Technical Specification: SMS Assistant with OpenAI Responses API, Botpress, and Twilio

Overview

This document outlines the architecture, integration flow, and technical specifications for implementing a student assistant for Ferrari Driving School using: - OpenAI Responses API for LLM-based reasoning and tool calling - Botpress for orchestration, flow control, and memory - Twilio (or another SMS provider) for two-way messaging

---

1. Architecture

```
graph TD
    A[Student (SMS)] --> B(Twilio or SMS Gateway)
    B --> C(Botpress)
    C --> D(OpenAI Responses API)
    C --> E(CRM API)
```

- Students interact via SMS.
- Messages are routed to Botpress from Twilio.
- Botpress manages conversational flow and memory.
- Botpress calls OpenAI Responses API for LLM decisions.
- OpenAI returns tool calls to be executed via Ferrari CRM REST API.

---

2. Components

2.1 Botpress

- Hosted on Botpress Cloud or self-hosted
- Manages:
- Session-based memory (e.g. driver's license)
- Input validations (via custom actions or code hooks)
- Flow transitions

Memory Schema (stored in `temp.user`):

```
{
  "drivers_license_number": "123456789",
  "classId": "CAR101",
  "lesson_duration": "2",
```

```
  "preferred_location": "BK"
}
```

Channels

- Twilio SMS
- Optional expansion: WhatsApp, Voice (via SIP trunk)

2.2 OpenAI Responses API

- Endpoint: `https://api.openai.com/v1/chat/completions`
- Model: `gpt-4o` or `gpt-4o-mini`
- Tool-calling: Schema-based tools (JSON) with `tool_choice: "auto"`

Example Tool Schema

```
{
  "type": "function",
  "function": {
    "name": "getStudentInfo",
    "description": "Retrieve student details by license number",
    "parameters": {
      "type": "object",
      "properties": {
        "studentId": {
          "type": "string",
          "description": "Driver's license number"
        }
      },
      "required": ["studentId"]
    }
  }
}
```

Example Call

```
{
  "model": "gpt-4o",
  "messages": [
    {"role": "system", "content": "You are an assistant for Ferrari Driving
School..."},
    {"role": "user", "content": "Book a lesson for 123456789"}
  ],
  "tools": [<tool schema objects>],
```

```
  "tool_choice": "auto"
}
```

2.3 CRM API (Ferrari Driving School)

- RESTful endpoints:
- `GET /students/{id}`
- `PATCH /students/{id}`
- `GET /lessons/availability`
- `POST /lessons`
- `POST /lessons/reschedule`
- `DELETE /lessons/{id}`
- Auth via API key or token header

---

3. Twilio SMS (or Alternative)

- Primary channel to reach users
- Botpress → Twilio Connector settings:
- Phone Number / Messaging Service SID
- Account SID
- Auth Token

Alternative SMS Providers

- Vonage (Nexmo)
- MessageBird
- Plivo
- Telnyx

---

4. Tool Flow Example: Booking a Lesson

1. User texts: "I want to book a lesson"
2. Botpress checks `temp.user.drivers_license_number`
3. If not found, prompts user: "Please provide your driver's license number."
4. Stores number, calls OpenAI with prompt
5. OpenAI returns `getStudentInfo` tool call
6. Botpress executes via CRM API
7. OpenAI follows up asking for:
8. Class type
9. Preferred date
10. Location (BK / AS)
11. Lesson duration
12. Botpress stores each step
13. When complete, OpenAI issues `bookAppointment` tool call

14. Botpress posts it to CRM and confirms via SMS

---

5. Error Handling

- CRM/API errors: Botpress catches and sends fallback:

  "We’re having trouble accessing your info. Please try again or call 718-278-6679."

- Timeout handling: Retry OpenAI call once; fall back to help message
- Validation: Phone, email, duration enums enforced in prompt & tool schema

---

6. Localization

- Initial: English only
- Expansion: Add Spanish via prompt instructions or translation API (e.g. DeepL, Google Translate)

---

7. Deployment & Monitoring

- Hosting:
- Botpress Cloud or Docker self-host
- CRM API protected behind rate limits
- Logs:
- Log all tool calls: name, studentId, latency, result
- Alert on failures or no response from OpenAI or CRM

---

8. Next Steps

- [ ] Clean up and normalize tool schemas (e.g., unified studentId key)
- [ ] Deploy Botpress with Twilio test channel
- [ ] Load OpenAI prompt + tools into Botpress/OpenAI bridge
- [ ] Pilot with internal test students
- [ ] Expand memory handling for reschedule/cancel flows