



WASIQ – Nancy AI – User Stories

11/2/2025

Contents

Phase 1 – FOUNDATION & SEMANTIC ENGINE (context engine + memory + intent)	3
US-1.1.1 - Context Persistence (Scope + Data Focus + Drill Path).....	3
US-1.1.2 - Intent Tracking (Goal Preservation)	4
US-1.1.3 - Persist Sessions & Restore Context	4
US-1.1.4 - Context & Intent View Panel	5
US-1.1.5 - Standardize Business Metric Definitions	6
US-1.1.6 - LLM Metric Registry Access.....	7
Phase 1 – VISUAL ANALYTICS	8
US-1.2.1 - Determine chart type (auto chart selection).....	8
US-1.2.2 - Render chart in UI.....	8
US-1.2.3 - Generate contextual drill-down suggestions.....	9
US-1.2.4 – Apply drill-down action	10
Phase 1- OUTPUT/ARTIFACTS	14
US-1.3.1 - Generate export payload	14
Phase 1 - Off-topic Decline & Guardrails	14
US-1.4.1 - Detect off-topic prompts.....	14
US-1.4.2 - Guardrail Enforcement: Prevent Hallucinated Fields / Invalid Query Columns ..	14
Phase 1 – DATA OPTIMIZATION.....	15

Phase 1 – FOUNDATION & SEMANTIC ENGINE (context engine + memory + intent)

US-1.1.1 - Context Persistence (Scope + Data Focus + Drill Path)

As a user, I want the AI to remember what we are currently analyzing - including the dataset, what entity we're focused on (such as transaction type, merchant, MCC, Sort Code), timeframe, and drill-down path - so I can ask natural follow-up questions without restating details.

Acceptance Criteria

Category	Criteria
Context Memory	1. System must store the following as conversational state: <ul style="list-style-type: none">• Data Domain: Settlement / Authorization• Entity Focus: merchant, MCC, BIN, processor, Sort Code• Granularity: daily/weekly/monthly grouping• Time Window: last 30/60/90 days, YTD• Drill Path: parent → child transitions
Natural-Language Follow-ups	2. User can say: “What about Discover?”, “Break that down” or “Show trend instead” and the AI continues the same analysis correctly using context.
Analytical Continuity	3. Context includes the current step of the analysis , e.g.: Declined Auth Spike → MCC 6012 → Merchant ABC
Context View	4. UI must display the active analysis context as a breadcrumb chip bar , e.g.: Authorization > Declined Auth Spike > MCC 6012 > Merchant ABC > Last 90 Days
Context Update Rules	5. When user changes context (“Switch to Settlement dataset”), context is overwritten.
Reset Rules	6. Saying: “Start over” / “Clear context” / “New analysis” resets state entirely.
Ambiguity Handling	7. If context becomes unclear, AI can ask a clarification: <i>“Are you referring to the dispute dataset or the authorization dataset?”</i>
Performance	8. Context retrieval must be < 200 ms.
Security	9. Context must not store sensitive data (PAN, cardholder, bank acct numbers).

US-1.1.2 - Intent Tracking (Goal Preservation)

As a user, I want the AI to remember **my analytical goal**, so that follow-up questions continue toward the same outcome without me repeating the goal.

Example:

User: "Why are disputes increasing for Merchant ABC?"

Later: "Break it down by MCC."

AI must **retain user intent** to identify root cause of increasing disputes, not generate general data.

Acceptance Criteria

Category	Criteria
Intent Storage	1. System identifies & stores analytical intent (root cause analysis, trend comparison, performance monitoring, etc.).
Goal Continuity	2. Follow-up questions continue toward that goal until user changes intent.
UI Transparency	3. UI displays the intent as part of the session context: Goal: Identify root cause of dispute increase
Natural Behavior	4. User can ask: "Go deeper", "show contributing MCCs", "what's driving it?" and AI knows the goal.
Re-alignment	5. If conflicting intent appears, system asks for clarification: "Would you like to change the goal to revenue optimization instead of dispute root cause?"
Reset	6. Saying "New goal" clears intent but keeps dataset.

US-1.1.3 - Persist Sessions & Restore Context

As a user, I want the system to save all sessions so when I return, the AI resumes context + intent + analysis path.

Acceptance Criteria

Category	Criteria
Stored session items	1. The following values must be saved per session: <ul style="list-style-type: none">• conversation history• active dataset (Settlement/Dispute/Auth)• context (entity, timeframe, path)• intent (goal)
Resume behavior	2. Upon resume, AI loads the last known context & intent automatically.
UI Transparency	3. User sees a session summary: "We were analyzing Dispute Spikes for MCC."
Limits	4. Should have no limits of save sessions
Performance	5. Restore time < 1 sec.

