AugmentRisk – Payments Workflow POC

Requirements Definition

Contents

[Overview 3](#_Toc211945458)

[Basic payment flow 3](#_Toc211945459)

[In Scope 4](#_Toc211945460)

[End-to-end walkthrough of the POC 4](#_Toc211945461)

[Menu structure 5](#_Toc211945462)

[Screens and data fields 5](#_Toc211945463)

[Manual Payment (Adhoc) Capture screen (Must Have) 5](#_Toc211945464)

[ERP Inbound queue screen (Must Have) 7](#_Toc211945465)

[Add Beneficiary screen (Optional) 8](#_Toc211945466)

[My Approvals screen (Must Have) 10](#_Toc211945467)

[Payment Release Queue screen (Must Have) 13](#_Toc211945468)

[Bank Acknowledgement screen (Combine with Bank Receipts) 14](#_Toc211945469)

[Bank Receipts screen (add Bank ACKs) - (Must Have) 15](#_Toc211945470)

[Audit Log screen (Optional) 17](#_Toc211945471)

[Configuration screen (Must Have) 18](#_Toc211945472)

[Home (Dashboard screen) - (Must Have) 18](#_Toc211945473)

[Features 20](#_Toc211945474)

[Duplicate Check 20](#_Toc211945475)

[Account Verification 20](#_Toc211945476)

[Anomaly Detection 21](#_Toc211945477)

[Auto Escalation 21](#_Toc211945478)

[CFO Approval 21](#_Toc211945479)

[Dual Authorisation for Foreign Payments 21](#_Toc211945480)

[User Roles 21](#_Toc211945481)

[Sample Data 23](#_Toc211945482)

[Beneficiaries Data 23](#_Toc211945483)

[Payments Data 23](#_Toc211945484)

[POC Demo Re-run (Must Have) 23](#_Toc211945485)

# Overview

This document outlines the requirements for the AugmentRisk Payments Workflow POC tailored for the UK market. The POC will simulate a payment processing system that supports manual payment capture and ERP integration, duplicate payment check, bank account verification, anomaly detection, approval workflows, audit traceability, and dashboard analytics.

# Basic payment flow

Below is a basic flow that has been provided by the client. This has been used as the based

A diagram of a company

AI-generated content may be incorrect.

# In Scope

* Manual Payment Capture with validation and beneficiary lookup.
* ERP payments feed simulation from inbound queue.
* Beneficiary creation and management.
* Duplicate Check engine with scoring and exception handling.
* Account Verification (CoP / IBAN / BIC).
* Approval workflow with escalation.
* Payment Release (dual auht where applicable).
* Bank Receipt ingestion (simulated) and reconciliation.
* Dashboards, KPIs, and exports (audit and operations).

# End-to-end walkthrough of the POC

The workflow defines the journey from payment creation to reconciliation, including validation, duplicate detection, verification, approval routing, and release.

1. Payment originates via manual capture or ERP feed (simulated import).
2. System performs duplicate checks using beneficiary, amount, date, and reference criteria.
3. If duplicate is detected, user must confirm non-duplicate or send to exceptions queue.
4. Account verification executes (confirmation of payee (CoP) for UK, IBAN/BIC check for foreign).
5. Workflow routes payments based on rules: amount thresholds, currency, and country.
6. Approval routing: Level 1, Level 2, and CFO (if > £500,000).
7. Foreign payments always require dual authorisation.
8. Once approved, release officer executes payment (bank API integration simulation).
9. Bank receipt ingested (Accepted, Rejected, or Partial) – simulated import.
10. Receipt Reconciliation updates payment status and dashboard KPIs.

