Project - High Level Design
on
Multimodal Hospitality Concept Visualizer

Course Name: Generative AI

Institution Name: Medicaps University – Datagami Skill Based Course

Student Name(s) & Enrolment Number(s):

Sr no	Student Name	Enrolment Number
1	Dhruv Dwivedi	EN22CS301333
2	Diya Jain	EN22CS301352
3	Bhavesh Thakur	EN22CS301265
4	Dishant Rathore	EN22CS301339
5	Chitralekha	EN22CS301304

Group Name: Group 04D3

Project Number: GAI-28

Industry Mentor Name: Prof. Suraj Nayak

University Mentor Name: Prof. Vineeta Rathore

Academic Year:2022-2026

Table of Contents

1. Introduction

1.1 Scope of the Document

This document describes the High-Level Design (HLD) of the **Multimodal Hospitality Concept Visualizer**.
The system integrates:

- Large Language Models (LLMs) for text generation
- Retrieval-Augmented Generation (RAG) for knowledge enhancement
- Stable Diffusion for image generation

The document explains architecture, components, data flow, APIs, and non-functional requirements.

1.2 Intended Audience

The intended users of the Multimodal Hospitality Concept Visualizer are:

1. Hospitality Designers

Architects and designers who want to visualize resort and hotel concepts quickly using AI-generated narratives and images.

2. Real Estate Developers

Developers who want conceptual visualization of hospitality projects before investing in detailed architectural planning.

3. Interior Designers

Professionals who need AI-assisted inspiration for interiors of luxury resorts, ecoretreats, wellness spas, and boutique hotels.

4. Hospitality Management Students

Students studying hotel management, architecture, or design who want conceptual understanding and visualization support.

5. AI Enthusiasts & Researchers

Individuals exploring multimodal AI systems combining LLMs, RAG, and image generation models.

1.3 System Overview

The Multimodal Hospitality Concept Visualizer is an AI-powered system that:

1. Accepts hospitality concept prompts (e.g., eco-resort, cliffside retreat, desert wellness spa).
2. Enhances prompts using **RAG (ChromaDB + Sentence Transformers)**.
3. Generates descriptive narratives using **Google Gemini 2.5 Flash**.
4. Produces high-quality architectural images using **Stable Diffusion v1.5**.
5. Displays multi-angle results (aerial, exterior, interior, landscape).

The system is deployed using **Gradio** and can be hosted on **Hugging Face Spaces** or local GPU.

2. System Design

2.1 Application Design

Architecture follows a modular AI pipeline:

User → Gradio UI → RAG Layer → LLM → Image Generator → Output Renderer Architecture

Type:

- Microservice-style modular architecture
- AI pipeline-based processing

2.2 Process Flow

1. User enters hospitality concept.
2. System embeds query using Sentence Transformers.
3. ChromaDB retrieves relevant documents.
4. Gemini enhances the prompt.
5. Enhanced prompt is sent to Stable Diffusion.
6. Images and narrative are generated.

7. Results displayed in UI.

2.3 Information Flow

Input:

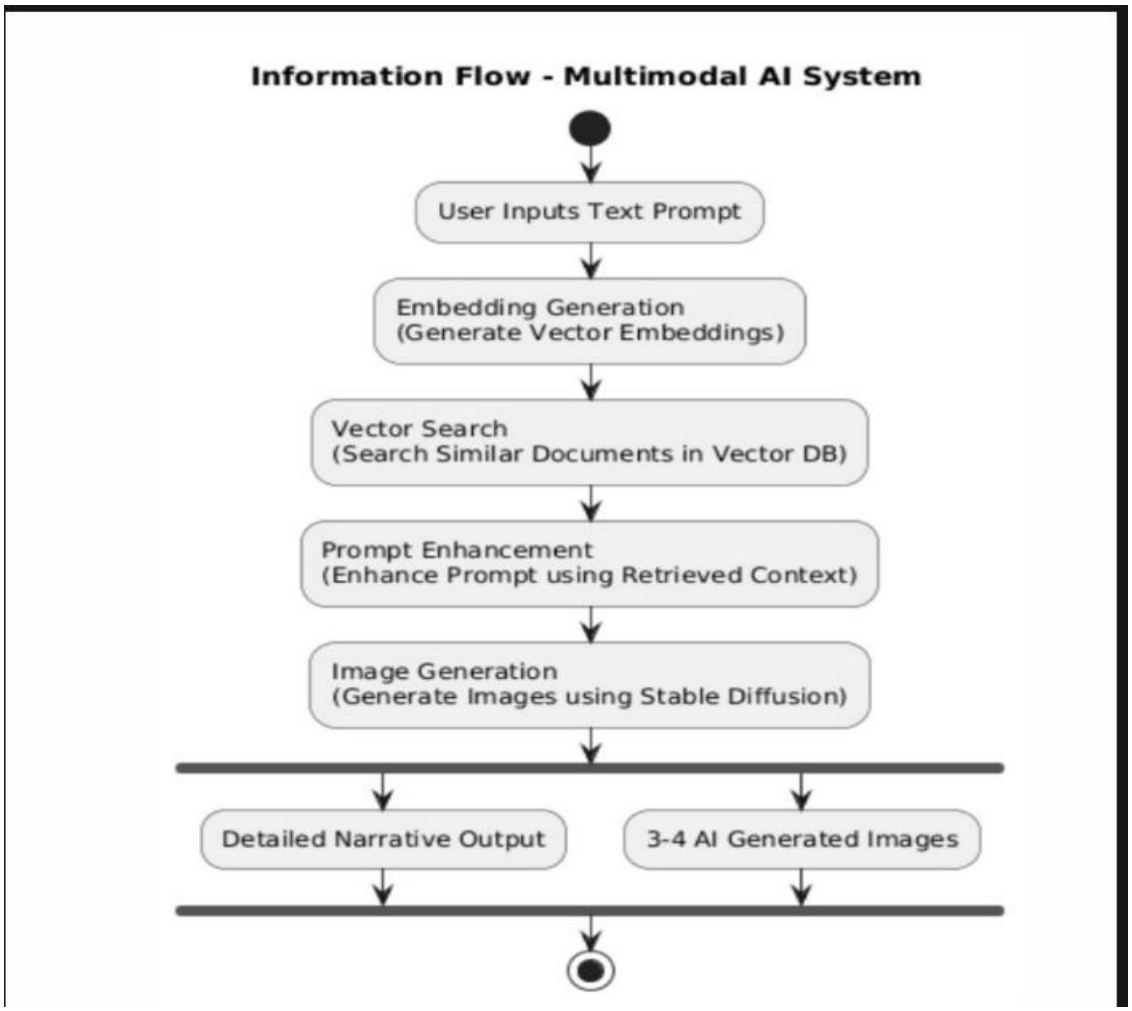
- Text Prompt

Processing:

- Embedding Generation
- Vector Search
- Prompt Enhancement
- Image Generation

Output:

- Detailed Narrative
- 3–4 AI Generated Images



2.4 Components Design

1. Frontend (Gradio)

- Text Input
- Generate Button
- Output Panel
- Download Options

2. RAG Layer

- Embedding Model: Sentence Transformers
- Vector Database: ChromaDB
- Semantic Search

3. LLM Layer

- Model: Google Gemini 2.5 Flash
- Task: Prompt enhancement + Narrative generation

4. Image Generation Layer

- Model: Stable Diffusion v1.5
- Library: Diffusers
- Hardware: GPU / CPU

5. Storage

- Temporary image storage
- Vector embeddings

2.5 Key Design Considerations

- Modular architecture for scalability
- GPU optimization for image generation
- Token management in Gemini
- Secure API key handling
- Low latency retrieval

2.6 API Catalogue

API Name	Method	Description
/generate	POST	Accepts prompt and returns narrative + images
/enhance_prompt	POST	Enhances prompt using Gemini
/retrieve_docs	POST	Retrieves related documents
/generate_image	POST	Generates image from enhanced prompt

3. Data Design

3.1 Data Model

Documents Collection

- doc_id
- content
- embedding_vector

User Request

- request_id
- input_prompt
- enhanced_prompt
- generated_images
- timestamp

3.2 Data Access Mechanism

- Embedding search via ChromaDB
- In-memory processing
- Temporary storage of generated images

3.3 Data Retention Policies

- Images stored temporarily
- Logs stored for debugging
- No personal user data retained

3.4 Data Migration

- Embeddings can be re-indexed
- Vector DB export/import supported

4. Interfaces

User Interface

- Web-based UI using Gradio

External Interfaces

- Gemini API
- Stable Diffusion model (local or HuggingFace)

5. State and Session Management

- Stateless architecture
- Each request processed independently
- Session maintained temporarily in Gradio runtime

6. Caching

- Embedding caching
- Model loading cached in memory
- Image model preloaded to reduce latency

7. Non-Functional Requirements

7.1 Security Aspects

- API keys stored in environment variables
- No hardcoded secrets
- HTTPS deployment
- Input validation

7.2 Performance Aspects

- GPU acceleration recommended

- Response time target:
    - Text generation: < 5 seconds
    - Image generation: 2-4 Min.
- Lazy loading of heavy models

8. References

- Google Gemini API Documentation
- Hugging Face Diffusers Documentation
- Chroma DB Official Documentation
- Sentence Transformers Documentation
- Gradio Documentation