
# Bilkent University
# Department of Computer Engineering

## Senior Design Project
## Group T2503
## VeriFact

## Analysis and Requirement Report

Orhun Ege Çelik - 22202321
Egehan Yıldız - 22203014
Alhassan Raad Jassim Al-Badri - 22201170
İrem Damla Karagöz - 22203691
Eray İşçi - 22201686

**Supervisor:** Sinem Sav
**Innovation Expert:** Mustafa Sakalsız
**Instructors:** İlker Burak Kurt, Mert Bıçakçı

**18/12/2025**

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

# Contents

# Analysis and Requirement Report

*VeriFact: A live claim-detection tool for Zoom*

## 1 Introduction

VeriFact is a real-time claim verification platform for online meetings. It helps presenters and participants make sure every claim has evidence. As a native Zoom application, it transcribes speech, detects factual claims, and checks them against uploaded relevant documents using a Retrieval-Augmented Generation (RAG) pipeline. Verified statements show color-coded indicators and clickable citations. Additionally, a wakeable Q&A agent provides permitted attendees with instant, source-based answers, serving as a secondary workflow that complements claim verification. By bringing together claim detection, secure document management and real-time fact checking, VeriFact reduces misinformation risks and helps organizations hold reliable, data-driven meetings.

## 2 Current System

Decision-making in this age happens in online meetings, yet organizations still lack a reliable way to verify factual claims as they are made. The current solution is quite manual: participants either trust speakers, take notes, or verify information after the meeting, at which point incorrect claims may have already influenced decisions and possibly the meeting's flow. Existing AI meeting tools such as Otter.ai, Fireflies.ai, MeetGeek, Granola AI, and Webex AI Companion focus on transcription, summarization, and meeting analytics, but they do not assess whether statements are factually correct or supported by internal documents.

Some more advanced tools partially overlap but still miss the core problem. Cluely provides real-time coaching and suggested responses for individuals, prioritizing performance rather than shared factual accuracy and raising ethical concerns. Glean offers powerful enterprise-wide search and post-meeting intelligence but lacks real-time claim verification during meetings. As a result, there is no solution that functions as a real-time fact-checking layer inside

meetings, leaving a clear gap for tools like VeriFact, which focuses on verifying claims against trusted documents, providing the speaker with near-instant verification for their claims.

# 3 Proposed System

## 3.1 Overview

The proposed system reflects VeriFact as a real-time claim verification infrastructure orchestrated with a pipeline style system, directly embedded into Zoom meetings through a bot. Essentially, the system operates by listening to meeting audio through a Zoom bot, transcribing the speech into segmented text, identifying check-worthy factual claims (claim detection), retrieving relevant evidence from both permanent and session-specific documents (RAG step), and evaluating the claim with cross referencing the RAG results. The architecture is designed to be quite modular, separating concerns across domains, allowing each component to scale independently and reliably.

## 3.2 Functional Requirements

### 3.2.1 Document Management (DM)

**REQ-DM-1:** The system shall support two document storage modes: Permanent Documents and Temporary Documents.

**REQ-DM-2:** The system shall maintain separate logical storage namespaces for permanent and temporary documents to ensure correct retrieval context.

**REQ-DM-3:** The system shall allow users to upload one or more documents before a meeting and during an active meeting session.

**REQ-DM-4:** The system shall validate file formats and reject unsupported formats with an appropriate error message.

**REQ-DM-5:** Upon upload, the system shall extract text, generate vector embeddings, and store the document in an S3-like bucket and vector database.

**REQ-DM-6:** The system shall restrict upload operations to users who have upload permissions. If a user without permission attempts to upload, the system shall display an “Insufficient Permissions” error.

**REQ-DM-7:** The system shall index and retrieve external documents only if the user has explicit access rights.

**REQ-DM-8:** When a user deletes a document, the system shall remove the document and all associated vector embeddings from all storage layers.

