Indian Institute of Technology Gandhinagar

CS 435: Human - Computer Interaction

Project Report

Morse-Code–Based Conversational Keyboard

for Visually Impaired Users

Submitted by:

Rutuja Swami

Sneha Gautam

Suryanivasini

Vedanshi Rayani

Abstract

Text entry on touchscreens is challenging for visually impaired users because traditional keyboards require precise spatial navigation. Alternatives like voice and gesture input are unreliable in noisy conditions or difficult to master. This project introduces a Morse-code–based conversational keyboard for Android that converts dot and dash inputs into text, supported by audio and vibration for non-visual accuracy. The report presents the system architecture, conceptual design, prototype features, user evaluation, ethical considerations, and the rationale for choosing Morse input. The final prototype shows high usability, strong learnability, and clear potential to improve accessible communication.

---

1. Introduction

Touchscreen keyboards present major challenges for visually impaired users due to the absence of tactile boundaries and the need for continuous visual guidance. Even with assistive tools such as TalkBack or magnification, typing accuracy and speed remain limited, and spatial layouts like QWERTY require precise finger placement that is difficult to maintain without sight.

This project aims to remove the need for spatial accuracy by replacing location-based input with a rhythm-based interaction model. The proposed Morse-code keyboard enables text entry through combinations of short and long taps, with a space mechanism for completing words. This approach reduces both cognitive and physical effort.

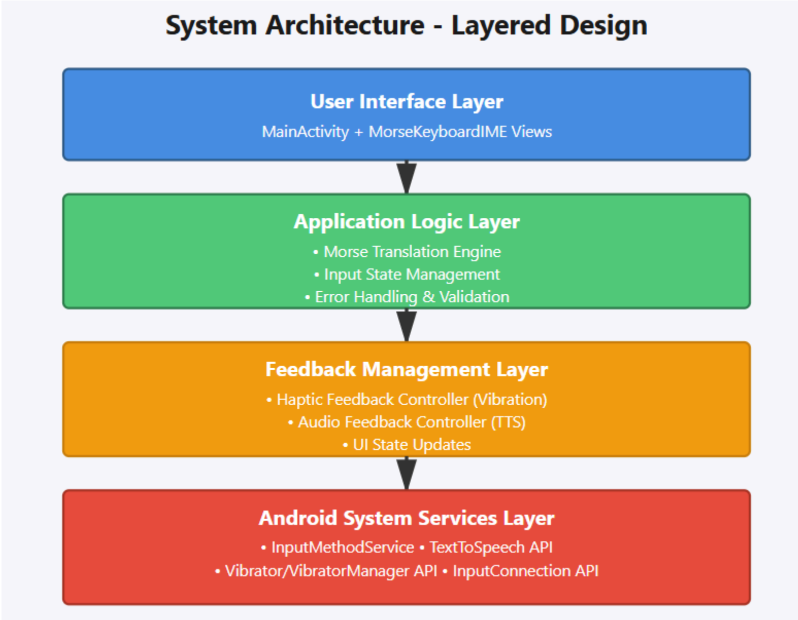
The prototype was chosen from two initial concepts—a Morse-code calculator and a Morse-code keyboard—with the keyboard selected for its broader practical relevance. The final design prioritizes accessibility, ease of learning, and dependable non-visual feedback, making it suitable for everyday smartphone use.