# Menu structure

The POC application will have the following main menu sections:

* Home (Dashboard screen)
* Payments
  + Manual capture (Adhoc)
  + ERP Inbound Queue (simulated feed)
* Beneficiaries
  + Add Beneficiaries
* My Approvals
* Payment Release Queue
* Bank Receipts (Bank Acks)
* Audit Logs
* Configuration

# Screens and data fields

## Manual Payment (Adhoc) Capture screen (Must Have)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Required | Control | Source/Default | Validation / Notes |
| Payment Reference | String (50) | Yes | Text | User | Unique value. |
| Invoice Number | String (50) | Yes | Text | User | Unique value. |
| Payment Date | Date | Yes | Date Picker | Today | DD/MM/YYYY. |
| Amount | Decimal(18,2) | Yes | Numeric | User | > 0. |
| Currency | ISO 4217 | Yes | Dropdown | GBP | Lock to GBP for domestic; allow others for foreign. |
| Domestic/Foreign | Enum | Yes | Dropdown | Domestic | Switch toggles form fields. |
| Beneficiary | Entity Ref | No | Dropdown (searchable) | Beneficiary Master | Optional. |
| Beneficiary Account Name | String(120) | Yes | Text | User | Editable if allowed. |
| Sort Code | String(6) | Yes | Text | User | NN-NN-NN or NNNNNN; UK format. |
| Account Number | String(8) | Yes | Text | User | 8 digits. |
| IBAN | String(34) | No | Text | User | Required if Foreign. |
| BIC/SWIFT | String(11) | No | Text | User | Required if Foreign. |
| Payment Method | Enum | Yes | Dropdown | Faster Payments | Faster Payments, BACS, CHAPS, SWIFT. |
| Remittance Advice | String(240) | No | Text | User | Shown to beneficiary. |
| Cost Centre | String(20) | No | Dropdown | Ref Data | Optional for POC. |
| GL Code | String(20) | No | Dropdown | Ref Data | Optional. |
| Reason Code | Enum | Yes | Dropdown | Ref Data | E.g., Supplier invoice, Refund, Payroll, Adhoc. |

* Actions:
  + Run payments duplicate check (auto on submit).
  + Run account verification (auto on submit).
  + Submit for Approval.
* Validation:
  + CoP for UK (match name)
  + IBAN checksum for foreign payments
  + Sort code and Account check
  + Amount > 0
  + Required fields present
* Rules shown:
  + Out-of-range warning if > £1,000,000
  + If foreign payment then dual approval banner must be shown.
* Each payment must have a payment details screen showing all the details of the selected payment. The payment must have a payment status showing where in the process the payment is (Pending Approval, Pending Release, Released, Pending Acknowledgement, Pending Receipt, Settled)

## ERP Inbound queue screen (Must Have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Source / Default | Validation / Notes |
| ERP Ref | String (50) | Yes | Generated random reference number | Must be unique per inbound record; duplicates rejected. |
| Beneficiary Name | String (120) | Yes | From Beneficiary Master lookup | Must match an existing active Beneficiary |
| Amount | Decimal (18,2) | Yes | From ERP feed or manual entry | Must be > 0; currency format GBP for domestic or foreign currency per Beneficiary. |
| Currency | ISO 4217 Code (3 chars) | Yes | From ERP feed / Beneficiary default | Must be valid ISO currency code (e.g. GBP, EUR, USD). |
| Method (Payment Type) | Enum (Faster Payments / BACS / CHAPS / SWIFT) | Yes | From ERP feed or selected from dropdown | Valid methods depend on Payment Type – SWIFT only for foreign; others for domestic. |
| Status | Enum (Received / Validated / Error / Submitted / Processed) | Yes | System-derived | Updates automatically as payment progresses through validation and workflow. |
| Created | DateTime (DD/MM/YYYY HH:MM) | Yes | System-generated | Read-only audit field; used for ageing and queue filters. |

* Grid of inbound payments with columns:
  + ERP Ref
  + Beneficiary name
  + Amount
  + Currency
  + Method (payment type)
  + Status
  + Created
* Row actions:
  + Submit to Workflow
* Actions:
  + Add new ERP payment. Allows the user to capture a new ERP payment.
  + On save – payment must be ready in grid for user to add to the workflow.
* Each payment must have a payment details screen showing all the details of the selected payment. The payment must have a payment status showing where in the process the payment is (Pending Approval, Pending Release, Released, Pending Acknowledgement, Pending Receipt, Settled, etc)

