Project - High Level Design

On

Marketing Planning Assistant Agent

Course Name: Agentic AI

Institution Name: Medicaps University – Datagami Skill Based Course

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

Sr no	Student Name	Enrolment Number
1.	Kavish Jha	EN22CS301494
2.	Krishna Pawar	EN22CS301524
3.	Kritika Purohit	EN22CS301527
4.	Kunal Patidar	EN22CS301538
5.	Jatin Patidar	EN22CS301458
6.	Javesh Khargonkar	EN22CS301460

Group Name: 06D5

Project Number: AAI-06

Industry Mentor Name:

University Mentor Name: Maya Yadav Baniya

Academic Year: 2025-2026

Table Content

1.INTRODUCTION

The Marketing Planning Assistant Agent is an AI-based system designed to automate the process of marketing planning. In traditional organizations, marketing planning is

activities such as competitor analysis, campaign strategy design, and execution scheduling are mostly performed manually. This makes the process time-consuming, error-prone, and heavily dependent on human expertise.
The proposed system aims to overcome these challenges by introducing an intelligent planning agent that can accept high-level marketing goals and convert them into structured and actionable execution plans. The system improves efficiency, consistency, and decision-making in marketing operations.

1.1. Scope of the document

The scope of this document is to present the High Level Design (HLD) of the Marketing Planning Assistant Agent. This document explains the overall structure of the system, major components, application design, process flow, data handling approach, interfaces, and non-functional requirements.

This HLD focuses only on conceptual and architectural design. Detailed coding, algorithms, class diagrams, and implementation details are not included. The document serves as a reference to understand how the system is designed and how different modules interact at a high level.

1.2. Intended Audience

- This document is intended for a wide range of stakeholders involved in the design, evaluation, and understanding of the system.
  The primary audience includes **students and project team members**, who use this document as a reference to understand the system structure and design decisions taken during development.
- The document is also intended for **faculty members and academic evaluators** who review the system design to assess conceptual clarity, logical structuring, and adherence to software engineering principles.
- Additionally, **system designers and technical reviewers** can use this document to gain a high-level understanding of the system architecture without examining code-level details. This document may also serve as a reference for future enhancement or redesign activities.

1.3. System overview

- The Marketing Planning Assistant Agent is an AI-driven system that supports marketing teams in planning and organizing marketing activities.
  The system accepts a **high-level marketing objective** such as market analysis, competitor advertisement study, or campaign planning.
- Once the objective is provided, the system analyzes it using intelligent planning and reasoning mechanisms. The objective is decomposed into smaller tasks, which are easier to manage and execute. These tasks are validated for feasibility and arranged in a logical sequence based on dependencies.
- The final output of the system is a structured and optimized marketing execution plan that helps reduce manual planning effort and improves efficiency and accuracy.

2. System Design

- The System Design section explains how the system is architected at a high level.
  The system follows a **modular and layered architecture**, where each module is responsible for a specific function. This approach ensures that the system is easy to maintain, scalable, and adaptable to future requirements.
- The design focuses on separation of concerns, ensuring that changes in one module do not affect other modules.

2.1. Application Design

- The application is designed as a set of loosely coupled modules that work together to achieve the system objectives.
  Each module has a clearly defined responsibility, which simplifies system maintenance and enhances readability.
- The major modules of the system are:
- **User Interface:** Responsible for accepting user input and displaying output
- **Planner Agent:** Performs goal analysis and task decomposition
- **Reasoning Engine:** Provides intelligent decision-making capabilities
- **Resource Validation Module:** Ensures feasibility of tasks
- **Execution Scheduler:** Determines task execution order and timeline
- **Output Generator:** Produces the final structured plan
- This modular design enables easy integration of additional features in future.

2.2. Process Flow

- The process flow defines the step-by-step working of the system from input to output:
- The user submits a high-level marketing goal.

- The system receives and interprets the goal.
- The Planner Agent decomposes the goal into smaller sub-tasks.
- Each task is validated for feasibility and resource availability.
- Task dependencies are identified and resolved.
- An execution schedule is generated.
- The final marketing execution plan is displayed to the user.
- This flow ensures systematic execution and minimizes planning errors.

.1.

2.3. Information Flow

Information flow describes how data moves across system components.
The marketing goal flows from the **User Interface** to the **Planner Agent**, where it is processed and decomposed into tasks.

Task-related information is forwarded to the **Resource Validation Module** for feasibility checks.
Validated task data is then passed to the **Execution Scheduler**, which generates a
Finally, the execution plan flows to the **Output Generator**, which presents the results to the user.

