HearWay

UID project 2025

Pop Alexandra
Moise Mihai
Radi Csaba

1. Project Idea and Need

HearWay is a smart mobility solution designed to help both deaf and blind users navigate urban environments safely and independently.

The system detects environmental sounds such as sirens, horns, alarms, and voices and converts them into visual and vibration-based alerts.

For deaf users, sounds are displayed on the screen as color-coded visual indicators (a red arrow showing direction), while for blind users, the bracelet provides directional vibration patterns and voice guidance that describes what type of sound is detected and where it comes from.

2. Description of the Application Domain

The project fits into the domain of inclusive urban mobility and assistive technology. It addresses the everyday challenges faced by people with hearing or visual disabilities when moving through noisy and unpredictable city environments.

HearWay enhances awareness and independence by translating sounds into multisensory feedback through vibration, voice, and visuals.

3. User Identification and Description

3.1. Main Users

Main users include individuals with hearing or visual impairments. Secondary users include family members, caretakers, or volunteers who assist them.

3.2. User Knowledge Levels

Users are familiar with smartphones or wearable devices at a basic to medium level. The system is designed to require minimal technical knowledge and support intuitive interaction.

3.3. Special User Requirements

Visually Impaired Users

Require strong haptic (vibration-based) and audio feedback rather than visuals.

Need clear, short voice guidance describing both the **type** and **direction** of detected sounds.

Interface must be compatible with screen readers and accessible gestures.

The bracelet must deliver distinct vibration patterns for different sounds (e.g., short pulses for pedestrian signals, continuous for sirens).

Hearing-Impaired Users

Require **visual alerts** and optional vibration cues instead of sound.

Color coding and arrow indicators must clearly show direction and urgency.

High-contrast mode and customizable brightness are essential for outdoor visibility.

Notifications should avoid flashing patterns that could cause discomfort or confusion.

4. Main Challenges

People with hearing and visual disabilities face serious safety risks while navigating modern urban environments due to their limited access to sound or visual cues.

Current mobility applications focus mainly on sighted and hearing users and are not designed to interpret real-time environmental sounds or to deliver adaptive, multimodal feedback.

The main challenges identified in this domain are:

- **Sound awareness gap:** Deaf users cannot perceive danger signals such as car horns, alarms, or ambulance sirens.
- **Visual awareness gap:** Blind users cannot see approaching vehicles or visual traffic signals and rely entirely on sound.
- **Lack of inclusive solutions:** Most existing navigation apps provide either visual or auditory feedback, not both in a synchronized way.
- **Environmental complexity:** Cities are unpredictable—traffic, noise pollution, and crowding can overwhelm or mislead users relying on a single sense.
- **Accessibility inconsistency:** Transport infrastructure and digital tools rarely follow universal accessibility standards.

Therefore, the challenge is to create a **smart, inclusive, and context-adaptive system** that converts real-world sounds into **visual, haptic, or vocal cues**, helping both deaf and blind people move independently, safely, and confidently through the city.

5. Context of Use

HearWay is designed for **dynamic, unpredictable, and often noisy urban environments** where users with hearing or visual disabilities need constant awareness of what happens around them.
The system operates through a mobile app and a smart bracelet that work together to provide **real-time sensory feedback** — visual, vocal, or vibration-based — depending on the user's disability type.

The most relevant contexts of use include:

- **Highly noisy environments:**
  Streets filled with cars, trams, or construction noises where deaf users cannot rely on sound, and blind users may have difficulty distinguishing specific noises.
  HearWay filters and interprets important sounds like sirens or horns and provides immediate alerts.
- **Time-sensitive situations:**
  Scenarios where quick reaction is essential, such as crossing streets, avoiding moving vehicles, or responding to emergency signals.
  The bracelet delivers instant vibration patterns, and the app highlights or announces the direction of the detected sound.
- **Situations with frequent interruptions:**
  Users may be distracted while walking, talking, or using public transport. HearWay runs continuously in the background, ensuring that alerts are not missed even when the phone screen is off.
- **Indoor and outdoor mobility:**
  The system works both in open areas (streets, crosswalks, parks) and in enclosed spaces (shopping centers, stations, universities), adapting the feedback intensity to ambient noise levels.
- **Accessibility conditions:**
  The app includes day/night visual modes, supports screen readers, and connects to assistive devices such as smartwatches or hearing aids, ensuring usability in varied lighting or weather conditions.

6. Current Applications Used by the Target Users

People with hearing or visual disabilities currently rely on a mix of accessibility and navigation tools, but none of them fully cover real-time sound detection combined with adaptive mobility support.

For hearing-impaired users:

- **SoundWatch (Google Research):** Detects common sounds like alarms, sirens, or dog barks and sends vibration notifications through a smartwatch.
  Advantage: Quick sound recognition.
  Limitation: No indication of direction or sound distance; cannot be integrated with navigation.
- **Ava:** Converts speech into live captions for conversations.
  Advantage: Improves communication in daily interactions.
  Limitation: Focused only on speech, not environmental awareness.

For visually-impaired users:

- **Be My Eyes:** Connects blind users with volunteers through a live video call for guidance.
  Advantage: Real human support and context understanding.
  Limitation: Requires an active helper; lacks autonomy.
- **Google Maps / Moovit:** Provide GPS-based navigation and route planning.
  Advantage: Accurate location tracking and public transport integration.
  Limitation: No accessibility adaptation (no sound interpretation or haptic guidance).
- **Seeing AI (Microsoft):** Uses AI to describe surroundings and read text aloud.
  Advantage: Good for object and text recognition.
  Limitation: Does not handle environmental sound awareness or direction-based alerts.

7. Coordination with Other Tools and Design Considerations

HearWay is designed to **cooperate seamlessly with existing mobility, accessibility, and smart-city tools** to provide users with a complete experience, rather than replacing what already works well.

Main integrations:

· **Navigation platforms** such as *Google Maps* or *Moovit* – HearWay uses their routing data and adds accessibility layers like sound detection and real-time hazard alerts.

· **Smart devices and wearables** – The bracelet communicates via Bluetooth with the mobile app to deliver synchronized vibration patterns. HearWay can also connect with smartwatches, AR glasses, or hearing aids for extended accessibility.

· **Public transport and city infrastructure APIs** – The system can receive information about traffic signals, emergency alerts, or public-transport schedules to anticipate potential risks or route changes.

· **Assistive technologies** – Compatibility with screen readers, voice assistants (e.g., Google Assistant), and accessibility shortcuts ensures inclusive interaction for all user categories.

Design considerations that enable this cooperation:

· **Unified design language:** same color codes, icons, and feedback patterns across all connected devices.

· **Modular architecture:** allows easy addition of new integrations without redesigning the core app.

· **Low-latency communication:** Bluetooth Low Energy and cloud synchronization for near-instant vibration and visual feedback.

· **Privacy and safety:** user data (e.g., location and sound recordings) are processed locally whenever possible, ensuring confidentiality.

· **Cross-platform accessibility:** interface layouts automatically adapt to mobile, tablet, or wearable formats while maintaining identical interaction flow.

By combining these integrations and design rules, HearWay can **blend naturally with the user's existing ecosystem** of mobility and accessibility tools, creating a connected and consistent experience that supports independent movement in the city.

8. Task Analysis

8.1. Travel Planning and Navigation Tasks

Task 1. Plan a Door-to-Door Commute

User	Deaf or blind person commuting to work or university.
Purpose	To plan an efficient, safe trip combining walking, bus, and metro routes.
Starting point	Landing page after authentication.
What they do	The user enters origin and destination, and the app generates an accessible multimodal route.
Context	Morning commute, often in noisy, crowded areas where audio announcements are hard to follow.
Steps	Enter locations → choose preferred transport → receive visual (for Deaf) or voice/vibration (for Blind) directions.

Phase 5: User Scenarios

Scenario A: Visual Planning (Success Case - Deaf User)Context: Ana is getting ready for work in the morning. She prefers checking the route visually to avoid noisy construction areas.

Start: On the Landing Page, Ana taps the large "Where to?" input field.

Action: She types "Office" and selects the saved location. The app instantly displays 3 route cards.

Decision: She notices "Route 2" has a "High Noise" warning icon, so she taps "Route 1" (Bus 24), which is marked as "Quiet".

System Reaction: The screen transitions to the map view with a high-contrast blue path.

Feedback: A message "Route Started" appears at the bottom, and she proceeds to the bus stop.

Scenario B: Audio/Haptic Planning (Alternative Success Case - Blind User)Context: Mircea is leaving the university and walking with a cane, so he cannot type easily.

Start: Mircea double-taps the screen to activate the microphone.

Action: He says, "Take me home."

System Reaction: The app's TTS (Text-to-Speech) responds: "Found two routes. Route 1 is 15 minutes by Metro. Route 2 is 25 minutes by Bus. Swipe right to listen to details."

Decision: Mircea swipes right, then double-taps to select Route 1.

Feedback: The bracelet gives a distinct "Success" vibration pattern (long buzz), confirming that turn-by-turn navigation has started.

Important components of the interface

- **Origin/Destination Input Fields:** Large, high-contrast text boxes with voice-to-text (dictation) support and full screen-reader compatibility.
- **Accessibility Mode Selector:** Clear toggle buttons (e.g., "Deaf/Visual Mode," "Blind/Audio Mode") that filter route results based on accessibility needs.
- **Route Options Cards:** A clear, scrollable list of 2-3 route options. Each card visually displays icons (bus, walk, metro), estimated time, and accessibility warnings (e.g., "Loud construction zone reported").
- **Interactive Map Preview:** A high-contrast map showing the route, which is hidden by default for screen-reader users but expandable for visual users.
- **"Start Trip" Button:** A large, clearly labeled confirmation button.

Data Types Used

- **`RouteRequest`:** Object containing origin (coordinates), destination (coordinates), and `UserProfile` settings.
- **`UserProfile`:** Stores user's preferences (e.g., `isBlind: true`, `isDeaf: false`, `preferQuietRoutes: true`).
- **`RouteObject`:** A list of route options, each containing steps, transport types (e.g., `BUS`, `WALK`), duration, and accessibility flags.
- **`GeoJSON`:** Standard format for displaying the route path on the map.

User interactions

Deaf User:

1. User taps the "Destination" field and types their address.

2. The app displays 3 "Route Options Cards" with clear icons and text (e.g., "Route 1: 20 min - Bus 24").
3. User taps "Route 1." The screen transitions to a high-contrast map showing the full path.
4. User taps the "Start Trip" button, which gives a short vibration confirmation.

Blind User:

1. User double-taps the screen to activate voice input: "Plan a route to the University."
2. App TTS (Text-to-Speech) responds: "Found 2 routes. Route 1 takes 20 minutes by Bus 24. Swipe right to hear Route 2."
3. User swipes right. App TTS: "Route 2 takes 30 minutes with one transfer."
4. User double-taps to select Route 1. App TTS: "Route 1 selected. Navigation starting." The bracelet gives a short haptic buzz.

Alignment to best practices

- **WCAG 2.2 Compliance:** All components use high-contrast text (Perceivable) and are fully operable via keyboard/screen reader (Operable).
- **Cross-Modal Redundancy:** Information is provided in at least two ways: visual (map, icons) and non-visual (TTS, haptics).
- **Reduced Cognitive Load:** The app presents a limited number of route choices (max 3) to prevent overwhelm.

Innovation

- **"Safe-Sound" Routing:** In addition to the *fastest* route, the app suggests an alternative "Quiet Route." This route uses real-time noise-level data to avoid loud construction or heavy traffic, which is ideal for blind users who rely on environmental sounds for orientation.

Phase 6:

Question	Answer & Motivation
1. Will the user be trying to produce the effect?	**YES.** The user has a specific goal (reaching a destination) and actively seeks input fields to enter the address.
2. Will the user see the correct control?	**YES.** The "Where to?" input field and "Start Trip" buttons are large and high-contras. For blind users, these elements are the first focused items for the screen reader.
3. Will the user see that the control produces the desired effect?	**YES.** The "Start Trip" button is explicitly labeled. The Route Cards clearly display transport icons (Bus/Metro) and time, matching the user's mental model of planning a trip.
4. Is there another control that the user might select instead of the correct one?	**POSSIBLE.** The "Accessibility Mode Selector" (Deaf/Blind toggle) might be clicked by mistake if placed too close to the route list. **Improvement:** Add more padding between settings and the route list.
5. Will the user understand the feedback to proceed correctly?	**YES.** The transition to the map screen is immediate. For blind users, the bracelet gives a confirmation haptic buzz, providing non-visual certainty that navigation has started.

Task 2. Receive Alerts for Route Changes

User	Traveler mid-commute using public transport.
Purpose	Stay informed about delays, detours, or cancellations without relying on sound.
Starting point	Active trip screen or background mode.
What they do	The user gets a vibration or on-screen alert when a change occurs.
Context	Inside a bus or metro where announcements are hard to hear or see.
Steps	App monitors route → detects update → sends visual icon (⚠) or vibration pattern → user adjusts journey.

Phase 5: User Scenarios

Scenario A: Visual Alert Handling (Success Case - Deaf User) Context: Ana is on the bus, watching a video on her phone, when a detour occurs.

Trigger: The bus deviates from the route due to an accident.

System Reaction: A red, animated banner drops down from the top of the screen: "UPDATE: Detour ahead. +5 min delay." The phone vibrates shortly to grab attention.

Action: Ana taps the banner to see the new path on the map.

Feedback: The map updates the blue line to show the new route, and the banner turns green: "Route Updated." Ana relaxes, knowing she is still on track.

Scenario B: Haptic Acknowledgement (Alternative Success Case - Blind User) Context: Mircea has his phone in his pocket while standing on a crowded bus.

Trigger: The bus route changes unexpectedly.

System Reaction: Mircea's bracelet emits the specific "Alert" vibration pattern (short-short-short).

Action: Instead of taking out his phone in the crowd, Mircea taps his bracelet once (the "Acknowledge" gesture).

