"Suno Saarthi": A Conversational Navigation Assistant to Enhance Driver Experience with AI in the Indian Context

PARTH BARTHWAL, IIIT-Delhi, India
MEHUL ARORA, IIIT-Delhi, India
YASH BHARDWAJ, IIIT-Delhi, India
NIKITA BHATIA, IIIT-Delhi, India

Fig. 1. Man driving using Google Maps, Pexels.com

This project proposes the development of "Saarthi," a conversational application that integrates with Google Maps to provide drivers with an intuitive, voice-based interface for route navigation. Current navigation solutions often lack interactive, bidirectional communication and struggle with mixed-language inputs common in India. By leveraging advanced AI models, including large language models (LLMs) fine-tuned for code-switching (e.g., Hindi-English), voice frameworks, and Google's Routes API, the system aims to enhance driver safety and convenience. It offers real-time, context-aware responses to driver queries, providing detailed route information related to flyovers, narrow roads, complex intersections, local landmarks, and potential hazards, all through natural language interaction. The application addresses the need for hands-free, detailed navigation assistance, particularly in challenging driving scenarios prevalent in India, such as high traffic density and rapidly changing road infrastructure. Saarthi aims to reduce driver anxiety, improve focus, and empower even tech-hesitant users to navigate independently and confidently by mimicking a knowledgeable co-passenger. The system's novelty lies in its seamless integration, ability to handle code-switching, proactive assistance, and potential for personalized navigation experiences.

ACM Reference Format:
Parth Barthwal, Mehul Arora, Yash Bhardwaj, and Nikita Bhatia. 2025. "Suno Saarthi": A Conversational Navigation Assistant to Enhance Driver Experience with AI in the Indian Context. In *HCAI Project Proposal, 2025.* ACM, New York, NY, USA, 20 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Problem Statement

Navigational technologies like Google Maps have revolutionised how individuals traverse urban and rural landscapes. However, in the Indian context, a significant demographic—particularly older adults and parents—often rely on a companion for verbal

Authors' Contact Information: Parth Barthwal, IIIT-Delhi, New Delhi, Delhi, India, parth21341@iiitd.ac.in; Mehul Arora, IIIT-Delhi, New Delhi, Delhi, India, mehul21066@iiitd.ac.in; Yash Bhardwaj, IIIT-Delhi, New Delhi, Delhi, India, yash22586@iiitd.ac.in; Nikita Bhatia, IIIT-Delhi, New Delhi, Delhi, India, nikita22323@iiitd.ac.in.

guidance. This reliance highlights a gap in existing navigation solutions, which predominantly offer unidirectional, text-based instructions without facilitating interactive, bidirectional communication. Our paper addresses this need by introducing a conversational navigation system tailored to the unique linguistic and cultural nuances of Indian users. By leveraging advanced LLMs, voice frameworks and Google's Routes API, the system aims to enhance driver safety and convenience by offering real-time, context-aware responses to queries about route details, such as flyovers, narrow roads, complex intersections, and landmarks in an Indian context.

2 Indian Context

This problem is especially prevalent in India due to a multitude of reasons. Some of them include:

(1) **Complex Road Infrastructure:** India's rapidly developing road network includes a mix of modern highways, flyovers, and complex intersections alongside narrow, congested urban streets. This diverse infrastructure can be challenging for drivers to navigate, especially in unfamiliar areas. The proposed system would be especially beneficial in providing detailed guidance for navigating flyovers and narrow roads, which are common features in Indian cities.
(2) **High Traffic Density:** Indian cities are known for their high traffic density and often chaotic driving conditions. In such environments, drivers need to make quick decisions while navigating through complex road layouts. A hands-free, voice-based navigation system would allow drivers to receive crucial information without diverting their attention from the road, enhancing safety in these challenging traffic scenarios.
(3) **Rapid Urbanization:** With India's ongoing rapid urbanization, many drivers frequently find themselves in unfamiliar areas of expanding cities. This increases the need for detailed, context-aware navigation assistance, especially for those not accustomed to navigating in urban environments.
(4) **Dependence on Co-Passengers for Navigation:** Many Indian drivers prefer human interaction for navigation due to comfort, language barriers, and the ability to provide contextual understanding. This reliance on co-passengers, while beneficial for shared responsibility and local knowledge, has limitations such as unavailability of knowledgeable companions and potential inaccuracies. The dependence on others for navigation can also hinder a driver's ability to navigate independently, especially in unfamiliar areas or complex traffic conditions.
(5) **Code-Switching and Mixed-Language Usage:** In India, it is common for users to switch between languages within a single conversation, a phenomenon known as code-switching. For instance, a user might alternate between Hindi and English while giving commands, such as "Give directions to the sabse paas waala hospital" ("Give directions to the nearest hospital"). Current voice recognition systems often struggle to accurately interpret these mixed-language inputs, resulting in errors and miscommunications that frustrate users and diminish the overall effectiveness of the navigation tool.

3 Literature Survey

This section reviews existing research and technologies relevant to the development of a conversational navigation assistant, focusing on Google Maps integration, conversational AI interfaces, and navigation assistance in complex driving environments.

3.1 Google Maps APIs and Navigation Systems

Google Maps provides robust navigation capabilities through its suite of APIs. The **Routes API** [6] is particularly useful for handling complex road structures, such as flyovers and narrow roads, ensuring accurate route guidance. Additionally, the **Directions API** [5] enables dynamic routing by offering features such as alternative routes, multimodal navigation, and real-time traffic insights. These APIs form the foundation for developing a real-time, conversational navigation assistant.

3.2 Conversational AI for Navigation

Conversational interfaces have gained traction in various domains, including navigation and information retrieval. Research on conversational AI for information search [9] suggests that natural language interfaces enhance usability by enabling users to retrieve relevant information quickly. Furthermore, studies on inclusive indoor navigation systems [13] highlight the advantages of voice-based guidance in complex environments, which is highly applicable to vehicle navigation. Google's advancements in AI-powered interactions [4] further strengthen the feasibility of implementing a natural language-based navigation assistant.

3.3 Navigation Assistance for Visually Impaired Users

Research on assistive technologies for visually impaired individuals [1] underscores the importance of multimodal interaction, integrating voice guidance with contextual awareness. While primarily designed for visually impaired users, these principles can be adapted to enhance driver experience, ensuring hands-free, voice-driven assistance for navigating challenging road conditions.

3.4 Emerging AI-Powered Navigation Technologies

Recent developments in AI-powered navigation tools, such as Google's conversational AI enhancements to Google Maps [11], demonstrate how intelligent voice assistants can improve the navigation experience. Additionally, research on human-computer interaction in navigation systems [8] highlights the importance of user-centric design in reducing cognitive load and improving trust in AI-powered navigation assistants.

3.5 Implications for Driver Experience

Integrating conversational AI with navigation enhances the driver experience by reducing cognitive overload and supporting real-time decision-making. Studies on dialog-based interaction for navigation [7] and natural language processing in route guidance [2] suggest that intelligent conversational agents can effectively assist users in interpreting complex navigation instructions, improving both efficiency and safety.

