**Task 2: Document Integration Needs with ACS and BAP Platforms**

**2.1 Gather Functional Integration Requirements (Voice Input → Intent Mapping)**

**2.1.1. Introduction**

Functional integration requirements define how user voice inputs are captured, processed, and translated into actionable intents by the system. This ensures seamless interaction between the speech recognition component, the Natural Language Processing (NLP) engine, and the application backend. The objective is to enable a voice-enabled conversational system that correctly interprets user queries, maps them to predefined intents, and returns accurate responses.

**2.1.2 Input Processing Requirements**

• Speech Capture:  
- Support microphone input with minimal latency.  
- Handle different accents, dialects, and background noise.  
- Provide multi-device input support (mobile, web, IVR, smart speakers).  
  
• Speech-to-Text (STT) Conversion:  
- Integration with ASR (Automatic Speech Recognition) engines like Google Speech API, Azure Cognitive Services, or open-source alternatives.  
- Ensure high transcription accuracy (>90%).  
- Real-time transcription for dynamic conversations.

**2.1.3. Intent Mapping Requirements**

• NLP Integration:  
- Process the transcribed text using NLP/NLU engines (Dialogflow, Rasa, Wit.ai, etc.).  
- Extract intents (user goals) and entities (specific parameters like date, location, product name).  
- Support multilingual intent recognition.  
  
• Intent Classification:  
- Use ML models or rule-based classification to identify intent.  
- Define fallback intent when confidence score < threshold (e.g., 0.6).  
- Maintain intent taxonomy aligned with business use cases.  
  
• Entity Extraction:  
- Support named entity recognition (NER) for parameters.  
- Allow context-aware entity extraction.

**2.1.4. Functional Flow (Voice Input → Intent Mapping)**

1. User Speaks: The system captures audio input.  
2. ASR Layer: Converts speech into raw text.  
3. Preprocessing: Cleans and normalizes text.  
4. NLP Engine:  
 - Classifies text into an intent.  
 - Extracts relevant entities/slots.  
5. Application Logic: Maps intent to backend service or workflow.  
6. Response Generation: Returns a voice or text response to the user.

**2.1.5. Non-Functional Requirements**

• Accuracy: Maintain intent detection accuracy above 85%.  
• Latency: Total processing time < 1.5 seconds.  
• Scalability: Handle concurrent voice inputs from thousands of users.  
• Security & Privacy: Ensure compliance with GDPR/CCPA for voice data handling.  
• Adaptability: Continuous learning from user feedback to refine intent models.

**2.1.6. Example Use Case**

Voice Input: “Book me a train ticket to Hyderabad for tomorrow evening.”  
- Intent: Book\_Train\_Ticket  
- Entities:  
 • Destination = Hyderabad  
 • Date = Tomorrow  
 • Time = Evening  
  
System maps this to the booking workflow and triggers the ticket reservation process.