DLMAIPAIUC - Project AI Use Case

STUDENT: HUGO ALBUQUERQUE COSME DA SILVA

ID:92126125

1 – CONCEPTION PHASE

1.1 - Project Overview

The GrandVista Hotel Booking Chatbot is an interactive, user-friendly AI-powered assistant designed to facilitate hotel room bookings. Leveraging Large Language Model's language understanding capabilities, the chatbot uses an agentic approach to provide a conversational booking experience for users, handling essential inquiries and capturing necessary booking information efficiently.

1.2 - Objectives

The goal of this project is to develop a Chatbot for booking a hotel room that will have the following features:

- Conversational Interface: Engage users through natural language conversations to collect booking details, powered by an AI agent.
- The following Information will be collected by the BOT:
  - Full Name
  - Check-in and Check-out Dates
  - Number of Guests
  - Preferred Payment Method
  - Breakfast Inclusion Preference

- The chatbot will manage invalid inputs and guide users to provide correct information.
- At the end of the conversation, the chatbot will summarize and confirm booking details with users.

1.3 - Architectural Overview

The chatbot is structured into modular components to ensure maintainability and scalability. The primary components include:

- Frontend Interface: Built with Streamlit, providing an interactive web-based platform for user interactions.
- Backend: Developed using FastAPI, It will allow the creation of different endpoints that will validate the user's message, send the message to be processed by the AI agent, and receive the agent's response.
- Chatbot "Engine": Utilizes LangGraph to StateGraph to structure the dialog flow and manage the state of the conversation. Each node in the graph may contain one or more LLM chains from LangChain to process the user's request.

- Persistence: To build a chatbot that can handle multiple simultaneous conversations while managing conversation history, LangGraph's built-in persistence layer will be used. Since the goal is not to build a large-scale application robust to very high throughput (in terms of concurrent users), using an in-memory database solution (SqliteSaver) will be more than enough.

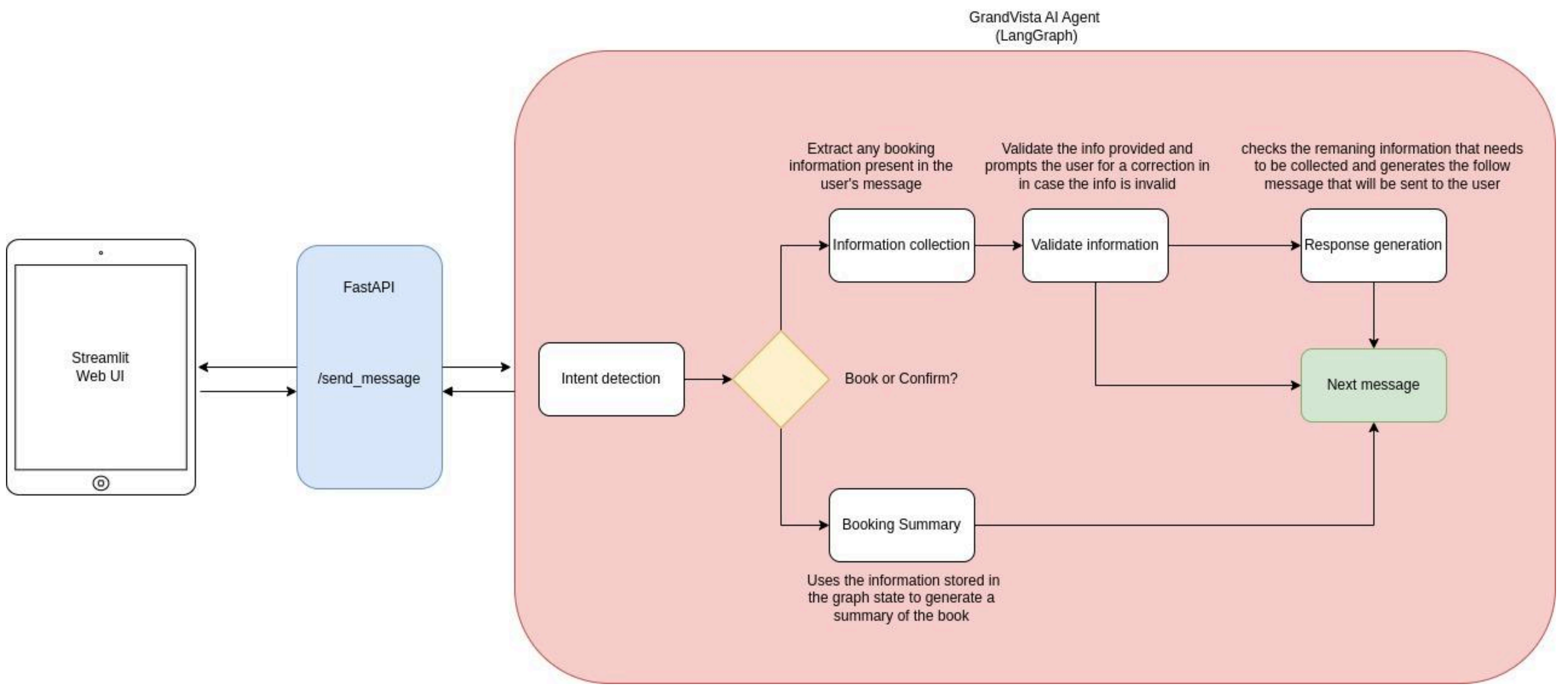
GrandVista AI Agent
(LangGraph)
Streamlit
Web UI
FastAPI
/send_message
Intent detection
Book or Confirm?
Extract any booking information present in the user's message
Information collection
Validate the info provided and prompts the user for a correction in in case the info is invalid
Validate information
checks the remaning information that needs to be collected and generates the follow message that will be sent to the user
Response generation
Next message
Booking Summary
Uses the information stored in the graph state to generate a summary of the book

Figure 1 - Solution Diagram

An example end-to-end workflow for the application would be:

1. The user starts a conversation by saying he wants to book a hotel. The user can provide a lot of information at once or merely introduce himself.
2. FastAPI acts as Middleware, decoupling the frontend from the backend. The frontend will communicate with the Ai agent through the */send_messages* endpoint via POST requests
3. The agent collects the relevant information present in the message, validates it, and generates a follow-up question based on the information that still needs to be collected
4. When all the information is collected, the agent generates a finalization prompt, summarizing the request.

Each node in the graph will contain a chain, where the appropriate logic will be built using prompt engineering. These details will be covered in depth in the next phase.