---

2. System Design and Architecture

2.1 Overview

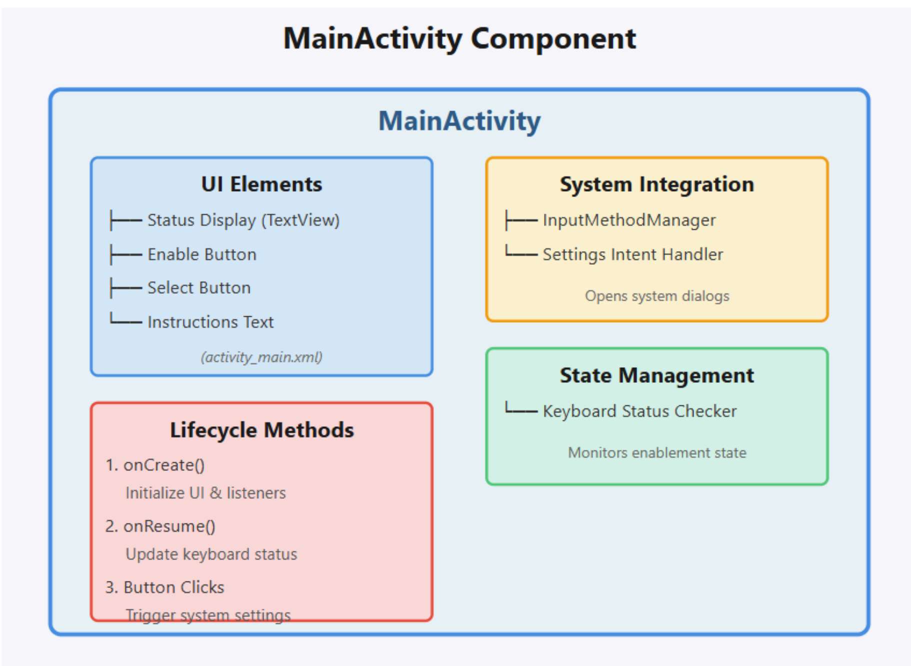
The keyboard is implemented as an Android Input Method Editor (IME), allowing it to function anywhere text input is required. This system-wide integration enables seamless use across messaging, browsing, search, and note-taking applications.



2.2 Component Architecture

A. Setup Component (MainActivity)

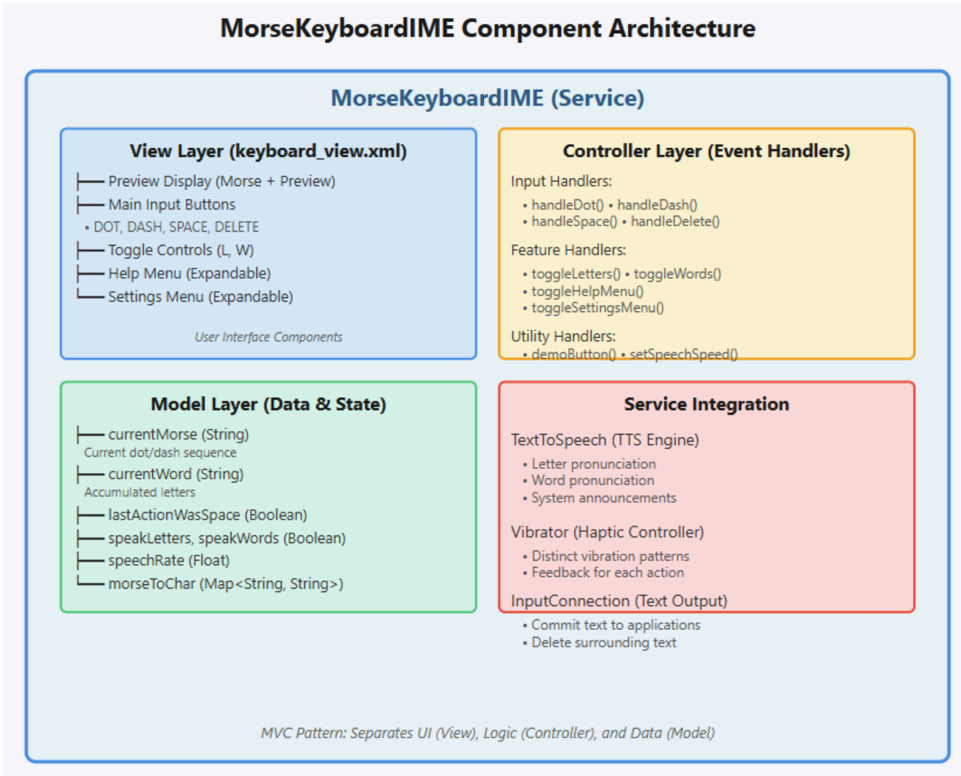
Purpose: Provides user onboarding and keyboard configuration interface



