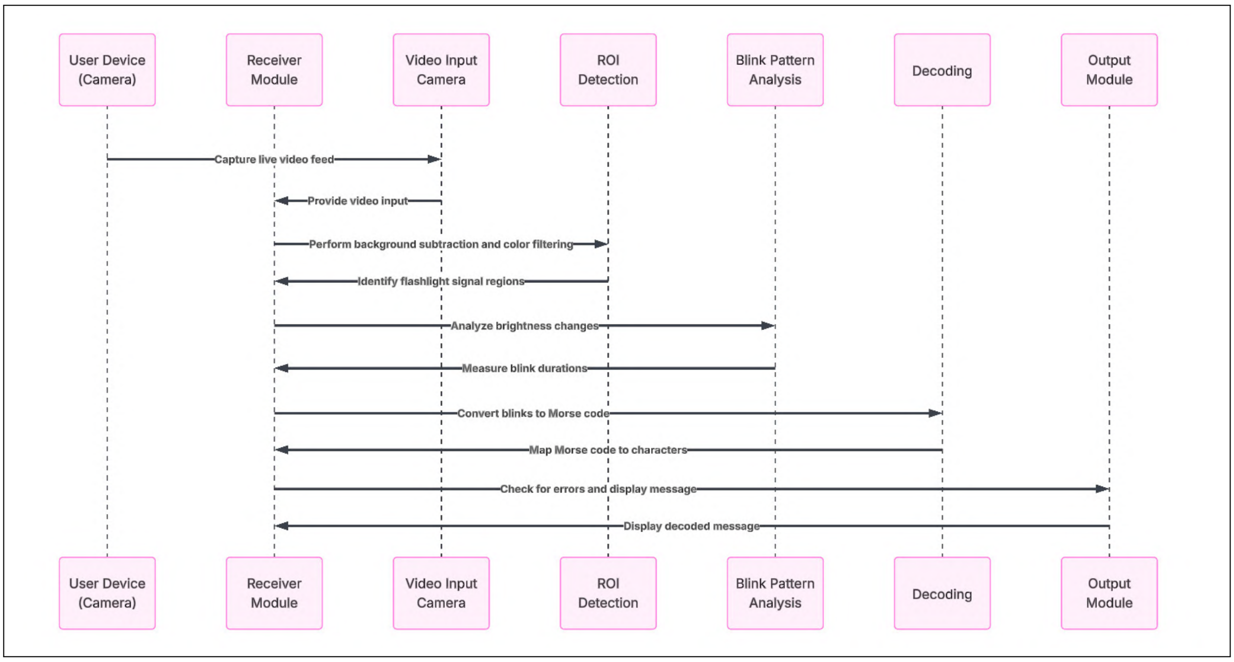


Figure 4.2: DataFlow diagram.

4.4 Tools and Technologies: Software and Hardware Requirements

Software Requirements

- Android Studio for mobile application development.
- OpenCV library for decoding flashlight signals. .
- Kotlin for creating the user interface.

Hardware Requirements

- Smartphones with functional flashlights for transmitting and receiving Morse code signals.
- Devices with cameras capable of capturing flashlight signals for decoding.
- Adequate storage and memory for running the mobile application.

4.5 Modules

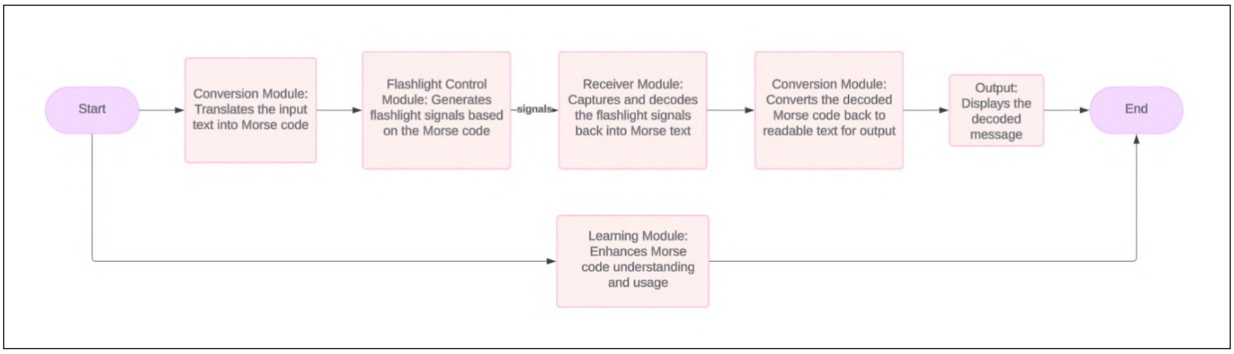


Figure 4.3: Module Flow diagram.

4.5.1 Input Module

The input module enables users to send messages via typing, ensuring seamless and accurate message capture. The typed input is entered through a text field with basic validations such as allowable characters and restrictions on length. When a message is produced, it is standardized into a format. This processed text is then handed over to the next stage for Morse code translation and output, ensuring consistency regardless of how it is input.

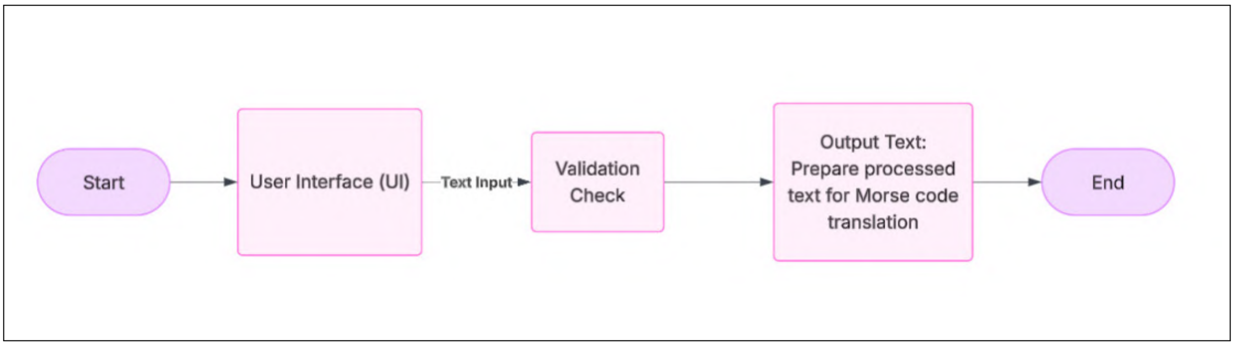


Figure 4.4: Input module.

4.5.2 Encoder Module

The encoder module implements an activity where converting of text to Morse code happens. As with any other modules, it begins with input validation before doing much of the job. Thus, its characters must at least belong to the allowable character set of Morse code which accepts only alphabets (A-Z), numbers (0-9), and some punctuation marks. For example, invalid characters, like special symbols or some emojis, are not converted. The system raises a warning concerning such invalidity for Morse conversion. After this validation, there will be a conversion process that involves getting all input into uppercase form because, in reality, Morse code does not draw any lines between uppercase and lowercase. Hence, this normalization-in-consistency will make a difference in output no matter how the input is presented in terms of casing. Each of these validated characters is then mapped to its corresponding Morse code symbol in a predetermined dictionary, which comprises all alphanumeric and some punctuation characters–such as 'A' to '.-' and '1' to '.—-', which allow for quick lookup during the conversion process.

The coding process assembles the Morse code sequence based on the entire message by joining the Morse symbols of the individual characters. To allow reading easily, it introduces a single space at the separation between the Morse codes of two adjacent characters and three spaces between Morse codes of separate words. This will allow the recipient to decode without effort distinguishing between characters and between words. The end result of the module will be a correctly formatted string in Morse code ready to be sent through the proper transmission lines using a flashlight control module. The output maintains character integrity and word separations to guarantee accurate transmission and decodes.

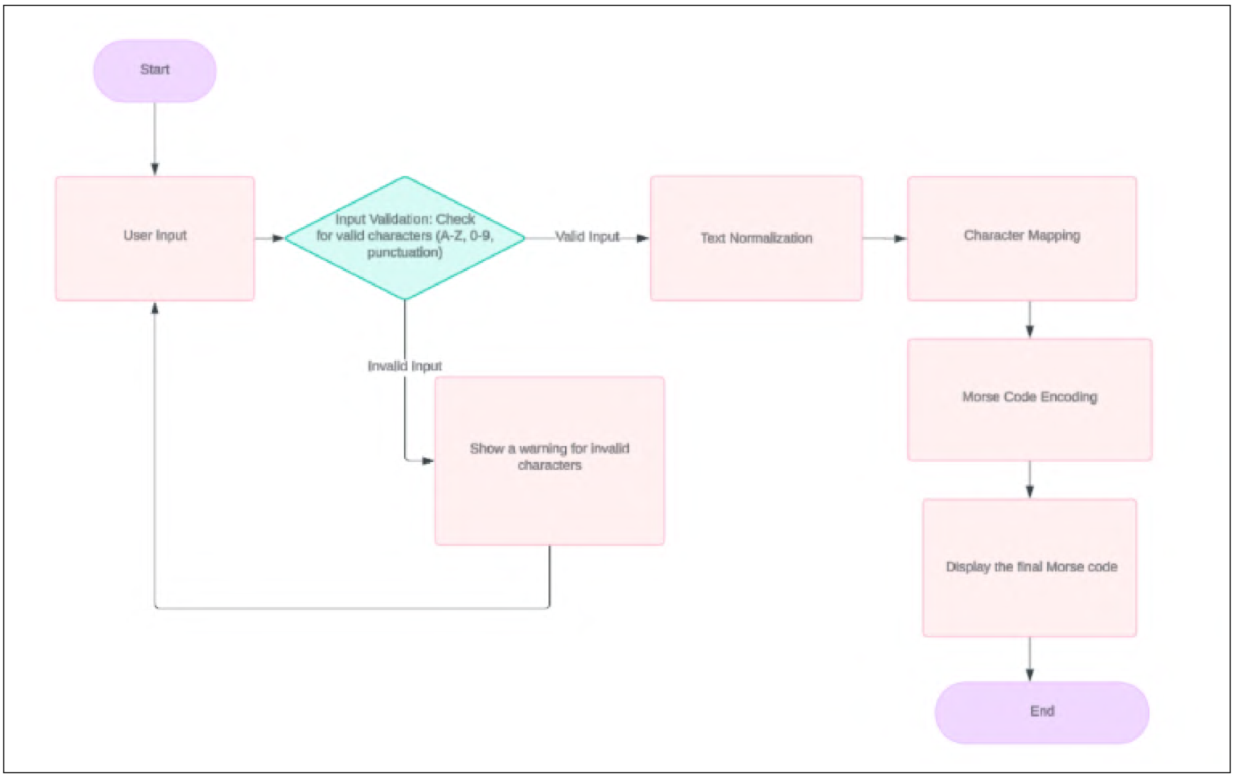


Figure 4.5: Encoder module.