System Reaction: The system interprets this as "I heard you." The TTS whispers in his earpiece: "Bus is detouring. New arrival time: 9:15 AM."

Feedback: Mircea confirms he understands and doesn't need to perform any other action.

Important components of the interface

- **High-Priority Modal Alert:** A full-screen overlay that blocks the app, used for critical changes (e.g., "Detour: Bus 24 is cancelled").
- **Non-Intrusive Notification Banner:** A small banner at the top of the navigation screen for minor updates (e.g., "Bus 24 delayed 5 minutes").
- **Haptic Alert System (Bracelet):** Uses distinct vibration patterns to signify different alert types (e.g., *short-short-short* for a delay, *long-buzz* for a cancellation).
- **Text-to-Speech (TTS) Announcer:** The voice guidance system, which will interrupt navigation to announce the alert.

Data Types Used

- **`AlertObject`:** Contains alert `type` (e.g., `DELAY`, `DETOUR`, `EMERGENCY`), `message` (text for display/speech), and optional `newRouteObject`.
- **`RouteStatus`:** A live object that monitors the user's GPS against the planned route and public transport APIs.
- **`NotificationQueue`:** Manages the order and priority of alerts to avoid overwhelming the user.

User interactions

Deaf User:

1. User is on the bus, looking at the app's map.
2. A red "Notification Banner" animates at the top: "UPDATE: Bus 24 is delayed 10 minutes."
3. The bracelet simultaneously gives a *short-short-short* vibration pattern.

4. User taps the banner to dismiss it.

Blind User:

1. User has the phone locked in their pocket; navigation is running in the background.
2. The bracelet gives a *long-buzz* vibration pattern (critical alert).
3. The phone's TTS automatically announces: "Route Alert: Bus 24 is on a detour and will skip your stop. A new route is available. Double-tap to accept."
4. User double-taps their phone screen (or a button on the bracelet) to accept the new route.

Alignment to best practices

- **Differentiated Alerts:** The severity of the alert matches the intensity of the notification (e.g., a minor delay is a small banner, a missed stop is a full modal/audio interruption).
- **Actionable Feedback:** Every alert is paired with a clear "what to do next" (e.g., "Show New Route," "Double-tap to accept").
- **Error Prevention:** The app proactively re-routes the user *before* they miss their stop, preventing a navigation failure.

Innovation

- **Haptic "Acknowledge" Gesture:** When an alert is sent, the user can tap their bracelet *once* to immediately silence the audio or visual notification. This confirms they have "heard" the alert and allows them to continue their journey with minimal interruption, which is especially useful in social situations.

Phase 6:

Question	Answer & Motivation
1. Will the user be trying to produce the effect?	**NO (Initially).** The user is passive (riding the bus). However, once the alert triggers (haptic/visual), the user *becomes* active, trying to resolve the issue.
2. Will the user see the correct control?	**YES.** The red "Notification Banner"and the distinct "short-short-short" vibration pattern are designed to break the user's focus and draw attention immediately.
3. Will the user see that the control produces the desired effect?	**YES.** The banner text "UPDATE: Bus 24 delayed" explains the situation clearly. The TTS prompt "Double-tap to accept" gives explicit instruction on how to proceed.
4. Is there another control that the user might select instead of the correct one?	**YES.** In a crowded vehicle, a user might accidentally tap the map instead of the small banner. **Improvement:** Use a full-screen Modal Alert for critical rerouting to prevent miss-taps.
5. Will the user understand the feedback to proceed correctly?	**YES.** Once tapped, the alert disappears, and the navigation updates. The "Haptic Acknowledge" gesture allows instant feedback without looking at the screen.

Task 3. Navigate in a Noisy or Crowded Environment

User	Deaf or blind commuter traveling alone.
Purpose	Know when to exit the bus or metro safely.
Starting point	Active navigation mode.
What they do	Follow haptic or visual cues that signal the correct stop.
Context	Dense crowd, moving vehicle, limited visibility.
Steps	GPS tracks progress → vibration/voice alert at next stop → confirmation signal when stop is reached.

Phase 5: Alternative User Scenario

Scenario A: Haptic-Only Navigation (Alternative Success Case - Blind User) Context: Mircea is traveling on a very old, noisy metro train during rush hour. The audio announcements are broken and inaudible due to the noise. He cannot take his phone out safely due to the crowd.

Start: Mircea holds the handrail with one hand and relies entirely on his HearWay bracelet.

System Reaction (Progress): As the train leaves the station *before* his destination, the bracelet vibrates with a specific code: two short pulses (indicating "Next stop is yours").

Action: Mircea prepares to move towards the door, guided solely by this tactile cue.

Trigger: The train decelerates and enters the station. The bracelet emits a "Heartbeat" pulse that matches the train's slowing speed.

Feedback: When the train comes to a complete halt, the bracelet delivers the distinct "Safe to Exit" pattern (one long, strong buzz). Mircea steps out confidently, knowing he is at the correct station without having heard a single announcement.

Important components of the interface

- **Proximity Progress Bar:** A large visual bar that fills up as the user gets closer to their target stop (for deaf users).

- **"Get Off Now" Alert:** A bright, full-screen flashing notification (e.g., green screen) combined with a specific, strong vibration pattern from the bracelet.
- **Audio Proximity Cue:** TTS announcements that increase in frequency (e.g., "Next stop is Central Station" ... "Arriving at Central Station" ... "Exit here").
- **Haptic Directional Guidance (for walking):** When exiting a station, the bracelet can vibrate on the left or right side to indicate which way to turn.

Data Types Used

- **GPSLocator**: Provides high-frequency, real-time user coordinates.
- **GeofenceObject**: A digital boundary (e.g., 50-meter radius) around the destination bus/metro stop.
- **HapticCommand**: A data packet sent to the bracelet (e.g., vibrate_pattern: 'STOP_APPROACHING', vibrate_direction: 'LEFT').

User interactions

Deaf User:

1. The user is on a crowded bus and looking at their phone.
2. The app shows a "Proximity Progress Bar" filling up as they approach the stop.
3. When the bus is 1 minute away, the screen shows a large text alert: "NEXT STOP: Central Station."
4. As the bus pulls into the stop, the screen flashes bright green with "GET OFF NOW," and the bracelet gives a continuous, strong buzz.

Blind User:

1. The user is on a noisy metro; the phone is in their pocket.
2. The app runs in the background. 1 minute from the stop, the bracelet gives a "prepare" pulse (e.g., *pulse... pulse...*).
3. The phone's TTS announces: "Arriving at Central Station. Prepare to exit."
4. When the metro doors open at the correct stop, the TTS says "Exit here," and the bracelet gives a continuous, strong buzz.

Alignment to best practices

- **Just-in-Time Alerts:** Feedback is delivered precisely when needed to prevent cognitive overload. The "get off" alert triggers *at* the stop, not 2 minutes before.
- **Environmental Adaptation:** The system relies on haptics and high-contrast visuals, which are effective even when the user cannot hear announcements or is being jostled in a crowd.
- **Clear Feedback:** The "Get Off Now" alert is impossible to misinterpret, using strong visual and haptic cues.

Innovation

Geofence Stop Confirmation: The app doesn't just rely on GPS *proximity*. It uses geofencing combined with the phone's accelerometer data. The *final* "Get Off Now" alert only triggers when the (1) GPS is inside the stop's geofence AND (2) the accelerometer confirms the vehicle has *stopped moving*. This prevents "false positives" if the bus stops in traffic just *before* the actual stop.

User Scenario: Navigating the Exit in a Crowded Bus (Task 3)

Goal: The user must safely and confidently exit a public transport vehicle at the correct stop, despite the crowded and noisy environment, relying on non-auditory cues.

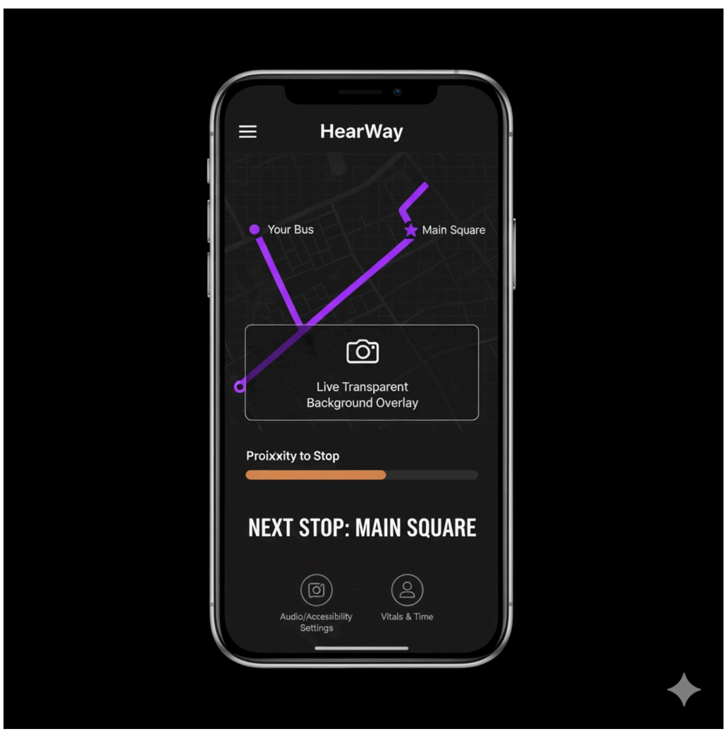
Starting Point: The user is authenticated and is currently on an active trip screen or has navigation running in background mode, two stops away from their destination.

A. Success Case: Safe & Confident Exit

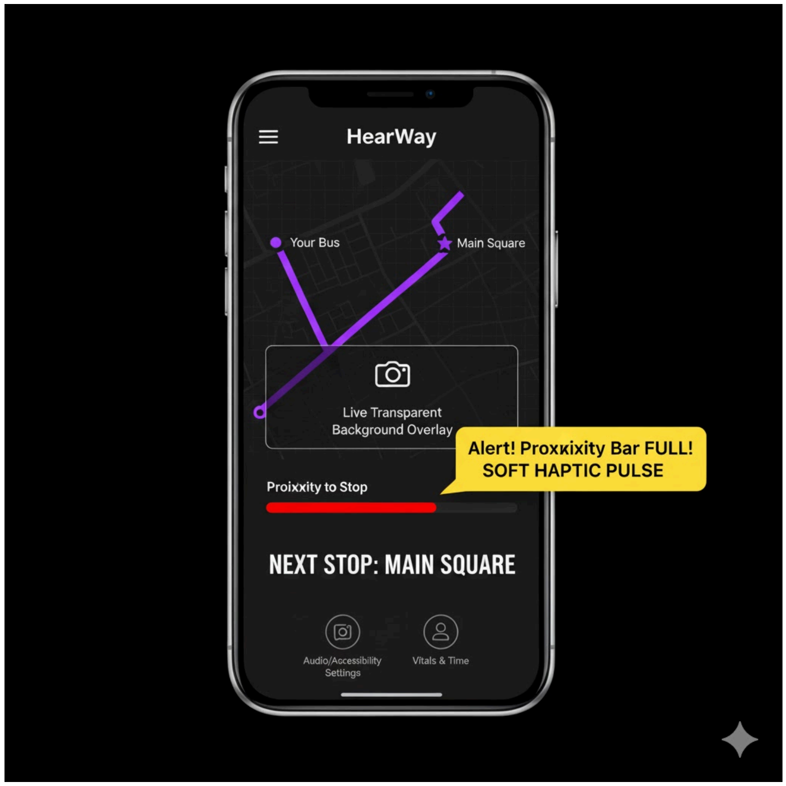
As the bus approaches the stop, the system begins its **Just-in-Time Alert** sequence. The **Proximity Progress Bar** on the Navigation Screen fills up completely for Ana (Hard-of-Hearing), changing to a high-contrast red, indicating the imminent arrival. Simultaneously, the smart bracelet gives a specific, low-level **haptic pattern** to Mircea (Blind). Mircea, having the phone locked in their pocket, double-taps the screen, and the Text-to-Speech (TTS) announces, "**Prepare to exit.**"

When the system confirms the vehicle has physically stopped (validated by GPS **Geofence** *and* accelerometer data), the system triggers the final, critical cue. The screen flashes bright green and displays the high-contrast text overlay: "**GET OFF NOW**". Simultaneously, the bracelet issues a continuous, strong **haptic buzz**—the "Get Off Now" Alert. The user immediately recognizes this unambiguous cue and exits the vehicle safely and confidently, completing the task. The system then announces, "**Journey successfully completed.**"

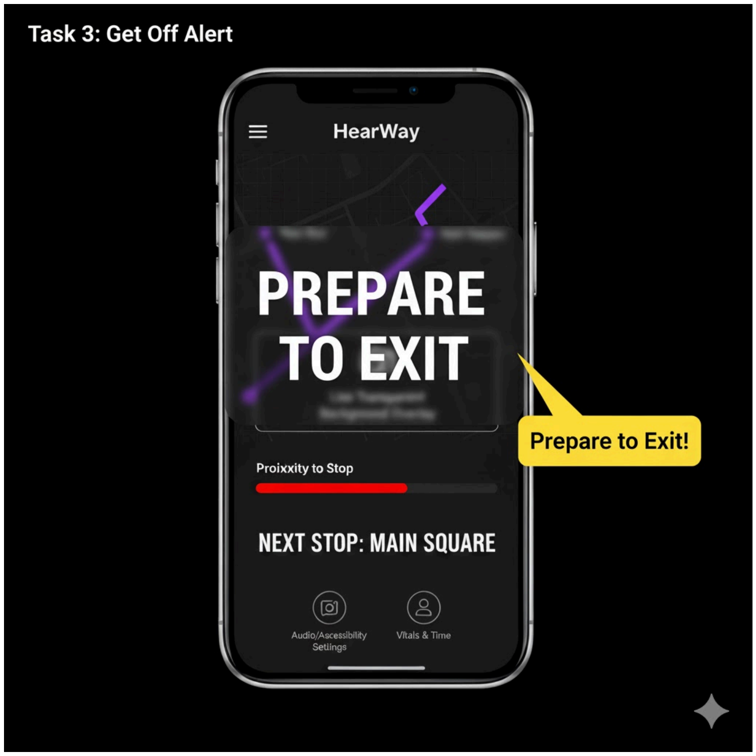
-first screen, showing the user's active journey on the HearWay app, with the map and essential navigation details, two stops away from the destination.



