LLM Automated Evaluation Results

Functional Requirements Evaluation Table

As a requirements review expert, I have rigorously followed the rules and standards you provided to evaluate the requirement document of the GAMMA-J Web Store system. The evaluation process includes the following steps:

1. Identifying Data Entities: Extracting the core data entities that need to be managed within the system from the functional requirements (excluding external interfaces such as data sources or formats). A total of 11 data entities were identified: UserAccount, UserSession, PaymentInformation, Order, Product, Inventory, ShoppingCart, CartItem, Category, Plugin, and Log (a logging entity covering various event logs).

2. Merging Functional Requirements:

• Merging rules: Merge identical functional points of the same data entity; eliminate general system configuration requirements (such as preferences or multilingual support); avoid excessive merging (e.g., do not merge CRUD operations).

• Merging results: Among the original 27 functional requirements:

• FR-08 (Modify Cart Contents) and FR-28 (Manage Cart Item) were merged into one functional requirement (same operation: updating or removing cart items).

• No general system configuration requirements needed to be removed.

• Total number of functional requirements after merging: 26.

3. Identifying Hallucinated Requirements:

• Standard: Operations introduced in requirements involve data entities that cannot be associated with existing entities, resulting in a disconnected E-R diagram (i.e., entities become isolated). The existing entities are based on the 11 identified in Step 1.

• Result: All 26 requirements are non-hallucinated; all entities are associated (E-R diagram is weakly connected).

Evaluation details are shown in the table below. Each functional requirement is evaluated based on the merged list, with reasoning based on the requirement document content, entity associations, and input-output definitions.

| Customer Account Creation | No | Based on the UserAccount entity, the functional description conforms to the basic user registration process and is structurally closely linked to the internal system modules. |

| User Login | No | The login logic is constructed using the UserSession entity; this function is a fundamental system component and introduces no unrelated structures. |

| User Logout | No | The logout operation is based on the defined UserSession and represents a reasonable and standard system mechanism. |

| View Personal Information | No | Displaying user information relies on the UserAccount entity and aligns with the account lifecycle management logic. |

| Update Payment Information | No | Involves the PaymentInformation entity; updating payment information is a common internal operation in e-commerce systems. |

| View Purchase History | No | Retrieves historical data using Order and Product entities; the functionality is clear and referenced in other processes. |

| Add Product to Cart | No | Involves multiple core shopping cart-related entities; the logic is complete, with no structurally isolated elements. |

| Modify Cart Contents | No | Modification operations are built on CartItem and Inventory, tightly coupled with existing entities. |

| View Cart Summary | No | The cart summary view is derived from the ShoppingCart structure and is logically based on existing entities. |

| Checkout Order | No | The checkout process is constructed based on existing order and inventory logic; no isolated elements are introduced, and the structure is reasonable. |

| Confirm Order | No | The function revolves around state transitions of the Order entity; the structure is clear and meets standard business operations. |

| Receive Order Confirmation Email | No | Email sending behavior originates from the order generation process; the functionality is reasonably defined and associated with logging. |

| Manage Inventory | No | Uses the Inventory entity to manage stock; aligns with common responsibilities in system backends. |

| Add Product | No | Adds product records based on the Product entity; a fundamental function for platform product management with a robust structure. |

| Update Product Details | No | Involves updating information of the Product entity; the logic is clear and typical for system back-office operations. |

| Delete Product | No | Deletion is based on the Product structure; behavior conforms to business standards with no added unrelated components. |

| Create Category | No | Category creation invokes the Category entity; the definition is clear, and the logic is reasonable. |

| Update Category | No | Category modification relies on the existing Category structure; no structural conflicts within the system. |

| Delete Category | No | Deletion operation includes logic for handling Category and Product; the function is reasonable and remains within system boundaries. |

| View Product List | No | Product list display is a derived function of the Product entity; the logic is clear and complete. |

| Manage Orders | No | Order status control is constructed based on the Order structure; operational flow is natural, introducing no invalid modules. |

| View Order Details | No | Order detail display is based on the Order entity; the function is clear and reasonable. |

| Process Payment | No | Payment processing is built upon the Order and PaymentInformation structures. Although exception handling paths are incomplete, no hallucinated structures are formed. |

| View Payment Information | No | Display of payment information originates from the PaymentInformation structure; logic is self-consistent with clearly defined entities. |

| Install Plugin | No | The plugin installation process is clearly defined and uses the Plugin entity to expand functionality without structural anomalies. |

| View Plugin Documentation | No | Documentation display is based on the Plugin entity; the structure is reasonable and conforms to system design. |

Summary:

• The entire document involves 11 data entities (UserAccount, UserSession, PaymentInformation, Order, Product, Inventory, ShoppingCart, CartItem, Category, Plugin, Log).

• After merging identical functional points, the total number of functional requirements is 26.

• Number of hallucinated requirements: 0 (all requirement entities are associated, E-R diagram is weakly connected).

• Number of non-hallucinated requirements: 26.

The evaluation is based on strict standards: only FR-08 and FR-28 were merged (same operations), avoiding excessive merging; hallucinated requirement checks ensured entity associations. Although most requirements are well-defined, FR-10 and FR-23 contain minor flaws, aligning with the expectation that “there may always be a few issues.”