**REQ-DM-9:** The system shall update document metadata when a stored file is replaced or modified by an authorized user.

### 3.2.2 Permission and Access Control (PAC)

**REQ-PAC-1:** The system shall allow the meeting host to modify user permissions during the meeting.

**REQ-PAC-2:** The system shall support the following permission types: Upload Documents, Delete Documents, View Evidence, Access Confidential Files, and to-be-determined additional permissions.

**REQ-PAC-3:** The system shall notify participants when their permissions change.

**REQ-PAC-4:** The system shall restrict evidence visibility based on user permissions.

**REQ-PAC-5:** The system shall enforce organization-level access controls when retrieving from external storage repositories.

**REQ-PAC-6:** The system shall hide refuting-evidence content (red claims) from users who do not have permission to access the underlying document.

### 3.2.3 In-Meeting Features (IMF)

#### 3.2.3.1 Meeting Initialization

**REQ-IMF-1:** When the user opens the VeriFact panel, the system shall display the real-time transcript, claim verification panel, document list, evidence window, and Q&A interface.

**REQ-IMF-2:** The system shall activate audio capture, transcription, claim detection, and retrieval pipelines when the meeting begins.

**REQ-IMF-3:** The system shall load both permanent and temporary documents selected for the meeting at initialization time.

#### **3.2.3.2** Participant Management

**REQ-IMF-4:** The system shall register new participants and enable audio capture with speaker identification.

**REQ-IMF-5:** The system shall update participant roles dynamically in response to host permission changes.

#### **3.2.3.3** Real-Time Transcription

**REQ-IMF-6:** The system shall transcribe audio into text in real time.

**REQ-IMF-7:** The system shall perform speaker attribution for all transcribed segments.

**REQ-IMF-8:** The system shall display transcriptions immediately.

#### **3.2.3.4** Real-Time Claim Detection

**REQ-IMF-9:** The system shall analyze each sentence using a ClaimBuster-derived BERT classifier to detect factual claims [1].

**REQ-IMF-10:** The system shall flag claims that exceed a configurable confidence threshold.

#### **3.2.3.5** Real-Time Claim Verification

Evidence Retrieval

**REQ-IMF-11:** The system shall retrieve evidence from permanent documents, temporary documents, and authorized external documents.

**REQ-IMF-12:** The system shall perform vector similarity search and return the top-k relevant passages.

Claim Classification

**REQ-IMF-13:** The system shall classify each claim into one of the following: SUPPORTS, REFUTES, or NOT-ENOUGH-INFO (NEI) [2].

**REQ-IMF-14:** The system shall compute a confidence score for each classification.

Visualization

**REQ-IMF-15:** Supported claims shall be highlighted in green and shown to all authorized users.

**REQ-IMF-16:** Refuted claims shall be highlighted in red, with evidence visible only to users who have the required access level.

**REQ-IMF-17:** NEI claims shall be shown in gray and added to the manual review queue.

#### **3.2.4** Q&A Agent (QA)

**REQ-QA-1:** The system shall activate Q&A mode when the user says “Hey Agent” or when text is submitted via the input panel.

**REQ-QA-2:** The system shall retrieve evidence from all authorized sources (permanent, temporary, external).

**REQ-QA-3:** The system shall present answers with citations including document names and page or section numbers.

### 3.2.5 Post-Meeting Review (PMR)

**REQ-PMR-1:** The system shall present post-meeting transcripts including speaker identities, detected claims, classifications, and associated evidence.

**REQ-PMR-2:** The system shall allow exporting the meeting summary in PDF, Markdown, or plain text formats.

## 3.3 Non-functional Requirements

### 3.3.1 Usability Requirements (USE)

**REQ-USE-1:** The system shall display evidence cards containing the source document name, page number, and a relevant excerpt (maximum 150 characters) without requiring additional user interaction.