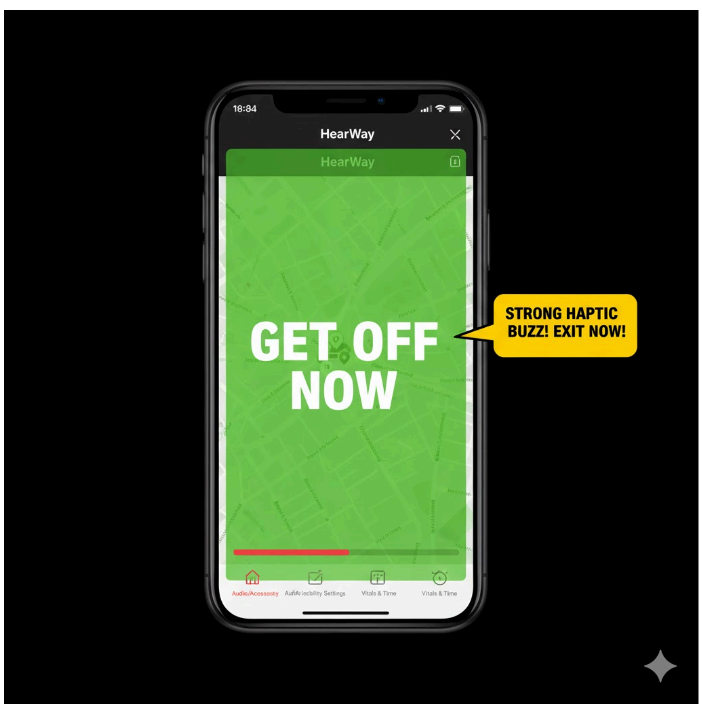
-second screen: As the bus gets 20 seconds away from the destination, the **Proximity Progress Bar** fills up completely and turns red. A soft haptic pattern is felt by the user.



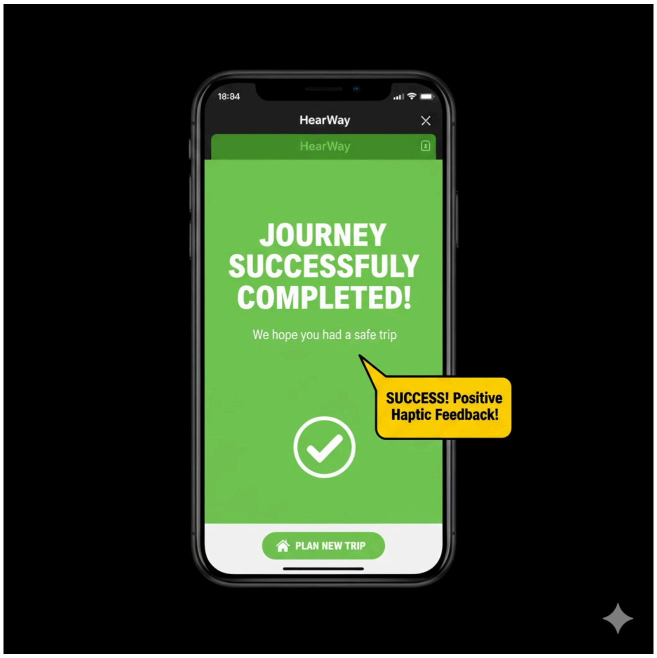
-third screen for the **Success Case** - **Navigate in a Noisy or Crowded Environment**:This screen displays the "PREPARE TO EXIT" instruction, making it visually clear for the user that the destination stop is next. This is also accompanied by a Text-to-Speech (TTS) announcement.



-Fourth screen for the **Success Case** of **Task 3: Navigate in a Noisy or Crowded Environment**:This is the critical "GET OFF NOW" alert. The system has confirmed the bus has stopped at the correct destination. The screen vividly displays "**GET OFF NOW**" with a bright green background, accompanied by a strong, continuous haptic buzz from the wearable.



-final screen for the **Success Case** of **Task 3: Navigate in a Noisy or Crowded Environment**. This screen confirms the successful completion of the journey, often accompanied by a positive haptic feedback or a TTS message like "Journey successfully completed."

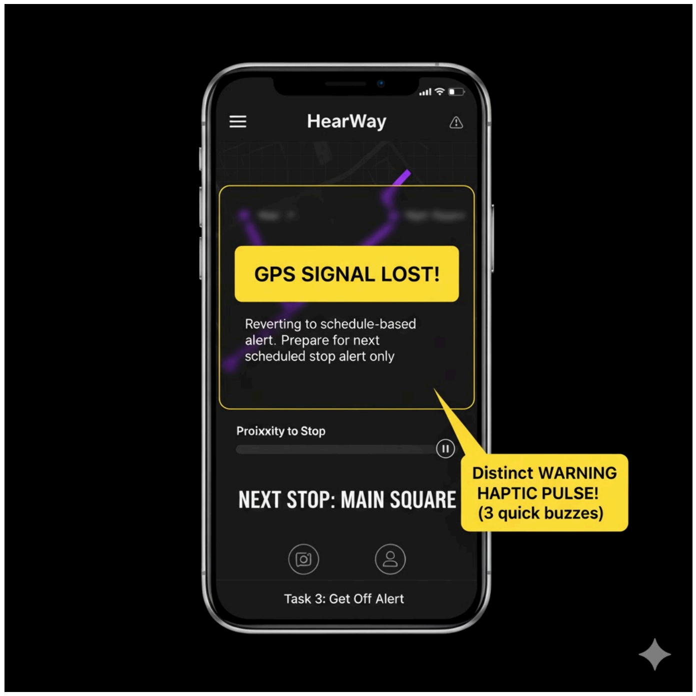


B. Error Case: GPS Signal Loss in a Tunnel

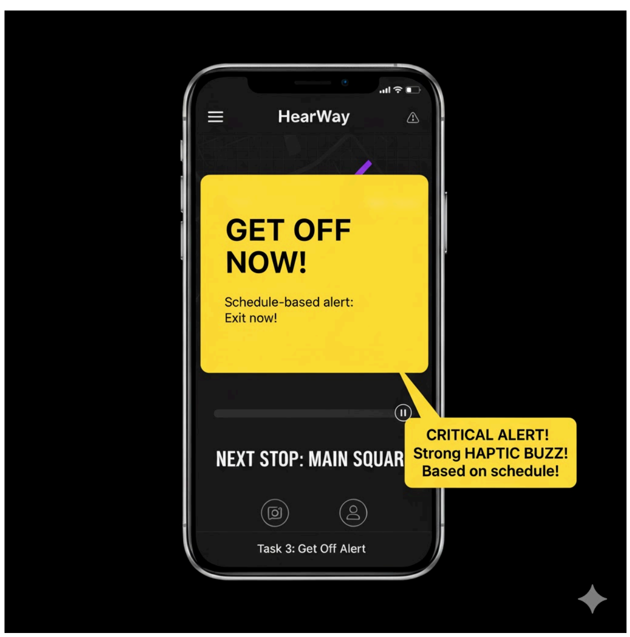
The vehicle enters an area with poor signal (e.g., a metro tunnel) just before the critical alert zone. The user continues to wait for the haptic alert.

System Error Reaction: The system displays a high-contrast **Warning Dialog** and immediately triggers a distinct **warning haptic pattern** (three quick pulses) to capture attention. Since GPS data is temporarily lost, the **Proximity Progress Bar** freezes. The TTS announces, "**GPS Signal Lost. Reverting to schedule-based alert. Prepare for the next scheduled stop alert only**." This prevents a hazardous "false positive" exit cue. The system continues to estimate arrival based on the scheduled time. When the time slot for the stop is reached, the system prioritizes safety and issues the definitive strong **haptic pattern** and "**GET OFF NOW**" visual cue, prompting the user to exit regardless of GPS status.

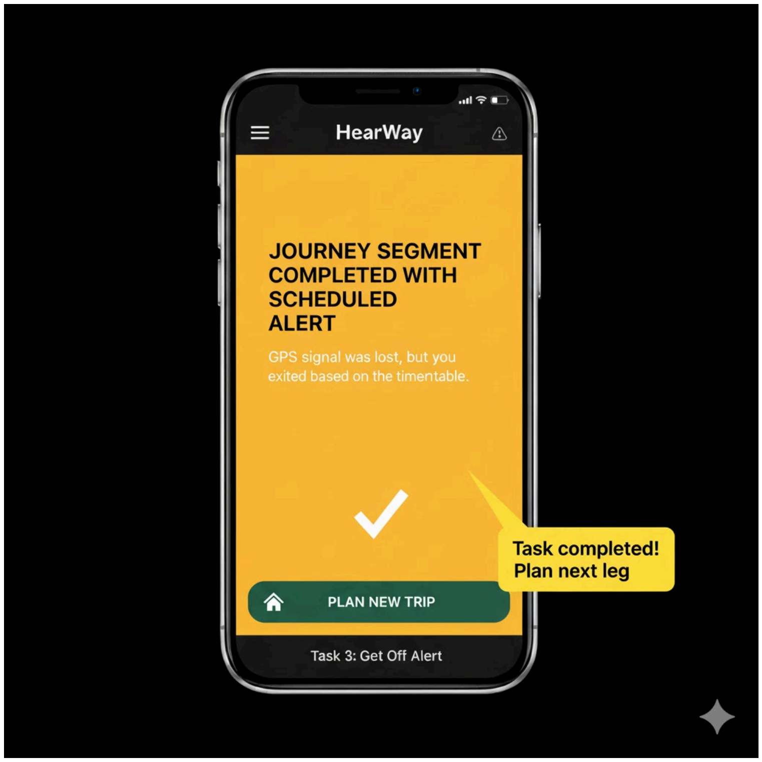
-first screen for the error scenario, depicting the **GPS Signal Lost** warning. The system detects a loss of signal and immediately alerts the user with a distinct warning haptic pattern and an on-screen message.



-second screen for the **Error Case: GPS Signal Loss in Tunnel** for **Task 3: Navigate in a Noisy or Crowded Environment**. Even with lost GPS, the system prioritizes safety. Based on the *scheduled arrival time* (as the system reverted to schedule-based alerts), the critical "GET OFF NOW" alert is eventually issued, ensuring the user still receives an unambiguous cue to exit.



-third and final screen for the **Error Case** of **Task 3: Navigate in a Noisy or Crowded Environment**. This screen confirms that despite the GPS error and scheduled alert, the journey segment was still completed. It provides closure and allows the user to plan their next step.



C. Motivation for Design Choices

The design is based on the principle of **Cross-modal Redundancy** to overcome the core challenges of the domain: **Sound Awareness Gap** and **Environmental Complexity**.

1. **Haptics and High Contrast:** Utilizing a specific, strong **vibration pattern** ensures the cue is received even when the user is jostled in a crowd or cannot see/hear standard announcements. The high-contrast green/text alert is a key part of **WCAG 2.2 Compliance**.
2. **Safety Innovation:** The **Geofence Stop Confirmation** is critical for **Error Prevention**. By requiring the accelerometer to confirm the vehicle has stopped *in addition* to GPS proximity, we prevent the "false positive" scenario where a bus stops prematurely in traffic, greatly enhancing user safety and trust.

Phase 6:

Question	Answer & Motivation
1. Will the user be trying to produce the effect?	**YES.** The user is anxiously waiting for the signal to exit. They are monitoring the app state closely.
2. Will the user see the correct control?	**YES.** The control is the entire screen flashing green and the bracelet vibrating continuously. It is impossible to miss.
3. Will the user see that the control produces the desired effect?	**YES.** The text "GET OFF NOW" is imperative and unambiguous. The strong haptic buzz is distinct from the softer "proximity" pulses, clearly signaling a change in state (Arrival).
4. Is there another control that the user might select instead of the correct one?	**NO.** In this "Success Case" scenario, the interface is simplified to show only the critical exit alert, removing distractions.
5. Will the user understand the feedback to proceed correctly?	**YES.** The feedback *is* the instruction. The combination of GPS geofencing and accelerometer data ensures the alert only happens when the bus actually stops, giving the user confidence to step out.

Task 4. Find the Nearest Accessible Transit Stop

User	New city visitor or student with sensory disability.
Purpose	Locate the closest stop with visual signage or tactile/voice assistance.
Starting point	“Nearby Stops” map section.
What they do	Use GPS search to identify accessible stations.
Context	Outdoor street navigation, possibly limited attention while walking.
Steps	Open map → search → display accessible options → start route.

Phase 5

Scenario A: Visual Discovery (Success Case – Deaf User)

Context: Ana is walking through an unfamiliar neighborhood looking for a stop that has clear visual signage.
Start: On the “Nearby Stops” screen, she taps the large **GPS Locator button**.
System Reaction: The map recenters and highlights three nearby accessible stops with large, high-contrast pins.
Action: Ana taps the **Filter** button and selects **“Visual Signage.”**
System Reaction: Pins update instantly to show only visually accessible stops.
Decision: She taps “Main Street Stop – 180 m.”
Feedback: A large preview card appears with walking time. Her bracelet gives **one short vibration**, confirming selection.
Outcome: She taps **Start Navigation**, and the map switches to a bold walking route with large arrows.

Scenario B: Audio/Haptic Discovery (Alternative Success Case – Blind User)

Context: Mircea steps outside the metro exit and needs the nearest stop with tactile paving.
Start: He opens “Nearby Stops.”
System Reaction: The TTS announces:

"Two accessible stops nearby. Stop 1: Main Street, 200 meters. Stop 2: Central Park, 400 meters."
Action: He swipes right to hear Stop 2, then swipes left to go back and selects Stop 1 with a double-tap.
System Reaction: The bracelet provides a **directional buzz** (left wrist), signaling the walking direction.
Feedback: A TTS message confirms:
"Navigation started. Walk forward 200 meters."
Outcome: Mircea heads toward the stop guided entirely via haptics.