This literature survey underscores the potential of AI-driven conversational navigation assistants to revolutionize driver interactions with route guidance systems, making navigation more intuitive, accessible, and user-friendly.

4 User-Centered Design Approach

The design of Saarthi is grounded in a user-centered methodology, informed by direct observations and interactions with diverse drivers in India. This section details the point of view (POV) statement that emerged from these engagements, the personas that encapsulate user needs, and the "How Might We" (HMW) questions that guided the system's development. Together, these elements highlight the critical gaps in existing navigation tools and illustrate how Saarthi addresses them through a tailored, AI-powered conversational approach.

4.1 POV Statements

- We met and observed many people who drive - especially older adults and parents - who are frustrated with current navigation apps. We saw that these apps don't offer a conversational interface and struggle with mixed Hindi-English commands, often forcing them to depend on family or co-passengers for guidance.
- We were surprised to learn that despite the near-universal use of smartphones, drivers still feel overwhelmed by existing navigation systems. Challenges like flyovers, narrow streets, and unpredictable traffic amplify their need for simple, human-like verbal directions.
- We wonder if a voice-first, AI-powered conversational navigation assistant could bridge this gap. Could such a system deliver natural, intuitive guidance that adapts to a driver's language preferences and integrates culturally relevant cues?
- We feel the need to develop "Saarthi"—a seamless Google Maps integration that offers real-time, mixed-language voice guidance, proactive assistance, and context-aware alerts. We believe this solution will reduce driver anxiety, enhance safety, and empower even tech-hesitant users to navigate independently and confidently.

4.2 Personas and User Needs

Through qualitative research, five distinct personas were identified, each representing a segment of Saarthi's target audience with unique navigation challenges and requirements in the Indian context.

Rajesh Verma – The Daily Commuter. **Age:** 38 **Location:** Bangalore **Occupation:** IT Professional

Background: Rajesh commutes 15 km daily from Whitefield to Bellandur, navigating Bangalore's heavy traffic using both a two-wheeler and a car. While tech-savvy, he finds traditional navigation apps too rigid and distracting, especially when switching between modes of travel.

Motivations:

- Save time by avoiding traffic congestion.
- Get accurate, real-time updates to reroute efficiently.
- Use a hands-free interface to stay focused on the road.

User Goals:

- Identify the fastest route using live traffic insights.
- Receive voice-based guidance for complex road layouts.
- Stay informed with proactive alerts about traffic events.

Challenges:

- Handling unpredictable traffic patterns.
- Managing cognitive load when switching vehicles.
- Navigating visual-heavy interfaces while driving.

Sunita Patel – The Working Mother. **Age:** 35 **Location:** Mumbai **Occupation:** School Teacher

Background: Balancing her job and parental responsibilities, Sunita faces Mumbai's chaotic traffic daily. Standard navigation apps often frustrate her with cluttered interfaces that lack safety-focused features.

Motivations:

- Ensure a secure commute for her children.
- Avoid unexpected delays disrupting her schedule.
- Use a distraction-free hands-free navigation system.

User Goals:

- Discover safer routes optimized for school zones.
- Receive alerts about unexpected changes (e.g., diversions, pedestrians).
- Get parking assistance to streamline her commute.

Challenges:

- Managing Mumbai's unpredictable streets safely.
- Adjusting to last-minute route changes without distraction.
- Overcoming visually complex app interfaces.

Arvind Singh – The Highway Truck Driver. **Age:** 45 **Location:** Delhi-Jaipur Highway **Occupation:** Logistics Driver

Background: A seasoned truck driver, Arvind regularly drives long distances but struggles with standard navigation apps that ignore truck-specific requirements like low-clearance bridges and restricted routes.

Motivations:

- Find truck-friendly routes that comply with regulations.
- Get updates on road conditions in remote areas.
- Use voice-guided navigation to reduce manual interactions.

User Goals:

- Identify truck-permitted routes to avoid fines.
- Receive early warnings about low-clearance bridges and narrow lanes.
- Locate rest stops and fuel stations efficiently.

Challenges:

- Lack of navigation apps catering to large vehicles.
- Coping with inadequate road signage and diversions.
- Avoiding frequent screen interactions during long hauls.

Dinesh Yadav – The Senior Citizen. **Age:** 68 **Location:** Lucknow **Occupation:** Retired Army Officer

Background: Dinesh, an experienced driver, now finds modern navigation interfaces overwhelming due to his slower reaction time and reduced eyesight. He mainly drives short, familiar distances but struggles with fast-paced instructions and text-heavy designs.

Motivations:

- Receive slow-paced, step-by-step navigation instructions.
- Avoid distraction-heavy, cluttered UI elements.
- Get proactive hazard alerts to allow sufficient reaction time.

User Goals:

- Follow landmark-based, easy-to-understand directions.
- Receive early warnings about road hazards.
- Use a navigation system designed for clarity rather than speed.

Challenges:

- Struggling with small text and rapid updates.
- Overloaded interfaces that don't accommodate slower reaction times.
- Anxiety from sudden, complex instructions.

Sneha Joshi – The Rider. **Age:** 35 **Location:** Pune **Occupation:** Delivery Executive

Background: Sneha relies on her scooty for quick deliveries across Pune, facing challenges with navigation apps that lack real-time awareness of two-wheeler-friendly routes and road hazards.

Motivations:

- Optimize routes to meet tight delivery schedules.
- Avoid hazardous road conditions like potholes and sudden congestion.
- Use voice-based navigation to maintain hands-free control.

User Goals:

- Get real-time lane-specific guidance suitable for two-wheelers.
- Receive hazard alerts for poor road conditions.
- Identify quick, two-wheeler-friendly detours through urban congestion.

Challenges:

- Navigating unpredictable urban roads efficiently.
- Lacking two-wheeler-specific route options in standard navigation apps.
- Balancing quick route adjustments with hands-free safety.

4.3 Design Challenges and Solutions

The insights from user observations and personas were distilled into five "How Might We" (HMW) questions, each addressing a critical design challenge and guiding Saarthi's feature development:

- **HMW 1: How might we design a voice-first navigation system that minimizes distractions while delivering real-time, context-aware guidance?** Saarthi incorporates a hands-free interface with conversational voice prompts, integrating real-time traffic updates and hazard alerts (e.g., accidents, diversions) to support drivers across vehicle types, ensuring focus remains on the road.
- **HMW 2: How might we create a navigation experience that adapts to users with varying tech proficiency and cognitive needs?** The system employs an adaptive interaction model that learns from prior user interactions, offering slow-paced, landmark-based directions for users like Dinesh and streamlined commands for busy individuals like Sneha, minimizing cognitive load.
- **HMW 3: How might we incorporate cultural and linguistic nuances into our system to resonate with a diverse user base?** Saarthi embeds mixed-language support (e.g., Hindi-English) and culturally relevant cues into its AI model, enabling natural communication that aligns with users' everyday language patterns and local contexts.