US-1.1.4 - Context & Intent View Panel

As a user, I want to inspect & modify what context the AI is currently using so I feel in control of the analysis.

Acceptance Criteria

Category	Criteria
UI	1. UI displays: <Data Domain> > <Focus Entity> > <Drill Path>
Edit	2. User can hover to remove part of path (e.g., delete MCC, keep only Merchant).
UI Transparency	3. UI displays the intent as part of the session context: Goal: Identify root cause of dispute increase
Change Intent	3. User can click "Change Goal" and select from predefined intents: <ul style="list-style-type: none">• root cause analysis• trend monitoring• comparison / benchmarking

US-1.1.5 - Standardize Business Metric Definitions

As a product/data team, we need a governed catalog of **business metrics** (formulas, inputs, allowed grains, constraints, versions) so that all users - and the AI agent - produce **consistent** analytics across Settlement, and Authorization domains.

Business Value

- Single source of truth → consistent answers
- Faster onboarding → less ambiguity

Functional Requirements

1. Registry storage

Store metric definitions in a registry (e.g., JSON docs) with version control.

2. Identity & naming

Each metric has a unique metricId and a human-friendly displayName.

3. Calculation spec

Definition includes either:

- a) atomic measure mapping (e.g., a field/aggregation), or
- b) derived formula that references other metrics/fields.

4. Context constraints

Allowed time grains (day/week/month), grouping dimensions (MCC, Merchant, BIN...), default time window, and disallowed combinations (e.g., cannot group by PAN).

5. Data lineage & source

Source entity (e.g., AllRepaySettlement), column bindings.

6. Status lifecycle

Draft → active → deprecated; only active are served to LLM for P1.

Example of metrics

1. Atomic metrics

- Settled_amount
 - Refund_amount
 - Credit_settled_amount
 - Debit_settled_amount
 - Credit_refund_amount
 - Debit_refund_amount
2. Derived metrics
 - Refund_rate
 - Approved_auth_rate
 - Declined_auth_rate

US-1.1.6 - LLM Metric Registry Access

As the AI orchestrator, I must **retrieve and apply** the official metric definition (from the registry) before planning/executing any query, so that calculations are consistent, safe, and within allowed constraints.

Business Value

Prevents formula drift and hallucinations; guarantees trust in numbers.

Functional Requirements

1. Lookup first

On each request containing a pre-defined business metric (e.g., “Approval Rate”), the orchestrator calls the corresponding id of the metrics.

If not found, return helpful alternatives (fuzzy suggestions) and do not generate SQL.

2. Constraint enforcement

Planner must respect allowedGrains, allowedGroupBys, filters, edgeCases, and resultFormat.

If user asks for a disallowed combo, return a suggestion (e.g., “Try grouping by MCC or Merchant”).

3. Provenance

Each result includes metricId, version, and source.entity in the query context the UI can display.

4. Fallback behavior

If registry is temporarily unavailable, return a bounded, safe error (“Metrics temporarily unavailable. Please retry.”). No ad-hoc formulas.

Phase 1 – VISUAL ANALYTICS

US-1.2.1 - Determine chart type (auto chart selection)

As AI, I want to select the correct chart type based on dataset.

Acceptance Criteria

Category	Criteria
Functional	<ol style="list-style-type: none">1. Time series → line chart.2. Category comparison → bar chart.3. KPI-only metric → KPI card.
UX	<ol style="list-style-type: none">4. Chart title auto generated (e.g., “Refund Amount by Month”).5. Legend auto populated.
Error Handling	<ol style="list-style-type: none">6. If AI selects unsupported chart type, fallback to table view.
Performance	<ol style="list-style-type: none">7. Chart rendering must not block UI thread.

US-1.2.2 - Render chart in UI

As a user, I want the chart displayed clearly so I can analyze visually.

Acceptance Criteria

Category	Criteria
Functional	<ol style="list-style-type: none">1. Chart renders with dataset passed from LLM.2. Hover tooltip shows dimension + value + format (currency/percent).3. Data table version is always accessible.
UX	<ol style="list-style-type: none">4. Chart supports full-screen expand.5. Users can toggle between Chart / Table / Summary view.
Error Handling	<ol style="list-style-type: none">6. If no data is returned, show: “<i>No data available for selected filters.</i>”

Performance	7. Initial chart must be rendered within < 5 seconds . 8. Lazy-load rendering for large datasets.
Security	9. No sensitive data displayed (no PAN, tokenized PAN).

US-1.2.3 - Generate contextual drill-down suggestions

As a user, I want the AI to propose 3–5 valid drill-down suggestions (e.g., by MCC, Merchant, Processor, Network, BIN, Region, Time grain) based on the current result (domain, dimensions present, metrics, anomalies) so I can explore the insight further without crafting complex prompts.

Business Value

Reduces user effort, guides consistent analysis paths, prevents invalid queries, and increases discoverability of insights.

