

Frontend Solution

Problem Statement:

Natural language translation engine for announcements and information dissemination at stations

Description:

Design a system to provide information in a desired Indian language on demand by passengers and other customers, in written and oral form. The system should be extendable to foreign languages for tourists as and when required. Limited vocabulary systems for common information about railway-needed services are acceptable. The scope of the system - announcements at stations, information over IVRS, information through chatbots and web interfaces. constraints to be considered - voice recognition in different languages; noisy ambience at stations; adequate computing power for on-the-fly content generation; and delivery on mobile devices.

Solution:

As the problem states that we have to provide a system that translate text to voice and give the desired information according to user's preference languages. The more I look into the solution for it I found out that this is the 90% frontend problem and only 10% backend problem.

Creating a user-friendly and efficient frontend for your multilingual information delivery system is crucial for ensuring user adoption and satisfaction

Here's how we are going to design our frontend:

1. User-Centric Design:

- ❖ Starting with gathering user requirements, understands the needs, preferences and constraints of different users, including passengers (regular travelers) and tourists.
- ❖ After gathering all the functional and non-functional requirements we are going to conduct user research and usability testing which gathers feedback and iterate on our design

2. Web And Mobile Interfaces:

- ❖ As we know nowadays everyone has mobile phone, so making a mobile interfaces is as much needed as web.
- ❖ We are going to develop both web and mobile responsive interfaces ensuring it adapts to various screen sizes and resolutions by keeping in mind that it still remain simple and intuitive so every literal and non-literal person can navigate through application

3. Multilingual Support:

- ❖ The most important feature that allows users to choose their desired Indian language or a foreign language for tourists. Will user clear and intuitive language icons or flags to represent languages.

4. Voice and Text Input:

- ❖ We are going to provide voice and text input options so that user either speak their queries or type them, depending on their preferences and the noise level at the station.

5. Voice Output:

- ❖ Incorporate text-to-speech (TTS) technology to convert translated text into spoken words. It will support various Indian languages and foreign language for tourists.

6. Simple and Intuitive Navigation:

- ❖ This is the most important intuitive part where we have to focus more. Cause there are lots of system which provide similar functionality but to surpass them we have to make simple and intuitive navigation (design) so that even a non-literal person can understand by the clean and straightforward navigation menu with easily recognizable icons or labels
- ❖ Implement a search bar for quick access to all the features if someone does not want explore. Everything should be accessible from the search bar.

7. Real-time Updates:

- ❖ Displaying real-time information is most crucial part of the application. Cause the passengers faces issues of real-time

information and they miss their train or get the wrong information.

- ❖ So we are going to display real-time information, such as train schedules, platform changes, and delays, prominently on the interface and highlight the critical announcements and alerts.

8. Interactive Maps:

- ❖ Include interactive maps of the station layout, highlighting important areas like platforms ticket counters, restrooms and waiting areas.

9. Accessibility Features:

- ❖ Implement accessibility features like high-contrast modes, screen readers, and options for users with disabilities.

10. Feedback Mechanism:

- ❖ Include a feedback for users to report the issues, provide suggestions, or request additional features.

11. Offline Mode:

- ❖ Design an offline mode that allows users to access essential information even when they have limited or no internet connectivity.

12. Scalability:

- ❖ Plan for scalability to accommodate a growing user base and additional features in the future.

13. FAQs and Help Guides:

- ❖ Provide user training materials and support resources, including FAQs and help guides.

14. Testing and Optimization:

- ❖ Continuously test and optimize the frontend for performance, responsiveness, and user satisfaction.

15. User Onboarding:

- ❖ Create an intuitive onboarding process for new users, guiding them through the features and languages selection. (It can be a video for a tutorial of the application)

16. Security and Privacy:

- ❖ Ensure that user data is protected and that the system complies with data privacy regulations.