## Add Beneficiary screen (Optional)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Source / Default | Validation / Notes |
| Beneficiary Name | String (140) | Yes | User input | Must be legal entity or individual name; used for Confirmation of Payee (CoP) match. |
| Sort Code | String (6) | Yes (for Domestic) | User input | UK format NN-NN-NN or NNNNNN; required for GBP payments. |
| Account Number | String (8) | Yes (for Domestic) | User input | Must be 8 digits |
| IBAN | String (34) | Yes (for Foreign) | User input | Must pass IBAN checksum; first two letters = country code. |
| BIC / SWIFT | String (11) | Yes (for Foreign) | User input | Must be 8 or 11 characters (A–Z, 0–9); validates bank code. |
| Currency | ISO 4217 Code (3 chars) | Yes | User input | Must be valid ISO code (e.g. GBP, EUR, USD); default to Beneficiary country currency. |
| Country | ISO 3166 Code (2 chars) | Yes | User selection / auto from IBAN prefix | Used to classify Domestic vs Foreign and drive dual auth rule. |
| Active Status | Boolean (Toggle) | Yes | Default = Active | Inactive beneficiaries hidden from dropdown lists. |
| Address Line 1 / City / Postcode | Strings | Optional | User input |  |
| Contact Email | String (120) | Optional | User input | Must be valid email format; used for remittance advice. |
| Verification Status | Enum (Pending / Passed / Failed) | Yes | System-generated (CoP / IBAN check) | Read-only; updated when account verified. |
| Verification Note | String (250) | No | System generated | See UI behavior below |
| Created / Modified By | String (User ID) | Yes | System-generated | For audit trail |
| Created Date | DateTime | Yes | System-generated | Timestamp for audit and reporting. |

* Actions:
  + On Save - Run duplicate check
    - Check for duplicates on account details:
      * Account number
      * IBAN
      * BIC/SWIFT
  + On Save - Run account verification check
    - Executes CoP or IBAN check
* Rules:
  + If Country = GB or IBAN prefix – GB then Domestic, else Foreign
  + If Country != GB, then Foreign, mark beneficiary as needing dual authorization.
  + Currency must match beneficiary country (GBP–>GB, EUR–>EU, USD–>US)
* UI behavior:
  + If the duplicate check fails, mark beneficiary as inactive and set verification note to “Possible duplicate beneficiary”.
  + Account verification:
    - Pending – CoP/IBAN check not yet passed.
    - Failed – cannot activate until corrected.
    - Passed – ready for use in workflow.

## My Approvals screen (Must Have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Source / Default | Validation / Notes |
| Payment Ref | String (50) | Yes | Populated from payment detail | Unique reference per payment; links to Payment Detail screen. |
| Beneficiary Name | String (120) | Yes | Populated from payment detail | Read-only display field. |
| Amount | Decimal (18,2) | Yes | Populated from payment detail | Used to drive approval routing logic (CFO threshold, out-of-range flag). |
| Currency | ISO 4217 Code (3 chars) | Yes | Populated from payment detail | Display only. |
| Payment Type | Enum (Domestic / Foreign) | Yes | Populated from payment detail | Used to enforce dual auth rule for Foreign payments. |
| Status | Enum (Pending / Approved / Rejected / Escalated / Released) | Yes | System-managed | Updates dynamically based on workflow stage. |
| Created By | String (User ID / Name) | Yes | From Payment Capture | Read only |
| Created Date | DateTime | Yes | System timestamp | Used for SLA ageing & escalation triggers. |
| Verification Status | Enum (Passed / Failed / Pending / Inconclusive) | Yes | From Account Verification service | Read-only; informs approver decision. |
| Duplicate Flag | Boolean | Yes | From Duplicate Check logic | “True” indicates potential duplicate; visual badge in UI |
| Out-of-Range Flag | Boolean | Yes | System-derived | Highlight if amount > £1,000,000. |
| Comments / Notes | Text | Optional | User input | Approver remarks (required on rejection). |
| Approval Deadline | DateTime | Yes | System-calculated (Created + SLA hours) | Used for auto-escalation if not actioned within SLA (2h). |