Important components of the interface

- **GPS Locator Button** – instantly centers map on user position
- **Accessible Stop Cards** – each card shows distance, accessibility type, and estimated walking time
- **Map View (optional)** – high-contrast map with large icons
- **"Start Navigation" Button** – starts haptic or voice guidance
- **Filter Button** – filters results by accessibility feature (visual, tactile, audio)

Data Types Used

- `StopObject`: {id, name, coordinates, accessibilityFeatures[]}
- `UserProfile`: {isBlind, isDeaf, preferredAccessibility}
- `GeoJSON`: for displaying stops on map
- `NavigationRoute`: walking route with accessibility flags

User interactions

Deaf user:

- Open map view → sees high-contrast pins and icons.
- Taps "Filter" → selects "Visual Signage."
- Chooses a stop → large visual arrows and vibration feedback confirm selection.

Blind user:

- Opens Nearby Stops → TTS: “Two accessible stops nearby. Stop 1, Main Street 200 meters. Stop 2, Central Park 400 meters.”
- Swipes right to hear the next option → double-taps to select.
- Bracelets vibrate directionally to guide walking.

Alignment to best practices

- WCAG 2.2 compliant (visual contrast + full screen-reader support).
- Cross-modal redundancy (visual + haptic + audio).
- Reduced cognitive load (only nearest 3–5 stops displayed).

Innovation

Accessibility Radar Mode: The bracelet gently vibrates faster as the user turns toward the nearest accessible stop — like a sonar beacon for navigation.

Phase 6:

Question	**Answer & Motivation**
1. Will the user be trying to produce the effect?	**YES.** The user is in an unfamiliar location and actively needs to find a specific type of stop (one with visual signage) to travel safely.
2. Will the user see the correct control?	**YES.** The "GPS Locator" and "Filter" buttons are prominent on the map interface. The pins on the high-contrast map are clearly visible.

3. Will the user see that the control produces the desired effect?	**YES.** The "Filter" button clearly implies narrowing down options. Selecting "Visual Signage" immediately updates the map pins, providing instant visual feedback.
4. Is there another control that the user might select instead of the correct one?	**POSSIBLE.** The user might tap a specific stop *before* filtering, wasting time checking details for inaccessible stops. **Improvement:** Highlight the "Filter" button or prompt "Set accessibility needs" on first use.
5. Will the user understand the feedback to proceed correctly?	**YES.** Selecting a stop triggers large visual arrows and vibration feedback, confirming the selection. The "Accessibility Radar" provides continuous feedback while walking.

8.2. Communication and Assistance Tasks

Task 5. Request Help from Transit Staff

User	Deaf traveler at a station.
Purpose	Ask for information quickly without speech or lip-reading.
Starting point	Assistance section or quick-access menu.
What they do	Select or edit a pre-written text message ("Where is platform 3?").
Context	Busy station with staff but no sign interpreter.
Steps	Open quick messages → choose template → show or send a message.

Phase 5: User Scenarios

Scenario A: Quick Visual Help Request (Success Case – Deaf User)

Context: Ana is at a crowded station and needs directions to Platform 3.
Start: She taps the **Assistance** button from the quick-access menu.
Action: A grid of large message tiles appears. She taps **"Where is Platform 3?"**
System Reaction: The app displays a **full-screen message**:
"I am Deaf. Where is Platform 3? Please write your response."
Feedback: The bracelet provides one short vibration to confirm the message is ready to show.
Outcome: She holds up the screen to the transit staff, who immediately understand.

Scenario B: Voice-to-Text Help Request (Alternative Success Case – Blind User)

Context: Mircea cannot see signage and the station is noisy.
Start: He uses a voice command: "Ask for help."
System Reaction: TTS responds:
"Say your question."
Action: Mircea says: "Where is Platform 3?"
System Reaction:

- The app sends the message digitally if possible
- TTS confirms: “Message sent.”
- The bracelet gives a **confirmation buzz**
  Outcome: A staff member receives the message and approaches to guide him.

Important components of the interface

- **Quick Message Grid:** Common messages in large, high-contrast tiles
- **Custom Message Input:** Allows text or voice dictation
- **Display Mode:** Full-screen readable message (“I am deaf, please write your response”)
- **Send Button:** Sends to connected station systems if supported
- **Haptic Confirmations:** Short vibration to confirm sent/displayed message

Data Types Used

- `MessageTemplate`: {id, text, category, icon}
- `UserProfile`: accessibility mode, preferred communication method
- `MessageHistory`: saved recent messages

User interactions

Deaf user:

- Open “Assistance” screen → taps “Where is platform 3?”
- The screen goes fullscreen with large text.
- If connected, the app shows: “Message sent to staff terminal.”

Blind user:

- Uses voice command: “Ask for help.”
- App TTS: “Say your question.”
- User: “Where is platform 3?”
- App sends message → TTS confirmation: “Message sent.”
- The bracelet gives a confirmation buzz.

Alignment to best practices

- Text and audio redundancy ensures universal access.
- Large, single-action buttons prevent mis-taps.
- WCAG-compliant text contrast for visibility.

Innovation

- **Dual Mode Messaging:** If station systems allow, HearWay automatically connects via BLE/NFC to transmit the question digitally to nearby staff devices, displaying the user’s profile and communication preferences instantly.

Phase 6:

Question	**Answer & Motivation**
1. Will the user be trying to produce the effect?	**YES.** The user is confused/lost in a busy station and needs information quickly but cannot communicate verbally.

2. Will the user see the correct control?	**YES.** The "Assistance" or "Quick Messages" section is accessible from the main menu. The grid of messages is designed with large tiles.
3. Will the user see that the control produces the desired effect?	**YES.** The tiles contain clear text (e.g., "Where is platform 3?"). The expectation is that tapping it will display or send the message.
4. Is there another control that the user might select instead of the correct one?	**NO.** The interface is simplified for quick access. However, distinguishing between "Show Screen" (to a human) and "Send Digital Alert" (to a system) must be clear to avoid confusion.
5. Will the user understand the feedback to proceed correctly?	**YES.** The screen immediately goes full-screen with large text, acting as a digital placard. This is unmistakable feedback that the phone is ready to be shown to staff.

Task 6. Report an Issue or Emergency Silently

User	Deaf or blind commuters facing a problem.
Purpose	Notify authorities without needing a phone call.
Starting point	Emergency button or “Report Issue” tab.
What they do	Send a silent alert including GPS and issue type.
Context	On public transport or the street, where speaking is impossible.
Steps	Tap emergency → confirm issue → automatic message sent to staff.

Phase 5: User Scenarios

Scenario A: Silent Emergency Report (Success Case)

Context: A suspicious person is following the user in a nearly empty tram.
Start: The user discreetly **long-presses the wearable button**.
System Reaction:

- A subtle haptic pulse confirms initiation
- A small banner appears: “Reporting initiated…”
  Action: The user sees (or hears via TTS) the **Emergency Type selection**:
- Medical
- Security
- Harassment
- Fire
- Other
  Decision: The user taps **“Security.”**

System Reaction:

- Report is sent with GPS location
- Wearable gives a **double haptic pulse**
- Phone shows: “Emergency reported. Help is on the way.”
  Outcome: The user remains discreet but reassured.

Scenario B: Accidental Activation (Error Case)

Context: Mircea accidentally presses the emergency button while adjusting his coat.
Start: The wearable gives the subtle initiation pulse.
System Reaction:

- Phone shows: “Reporting initiated…”
- Confirmation screen appears
  Action: Mircea does nothing.
  System Recovery:
- After 5 seconds, the system cancels
- Wearable gives a **short cancellation buzz**
- Phone returns to the previous screen
  Outcome: No false alert is sent.

Important components of the interface

- **Emergency Button:** Large, always visible in app and on wearable
- **Alert Type Selector:** Simple icons or voice menu for blind users
- **Live Status Screen:** “Alert sent. Help on the way.”
- **Auto Location Share:** Background GPS tracking
- **Silent Mode:** Ensures discretion during emergencies

Data Types Used

- `AlertObject`: {type, timestamp, location, status}
- `UserProfile`: {emergencyContacts[], disabilityType}
- `GeoLocation`: coordinates + address
- `StatusTracker`: monitors delivery confirmation

User interactions

Deaf user:

- Long-press bracelet → continuous vibration confirms activation.
- The app shows a red screen: “Emergency sent.”
- Visual countdown in case of accidental trigger (cancel within 5 seconds).

Blind user:

- Press and hold the physical button → short buzz confirms.
- TTS: “Emergency alert sent to contacts.”
- If safe, voice prompt: “Say alert type: Medical, Lost, or Danger.”
- Bracelet vibrates once for success.

Alignment to best practices

- One-step access minimizes interaction under stress.
- Distinct haptic pattern ensures confirmation without sight/sound.
- WCAG-compliant red/white contrast for visibility.

- **Stealth Mode:** When triggered in silent mode, HearWay automatically sends the alert while keeping the screen dim to avoid drawing attention.

User Scenario: Reporting an Emergency Silently (Task 6)

Goal: The user must silently and quickly report an issue or emergency to transit authorities or emergency services without speaking or drawing attention to themselves.

Starting Point: The user perceives a dangerous situation (e.g., suspicious behavior, medical emergency) while on public transport or at a station.

A. Success Case: Silent Emergency Report

Upon perceiving a dangerous situation, the user immediately **long-presses a designated button on their smart wearable** (e.g., a smart watch or a dedicated HearWay bracelet button). This action triggers a discreet but persistent **haptic pulse** on the wearable, confirming that the reporting process has been initiated. If the phone screen is active, a small, non-intrusive "**Reporting initiated...**" banner appears at the top, without drawing attention.

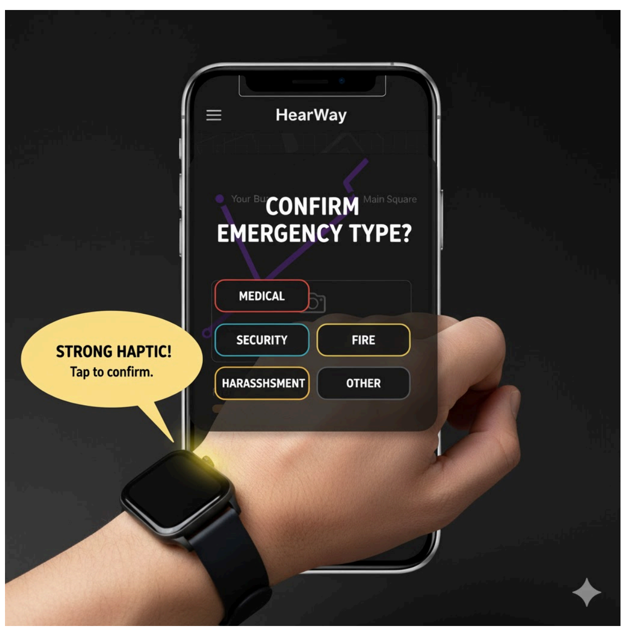
Immediately after initiation, the user receives a stronger, distinct **haptic pattern** on the wearable and, if the phone is active, a clear visual prompt asking "**Confirm Emergency Type?**" with large, high-contrast buttons for options like "Medical," "Security," "Fire," "Harassment," or "Other." The user quickly **taps the appropriate option** on the screen or performs a specific, predefined gesture (e.g., another long-press for "Security") on the wearable.

Once the emergency type is confirmed, the report is instantly sent to the designated authorities (transit security or 911, along with the user's precise GPS location). The system then provides a final, reassuring **double haptic pulse** on the wearable and displays a full-screen, high-contrast green message on the phone: "**Emergency Reported. Help is on the way. Remain calm.**" The message explicitly confirms that the location has been sent, providing crucial reassurance to the user in a stressful situation.

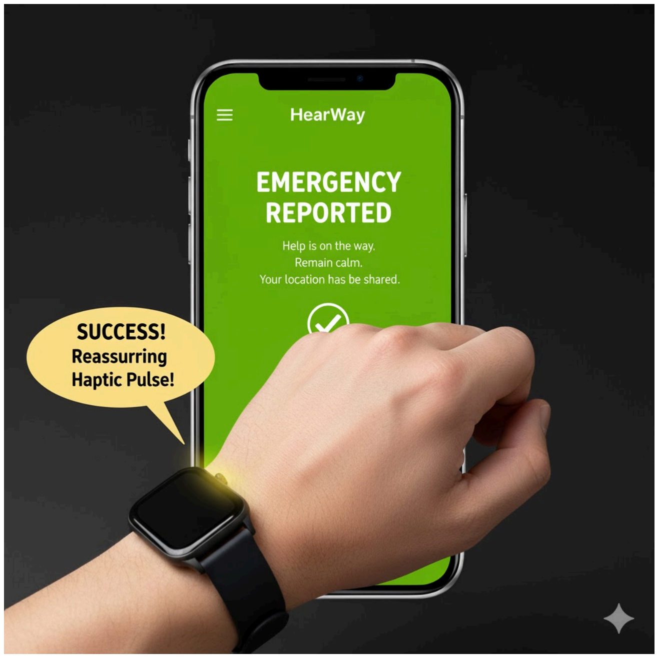
-first screen, showing the user in a normal app state (e.g., on a navigation screen), just before initiating the emergency report. The user perceives a threat and is about to perform the "long-press" action on their wearable.

-second screen for the **Success Case** of **Task 6: Report an Issue or Emergency Silently**: This screen shows the system's immediate response after the user initiates the report. A discreet "Reporting initiated..." banner appears on the phone, and a persistent haptic pulse is felt on the wearable, confirming the action without drawing undue attention.

-Third screen for the **Success Case** of **Task 6: Report an Issue or Emergency Silently**: This screen shows the crucial "Confirm Emergency Type?" overlay. The user is prompted with large, high-contrast buttons to select the nature of the emergency, and receives a stronger haptic pattern to guide their attention.