2.4. Components Design

Each system component plays a critical role:

- **User Interface:** Enables user interaction and input validation
- **Planner Agent:** Acts as the core intelligence of the system
- **Reasoning Engine:** Supports logical planning and decision-making
- **Resource Validator:** Ensures task feasibility
- **Execution Scheduler:** Handles task sequencing and scheduling
- **Output Generator:** Presents the final execution plan
- Each component is designed independently to ensure flexibility.

2.5. Key Design Considerations

The following design considerations were prioritized:

- Modularity and separation of concerns
- Scalability for handling complex marketing goals
- Maintainability and ease of enhancement
- Reliability and fault tolerance
- AI-driven autonomous planning

- These considerations ensure a robust and future-ready system.

2.6. API Catalogue

The system exposes high-level APIs that enable interaction between components:

- **submitGoal():** Accepts marketing objective from the user
- **decomposeTask():** Breaks down goal into sub-tasks
- **validateResources():** Validates feasibility of tasks
- **generateSchedule():** Generates structured execution timeline
- These APIs abstract internal processing and simplify system interaction.

3. Data Design

Data Design defines how data is represented and handled within the system. The system uses lightweight in-memory data structures to store goals, tasks, and schedules during execution.

3.1 Data Model

The primary data entities include:

- **Marketing Goal:** Represents the user objective
- **Task:** Represents individual actions derived from the goal
- **Resource:** Represents required resources
- **Execution Schedule:** Represents task order and timeline
- Each entity contains attributes that support planning and scheduling.

3.2 Data Access Mechanism

Data access is handled through in-memory structures such as lists and dictionaries. This approach avoids the overhead of database management and keeps the system lightweight.

3.3 Data Retention Policies

All data is retained only for the duration of system execution.
After the final plan is generated, data is discarded, ensuring efficient memory usage and improved security.

3.4 Data Migration

- Data Migration refers to the process of transferring data from one system, storage, or format to another.
  In the context of the Marketing Planning Assistant Agent, data migration is **not applicable in the current system design** because the system does not rely on persistent storage or external databases.
- All data, including marketing goals, tasks, and execution schedules, is processed **in-memory** during runtime. Once the system generates the final marketing execution plan, all temporary data is discarded. As a result, there is no requirement to migrate data between systems or storage layers.
- The absence of data migration simplifies the overall system architecture and reduces complexity. It also eliminates risks associated with data inconsistency, data loss, or compatibility issues.
- However, in future enhancements, if the system is extended to include **database storage**, **user profiles**, or **historical marketing plans**, data migration strategies may be required.

4. Interfaces

- The interface defines how the user interacts with the Marketing Planning Assistant Agent. In the current design, the system provides a **Command Line Interface (CLI)** that allows users to input high-level marketing goals and receive structured execution plans as output.
- The interface is kept simple to ensure ease of use and quick interaction. Users are not required to have technical expertise to operate the system.
  The design also supports extensibility, meaning that in future versions, the interface can be enhanced to include a **web-based dashboard** or **API-based interface** for integration with other marketing tools.
- The interface acts as a bridge between the user and the internal system components, ensuring smooth communication and data exchange.

5. State and Session Management

- State and session management describes how the system handles user sessions and execution state.
  The Marketing Planning Assistant Agent is designed as a **stateless system**, meaning that each execution is independent and does not rely on previous executions.
- No session data is stored after the execution is completed. This design simplifies system architecture and improves scalability.
  A stateless approach ensures that multiple users can interact with the system without interference and reduces the complexity of managing user sessions.

- If required in future, session management mechanisms can be introduced to support long-running tasks or personalized user experiences.

6. Caching

Caching refers to temporarily storing frequently accessed data to improve system performance.
In the current design, caching is **not implemented**, as the system performs lightweight, in-memory processing and does not handle repetitive data requests.

- However, caching can be considered as a future enhancement.
  For example, caching previously generated execution plans or commonly used task templates can reduce computation time and improve response speed.
- The decision to exclude caching in the current design helps maintain simplicity and reduces system overhead.

7. Non-Functional Requirements

Non-functional requirements define the quality attributes of the system rather than its functional behavior.
These requirements ensure that the system performs efficiently, securely, and reliably under expected operating conditions.

The key non-functional requirements considered for this system include **security** and **performance**.

