HIGHTOWER.DIGITAL® · CREATE EXCELLENCE!!™

Domain Owner — Use Case Report

IT REQUIREMENTS USE CASE · DO1 · VERSION R1.A

Author: LW Atkinson (CEO/CTO), HighTower Digital Inc.

Date: 06/25/2026 · **Classification:** HDI Confidential

Methods: Agile Scrum + Lean + XP · **Product:** IT Requirements As-A-Service (MSP SLA #1–5)

File: DO1_UCR1a

5 Domain Owner (Domain / Brand Spec Document)

The Domain Owner sits in **Brand Management** @ **Enterprise** and is accountable for **domain integrity and brand consistency** across every requirement. Grounded in **Domain-Driven Design (DDD)** — bounded contexts, ubiquitous language, and a shared domain model — the Domain/Brand Spec ensures terminology, tone, accessibility, and visual identity are encoded as requirements, not afterthoughts. For HighTower.Digital this protects the "Create Excellence!!" brand promise and the shield identity across all client-facing surfaces.

5.1 Validated Design Document

5.1.1 STORY INDEX

Story	Domain concern	Standard
DM-01	Ubiquitous language enforcement	DDD
DM-02	Brand token system (color/type)	Design system
DM-03	Accessibility (WCAG 2.2 AA)	WCAG
DM-04	Bounded-context API naming	DDD

5.1.2 STORY BACKLOG

Domain stories are cross-cutting; the Domain Owner reviews every other backlog for terminology and brand conformance before "Done."

5.1.3 STORY LIST (FEATURES / FUNCTIONS)

The **ubiquitous language glossary** (Appendix D) is the single source of truth; code, UI, and docs must use the same terms.

5.2 Frontend Feature List

5.2.1 FEATURE CARD — BRAND TOKEN PROVIDER

FRONTEND · FC-FE-DM-01

Feature: Brand token provider (React/TS) — injects HighTower.Digital design tokens (navy/blue/cyan/purple) so every surface is on-brand by default.

(1) Header code

```
export const hdTokens = {
  navy: "#0a1f44", blue: "#1b4fd8", cyan: "#22b8e6", purple: "#7b4fe0",
} as const;
```

(2) Function code

```
import { createContext, useContext } from "react";
const BrandCtx = createContext(hdTokens);
export const useBrand = () => useContext(BrandCtx);
export const BrandProvider = ({ children }: { children: React.ReactNode }) =>
  <BrandCtx.Provider value={hdTokens}>{children}</BrandCtx.Provider>;
```

(3) Footer code

```
export { hdTokens, useBrand, BrandProvider };
```

(4) Use Case Example — Any component calls `useBrand()` to render with approved brand colors — no hardcoded hex values pass review.

(5) Step-by-Step Example — Wrap app in `BrandProvider` → components consume `useBrand()` → lint rule blocks raw hex outside tokens.

5.3 Middleware Features List

5.3.1 FEATURE CARD — UBIQUITOUS LANGUAGE VALIDATOR

MIDDLEWARE · FC-MW-DM-01

Feature: Ubiquitous-language validator (Node.js) — rejects API payloads using non-canonical domain terms (e.g., "ticket" vs "trouble-ticket").

(1) Header code

```
const canonical: Record<string,string> = { ticket: "troubleTicket", req: "requirement" };
```

(2) Function code

```
export function normalizeTerms(payload: Record<string, unknown>) {
  const out: Record<string, unknown> = {};
  for (const [k, v] of Object.entries(payload)) out[canonical[k] ?? k] = v;
  return out;
}
```

(3) Footer code

```
export { normalizeTerms };
```

(4) Use Case Example — A legacy client sends `{ ticket: ... }`; middleware normalizes to the ubiquitous `troubleTicket` before persistence.

(5) Step-by-Step Example — Receive payload → map keys to canonical terms → forward normalized object downstream.

5.4 Backend Feature List

5.4.1 FEATURE CARD — BOUNDED-CONTEXT REGISTRY

BACKEND · FC-BE-DM-01

Feature: Bounded-context registry (Python) — declares which domain owns which entity, preventing model leakage across contexts.

(1) Header code

```python
from dataclasses import dataclass, field
@dataclass
class BoundedContext:
    name: str
    entities: set[str] = field(default_factory=set)
```

(2) Function code

```python
def owns(ctx: BoundedContext, entity: str) -> bool:
    return entity in ctx.entities
```

(3) Footer code

```python
__all__ = ["BoundedContext", "owns"]
```

(4) Use Case Example — A service checks `owns(billing_ctx, "Invoice")` before mutating an entity, enforcing context boundaries.

(5) Step-by-Step Example — Register contexts → assign entities → guard mutations with `owns()` checks.