-Fourth and final screen for the **Success Case** of **Task 6: Report an Issue or Emergency Silently**: This screen provides the ultimate confirmation that the emergency report has been sent. A full-screen, high-contrast green message assures the user that help is on the way and their location has been shared, accompanied by a reassuring double haptic pulse.

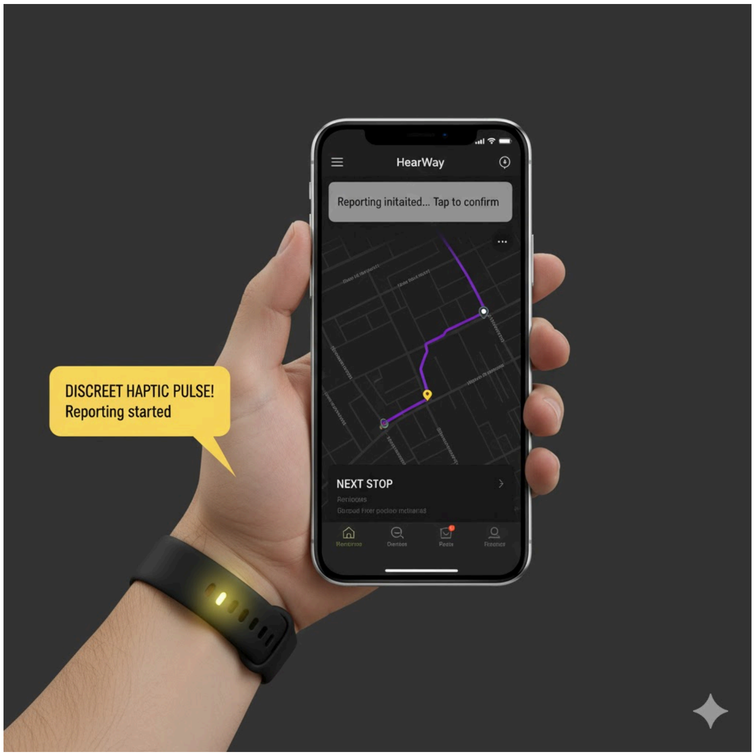


B. Error Case: Accidental Activation / No Confirmation

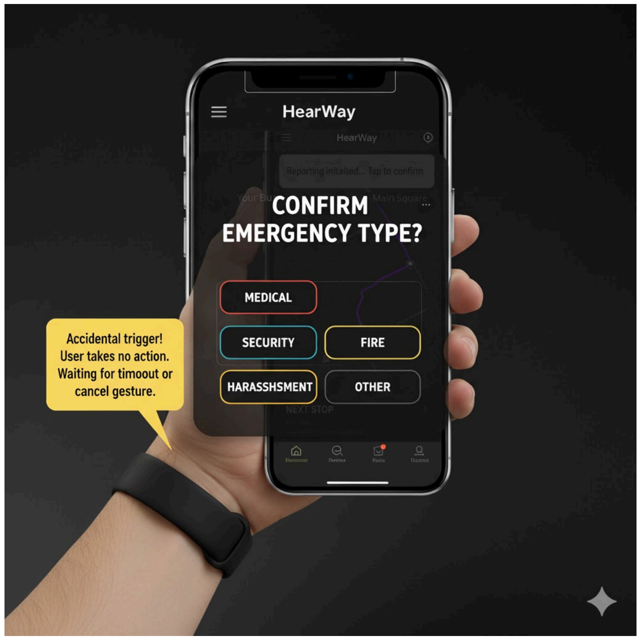
While reaching for something in their bag, the user accidentally **long-presses the designated button/gesture** on their wearable. The system responds by sending the initial discreet **haptic pulse**, and the "Confirm Emergency Type?" prompt appears on the phone screen (if active).

The user immediately realizes it was an accident and does not interact further with the confirmation prompt. After a short, predefined timeout of 5 seconds (or if the user performs a specific "cancel" gesture, like two quick taps), the system automatically **cancels the report**. A quick, single **haptic "cancellation" pulse** confirms that no report was sent. The app then reverts to its previous state (e.g., the navigation screen) without sending a false alarm, and no TTS announcement is made unless the user is in a private context where such an announcement would not draw unwanted attention.

-first screen for the error scenario. It shows the initial state after an accidental long-press, with the "Reporting initiated..." banner appearing on the phone and the discreet haptic pulse being sent to the wearable. The user realizes it was an unintended action.



-second screen for the **Error Case: Accidental Activation / No Confirmation** for **Task 6: Report an Issue or Emergency Silently**: This screen shows the "Confirm Emergency Type?" overlay appearing. However, since the user realized it was an accidental trigger, they **do not interact** with the confirmation options. The system is now awaiting a confirmation or a timeout.



-Third and final screen for the **Error Case: Accidental Activation / No Confirmation** for **Task 6: Report an Issue or Emergency Silently**: This screen shows the system automatically **cancelling the report** due to lack of confirmation (timeout). A discreet cancellation haptic pulse is sent, and the app reverts to its previous state without sending a false alarm.

C. Motivation for Design Choices

The design for Task 6 prioritizes **speed, discretion, and accuracy** in high-stress situations, directly addressing the core safety mission of HearWay.

1. **"Stealth Mode" Activation:** The use of a **long-press on a wearable button** or a specific gesture on a locked phone screen is motivated by the critical need for **discreet activation**. This design choice prevents users from drawing unwanted attention to themselves during an emergency, which is paramount for safety in situations involving harassment or security threats.
2. **Confirmation and Error Prevention:** A **two-step confirmation process** (initiate then confirm type) is implemented to prevent accidental false alarms while still ensuring rapid reporting. The immediate haptic feedback at each step provides non-visual assurance to the user, enhancing confidence under stress. The short timeout for confirmation is a crucial aspect for preventing unconfirmed reports from being sent inadvertently.
3. **GPS Integration:** The automatic inclusion of **GPS location data** with every emergency report is an essential feature for enabling rapid response by authorities, directly contributing to the primary safety objective of the HearWay project. This ensures that help can be dispatched precisely and efficiently, even if the user cannot verbally communicate their location.

Phase 6:

Question	Answer & Motivation
1. Will the user be trying to produce the effect?	**YES.** The user perceives a threat and wants to alert authorities without drawing attention to themselves (Stealth Mode)[10].
2. Will the user see the correct control?	**YES (Tactile).** Since this is a "Stealth" task, the user doesn't look for a visual button but feels for the physical button on the wearable/bracelet to perform the "Long-press"[11].
3. Will the user see that the control produces the desired effect?	**YES.** The user receives a **discreet haptic pulse**[12]. This non-visual confirmation is critical since looking at the screen might be dangerous.
4. Is there another control that the user might select instead of the correct one?	**RISK MANAGED.** An accidental short tap might not trigger it (requires long-press). If triggered accidentally, the "Confirm Emergency Type" step [13] acts as a safety buffer to prevent false alarms.
5. Will the user understand the feedback to proceed correctly?	**YES.** The two-step haptic feedback (initial pulse -> confirmation pulse) guides the user through the "Initiate -> Confirm" flow without needing to look at the device[14].

Task 7. Communicate with Rideshare or Taxi Drivers

User	Deaf users taking private transport.
Purpose	Communicate basic instructions easily.
Starting point	After booking a ride in the app.
What they do	Use text-to-speech or large-text templates (“Please stop here”).
Context	Inside a car or pickup zone under time pressure.
Steps	Select message → app displays or reads aloud → driver understands instruction.

Important components of the interface

- Pre-set message templates for quick communication. For example: “Please stop here”, “I am nearby”, “I cannot speak, thank you”
- Large text speech bubble display to show text clearly to the driver.
  High contrast, big font
- Live speech-to-text captioning to show Real-time subtitles for driver speech.
- Transparent/live camera background to see surroundings while reading.
  Similar to Google Live Translate background opacity mode
- Haptic confirmation to provide feedback when messages are sent.
- Voice navigation like saying out loud “Swipe right for next message” or
  “Double tap to send”
- Integration of Braille display support, which is an external hardware attachable to a mobile display.
- Haptic patterns to differentiate between feedback categories.
  For example short buzz is navigation, long buzz is confirmation
- Haptic menu navigation enables vibrations to map to menu states
- Tactile gesture shortcuts, for example two-finger slide left to cancel

Data Types Used

- Message templates: Each message has both a text and a text-to-speech component.
- User preferences: Accessibility preferences are persisted
- Voice integration logs: For debugging UX issues

- Transcription transcript: On-device audio to text transcription
- Camera feed overlay settings for transparent UI mode

User interactions

Deaf people

Open ride assist -> Quick buttons:
[I AM HERE] [PLEASE STOP HERE] [I AM DEAF] [THANK YOU]

Blind people

Pickup zone -> App auto opens Ride Assist -> Voice instructions:
“Swipe right to browse messages”
“Double tap to send”

Feedback

Visual confirmation for deaf
Voice confirmation for blind
Vibration confirmation for deaf-bind

Alignment to best practices

WCAG 2.2 accessibility supports Text scaling, contrast, captions and speech
Cross-modal redundancy enables Haptic, audio and text
Cognitive load reduction allows the usage of templates and simple gestures
Universal design enables the application usage with or without disability settings
Real-world constraints enables using the application under time pressure, in a noisy environment

Innovation

Transparent Live Background Text Overlay (For Deaf Users)

Allow users to see outside while reading UI to increase situational awareness. Works by running the camera behind the UI, having an adjustable background opacity slider, and displaying a large text overlay like HUD. Similar to Google Lens overlay text, or Samsung Translate assistant transparency mode.

Phone as Hearing-Aid-Like Caption Device

Use iOS Live Transcribe API (Speech framework) or Android Live Caption / Live Transcribe APIs to enable Real-time speech-to text to display a person’s voice like a subtitle

User Scenario: Communicate with Rideshare or Taxi Drivers (Task 7)

Goal: The user must effectively communicate pickup details, destination changes, or other important information with a rideshare/taxi driver, overcoming potential auditory communication barriers.

Starting Point: The user has booked a rideshare/taxi and is either waiting for the driver to arrive at the pickup spot or is already in the vehicle and needs to convey information.

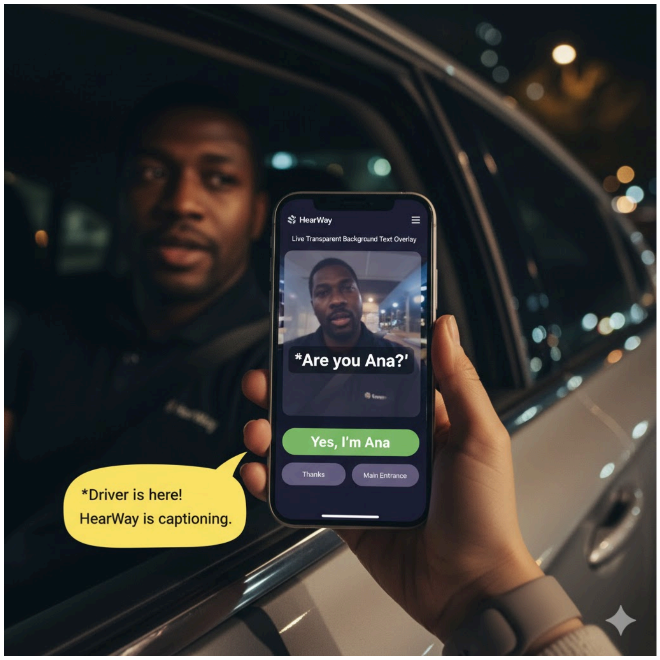
A. Success Case: Smooth Communication for Pickup For Deaf People

As the rideshare driver arrives at the pickup spot, they attempt to verbally confirm the user's identity or ask a question like, "Are you [User's Name]?" Ana (Hard-of-Hearing) might not hear the verbal query but sees the driver's lips moving or observes the driver looking directly at her. Mircea (Blind) receives a "Driver approaching" haptic alert from the HearWay app, followed by a Text-to-Speech (TTS) announcement of the driver's words: "Driver has arrived. Driver says: 'Are you Mircea?'" At this moment, Ana's phone, held up, displays a **Live Transparent Background Text Overlay**, captioning the driver's speech in real-time (e.g., "Are you Ana?").

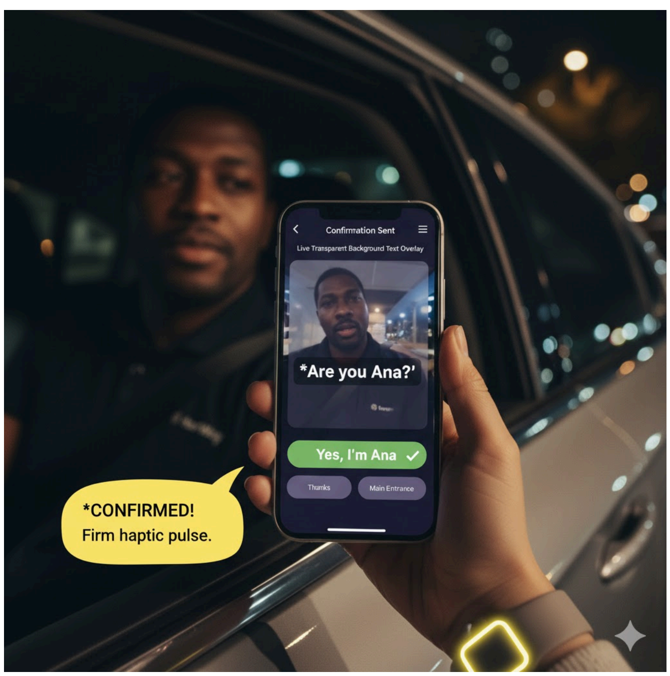
Upon seeing the caption or hearing the TTS, Ana immediately taps a large "**Yes, I'm Ana**" button displayed on her phone, or simply nods while showing the phone screen to the driver for visual confirmation. Mircea responds vocally with "Yes" or uses a predefined message by interacting with the app. The system briefly displays "**Confirmation sent**" to the user and sends a single, firm haptic pulse to the wearable, confirming the successful initial interaction.