4.5.3 Flashlight Control Module

The module controlling the flashlight transmits Morse code signals using the mobile flashlight according to translated Morse-code sequence. The signals are transmitted with appropriate accuracy by timing the flashes and pauses effectively. Thus, the module converts Morse code, a sequence of dots, dashes, and spaces, into a set of flashlight flashes-the dots being represented by short flashes and dashes by longer ones - with pauses occurring between letters and words and having extra markers to denote the start and end of the message so the transmission is kept intact.

This module uses Kotlin to control the flashlight via Android's Camera2 API. Key functionalities include toggling the flashlight on and off using the setTorchMode method of the CameraManager class. Timing of the signals is regulated using Thread.sleep() to ensure accuracy.

The output of this module is a sequence of controlled light flashes that represent the user's message. This output is then passed to the Receiver Module, which detects and decodes the flashes back into text.

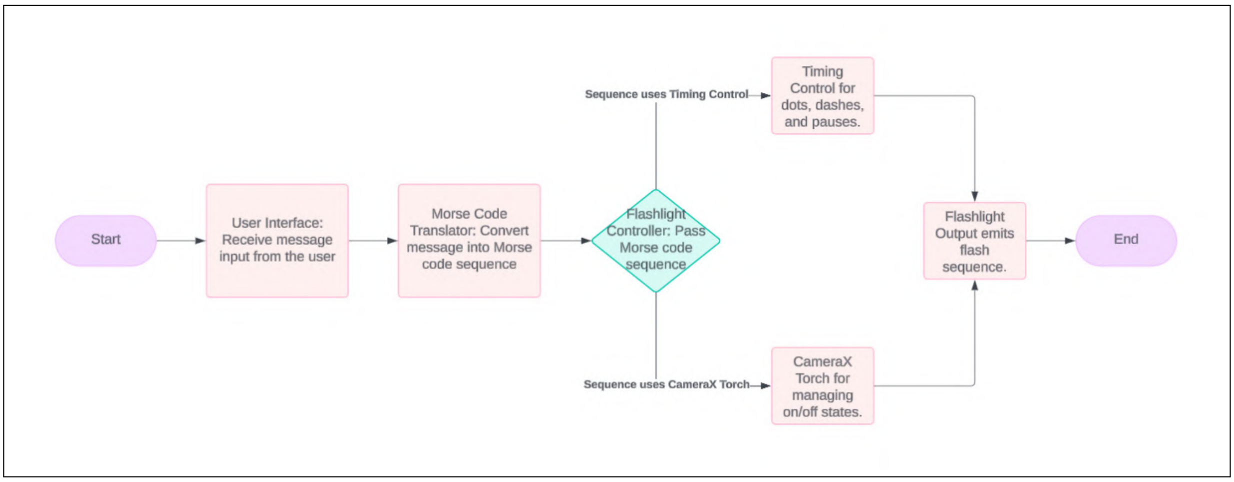


Figure 4.6: Flashlight control module.

4.5.4 Receiver Module

The Receiver Module records live video using the OpenCV VideoCapture function and takes each frame as a snapshot in a sequence. The rate at which frames are captured can be predetermined (for example, 30 fps) for smooth real-time processing. The live video stream is analyzed for changes in light intensity corresponding to Morse code signals induced by rapid brightness changes, such as a flashlight being turned on or off, wherein each frame is analyzed individually in order to detect the rapid change.

The module then identifies Regions of Interest (ROIs) in each frame by first converting them to grayscale to focus on light intensity while ignoring color data. Applying Gaussian blur filters out noises, and background subtraction extracts movement of objects such as lights going on/off from the relative motion picture. These steps yield ROIs that likely contain flashlight signals which are ready for tracking and analysis.

The Blink Pattern includes a significant difference in brightness across two adjacent frames, indicating the on and off states of the flashlight, i.e., the blinks. Minor fluctuations are filtered by a threshold since it should only process meaningful changes in brightness. For blinks, the module records its duration through timestamps in every frame to identify dots and dashes in Morse as short blinks and long blinks respectively. It then groups the pauses into types such as intra-character (within a letter), inter-character (between letters), and word, with the guide of standard Morse timing. This process effectively converts light signals into a sequence of dots and dashes.

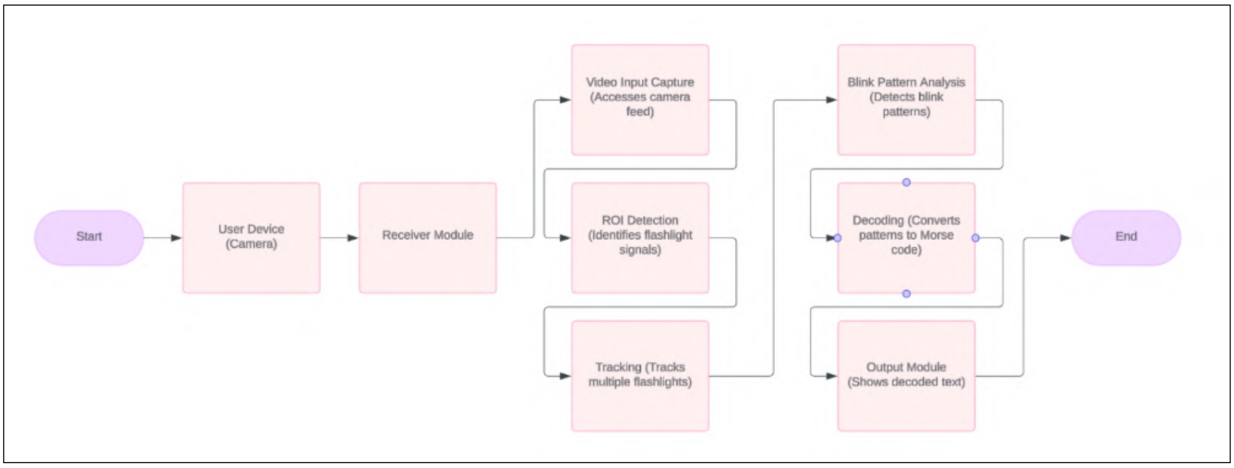


Figure 4.7: Receiver module.

4.5.5 Decoder Module

The module Decoder transforms sequences of Morse code back to text. The Morse code input includes dots (.), dashes (-), single spaces (separating characters), and triple spaces (separating words). For example, the following input is passed into the module for decoding: .... . .-.. .-.. — .– — .-. .-.. -...

The decoding is initiated through signal mapping. The module identifies dots and dashes in the sequence and maps them to the corresponding letters from a predefined Morse code dictionary. Single spaces are interpreted as separators between characters and enable the grouping of dots and dashes into individualized character mappings. Similarly, it has a limit for tripled spaces in words.

After decoding each character, the module assembles them into words, spacing between them. For example: the Morse code input .... . .-.. .-.. — .– — .-. .-.. -.. is converted to text output HELLO WORLD. The last output is nothing but simply a plain text message fully decoded from that Morse code input. It's in a format that can be read easily.

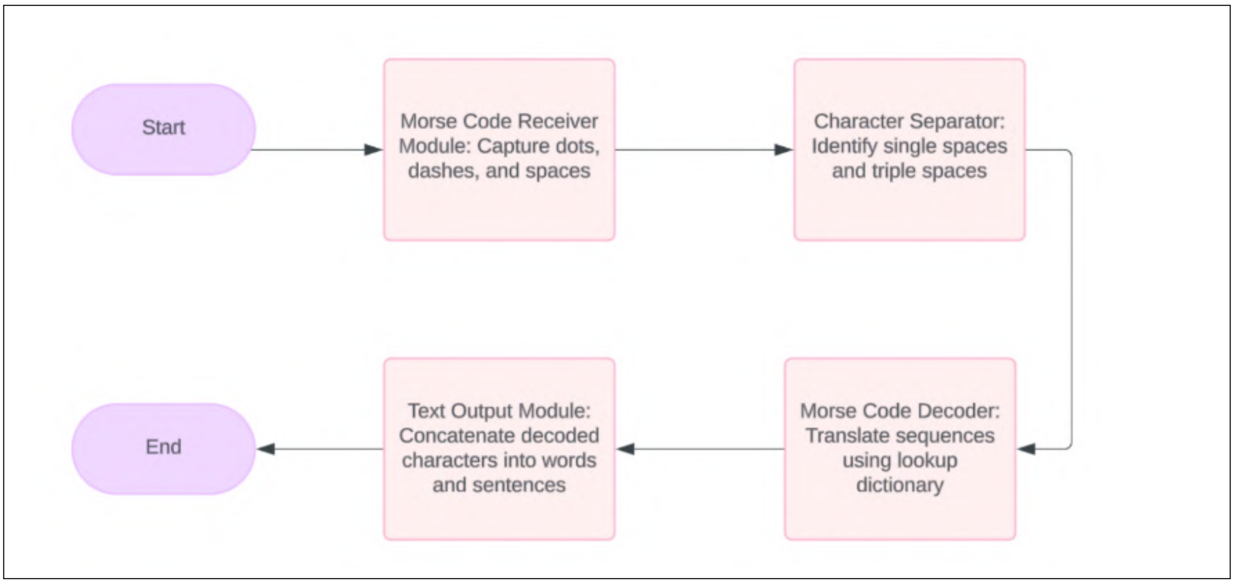


Figure 4.8: Decoder module.

4.5.6 Conclusion

The chapter makes a detailed analysis of the core modules that form the application of Morse Talk. Each of them is designed to facilitate people's messages converted to Morse code and transmitted as such. These modules are discussed:

1. **Input Module:** The input will carry the message through text. It will also include an active mechanism for maintaining input robustness through validation mechanisms, ensuring accurate text transmission.

2. **Encoder Module:** The module helps encode text-based messages into Morse code. The operations involved are text normalization; mapping each character as defined in the dictionary, systematic formatting of Morse code sequences, that guarantees the accuracy and consistency of outputs.

3. **Flashlight Control Module:** This is the module which is responsible for emission of Morse code through flashlight signals. This module accurately controls the flashlight to produce dots, dashes, and pauses in the signals.

4. **Receiver Module:** It receives flashlight signals through the mobile camera. It then analyzes and processes video frames to detect and isolate microscopic light sources

and analyze blink patterns. Involved are the techniques such as ROI detection, tracking algorithms, and brightness analysis for reliable Morse code decoding.

5. **Decoder Module:** This is actually that part of the decoder that produces readable text from Morse code. It will take dots, dashes, and spaces and shape them into letters/words according to a predefined Morse code dictionary and completes the communication cycle.

Integration and Workflow: The integration of these modules ensures seamless operation, enabling real-time message transmission and reception using Morse code. The modular approach not only enhances the project's functionality but also provides a scalable framework for future improvements.

