Hackathon Brief

0. Context & Scope

This hackathon is about prototyping a solution that turns customer meetings into structured, reusable insights for two main users:

1. **Sales Representatives** who need to remember what was said in previous meetings and follow up effectively.
2. **Product Management (VoC stakeholder)** who need to see patterns across many meetings, such as top pain points and competitor mentions.

We assume the company uses a CRM (for example Salesforce). For the hackathon, you will not integrate with a live CRM. Instead, you will work with mock "Salesforce-like" data that includes:

- Accounts
- Opportunities
- Activities (meetings) with free-text notes

Your solution should show how CRM-shaped data and meeting transcripts can be used to support the two personas, even if everything runs on mocked data.

---

1. Personas

1.1 Persona 1: Sales Representative

Name: Alex Berg
Role: Sales Representative (Account Executive)
Primary responsibility: Progress deals by running effective customer meetings
Primary systems: Salesforce, calendar, video conferencing, email

Goal
When meeting a potential customer, remember what has been said in previous meetings.

Context
Alex runs multiple deals in parallel and speaks to the same customer repeatedly over time.

Core needs

- Record customer meetings with minimal effort

- Recall what was discussed in previous meetings before the next call
- Capture decisions, objections, and next steps without heavy manual note-taking

Key pain points

- Meeting recordings and notes live in different tools
- Important details are forgotten or inconsistently documented
- Writing notes after meetings is time-consuming and often delayed

Success looks like

- Meetings are recorded by default
- Previous conversations are easy to review before a call
- Meeting outcomes are captured accurately without extra work

Inevitable activities
What does the user invariably need to do to achieve the goal?

1. A customer meeting is scheduled.
2. The sales rep prepares for the meeting.
3. The sales rep looks up previous interactions with the customer.
4. The meeting takes place.
5. New information is shared by the customer.
6. The sales rep decides on next steps.
7. The outcome of the meeting is documented.
8. A follow-up action is taken or planned.

1.2 Persona 2: Product Management (VoC)

Name: Priya Nordin
Role: Product Manager (Voice of Customer Stakeholder)
Primary responsibility: Translate customer conversations into revenue-relevant insights
Primary systems: Teams, Salesforce, spreadsheets, presentations, internal docs

Goal
Gain insights from customer meetings that help generate more profitable revenue.

Context
Priya is not present in most customer meetings and depends on recorded conversations and summaries created by sales.

Core needs

- Access customer meeting recordings and derived insights
- Identify recurring pain points, feature requests, and objections

- Understand which competitors are mentioned and in what context

Key pain points

- Customer insights depend on how well sales documented meetings
- Missing or inconsistent recordings reduce confidence in conclusions
- It is hard to defend insights without direct customer language

Success looks like

- Customer meetings are consistently recorded
- Insights are based on actual customer statements, not second-hand summaries
- Themes and competitor mentions can be validated with examples

Inevitable activities (hackathon scope)
What does the user invariably need to do to achieve the goal?

1. Customer meetings take place with sales, without the VoC user present.
2. Customer statements are gathered manually from Salesforce or by talking to sales.
3. Similar statements are grouped into themes.
4. Themes are ranked by frequency.
5. Competitors mentioned in meetings are identified.
6. The context of each competitor mention is captured.
7. A concise summary of themes and competitors is shared.

---

2. High-Impact Suboptimalities

Here we have mapped out where the current workflows are clearly weaker than they could be

2.1 Sales Representative

Inevitable activities

1. A customer meeting is scheduled.
2. The sales rep prepares for the meeting.
3. The sales rep looks up previous interactions with the customer.
4. The meeting takes place.
5. New information is shared by the customer.
6. The sales rep decides on next steps.
7. The outcome of the meeting is documented.
8. A follow-up action is taken or planned.

Suboptimalities

1. **Scattered preparation (2 + 3)**
   - Previous interactions are spread across Salesforce, email, and personal notes.
   - It is hard to quickly see what actually matters for the upcoming meeting.
2. **Context loss during and after the meeting (4 + 5)**
   - The rep is focused on the conversation, not on detailed note-taking.
   - Important pain points, objections, and commitments are often only partially captured or remembered.
3. **Inconsistent and delayed documentation (7)**
   - Notes are written later, if at all, and vary a lot in structure and quality.
   - Free-text notes are hard to reuse for future meetings or analysis.
4. **Manual and fragmented follow-up (6 + 8)**
   - Next steps and action items are not consistently extracted and tracked.
   - Follow-ups rely heavily on the rep’s memory and personal habits.