- **HMW 4: How might we build a navigation system that adapts dynamically to different vehicle types and road challenges?** Vehicle-specific routing and context-aware alerts are implemented, providing lane guidance for two-wheelers, truck-friendly paths for drivers like Arvind, and hazard notifications for urban complexities, ensuring tailored, timely assistance.
- **HMW 5: How might we design a navigation solution that actively reduces driver anxiety and enhances road safety?** Saarthi delivers proactive, clear voice instructions and real-time alerts about potential hazards (e.g., flyovers, pedestrian zones), empowering informed decision-making and reducing stress across diverse driving scenarios.

5 Methodology

5.1 Co-Design Approach

To address the pain points and insights we gathered from our research with personas and HMW questions, we decided to co-design. Co-design is a user-centered design methodology that actively involves stakeholders in the development process, ensuring that the final product aligns with user needs and expectations [**?**]. By collaborating with potential users, co-design allows for iterative refinement, integrating feedback at various stages.

5.1.1 Feature Categorization. Below is a detailed categorization of the features based on their functionality and relevance:

Category	**Feature ID**	**Feature Description**
Activation & Entry Point	1	Auto-Activation – NavBuddy starts automatically when a destination is set in Google Maps.
	2	Manual Activation – Press a button to enable NavBuddy's voice assistance manually.
	3	Activation Indicator – A small icon or message confirms NavBuddy is active.
	4	Passive Listening Mode – The system remains in standby, waiting for voice input.
	5	Wake Word Activation – User can say "Hey NavBuddy" to activate hands-free.
	6	Seamless Google Maps Integration – Works within Google Maps without switching apps.
	7	Driving Mode Toggle – Users can manually activate or deactivate NavBuddy's voice assistance.
	8	Battery & Data Efficiency Mode – Minimizes GPS usage when navigation is paused or in standby.
Voice-Based Navigation Controls	9	Announce Route Summary – NavBuddy gives an initial route summary before navigation begins.
	10	Periodic Voice Instructions – System provides turn-by-turn guidance at key points.
	11	Landmark-Based Directions – Uses familiar landmarks instead of just street names.
	12	Mixed-Language Support – Allows users to switch between Hindi-English or other languages seamlessly.
	13	Adjust Navigation Style – Users can set preference for fast instructions, detailed instructions, or slow-paced guidance.
	14	Context-Aware Guidance – Adjusts instructions based on traffic conditions and driving speed.
	15	Short & Clear Instructions – Avoids long, complex sentences to reduce cognitive load.
	16	Intersection & Lane Guidance – Clear instructions for multi-lane highways and confusing junctions.
	17	Alternative Route Suggestions – Recommends faster or safer routes proactively.
User Interaction & Clarifications	18	Listening Window Activation – A pulsing microphone icon appears after each instruction, indicating NavBuddy is ready for clarifications.

Category	Feature ID	Feature Description
	19	User Clarification Queries – "Should I take the flyover?" / "Repeat last instruction" / "Show me an alternate route."
	20	System Echo & Confirmation – NavBuddy repeats the user's query before adjusting navigation.
	21	Smart Interruptions – If a user speaks while NavBuddy is giving instructions, it pauses and listens.
	22	Revised Directions – System dynamically adjusts based on user queries.
	23	Gesture-Based Controls – Nod to confirm, shake head to reject alternative routes.
	24	Predictive Assistance – System detects hesitation or deviation and prompts: "Need a different route?"
	25	Adaptive Listening Duration – Adjusts the time it listens for queries based on past user behavior.
	26	Mute & Unmute via Voice – "Mute NavBuddy" / "Resume voice guidance."
	27	Bookmark a Location – "Save this gas station" / "Mark this as a preferred route."
Visual Interface & Feedback Elements	28	Minimalist UI Overlay – A semi-transparent instruction box appears at the bottom of the screen.
	29	Route Highlights – Key segments like toll booths, exits, or congestion points are visually marked.
	30	Distraction-Free Design – Large fonts, high contrast, and minimal clutter.
	31	Listening Indicator – Pulsing icon or animation signals when NavBuddy is actively listening.
	32	Silent Mode Option – User can disable voice but keep visual cues.
	33	Dark Mode & High-Contrast Theme – Adapts to different lighting conditions.
	34	Customizable Text Size – Allows users to adjust text for readability.
	35	Night Mode Adaptation – Reduces screen brightness and glare automatically.
	36	Vibration Feedback – A subtle vibration confirms when a command is recognized.
Advanced Features & Adaptability	37	Vehicle Mode Adaptation – Different guidance styles for cars, two-wheelers, or trucks.
	38	Proactive Traffic Alerts – Warns about roadblocks, accidents, or diversions.
	39	Hands-Free Route Adjustments – "Hey NavBuddy, avoid highways" or "Find a gas station."
	40	Destination-Based Mode – Adjusts voice guidance based on urban vs. highway driving.
	41	Emergency Assistance – Quick voice-activated SOS or police contact.
	42	Fuel & Charging Station Alerts – Suggests stops based on fuel level (if connected to the car system).
	43	Adaptive Driving Mode – Recognizes user's preferred route style and adjusts recommendations.
	44	Speed Limit Reminders – Voice alerts when exceeding speed limits in specific zones.
	45	School & Pedestrian Zone Warnings – Alerts drivers to high-risk areas.
	46	Crowdsourced Traffic Intelligence – Uses driver-reported data to warn about hazards.

5.1.2 Data to Extract from Card Sorting. The following data points were extracted from the card sorting exercise to inform the design and prioritization of features:

(1) **Category Agreement Analysis** (How consistently users categorize each feature)

- **What to look for:**
  - Identify high-agreement features (cards most users placed in the same category).
  - Identify low-agreement features (cards that were placed in multiple categories).
  - If users consistently misplace a feature, reconsider its placement or label clarity.
- **Example:**
  - If "Wake Word Activation" appears in both Activation & Entry Point and User Interaction, it suggests overlap. Consider merging or rewording.

(2) **Key Feature Prioritization** (Which features users find essential or secondary)
- **What to look for:**
  - Count how many users prioritized each feature when sorting.
  - Identify most frequently selected features as critical for early design phases.
  - Features with lower priority might be optional or secondary.
- **Example:**
  - If "Landmark-Based Directions" is consistently grouped under Voice-Based Navigation Controls, it's a core function.
  - If "Gesture-Based Controls" appears sporadically, it may be a nice-to-have feature for later.

(3) **Naming & Labeling Preferences** (How users interpret terms and groupings)
- **What to look for:**
  - Do users rename categories or suggest alternative labels?
  - Are certain terms confusing or misunderstood?
  - Consider rewording UI labels based on common preferences.
- **Example:**
  - Users may rename "Listening Window Activation" to "Clarification Mode", indicating better terminology.

(4) **Feature Overlap & Consolidation** (Where features are too similar or redundant)
- **What to look for:**
  - If users place two similar features in different categories, consider merging them.
  - Identify features that might be unnecessary duplications.
- **Example:**
  - If "User Clarification Queries" and "Revised Directions" always appear together, they might be merged under a single "Dynamic Route Adjustments" feature.

(5) **User Segmentation Insights** (How different personas categorize features differently)
- **What to look for:**
  - Compare sorting patterns across different user types (e.g., tech-savvy vs. elderly drivers).
  - Identify persona-based UI needs.
- **Example:**
  - Older users might sort "Large Text Display" under Essential Navigation rather than Visual Interface, suggesting accessibility needs.
  - Truck drivers might prioritize "Truck-Friendly Route Suggestions", while urban commuters focus on "Proactive Traffic Alerts".

