

Part 1. Quality Scenario Catalog (QS Doc)

At least two concrete scenarios are provided for each of the seven mandatory Quality Attributes (QAs), structured using the six-part scenario template.

I. Availability (A)

Availability refers to the system's ability to be ready to carry out its task when needed, encompassing the masking or repairing of faults such that they do not become failures.

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
A.1	1,000 Concurrent End Users	External Payment Service times out or fails (Fault: crash/timing)	Flash Sale Peak Load (Overloaded Mode)	External Payment Service Connector / Order Processing Logic	The system employs a Circuit Breaker pattern to stop attempting immediate payment and utilizes Graceful Degradation by routing orders to a queue for asynchronous processing.	99% of order requests submitted are successfully accepted (queued or completed), and the Mean Time to Repair (MTTR) the payment connection fault is less than 5 minutes .

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
A.2	External Payment Service (Mock)	Transient payment failure (e.g., communication timeout, temporary card processing failure, momentary cash payment validation error)	Normal Operation / Degraded Operation	Order Processing Logic, Sale Persistence Transaction	The system immediately executes a Rollback of any partial state changes (such as potential log entries or resource reservations) and automatically performs a Retry of the payment and full transaction logic (up to $N=3$ attempts).	99% of transactions that initially fail due to transient payment errors are successfully completed within 5 seconds (Time to detect + Time to repair).
A.3	External Payment Service (Mock)	Payment Service reports permanent failure (e.g., "Card Declined," invalid API key)	Normal Operation	External Payment Service Connector/Order Processing Logic	The system applies a Rollback of any pending transaction steps, logs the failure (Audit tactic), and displays clear error feedback to the end user, prompting them to select a different payment method.	The entire transaction pipeline (from payment capture attempt to informing the user) results in zero unintended side effects (zero stock decrement, zero sale persistence).

II. Security (S)

Security is the measure of the system's ability to protect data and information from unauthorized access, focusing on confidentiality, integrity, and availability.

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
S.1	External Partner (VAR) System	Attempted catalog ingest request arrives with invalid or expired API key (Stimulus: Unauthorized attempt to access services)	Runtime (System Online, receiving partner feed)	Partner Catalog Ingest API Endpoint/Authorization Mechanism	The system uses the Authenticate Actors tactic to immediately deny the request, logs the access failure, and notifies the administrator.	100% of attempts originating from unauthorized external sources are denied access, measured by zero instances of successful data manipulation.
S.2	Malicious External Partner Feed (Data)	Input data field (e.g., Product Description in CSV/JSON) contains a known SQL Injection payload (Stimulus: unauthorized attempt to change data)	Integration / Ingestion process	Partner Catalog Ingest Module (data validation layer)	The system implements Validate Input (sanitization/filtering) on all incoming external data fields before persistence.	Zero malicious data payloads successfully reach the PostgreSQL database, measured by 100% adherence to defined database integrity constraints.

III. Modifiability (M)

Modifiability focuses on lowering the cost and risk of making changes, such as modifying functionality or adapting to new technology. Architectural decisions enable managing change as the system evolves.

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
M.1	Developer/Architect	Request to support a new partner integration format (e.g., XML)	Design/Development Time	Partner Catalog Ingest Module and internal interfaces	The module design uses the Use an Intermediary / Encapsulate tactic (e.g., an Adapter pattern) such that the new XML format parser can be added without modifying the existing CSV/JSON parsers or core domain logic.	The new XML format integration is completed, tested, and deployed with less than 20 person-hours of effort.
M.2	Product Owner / System Administrator	Request to instantly disable the "Flash Sale" feature post-deployment due to unexpected bug	Runtime	Flash Sale Pricing/Display Logic	The system employs a Feature Toggle mechanism (Defer Binding via configuration) allowing the feature to be disabled via an external configuration flag.	The feature is disabled and confirmed as inactive across all users within 5 seconds of the configuration change, requiring zero code changes or redeployment.