4.6 Work Division

Dhiraj Bobby will handle the text-to-Morse conversion and its transmission using the mobile flashlight. He will develop the logic for translating text into Morse code and integrate the flashlight's use with Android's Camera API. Additionally, he will ensure that the transmission speed is well-balanced and that the flashlight signals follow a controlled pattern. His key deliverables include a reliable Morse code transmission system with configurable settings, thoroughly tested for accuracy.

Kalidas Jayakumar will work on image processing for capturing the flashlight duration. He will use OpenCV to analyze video frames and accurately detect the presence of light patterns corresponding to Morse code signals. His responsibility includes ensuring that the video feed processes light signals efficiently under varying lighting conditions. His deliverables include a robust image processing module that detects and captures Morse code transmissions with high precision.

Jerin Vincent will be responsible for decoding the captured flashlight duration into text. He will implement algorithms to analyze the detected light durations and convert them into readable text, ensuring accurate real-time decoding. His work will also involve optimizing the processing speed and error correction for better accuracy. His deliverables include a fully functional decoding system and comprehensive documentation of the decoding process.

John Kurian will be in charge of designing the user interface (UI) for the project using Jetpack Compose and XML. He will ensure that the application is user-friendly and integrates all the modules seamlessly. He will also conduct end-to-end testing to verify that all components—text input, Morse code transmission, signal capturing, and decoding—work together as a complete system. His deliverables include a polished, responsive UI and a fully integrated application ready for deployment.

4.6.1 Key Deliverables

The project aims to develop a prototype of the MorseTalk system that is completely functional and is intended to display its important features and usages. The heart of such a prototype is a mobile application that translates the text into Morse code. The smartphone flashlight acts like a torch to lit up based on time scales triggering dots and dashes. The other device will have the function of a decoder that receives all the flashlight signals from the other device and displays it in text form, thus depicting evidence of real-time communication.

The mobile application will have a basic accessible interface that will be easy for a wide array of users. It will contain tools for learning and practicing Morse code through interactive and adaptive activities, making the experience fun and effective for users and providing a basic level knowledge of how Morse works .

There will also be a complete documentation accompanying the prototype on system architecture, functionality, and implementation processes, which will serve as a guide for further development and any future deployment in real-world scenarios. This deliverable should indicate that the MorseTalk system would not only be practical but will also add the required user likeness, thus opening doors to further evolution and advancement on the use of Morse code.

4.6.2 Gantt Chart

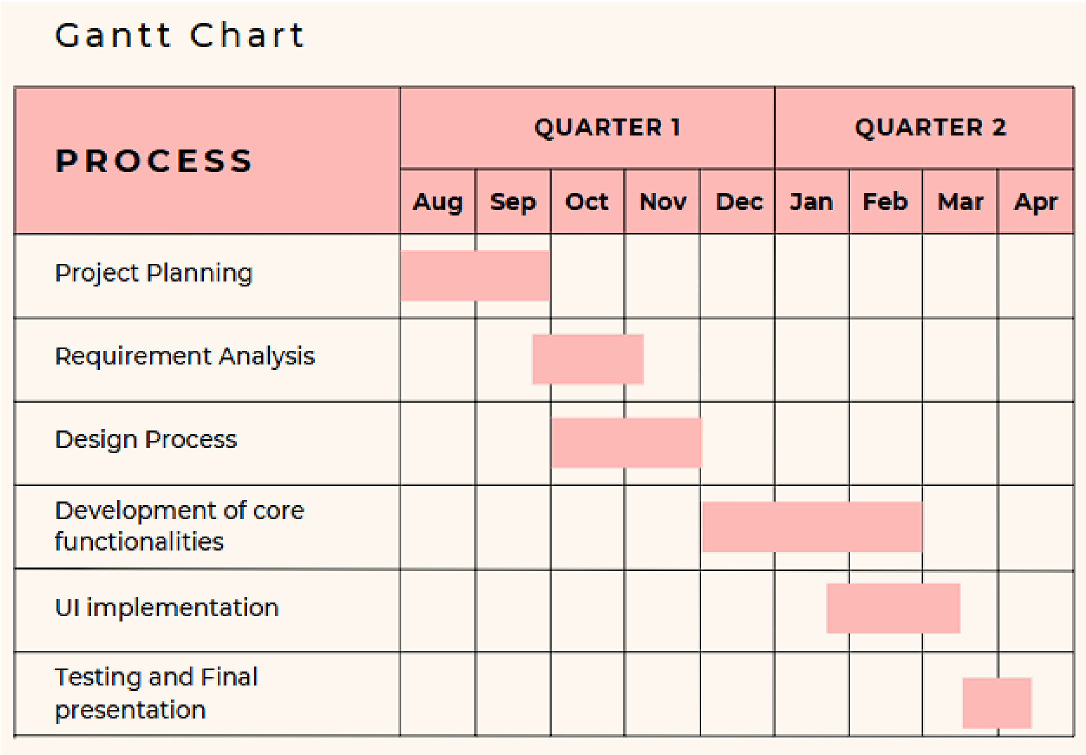


Figure 4.9: Gantt Chart.

Chapter 5

Results and Discussions

5.1 Introduction

This chapter presents the results, analysis, and performance evaluation of the Morse Talk project. The main objective of the project was to develop an efficient Morse code-based communication system using flashlight signals and camera-based decoding. The outcomes are analyzed based on signal transmission accuracy, real-time performance, and error handling.

5.2 Results

5.2.1 Signal Transmission Analysis

The following results illustrate the efficiency of the Morse code transmission process:

- **Flashlight Signal Output:** The flashlight successfully blinked Morse code sequences representing text.

5.2.2 Decoding Performance

- The system successfully reconstructed text from received Morse code signals with high accuracy.
- The decoding module showed minimal latency in processing and translating blinks to readable text.

5.3 Discussion

5.3.1 Comparison with Existing Methods

- Compared to traditional Morse code systems, Morse Talk provides an automated and real-time transmission process.
- The use of camera-based detection enhances the reliability of decoding compared to manual interpretation methods.

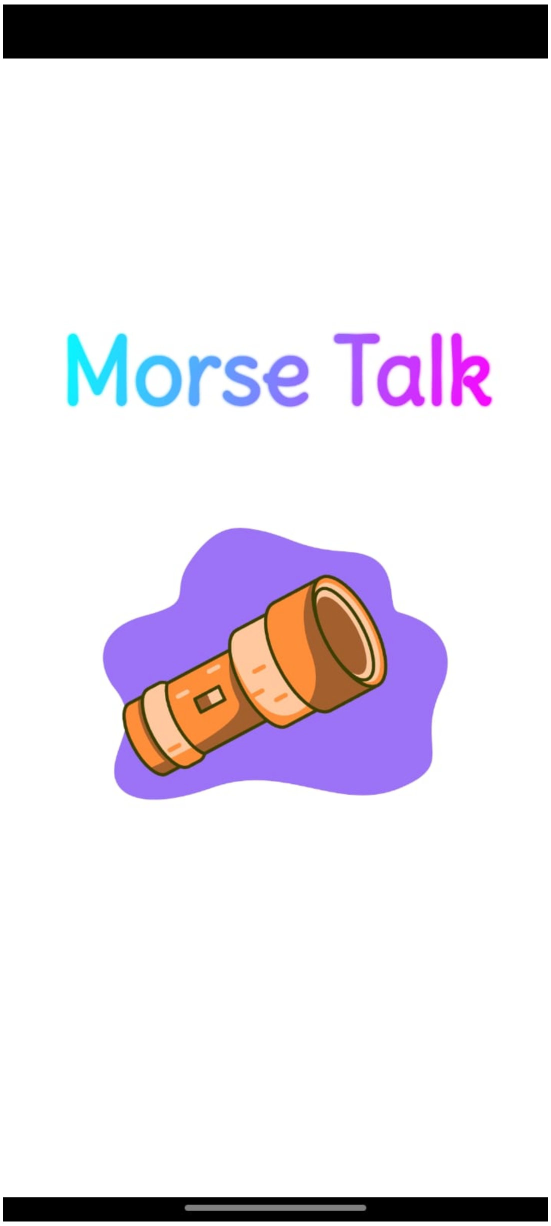
5.3.2 Challenges Encountered

- High ambient light interference sometimes disrupted signal recognition.
- Processing delays were observed in low-light conditions where camera sensitivity needed adjustments.
- The accuracy of the receiver module was affected by fast blinking sequences, requiring refined tracking algorithms.

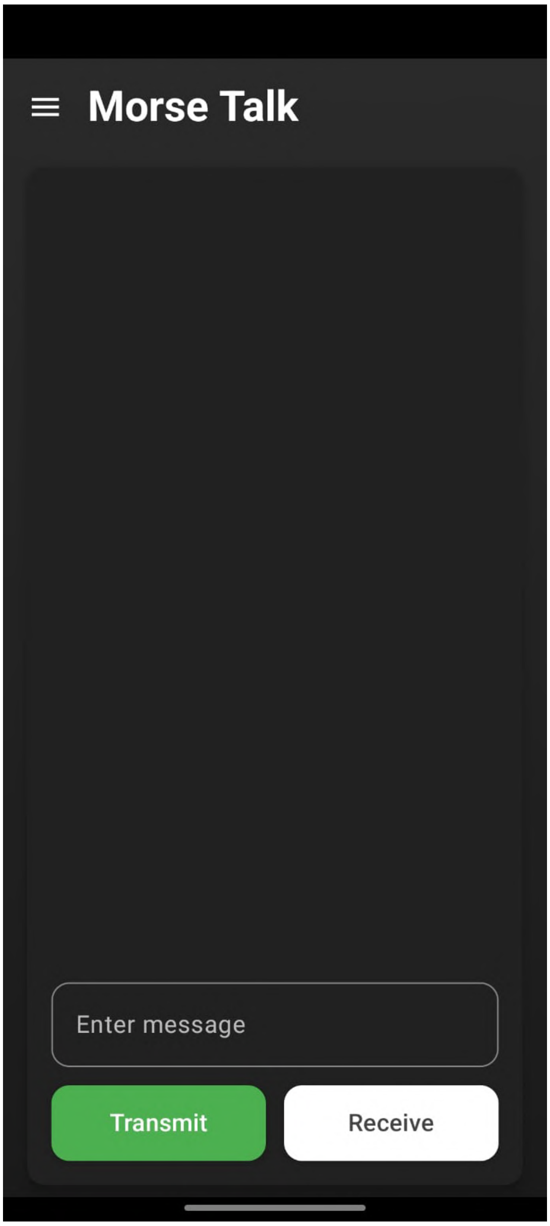
5.3.3 Future Enhancements

- Implementation of adaptive brightness filtering to reduce interference from external light sources.
- Optimization of the receiver module for real-time, high-speed signal decoding.
- Dataset collection for improved machine learning-based recognition.