B. Keyboard Service Component (MorseKeyboardIME)

Purpose: Core IME service providing Morse code input functionality

Architecture Pattern: Service-based with MVC-like structure

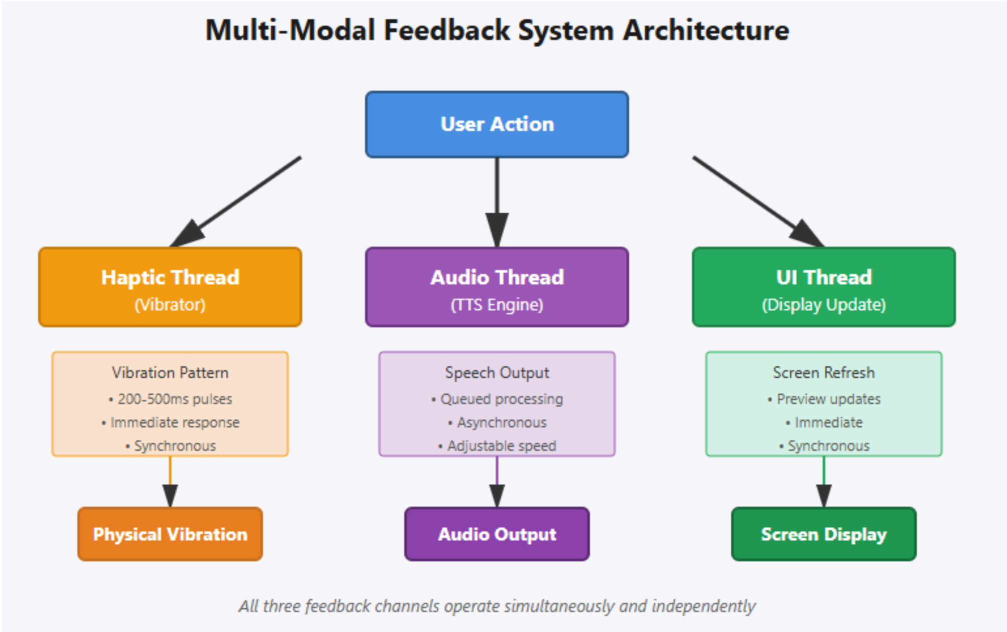


2.3. State Management Architecture

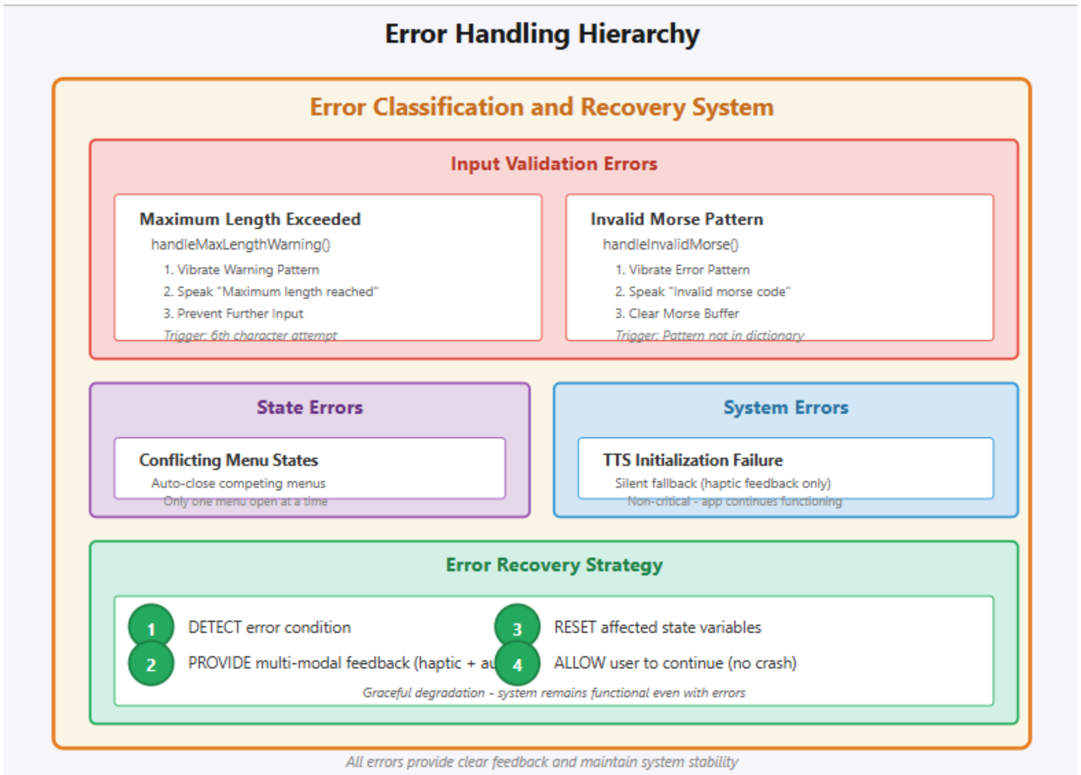
State Variables:

Variable	Type	Purpose	Scope
`currentMorse`	String	Stores current dot/dash sequence	Session
`currentWord`	String	Accumulates letters for word pronunciation	Session
`lastActionWasSpace`	Boolean	Tracks consecutive space presses	Session
`speakLetters`	Boolean	Controls letter pronunciation	Persistent
`speakWords`	Boolean	Controls word pronunciation	Persistent
`isHelpExpanded`	Boolean	Help menu visibility state	Session
`isSettingsExpanded`	Boolean	Settings menu visibility state	Session
`speechRate`	Float	TTS speed multiplier	Persistent

2.4. Feedback System Architecture



2.5. Error Handling Architecture



This comprehensive System Design and Architecture documentation demonstrates a well-structured, maintainable, and accessible keyboard application designed specifically for visually impaired users, with clear separation of concerns, robust error handling, and efficient use of Android system APIs.

3. Compromises in Prototyping

Our high-fidelity prototype adopts a vertical compromise approach, focusing on perfecting the core Morse-code typing interaction rather than developing a full messaging application. The prototype delivers a highly functional keypad that enables text input in any Android app through dot–dash sequences. It is designed for visually impaired users, incorporating auditory feedback, vibration confirmation, a help guide, and customizable input settings to support accurate and confident typing.

To achieve this depth, the system's overall breadth was intentionally limited. Features such as a complete messaging interface, contact management, and full app navigation were simplified or omitted, allowing resources to be concentrated on creating a realistic, responsive, and accessible typing experience. Consequently, the prototype offers strong functional depth in its primary task while intentionally sacrificing the broader feature set characteristic of a full application.

---

4. Conceptual Model

The conceptual model of the Morse-code–based keypad specifies the user actions and concepts required for interaction. Its goal is to enable visually impaired users to type on any Android device using dot, dash, space, and delete inputs, each supported by auditory and vibrational feedback. Characters are formed through sequential Morse inputs, automatically decoded, and entered into any text field on the device.

Development of the model began with an examination of the problem space, focusing on issues such as limited tactile guidance, accidental presses, and slow text entry on touchscreens. Through brainstorming, observation, and empathy-building activities, we identified strategies to simplify input and support inclusive interaction. The model emphasizes minimal visual dependency, consistent feedback, and a low learning curve.

Alternative concepts—including gesture-based typing and voice input—were evaluated, but Morse code was chosen for its simplicity, predictability, and suitability for non-visual use. Scenarios and early prototypes informed the interaction flow and refined user behavior patterns.

4.1 Interface Metaphor

The interface metaphor is based on a physical telegraph key, where dots and dashes form letters. This links familiar historical interactions with modern touch input, enabling users to type through rhythmic taps rather than locating keys.

Metaphor Evaluation

Structure: The telegraph metaphor provides a clear sequence-based system where short and long taps generate letters.

Relevance to the problem: It supports non-visual typing without reliance on visual scanning, aligning with the needs of visually impaired users.

Ease of representation: Large touch buttons for dot, dash, space, and delete effectively mirror the telegraph workflow.

Audience understanding: A help guide explains key functions, while auditory and vibration cues make the metaphor intuitive.

Extensibility: The metaphor can expand to include shortcuts, predictive text, and customizable Morse sets.

4.2 Design

The design emphasizes simplicity, accessibility, and low cognitive load. Large, high-contrast buttons for dot, dash, space, and delete support accurate tapping. Each input provides vibration and spoken feedback to confirm entries, enabling confident non-visual typing. A help guide teaches Morse sequences and offers practice. Customization settings allow users to adjust audio speed and feedback type.