**REQ-USE-2:** The system shall auto-save document upload progress and allow users to resume interrupted uploads.

**REQ-USE-3:** The system shall provide access to permission management for document uploads within no more than two user interactions from the main panel.

**REQ-USE-4:** If a query requires more than 8 seconds to process, the system shall provide a temporary acknowledgment message (e.g., “Searching for information…”).

### 3.3.2 Reliability Requirements (REL)

**REQ-REL-1:** The system shall enforce document retention policies by automatically archiving documents older than 365 days to cold storage with a retrieval time of no more than 24 hours. Permanent documents are held to this restriction, whereas temporary documents will be deleted after the meeting ends.

**REQ-REL-2:** If the claim classification service becomes unavailable, the system shall buffer up to 50 unprocessed claims and process them when service is restored.

**REQ-REL-3:** If the Q&A bot fails to respond within 10 seconds, the system shall notify the user with “Agent is temporarily unavailable” and log the event for diagnostic purposes.

**REQ-REL-4:** The speaker identification subsystem shall attribute speech to the correct participant ID with at least 95% accuracy when multiple speakers are present.

**REQ-REL-5:** The RAG retrieval subsystem shall return at least one evidence snippet with cosine similarity > 0.6 for at least 90% of detected claims.

### 3.3.3 Performance Requirements (PERF)

**REQ-PERF-1:** The system shall ensure that end-to-end latency from speech utterance to claim verification output does not exceed 5 seconds for 95% of claims, distributed as:

- Speech-to-text: ≤ 2 seconds
- Claim classification: ≤ 0.5 seconds
- RAG retrieval: ≤ 1.5 seconds
- Evidence classification: ≤ 1 second

**REQ-PERF-2:** Document upload and vectorization shall support parallel processing and achieve the following completion times:

- Small documents (≤10 pages): ≤ 15 seconds
  Medium documents (11-50 pages): ≤ 45 seconds
- Large documents (51-100 pages): ≤ 90 seconds
  Processing throughput shall average at least 1.1 pages per second.

**REQ-PERF-3:** The system shall support concurrent document uploads by up to 3 users within the same meeting without degradation in processing time.

**REQ-PERF-4:** The Q&A bot shall acknowledge user queries within 1 second of activation phrase detection.

**REQ-PERF-5:** The claim classification service shall process at least 20 claims per second under load.

### 3.3.4 Supportability Requirements (SUP)

**REQ-SUP-1:** The RAG retrieval configuration parameters (e.g., top-k, similarity threshold) shall be adjustable through environment variables or configuration files.

**REQ-SUP-2:** Database schema migrations shall be backward-compatible for at least one version to support safe rollbacks.

### 3.3.5 Scalability Requirements (SCA)

**REQ-SCA-1:** The system shall support up to 50 concurrent meetings without measurable degradation in per-meeting performance metrics.

**REQ-SCA-2:** Each meeting shall support up to 50 participants with claim verification services active for the host and designated presenters.

**REQ-SCA-3:** The system shall support organizations with up to 1,000 registered users accessing the shared document repository.

**REQ-SCA-4:** The vector database shall scale to 1 million embedded document chunks while maintaining query response time below 150 ms.

**REQ-SCA-5:** The RAG pipeline shall support parallel processing of up to 10 concurrent claims within a single meeting with latency overhead not exceeding 20%.

### 3.4 Pseudo Requirements

1. The system shall run as a native Zoom application and provide an in-meeting side panel interface.
2. The system shall capture live meeting audio via a Zoom bot and convert it into real-time transcript segments with speaker attribution.
3. The system shall detect factual claims from the live transcript and flag claims exceeding a configurable confidence threshold.
4. The system shall retrieve relevant evidence for detected claims from authorized sources, including permanent documents, temporary meeting documents, and external documents (when permitted).
5. The system shall classify detected claims as SUPPORTS, REFUTES, or NOT-ENOUGH-INFO (NEI) and compute a confidence score for each result.
6. The system shall display verification results in the meeting UI using color-coded indicators and show evidence excerpts and citations to authorized users.
7. The system shall support two document modes (Permanent and Temporary), allow uploading before or during a meeting, and generate/store embeddings for retrieval.