If the driver then proceeds to ask for confirmation of the destination (e.g., "Confirming destination: [Address]?") or requires specific instructions (e.g., "Which entrance should I use?"), Ana's **Live Transparent Background Text Overlay** continues to caption the driver's speech in real-time. She can then quickly tap a pre-set message like "**Please use the main entrance**" or type a quick, custom response. For Mircea, the TTS reads the driver's question, and he selects an appropriate pre-set message (e.g., "Main Entrance"), which the HearWay app then speaks aloud to the driver, ensuring clear and efficient communication throughout the journey.

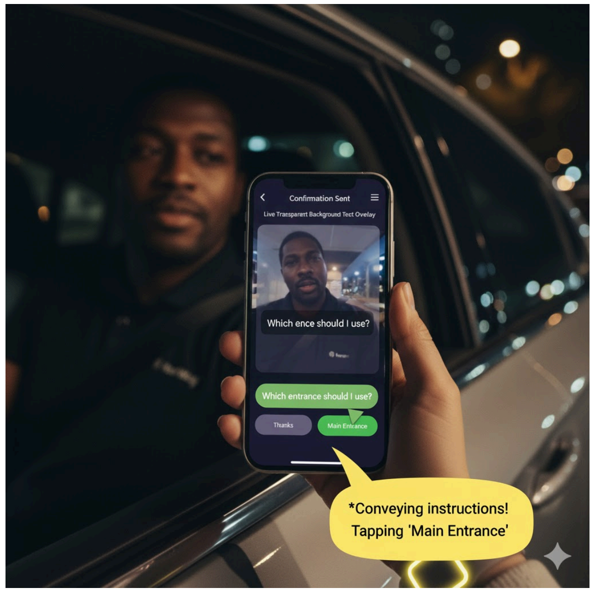
- first screen, depicting the user waiting for their rideshare driver. The driver has arrived and is attempting to initiate verbal communication, which the user (Ana) might not hear. The HearWay app is ready.



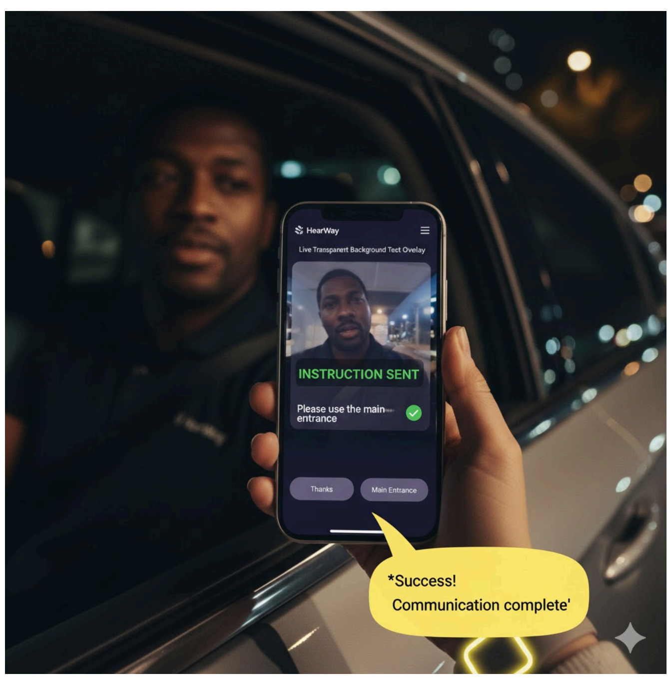
- second screen for the **Success Case** of **Task 7: Communicate with Rideshare or Taxi Drivers**: This screen shows the user confirming their identity to the driver. They tap the "Yes, I'm Ana" button, and the system provides confirmation on the screen and a firm haptic pulse.



- third screen for the **Success Case** of **Task 7: Communicate with Rideshare or Taxi Drivers**: This screen shows the user conveying specific instructions (e.g., "Main Entrance") to the driver. The Live Transparent Background Text Overlay continues to caption the driver, and the user taps a pre-set message for clear communication.



- fourth and final screen for the **Success Case** of **Task 7: Communicate with Rideshare or Taxi Drivers**: This screen confirms that the instruction has been successfully conveyed to the driver. The app shows the message sent and provides a clear, positive indicator of successful communication, allowing the journey to proceed smoothly.



B. Success Case: Smooth Communication for Pickup For BlindPeople

As the rideshare driver approaches the pickup location, the HearWay app automatically notifies the driver that the passenger is blind and may need verbal guidance when approaching the vehicle. The driver receives a brief message such as: "Passenger is blind. Please approach calmly, greet verbally, and offer assistance if requested."

At the same time, the user's app sends a distinct haptic alert to indicate "Driver approaching." A moment later, Mircea (Blind) hears a Text-to-Speech (TTS) announcement:
"Your driver is arriving shortly."

When the driver reaches the pickup spot, he follows the provided guidance—gently calling out from a respectful distance:
"Hello! I'm your driver. Are you Mircea?"

The HearWay app picks this up and immediately delivers it as TTS:
"The driver says: 'Hello! I'm your driver. Are you Mircea?'"

Mircea responds naturally by speaking:
"Yes, I'm Mircea."
If the environment is noisy, he can alternatively tap the volume button which transmits via TTS:
"Yes, I'm Mircea."

The app confirms the interaction through audio and haptics:
TTS: "Confirmation sent."
Haptic: a short pulse indicating success.

If the driver then asks for confirmation of the destination or needs further guidance—for example, "We're heading to 42 Market Street, correct?" or "Would you like assistance getting into the car?"—the HearWay app continues to capture and read the driver's questions aloud:

"Driver says: 'Would you like assistance getting into the car?'"

Mircea can answer verbally:
"Yes, please guide me to the door."
"No assistance needed."
"I'm on your right side."

Once the correct destination or instruction has been sent, the app provides another brief haptic confirmation and TTS feedback such as:
"Instruction sent to driver."

The driver, already aware that the passenger is blind, offers support safely and respectfully—guiding Mircea to the car door only if requested—and the ride begins without confusion or discomfort.

- First screen, Driver arrival notification for the **Success Case** of **Task 7: Communication for Pickup for BlindPeople**: The app notifies the driver that the passenger is blind and provides quick guidance on how to approach a blind person in public. This activity also provides a haptic cue that announces the arrival of the driver either on the phone or on a wearable device.

- second screen, Identity confirmation for the **Success Case** of **Task 7: Communication for Pickup for BlindPeople**: The user responds verbally, or taps the volume button. This activity provides a haptic cue that announces that the confirmation was sent either on the phone or on a wearable device.

- third screen, successful route communication for the **Success Case** of **Task 7: Communication for Pickup for BlindPeople**: To support situational awareness, the app delivers a pattern of haptic signals that indicate the remaining travel time, helping the user gauge how far they are from home.

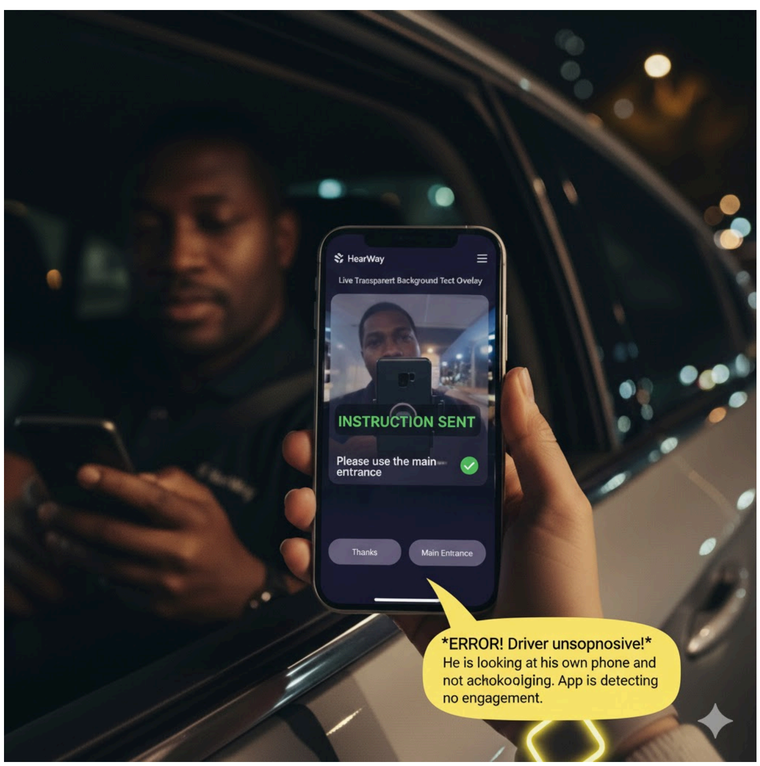
C. Error Case: Driver Non-Compliance / Lack of Attention

Ana attempts to show the driver her phone with a pre-set message confirming the destination (e.g., a large text display "Destination: Main Square"). However, the driver glances dismissively, indicates they don't understand, or is distracted (e.g., looking at their own phone), relying solely on verbal communication that Ana cannot perceive. The HearWay app detects a lack of acknowledged interaction (e.g., if the message remains displayed for 10 seconds without an input or acknowledgement).

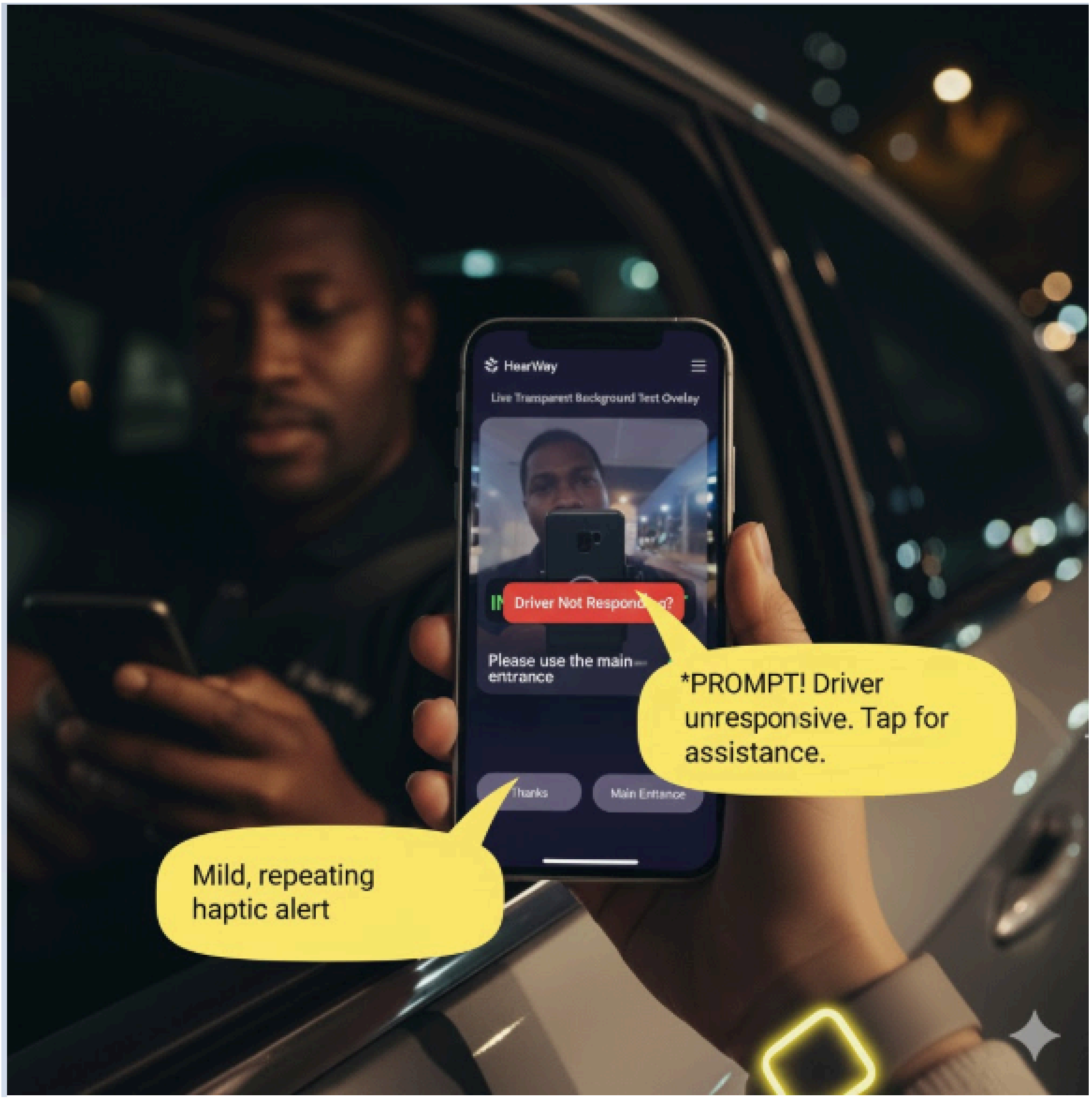
After this period of unacknowledged interaction, the system subtly prompts the user: "**Driver unresponsive? Tap here for further assistance.**" A mild, persistent haptic pattern is sent to the wearable, discreetly alerting Ana to the communication breakdown. On the phone screen, a small, red, pulsing "**Driver Not Responding?**" button appears, providing an immediate call to action.

The user taps the "Driver Not Responding?" button. The app then presents a menu of options: "**Call Driver (via text relay service)**", "**Report Issue to Rideshare Company**", or "**Cancel Ride**". If "Report Issue" is selected, the app immediately generates a pre-written message to the rideshare company (including the driver's ID and details of the communication issue), ready for immediate sending, empowering Ana to resolve the situation safely and effectively.

