**Conditions Portal — Business Requirements Document (BRD)**

**1) Purpose & Vision**

Enable processors and underwriters to **identify, request, track, and clear loan conditions** inside Byte via a Clear-styled, embedded experience. Reduce swivel-chair work across Byte, Clear, and Clear Docs; automate repetitive adds; and make cross-role communication auditable and transparent.

**2) In-Scope (MVP)**

* Embedded **Conditions Portal** page in Byte shell (iframe / embedded app).
* **Two primary roles:** Processor and Underwriter, each with tailored quick filters and actions; **Active** filter is role-aware.
* **Automated Conditions Engine** runs (a) before initial processor intake and (b) on handoff to underwriting; underwriter can also rerun and reconcile adds/removes. Not AUS; uses the same rules engine family as disclosures.
* **Add Conditions** (standard list, “My Lists,” and Custom Conditions/templates).
* **Borrower Request workflow**: queue requested docs, generate/edit borrower-friendly descriptions, assemble/send email; notifications are **gated** until processor/LO sends.
* **Document Review wizard**: accept/reject, **split**, **merge**, re-associate, annotate (burn-on-save dependent on Byte API).
* **Status progression** per role (e.g., Ready for Underwriting, Cleared/Not Cleared), plus **notes** with unread badges.
* **Loan status update** happens in Byte’s native screen; portal collects the handoff note first, then deep-links the user.

**3) Out of Scope (MVP)**

* Replacing Byte’s **loan status** UI or triggering Byte macros from inside the Portal.
* End-to-end role coverage beyond Processor/Underwriter (e.g., full ops). (Discussed as a broader initiative implication.)

**4) Users, Roles, Permissions**

* **Processor**: sees Processor-centric Active filter; can attach/upload docs, prepare borrower requests, mark conditions *Ready for Underwriting*, update handoff notes.
* **Underwriter**: sees Underwriter-centric Active filter; can add conditions (with automation), clear/not-clear conditions, leave notes, rerun rules & reconcile.
* **Privilege example**: “Add conditions” button visible to Processor Level 3 and up.

**5) Core Flows (Happy Paths)**

**5.1 Processor — Initial Intake**

1. Portal opens within Byte shell; quick filters appear (Active, etc.).
2. System has already run **Automated Conditions** and de-duped AUS docs.
3. Processor attaches existing Clear Docs or uploads new files to a selected condition; enters **Review** to accept & mark **Ready for Underwriting**.
4. Processor clicks **Update Loan Status** → enters note → routed to Byte’s status screen.

**5.2 Underwriter — Add & Curate Conditions**

1. Reads unread **notes**; badge shows count; panel lists condition- and loan-level notes.
2. **Automated Conditions (UW pass)** has run; UW may also **rerun & reconcile** (add/remove suggestions).
3. UW **Add Conditions** screen: category filters, search, multi-select, “My Lists,” and **Custom Conditions** with optional **save as template** (user-level).

**5.3 Processor — Send Borrower Requests**

1. System collates **Borrower-facing** requests for conditions in “Need Borrower Request/New.”
2. Processor edits human-friendly descriptions and **adds to email** (AI-assisted intro allowed). (Per B&L parity; specific AI prompt controls are productized elsewhere.)

**5.4 Document Review Wizard (both roles)**

* Inbound sources: borrower portal bucketed uploads **and** misc bucket; email drag-and-drop; vendor/third-party docs.
* Actions: **Accept**, **Reject** (optionally queue re-request + inactivate doc + add note with link), **Split**, **Merge**, **Re-associate** to different condition, annotate (Byte API “burn-on-save” dependency).
* Status results by role:
  + **Processor**: mark **Ready for Underwriting** (condition-level) without changing loan status.
  + **Underwriter**: **Cleared** or **Not Cleared** (downgrade); “Suspend” pertains to loan status, not the condition (design TBD).

**6) Functional Requirements (by module)**

**6.1 Conditions Grid (Embedded in Byte)**

* Must render inside Byte with Clear styling and columnar grid of conditions.
* Quick filters include **Active** (role-aware rules), plus category/stage/status filters. Display only relevant actions per role.
* Row shows: Code, Title, Description, Borrower, Status, Docs count, Notes (unread badge).

**6.2 Add Conditions**

* **All/By Category** list; search with incremental filter; multi-select; **Add** appends rows immediately.
* **My Lists**: create, name, save sets of conditions for reuse; supports standard + custom items.
* **Custom Conditions** (formerly ad-hoc): full fields (name, long description, category, stage, borrower/internal), optional **document request** pairing, and **Save as Template** (user scope; org-level via Admin only).

**6.3 Automated Conditions (Rules Engine)**

* **Auto-run** on: loan first assigned to Processor; on Processor→UW handoff.
* **Manual run (UW)**: “Run Automated Conditions” → **Reconcile** modal (Add X / Remove Y / Unchanged).
* **Engine** is **not AUS**; it is the disclosures-class engine/tooling; SME: **Kathy Hodgson** for details.

**6.4 Borrower Requests**

* Conditions can **carry a paired document request** so Processors don’t re-author the obvious asks.
* **Queue first** (no borrower notification until Processor/LO sends); maintain edited description for reuse on re-requests.
* **Email build**: assemble selected requests + AI intro; log the outbound message/action.

**6.5 Document Review**

* Source aggregation (borrower portal—assigned & misc), email drag-and-drop directly to a condition, and Clear Docs picker.
* **Split/Merge** UX mirrors Clear Docs capabilities, plus association to one or more conditions (within Byte data model constraints).
* **Accept/Reject**: Accept advances; Reject inactivates doc, adds note (with link), and (optionally) re-queues borrower request.

**6.6 Notes & Badging**

* Condition-level and loan-level **notes**; unread badge shown at top; “View All Notes” panel; mark read/unread.

**6.7 Handoff to Byte Loan Status**

* **Single “Update Loan Status”** CTA opens note modal → routes to Byte status page where macros fire.

**7) Status Model (Condition level)**

* **New** (UW just added) → **Need Borrower Request** (eligible for queue) → **Requested** (email sent) → **Docs Received / In Review** (wizard) → **Ready for Underwriting** (Processor) → **Cleared** (UW) or **Not Cleared** (UW; downgrades back to requested/not-ready). Exact Byte mapping for *Ready for Underwriting* to be finalized; prior mapping referenced “Submitted/Submit to Underwriting.”

Note: **Suspend** applies to the *loan* (not the condition) and needs design/requirements finalization.

**8) Data & Configuration (prototype-ready)**

* **Entities**: Condition, Document, DocumentRequest, Note, Borrower, Loan, ConditionTemplate, MyList, AutomationRun (with add/remove deltas). (Ref. demo data model needs for unread flags, etc.)
* **Keys**: Condition has code (ex: APP-100, ASSET-500, CREDIT-315), title, description, category (Assets, Income, Credit, Title, Appraisal, Program e.g., FHA/VA/Conv/USDA), stage, assigneeRole, status, pairedRequestId, notes[], docs[]. (Codes surfaced in walkthrough examples.)

**9) Baseline Conditions Library (taxonomy & grouping)**

Maintain **separate tabs/groups** and keep them distinct in UI and test plans:

* **Program**: FHA, VA, Conventional, USDA
* **Cross-cutting**: **Assets**, **Income** (generally required on all loans), **Credit**, **Title**, **Appraisal**, **Identity/KYC**, **Compliance**  
  Attach and reference **Standard Underwriting Conditions.xlsx** as the source of truth; surface these groups as **left-rail filters** and as **“category” chips** in Add Conditions. (Per your direction; file contains the standard set to be applied at runtime by rules where applicable.)