8. The system shall enforce permission-based access control for uploading/deleting documents and viewing sensitive evidence during verification.
9. The system shall provide an in-meeting Q&A mode activated by a wake phrase (e.g., “Hey Agent”) or text input, returning answers grounded in authorized documents with citations.
10. The system shall support post-meeting review by presenting transcripts, detected claims, classifications, and associated evidence, with export options such as PDF/Markdown/text.

## 3.5 System Models

### 3.5.1 Scenarios

**Scenario 1.1: Verification - Evidence Supports Claim**

The "Supporting Path" where the speaker states something correct based on stored data.

**Table 1. Verification Scenario: Evidence Supports Claim**

| **Scenario 1.1** | **Verification: Claim Supported by Evidence** |
|---|---|
| **Pre-conditions** | 1. The meeting is active, and VeriFact Bot is invited.<br>2. A Permanent or Temporary document exists that confirms the speaker's statement.<br>3. The document's confidence score exceeds the verification threshold. |
| **Participating Actor** | Speaker (Participant), VeriFact Bot, System |
| **Main Flow of Events** | 1. A Participant makes a factual statement (e.g., "Our Q3 revenue grew by 15%").<br>2. The system detects the claim (UC_DetectClaim) and queries the vector database.<br>3. The system retrieves a document section with high semantic similarity (UC_RetrieveEvidence). |

| | |
|---|---|
| | 4. The system compares the claim against the evidence and determines they align.<br>5. The system marks the claim as "Supported".<br>6. A green "Supported" badge appears next to the transcript line (UC_ShowVerdict). |
| **Post-conditions** | 1. The claim is visibly marked as verified.<br>2. A citation link is generated, allowing authorized members to view the source (UC_ViewIndicator). |

**Scenario 1.2: Verification - Evidence Refutes Claim (Contradiction)**

The "Correction Path" where the speaker states something incorrect based on stored data.

**Table 2. Verification Scenario: Evidence Refutes Claim**

| **Scenario 1.2** | **Verification: Claim Contradicted (Refuted)** |
|---|---|
| **Pre-conditions** | 1. The meeting is active, and the Bot is listening.<br>2. The claim is factual in nature<br>3. A document exists that explicitly contradicts the speaker's statement.<br>4. The contradiction confidence is high. |
| **Participating Actor** | Speaker (Participant), VeriFact Bot, System |
| **Main Flow of Events** | 1. A Participant makes a claim (e.g., "The project deadline is set for December 30th").<br>2. The system queries the knowledge base and finds a project charter stating the deadline is "November 30th" (UC_RetrieveEvidence).<br>3. The system detects a logical conflict between the spoken date and the stored date.<br>4. The system marks the claim as "Refuted". |

| | |
|---|---|
| | 5. A red "Refuted" warning/badge is displayed to participants (UC_ShowVerdict).<br>6. The system highlights the specific conflicting evidence snippet in the UI. |
| **Post-conditions** | 1. Participants are alerted to the inaccuracy.<br>2. The transcript logs the correction for the meeting minutes.<br>3. A citation link is generated, allowing authorized members to view the source of contradicting evidence (UC_ViewIndicator). |

**Scenario 1.3: Verification - Evidence Not Found (NEI)**

The "Missing Data Path" where the system cannot verify the statement.