7.1 Security Aspects

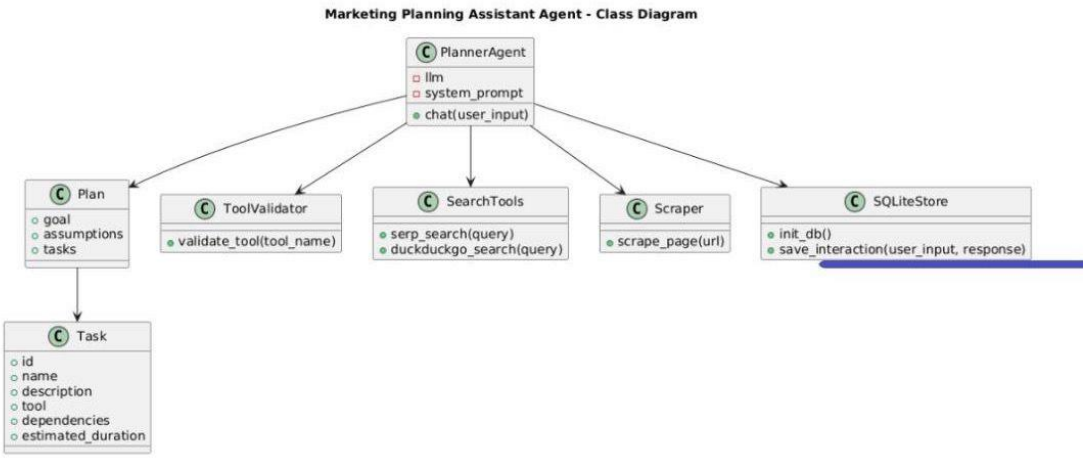
- Security is an important aspect of system design, especially when dealing with user input and planning data.
  In the current version, the system does not store sensitive user information, which minimizes security risks.
- User inputs are processed temporarily during execution and are not persisted.
  This approach reduces the possibility of unauthorized data access.
- In future versions, additional security mechanisms such as **authentication**, **authorization**, and **secure API access** can be implemented to enhance system security when integrating with external services or web-based interfaces.

7.2 Performance Aspects

- Performance defines how efficiently the system responds to user requests.
  The Marketing Planning Assistant Agent is designed to provide **fast response times** by using in-memory data processing and lightweight architecture.
- The modular design ensures that tasks such as planning, validation, and scheduling are handled efficiently.
  Since the system does not depend on external databases or heavy computations, it is capable of generating execution plans quickly.
- Future performance improvements may include optimization of task scheduling algorithms and the introduction of caching mechanisms.

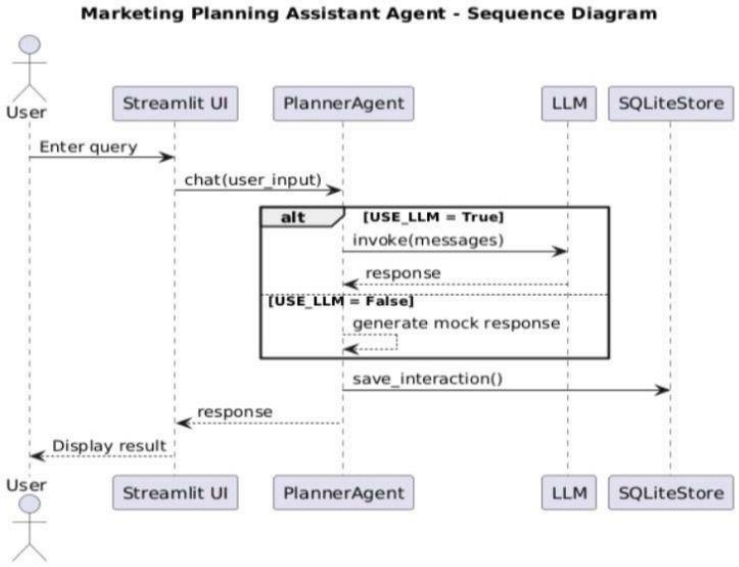
8. Diagram & Flowcharts :