(6) **Unexpected User Behavior & Insights** (What surprised you?)
- **What to look for:**
  - Did users group a feature in an unexpected category?
  - Are users suggesting new features based on how they categorized items?
- **Example:**
  - Users might place "Silent Mode Option" under User Interaction rather than Visual Interface, indicating they see it as a control, not just a display setting.

5.1.3 Card Sorting Results. Tables 2 to 6 show the feature categorization by each participant.
Further analysis of feature agreement, prioritization, and labeling insights follows in the next section.

Table 2. Card Sorting Results for Rajesh

Category	Features (IDs)
Activation & Entry Point	1, 6, 3, 5, 4, 8, 7, 2
Voice-Based Navigation Controls	14, 17, 16, 9, 10, 13, 12, 15, 11
User Interaction & Clarifications	19, 24, 22, 21, 18, 25, 20, 26, 27, 23
Visual Interface & Feedback Elements	30, 28, 31, 35, 32, 33, 34, 29, 36
Advanced Features & Adaptability	38, 39, 46, 37, 40, 45, 42, 44, 41, 43

Table 3. Card Sorting Results for Sunita

Category	Features (IDs)
Activation & Entry Point	1, 6, 3, 5, 4, 7, 2, 8
Voice-Based Navigation Controls	10, 16, 11, 14, 12, 15, 13, 17, 9
User Interaction & Clarifications	19, 22, 24, 18, 21, 25, 20, 26, 23, 27
Visual Interface & Feedback Elements	30, 28, 31, 35, 32, 33, 34, 29, 36
Advanced Features & Adaptability	45, 38, 46, 40, 39, 37, 42, 44, 41, 43

Table 4. Card Sorting Results for Arvind

Category	Features (IDs)
Activation & Entry Point	1, 6, 3, 5, 4, 7, 8, 2
Voice-Based Navigation Controls	10, 16, 14, 9, 11, 17, 12, 13, 15
User Interaction & Clarifications	19, 22, 24, 18, 21, 20, 25, 26, 27, 23
Visual Interface & Feedback Elements	30, 28, 31, 29, 35, 32, 33, 34, 36
Advanced Features & Adaptability	37, 38, 40, 42, 39, 44, 45, 41, 46, 43

Table 5. Card Sorting Results for Dinesh

Category	Features (IDs)
Activation & Entry Point	2, 3, 6, 5, 4, 7, 1, 8
Voice-Based Navigation Controls	9, 15, 11, 13, 12, 14, 16, 10, 17
User Interaction & Clarifications	18, 21, 20, 25, 19, 22, 24, 26, 27, 23
Visual Interface & Feedback Elements	30, 34, 28, 33, 32, 35, 31, 29, 36
Advanced Features & Adaptability	40, 37, 42, 38, 41, 39, 44, 45, 46, 43

Table 6. Card Sorting Results for Sneha

Category	Features (IDs)
Activation & Entry Point	5, 6, 2, 7, 3, 1, 8, 4
Voice-Based Navigation Controls	15, 10, 12, 11, 9, 16, 14, 13, 17
User Interaction & Clarifications	19, 21, 18, 22, 26, 24, 20, 27, 25, 23
Visual Interface & Feedback Elements	30, 28, 33, 32, 34, 31, 29, 35, 36
Advanced Features & Adaptability	38, 42, 40, 37, 39, 41, 44, 45, 46, 43

5.1.4 *Results.*

- **Category 1: Activation & Entry Point**
  - **Strong Agreement (Most users placed these together)**
    - 1 (Auto-Activation), 6 (Announce Route Summary), 3 (Activation Indicator), 5 (Wake Word Activation), 4 (Passive Listening Mode), 7 (Driving Mode Toggle)

    Everyone placed these in Activation & Entry Point, meaning they should be key features in the UI design.
  - **Disagreements / Outliers**
    - 2 (Manual Activation) – Dinesh ranked this first, while others placed it later in their list. This suggests some users prefer manual control, while most rely on auto-activation.

    * 8 (Battery & Data Efficiency Mode) – Appears inconsistently ranked, meaning it might be a secondary setting rather than a core function.

- **Category 2: Voice-Based Navigation Controls**
  - **Strong Agreement**
    * 10 (Periodic Voice Instructions), 16 (Intersection & Lane Guidance), 14 (Context-Aware Guidance), 9 (Mixed-Language Support), 11 (Landmark-Based Directions)

    These are crucial voice-based features that should be prioritized in the NavBuddy voice experience.
  - **Disagreements / Outliers**
    * 17 (Alternative Route Suggestions) – Ranked higher by Rajesh and Arvind but lower by Sunita and Sneha. This suggests that tech-savvy users value dynamic routing more, while casual users may not frequently use rerouting.
    * 15 (Adjust Navigation Style) – Highly ranked by Dinesh and Sneha, but lower in others' lists. This might be critical for older users or those who prefer slower-paced guidance.
- **Category 3: User Interaction & Clarifications**
  - **Strong Agreement**
    * 19 (Listening Window Activation), 22 (Smart Interruptions), 24 (Predictive Assistance), 21 (User Clarification Queries)

    These core features should be designed to be clear, responsive, and easy to use.
  - **Disagreements / Outliers**
    * 23 (Bookmark a Location) – Low-ranked across all users, suggesting it may not be an essential feature for voice interaction.
    * 25 (Adaptive Listening Duration) – Higher for Rajesh & Arvind, lower for Sunita & Sneha, meaning tech-oriented users might expect smarter listening adjustments.
- **Category 4: Visual Interface & Feedback Elements**
  - **Strong Agreement**
    * 30 (Minimalist UI Overlay), 28 (Route Highlights), 31 (Listening Indicator), 35 (Night Mode Adaptation)

    These should be core UI design elements, ensuring clarity and reduced distraction.
  - **Disagreements / Outliers**
    * 34 (Customizable Text Size) – Higher for Dinesh, suggesting older users prioritize accessibility options more than younger drivers.
    * 32 (Silent Mode Option) – Mixed rankings, meaning it should be an optional setting rather than a primary feature.
- **Category 5: Advanced Features & Adaptability**
  - **Strong Agreement**
    * 38 (Proactive Traffic Alerts), 40 (Destination-Based Mode), 37 (Vehicle Mode Adaptation), 42 (Fuel & Charging Station Alerts)

    These features were consistently ranked high, confirming they should be key enhancements to NavBuddy.
  - **Disagreements / Outliers**
    * 46 (Crowdsourced Traffic Intelligence) – Ranked highly by Arvind & Sneha but lower for others, meaning frequent highway drivers find this more useful than urban commuters.
    * 44 (Speed Limit Reminders) – Lower-ranked, possibly indicating users don't expect this from NavBuddy, as their car's dashboard already shows speed limits.

5.2 User Taskflows

We have designed Saarthi's User Task Flows with a mindset to enable a seamless, bidirectional conversational navigation experience, minimizing driver distraction while providing clear, context-aware guidance. The flow captures the entire navigation session, from when the destination is entered into Google Maps to when the journey ends, by clearly defining each user and system interaction in atomic steps.