The prototype functions system-wide on Android, allowing the keypad to be used in any application—messages, search bars, notes, and more—ensuring consistency and removing the need to switch interfaces.

4.3 Design Space

The design space covers multiple non-visual text-entry approaches, including:

- Voice typing (fast but unreliable in noisy or private contexts)
- Braille input (efficient but requires extensive training)
- Gesture-based typing (hard to memorize, error-prone)
- Prediction-based minimal keypads (less accurate for non-visual use)

Our solution occupies the design-space point of a Morse-code keypad with tactile and audio feedback, where:

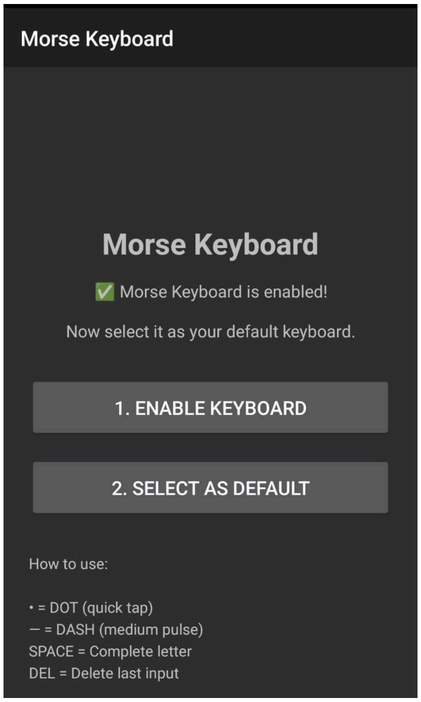
- Input is minimal (dot and dash)
- Learning is simpler than Braille
- Feedback is multimodal
- Visual dependency is removed
- Use is possible in silent, noisy, or dark environments

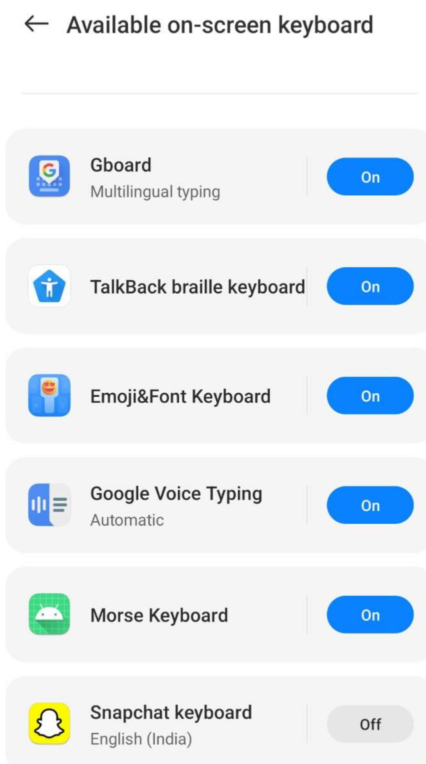
By narrowing the space to rhythmic, sequential input, the design achieves a balance of learnability, accessibility, and broad usability.

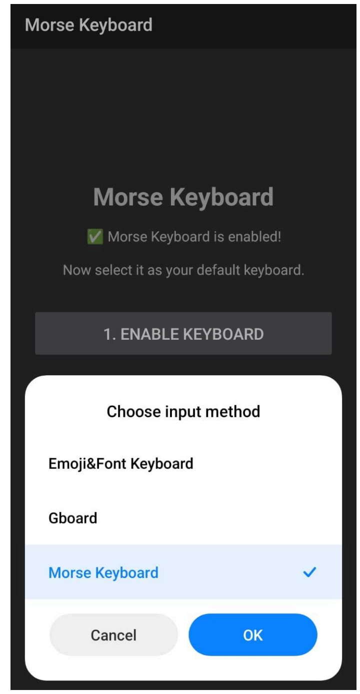
5. Prototype Description and Functionalities