In Scope

- Generating suggestion objects for the **current answer step**
- Validating suggestions against semantic layer (allowed dimensions/filters/time grains)
- Ranking suggestions by relevance (e.g., anomaly contribution, coverage, business priority)

Functional Requirements

1. Context-aware generation

Suggestions must be derived from: current domain (Settlement/Dispute/Auth), active dimensions/filters, time window, intent (e.g., root cause), recent anomalies (spikes, deltas), and allowed drillable dimensions from the semantic model.

2. Validity check

Each suggestion must pass schema validation (dimension exists, data type compatible, role valid at current grain).

3. Cardinality & caps

Return 3–5 suggestions. If fewer than 3 valid options exist, return all valid ones with a note “Limited drill-downs available.”

4. Ranking

Rank by: (a) anomaly contribution potential, (b) business relevance priority (MCC, Merchant, Processor, Network), (c) coverage (how much data is explainable by that split), (d) freshness (recent changes).

5. Explainability

Each suggestion must include a short rationale (≤ 120 chars), e.g., “MCC 6012 contributed 38% of increase.”

6. Idempotence

Re-generating suggestions for the same step should be stable unless the underlying data or context changed.

US-1.2.4 – Apply drill-down action

As a user, when I click a drill-down suggestion chip, the system should execute the associated query plan (respecting current **context + intent**), render the new result (chart + table + summary), and append it as the next step in the same conversation with an updated **drill path**.

Business Value

One-click continuity speeds analysis, maintains provenance, and prevents users from writing complex prompts.

In Scope

- Converting suggestion → safe query plan
- Executing with inherited context (filters/time window/intent)
- Updating UI (chart/table/summary) and breadcrumb
- Appending to conversation history

Functional Requirements

1. Inheritance

The action inherits: domain, active filters, time window, grain, and intent from the prior turn.

2. Query Plan Creation

Suggestion parameters (e.g., dimension:"mcc", topK:10) are translated into a validated query plan using the semantic layer (group by MCC, measure = selected metric, limit top 10 by metric).

3. Execution

Execute server-side aggregation (fabric/synapse/cosmos materialized views).

Enforce guardrails: allowed fields, allowed grains, max points.

4. UI Update

Render chart + table + 1–3 Key Insights bullets.

Update breadcrumb: e.g., Settlement > Refund Spike > by MCC.

Preserve prior step above; new step appended below.

5. Undo / Back

Provide “Back to previous step” control (client-side cached result).

6. Provenance

Store the plan summary (dimension, metric, filters) in the conversation state for export/audit.

US-1.2.5 – Load last 10 sessions

As a returning user, I want to see my most recent chat sessions (up to 10) with key context (title, last activity, goal/intent, domain), so I can quickly resume my work without hunting through history.

Business Value:

Reduces time-to-resume, promotes continuity, and increases adoption.

Acceptance Criteria

Category	Criteria
Functional	<ol style="list-style-type: none"> 1. On load of the Home screen, the app fetches the 10 most recent sessions for the authenticated user ordered by the last modified date/timestamp. 2. Each session card shows: <ul style="list-style-type: none"> • Session title (first user prompt truncated to 80 chars) • Last activity timestamp (relative + absolute on hover) • Domain(s): Settlement / Authorization • Current goal/intent (if present) • Drill path breadcrumb (last 1–2 hops, truncated) • Session length (message count) 3. Clicking a session opens it and restores conversation, context, intent, drill path. 4. If the user has no saved sessions, show an empty state with 3 starter prompts.
UX	<ol style="list-style-type: none"> 5. Display as a list or grid of cards with sorting by Last Activity (default) and Title. 6. Provide a “View All” capability when more than 10 exist. 7. Cards support keyboard navigation; pressing Enter opens the session.
Error Handling	If the fetch fails, show non-technical message: “We couldn’t load your recent sessions. Try again.” with Retry.
Performance	
Security	Data must be scoped per user/tenant; no cross-tenant exposure.

US-1.2.6 – Search sessions

As a user, I want to search my past sessions by keywords, intent/goal, domain (Settlement/Authorization), and date range, so I can quickly find and reopen a specific analysis thread.

Acceptance Criteria

Category	Criteria
Functional	<ol style="list-style-type: none"> 1. Search inputs: <ul style="list-style-type: none"> • Free-text keyword (matches title and lightweight message index/summary) • Sort by: Relevance (default), Last Activity, Title 2. Results return: sessionId, title, lastActivity, domain(s), intent, snippet (matched context), and message count. 3. Pagination supported (page size 20, cursor/offset). 4. Clicking a result opens the session and restores context/intent/drill path.
UX	<ol style="list-style-type: none"> 1. Search box with placeholder “Search sessions (title, intent, or keyword)”. 2. Facets (chips or sidebar): Domain, Intent, Date Range, Has Export. 3. Highlight matched keywords in title/snippet. 4. “Clear filters” control returns to default state.
Error Handling	No results → show helpful message and quick filters (e.g., expand date range).
Performance	
Security	<ol style="list-style-type: none"> 1. All queries enforce user/tenant scope and role checks. 2. Search results must not leak sensitive content (no PAN/PII; snippets are from summaries with masking).