The process begins as soon as the user inputs their destination in Google Maps. Saarthi automatically activates its continuous navigation mode and immediately announces the route along with initial instructions, informing the user that they can ask for clarifications at any time. The system then delivers periodic voice instructions and overlays visual guidance on the map. After each instruction, a short listening window is initiated to allow the user to ask for clarification if necessary.

We have also made the task flow adaptive to the technology the user has access to: In a CarPlay/Android Car environment, the user may press an activation button on the steering wheel to trigger the listening window; on mobile devices, the user speaks a wake word. If a clarification query is received, the system captures and echoes the input, processes the query, updates the instructions accordingly, and resumes continuous guidance. Otherwise, if no query is detected and the destination has not been reached, the system temporarily enters dormant mode before reactivating the listening window.

Figure 2 presents the task flow diagram.

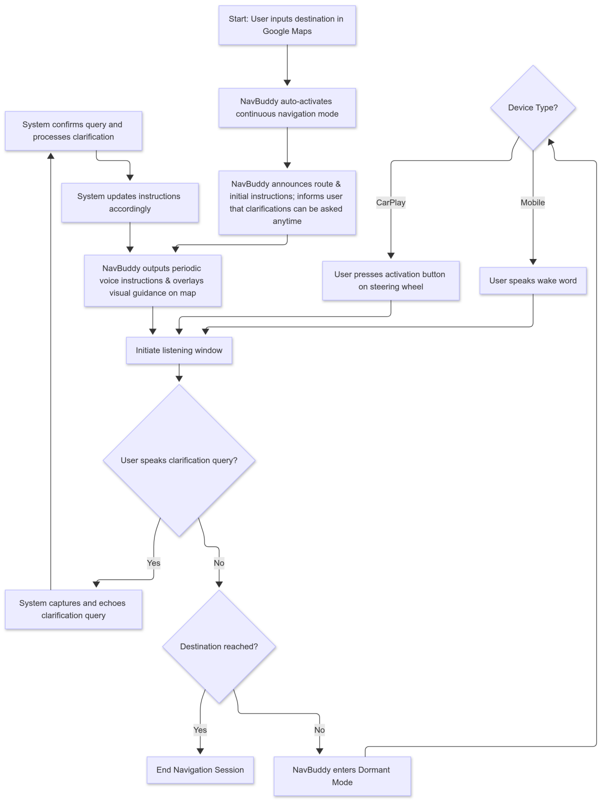

Fig. 2. Detailed User Task Flow for Saarthi

5.3 Low-Fidelity Prototype

The proposed conversational AI navigation assistant introduces several innovative features that set it apart from traditional navigation systems, addressing both emotional and technical aspects of the user experience. A Sketch for the same is provided at: figma

Figure 3 presents the Lo-Fi prototype.

5.4 Screen 1: Activation & Entry Point

Layout & Elements:

- **Google Maps Background** – Standard map interface
- **Saarthi Activation Indicator** – Small floating icon (top right corner)
- **Wake Word Prompt** – Subtle text: “Say ’Suno Saarthi’ for assistance”
- **Soft Chime Feedback** – Audible confirmation upon activation

User Interaction Flow:

- User sets a destination in Google Maps
- Saarthi auto-activates, overlay appears with activation chime
- Minimal UI overlay with Saarthi logo & “Listening Mode” text
- If first-time user, a small tooltip appears: “Saarthi chalu hai. Aap padhav puchh sakte hain.”

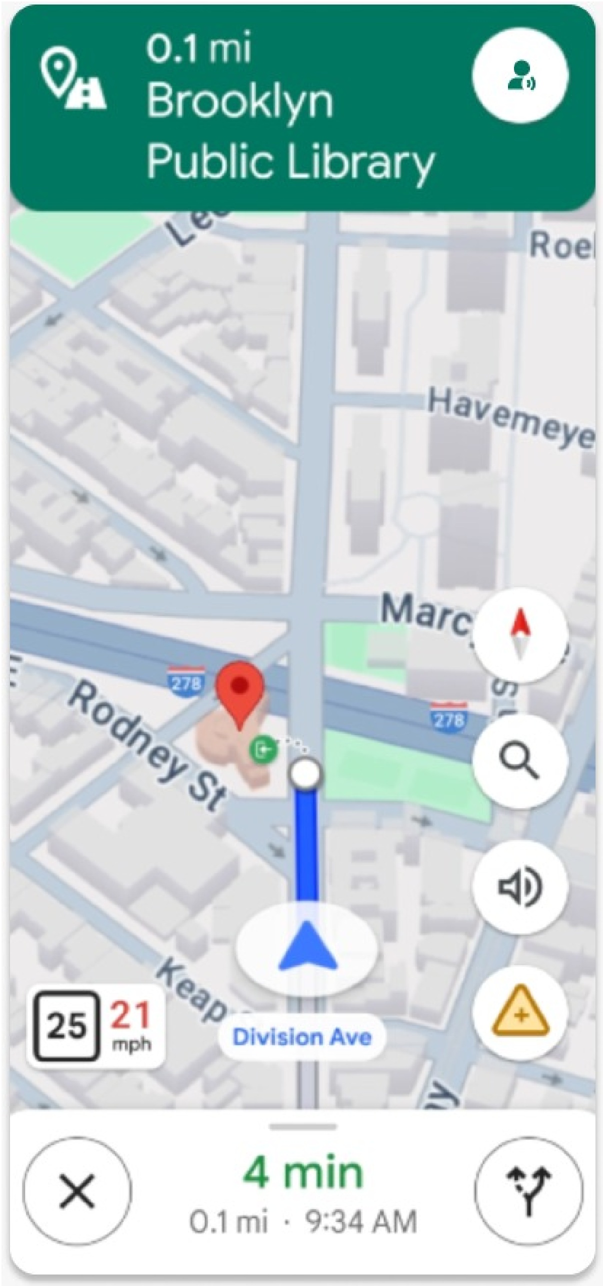

Fig. 3. Lo-Fi Design

5.5 Screen 2: Assisted Driving Mode (Main Navigation)

Layout & Elements:

- extbfFull-Screen Google Maps View – Main navigation screen
- extbfInstruction Overlay (Bottom of Screen) – Displays real-time navigation prompts
- extbfVoice Listening Indicator – Small pulsing mic icon after each instruction
- extbfNon-Intrusive Notification Zone (Top Left) – Alerts for roadblocks, speed warnings, congestion
- extbfAlternate Route Suggestion Panel – Appears dynamically when traffic is detected

User Interaction Flow:

- Saarthi provides turn-by-turn guidance (e.g., "Aage 500 meter mein flyover ka right le lo")
- Before major turns, soft chime plays instead of full voice command
- If traffic is ahead, an on-screen alert blinks + soft chime: "Traffic ahead. Alternate route chahiye?"
- Listening Mode opens briefly after instructions (pulsing mic icon for 5 seconds)

- User may ask clarifications (e.g., "Yeh flyover lena zaroori hai?")
- Saarthi responds if needed or auto-closes Listening Mode if no input

