Google Developer Groups
Cloud • New Delhi

Turgon

Girls Gone Git

Round I

IDEA SUBMISSION

HACKFEST 2.0

Google Developer Groups
Cloud • New Delhi

Turgon

Team Details:

S no.	Team Member Name	Team Member Role	Team Member Mail ID
1.	Diksha Sharma	Team Lead + NLP Engineer	dikshasharma5014@gmail.com
2.	Shubhra Negi	Frontend Developer	shubhranegi27@gmail.com
3.	Misthi Pandey	**Data Engineer + Evaluation**	misthikpandey@gmail.com
4.	Khyati Bajaj	**Backend Developer**	khyatibajaj903@gmail.com

HACKFEST 2.0

Google Developer Groups
Cloud • New Delhi

Turgon

Track Chosen:

Track Chosen: Business Requirements Document (BRD) Generation Agent

Problem Description:

Business requirements are scattered across emails, meetings, Slack threads, and documents. Manually synthesizing them into a BRD takes days, misses critical details, and breaks down entirely when sources contradict each other. Every delayed project, misbuilt feature, and blown budget traces back to this exact problem — bad or missing requirements documentation.

HACKFEST 2.0

Google Developer Groups
Cloud · New Delhi

Turgon

Proposed Solution / Idea Overview:

BRDGen — Requirements Intelligence Pipeline

BRDGen is a 9-stage AI pipeline that ingests emails, meeting transcripts, and Slack messages — and automatically produces a structured, cited, conflict-aware Business Requirements Document.

The core principle: **our LLM never generates requirements. It only renders them.** Every requirement is extracted, verified, and sourced by the pipeline first. The LLM only converts proven data into professional prose. Hallucination is architecturally impossible — the model cannot introduce content that was never extracted from a real source.

The core principle: **our LLM never generates requirements. It only renders them.** Every requirement is extracted, verified, and sourced by the pipeline first. The LLM only converts proven data into professional prose. Hallucination is architecturally impossible — the model cannot introduce content that was never extracted from a real source.

Google Developer Groups
Cloud • New Delhi

Turgon

Innovative Solution:

Three Intelligence Layers That Make Us Different

🔵 **Semantic Equivalence Engine** Detects when different people said the same requirement differently across channels. Merges them into one canonical requirement. Cross-channel corroboration becomes a confidence score.

🔴 **Conflict Detection Layer** Classifies conflict type — numerical, scope, timeline, authority vs. expertise. Weights sources by role and recency. Surfaces both positions with a recommended resolution. We never silently pick one.

🟢 **Provenance Graph** Every BRD sentence traces back to its exact source — email, meeting moment, or Slack message, with sender and timestamp. Fully auditable. One click from claim to original evidence.

Google Developer Groups
Cloud • New Delhi

Tech Stack

Turgon

📥 Data Layer

- Enron Email Dataset (500K emails)
- AMI Meeting Corpus (279 transcripts)
- Synthetic Slack messages
- mailparser + xml.etree

🧠 Intelligence Layer

- MiniLM — semantic similarity
- BART (zero-shot) — noise filtering
- spaCy — NLP & coreference
- BERTopic — project clustering
- Cosine similarity — conflict detection

🔗 Structure & Memory Layer

- networkx — provenance graph
- Timestamped requirement objects — versioning
- Confidence scoring — corroboration tracking

⚙️ Output Layer

- GPT-4o / Claude API — BRD generation
- Streamlit — frontend UI
- AMI ground truth — evaluation/benchmarking

HACKFEST 2.0

Google Developer Groups
Cloud • New Delhi

Turgon

Implementation Plan

Phase 1 📥 Ingestion & Parsing

- Parse Enron emails, AMI transcripts, synthetic Slack
- Normalize all sources into unified schema
- Extract sender, role, timestamp, content

Phase 2 🧠 Intelligence Layer

- Filter noise using zero-shot classification
- Cluster semantically identical requirements
- Detect conflicts & track requirement versions

Phase 3 ✍️ Generation & Citations

- Build provenance graph for every requirement
- Pass verified, structured data to LLM
- Generate cited, section-wise BRD output

Phase 4 📊 UI & Evaluation

- Display BRD with citations on Streamlit
- Show conflict report & BRD Health Score
- Benchmark accuracy against AMI ground truth

Google Developer Groups
Cloud • New Delhi

Turgon

What Makes Us Unique

What Makes BRDGen Different

⚡ **01 — LLM as Renderer, Not Reasoner** The pipeline extracts and verifies every requirement first. The LLM only writes prose. Hallucination is structurally impossible.

🔍 **02 — Typed Conflict Detection** We don't just flag conflicts — we classify them by type, attribute them to specific decision makers, and recommend resolution. No silent picks.

🔗 **03 — Full Provenance Graph** Every BRD sentence traces back to its exact source — email, meeting, or Slack — with sender, timestamp, and channel. Fully auditable.

🎯 **04 — Honest Uncertainty** Vague requirements go to a Parking Lot. Cancelled ones are formally documented. Nothing is silently dropped or silently promoted.

📊 **05 — Benchmarked, Not Just Demoed** We evaluate against AMI ground truth summaries and report real precision/recall numbers — not just a polished happy-path demo.

HACKFEST 2.0

Google Developer Groups
Cloud • New Delhi

Turgon

BEFORE-AFTER

Before BRDGen

- 3–5 days of manual synthesis
- Requirements missed or misunderstood
- Contradictions go unnoticed
- No source trail — "who said this?"
- BRD outdated before it's finished

After BRDGen

- BRD generated in minutes
- Every requirement sourced & cited
- Conflicts flagged and classified
- Full audit trail — one click to origin
- Continuously updatable via natural language

https://cartoondealer.com/image/305049712/messy-office-environment-documents-flying-disarray.html

HACKFEST 2.0

Google Developer Groups
Cloud • New Delhi

Turgon

Future Scope:

🔄 **Real-Time BRD Updates** Live API integration with Gmail, Slack & Fireflies — BRD auto-updates as decisions are made

🧭 **Full Requirement Lifecycle Tracking** Trace every requirement from first mention → development → testing → deployment

🎭 **Stakeholder Sentiment Layer** Detect resistance, enthusiasm & uncertainty from tone — not just content

🌍 **Multilingual Requirement Extraction** Capture decisions in any language, synthesize into one English BRD

🔌 **One-Click JIRA / Notion Export** Push verified requirements directly into your project management tool — zero copy-paste

🛡️ **Role-Based Access & Approval Workflows** Stakeholders approve their own requirements before BRD is finalized — human-in-the-loop built in

"BRDGen today automates what takes days. BRDGen tomorrow becomes the single source of truth for every project requirement, from first conversation to final deployment."

HACKFEST 2.0