5.4 Outputs

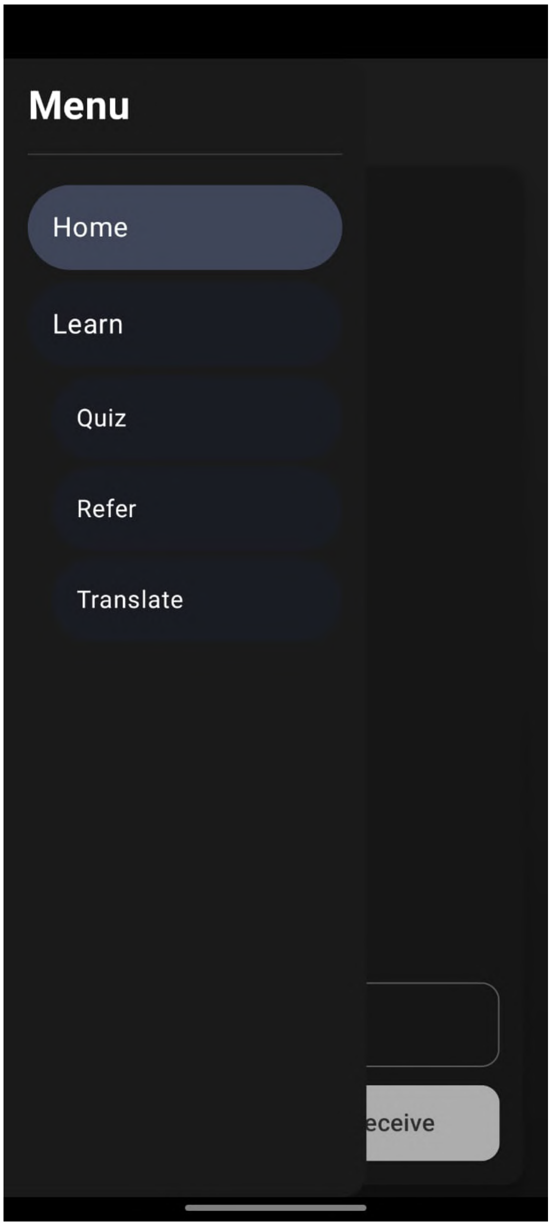


(a) Project Logo



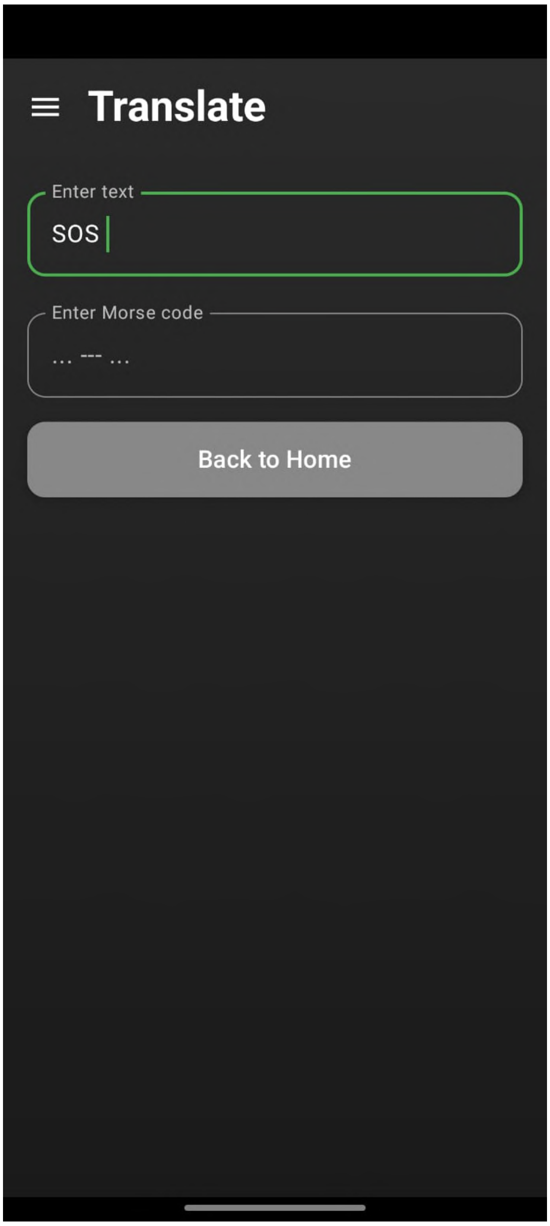
(b) Home Screen

(a) Sidebar Menu

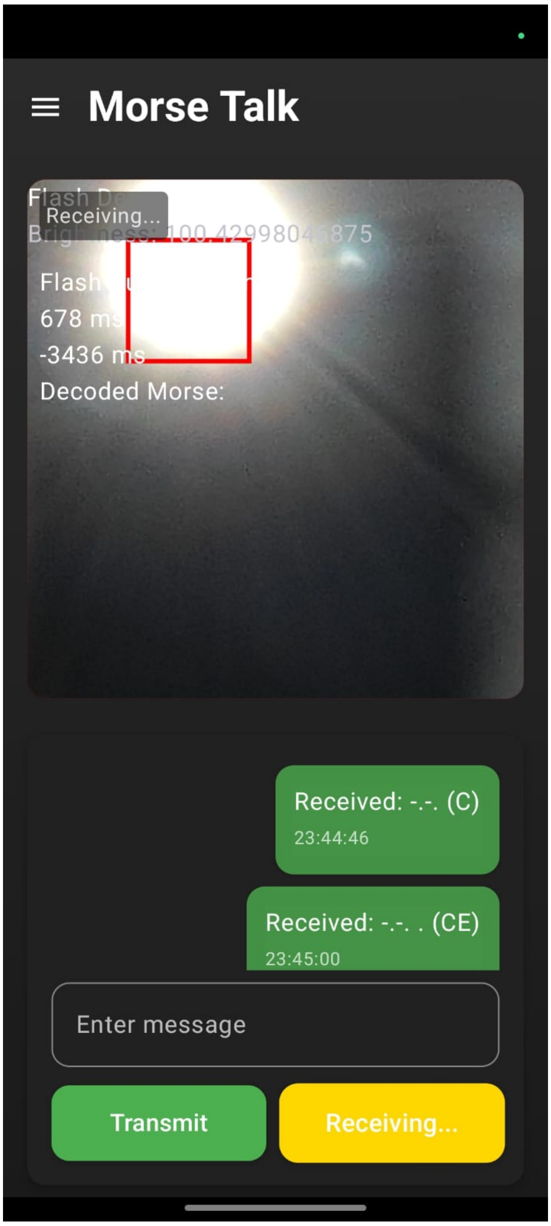


(b) Sidebar Expanded

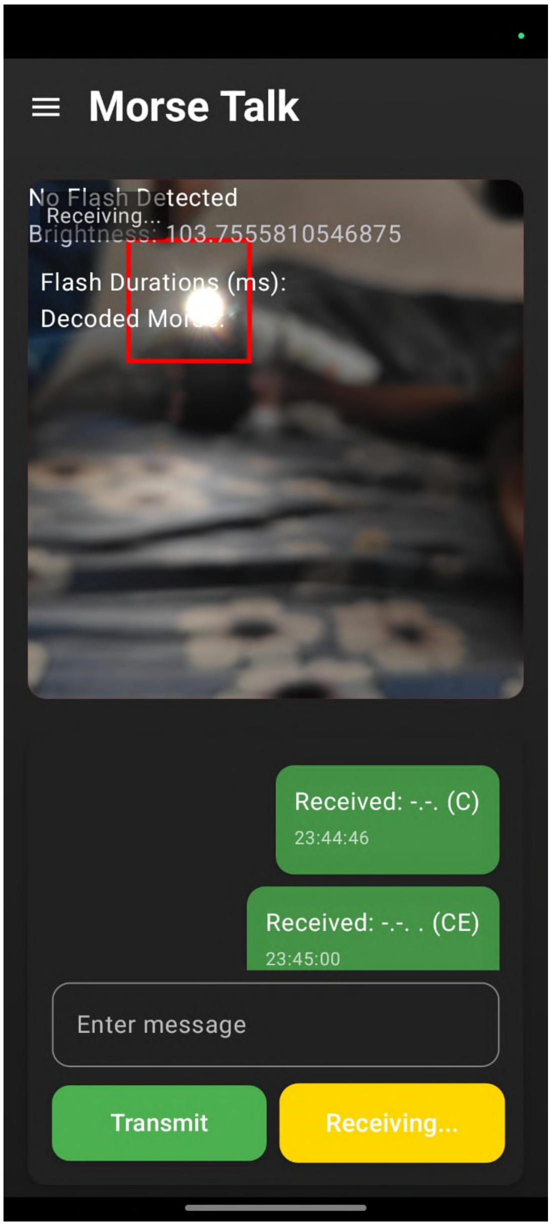
(a) Reference Section



(b) Translation Feature



(a) Flash Detection



(b) No Flash Detected

Chapter 6

Conclusion

The MorseTalk project denotes a considerable stride towards the integration of past forms of communication with contemporary tech reaches into solving major emergent communication hitches. It reinstates Morse Code into the apparatus using flashing signals and processing images at sophisticated levels to bring offline communications possible when all other technologies fail. Its introduction pointed out how relevant this was and thus set the stage for producing a simple yet versatile mobile application.

The literature survey revealed a treasure trove of information-like other works and technologies. State-of-the-art development for visible light communication, image processing, and signal modulation has been reviewed. The merits of these methods were rendered while gaps-in terms of scalability, real-time processing, and adaptability-have been pinpointed and targeted by this project. The effort is expected to morph problematic scenarios into accessible, dependable, and functional experiences by drawing from previous studies while integrating with innovative approaches.

A detailed chapter on methods gave a modular perspective of the entire system for design and implementation, highlighting how well it is integrated. From getting the user's input via text and encoding and transmitting Morse signals through flashlight to decoding light patterns back to text using advanced image processing techniques, the project promises a whole seamless system. The application is designed to work offline for reliability even in the absence of a network, along with having a very user-friendly interface for learning and practical usage of Morse code.

References

[1] R. Deepthi and S. Sankaraiah, "Implementation of mobile platform using qt and opencv for image processing applications," *2011 IEEE Conference on Open Systems (ICOS2011)*, 2011.

[2] S. J. Kulkarni, A. P. Ambekar, K. N. Gopale, P. V. Bhandare, and R. R. Chakre, "Lifi based visible light communication," *International Research Journal of Engineering and Technology (IRJET)*, 2021.

[3] S. R. Rupanagudi, U. Mishra, V. G. Bhat *et al.*, "A video processing based lifi system using morse code based modulation," *IEEE*, 2020.

[4] S.-C. Chen, C.-M. Wu, and S.-B. Su, "Image morse code text input system using real-time mouth image recognition," *4th International Conference on Intelligent and Advanced Systems (ICIAS)*, 2012.