2.2 Product Management (VoC)

Inevitable activities (hackathon scope)

1. Customer meetings take place without the VoC user present.
2. A set of customer meetings is selected for review.
3. Customer statements from those meetings are reviewed.
4. Similar statements are grouped into themes.
5. Themes are ranked by frequency.
6. Competitors mentioned in meetings are identified.
7. The context of each competitor mention is captured.
8. A concise summary of themes and competitors is shared.

Suboptimalities

1. **Loss of customer data (2)**
   - Sampled feedback is often inconsistent based on quality of the sales rep documentation efforts and therefore not representative.
   - Valueble customer statements are not captured.
2. **Manual review of statements (3)**
   - Reading through sales rep notes is slow and hard to scale.
   - Important mentions of pain points, objections, or requests can be missed.
3. **Unstructured theme creation and ranking (4 + 5)**
   - Themes are often defined informally and not connected to actual counts.
   - It is difficult to say how frequent a theme truly is across meetings.
4. **Inconsistent competitor tracking (6 + 7)**
   - Competitor names and contexts are not systematically captured.
   - It is hard to answer who appears most often, and in which situations.
5. **Time-consuming summary creation (8)**
   - Preparing VoC summaries requires repeated manual synthesis.

    - It is difficult to back up insights with concrete examples when there are no meeting transcripts.

2.3 Coordination Between Sales Rep and Product Manager (PM)

1. Inconsistent meeting capture
    - If meetings are not reliably recorded or transcribed, PM has weak input.
2. No shared structure for notes
    - Reps write notes in free text, PM must reconstruct themes manually.
3. Insights do not flow back to reps
    - Aggregated customer conversations don´t automatically turn into concrete support for reps in their next meetings as this is currently a manual task going into the CRM system.

These coordination gaps mean that the same information is effectively re-created multiple times instead of flowing through a shared pipeline.

---

3. Game-Changing Solution Structure

A shared structure that improves execution of the activities above

3.1 Core Idea

Create a solution where every recorded customer meeting becomes a structured asset that serves:

- **Sales reps:** pre-meeting briefings and post-meeting summaries
- **Product Manager (VoC):** aggregated themes and competitor insights

The same input (meeting transcript or notes) is processed once and reused for both personas.

By making recording the meeting part of the sales rep's core workflow, the solution:

- Reduces dependency on rep discipline
- Improves VoC data quality automatically
- Creates value for both personas from the same data

For the hackathon, this can be simple:

- "Record" can mean: upload audio, attach a transcript, or simulate recording ingestion
- The key is that the system owns the meeting capture, not just the analysis

3.2 Key Data Objects

1. **Meeting**
    - Linked to: account, opportunity, product, date, participants
    - Contains: raw transcript or pasted notes
2. **Extracted insights (per meeting)**
    - Pain points
    - Feature requests
    - Objections
    - Competitors mentioned (with short context)
    - Next steps and action items
    - Short, structured meeting summary
3. **Aggregate views**
    - For a chosen slice (product, time range, etc.):
        - Top pain points (with frequency)
        - Top feature requests (with frequency)
        - Top objections (with frequency)
        - Competitors mentioned (with frequency and example contexts)

For the hackathon, these objects can be backed by the provided mocked "Salesforce-like" data (accounts, opportunities, activities with notes_raw and tags).

3.3 Sales Rep Flow (User 1)

Before the meeting

1. The rep selects an account or opportunity.
2. The system shows a **Briefing Card** with:
    - Last few meetings for this account or opportunity
    - Previously extracted pain points, feature requests, and objections
    - Previously mentioned competitors and their contexts
    - Open action items from earlier meetings

During / after the meeting

3. The meeting is recorded, or a transcript is uploaded into the solution.
4. The system processes the transcript and extracts:
    - Updated pain points, feature requests, and objections
    - New competitor mentions with short context
    - A structured summary and proposed action items
5. The rep receives:
    - A meeting summary to review and adjust if needed
    - A list of action items

    - An optional follow-up email outline

6. The activity is stored in the mocked “Salesforce-like” database, linked to the account and opportunity.

This structure improves preparation, documentation, and follow-up without requiring the rep to write everything manually.