8.1 Class Diagram

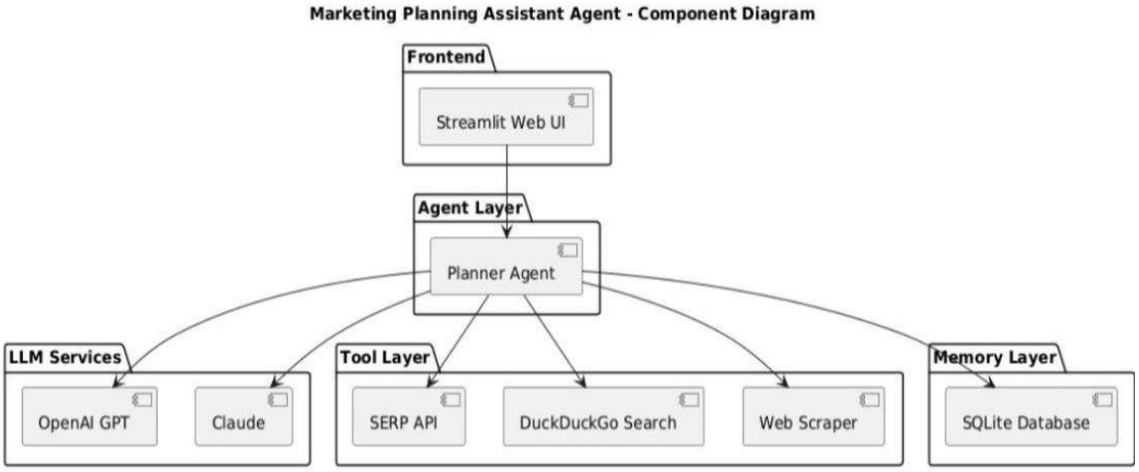


8.2 Activity Diagram

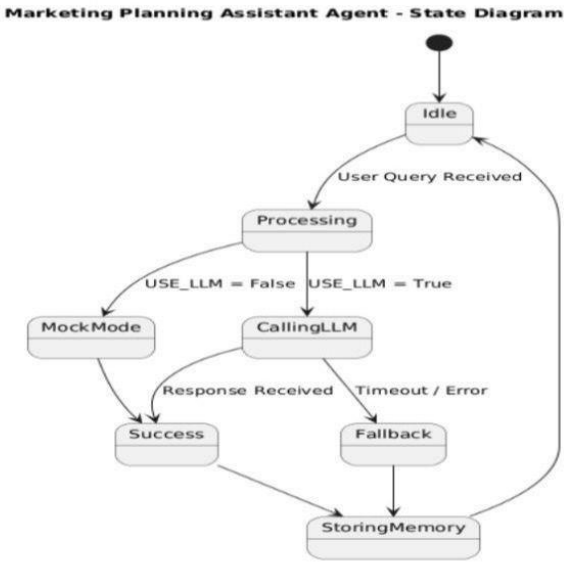
8.3 Sequence Diagram



8.4 Component Diagram



8.5 State Diagram



8.6 Database Design & ER Diagram

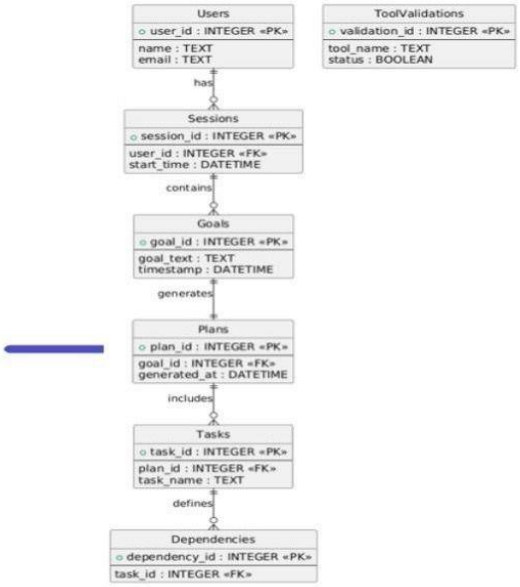
Our system's data model uses interconnected entities for marketing plan generation.

Core Entities and Attributes

- Users: user_id, name, email
- Sessions: session_id, user_id, start_time
- Goals: goal_id, goal_text, timestamp
- Plans: plan_id, goal_id, generated_at
- Tasks: task_id, plan_id, task_name
- Dependencies: dependency_id, task_id
- Tool Validations: validation_id, tool_name, status

Key Relationships

- User → Sessions
- Session → Goals
- Goal → Plan
- Plan → Tasks
- Task → Dependencies



References

The references section lists the technical resources and documentation used for understanding and designing the system.
These references provide theoretical and practical background related to AI planning, system architecture, and software design principles.

The key references include:

- Python Official Documentation
- LangChain Documentation
- CrewAI Documentation
- OpenAI API Documentation

These resources support the concepts applied in the system design and can be used for further study or enhancement.