

Panel Defense and AI Tool Appendix

1. WRITTEN PANEL DEFENSE

1.1 The Friction I Picked and Why

I chose F2 — the Manual Itemisation & Math Bottleneck: in PhonePe's group-split flow the planner divides a bill item-by-item by hand and leaves the app to apportion 18% GST and service charge on a calculator. I picked it over discoverability (F1) and awkward reminders (F5) because those are real but solvable with UI, copy and notification changes — they don't need AI. F2 sits at the heart of the flow's core job, splitting fairly, and recurs on almost every use. It is also the one friction whose fix cascades: reading the bill to itemise it (F2) simultaneously resolves proportional tax (F4) and bill-proof (F3).

1.2 The AI Feature Summary

- **Core Mechanism:** Smart Split is an AI-powered extraction layer injected at the expense-entry step. The planner photographs the receipt to initiate the flow (File 4, Screens 1–2).
- **The Input:** A raw bill image paired with manual tap-assignments or unstructured natural language inputs (e.g., "I had the pizza").
- **The Processing:** The OCR pipeline extracts line items, taxes, and the bill total. The allocation model then distributes GST and service charges proportionally based on individual item values.
- **The Output:** A structured, per-person payment request dispatched through PhonePe's existing UPI rails. Receivers are served an itemised proof of their specific share (Screen 5).
- **The Guardrails & Fallbacks: Reconciliation Block:** A hard guardrail blocks the dispatch of any split where the sum of individual shares does not perfectly match the extracted bill total.
 - **Human-in-the-Loop:** All AI-extracted amounts remain manually editable prior to execution.

1.3 Responsible AI Decision Defense

The decision I defend is the sum-of-shares guardrail: a split cannot be sent unless every per-person share adds up exactly to the bill total, and any low-confidence item is flagged first. I made this choice because the feature's entire promise is fairness — a silently wrong amount doesn't just annoy a user, it makes a friend overpay and creates social conflict. The trade-off was friction versus safety: blocking a send adds a step when the AI is unsure. I accepted that, because in payments a fast wrong answer is far worse than a slightly slower correct one. Pairing the guardrail with a no-wait manual fallback means we never trap the user while still refusing to ship a broken split.

1.4 North Star Metric and Kill-Switch Defense

My North Star is the weekly value of bills successfully split-and-settled through Smart Split. I chose it because it is the only metric that proves the whole chain worked — scanned, split fairly, and actually paid — not a vanity count of scans that may never settle. It aligns user value (friends pay without chasing) with business value (real transaction volume and DAU/MAU). I would activate the kill switch if hallucination rate exceeds 2%, dispute rate breaches its threshold, P95 latency passes 8 seconds, or any PII incident occurs — each directly attacks the fairness and trust the North Star depends on.

1.5 The Hardest Question I Anticipate

Hardest question: 'If the AI misreads a bill and a friend overpays, PhonePe has facilitated a wrong money transfer — what is your liability and why ship this?' My answer: we treat that as the primary risk, not an edge case. First, we reduce the chance — the sum-of-shares guardrail blocks any split that doesn't reconcile, and

low-confidence items are flagged before sending. Second, the human stays in control — nothing is auto-sent; the planner reviews an itemised, editable breakdown and every receiver sees the proof before paying. Third, we provide recourse — a two-minute human escalation and a payments-specialist dispute path that can reverse and correct a split. So PhonePe isn't making an unaccountable financial decision for the user; it offers a reviewable suggestion with guardrails, a hard reconciliation check, and a clear dispute route — more controlled than today's manual split, where a mental-maths error has none of those protections.

2. ANTICIPATED TOUGH QUESTIONS

#	Function	Tough question	Your answer
1	Engineering	If the OCR vendor has a global outage, what is your fallback — and will you pay for two AI vendors at once?	A circuit-breaker drops instantly to manual split (no wait) and auto-retries later, so the flow never hard-fails. We start single-vendor to control cost and add a secondary OCR provider only once volume justifies it, with the abstraction layer already built so failover is a config change.
2	Design	How do you stop users blindly trusting a wrong AI split?	The result screen always shows the itemised breakdown and an AI disclaimer, flags low-confidence items in amber, and asks 'Was this split correct?'. The reconciliation guardrail makes obviously-wrong totals impossible to send, so trust is earned through visible proof, not a black box.
3	Legal	Bill photos can contain PII and financial data — how is that compliant?	Images are encrypted in transit and at rest, card digits and names are redacted before processing, the raw image is never posted to the group thread, and retention is limited to the active split. Handling follows applicable RBI and data-protection norms; data is never used for advertising.
4	Finance	A free feature with per-scan AI cost — how is this not a money pit?	Cost is under Rs 0.50 per scan; each settled split drives incremental UPI/wallet transactions and higher DAU/MAU. The value of a retained, more-active user exceeds the per-scan cost, so unit economics are positive even before cross-sell. We cap exposure with the kill switch and per-user rate limits.
5	Product / Business	Splitwise and Paytm exist — why will users switch?	Smart Split lives inside the app where the money already moves, so there's no second app and no manual re-entry. Itemised AI splitting plus automatic tax apportionment beats Paytm's share-based split and Splitwise's manual entry, and settlement happens on PhonePe's rails in one tap.

3. AI TOOLS USAGE APPENDIX

Every AI tool used to research, design and build this submission is logged below.

Tool used	Purpose	Prompt Used	How I reviewed or modified output
Claude (Anthropic, Cowork)	Drafting and structuring the PRD, Responsible-AI, metrics, GTM and panel-defense sections	'Audit PhonePe's group-bill-split flow, pick one friction solvable with AI, and draft the capstone sections for an AI receipt-splitting feature, kept to one flow.'	Reviewed every section against the rubric, kept the single-flow rule, added my own judgment (rejected alternatives, edge cases, uncertainty), corrected figures and verified all numbers.
Figma (MCP / Plugin API)	Building the 9 high-fidelity Smart Split prototype screens	'Create PhonePe-branded mobile screens for the Smart Split flow: entry, scan, processing, OCR-uncertainty, AI output, error, fallback, human-in-loop, confirmation.'	Inspected each frame for layout, branding and copy; re-rendered at 2× and verified captions match the flow.
Make (Integromat)	Building the backend orchestration scenario (#6109702)	'Build a scenario: webhook on bill upload → OCR → parse JSON → iterate items → create a split request per member.'	Verified the module chain and data mapping, confirmed it saved in the team workspace, and captioned it accurately in File 4.
Google Cloud Vision (designed)	The in-product OCR capability the feature relies on	Configured as DOCUMENT_TEXT_DETECTION on the uploaded bill image in the scenario above.	Specified as a dependency; accuracy, latency and cost targets defined in Files 2, 3 and 5; not a content-generation tool for this submission.

Where AI was not used: the friction selection, the metric and guardrail choices, the rejected alternatives, the edge cases, all final figures, and the panel-defense arguments were decided and verified by me.