* Actions:
  + - Multi-select for bulk approval.
    - Approval of a payment moves the payment to the release queue.
    - Rejection, send payment back to ERP queue, or back to payment initiator.
* Rules
  + - If logged in user is the CFO, show all payments for approval
    - If logged in user is CFO, provide toggle to show only >500 000 payments
    - If logged in user is NOT the CFO, show all payments, but don’t allow approval of payments > 500 000.
    - If any payment is >1 000 000, highlight payment. Payment is only allowed to be approved by CFO.
    - If logged in user is CFO and payment is > 1 000 000, on approval show modal confirming the payment.
* UI behavior
  + - Colour coding of payments according to status:
      * + Green – Approved
        + Orange - Pending within SLA
        + Red – Pending outside of SLA (Overdue)
        + Grayed out – Rejected
    - Filter options:
      * + All payments
        + CFO – High Value (>500k)
        + Foreign Payments
        + Overdue
        + Pending Approval (Default)

## Payment Release Queue screen (Must Have)

This screen contains all the fields from the My Approvals screen with the below additional fields:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Source / Default | Validation / Notes |
| Release Status | Enum (Pending / Released / Failed / Rejected) | Yes | System-managed | Updates dynamically after release. |
| Released By | String | Optional | Populated when payment released | Read-only after release. |
| Released Date | DateTime | Optional | Populated when payment released | Read-only after release. |
| Release Batch ID | String (50) | Optional | Auto-generated per release batch | Used for export and reconciliation. |

* Actions:
  + Click on Payment Ref opens the full details of the payment and all the history of the approvals, notes (if available).
  + Multi select to create a batch payment release
  + Release – send payments to bank via API (simulated) and updates the Release status.
  + Release All – groups all payments into a single batch and does the release, updates the Release Status.
  + Reject – send payment back to approver.
  + Export Release – creates a CSV file with all payments and details.
* Filter:
  + Payment type
  + Status
  + Date
  + Amount
  + Beneficiary
  + Default filter – Pending release
* UI behavior:
  + Release confirmation
    - When Release or Release All is clicked display a modal that shows:
    - “You are about to release payment **PMT-000102** to **ACME LTD** for £600,000.  
      This action cannot be undone. Proceed?”
    - Requires click on **Confirm Release**.
  + When Release or Release All is clicked, and confirmed display a modal that shows:
    - Number of payments released
    - Total amount
    - Batch ID

## Bank Acknowledgement screen (Combine with Bank Receipts)

|  |  |  |
| --- | --- | --- |
| Field | Description | Example / Notes |
| PaymentReference | Unique ID per transaction from payment | PMT00012345 |
| AccountNumber / SortCode / IBAN | Beneficiary account details | Masked for privacy in some ACKs |
| Amount | Payment amount | 5000.00 |
| Currency | Transaction currency | GBP |
| Status | Bank processing status | Accepted, Rejected, Pending |
| ReasonCode / RejectCode | Code explaining a rejection or hold | R01 / R02 / R03 |
| BankResponseMessage | Text description | invalid account, account closed, insufficient funds |
| ReceivedTimestamp | When the bank validated the transaction | 2025-10-21T09:35:12Z |

* For the POC we don’t have to implement the Header or Trailer sections of the ACKs.
* This is only an information screen.
  + No actions required on this screen.

## Bank Receipts screen (add Bank ACKs) - (Must Have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Source / Default | Validation / Notes |
| Receipt ID | String (50) | Yes | Auto-generated | Unique identifier for each receipt record. |
| Bank Ref | String (100) | Yes | From mock bank API | Must be unique per bank message. |
| Payment Ref | String (50) | Yes | Matched against internal payment | Cross-referenced with released payments |
| Beneficiary Name | String (120) | Yes | From matched payment | Read-only. |
| Amount | Decimal (18,2) | Yes | From bank message | Must match released payment |
| Currency | ISO 4217 Code (3 chars) | Yes | From bank message | Must match released payment. |
| Rail / Method | Enum (Faster / BACS / CHAPS / SWIFT) | Yes | From payment release | Used for reporting and grouping. |
| ACK Status | Enum (Accepted, Rejected, Pending) | Yes | From bank message | Reason or notes from bank |
| Status | Enum (ACCEPTED / REJECTED / PENDING / PARTIAL) | Yes | From bank message | Read only |
| Status Reason | String (255) | Optional | From bank message | Explains REJECTED or PARTIAL statuses ( AccountClosed, Insufficient funds). |
| Value Date | Date | Yes | From bank message | Payment settlement date. |
| Received Date | DateTime | Yes | System timestamp on receipt ingestion | Used for ageing. |
| Matched Payment ID | String (UUID / internal ID) | Yes | System-generated on match | Used for automated linking to Payment table. |
| Match Status | Enum (Matched / Unmatched / Partial) | Yes | System-calculated | Read only |