[5] X. Wang, M. Zhang, H. Zhou, X. Lin, and X. Ren, "Robust real-time automatic recognition for maritime optical morse communications," *Applied Sciences*, 2020.

Appendix A: Presentation

MorseTalk

INNOVATIVE APPROACH TO EMERGENCY COMMUNICATION DURING DISASTERS

Dhiraj Bobby Jerin Vincent John Kurian Kalidas Jayakumar

CONTENTS

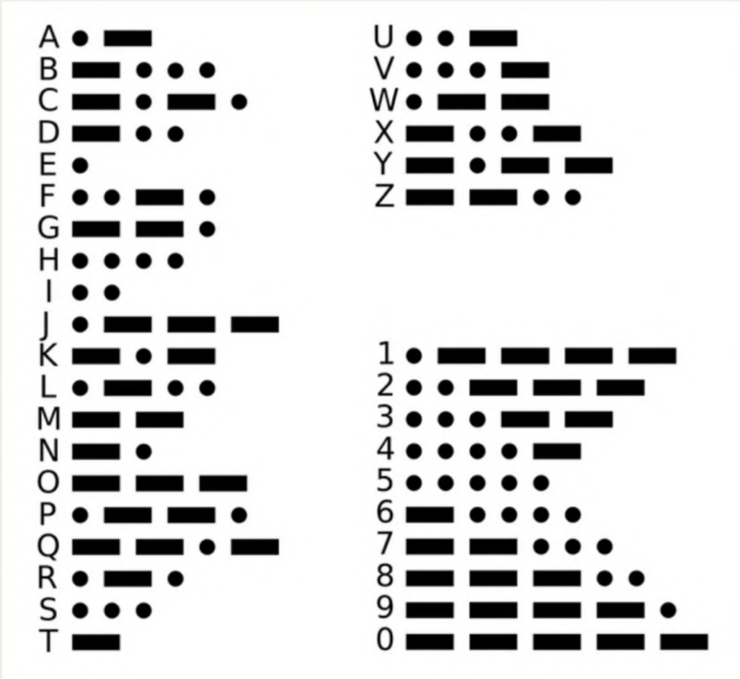
PROBLEM DEFINITION

- In emergencies, individuals may need to communicate with someone visible in their line of sight but beyond verbal range.
- Mobile networks may be unreliable, and shouting can be ineffective due to distance or noise.
- A simple, effective, and accessible visual communication solution is essential, without needing specialized equipment or a stable network.

PROJECT OBJECTIVE

- The project aims to develop a mobile app using Morse code via a smartphone’s flashlight for visual communication in emergencies.
- The app enables quick, effective message transmission when traditional communication methods are unavailable.
- It includes an interactive learning feature to teach Morse code, enhancing accessibility and engagement.

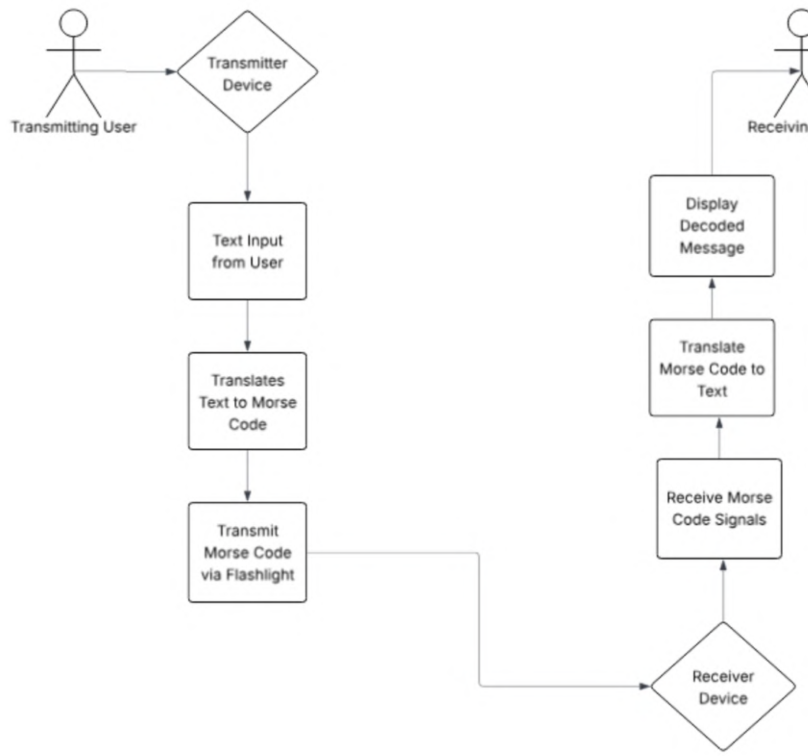
MORSE CODE CHART

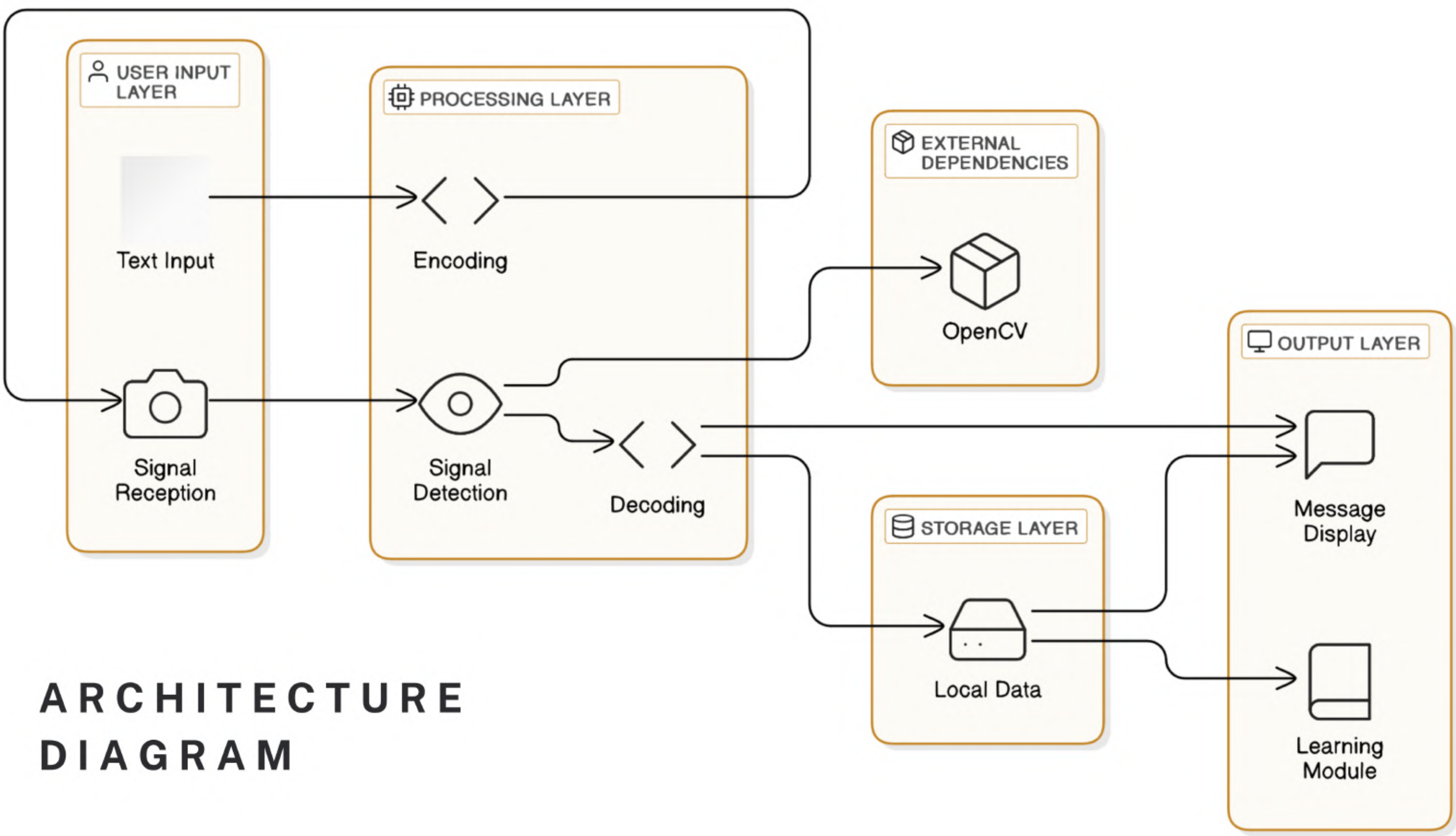


MODULES

1. **INPUT MODULE**
    - Captures user input through text and validates the input.
2. **ENCODER MODULE**
    - Translates valid text input into Morse code, ensuring proper character mapping and formatting for accurate transmission.
3. **FLASHLIGHT CONTROL MODULE**
    - Generates Morse code signals by controlling the mobile flashlight with precise timing for dots, dashes, and pauses.
4. **RECEIVER MODULE**
    - Captures and decodes flashlight signals, utilizing video processing techniques to translate light patterns into readable text.
5. **DECODER MODULE**
    - Converts the Morse code generated by receiver module back to text .

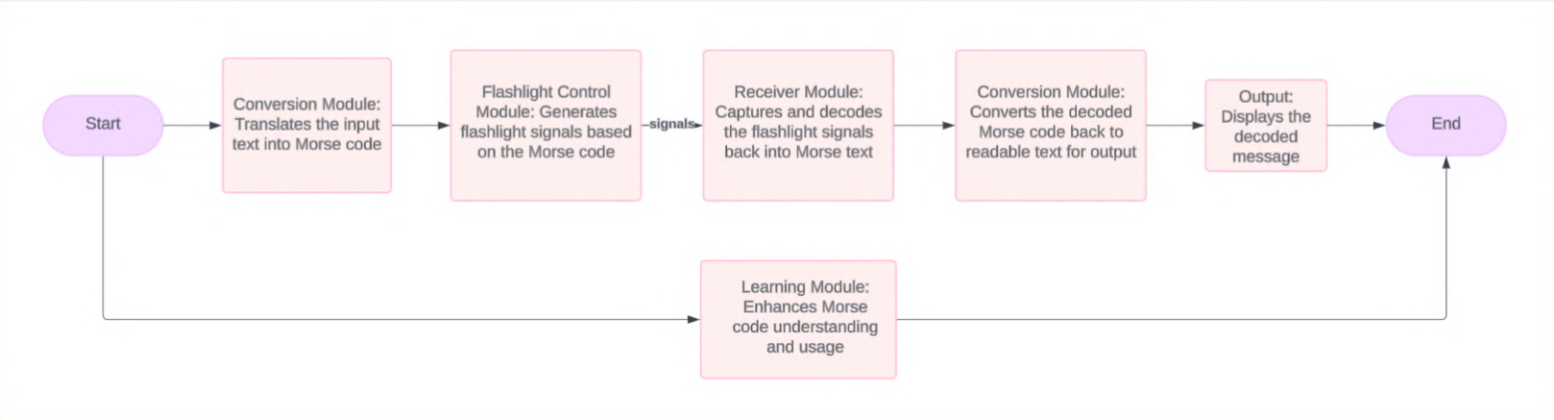
USE CASE DIAGRAM





ARCHITECTURE DIAGRAM

MODULES



INPUT MODULE

Text Input Handling

- If the user opts to type the message, the module accepts the input through a text field, ensuring basic validation such as allowable character checks and length constraints.