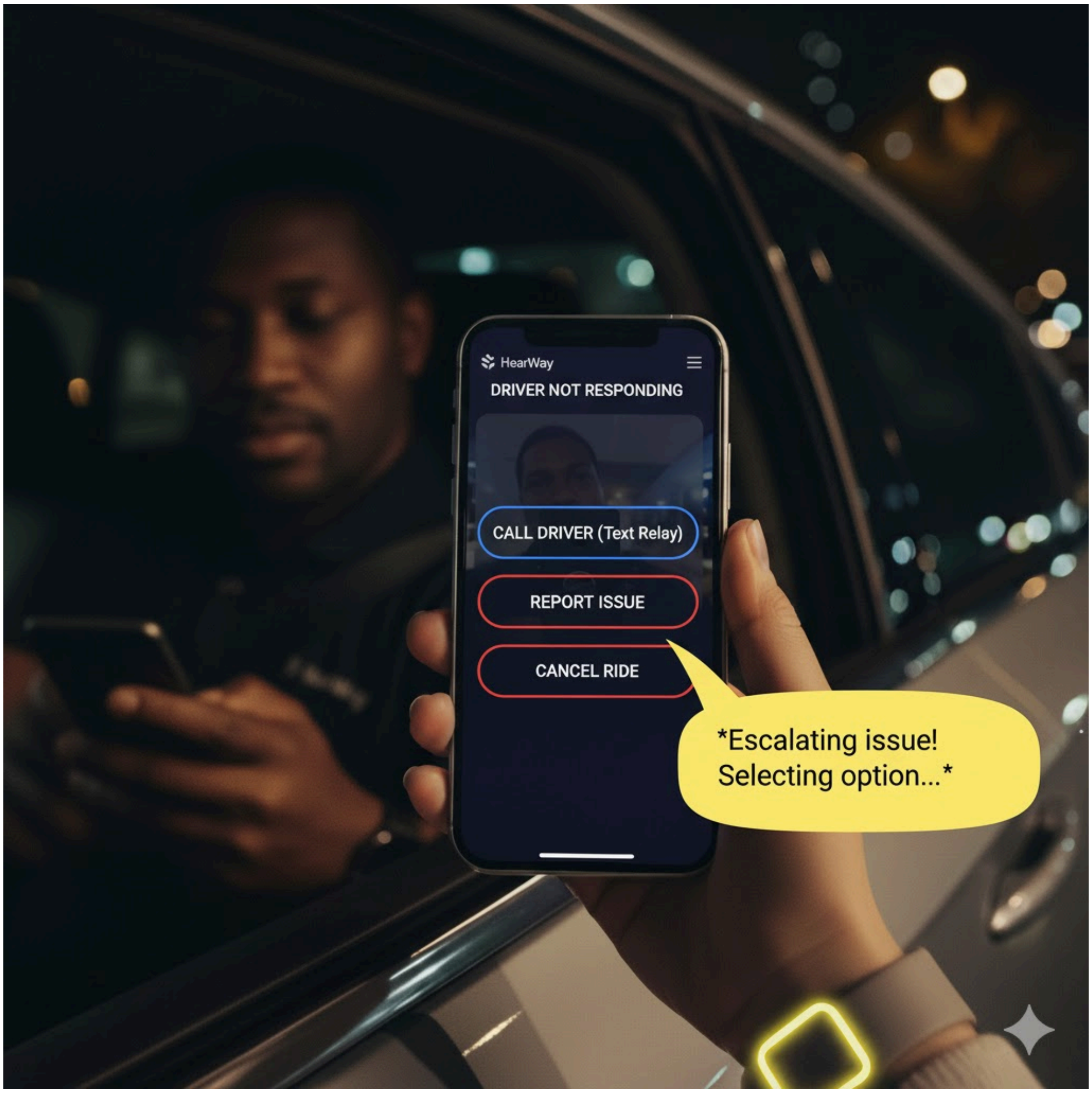
- First screen for the Error Case of Task 7: Communicate with Rideshare or Taxi Drivers: This image will accurately show the user's attempt to communicate, and the driver being unresponsive, with the app detecting this lack of engagement.



- second screen for the Error Case: Driver Non-Compliance / Lack of Attention for Task 7: Communicate with Rideshare or Taxi Drivers:This screen shows the system subtly prompting the user for further action after detecting the driver's unresponsiveness. A "Driver Not Responding?" button appears, accompanied by a gentle, repeating haptic alert.



- third screen for the Error Case: Driver Non-Compliance / Lack of Attention for Task 7: Communicate with Rideshare or Taxi Drivers: This screen shows the user engaging the "Driver Not Responding?" button, leading to a menu with options to escalate the issue. This empowers the user to either try an alternative communication method, report the issue to the rideshare company, or cancel the ride.



C. Motivation for Design Choices

The design for Task 7 is fundamentally focused on **bridging communication gaps, ensuring user control, and maintaining safety** in dynamic social interactions that are often challenging for individuals with sensory impairments.

1. **Real-Time Multimodal Communication:** The **Live Transparent Background Text Overlay** is a core innovation specifically designed for Deaf users. It allows them to simultaneously maintain critical environmental awareness (seeing the driver's face, the road ahead) while understanding spoken communication through real-time captions. For Blind users, the combination of **Text-to-Speech (TTS) for driver's speech** and the ability to input responses via voice or pre-set messages ensures a seamless and real-time interactive experience.
2. **Pre-set Messages & Visual Cues:** The inclusion of pre-set, commonly used messages (e.g., "My destination is X", "Please wait a moment") is crucial for minimizing the need for complex typing or voice commands, facilitating quick and efficient communication, especially under time pressure. Large text and clear buttons provide essential visual cues for rapid comprehension by both the user and the driver.
3. **Error Handling & User Empowerment:** The "Driver Not Responding?" escalation mechanism is a vital component for **user empowerment and safety**. It provides a clear, actionable path when direct communication fails, ensuring that the user is not left vulnerable due to a lack of understanding or an unresponsive driver. This proactive error handling strategy is designed to prevent frustration and mitigate potential safety risks that can arise from miscommunication or driver non-compliance, thereby reinforcing the HearWay project's commitment to independent and safe mobility.

Phase 6:

Question	**Answer & Motivation**
1. Will the user be trying to produce the effect?	**YES.** The user has a hearing disability. The application allowed her to understand the message conveyed by the driver.
2. Will the user see the correct control?	**YES.** The user will see a large button displaying template answers enabling the user to quickly respond in a real world environment.
3. Will the user see that the control produces the desired effect?	**YES.** The button is labeled with the exact phrase. The icon (e.g., a hand or stop sign) reinforces the meaning.
4. Is there another control that the user might select instead of the correct one?	**POSSIBLE.** If the car is moving on a bumpy road, the user may accidentally tap a nearby quick-message button (e.g., “Thank you”) instead of “Stop Here.”. To reduce this risk, the interface could increase spacing between the quick-message options or group messages into categories so that conflicting or easily mistaken options - such as selecting “Thank you” instead of “I’m feeling carsick, could you open the window” - do not appear on the same screen.
5. Will the user understand the feedback to proceed correctly?	**YES.** The app displays the message in a large speech bubble and activates TTS (Text-to-Speech) for the driver. A confirmation haptic pulse lets the user know the message was "spoken".

8.3. Scheduling and Repeating Routine Tasks

Task 8. Save Favorite Routes and Recurring Commutes

User	Regular commuter (student or worker).
Purpose	Save time when planning frequent trips.
Starting point	After completing a successful route.
What they do	Add route to favorites for one-tap reuse.
Context	Usually at home before leaving for work/school.
Steps	Open trip history → tap “Save Route” → confirm name → future quick access.

Important components of the interface

- Home Screen Shortcut to enable quick access to saved routes
- Trip History Screen to find last completed routes
- Route Detail Card to show route summary
- Save route screen to name and confirm a route
- Accessibility Mode Selector to enable the user profile Blind/Deaf/Combined

Data Types Used

- Route Info provides a history of Routes
- Favorite Routes enable quick reuse
- User preferences: Accessibility preferences are persisted
- Interaction Logs to improve UX

User interactions

Deaf people

Tap “History” -> Previously completed route -> Save route -> Pop-up Enter name
Output
Large text success screen.
Transparent-background UI so environment stays visible

Blind people

Long-press screen -> Voice feedback: “Trip complete. Save this route?” -> Volume up = Yes, Volume-down = No. If yes -> Dictation prompt: “Say route name”
TTS confirmation: “Route saved as Work Route”
Haptic pattern for success.
Navigation:
Swipe up/down to move through history.
Double tap to select
Long-press to preview route in audio
Or Braille keyboard
Confirmation via vibration codes

Alignment to best practices

WCAG 2.2 accessibility supports Text scaling, contrast, captions and speech
Cross-modal redundancy enables Haptic, audio and text
Cognitive load reduction allows the usage of templates and simple gestures
Universal design enables the application usage with or without disability settings
Real-world constraints enables using the application under time pressure, in a noisy environment

Innovation

Transparent Live Background Text Overlay (For Deaf Users)

Allow users to see outside while reading UI to increase situational awareness. Works by running the camera behind the UI, having an adjustable background opacity slider, and displaying a large text overlay like HUD. Similar to Google Lens overlay text, or Samsung Translate assistant transparency mode.

Phone as Hearing-Aid-Like Caption Device

Use iOS Live Transcribe API (Speech framework) or Android Live Caption / Live Transcribe APIs to enable Real-time speech-to text to display a person’s voice like a subtitle

Phase 6:

Question	Answer & Motivation
1. Will the user be trying to produce the effect?	**YES.** The user wants to avoid re-entering the same details next time and intends to save the successful route.
2. Will the user see the correct control?	**YES.** On the "Trip Summary" screen, the "Save Route" button is usually placed at the bottom. For blind users, the voice prompt explicitly asks "Trip complete. Save this route?".
3. Will the user see that the control produces the desired effect?	**YES.** The label "Save Route" or the "Heart/Star" icon is standard. The voice prompt is a direct question requiring a Yes/No action.
4. Is there another control that the user might select instead of the correct one?	**NO.** The flow is linear. The user is guided to make this decision before returning to the home screen.
5. Will the user understand the feedback to proceed correctly?	**YES.** The app confirms visually ("Route Saved") and verbally ("Route saved as [Name]"). The route immediately appears in the "Favorites" tab.

Task 9. Set Up Vibration-Based Departure Reminders

User	Deaf commuter.
Purpose	Get a tactile reminder before a scheduled bus/train.
Starting point	Notifications or Settings page.
What they do	Set a vibration pattern and advance time.
Context	Waiting indoors or multitasking before trip.
Steps	Open reminder settings → choose transport → pick alert time → bracelet vibrates before departure.

Important components of the interface

- Reminder Setup screen to create vibration alerts
- Transport selector enables searching by spoken input or vibration scrolling
- Time Offset picker allows users to adjust notifications such as 10 minutes before event
- Vibration Pattern Designer would allow users to associate messages with vibration patterns.
- Active Reminders List provides an overview of the existing reminders, and allows enabling/disabling or changing them

Data Types Used

- UserProfile for personalization
- TransportInfo to store routeNumber, stops, departure times, transport types, may sync with a public transport public API
- Reminder to trigger alarms, communicate with wearables
- VibrationPattern to associate different meaning to different vibrations
- DeviceInfo to allow vibration with multiple devices like smartwatches or bracelets.

User interactions

Deaf people

User Tap “Reminder” -> Choose route -> Pick talent time -> Choose vibration pattern

App output
Large text success screen.
Transparent-background UI so environment stays visible

Blind people

User Long-press screen ->
App TTS: “Add reminder?” -> Volume up = Yes, Volume-down = No.
User yes
APP TTS: “Which Route?”
User Speaks “Bus 31”
The app pulls the schedule and reads nearest departures.
The user swipes until hearing the correct stop.
User double taps.
App TTS: “When to alert? 5, 10, 15 minutes”
User presses volume up 2 times -> selecting 15 minutes
Haptic pattern for success.
Navigation:
Swipe up/down to move through history.
Double tap to select
Long-press to preview route in audio
Or Braille keyboard
Confirmation via vibration codes

Alignment to best practices

WCAG 2.2 accessibility supports Text scaling, contrast, captions and speech
Cross-modal redundancy enables Haptic, audio and text
Cognitive load reduction allows the usage of templates and simple gestures
Universal design enables the application usage with or without disability settings
Real-world constraints enables using the application under time pressure, in a noisy environment

Innovation

Transparent Live Background Text Overlay (For Deaf Users)

Allow users to see outside while reading UI to increase situational awareness. Works by running the camera behind the UI, having an adjustable background opacity slider, and displaying a large text overlay like HUD. Similar to Google Lens overlay text, or Samsung Translate assistant transparency mode.

Phone as Hearing-Aid-Like Caption Device

Use iOS Live Transcribe API (Speech framework) or Android Live Caption / Live Transcribe APIs to enable Real-time speech-to text to display a person’s voice like a subtitle.

Phase 6:

Question	**Answer & Motivation**
1. Will the user be trying to produce the effect?	**YES.** The user is multitasking or waiting indoors and relies on the tactile alert to leave on time.
2. Will the user see the correct control?	**YES.** The "Set Reminder" icon (bell/clock) is visible next to the departure time in the schedule view.
3. Will the user see that the control produces the desired effect?	**YES.** Tapping it opens the "Time Offset" picker (e.g., "Alert 10 min before"). The connection between the bell icon and an alarm is intuitive.
4. Is there another control that the user might select instead of the correct one?	**YES.** A user might confuse "Departure Time" with "Alert Time". They might set the alarm *at* the departure time (too late) instead of *before*. **Improvement:** Clearly label the option as "Remind me [X] min BEFORE departure".
5. Will the user understand the feedback to proceed correctly?	**YES.** The bracelet gives a sample vibration pattern immediately upon setting the alarm to confirm the haptic link is working.

Phase 8: Heuristic Evaluation Report

Cluster of Tasks Analyzed:

1. **Task 3:** Navigate in a Noisy or Crowded Environment (Get Off Alert).
2. **Task 6:** Report an Issue or Emergency Silently (Emergency SOS).
3. **Task 7:** Communicate with Rideshare/Taxi Drivers (Ride Assist).

1. Visibility of System Status

(The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.)