**Table 3. Verification Scenario: Not Enough Information (NEI)**

| **Scenario 1.3** | **Verification: Not Enough Information (NEI)** |
|---|---|
| **Pre-conditions** | 1. The meeting is active.<br>2. The claim is factual in nature.<br>3. No relevant documents exist in the Permanent or Temporary sets, OR the similarity score is below the confidence threshold. |
| **Participating Actor** | Speaker (Participant), System |
| **Main Flow of Events** | 1. A Participant makes a niche claim (e.g., "The legacy server IP is 192.168.1.55").<br>2. The system searches all available document embeddings (UC_RetrieveEvidence).<br>3. The search returns either no results or results with low relevance scores.<br>4. The system classifies the verdict as "Not Enough Information" (NEI).<br>5. A neutral/gray "Unverified" or "NEI" indicator is shown (UC_ShowVerdict). |

| | |
|---|---|
| **Post-conditions** | 1. The claim remains in the transcript without a verification badge (or with a neutral one).<br>2. No citation links are generated. |

**Scenario 2.1: Agent Q&A - Answer Successfully Retrieved**

The "Support Path" where the Agent finds clear evidence and answers the user's question.

**Table 4. Q&A Agent Scenario: Answer Successfully Retrieved**

| **Scenario 2.1** | **Q&A: Agent Answers with Evidence** |
|---|---|
| **Pre-conditions** | 1. VeriFact Bot is active in the meeting.<br>2. A Member (or higher) initiates the request ("Hey Agent").<br>3. Documents answering the question exist and the user has permission to view them. |
| **Participating Actor** | Member, VeriFact Agent, System |
| **Main Flow of Events** | 1. A Member asks a specific question (e.g., "What is the budget cap for Q4 marketing?").<br>2. The system interprets the intent and searches the vector database (UC_RetrieveEvidence).<br>3. The system finds a matching section in the "2024 Budget Plan" with high relevance.<br>4. The Agent generates a natural language answer: "The Q4 marketing budget is capped at $50,000." (UC_QA_Answer)<br>5. The Agent appends a citation to the source document and page number (UC_QA_Cite).<br>6. The answer and citation are displayed in the chat interface (UC_QA_Display). |
| **Post-conditions** | 1. The user receives the correct answer.<br>2. The user can click the citation to open the source document (since they have permission). |

**Scenario 2.2: Agent Q&A - Conflicting Information Found**

The "Conflict Path" where the Agent finds two different answers in the documents.

**Table 5. Q&A Agent Scenario: Conflicting Information Detected**

| **Scenario 2.2** | **Q&A: Agent Reports Conflicting Evidence** |
|---|---|
| **Pre-conditions** | 1. VeriFact Bot is active.<br>2. The Knowledge Base contains outdated or contradictory documents (e.g., "Draft v1" vs "Final v2") that the user has access to. |
| **Participating Actor** | Member, VeriFact Agent, System |
| **Main Flow of Events** | 1. A Member asks: "What is the project deadline?"<br>2. The system retrieves two high-ranking snippets: one says "Nov 15th" (from an email) and another says "Dec 1st" (from the Project Charter).<br>3. The Agent detects the inconsistency between the retrieved contexts.<br>4. The Agent generates a synthesized answer highlighting the conflict: "I found conflicting information. The Project Charter states Dec 1st, but a recent email mentions Nov 15th."<br>5. The Agent cites both sources so the user can investigate (UC_QA_Cite). |
| **Post-conditions** | 1. The user is aware of the discrepancy in their documentation.<br>2. Both sources are linked for manual verification. |

**Scenario 2.3: Agent Q&A - Answer Not Found (NEI)**

The "Missing Data Path" where the Agent cannot answer based on current knowledge.

**Table 6. Q&A Agent Scenario: Answer Not Found in Knowledge Base**

| Scenario 2.3 | Q&A: Not Enough Information (NEI) |
|---|---|
| **Pre-conditions** | 1. VeriFact Bot is active.<br>2. The answer does not exist in the uploaded documents, OR the user does not have permission to view the documents that contain the answer. |
| **Participating Actor** | Member, VeriFact Agent |
| **Main Flow of Events** | 1. A Member asks a question about an external topic (e.g., "What is the competitor's stock price today?").<br>2. The system searches the authorized document embeddings (UC_RetrieveEvidence).<br>3. The search returns low relevance scores (below the confidence threshold).<br>4. The Agent generates a fallback response: "I cannot find information regarding that topic in the current document set."<br>5. No citations are displayed. |
| **Post-conditions** | 1. The user is informed that the Knowledge Base is insufficient for this query.<br>2. An answer explaining the fallback is provided. |

**Scenario 3: Permission-Based Evidence Inspection**

This scenario addresses the Member vs Admin constraint (UC_OpenDoc note).

