



✓ Flutter Widget Test Report

File: test/screens/dashboard/pod_status_screen_test.dart

Run Time: ~00:15 seconds

Status: ✓ All tests passed (16/16)

📁 Test Summary

#	Test Description	Status
1	Displays "No device connected" when disconnected	✓ Passed
2	Displays loading state initially	✓ Passed
3	Displays "Stream Error" on stream error	✓ Passed
4	Displays "No device connected" overlay with correct style	✓ Passed
5	Displays received data correctly	✓ Passed
6	Displays detailed pod data Low/Mid	✓ Passed
7	Displays detailed pod data Low/High	✓ Passed
8	Displays detailed pod data Mid/Low	✓ Passed
9	Displays detailed pod data High/Mid	✓ Passed
10	Displays detailed pod data Mid/High	✓ Passed
11	Displays detailed pod data High/High	✓ Passed
12	Displays detailed pod data Low/Low	✓ Passed
13	Displays detailed pod data Mid/Mid	✓ Passed
14	⚠ Battery edge case set to 300%	✓ Passed
15	⚠ Pods edge case set to 300ml (beyond 200ml max)	✓ Passed
16	Final test result	✓ All passed

🔍 Observations

- The core UI components of `PodStatusScreen` performed as expected under all standard and edge data conditions.
 - Your test suite now includes **extreme values** to test the **limits** of the app's response logic — a strong step toward robustness and patient safety.
-

⚠️ Edge Case Concerns

! Battery at 300%

This value exceeds physiological and hardware constraints. Displaying this without a warning could:

- Confuse users
- Indicate a firmware or transmission error
- Lead to a **false sense of security**

! Pod Fill at 300ml (beyond safe max of 200ml)

This also exceeds normal capacity. Unchecked, it may:

- Hide a device malfunction
 - Risk **overdosing** if misinterpreted
-

✓ Recommendations

⚠️ Even though these edge cases **passed the UI rendering test**, they highlight a **critical safety issue** in logic validation.

Suggested Safeguards:

1. **Validation Layer:** Clamp or reject incoming values outside safe thresholds (e.g., `battery > 100`, `fill > 200`).
2. **Warning UI:** Display a warning banner, toast, or modal with:

- "Unexpected sensor value received"
 - "Please reconnect or contact support"
3. **Log the incident:** Include local logging or cloud error reporting for anomalies.
 4. **Unit + Widget Tests:** Add tests that **expect** an error/warning UI when out-of-bounds data is received.
-