- **Compliance:** HearWay provides continuous, multi-sensory feedback (Visual, Haptic, and Audio) across all three tasks, ensuring the user is never left guessing the system's state.
- **Specific Examples:**
  - **Task 3 (Navigation):** The interface changes background colors dynamically (Black $\rightarrow$ Orange $\rightarrow$ Green) to indicate proximity. The **Progress Bar** gives immediate visual status, while the **Haptic Pulse** confirms state changes blindly.
  - **Task 6 (Emergency):** The "Selecting" state uses a **Red Thick Border** to highlight the active choice. Upon confirmation, the entire screen turns **Green** with a giant checkmark, instantly confirming that the distress signal was sent.
  - **Task 7 (Communication):** When a quick message is selected, a "Message Sent" overlay appears with a checkmark and AnimatedVisibility, confirming that the TTS engine has spoken the phrase.

2. Match Between System and the Real World

(The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.)

- **Compliance:** The system uses natural language and mimics physical interactions. The haptic feedback intensity correlates with real-world urgency.
- **Specific Examples:**
  - **Task 6 (Emergency):** The categories (Medical, Fire, Harassment) use plain language. The use of **Volume Buttons** as a physical trigger mimics the tactile nature of a physical panic button or alarm, which is more natural in a high-stress situation than a touchscreen.
  - **Task 7 (Communication):** The **Camera Overlay** ("Eyes on the World") integrates the digital UI directly into the physical reality, allowing deaf users to see the driver's lips/face while "speaking" through the app.
  - **Task 3 (Navigation):** The command "GET OFF NOW" matches the imperative internal monologue of a commuter arriving at their stop.

3. User Control and Freedom

(Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state.)

- **Compliance:** Every critical task has a "Fail-Safe" or "Emergency Exit" accessible via hardware or clear UI buttons.
- **Specific Examples:**
    - **Task 6 (Emergency):** The **10-second Timeout** acts as an automatic "Undo" for accidental pocket triggers. If no interaction is detected, the system cancels the SOS state ("Emergency cancelled due to timeout").
    - **Task 3 (Navigation):** The user can cancel an active journey instantly by pressing **Volume Down**, avoiding the need to hunt for a small "X" button while moving.
    - **Task 7 (Communication):** The user can exit the camera overlay mode immediately via the standard Back button or gesture, returning to the safe dashboard.

4. Consistency and Standards

(Users should not have to wonder whether different words, situations, or actions mean the same thing.)

- **Compliance:** The interaction model for Hardware Keys is strictly consistent across the entire application.
- **Specific Examples:**
    - **Universal Hardware Logic:** across Task 3, Task 6, and Task 7 (and Planning), the **Volume Down** key is consistently used to **Cycle/Navigate/Cancel**, while the **Volume Up** key is consistently used to **Select/Confirm**.
    - **Color Standards: Green** always signifies "Success/Safe" (Journey Complete, Emergency Sent, Message Spoken), while **Red/Orange** signifies "Attention/Action Required."

5. Error Prevention

(Even better than good error messages is a careful design which prevents a problem from occurring in the first place.)

- **Compliance:** The system anticipates high-stress errors (trembling hands, pocket dials, signal loss) and implements constraints to prevent them.
- **Specific Examples:**
    - **Task 6 (Emergency):** The **Timeout Mechanism** prevents false alarms. The system requires a distinct **Volume Up** confirmation action to send the alert; simply entering the screen is not enough.
    - **Task 3 (Navigation):** The **Graceful Degradation** logic handles GPS signal loss. Instead of freezing or crashing, the system proactively switches to a time-based algorithm to prevent the user from missing their stop.
    - **Task 7 (Communication):** Large, pre-set Quick Response buttons prevent typos that would likely occur if a user tried to type on a virtual keyboard while entering a taxi in a hurry.

6. Recognition Rather Than Recall

(Minimize the user's memory load by making objects, actions, and options visible.)

- **Compliance:** The interface relies on visible menus and distinct options rather than requiring the user to memorize commands.
- **Specific Examples:**
  - **Task 6 (Emergency):** The **LazyVerticalGrid** presents all emergency types at once with large icons and text. The user recognizes "Fire" or "Medical" immediately without recalling a specific number to dial.
  - **Task 7 (Communication):** Common phrases ("I am here", "Thanks") are presented as visible options, reducing the cognitive load of formulating a sentence during a social interaction.

7. Flexibility and Efficiency of Use

(Accelerators — unseen by the novice user — may often speed up the interaction for the expert user.)

- **Compliance:** The system is **Multimodal**, catering effectively to blind users (Audio/Haptic centric), deaf users (Visual centric), and expert users (Hardware centric).
- **Specific Examples:**
  - **Stealth Mode (Task 6):** An expert or blind user can report an emergency from their pocket using only the Volume Keys (Vol Down x2 -> Vol Up = Report Fire), without ever looking at the screen.
  - **Dual Output (Task 7):** The app speaks for the mute/deaf user (TTS) while showing captions for the hearing driver, bridging the communication gap efficiently.

8. Aesthetic and Minimalist Design

(Dialogues should not contain information which is irrelevant or rarely needed.)

- **Compliance:** High-stress screens are stripped of all decorative elements, focusing purely on the signal-to-noise ratio.
- **Specific Examples:**
  - **Task 3 (Navigation):** The "GET OFF NOW" screen removes the map and menus, displaying only the green background and massive text.
  - **Task 6 (Emergency):** The grid uses a High-Contrast (Black background, White text, Red borders) design with no distractions, ensuring the user focuses solely on selecting the help type.
  - **Task 7 (Communication):** The UI elements are semi-transparent overlays, maximizing the visibility of the real world through the camera.

9. Help Users Recognize, Diagnose, and Recover from Errors

(Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.)

- **Compliance:** Error handling provides constructive solutions rather than technical codes.
- **Specific Examples:**
    - **Task 3 (GPS Loss):** The system explicitly states "GPS Signal Lost" and immediately offers the solution: "Reverting to schedule-based alert."
    - **Task 6 (Timeout):** If the user hesitates too long, the system announces "Emergency cancelled," confirming the state change so the user knows no alert was sent (avoiding panic about false alarms).
    - **Task 7 (Unresponsive):** If the driver doesn't react, the user is prompted with "Driver Unresponsive?" and offered alternative actions (Show Screen, Flashlight).

10. Help and Documentation

(It may be necessary to provide help and documentation, preferably contextual.)

- **Compliance:** Documentation is **"Just-in-Time"** and audio-based, integrated directly into the task flow.
- **Specific Examples:**
    - **Task 6 (Emergency):** Upon entering the screen, the TTS immediately announces: "Use volume keys to select. Volume Up to confirm." This removes the need for a tutorial manual during a crisis.
    - **Task 7 (Communication):** First-time usage prompts explain that the camera is active to help communicate, ensuring the user understands the context.

Heuristic Report

Visibility of System Status

Finding 1. Additional feedback for blind users to identify their bus when multiple buses are available in the bus-station at the same time.

Severity: 3/4

Heuristic Violated: #1

Description: Blind users are notified immediately via TTS, but s may not perceive which bus is theirs, and the app should help with finding that out.

Recommendation: Introduce additional haptic and audio cues for blind users.

Finding 2: Countdown in `EmergencyScreen` shows time left for cancellation; TTS reads only once at start.

Severity: 2/4

Heuristic Violated: #1

Description: Blind users may not know remaining seconds unless reactivating TTS;

Recommendation: Consider periodic TTS countdown updates or haptic ticks as progress bar for clarity.

Match Between System and Real World

Finding 3: Bus arrival information is visual; blind users rely solely on TTS without integration with station announcements.

Severity: 4/4

Heuristic Violated: #2 Match between the System and the Real World

Description: The system does not integrate with real-world auditory cues at stations, causing potential confusion for blind users.

Recommendation: Integrate station PA systems or allow apps to pick up announcements via API if available; or provide haptic/voice alerts synced with arrivals.

Finding 4: Route and stop options sometimes described in technical terms (`RouteOption`, `StopInfo`) rather than plain language.

Severity: 2/4

Heuristic Violated: #2 Match between the System and the Real World

Recommendation: Use simple natural language for blind TTS messages and visual labels; avoid jargon.

User Control and Freedom

Finding 5: Users cannot cancel some actions (e.g., message sent in `RequestHelpScreen` or emergency once confirmed).

Severity: 3/4

Heuristic Violated: #3 User Control and Freedom

Description: Blind users have no “undo” option after sending a help message; emergency auto-send cannot be stopped after confirmation.

Recommendation: Add a short “cancel” window (e.g., 3–5 seconds) and/or allow users to revoke non-critical messages.

Consistency and Standards

Finding 6: Style inconsistency across screens (buttons vs. grid boxes, color coding).

Severity: 3/4

Heuristic Violated: #4 Consistency and Standards

Description: Some screens use rounded colored buttons for blind cues, others rectangular grids; inconsistent typography, padding, and spacing.

Recommendation: Establish a single style guide; standardize button shapes, color semantics, font sizes, and grid/list usage.

Error Prevention

Finding 7: Emergency messages sent instantly; potential for false triggers.

Severity: 4/4

Heuristic Violated: #5 Error Prevention

Recommendation: Introduce confirmation dialogs, vibration patterns, or volume key combination to reduce accidental activation.

Finding 8: GPS Signal Loss in Underground Areas

Severity: 3/4
Heuristic Violated: #5 Error Prevention

Description: Blind users relying on navigation may lose location accuracy in metro stations or tunnels due to GPS signal obstruction. Currently, the app may provide incorrect route instruction, causing confusion or delays. Sighted users are less affected because they can visually confirm routes, but blind users have no alternative cues.

Recommendations:
Use historical and statistical data to estimate expected arrival times for stations or stops.

If the user's estimated arrival time has passed and GPS is unavailable, provide calm, informative TTS or visual messages such as: "You should have arrived at Stop X by now. Please wait a moment; the vehicle may be delayed or GPS signal is unavailable".
Include guidance on next steps
Ensure messages are accessible for blind users via TTS.

Recognition Rather Than Recall

Finding 9: Communication templates currently exist in the navigation section instead of a dedicated communication section.

Severity: 3/4

Heuristic Violated: #6 Recognition rather than recall

Description: Users must remember what template is available or where to find it.

Recommendation: Move communication templates to a central "Communication" area; add labels and previews for recognition.

Flexibility and Efficiency of Use

Finding 10: Options like alternative routes and nearby stops may not provide new information; repeated in multiple sections.

Severity: 2/4

Heuristic Violated: #7 Flexibility and Efficiency of Use

Recommendation: Streamline options; combine related functionality to reduce cognitive load; offer shortcuts for experienced users (e.g., quick-select last-used route).

Aesthetic and Minimalist Design

Finding 11: Overuse of large icons, varied colors, and redundant confirmation screens can distract sighted users.

Severity: 2/4

Heuristic Violated: #8 Aesthetic and Minimalist Design

Recommendation: Simplify layouts; minimize repetitive screens (e.g., `RoutePlanning` vs. `FindStops` confirmations) and remove non-essential graphics for a cleaner interface.

Help Users Recognize, Diagnose, and Recover from Errors

Finding 12: No recovery options for failed camera permission in `Communicate` beyond returning to previous screen.

Severity: 3/4

Heuristic Violated: #9 Help Users Recognize, Diagnose, and Recover from Errors

Recommendation: Provide detailed guidance: “Enable camera in settings to use communication feature,” include retry button, and clarify error messages.

Help and Documentation

Finding 13: TTS messages are context-specific, but no help or tutorial is available for first-time users.

Severity: 2/4

Heuristic Violated: #10 Help and Documentation

Recommendation: Add a short onboarding tutorial for blind users explaining navigation, volume key actions, and emergency reporting.

Summary of Key Problems

Finding	Description	Severity	Recommendation
3	Blind users cannot correlate bus arrival TTS with real-world PA announcements	4	Integrate with station announcements or alert APIs
6	Style inconsistency (button shapes, colors, grids)	3	Adopt a consistent style guide
5	Emergency messages have no undo	3	Add short cancel window
9	Communication templates misplaced	3	Move to dedicated Communication section
10	Alternative routes/nearby stops redundant	2	Streamline options; remove redundancy

Overall Compliance: ~60%

Strengths: Strong accessibility support for blind users (TTS + haptics), feedback for actions, multiple navigation aids.

Weaknesses: Redundancy, inconsistent UI, partial integration with real-world data, limited error recovery, some overcomplicated flows.

Planned Improvements

- **Accessibility-first refinement**
  Blind accessibility API fully enabled; remove unnecessary visual elements when the user is blind.
- **UI Standardization**
  Unified button and grid styles, consistent typography, and color semantics.
- **Error Prevention**
  Emergency and help actions receive brief confirmation windows.
- **Information Placement**
  Move communication templates to the Communication section.
  Merge redundant navigation and stop selection options.
- **User Education**
  Add onboarding tutorial with TTS guidance.
- **Integration with Real-World Signals**

Explore station PA / alert integration for bus arrivals.

12. Conclusion & Future Roadmap

Our Heuristic Evaluation reveals a product with a **strong architectural core** but critical edge-case vulnerabilities.

On one hand, **HearWay succeeds in its primary mission**: breaking down accessibility barriers. The "Compliance" section demonstrates that our multimodal approach (integrating Hardware Keys, Haptics, and TTS) effectively creates an inclusive experience for blind and deaf users, satisfying the fundamental requirements of Visibility and Flexibility.

On the other hand, the "Critical Analysis" exposes that while the **happy path** works well, the **stress cases** need reinforcement. The violations identified (particularly regarding the passive SOS timeout (Finding 7) and GPS reliance in tunnels (Finding 8))highlight that adherence to safety standards (#5 Error Prevention) requires active, fail-safe mechanisms, not just passive alerts.

The Verdict: The prototype is roughly **60% production-ready**. The remaining 40% does not require a redesign of the core features, but rather a **strict refinement phase** focused on consistency (standardizing haptics) and active error prevention (hardware aborts for SOS). By addressing the "High Severity" violations identified in this report, HearWay will transition from a functional prototype to a reliable safety tool.