3.4 VoC Flow (User 2)

VoC analysis cycle

1. The VoC user selects a set of meetings to analyze, for example:
    - Product: HID Visitor Management or PKI-as-a-Service
    - Date range (for example last quarter)
    - Industry, region, or deal stage
2. For that set, the system aggregates extracted insights into:
    - Top pain points, with counts
    - Top feature requests, with counts
    - Top objections, with counts
    - Competitors mentioned, with counts
    - Example quotes or snippets for each theme and competitor
3. The system generates a **VoC summary skeleton**, for example:
    - “Top 5 pain points for PIAM opportunities this period”
    - “Top 5 objections and how often they appeared”
    - “Competitors most often mentioned and in what context”
4. The VoC user reviews, adjusts wording, and exports or presents the summary.

This structure reduces manual reading, grouping, and counting, while keeping the human in control of interpretation.

3.5 Coordination Design

The same solution supports both personas:

- **Sales reps** benefit from:
    - Better pre-meeting context
    - Automated summaries
    - Clearer action items
- **VoC stakeholders** benefit from:
    - Consistent meeting capture
    - Structured, reusable insights
    - Faster creation of theme and competitor overviews

3.6 Example User Journeys (suggested)

The flows below are **only** examples of how the 1-day hackathon scope could be designed. You are free to adjust it, challenge it and innovate freely, as long as the solution adhere to the user needs and expected business outcomes.

3.6.1 Sales Rep Journey (User 1 – Alex)

Goal: Prepare for a meeting, run it, and capture outcomes in a structured way that can feed both CRM and VoC.

1. Log in & select account
    - Alex opens the solution.
    - Alex sees a list of accounts/opportunities (from mocked CRM data) and selects one.
2. View past activity timeline
    - The solution shows a timeline of past activities/meetings for that account or opportunity, using the mocked activity data.
    - Each activity shows: date, short title, and a summary of extracted insights (pain points, objections, feature requests, competitor mentions).
3. Prepare for the next meeting
    - Alex clicks "Prepare for next meeting".
    - The solution displays a Briefing Card summarizing:
        - Latest pain points
        - Latest objections
        - Any feature requests
        - Competitors mentioned so far
        - Open action items
4. Record the meeting
    - When the meeting happens, Alex:
        - either records audio and uploads it,
        - or pastes a meeting transcript into the solution.
5. System processes the meeting
    - The solution takes the transcript and automatically extracts:
        - Pain points
        - Feature requests
        - Objections
        - Competitors mentioned, with short context
        - Proposed action items
        - A short, structured meeting summary
6. Human-in-the-loop review
    - Alex reviews the generated summary and action items in the UI.
    - Alex can edit or confirm the content.
7. Push changes to CRM (mocked)
    - After review, Alex clicks "Save to CRM".

    - The solution writes the updated activity back into the mocked "Salesforce-like" data model (for example as a new Activity with structured fields + summary).
8. Optional: generate follow-up email
    - Alex can click "Generate follow-up email".
    - The solution suggests an email draft (recap + next steps) based on the meeting summary and action items.
    - Alex can copy/paste this into their usual email client.

The result:

- Alex has better prep and less manual note-taking.
- The meeting is captured in a structured way that can be reused for VoC analysis.

3.6.2 VoC User Journey (User 2 – Priya)

Goal: See top customer themes and competitor mentions across many meetings for a specific product.

1. Log in & select product
    - Priya opens the solution.
    - Priya selects a product (for example "PIAM Visitor Management" or "PKI-as-a-Service").
2. Define analysis slice (optional filters)
    - Optionally, Priya can filter by:
        - Date range (for example "last 30 days")
        - Industry, region, or deal stage (based on mocked CRM data)
3. View VoC overview
    - The solution shows a VoC dashboard for the selected slice, containing:
        - Top pain points, with counts
        - Top feature requests, with counts
        - Top objections, with counts
        - Competitors mentioned, with counts
4. Inspect example snippets
    - For each theme or competitor, Priya can click to see:
        - Short example snippets or quotes taken from individual meetings
        - Links back to the underlying meetings/activities in the mocked data
5. Create a VoC summary
    - The solution can optionally generate a VoC summary skeleton, such as:
        - "Top 5 pain points for PIAM opportunities this period"
        - "Top 5 objections and their frequency"
        - "Competitors most often mentioned and in what context"
    - Priya can export or copy this into a slide or document.

The result:

- Priya can answer “What are customers struggling with?” and “Who do they compare us to?” using actual meeting data, not just anecdote.