* Actions:
  + Reconciliation:
    - Primary match is on PaymentRef
    - Secondary match is on:
      * Beneficiary name
      * Amount
      * Value date
    - If multiple candidates, mark as “Partial Match”
    - If matched, mark as “Settled”
    - If no match, mark as "Unmatched”
* UI behavior
  + Green – Settled
  + Orange – Partial Match
  + Blue – Matched, but rejected (Account closed, insufficient funds)
  + Red - Unmatched

## Audit Log screen (Optional)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Required | Source / Default | Validation / Notes |
| Batch ID | String (50) | Yes | Auto-generated during batch release | Unique identifier per release batch (e.g. BATCH-00045). |
| Release Date | DateTime | Yes | System-generated at time of release | Timestamp when batch was successfully released. |
| Released By | String (User Name / ID) | Yes | From authenticated session | Identifies the Release Officer responsible. |
| No. of Payments | Integer | Yes | Count of payments in batch | Auto-calculated during batch generation. |
| Total Amount | Decimal (18,2) | Yes | Sum of all payment amounts in batch | Display formatted (e.g., £1,200,000.00). |
| Status | Enum (Released / Failed) | Yes | System-managed | Indicates batch transmission result. |
| Receipt Count | Integer | Optional | Derived from Bank Receipts module | Number of receipts matched to this batch. |
| Download Audit File | Button | Yes | System-generated on release | Enables CSV or PDF export of all payments within batch. |
| Comments / Notes | Text | Optional | User entry | Optional remarks when creating or verifying batch. |

* Actions
  + Select batch to download.
* Filter:
  + Release date
  + Release by
  + Status

## Configuration screen (Must Have)

Screen to set the values of the following:

* CFO Approval Threshold: [Numeric, default 500 000]
* Anomaly Detection Threshold: [Numeric, default 1 000 000]
* Escalation Time Limit: [Numeric, 2h]
* CoP/Account Verification Toggle: [Checkbox]
* Dual Authorisation for Foreign Payments: [Checkbox]

## Home (Dashboard screen) - (Must Have)

Screen to show the KPI metrics for a selected period. Period must be global period for all metrics on the page.

Period values can be:

* Today
* Past 7 days
* Past month
* Past year

**M1. Payments Ready to be Released**

* **Definition:** Payment.status = 'Approved' AND Verification.status = 'Passed' AND DuplicateCheck.status = ‘Passed’ AND Payment.status IS NOT ‘Released’
* **Count:** integer
* **Click-through:** opens *Release Queue* page.

**M2. Payments Requiring Approval**

* **Definition:** Payment.status IN ('Submitted','In Approval') (awaiting any approver incl. CFO)
* **Count:** integer
* **Click-through:** opens My Approvals filtered by user.

**M3. Receipts from Bank**

* **Definition:** receipts ingested for period.
* **Breakdown:** total count of receipts from bank. Can be shown in their different statuses.
* **Click-through:** opens *Bank Receipts* screen.

**M4. Account Verification Stats (Passed, Failed, Outstanding)**

* **Definition:** verification outcomes over Last Week / Last Month / Last Year
* **Aggregation:** counts by outcome per period bucket
* **Graph:** stacked bar (three outcomes), toggle period (Week/Month/Year)

**M5. Anomalies Detected**

1. **Definition:** count of payments with any of:
   * amount > OUT\_OF\_RANGE\_THRESHOLD (default £1,000,000)
   * DuplicateFlag = true
   * Verification.status = 'Failed'
2. **Count:** total anomalies in selected period.