5.5 Technical Requirements

5.5.1 BUSINESS CASES

BC-05: *"Consistent ubiquitous language and brand tokens reduce onboarding time and prevent costly brand/compliance rework across client deliverables."*

5.5.2 EDGE CASES

- Unmapped term submitted → flagged for glossary review, not silently passed.
- Color contrast below WCAG 2.2 AA → blocked at design-token lint.
- Entity claimed by two contexts → registry raises a boundary conflict.

5.5.3 USE CASES

UC-05 Enforce Domain & Brand Integrity: validate terms → enforce brand tokens → check accessibility → guard context boundaries.

5.5.4 'TECHNICAL REQUIREMENTS' CARD

TECH REQ CARD · TR-DM-01

Title: Brand & accessibility conformance gate **Statement:** *All client-facing surfaces shall use approved brand tokens and meet WCAG 2.2 AA contrast; raw hex values and sub-AA contrast shall fail CI.*
Priority: MUST · **Verification:** Design-token lint + automated a11y test (axe)

5.5.5 TECH REQUIREMENTS INDEX + PRIORITY

ID	Requirement	Priority	Method
TR-DM-01	Brand & a11y conformance gate	MUST	XP CI
TR-DM-02	Ubiquitous language validator	MUST	DDD
TR-DM-03	Bounded-context registry	SHOULD	DDD
TR-DM-04	Design token system	SHOULD	Design system
TR-DM-05	Brand telemetry/audit	COULD	Lean metrics

Appendix D — Glossary (Ubiquitous Language)

Term	Definition
Trouble-Ticket	Canonical term for a submitted requirement/incident entering the SLA pipeline.
Developer-Ready Spec	A specification with API schema, data model, acceptance criteria, and edge cases — zero open interpretation.
SLA Tier	Advisory (4 h), Partner (1 h), Emergency Owner (15 min) response thresholds.
RTM	Requirements Traceability Matrix linking need → story → test.
NFR	Non-Functional Requirement (performance, scalability, reliability, security, etc.).
ADR	Architecture Decision Record — context, decision, consequences.
Bounded Context	DDD boundary within which a domain model and its terms are consistent.
INVEST	Independent, Negotiable, Valuable, Estimable, Small, Testable.
WSJF	Weighted Shortest Job First — Cost of Delay ÷ Job Size.

Appendix E — References

1. ISO/IEC/IEEE 29148:2018 — *Systems and software engineering — Life cycle processes — Requirements engineering.* IEEE SA. https://standards.ieee.org/standard/29148-2018.html
2. ReqView — *ISO/IEC/IEEE 29148 Requirements Specification Templates (StRS/SyRS/SRS/BRS/OpsCon).* https://www.reqview.com/doc/iso-iec-ieee-29148-templates
3. Schwaber, K. & Sutherland, J. — *The 2020 Scrum Guide.* https://scrumguides.org/scrum-guide.html
4. IIBA — *A Guide to the Business Analysis Body of Knowledge (BABOK® Guide v3).* https://www.iiba.org/
5. Blank, S. — *Jobs-To-Be-Done as the Front End of Customer Discovery.* https://steveblank.com/2021/11/04/market-definition-its-the-front-end-of-customer-discovery/
6. Ulwick, T. — *Jobs to Be Done (JTBD): The Original Framework.* Strategyn. https://strategyn.com/jobs-to-be-done/
7. Fitzpatrick, R. — *The Mom Test* (customer discovery interview method).
8. AltexSoft — *Nonfunctional Requirements: Examples, Types and Best Practices.* https://www.altexsoft.com/blog/non-functional-requirements/

9. TestQuality — *Gherkin User Stories & Acceptance Criteria Guide (2026).* https://testquality.com/gherkin-user-stories-acceptance-criteria-guide
10. Boost / PlatinumEdge — *INVEST Criteria for Agile User Stories.* https://www.boost.co.nz/blog/2021/10/invest-criteria
11. Architecture Decision Records. https://adr.github.io/
12. KanbanZone — *Agile Frameworks Compared: Scrum, Kanban, Lean, XP (2024).* https://kanbanzone.com/2024/agile-frameworks-compared/
13. PPM Express — *From RICE to WSJF: Prioritization Techniques.* https://www.ppm.express/blog/13-prioritization-techniques
14. GetProductPeople — *Prioritization Techniques: RICE, MoSCoW, ICE & Kano.* https://www.getproductpeople.com/blog/prioritization-techniques-rice-moscow-ice-kano
15. HighTower.Digital — *MSP SLA #1+5 — IT Requirements As-A-Service.* https://www.hightower.digital/msp-sla-15

Document control. HDI Confidential · Author: LW Atkinson (CEO/CTO) · Version R1.A · 06/25/2026.