**Table 7. Access Control Scenario: Restricted Evidence Inspection**

| Scenario 3 | Attempting to View Restricted Evidence |
|---|---|
| **Pre-conditions** | 1. A claim has been verified (Supported or Refuted).<br>2. The supporting evidence comes from a document with "Admin-Only" permissions.<br>3. The user attempting to click the citation is a standard Member (not Admin/Owner). |

| | |
|---|---|
| **Participating Actor** | Member, System |
| **Main Flow of Events** | 1. The Member sees a claim marked as "Refuted" and clicks the citation to see why (UC_ViewIndicator).<br>2. The system checks the user's role against the document's permission level (UC_OpenDoc check).<br>3. The system identifies the user as a Member, but the document requires Admin access.<br>4. The system denies the request to open the full document text.<br>5. The system displays a "Restricted Access" toast notification (e.g., "You do not have permission to view this source document"). |
| **Post-conditions** | 1. The user knows the claim is refuted but cannot read the confidential source text.<br>2. Security of the underlying document is maintained. |

**Scenario 4: Organization Onboarding**

The first step for a new organization (Handling UC_CreateOrg).

**Table 8. Administration Scenario: Organization Onboarding**

| **Scenario 4** | **Creating a New Organization Environment** |
|---|---|
| **Pre-conditions** | 1. A new client has purchased VeriFact.<br>2. The user is the designated Owner. |
| **Participating Actor** | Owner, System |
| **Main Flow of Events** | 1. The owner logs into the web portal for the first time.<br>2. The Owner initiates "Create Organization" (UC_CreateOrg).<br>3. The System provisions a secure, isolated tenant (database shard) for this Org.<br>4. The Owner invites other users as Admins or Members. |

| | |
|---|---|
| | 5. The Owner uploads the first set of "Permanent Documents" (e.g., Company Handbook) and selects the default accessibility level for each (e.g., "All Members" or "Admins Only"). |
| **Post-conditions** | 1. The Organization is active.<br>2. The System is ready to accept meeting connections from this Org's hosts. |

**Scenario 5: Initializing Meeting with Organization Context**

Security setup is verified for organizations before any verification activity begins. The critical "handshake" that prevents data leaks between different companies using the same software.

**Table 9. Initialization Scenario: Secure Meeting Context Loading**

| **Scenario 5** | **Secure Bot Initialization** |
|---|---|
| **Pre-conditions** | 1. A Zoom meeting is in progress.<br>2. The Host belongs to a valid Organization in VeriFact. |
| **Participating Actor** | Meeting Host, VeriFact Bot, System |
| **Main Flow of Events** | 1. The Host invites the VeriFact Bot into the Zoom meeting (UC_InviteBot).<br>2. The Bot joins the waiting room or meeting.<br>3. The System identifies the Host's Organization ID based on their credentials.<br>4. The System loads the specific "Organization Context" (loading only that Org's Permanent Documents and relevant Temporary Documents) (UC_InitMeeting).<br>5. The Bot posts a "Ready" message: "VeriFact is active for [Organization Name]." |
| **Post-conditions** | 1. The Bot is active and listening. |

| | |
|---|---|
| | 2. Data strictness is enforced: The bot cannot access documents from any other Organization. |

**Scenario 6: Granting Permissions During a Meeting**

When the admin wants to change the accessibility level of certain documents, they can do it during meetings. This keeps the meeting flowing while maintaining security.