**M6. Escalations Triggered**

* **Definition:** number of escalationevents raised in selected period.
  + Escalation occurs when pending approval exceeds SLA\_HOURS (2 hours)
* **Click-through:** opens My Approvals screen.

**M7. Payments near SLA**

* **Definition:** number of payments where the SLA is < 2h away.
* **Graph:** Bar chart showing:
  + Payments within 30min or less of SLA
  + Payments within 1 hour or less of SLA
  + Payments within 2 hours or less of SLA
* **Click-through:** opens My Approvals screen.

# Features

## Duplicate Check

The system will detect duplicate beneficiaries and payments using exact matching. Exact match checks fields like Beneficiary, Invoice Number, Amount, and Date.

## Account Verification

The system will verify bank details using:

* Confirmation of Payee (CoP): Matches account holder name with bank records (simulated).
  + To simulate the CoP process we need to do the following:
    - If **Beneficiary Name** contains ‘Test’, the verification must fail, all other must pass.
* IBAN Validation: Checks format and country-specific rules.
  + We will simulate this as:
    - If IBAN starts with ‘ZZ’, verification must fail.
    - If IBAN is less than 22 chars, verification must fail.
    - All else must pass.
* Sort Code and Account Number: Validated using modulus checks and bank directory services (simulated)
  + - If the 6-digit **Sort code** ends with ‘00’, the verification must fail, all other must pass.
    - If the 8-digit **account number** ends with an odd number, the verification must fail.
    - If the 8-digit **account number** ends with an even number, the verification will pass.

## Anomaly Detection

Payments above £1,000,000 will be flagged as anomalies, and a warning message should be shown to the user.

## Auto Escalation

Payments not approved within 2 hours (configurable) will be escalated to the next approver or CFO.

For the POC we will make the 2 hours (120 minutes) = 2 minute (just for demo purposes)

This must be configurable from the Configuration screen.

## CFO Approval

Payments above £500,000 cannot bypass CFO approval. The system will enforce this rule and route such payments to the CFO user.

## Dual Authorisation for Foreign Payments

All foreign payments require dual authorisation regardless of amount. Two separate users must approve the payment before that payment will be moved to the release queue.

# User Roles

The following users must be configured on the application:

1. **Super User**

Username: super

Password: 1234

Must have access to everything on the application.

1. **Payments Initiator**

Username: pi

Password: 1234

Must be able to initiate payments (manual capture and ERP queue). No approval or release capability. Has capability to create a new beneficiary.

1. **Approver 1**

Username: a1

Password: 1234

Must be able to see all payments and approve payments. No release capability.

1. **Approver 2**

Username: a2

Password: 1234

Must be able to see all payments and approve payments that have been escalated to them on trigger of payment outside of SLA.

1. **CFO**

Username: cfo

Password: 1234

Must be able to see all payments and approve all payments. Payments > 500 000 must be assigned to the CFO for approval by default and not be assigned to approver 1 or approver 2.

1. **Configurator**

Username: config

Password: 1234

Must have access to change the config values on the configuration screen.

All other users will be able to see the configuration screen and values, but will not have the ability to update them.

1. **Release Officer**

Username: ro

Password: 1234

Must be able to release payments in the release queue. This user will not have the ability to capture or approve any payments, but will be able to see all payments and payments awaiting approval.

# Sample Data

## Beneficiaries Data

## Payments Data

# POC Demo Re-run (Must Have)

How do we make the demo re-runnable?

So that one can revert back to a knows starting point in terms of data (payments) and redo the demo with the same data and scenarios easily and quickly?

Suggestions:

* **Gawie** – create scripts to revert the DB back to the original state. (These scripts could all execute in order at the click of a button on the front-end).
* **Diederick** – can also restore a backup of a DB. Create a screen with just one button to drop current DB and restore a backup of a DB. Melissa did this for the DataShift Demo.