5.1 Installation and Enabling keyboard

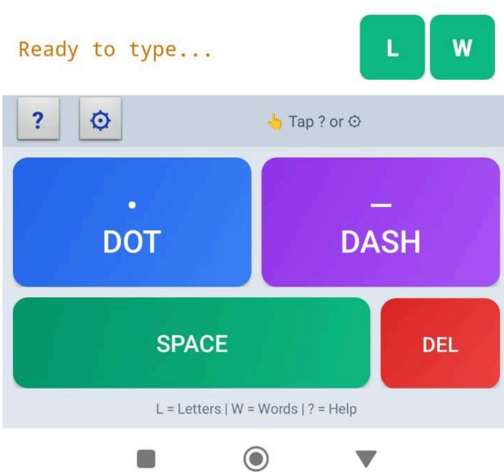
Step 1 | Step 2 | Step 3







5.2 Main Keyboard Screen

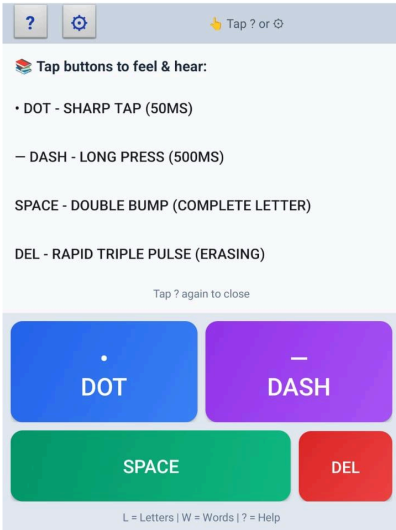


The keyboard consists of four large buttons: dot, dash, space, and delete. Dot and dash occupy the central region of the screen to allow comfortable tapping. Space and delete are positioned below them. The top corners contain icons for the help guide and settings.

Each action delivers consistent feedback:

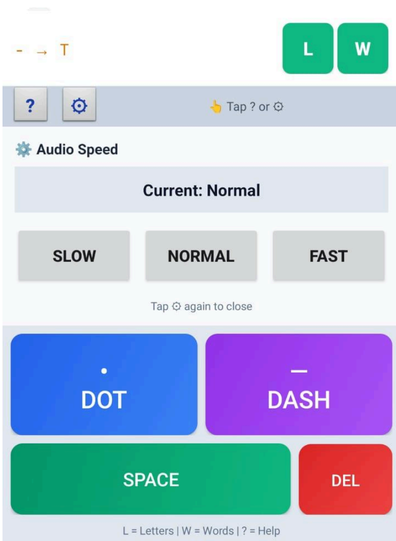
- A dot tap triggers a short vibration.
- A dash tap triggers a longer vibration.
- Converting a sequence with space triggers spoken output (depending on preferences).
- Errors generate a spoken message and a distinct vibration pattern.

5.2 Help Screen



The help screen is designed as an interactive learning tool. Users can tap on entries corresponding to dot, dash, space, and delete to learn their vibration patterns. This hands-on learning style supports users who are unfamiliar with Morse code or have not used rhythmic input systems before.