**Table 10. Access Control Scenario: Granting Permissions During Meeting**

| **Scenario 6** | **Permission Granting During Meeting** |
|---|---|
| **Pre-conditions** | 1. The meeting is active.<br>2. A standard Member attempts to view a citation but is denied access (as per Scenario 4).<br>3. An Admin is present in the meeting (or accessible via the dashboard). |
| **Participating Actor** | Admin, System |
| **Main Flow of Events** | 1. The Admin updates the access permission of a document from the dashboard(e.g., changing “Only Owner” to "All Members" access).<br>2. The system updates the Member's session permissions immediately.<br>3. The Member clicks the citation again and successfully opens the proof document (UC_OpenDoc). |
| **Post-conditions** | 1. The Member can view the previously restricted evidence.<br>2. The access log records the permission change. |

**Scenario 7: Updating the Permanent Knowledge Base**

Companies often update policies. This scenario handles updating and controlling the maintenance lifecycle of documents. The system must delete the old mathematical embeddings and replace them with the new ones.

**Table 11. Knowledge Base Scenario: Updating Permanent Documents**

| **Scenario 7** | **Updating an Outdated Policy Document** |
|---|---|
| **Pre-conditions** | 1. A document named "HR Policy 2024" currently exists in the Permanent Document set.<br>2. An Admin has the new "HR Policy 2025" file ready for upload. |
| **Participating Actor** | Admin, System |
| **Main Flow of Events** | 1. The Admin navigates to the Permanent Documents management console.<br>2. The Admin selects "HR Policy 2024" and chooses "Update/Replace" (UC_UpdatePerm).<br>3. The Admin uploads the "HR Policy 2025" file.<br>4. The system deletes the vector embeddings associated with the old 2024 version.<br>5. The system generates and stores new embeddings for the 2025 version.<br>6. The system confirms the update to the Admin. |
| **Post-conditions** | 1. Future claims/queries will be verified against the 2025 policy.<br>2. The 2024 policy is no longer accessible for verification. |

**Scenario 8: Uploading and Using Temporary Meeting Documents**

Handling the temporary documents data that should only exist for one meeting (e.g., a draft contract).

**Table 12. Document Management Scenario: Uploading Temporary Files**

| **Scenario 8** | **Uploading and Using Temporary Meeting Documents** |
|---|---|
| **Pre-conditions** | 1. The user acting is an Admin or Owner.<br>2. A meeting context has been initialized or is in preparation.<br>3. The documents to be uploaded are available. |

| | |
|---|---|
| **Participating Actor** | Admin (or Owner), System |
| **Main Flow of Events** | 1. An Admin uploads temporary documents intended for the current meeting (UC_UploadTemp).<br>2. The system sets the permission level for these documents (e.g., "Meeting Participants Only" or "Admins Only").<br>3. The system parses and indexes the documents into the knowledge base.<br>4. During the meeting, claims are verified using evidence from both these temporary docs and permanent docs.<br>5. After the meeting concludes (or lifecycle ends), the system triggers expiration. |
| **Post-conditions** | 1. Temporary documents are removed/expired from the system (UC_ExpireTemp).<br>2. No derived knowledge from temporary docs remains accessible after the meeting scope ends. |

**Scenario 9: Post-Meeting Cleanup of Temporary Documents**

Ensuring temporary documents are cleaned up and do not remain in the database or used in future meetings (Handling UC_ExpireTemp).