5.6 Screen 3: Exit & Dormant Mode

Layout & Elements:

- **Arrival Confirmation Overlay** – Soft chime + "Aap manzil par pohonch gaye hain"
- **UI Auto-Fades Out** – After arrival confirmation
- **Standby Mode Indicator (Small Icon, Bottom Right)** – If inactive for 5 mins
- **Wake-Up Prompt** – "Suno Saarthi kehkar wapas bula sakte hain"

User Interaction Flow:

- User reaches destination, soft chime confirms arrival
- Saarthi UI disappears after a few seconds
- If no user interaction, Saarthi enters Dormant Mode (standby indicator remains visible)
- If needed again, user can wake Saarthi via "Suno Saarthi" or tap UI icon

5.7 Additional Notes

- **Minimalist Design** – UI remains unobtrusive and prioritizes voice-first interaction
- **Smart Audio Feedback** – Chimes replace unnecessary voice prompts to reduce distraction
- **Hands-Free Optimization** – Designed to minimize user input while driving

5.8 Next Steps

- Implement these wireframe ideas into Figma or Adobe XD for visual representation
- Create interactive prototype to test UI behavior & user flow
- Gather user feedback to refine Listening Mode & alerts

5.9 Key Features to Prioritize in Design

- Auto-Activation & Voice-Based Entry (1, 5, 3, 6, 4)
- Clear, Concise Voice Instructions with Mixed-Language Support (10, 9, 11, 16)
- Listening Window & Clarifications (19, 22, 24, 21)
- Minimalist UI with Visual Cues (30, 28, 31, 35)
- Proactive Traffic & Vehicle Mode Adaptability (38, 40, 37, 42)

5.10 Features That Should Be Optional or Settings-Based

- **Battery & Data Efficiency Mode (8)** – Keep as an advanced setting.
- **Adjust Navigation Style (15)** – Allow customization, but default to standard mode.
- **Silent Mode Option (32)** – Some users will want full voice interaction, others may prefer only visual cues.
- **Speed Limit Reminders (44)** – Useful but secondary; cars already show speed limits.

6 Mid-Fidelity Prototype

6.1 Design Pivot: From Map Add-on to Standalone App

Our decision to pivot from a Google Maps add-on to a standalone app stems from our commitment to enhancing user experience through a conversational interface that operates independently of traditional map views. This new direction allows us to focus on delivering essential navigation information in a clear and engaging format.

This pivot is driven by the understanding that many users seek a navigation solution that prioritizes conversation and clarity over traditional mapping. By creating an app that stands alone, we can offer a more engaging and efficient experience tailored to modern needs, while still providing access to comprehensive mapping resources when desired. This approach not only differentiates our product but also aligns with our vision of making navigation accessible and user-friendly without being overly reliant on visual maps.

Fig. 4. Mid-Fidelity UI Mockup 1: Navigation assistant with clear direction, distance, and voice interaction

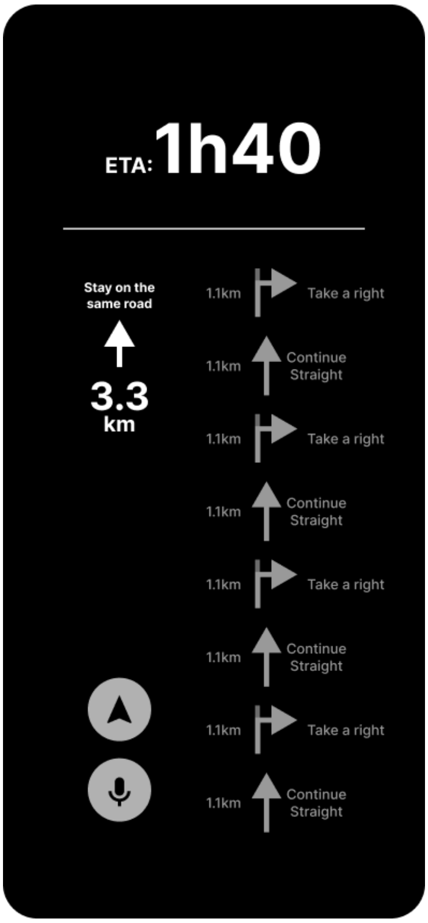

Fig. 5. Mid-Fidelity UI Mockup 2: Draggable interface with sequential route steps

6.2 Mid-Fi Screen Overview

6.3 Key Features Highlighted

- **Bold ETA Display:** Users see a prominently displayed Estimated Time of Arrival (ETA) at the top, ensuring they are always aware of their remaining travel time.
- **Visual Distance and Direction:** Clear, minimal iconography is used to show directions (e.g., continue straight, turn right) alongside a distance value. This simplifies visual parsing and reduces distraction.
- **Voice Activation Options:**
  - A dedicated mic button is available on the screen for manual activation.
  - Users may also activate the assistant hands-free by saying the wake word "Suno Saarthi" or by tapping anywhere on the screen.
- **Reflective User Queries:** The question asked by the user is visually reflected on-screen to confirm that the assistant correctly interpreted the voice input, providing instant feedback and clarity.
- **Future Navigation Steps:** Upcoming directions are shown (e.g., turn right in 1.1 km), giving users the ability to anticipate route changes without needing to view a full map.
- **Draggable Route Timeline (as shown in Figure 5):** The route preview is designed to be draggable vertically, allowing users to explore future instructions with ease.
- **Bespoke Iconography:** The app uses custom-designed directional icons tailored for driving contexts in India, ensuring both familiarity and clarity.
- **Seamless Map Integration:** While the interface is designed to function independently of traditional maps, an option is available to switch to Google Maps if the user prefers a detailed map view.

7 Baseline Implementation and Functionalities

7.1 Overview

The baseline implementation of *Suno Saarthi* showcases a locally-running, privacy-respecting conversational driving assistant. This version focuses on demonstrating end-to-end interaction, starting from wake-word-based audio activation to a context-aware, AI-generated response. Key elements include:

- Wake-word detection via real-time microphone input
- Integration of a lightweight open-source language model (`TinyLlama/TinyLlama-1.1B-Chat-v1.0`)
- Modular system design for future integrations
- Basic response logic for navigation-related queries

7.2 Key Functionalities Implemented

Feature	Description
Wake Word Detection	The assistant continuously listens in standby mode and activates when the user says "Suno Saarthi" or anything along those lines. Implemented using Python's `SpeechRecognition` with `PyAudio`.
Conversational AI	Once activated, user queries are sent to a locally-running instance of `TinyLlama/TinyLlama-1.1B-Chat-v1.0` via Hugging Face `transformers`. The model generates human-like responses to driving-related or casual questions. A small model has been used only to check the working of the system.
Navigation Response Stub	Queries about routes (e.g., "flyover ahead" or "shortcut to Noida") are handled through a rule-based stub. This will later be replaced with Google Maps API integration.
Modular Architecture	The system is split into modules: `wake_word.py`, `llm_interface.py`, `navigation.py`, and `main.py`. This ensures ease of testing and extensibility.