Phase 1 – OUTPUT/ARTIFACTS

US-1.3.1 - Generate export payload

Acceptance Criteria

Category	Criteria
Functional	1. Collects prompts, responses, charts (as images), insight summaries.
UX	2. User sees preview screen with page breaks.
Error Handling	3. If missing chart image, PDF replaces with placeholder text.
Performance	4. Export start → finish < 8s.

Phase 1 - Off-topic Decline & Guardrails

US-1.4.1 - Detect off-topic prompts

Examples of off-topic prompts: weather, movies, relationship advice.

Acceptance Criteria

Category	Criteria
Functional	1. If prompt contains irrelevant domain intent, classify as out-of-scope.
UX	2. Response template: <i>"This workspace only answers questions about Settlement/Authorization/Dispute/ACH.. analytics."</i>
Error Handling	3. Log off-topic prompts as "intent=declined".

US-1.4.2 - Guardrail Enforcement: Prevent Hallucinated Fields / Invalid Query Columns

As a user,

I want the AI to only query valid fields from the approved WASIQ data schema (Settlement,

Dispute, Authorization, ACH, Authorizations), so that the AI never fabricates (“hallucinates”) field names or executes invalid queries.

When a user requests a field, metric, or data element that does not exist in the schema, the system must stop the query and instead return guidance with valid field options.

This prevents incorrect analysis, build failures, and user frustration.

It ensures that the conversational AI agent behaves like a governed analytics solution, not a hallucinating chatbot.

Functional Requirements - Behavior

1. Before generating any SQL/DAX/query, the agent must perform a schema validation step.
2. The schema is retrieved from an approved metadata source (semantic model, Cosmos metadata collection, etc.).
3. If the prompt includes a field/metric that does not exist in the schema:
 - o The system must stop query generation.
 - o The AI must respond with clarification including closest matching valid field names (fuzzy matching allowed).
4. The agent is explicitly prohibited from inventing fields, tables, formulas, or metrics.
5. The agent must always use business-friendly field names from the semantic layer when responding in natural language.

Phase 1 – DATA OPTIMIZATION

US-1.5.1 - Partitioning Improvement (reduce cross-partition scans)

Partition Cosmos data for analytical workloads.

US-1.5.2 - Aggregated / Materialized Views (Daily Merchant Summary)

Daily pre-aggregated data for fast trend queries.

US-1.5.3 - Sensitive Data Masking (PAN/tokenized fields)

Apply compliance masking before LLM ever sees the data.

Phase 1 – USER MANAGEMENT AND AUTHENTICATION

US-1.6.1 - Sign in to AI Chat using Microsoft Entra ID (SSO)

As a WASIQ user, I want to sign in to the AI chat tool using my Microsoft Entra ID (Azure AD) account, so I can access the application securely with my company credentials and have my permissions (roles / data access) applied automatically.

Assumptions / Notes

- The chat tool is a confidential client (web app) registered in Entra ID with redirect URIs set for PROD/QA/STAG.
- Session is cookie-based; tokens are stored server-side or in httpOnly secure cookies.
- RLS/permissions are derived from Entra ID (roles, groups, tenant, user objectId).

Acceptance Criteria

A. Sign-in & SSO

1. Given an unauthenticated user visits any link
When no valid session exists
Then the app redirects to Entra ID for sign-in and, after success, returns to the original URL (relay state preserved).
2. Given a user is already signed into Microsoft 365 in the same browser
When they open the chat tool
Then Seamless SSO occurs without re-entering credentials (subject to tenant Conditional Access).
3. Given a user is not assigned to the enterprise app (license/assignment missing)
When they attempt sign-in
Then the app shows “Access not granted. Please request access.” and does not create a session.

B. Tokens, Session & Logout

4. On successful sign-in, the app establishes a server-side session bound to the user and stores: userId (OID), tenantId, displayName, email/UPN, and role/group claims needed for authorization.
5. ID token is validated (issuer, audience, signature, nonce). Access token is acquired for backend/API; refresh via on-behalf-of/refresh token as configured. No tokens are stored in localStorage/sessionStorage.
6. Idle timeout: session auto-signs-out after 30 minutes of inactivity; absolute timeout at 8 hours. Both values configurable per environment.

7. Logout: clicking “Sign out” clears the session and calls the Entra front-channel logout endpoint; user is redirected to the public landing page (/login).