- The processed text string is then passed to the next stage, which involves translating the message into Morse code. The output ensures consistency and facilitates accurate Morse code translation.

ENCODER MODULE

- This module focuses on the process of translating text into Morse code.
- Input Validation: Ensures only valid characters (A-Z, numbers, some punctuation like commas, question marks) are processed.

ENCODER MODULE

Text Normalization

- Text Normalization: Standardizes input for processing.
- Case Handling: Converts all text to uppercase since Morse code ignores case.
- Uniformity: Ensures consistent output (e.g., "Hello" or "HELLO" yields same result).

ENCODER MODULE

Character Mapping

- Character Mapping: Maps each character to its Morse code symbol using a dictionary.
- Dictionary: Predefined, hard-coded with mappings (e.g., ‘A’ to “.-”, ‘1’ to “.----”).
- Process: Quickly looks up each input character and returns its Morse equivalent.
- Efficiency: Hard-coded dictionary ensures fast, reliable conversions.

ENCODER MODULE

Encoding Process

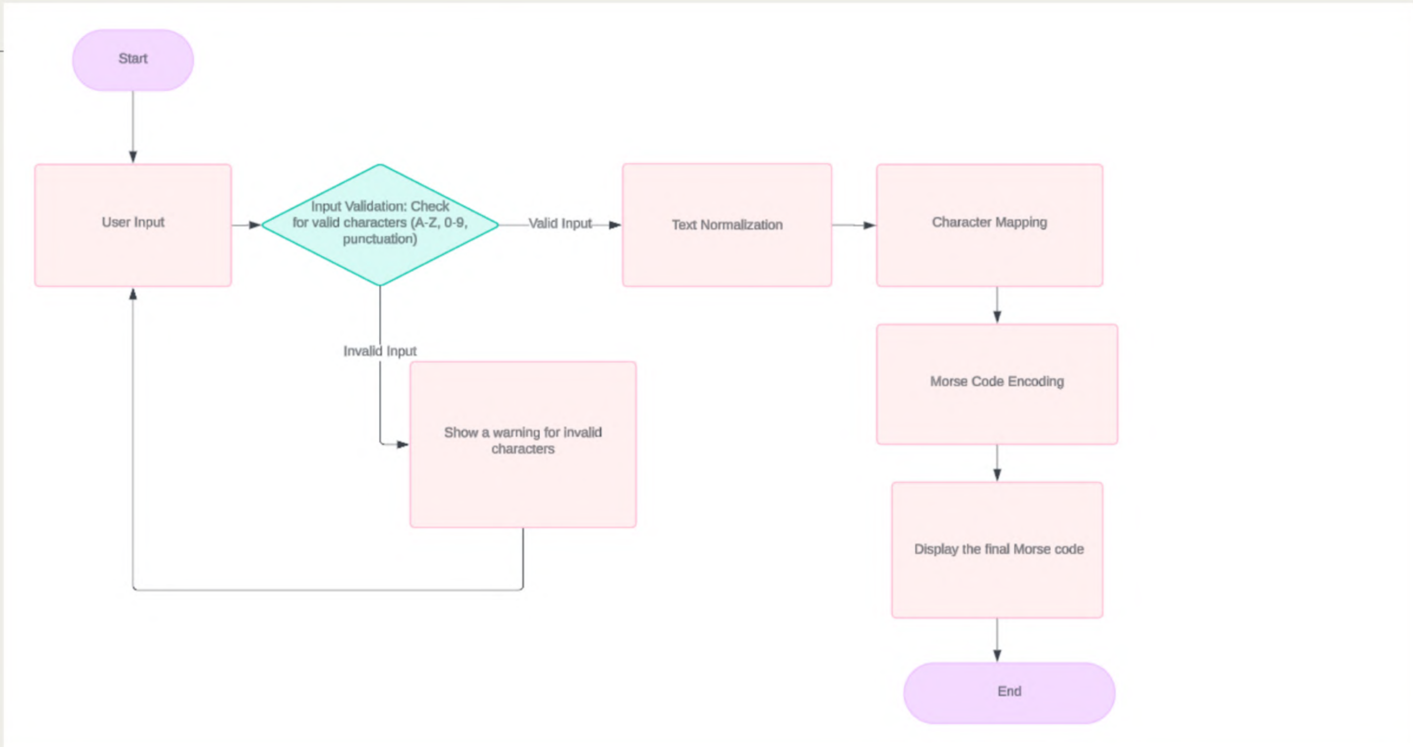
- Encoding Process: Builds Morse code sequence for the full message.
- Concatenation: Combines Morse symbols for each character.
- Character Separation: Adds one space between each character’s Morse code.
- Word Separation: Added a morse code symbol(..--) for word separation.

ENCODER MODULE

Module Output

- The final output of the Conversion Module is a properly formatted Morse code string, ready for transmission using the flashlight control module.
- The output maintains the integrity of both individual characters and word separation for accurate transmission and decoding.

ENCODER MODULE



FLASHLIGHT CONTROL MODULE

Morse Code Signal Generation

- This module controls the flashlight to emit Morse code signals based on the translated sequence.
- Dots (·) are represented by short flashes, and dashes (–) by longer flashes.
- Pauses are used to clearly separate individual characters.
- Ensures accurate timing of flashes and gaps for proper Morse code transmission.

FLASHLIGHT CONTROL MODULE

Flashlight Control with Kotlin and CameraX

- The flashlight functionality in the project is managed using Kotlin's CameraX API, which provides easy access to control the device's flashlight. It uses two key functions:
  - cameraControl?.enableTorch(true) to turn the flashlight on.
  - cameraControl?.enableTorch(false) to turn the flashlight off.

Timing and Signal Regulation

- The module carefully manages the timing of the flashlight's on/off states using delay functions to ensure accurate signaling.
- This precise control over timing guarantees that the recipient can correctly distinguish between dots, dashes, and pauses during the transmission.

FLASHLIGHT CONTROL MODULE

Each signal is timed as follows:

- Dots are flashed for 200ms, while dashes are flashed for 600ms.

- Pauses are added to maintain the structure of Morse code:
  - Inter-character pause (between letters): 1000ms,

- This ensures that the message is transmitted clearly and in a format that the receiving module can decode without errors.

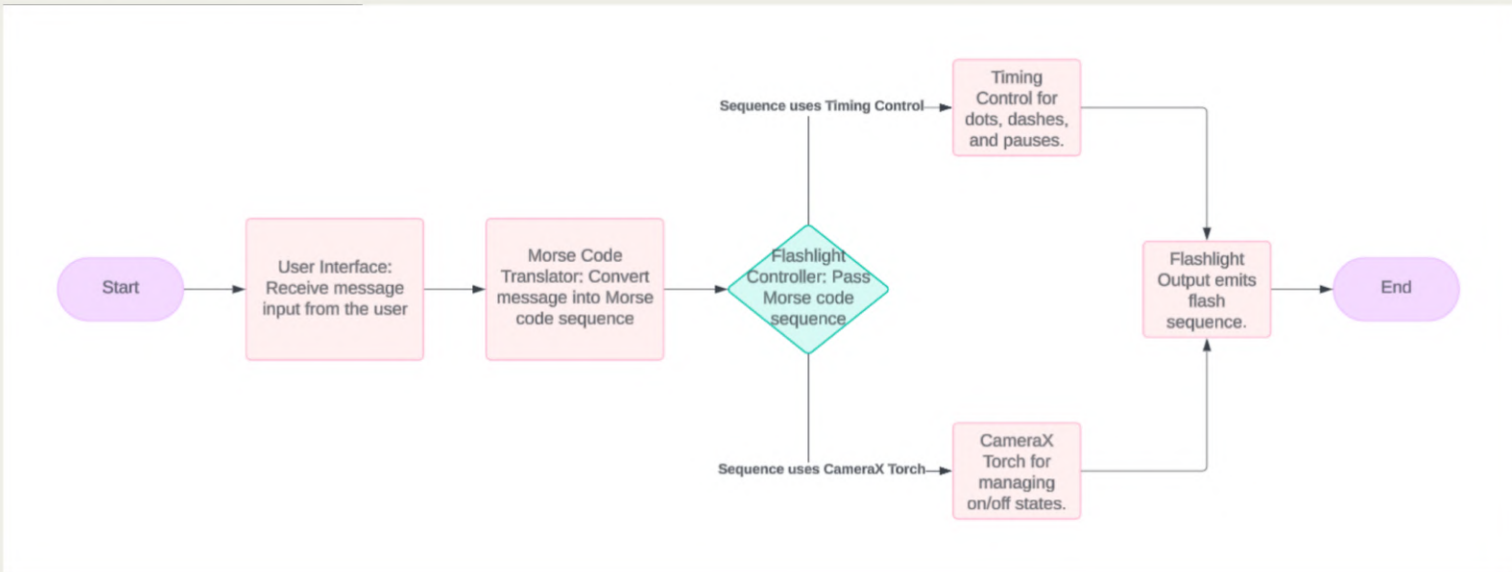
FLASHLIGHT CONTROL MODULE

Module Output

- The Flashlight Signal Transmission Module outputs a sequence of controlled light flashes, which represent the user's message in Morse code format.

- This sequence is passed to the next module — the Receiver Module — which is responsible for detecting and decoding these flashes back into text

FLASHLIGHT CONTROL MODULE



RECEIVER MODULE

1. Video Input Capture

- The module uses cameraX's Image Analysis use case to connect with the camera and capture live video frames continuously, treating each frame as a snapshot in a sequence.
- Frames are captured at a specified rate (e.g., 30 frames per second) for smooth real-time processing.
- The captured frames form a live video stream, which is then fed into the subsequent steps of the module for further analysis, such as detecting changes in light intensity that correspond to Morse code signals.
- By processing each frame individually, the module can detect rapid changes in brightness, allowing it to identify when a flashlight turns on or off.

RECEIVER MODULE

2.Region of Interest (ROI) Detection:

- The module processes each video frame to identify potential flashlight signal areas, starting with grayscale conversion.
  - This step reduces the complexity of the image by focusing on light intensity, ignoring color information
- Next, Gaussian blurring is applied to the grayscale image to smooth the frame and reduce noise, allowing the system to focus on significant light intensity changes.

