Fantastisch 👏 — dit wordt een mooie, duidelijke samenvatting van twee grote stappen:

**TDA-123 Search & Filters Redesign** en **Map Interaction Hardening**.

Hieronder vind je een kant-en-klare Jira-comment in professioneel Engels (zoals je team dat zou verwachten), opgedeeld in twee delen — één voor de *Search & Filters Redesign* (fase 1) en één voor de *Map Interaction Hardening* (fase 2).

**🧩**

**TDA-123 – Search & Filters Redesign**

**Scope & Objectives**

We redesigned the search and filtering experience within the Bottom Sheet to make it consistent, reactive, and performant while keeping all existing functionality intact.

The goal was to unify filtering, local search, and UI state management without introducing new components or regressions.

**Implementation summary**

* Added a new composable useSearch hook:
  + Provides **debounced search** (≈ 350 ms) and **in-memory LRU cache** scoped to the session.
  + Handles caching and memoization to avoid redundant filtering and re-renders.
  + Supplies lightweight local autocomplete suggestions (top 8 matches from loaded locations).
* Refactored Filters.tsx:
  + Migrated to **shadcn/ui** components for consistent styling and behavior (Input, Button, Badge).
  + Added **search icons** (lucide-react) for Search / Clear.
  + Replaced the old <select> category filter with chip-style category buttons.
  + Integrated suggestions dropdown and keyboard navigation (↑ / ↓ / Enter / Esc).
* Updated App.tsx:
  + Integrated useSearch and removed the old useMemo debounce logic.
  + Unified map, list, and Bottom Sheet sync; maintained existing onSelect and onFilterChange flows.
* Minor polish:
  + Added transition-colors for smooth visual feedback in LocationList.tsx.
  + Created Docs/frontend-search.md documenting debounce, caching, and event flow.

**User-experience improvements**

* Real-time, debounced search reduces unnecessary renders and API calls.
* Category chips make filtering immediate and intuitive.
* Autocomplete suggestions make discovery faster on mobile.
* “Clear” icon and consistent keyboard support improve accessibility.
* Bottom Sheet remains reactive and consistent with the design system.

**Result**

* Search/filter UX feels instant and consistent with Google Maps-style mobile interactions.
* No extra network requests introduced (still fully local filtering).
* Existing map/list selection logic preserved 1:1.

**🗺️**

**Map Interaction Hardening (Follow-up to TDA-123)**

**Context**

After TDA-123, we uncovered a deeper issue in the map interaction layer:

markers and clusters could become unresponsive (“dead map”) after the first selection → deselection cycle due to overlapping layers, pointer-event side effects, and lifecycle issues in Mapbox GL listener management.

**Diagnosis & fixes (progressive hardening)**

1. **Overlay interference**
   * .tda-bs-backdrop sometimes remained active even when visually hidden.
   * Fixed with display:none / pointer-events:none when not visible.
2. **Popup & background click logic**
   * Background clicks weren’t always recognized as true “deselect” events.
   * Introduced a robust hit-test in MapView.tsx that differentiates between interactive layers, highlight rings, and empty map clicks.
   * Popup close events now also trigger onMapClick() → selectedId = null.
3. **Cursor behavior**
   * Added centralized cursor management (mousemove listener) that inspects current hover hits and sets the cursor accordingly.
   * Prevents the “stuck pointer” issue where the whole map behaved as one giant clickable area.
4. **Persistent listener lifecycle**
   * **Root cause:** event handlers in MarkerLayer.tsx were bound inside an effect depending on changing props.

Cleanup ran on deselect but re-attachment didn’t.

* + **Fix:**
    - Attach listeners (click on clusters/points, mousemove for cursor) **once per map instance**, cleanup only on unmount.
    - Store handlers in stable refs (clusterClickRef, pointClickRef, mouseMoveRef) and the latest onSelect in onSelectRef.
    - tsc build now passes cleanly (no TS2304/TS2552).

**Behavior after fix**

* ✅ Markers and clusters stay fully clickable after any number of selection / deselection cycles.
* ✅ Hover cursor switches correctly (grab ↔ pointer) depending on feature under mouse.
* ✅ Popup close and background clicks deselect cleanly and remove highlight.
* ✅ No full-screen overlays block the map.
* ✅ Map remains responsive and smooth on both desktop and mobile.

**Result**

* **Interactivity hardened**: the map can no longer enter a “dead” or “stuck pointer” state.
* **Build stability**: all scope-related TS errors resolved.
* **UX consistency**: the map now behaves predictably like Google Maps—responsive, tactile, and state-safe.

**✅**

**Overall outcome**

Both parts combined deliver a major upgrade in usability and technical resilience:

* Unified, performant search/filter experience inside the Bottom Sheet.
* Fully reliable marker / cluster interaction model across all selection states.
* Clean, maintainable code (type-safe, single source of truth for handlers and cursor).

**Status:** Feature complete, stable in production build.

**Next step:** Merge → QA regression test → close TDA-123.