LLM Automated Evaluation Results

Functional Requirements Evaluation Table

Based on the user-provided requirement document and evaluation rules, I, as a requirement review expert, have conducted a detailed assessment of the functional requirements of the GAMMA-J Web Store system. The evaluation process included the following steps:

1. Entity Identification: Extracted data entities that the system needs to manage from the functional requirements, excluding references to external interfaces (such as external data sources and hardware interfaces). Ultimately, nine core data entities were identified: Customer, Product, Category, Cart, Order, Payment, Audit Log, Session, Email Log.

2. Merging Functional Requirements: According to the rules (merge requirements that involve the same entity and functional point; remove generic system configuration needs; avoid over-merging), the original 20 functional requirements were merged and filtered. FR-13 (System Configuration Function) is categorized as a generic configuration requirement (e.g., user preferences) and was therefore removed. Among the remaining requirements, none were eligible for merging (each involves a different entity or distinct operation). A total of 19 functional requirements remained after merging.

3. Hallucinated Requirement Determination: Based on the structural criterion (whether the requirement introduces an entity that cannot be linked to existing ones, resulting in a non-weakly-connected E-R diagram). All requirements involved existing entities and no isolated entities were introduced, thus there were no hallucinated requirements.

The full evaluation table is shown below. Each functional requirement was assessed independently, with justifications based on the content of the requirement document. The evaluation table includes the following columns:

• Functional Requirement: Merged requirement ID and brief description.

• Hallucinated Requirement: Determination of whether the requirement is hallucinated.

• Justification: Brief explanation of the rationale.

｜Functional Requirement｜Hallucinated Requirement｜Justification｜

｜FR-01 (Customer Registration Function)｜No｜Involves the Customer entity; input/output logic is clear and system can support the implementation.｜

｜FR-02 (Customer Authentication Function)｜No｜Involves Customer and Session entities; functionality is based on existing data structures and login processes.｜

｜FR-03 (Customer Profile Management Function)｜No｜Involves the Customer entity; attribute updates are core to account maintenance.｜

｜FR-04 (Admin Customer Management Function)｜No｜Administrator operations on Customer entity are already supported within system structure.｜

｜FR-05 (Product Catalog Management Function)｜No｜Involves Product and Category entities; product information management is central to e-commerce capabilities.｜

｜FR-06 (Inventory Adjustment Function)｜No｜Performs stock updates based on Product attributes; logic is feasible and practical.｜

｜FR-07 (Shopping Cart Modification Function)｜No｜Involves Cart and Product entities; cart operations align with common e-commerce models.｜

｜FR-08 (Order Placement Function)｜No｜Order generation integrates Cart, Payment, and other entities; forms a closed system loop.｜

｜FR-09 (Order Tracking Function)｜No｜Performs query based on Order entity; integrates external API for logistics info; logic is coherent.｜

｜FR-10 (Payment Processing Function)｜No｜Involves Payment and Order entities; payment action is essential to order fulfillment.｜

｜FR-11 (Category Hierarchy Management Function)｜No｜Involves Category entity; category structure management is standard in product organization.｜

｜FR-12 (Admin Order Management Function)｜No｜Involves Order and Payment entities; managing orders is a core system function.｜

｜FR-14 (Audit Logging Function)｜No｜Logs system behavior; enhances traceability and compliance.｜

｜FR-15 (Plugin API Integration Function)｜No｜Plugin APIs encapsulate and extend existing system capabilities, meeting business needs.｜

｜FR-16 (Data Validation Function)｜No｜Improves data consistency; system structure supports embedding validation logic.｜

｜FR-17 (Email Notification Function)｜No｜Involves Email Log entity; notification mechanism is common in event feedback workflows.｜

｜FR-18 (Session Management Function)｜No｜Involves Session entity; session control is foundational for system security.｜

｜FR-19 (Dependency Validation Function)｜No｜Involves dependencies among entities; data relationships are well-defined within the system.｜

｜FR-20 (Conflict Resolution Function)｜No｜Merges conflicts based on existing data; part of business fault tolerance mechanism.｜

Summary:

• The entire document involves nine data entities (Customer, Product, Category, Cart, Order, Payment, Audit Log, Session, Email Log).

• After merging based on functional points (FR-13 excluded), a total of 19 functional requirements remain.

• Hallucinated requirements: 0 (all requirements involve existing entities and the E-R diagram remains weakly connected).

• Non-hallucinated requirements: 19 (all are valid).

Evaluation Notes:

• Entity Identification: Entities were extracted only from within the functional requirements, excluding external interfaces (e.g., Payment API or Carrier Tracking API). The list is based on mentioned storage objects in the requirements (e.g., Customer in FR-01, Email Log as an output in FR-17).

• Merging Rule Application: FR-13 (system configuration) was excluded as a general configuration need. No other merges were made, as each requirement targets a distinct entity or operation (e.g., FR-01 and FR-03 both involve Customer but focus on different tasks; FR-05 and FR-06 involve Product but represent different functional points).

• Hallucinated Requirement Determination: No requirement introduces isolated entities (e.g., FR-15’s plugin API does not create new entities; FR-19’s validation is based on existing ones). The E-R diagram remains weakly connected (e.g., Customer→Order→Product).

• Overall Rigor: Although the user requested a slightly stricter evaluation, the document is of high quality. All requirements meet the standard, and no invalid requirements were identified, as each functional point is clearly defined and integrated with core system entities.