.

RECEIVER MODULE

3. Blink Pattern Analysis:

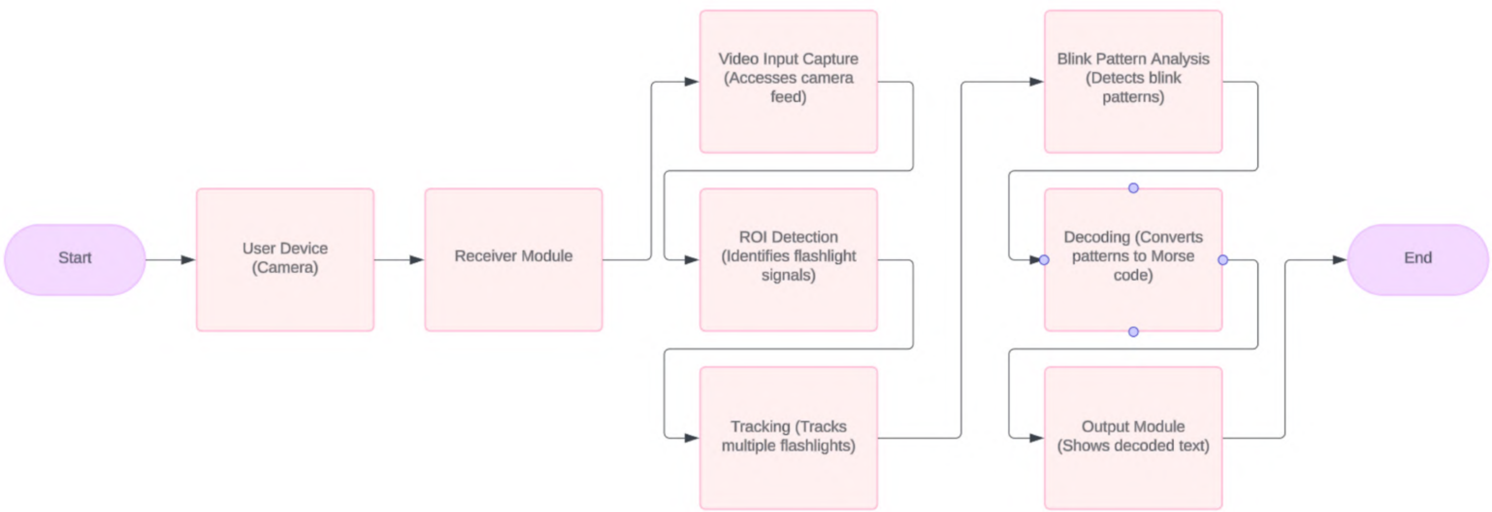
- Analyzes frame sequences to detect brightness changes indicating flashlight on/off states.
- Frame Differencing: Compares consecutive frames to identify significant intensity shifts, signaling flashlight blinks.
- Applies a threshold to ignore minor variations and focus only on actual blink events.
- Duration Measurement: Calculates the time between blink start and end using frame timestamps.
- Blink duration helps classify signals as dots, dashes, or gaps in Morse code.

RECEIVER MODULE

3. Blink Pattern Analysis:

- Short blinks represent dots (·); longer blinks represent dashes (–) in Morse code.
- Blink Classification: Based on duration, blinks are classified as dots or dashes.
- Pause Detection:
  - Inter-character pauses separate letters.
  - Word gap symbol is used to detect word gaps.
- Accurate timing analysis enables reliable conversion of blink patterns into Morse code sequences for decoding.

RECEIVER MODULE



DECODER MODULE

This module focuses on the process of translating the captured Morse code back to text.

Input Handling

- The module accepts a Morse code sequence as input, consisting of dots (.), dashes (-), single spaces between characters, and space symbol("..--") between words

- Example Input: .... . .-.. .-.. --- .-- --- .-. .-.. -..

- This Morse code string is passed into the system for decoding.

DECODER MODULE

Signal Decoding (Dot, Dash, and Space Mapping)

- Dot and Dash Identification: Detects dots (.) and dashes (-) and maps them to letters using a predefined Morse code dictionary.

- Character Separation: Uses a single space to separate Morse symbols into individual characters.

- Word Separation: Interprets the space symbol(..--) as a separator between words in the decoded message.

DECODER MODULE

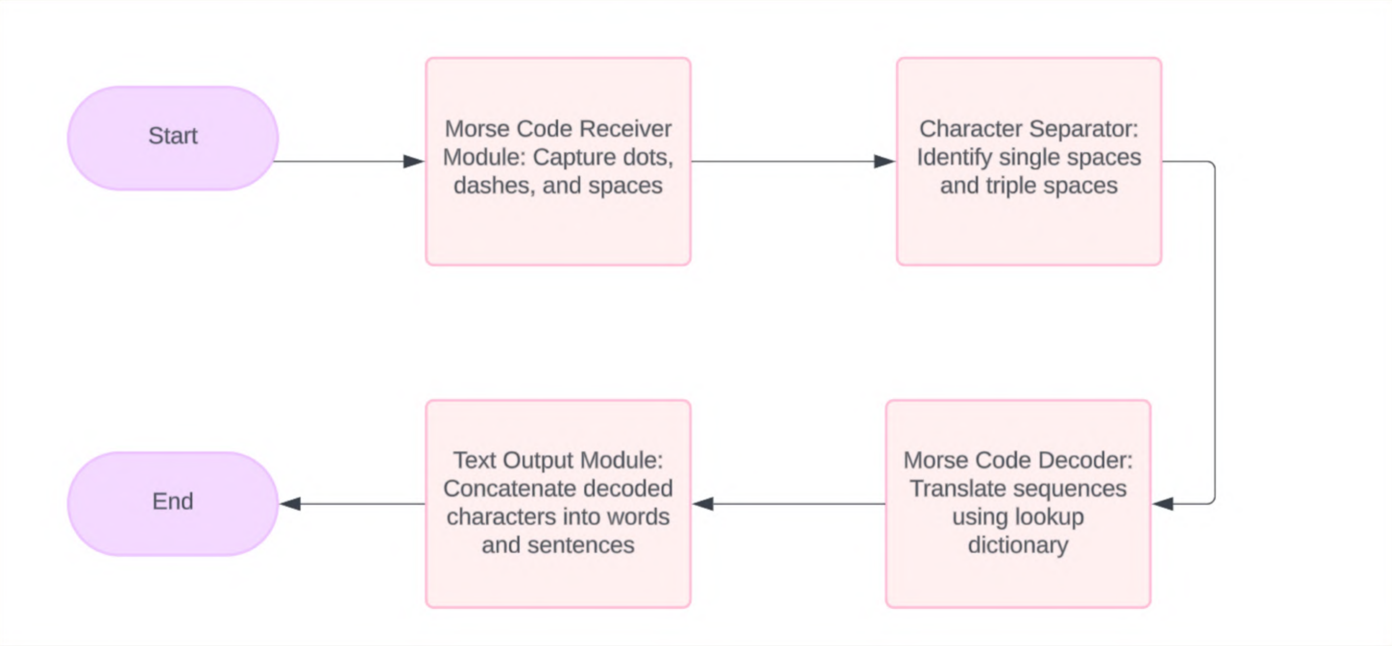
Decoding Process

- After each character is decoded, the module concatenates them into words, with a space between each word.
- Input: Morse code: .... . .-.. .-.. --- ..-- .-- --- .-. .-.. -..
- Output: Text: "HELLO WORLD"

Module Output

- The module outputs a plain text message, fully decoded from the Morse code input, allowing for easy readability.

DECODER MODULE



LEARNING MODULE

- Interactive Quiz – Tests Morse code knowledge with engaging questions

.

- Translation Region – Converts English to Morse and Morse to English in real-time.
- Reference Sheet – Provides a quick lookup table for Morse code mappings.
- Hands-on Learning – Enhances understanding through practical usage and instant feedback.

ASSUMPTIONS

- Users are within line of sight of the intended recipient during emergency situations.
- Traditional communication methods, such as mobile networks, are either unavailable or unreliable.
- Users have access to a smartphone equipped with a flashlight and sufficient battery power.
- Ambient light conditions allow the receiver to detect the flashlight signals effectively.
- The devices used have adequate camera quality to detect variations in light intensity for Morse code.
- The distance between sender and receiver is within a range where the flashlight can be visually distinguished.

WORK BREAKDOWN AND RESPONSIBILITIES

Dhiraj Bobby : UI Development and Application Logic

- Dhiraj will be responsible for designing and implementing the user interface of the mobile application using Kotlin, ensuring it is intuitive and easy to use.
- He will also develop the core application logic that connects the various features, managing data locally to ensure offline functionality.
- He will focus on providing a cohesive and responsive user experience, incorporating features such as the flashlight control and Morse code translation.

WORK BREAKDOWN AND RESPONSIBILITIES

Kalidas Jayakumar : Morse Code Translation and Flashlight Control

- Kalidas will be responsible for developing the text-to-Morse code translation module.
- This will involve creating an algorithm to convert the input English text into a sequence of dots and dashes, adhering to standard Morse code conventions.
- Furthermore, he will implement a control mechanism for the smartphone's flashlight to generate the light patterns corresponding to the Morse code, ensuring precise timing and accurate representation of each symbol.

WORK BREAKDOWN AND RESPONSIBILITIES

John Kurian : Image Processing for Signal Detection

- John will handle the image processing tasks, focusing on detecting and decoding Morse code signals transmitted via the smartphone flashlight.

- This involves developing an algorithm using OpenCV to identify variations in light intensity in the captured video frames and accurately differentiate the flashlight signals from ambient light sources.

WORK BREAKDOWN AND RESPONSIBILITIES

Jerin Vincent : Learning Module and Decoding

- Jerin will be responsible for developing the learning module that helps users to learn morsecode through quizzes.

- He will also handle a part of the decoding phase of the project, making sure that the flashlight decoding using the device's camera remains accurate.

.

