Preliminary Capstone Project Writeup (English)

Title: Warehouse Analytics Dashboard with AI Agents

Subtitle: Autonomous, Multi-Agent System for Real-Time Supply Chain Optimization

Submission Track: Enterprise Agents

Project Description (The Pitch & Value)

1. The Problem: From Data Log to Strategic Insight

Traditional Warehouse Management Systems (WMS or SGA) excel at granular operational management but rarely offer intelligent data analysis to support tactical and strategic decision-making. Business analysts frequently require manual data exports and external business intelligence (BI) tools to gain insights, resulting in time-consuming searches and a lack of real-time visibility. The goal of this project is to address this gap by developing a smart module integrated with a WMS.

****Architectural Philosophy: Augmented Intelligence over Artificial Dependency****

This project champions a design philosophy where AI enhances rather than replaces traditional analytics. The architecture deliberately separates:

1. ****Core Analytical Engine****: Traditional dashboard with filters, charts, and KPIs that work independently

2. ****AI Intelligence Layer****: Specialized agents that provide conversational insights when available

This approach ensures that:

- ****Business operations continue uninterrupted**** during AI service outages

- ****Users build trust**** through reliable core functionality before adopting AI features

- ****Progressive adoption**** allows organizations to integrate AI at their own pace

- **Cost optimization** by using AI only for high-value insights rather than basic reporting

The result is a system that delivers the sophistication of AI-powered consultation without creating critical dependencies that could disrupt business intelligence workflows.

2. The Solution: AI-Powered Logistics Consultation

We have built a proof-of-concept **Warehouse Analytics Dashboard with an integrated AI Assistant**. This solution offers immediate, AI-powered consultation, leveraging the inherent flexibility of agentic AI to analyze warehouse data (expeditions and stock) via natural language queries. The primary value proposition is offering the analytical depth of a specialized logistics consultant alongside data visualization, streamlining inventory optimization, proactive demand forecasting, and real-time client service level monitoring.

3. Architecture and Implementation (The How)

The system employs a **Level 3 Collaborative Multi-Agent System (MAS)** (Hierarchical Orchestrator Pattern), reflecting a human organizational structure. This architecture is built on the **Google Agent Development Kit (ADK)** and hosted via a robust **FastAPI** backend integrated with a **Plotly Dash** frontend. Also it follows a ****decoupled microservices design**** hosted via Docker containers.

1. **Orchestrator Agent:** The primary agent (warehouse_orchestrator_agent) acts as the central router and coordinator. It interprets the user's complex query and delegates the task to the most appropriate specialized agent.

2. **Specialized Agents:** The Orchestrator uses three domain-specific agents as **tools** (AgentTool):

   ◦ client_service_agent: Expert in client performance and service levels.

   ◦ reference_expeditions_agent: Expert in material demand analysis and forecasting.

   ◦ stock_analysis_agent: Expert in inventory aging and stock optimization.

3. **Core Tools:** Each specialized agent is equipped with modular **Custom Tools** (Python functions) that access and analyze the data (e.g., get_top_clients, forecast_next_month_demand). This allows the agents to execute actions and retrieve real-time data beyond their training context.

4. **Model Selection:** All agents are powered by **Gemini 2.5 Flash**. This choice was deliberate: given high **rate limit errors (429 RESOURCE_EXHAUSTED)** experienced with the heavier gemini-2.5-pro model on the Free Tier, migrating to the more efficient Flash model ensured the system's stability and ability to handle the concurrent tool calls inherent to a multi-agent architecture.

Decoupled Architecture for Enhanced User Experience:

The system is designed with a clear separation between the analytical dashboard and AI intelligence layer, ensuring:

- **Operational Resilience**: The dashboard remains fully functional even during AI service interruptions

- **Progressive Enhancement**: Users access core analytics immediately, with AI insights as a value-added layer

- **Performance Optimization**: Heavy AI processing is isolated from the responsive UI layer

- **Business Continuity**: Critical warehouse analytics are never blocked by API rate limits or connectivity issues

4. **Fulfillment of Key Concepts (Technical Implementation)**

This project successfully implements **seven (7)** of the core concepts required for the Capstone Project, demonstrating a strong architectural foundation:

1. **Multi-agent system:** Implemented as a hierarchical structure where the orchestrator_agent uses three specialized agents as tools.

2. **Agent powered by an LLM:** All agents are instances of LlmAgent and are powered by **Gemini 2.5 Flash**.

3. **Custom Tools:** Modular Python functions (e.g., get_top_clients, get_avg_time_in_warehouse) are defined and used by the agents to perform data analysis and KPI calculation.

4. **Sessions & State Management:** The system uses InMemorySessionService from the ADK to manage conversation history, enabling the Orchestrator to maintain context for follow-up queries within a single session (session_id).

5. Tracing (Trajectory Auditability): The application integrates OpenTelemetry Tracing via an ADK Plugin (`minimal_tracing_plugin`). This provides complete auditability of the agent's trajectory ("The Trajectory is the Truth"), validating Orchestrator's planning and tool selection (as demonstrated in the analysis of the Service Level and Inventory Aging traces).

6. **Observability & Robustness (Logging & Metrics):** The application integrates explicit logging (setup_logger) in the FastAPI backend and data loading. Crucially, the system is engineered for **Robustness** by implementing an **Exponential Backoff** mechanism on API calls (4 attempts with exp_base=2) for transient errors, including the critical 429 RESOURCE_EXHAUSTED error.

7. **Production-Ready Deployment:** The system is containerized using Docker with separate services for API and frontend, demonstrating enterprise-grade deployment readiness and qualifying for the **5-point deployment bonus**.

5. **Quality and Future Evolution**

User-Centric Architecture Philosophy:

The current implementation embodies a **progressive enhancement** approach where:

- **Core analytics are always available** through the traditional dashboard interface

- **AI intelligence augments rather than replaces** existing functionality

- **Graceful degradation** ensures service continuity during AI service interruptions

- **Modular design** allows independent scaling of analytical and AI components

This design philosophy ensures that business intelligence delivery is never completely dependent on external AI services, while still leveraging their advanced capabilities when available.

****Technical Quality Pillars:****

The implementation adheres to the **Agent Quality principle that **Robustness** is essential. The deliberate choice of applying **Exponential Backoff** confirms that quality is treated as an architectural pillar, designed to mitigate unpredictable external failures (like API rate limits). Errors are managed gracefully by the FastAPI service, which prevents the client application (Dash) from crashing, ensuring a stable user experience.**

****Next Steps for Enhancement:** To secure the maximum score, future work will focus on the remaining pillars of Agent Quality:**

- **Agent Evaluation: Define a small golden dataset of scenarios to validate that the Orchestrator correctly routes queries to the appropriate specialized agents, demonstrating the strategic success of the multi-agent design.**

This agent was developed as part of the Google AI Agents Intensive Course Capstone Project.