


Challenger settings

Challenge intensity 

☐ mild


☒ spicy

Agent Selector

Tip: You can also explicitly type an agent tag in your message, e.g.,

`@novelty_radar` or `@bug_logger` .

Preferred Agent (optional)

Auto (Smart Intent Detection) 


Upload Document

Requirements / Logs / Specification (txt / pdf)


Drag and drop file here


Limit 200MB per file • TXT, PDF

Browse files



Requirements Specifica...
38.6KB





Preview of Uploaded Document
(first 1000 characters)

Defying Gravity in Testing



@AssumptionBuster I’m trying to distinguish if some state still lingers after a reservation is canceled then reserved again. What are the main blind spots or limitations in my testing logic?



- Logic and Assumption Elicitation:
 - The hypothesis assumes cancellation fully clears all reservation-related state before rebooking.
 - The requirement FR-8 implies the UI must update remaining slots and status badges after changes.
- Adversarial Critique — Assumption Weaknesses:
 - System's in-memory architecture increases risk that client-side variables persist across cancel and re-reserve cycles.
 - The logic may ignore race conditions, caching, User ID association, and admin overrides that alter state lifecycle.
- Persona-Based Debating:
 - As an attacker, could forged or mismatched User IDs cause lingering state attribution errors?
 - As a concurrent-user scenario, might overlapping actions create stale UI or inconsistent in-memory counts?
- Reflective Challenge:
 - Which exact observable behaviors would indicate lingering state rather than expected eventual consistency?
 - Specify timing boundaries, User ID handling, and admin actions that the hypothesis depends on.

Type here... (e.g., '@Brainstormer Find unseen
exploration paths in the unloaded GSN')