HARDWARE & SOFTWARE REQUIREMENTS

MINIMUM HARDWARE REQUIREMENTS:

- Operating System: Android 7.0 (Nougat) or higher
- Memory (RAM): 4 GB
- Internal Storage: 8 GB available
- Processor: Qualcomm Snapdragon 600 series or equivalent
- Display Resolution: 720 x 1280 pixels
- Camera and Flashlight: Required for Morse code detection and transmission

HARDWARE & SOFTWARE REQUIREMENTS

SOFTWARE REQUIREMENTS:

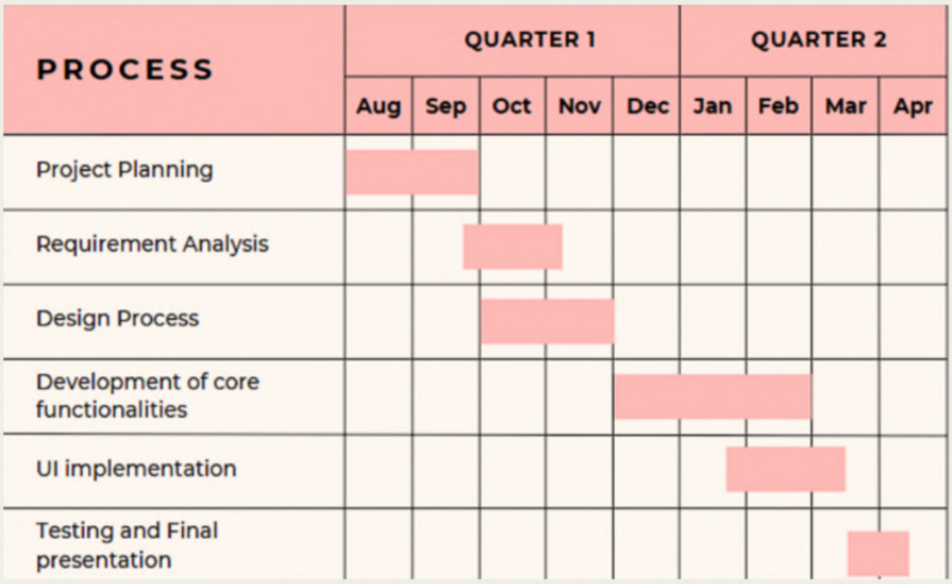
- Development Framework: Kotlin and Android Studio for mobile app development

- Programming Languages: Python for image processing tasks, Java or Kotlin to implement flashlight control functionality.

- Image Processing Library: OpenCV for real-time detection of Morse code signals

- Integrated Development Environment (IDE): Android Studio or Visual Studio Code for application development.

GANTT CHART



The Gantt chart illustrates the timeline and progression of each development phase of the MorseTalk project, ensuring a well-organized approach to achieving project milestones.

BUDGET

- Open-Source Tools:
    - The project uses free open-source software, including OpenCV and Vosk.

- Device Usage:
    - Development and testing are carried out using the team's personal devices, eliminating hardware costs.

- Cost-Effective Approach:
    - The project design focuses on utilizing available resources and expertise, ensuring a minimal budget.

FREE OF COST

RISK AND CHALLENGES

Limited pixel intensity at long distance

- As distance increases, flashlight signals appear dimmer on camera, lowering pixel intensity and affecting accurate Morse code detection. A pixel count method can be an alternative for better signal recognition.

Hardware Limitations and Environmental Interference

- The application's reliance on the device's camera and flashlight for Morse code transmission and detection may be affected by hardware limitations and environmental conditions, such as poor lighting, which could compromise reliability.

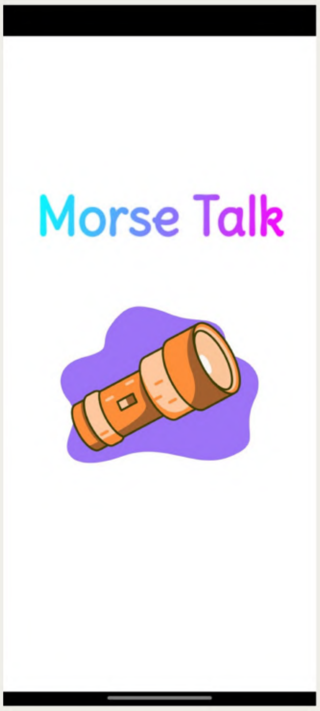
CONCLUSION

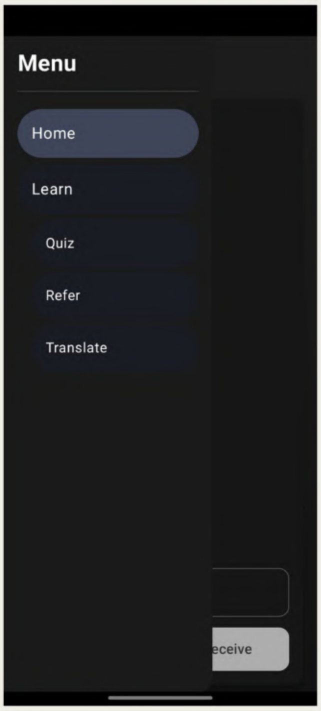
- Reliable in Critical Situations: MorseTalk transforms text into Morse code using the phone’s flashlight, providing a dependable way to communicate when other systems fail.

- Offline Functionality: Works without internet, making it useful in disasters or areas with poor network coverage.

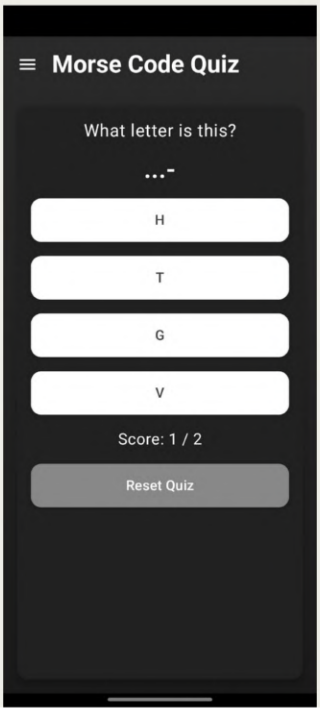
- Visual Communication: Ensures messages are seen via flashlight signals, offering an effective method when traditional communication is impossible.

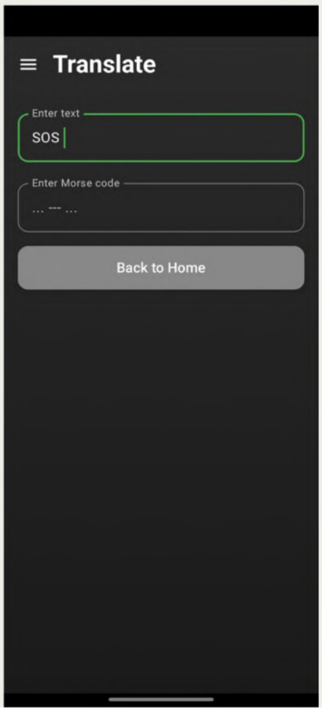
OUTPUT



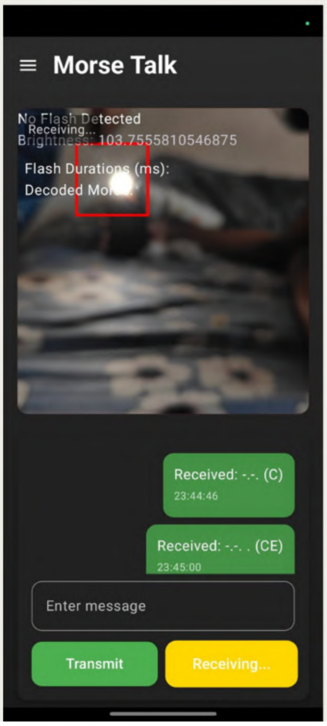


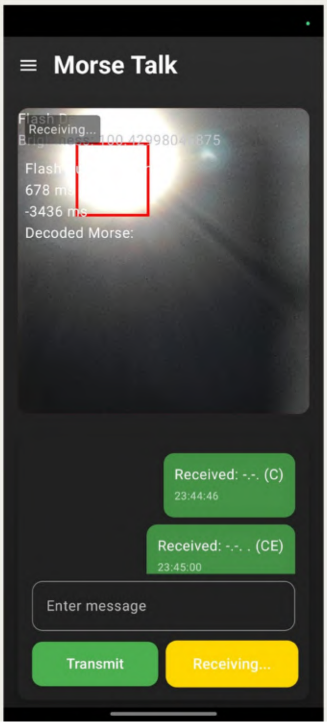
OUTPUT





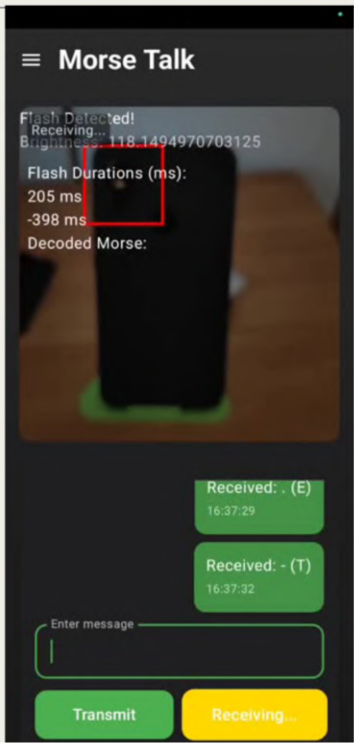
OUTPUT





OUTPUT

OUTPUT



REFERENCES

- Kulkarni, S.J., Ambekar, A.P., Gopale, K.N., Bhandare, P.V. and Chakre, M.R.R., 2021. LiFi Based Visible Light Communication. *International Research Journal of Engineering and Technology.*

- Kumar, K., Srikar, V.S., Swapnika, Y., Sravani, V.S. and Aditya, N., 2020. A novel approach for Morse code detection from eye blinks and decoding using OpenCV. *International Journal for Research in Applied Science & Engineering Technology (IJRASET), 8.*

- Deepthi, R.S. and Sankaraiah, S., 2011, September. Implementation of mobile platform using Qt and OpenCV for image processing applications. *In 2011 IEEE Conference on Open Systems (pp. 284-289). IEEE.*

- P. Nalajala, B. Godavarth, M. L. Raviteja and D. Simhadri, "Morse code generator using Microcontroller with alphanumeric keypad," *2016 International Conference on Electrical, Electronics, and Optimization Techniques.*

- R. Li, M. Nguyen and W. Q. Yan, "Morse Codes Enter Using Finger Gesture Recognition," *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*

Thank you!

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Appendix B

Vision: To become a Centre of Excellence in Computer Science & Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Mission: To inspire and nurture students, with up-to-date knowledge in Computer Science & Engineering, Ethics, Team Spirit, Leadership Abilities, Innovation and Creativity to come out with solutions meeting the societal needs.

Program Outcomes:

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes:

PSO1: Computer Science Specific Skills: The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills: The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills: The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

CO1: Model and solve real world problems by applying knowledge across domains.

CO2: Develop products, processes, or technologies for sustainable and socially relevant applications.

CO3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks.

CO4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms.

CO5: Identify technology/research gaps and propose innovative/creative solutions.

CO6: Organize and communicate technical and scientific findings effectively in written and oral forms.

Appendix C: CO-PO-PSO Mapping

Appendix C

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO 2	2	2	2		1	3	3	1	1		1	1		2	
CO 3									3	2	2	1			3
CO 4					2			3	2	2	3	2			3
CO 5	2	3	3	1	2							1	3		
CO 6					2			2	2	3	1	1			3

3/2/1: high/medium/low