AI Case Study Report

Gökberk Ataer Bozdemir

1. Introduction

This report contains my work on the AI case study, where I was asked to develop a system that enables users to search for hotel rooms based on their visual preferences. The goal was to match user queries such as 'double room with sea view' to relevant hotel room images. I approached this task with a problem-solving mindset, using a modular architecture and multiple technologies to replicate a real-world solution.

2. Case Objective

The case required building a system that could:
- Analyze hotel room images to extract descriptive information,
- Understand user queries expressed in natural language,
- Return image URLs that match user-defined criteria,
- Optionally use agent-to-agent or multi-module communication architecture.

3. Technologies Used

- GPT-4V: I used GPT-4 with vision capabilities to analyze each of the 15 hotel room images and extract detailed descriptions. This acted as an image-to-text tool.
- Python & Pandas: I used these technologies to organize extracted metadata, to build structured tables, and to apply rule-based filtering.
- Semantic Logic Matching: I implemented logic similar to vector-based search by analyzing how user queries semantically align with image content, based on caption data and feature tags.
- Modular Agent-like Design: The system is structured around separate functional units resembling agent-based communication.

4. Agent-Based Architecture

Although not implemented as distributed services, I designed the solution with logical separation of responsibilities, aligning with the concept of agent-to-agent communication. Each module acted independently and passed structured information to the next. The roles are as follows:
- Image Analysis Agent: Processes the image and produces a descriptive caption. (Executed via GPT-4V.)
- Query Understanding Agent: Parses user queries into filterable parameters like 'bed_type: double' or 'view: sea'.
- Matching Agent: Performs both rule-based and semantic filtering to return relevant image matches.

This modular design simulates agent interaction without the need for a complex infrastructure or asynchronous communication frameworks, fulfilling the optional architectural request while keeping the system simple and readable.

5. System Flow

This system was built in a modular way to simulate a structured visual filtering pipeline. Although I did not code a full end-to-end application, I followed each logic step manually or with the help of specific tools. Below is a breakdown of the full flow from image input to final output:

1. Image Input & Captioning

Input: 15 hotel room images (15 from different web sites)
Process: I uploaded each image to ChatGPT and received a caption that describes visible elements like bed type, view, balcony, furniture, and lighting.

Example: For Image 1, GPT-4 Vision returned:
"Bright room with a large double bed, wide windows facing a city view, a desk on the side, and minimalistic decor."

2. Feature Extraction

Tool used: Manual inspection + structured tagging in Cursor
Process:
From each caption, I extracted key room attributes manually. I entered these in a table with fields like:

- Room Type, View, Balcony, AC, Desk, Capacity

Example:
Caption says “double bed”, so Room Type = Double
No mention of AC → marked as “Unclear”
"Wide windows facing sea" → View = Sea

I organized this in Cursor, which helped keep my notes and logic structured like an IDE.

3. Query Parsing

Input: 4 natural language queries
Process:
I broke each query down into attributes that I could match against the structured table. For example:

- Query 1: “Double rooms with a sea view”
  → Room Type = Double, View = Sea

This matching logic was written and applied manually in a spreadsheet.

6. Query Matching Results

Query 1 – Double rooms with a sea view:

- Image 5: https://imgur.com/XSNx12W

Query 2 – Rooms with a balcony and air conditioning, with a city view:

(There aren't any compatible images with this query but the images down below are the closest ones.)

- Image 13: https://imgur.com/l1HVcDk

- Image 15: https://imgur.com/nWFJVhi

(However, even these ones don't have any air conditioning or a balcony. They only have a city view.)

- Image 4: https://imgur.com/1SZLeYz (Even though this image was the closest one to the query, the AI didn't select it)

Query 3 – Triple rooms with a desk:

- Image 6: https://imgur.com/ykNviKl (Although there isn't a clear view of a desk, since there was a chair, the AI interpreted that there would be a table.)

- Image 11: https://imgur.com/iNiahfA

Query 4 – Rooms with a maximum capacity of 4 people:

- Image 6: https://imgur.com/ykNviKl

- Image 11: https://imgur.com/iNiahfA

7. Final Remarks

Throughout this project, I aimed to simulate how an AI-based search experience could be designed with a combination of structured metadata, natural language understanding, and modular thinking. While there are limitations due to the infrastructure constraints (e.g., vector indexing or real-time UI), I believe the proposed approach is practical and scalable with further development.