Table 7. Implemented Baseline Features

7.3 Tech Stack

- **Voice Input:** Python, `PyAudio`, `SpeechRecognition`
- **Language Model:** `TinyLlama/TinyLlama-1.1B-Chat-v1.0` via Hugging Face `transformers`

- **Routing Logic:** Stubbed Python module, ready to be integrated with real APIs
- **Execution Environment:** Python 3.10+ on Ubuntu 22.04 (tested locally)

7.4 Terminal Interaction Snapshot

```
[STANDBY] Listening for wake word...
[AWAKE] 'Suno Saarthi' heard. How can I help?

[ACTIVE] Awaiting your command for ~5 seconds...
User: Can I take the flyover?
Saarthi (Route Info): There is a flyover 2 km ahead. Stay in the right lane.
```

7.5 Design Justifications

Decision	Rationale
TinyLlama LLM	Lightweight, locally runnable model suitable for CPU-level inference. Ideal for real-time voice applications in constrained environments. Mainly for initial phase.
Microphone-based Wake Word	Enables hands-free use in vehicles; keeps driver focus on the road. Low-latency activation using known libraries.
Stubbed Navigation	Simplifies early testing and showcases how AI responses can integrate with route queries. Modular design enables API swap later.
CLI-first Baseline	Terminal interface allows for debugging, fast iterations, and easy data collection for future fine-tuning or UI development.

Table 8. Design Rationale for Baseline

7.6 Sample Queries and Responses

User Query	System Response
"Suno Saarthi"	Wake-word detected. System activates.
"Can I take the flyover?"	"There is a flyover 2 km ahead. Stay in the right lane."
"Tell me a joke"	LLM-generated road-friendly jokes (e.g., "Why did the bike fall over? It was two-tired!")
"Shortcut to Noida?"	"There's a shortcut via MG Road, but it might have traffic."
"Quit"	System exits gracefully and returns to terminal.

Table 9. Example Interactions with Saarthi

7.7 Known Limitations and Next Steps

- **No Text-to-Speech:** System currently replies via text in terminal. Next iteration will include voice feedback using TTS engines.
- **No Continuous Listening:** Voice input is time-limited to 5-second intervals. A streaming or threaded audio listener is planned.
- **Navigation Hardcoded:** Static route responses to be replaced with real-time data from Google Maps Directions API.
- **No GUI Yet:** CLI interface is for development/testing. Figma-based UI design is pretty much done and integration with backend is planned to be done soon.

8 Preliminary Results

The baseline implementation of *Suno Saarthi* demonstrates a successful end-to-end conversational navigation flow in a local development setting. The system is able to:

- Detect a wake word ("Suno Saarthi") through real-time audio input.
- Accept user queries via voice and transcribe them accurately.
- Generate appropriate responses using a lightweight, locally-hosted LLM (`TinyLlama/TinyLlama-1.1B-Chat-v1.0`).

- Provide route-related stub responses (e.g., flyover, shortcuts) in a conversational format.

This marks a significant milestone in proving the feasibility of a voice-first standalone navigation interface that can operate independently of traditional visual-heavy apps (Google Maps). This helps improve road safety, and reduces cognitive load so that user can spend more time focusing on the road.

System Successes

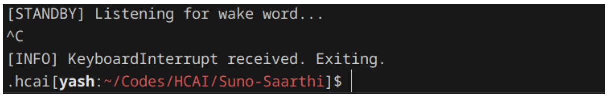

Fig. 6. The system can successfully listen for "Suno Saarthi".

- **Conversational Flow:** Basic queries like “Can I take the flyover?” or “How far is the turn?” are processed smoothly with relevant and natural replies.
- **Wake Word Responsiveness:** The assistant reliably detects and responds to the “Suno Saarthi” trigger with minimal delay.
- **Modular Integration:** The system architecture—comprising separate modules for audio input, LLM integration, and routing—supports scalability and is ready for seamless API and GUI integration.

Challenges Encountered

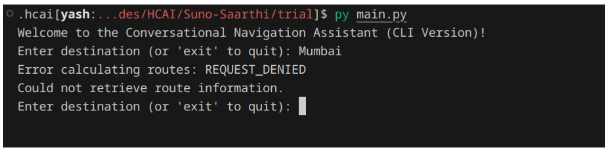

Fig. 7. There is an issue with Directions API due to recent integration of it into Routes API.

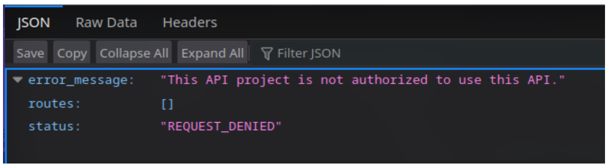

Fig. 8. The API fails to fetch information which points to an issue with GCP.

- **Google Directions API Limitation:** During early integration attempts, the Directions API faced issues related to billing configuration, unstable quota limits, and occasional regional access failures. This limited our ability to dynamically fetch turn-by-turn instructions in real-time.
- **High RAM Usage in GCP-Firefox Environment:** When running the prototype on Google Cloud Platform (GCP) via Firefox, we observed excessive memory usage exceeding 8+ GB. This was caused by simultaneous browser-based model inference and microphone streaming, leading to browser instability and crashes.

Planned Fixes

- **API Reliability Improvements:** Shift the Directions API integration to a backend service using Flask or FastAPI, where API requests can be handled, parsed, and preprocessed server-side for more consistent response delivery.
- **Browser Optimization:** Migrate prototype testing to Chrome or Brave for improved memory handling. Additionally, we plan to decouple model inference from the frontend by offloading LLM queries to a lightweight local server.

These results validate the core interaction flow and highlight areas for engineering improvements in the next development cycle.

9 Evaluation Metrics and Methodology

To evaluate *Suno Saarthi*, we focus on three key areas:

(1) **System Performance and Responsiveness**
(2) **User Experience and Task Effectiveness**
(3) **Technology Acceptance (TAM-based)**

9.1 System Performance and Responsiveness

Goal: Ensure the system offers a smooth, natural, and hands-free interaction experience that minimizes distraction while driving.

- **Hands-Free Compliance Rate**
  - *Why it matters:* Drivers should be able to complete tasks using voice only, without touching the phone.
  - *How we measure:* Log each interaction type (voice vs manual).
  - *Success Criteria:* At least 90% of interactions should be voice-based.
- **Voice Command Success Rate**
  - *Why it matters:* The system must accurately understand and respond to user queries.
  - *How we measure:* Log all voice inputs, check for recognition accuracy and successful execution.
  - *Success Criteria:* 85%+ of voice commands should be recognized and executed correctly.
- **System Response Latency**
  - *Why it matters:* Delays in response can interrupt the flow of conversation.
  - *How we measure:* Measure time between end of user's speech and start of system's response.
  - *Success Criteria:* Average response time should be under 1.5 seconds.

9.2 Usability and Task Effectiveness

Goal: The system should help users complete driving-related tasks smoothly using only voice.

