System Requirements for Mothusi: AI-Powered Assistive Smart Assistant

1. Functional Requirements

1.1 Object Recognition

* The system shall capture images from the environment using a camera.
* The system shall pre-process the captured images to make them suitable for object detection (e.g., resizing, normalization).
* The system shall use a pre-trained AI model (YOLOv11) to detect and classify objects from the camera feed in real-time.
* The system shall output the detected object labels in real-time.
* The system shall trigger a voice feedback response to announce the detected object, such as "Object detected: Cat."

1.2 Voice Command Recognition

* The system shall use a microphone to capture voice commands from the user.
* The system shall process voice input using a speech recognition API or model (e.g., Google Speech API or TensorFlow Lite).
* The system shall recognize predefined voice commands such as "What is this?" and "Identify object."
* The system shall interpret the voice commands and map them to corresponding actions (e.g., start object recognition).
* The system shall provide feedback to the user about the status of the command, such as "Identifying object."

1.3 Voice Feedback

* The system shall convert text responses (e.g., detected object labels or command responses) into speech using a Text-to-Speech (TTS) engine (e.g., Google Text-to-Speech or pyttsx3).
* The system shall provide real-time voice feedback to the user through a speaker.
* The system shall ensure that the feedback is context-sensitive, responding to user queries or announcing detected objects appropriately.

1.4 User Interaction

* The system shall be able to handle multiple voice commands and respond with appropriate actions sequentially.
* The system shall provide verbal instructions to guide the user through its functionalities (e.g., "Please say a command").
* The system shall allow users to stop or reset an action with a predefined voice command such as "Stop" or "Cancel."

1.5 System Initialization and Setup

* The system shall initialize all components (camera, microphone, speaker, and AI models) upon startup.
* The system shall provide voice feedback to indicate that it is ready for use (e.g., "System ready").
* The system shall include setup instructions for hardware components (camera, microphone, speaker) and software dependencies (e.g., TensorFlow Lite, OpenCV).

2. Non-Functional Requirements

2.1 Performance

* The system shall process object recognition and provide results within 1 second from image capture in real-time.
* The system shall respond to voice commands within 2 seconds after command recognition.
* The system shall provide voice feedback with less than 1 second delay after object detection or command processing.

2.2 Usability

* The system shall provide clear and understandable voice feedback for the user.
* The system shall be able to handle different accents and variations in voice commands with at least 85% accuracy.
* The system shall ensure that voice commands are easy to remember and use (e.g., simple and intuitive commands such as "What is this?").

2.3 Accessibility

* The system shall be designed specifically for individuals with disabilities, ensuring it is accessible for users with visual impairments or motor difficulties.
* The system shall allow users to interact with it solely via voice, without requiring physical input.

2.4 Scalability

* The system shall support adding new objects to the recognition database with minimal reconfiguration.
* The system shall allow for adding new voice commands and responses without significant changes to the underlying architecture.

2.5 Reliability

* The system shall be available for use 95% of the time, minimizing downtime due to hardware or software issues.
* The system shall provide error handling for failed object detection or unrecognized commands, providing feedback to the user (e.g., "I didn't understand that command").

2.7 Resource Constraints

* The system shall be optimized for embedded hardware (e.g., Raspberry Pi or NVIDIA Jetson) with limited processing power and memory.
* The system shall consume less than 500MB of RAM during operation, ensuring smooth execution on resource-limited devices.

2.8 Portability

* The system shall be portable across different hardware platforms, including Raspberry Pi, NVIDIA Jetson, or similar embedded systems.
* The system shall be compatible with various camera and microphone hardware, provided they meet basic resolution and audio quality standards.

2.9 Maintainability

* The system shall use modular code design, making it easy to modify or extend specific functionalities (e.g., adding a new object detection model).
* The system shall include detailed documentation for setup, usage, and maintenance to facilitate future development and troubleshooting.

3. System Constraints

* The system shall use open-source libraries (e.g., TensorFlow Lite, OpenCV) to minimize development costs and maintain transparency.
* The system shall operate offline and should not depend on continuous internet connectivity, except for initial setup or updates.
* The system shall support voice commands in English as the primary language, with possible future extensions for other languages.