IV. Performance (P)

Performance relates to the system's ability to meet timing and throughput requirements when events occur. Tactics control the time or resources used to generate a response.

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
P.1	Automated Load Testing Tool (e.g., simulating 1,000 users)	1,000 order placement requests arrive per second (Stimulus: Event stream)	Peak Load / Overloaded Mode during Flash Sale	Order Submission Endpoint / API	The system uses the Manage Event Arrival tactic (Throttling/Queuing) to limit concurrent processing, prioritizing throughput over unbounded latency.	The average latency for 95% of accepted order requests remains below 500 milliseconds .
P.2	Multiple concurrent internal processes (e.g., multiple web workers)	Multiple simultaneous transactions attempt to modify the stock level for the same product	Normal Operation	Product / Inventory Database records	The system utilizes the Introduce Concurrency tactic (e.g., database transaction locking/isolation levels) to ensure efficient shared resource access.	Database lock wait time (blocked time) for critical stock updates remains below 50 milliseconds during the peak load window.

V. Integrability (I)

Integrability concerns the costs and technical risks of making separately developed components cooperate, particularly when integrating components supplied by external vendors (like partners/VARs).

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
I.1	External Reseller API (e.g., legacy SOAP/XML communication)	Request to onboard the reseller's checkout API, which uses different data semantic protocols (Stimulus: Add new component)	Integration / Development Time	New Reseller API Connector	The system applies the Tailor Interface tactic via an Adapter pattern to translate data formats and protocol sequences between the external system and the internal order service.	The new Reseller API is integrated, tested, and operationalized in less than 40 person-hours of effort.
I.2	Internal Development Team	A new internal reporting service needs to consume incoming Partner Catalog data feed for real-time reporting	Development Time	Partner Catalog Ingest System	The Partner Ingest system uses the Use an Intermediary tactic (Publish-Subscribe pattern) to broadcast data updates, reducing direct coupling.	Adding the new reporting consumer requires modification of zero lines of code in the existing Partner Catalog Ingest module.

VI. Testability (T)

Testability refers to the ease with which software can be made to demonstrate its faults through testing, promoting control and observability.

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
T.1	Automated Testing Tool (Load simulation script)	Execution of a load test script simulating 500 Flash Sale transactions	Staging / Testing environment	Entire system (Order flow, payment, stock update)	The testing infrastructure successfully employs the Record/Playback tactic to re-create the exact state and traffic patterns that caused performance degradation in a previous run.	The effort required to replicate the exact flash sale workload condition (including system state and input data) is reduced to less than 1 hour .
T.2	Unit Tester / Developer	Need to verify the logic of the Order Processor when the external Payment Service returns a transient failure	Development time	Order Processor module	The developer uses Dependency Injection to substitute the real Payment Service dependency with a mock object that simulates a transient timeout or failure.	The test case executes and validates the full retry/rollback logic in less than 5 seconds (Time to perform tests).

VII. Usability (U)

Usability is concerned with how easy it is for the user to accomplish a desired task, minimizing the impact of user errors, and increasing confidence and satisfaction.

#	Source	Stimulus	Environment	Artifact	Response	Response Measure
U.1	End User	The simulated Payment Service declines the transaction (Stimulus: Minimize impact of errors)	Runtime (Checkout process)	User Interface (Checkout Page)	The system employs the Minimize Impact of User Errors tactic by providing clear, immediate, actionable error feedback that suggests an alternative payment method.	User successfully completes a modified transaction (after initial failure) in less than 90 seconds (Task time/Number of errors).
U.2	End User	User clicks "Confirm Order," triggering a lengthy sequence of stock checks and database transactions (Stimulus: Use a system efficiently)	Peak Load	User Interface (Order Confirmation/ Wait screen)	The system uses the Maintain System Model tactic (Progress Indicator) to provide immediate feedback, showing the current state and estimated completion time.	User satisfaction score (e.g., SUS score) for transactions taking longer than 10 seconds remains above 80% .