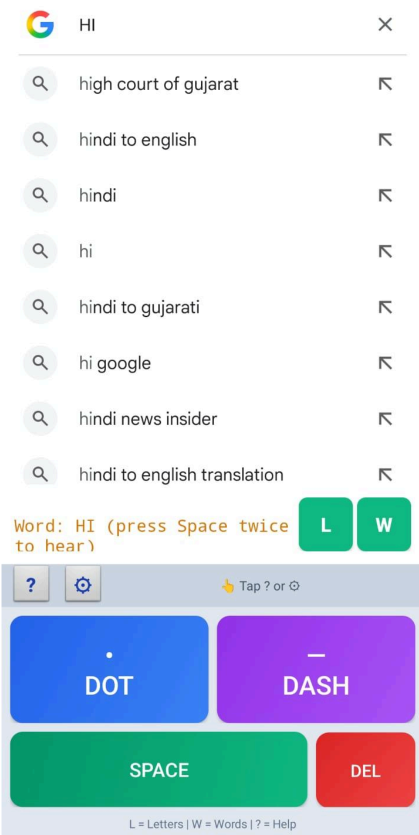
5.3 Settings Screen

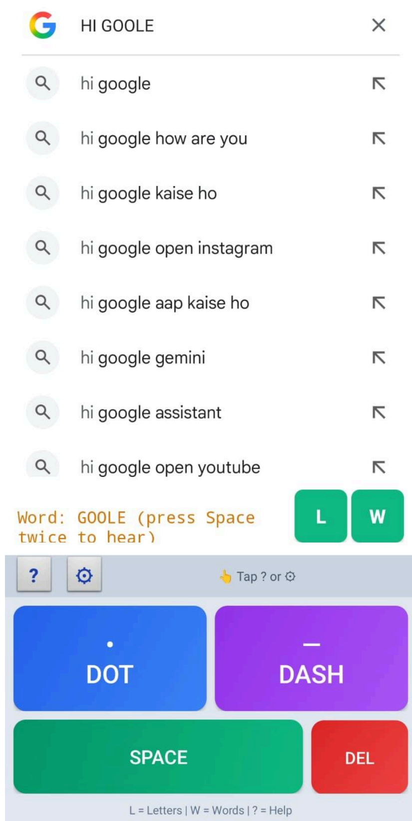


The settings menu allows users to choose how much audio feedback they want. Beginners often select the “letters only” or “both letters and words” modes, while advanced users prefer hearing only completed words. Speech rate control ensures that output remains tuned to the user’s comfort level.

5.4 Interaction Examples

- To type the letter "H", the user taps dot, then dash, then space.
- To type the word “GOOGLE,” individual letters are typed with space after each one. A second press of space speaks and inserts the whole word.
- Errors such as invalid Morse sequences are automatically detected and communicated clearly.
- Delete removes one dot or dash at a time, enabling incremental correction.





---

6. Interaction Behavior

The behavior of the system is governed by consistency and clarity. Users receive immediate feedback for every action, which reduces uncertainty and promotes confidence.

6.1 Typing Flow

Typing follows a structured rhythm:

1. Input dots and dashes.
2. Press space to attempt letter conversion.
3. Continue typing or press space again to complete the word.

This predictable rhythm makes the system easy to learn and fast to use.

6.2 Error Handling

The system distinguishes between several types of errors. Each error results in a spoken message

Wrong pattern - “Invalid Morse Code”
Buffer Overflow - “Maximum Length Reached”

and vibration (dual pulse) so the user immediately understands what has gone wrong and can correct it without confusion.

6.3 Deletion Behavior

By deleting only the most recent symbol, the system prevents the frustration of losing entire letters due to a minor mistake. This fine-grained correction mechanism proved extremely useful in user testing.

---

7. User Evaluation

Six participants took part in usability testing. Two were visually impaired, and the remaining four were sighted users who were blindfolded for the session. Tasks included learning vibration patterns, typing individual letters, typing full words, correcting errors, and adjusting settings.

- Accuracy for letter recognition increased rapidly with practice.
- Word-level accuracy reached promising levels within a short session.
- Error rates decreased as users internalized the rhythm of the input method.
- Satisfaction ratings were consistently high, indicating strong usability and perceived usefulness.

	Participant	TaskTime	Errors	Satisfaction
0	P1	42	4	7
1	P2	39	3	8
2	P3	46	5	7
3	P4	44	4	8
4	P5	55	2	9
5	P6	58	3	9

A detailed analysis of the user evaluation study is provided in the ‘Evaluation Data’ file attached.

---

8. Ethics and Risk Assessment

8.1 Ethical Considerations

Keyboards handle sensitive data, so the prototype ensures complete privacy by avoiding any form of data storage or transmission. All input remains strictly on the user’s device.

8.2 Potential Risks and Mitigations

- **Audio privacy risks:** These are addressed by allowing a user-choice to silent word or letter feedback or both.
- **Audio feedback customization:** Users can modify audio speed to suit their comfort levels.

- **Learning curve:** The help screen and vibration demonstrations guide new users effectively.

The system therefore maintains a low ethical and functional risk profile.

---

9. Conclusion

The Morse-code–based keyboard provides a powerful alternative to traditional touchscreen typing for visually impaired users. By reducing input to rhythmic sequences supported by multimodal feedback, the system overcomes many of the challenges associated with spatial layouts. User evaluations confirm that the method is learnable, accessible, and effective for non-visual environments.

Overall, the project showcases how rhythm-based interaction models can transform accessibility and provide visually impaired users with more reliable and empowering communication tools.