- **Task Completion Rate (TCR)**
  - *What we measure:* Can users complete navigation tasks like rerouting, finding landmarks, or asking road questions without help?
  - *Success Criteria:* 85%+ completion rate via voice alone.
- **Clarification Request Analysis**
  - *What we measure:* Are users asking follow-up questions due to confusion, or because they're engaged?
  - *Success Criteria:* 70%+ of clarifications should reflect healthy engagement, not confusion.
- **User Satisfaction Score**
  - *What we measure:* After each session, users rate their experience (1–5 scale) based on ease of understanding, assistant helpfulness, and responsiveness.
  - *Success Criteria:* Average satisfaction score of 4.0 or higher.

9.3 Perceived Usefulness and Ease of Use (TAM-Based)

Based on the Technology Acceptance Model (TAM), we evaluate how users *perceive* the usefulness and usability of Saarthi.

- **Perceived Usefulness (PU)**
  - *Statement:* "Using Suno Saarthi improves my ability to navigate confidently."
  - *How we measure:* 5-point Likert scale in post-use surveys.
  - *Success Criteria:* Average PU score of 4.0 or higher.
- **Perceived Ease of Use (PEOU)**
  - *Statement:* "Interacting with the assistant was simple and natural."
  - *How we measure:* 5-point Likert scale.
  - *Success Criteria:* Average PEOU score of 4.0 or higher.
- **Behavioral Intention to Use (BI)**
  - *Statement:* "I would prefer using this assistant over regular navigation apps in future drives."
  - *Success Criteria:* BI score of 4.0 or higher indicates adoption intent.

Evaluation Summary

Metric	Method	Target
Hands-Free Compliance Rate	Log interaction types (voice/manual)	> 90% voice-only
Voice Command Success Rate	Log accuracy and successful action	> 85%
System Response Latency	Measure delay between speech and response	< 1.5s average
Task Completion Rate (TCR)	Voice-only success in key tasks	> 85%
Clarification Request Quality	Analyze proactive vs. confusion queries	> 70% proactive
User Satisfaction	5-point Likert scale	> 4.0 avg.
TAM: PU / PEOU / BI	Post-use survey	PU/PEOU/BI > 4.0

Table 10. Evaluation Metrics Summary

10 Novelty of the System

10.1 Emotional Novelty

- **Human-like interaction:** The system mimics natural dialogue by **allowing for bidirectional conversation** letting users communicate as if talking to a knowledgeable co-passenger, reducing the feeling of isolation during solo drives.
- **Anxiety reduction:** By providing proactive assistance and adaptive responses based on driving context, the system helps alleviate anxiety associated with navigating complex or unfamiliar routes.
- **Personalized experience:** The system's ability to learn user preferences over time creates a sense of familiarity and trust, enhancing the emotional connection between the driver and the navigation assistant.

10.2 Technical Novelty

- **Advanced NLP integration:** The system combines Google's Routes API with state-of-the-art natural language processing, enabling it to understand and respond to complex, multi-part queries about specific road features like flyovers and narrow roads.
- **Context-aware navigation:** By integrating real-time data processing with user queries, the system provides timely and accurate guidance that adapts to changing road conditions and traffic patterns.
- **Multimodal accessibility:** The combination of voice-based interaction, visual impairment support, and potential multilingual capabilities makes the system accessible to a diverse range of users, including those with disabilities or language barriers.
- **Seamless cross-device integration:** The system's ability to synchronize across personal devices ensures a cohesive navigation experience, allowing users to plan routes on one device and seamlessly continue navigation on another.

References

[1] Mustufa Haider Abidi. 2024. A Comprehensive Review of Navigation Systems for Visually Impaired Individuals. *ResearchGate* (2024). \url{https://www.researchgate.net/publication/380860929_A_Comprehensive_Review_of_Navigation_Systems_for_Visually_Impaired_Individuals} Accessed: 2025-02-02.
[2] Joyce Chai. 2001. ACL Paper H01-1012. *ACL Anthology* (2001). \url{https://aclanthology.org/H01-1012.pdf} Accessed: 2025-02-02.
[3] Google Developers. 2024. Google Maps Documentation. https://developers.google.com/maps/documentation.
[4] Google AI. 2024. Google AI Developer Guide.
urlhttps://ai.google.dev/. Accessed: 2025-02-02.
[5] Google Developers. 2024. Directions API: Overview.
urlhttps://developers.google.com/maps/documentation/directions/overview. Accessed: 2025-02-02.
[6] Google Developers. 2024. Routes API: Flyover Narrow.
urlhttps://developers.google.com/maps/documentation/routes/flyover-narrow. Accessed: 2025-02-02.
[7] Srinivasan Janarthanam. 2013. ACL Paper W13-4025. *ACL Anthology* (2013). \url{https://aclanthology.org/W13-4025.pdf} Accessed: 2025-02-02.
[8] Anthony Bokolo Jnr. 2024. Human-Computer Interaction Research. *International Journal of Human-Computer Interaction* (2024). doi:10.1080/10447318.2024.2344145 Accessed: 2025-02-02.
[9] Vera Liao. 2020. Conversational Interfaces for Information Search. *ResearchGate* (2020). \url{https://www.researchgate.net/publication/341765332_Conversational_Interfaces_for_Information_Search} Accessed: 2025-02-02.
[10] Near Media. 2024. Conversational UI and Indoor Navigation. https://www.youtube.com/watch?v=m1ULM77nGmc&ab_channel=NearMedia.
[11] Near Media. 2024. Video: Google Maps Getting Conversational AI.
urlhttps://www.youtube.com/watch?v=m1ULM77nGmc. Accessed: 2025-02-02.
[12] NNGroup. 2024. Card Sorting. https://www.nngroup.com/articles/card-sorting/.
[13] Verena Traubinger. 2023. Inclusive Indoor-Navigation Systems With Conversational User Interfaces. *ResearchGate* (2023). \url{https://www.researchgate.net/publication/376229361_Inclusive_Indoor-Navigation_Systems_With_Conversational_User_Interfaces} Accessed: 2025-02-02.

[14] Unknown. 2001. Conversational User Interfaces. https://aclanthology.org/H01-1012.pdf. In *Association for Computational Linguistics Proceedings*.

[15] Unknown. 2020. Conversational Interfaces for Information Search. https://www.researchgate.net/publication/341765332_Conversational_Interfaces_for_Information_Search.

[16] Unknown. 2021. Principles of Evidence-Based Education in the Development of Giftedness. https://www.researchgate.net/publication/348110456_Principles_of_Evidence-Based_Education_in_the_Development_of_Giftedness.

[17] Unknown. 2024. Article from ACM Digital Library. *ACM Transactions* XX, X (2024), XX–XX. doi:10.1145/3491102

[18] Unknown. 2024. A Comprehensive Review of Navigation Systems for Visually Impaired Individuals. https://www.researchgate.net/publication/380860929_A_Comprehensive_Review_of_Navigation_Systems_for_Visually_Impaired_Individuals.

[19] Unknown. 2024. Inclusive Indoor-Navigation Systems With Conversational User Interfaces. https://www.researchgate.net/publication/376229361_Inclusive_Indoor-Navigation_Systems_With_Conversational_User_Interfaces.

[20] Unknown. 2024. Navigation Systems for the Visually Impaired. *International Journal of Human-Computer Interaction* XX, X (2024), XX–XX. doi:10.1080/10447318.2024.2344145