**Project Documentation**

**Project Title:** AI Enabled Conversational IVR Modernization Framework  
**Prepared for:** Saadhana mam (mentor of Infosys springboard batch 3)  
**Prepared by:** Gayatri Kotawar  
**Date:** 10 October 2025

**1. Introduction**

This document aims to assess the current IVR (Interactive Voice Response) system, which is built on older VXML technology, and lay out a clear plan for integrating a modern, AI-powered Conversational IVR.

I will use the **"Flight Booking Support"** use case for making of this project.

* **Task 1: Review of Existing IVR Architecture & Capabilities**

Our current system is a **Traditional, VXML-based IVR**.

**Work of the current system (The Current Architecture):**

1. **Call Comes In:** A customer calls the airline's support number.
2. **VXML Server Answers:** A server, running VoiceXML scripts, picks up the call.
3. **Play Prompts & Get Input:** It plays a pre-recorded menu: *"Welcome. For new bookings, press 1. For existing bookings, press 2. For flight status, press 3."*
4. **DTMF (Touch-Tone) Input:** The system only understands button presses on the keypad. If you press 2, it moves to the next script.
5. **Simple Backend Check:** It might ask for a 6-digit Booking Reference (PNR) and check a database.
6. **Action:** It either plays back pre-recorded information ("Your flight is on time") or transfers the call to a human agent.

* **Capabilities of the Current System:**

1. **In Basic Call Routing It** Sends calls to the right department (Booking, Support, Status).
2. **Simple Data Retrieval in which it** Can fetch basic information like flight status or booking details if the user provides an exact ID.
3. **Stable & Predictable:** It's a known technology that works reliably for simple tasks.

* **Limitations of the Current System:**

1. **Frustrating User Experience:** Customers are forced through "menu mazes." A simple question like "Is my flight delayed?" requires multiple button presses.
2. **No Natural Language Understanding:** It cannot understand a customer who says, "My flight to Miami is cancelled, I need to get on the next available one." It only understands PRESS 1.
3. **No Context:** Every interaction is isolated. The system doesn't remember who you are or what you said before.
4. **Limited Self-Service:** It can't handle complex requests like changing a flight date or solving a baggage issue. This leads to a high rate of calls being transferred to agents.

* **Task 2: Integration Plan with ACS & BAP for Flight Support**

Here is how we will align our new **AI Conversational IVR** with the existing ACS and BAP platforms.

**Chosen Use Case: Flight Booking Support**

**Step-by-Step Integration Process:**

1. **Voice Input & Understanding (AI IVR's Job):**
   * The customer speaks naturally.
   * The **ASR (Automatic Speech Recognition)** converts the speech to text: I need to change my flight from Toronto to Paris next week.
   * The **NLU (Natural Language Understanding)** identifies the **Intent** (change flight) and key **Entities**.
2. **Integration with ACS (Automated Customer Service):**
   * The AI IVR uses an API to ask the ACS that
     + *Who is this caller?*
     + *Are they a premium member?*
     + *What is their travel history?*
   * Which will help the ACS provides the customer profile. This allows the IVR to personalize the interaction.
3. **Integration with BAP (Business Application Platform):**
   * The AI IVR uses another API to ask the BAP
   * The BAP is the brain of the airline's operations. It provides real-time, actionable data. This allows the IVR to do more than just provide information it can perform actions.
4. **Action and Fulfilment:**
   * The AI IVR, now equipped with customer data from the ACS and flight data from the BAP, can have a true conversation.
   * Once the customer chooses, the IVR instructs the BAP to make the change and confirms the new itinerary.
5. **Seamless Handoff to Agent (if needed):**
   * If the request is too complex (e.g: rebooking of 10 members from same family), the AI IVR transfers the call to a live agent.
   * it will send the **entire conversation context** to the agent's screen via the ACS. The customer doesn't have to repeat themselves.

**4. Task 3: Technical Challenges, Constraints & Gaps**

Key challenges to solve:

**1. Compatibility Gap: "Static Scripts" vs. "Dynamic Conversations"**

* **Challenge:** VXML systems are built on static, pre-defined scripts. Our new AI system is dynamic and learns from conversations. They speak different languages.
* **Solution:** We need a robust **API Middleware Layer**. This layer will act as a "translator," taking the structured requests from the AI IVR and converting them into the specific API calls that the ACS and BAP platforms understand.

**2. Data Integration & Real-Time Access**

* **Challenge:** The AI's effectiveness depends on live, accurate data. If the BAP's flight inventory API is slow or the ACS's customer profile is outdated, the AI will give wrong answers.
* **Solution:** We must perform a thorough audit of all existing APIs. We may need to work with the teams managing ACS and BAP to create new, faster endpoints specifically designed for conversational AI.

**3. Security and Compliance**

* **Challenge:** The old VXML system had limited access to sensitive data. The new AI system will need to access customer profiles and payment information to be truly helpful. This introduces security risks.
* **Solution:** Implement strict security protocols for all data passing between systems. Use token-based authentication and ensure the AI IVR is PCI-DSS compliant if handling payments.

**4. Error Handling & "Graceful Failure"**

* **Challenge:** What happens if the BAP system is down? The old IVR would just play a "system busy" message. The AI IVR needs to be smart enough to handle this gracefully.
* **Solution:** Design sophisticated fallback mechanisms. For example, if the flight status API fails, the AI should say, "I'm having trouble accessing the live flight data right now, but I can connect you directly to an agent who can help," and then transfer the call.

**5. Performance & Scalability**

* **Challenge:** A VXML system can handle a predictable number of calls. A successful AI IVR might see a huge surge in usage, especially during flight disruptions. The cloud infrastructure must be able to scale instantly.
* **Solution:** Build the new AI IVR on a modern, cloud-native platform (like AWS, Google Cloud, or Azure) that supports auto-scaling to handle thousands of simultaneous calls.