**Table 13. Lifecycle Scenario: Post-Meeting Cleanup of Temporary Data**

| **Scenario 9** | **Automatic Expiration of Temporary Documents** |
|---|---|
| **Pre-conditions** | 1. A meeting with "Temporary Documents" (e.g., highly sensitive M&A draft) has just concluded.<br>2. The Zoom API sends a "Meeting Ended" signal to VeriFact. |
| **Participating Actor** | System |

| | |
|---|---|
| **Main Flow of Events** | 1. The System receives the "End of Meeting" trigger.<br>2. The System identifies all documents tagged as "Temporary" for this specific meeting ID.<br>3. The System permanently deletes the vector embeddings and file references for these documents (UC_ExpireTemp).<br>4. The System logs the deletion event for compliance auditing.<br>5. The System retains the verification logs (the transcript verdict) but removes the link to the source text. |
| **Post-conditions** | 1. Sensitive temporary data is deleted fully from the Knowledge Base.<br>2. Future meetings cannot access this data. |

**Scenario 10: Deleting a Permanent Document**

Removing permanent data entirely from organization's documents (Handling UC_DeletePerm).

**Table 14. Knowledge Base Scenario: Deleting Permanent Documents**

| **Scenario 10** | **Deleting Permanent Documents** |
|---|---|
| **Pre-conditions** | 1. A document exists in the Permanent set (e.g., "Project X_2020.pdf").<br>2. The project has been cancelled, and the data is no longer relevant.<br>3. An Admin initiates the deletion. |
| **Participating Actor** | Admin, System |
| **Main Flow of Events** | 1. The Admin navigates to the Permanent Documents dashboard.<br>2. The Admin selects the obsolete document and clicks "Delete" (UC_DeletePerm).<br>3. The System displays a confirmation warning.<br>4. The Admin confirms the action. |

| | |
|---|---|
| | 5. The System permanently removes the file and its associated vector embeddings from the Knowledge Base. |
| **Post-conditions** | 1. The document is deleted.<br>2. Future queries will never reference this data again (preventing "hallucinations" based on dead projects). |

**Scenario 11: Filtering Non-Factual Statements**

Preventing "Alert Fatigue" by ensuring the bot ignores subjective opinions.

**Table 15. Detection Scenario: Filtering Non-Factual Statements**

| **Scenario 11** | **Filtering Opinions vs. Facts** |
|---|---|
| **Pre-conditions** | 1. The meeting is active.<br>2. The participant is speaking subjectively or casually. |
| **Participating Actor** | Speaker (Participant), System |
| **Main Flow of Events** | 1. The participant says: "I feel like our team synergy is improving drastically."<br>2. The System transcribes the audio.<br>3. The Fact-Check classifier analyzes the sentence structure (UC_DetectClaim).<br>4. The System determines this is an opinion/subjective statement, not a verifyable fact.<br>5. The System skips the database query and verification step.<br>6. No badge (Green/Red/Gray) is displayed; the transcript remains plain text. |
| **Post-conditions** | 1. The system saves resources by not querying the database.<br>2. The user interface remains clean and uncluttered. |

**Scenario 12: Correction of Temporary Meeting Materials**

An Admin might upload the wrong "Draft Contract" before the meeting. They need to fix it without cancelling the whole meeting setup.

**Table 16. Maintenance Scenario: Correction of Temporary Materials**

| **Scenario 12** | **Deleting or Replacing a Temporary Document** |
|---|---|
| **Pre-conditions** | 1. An Admin previously uploaded a document to the Temporary set (e.g., "Budget_v1.pdf").<br>2. The meeting has not yet ended.<br>3. The Admin realizes the file is incorrect or no longer needed. |
| **Participating Actor** | Admin, System |
| **Main Flow of Events** | 1. The Admin views the "Meeting Prep" dashboard.<br>2. The Admin selects the incorrect temporary file.<br>3. The Admin chooses "Delete" (or "Replace" with "Budget_v2.pdf").<br>4. The System removes the file and immediately purges its vector embeddings from the temporary index (UC_DeleteTemp).<br>5. (If replacing) The System indexes the new file (UC_UpdateTemp) and reconfirms the accessibility setting of the updated file. |
| **Post-conditions** | 1. The Bot will no longer use the incorrect file for verification.<br>2. The risk of verifying claims against bad data is removed. |

### 3.5.2 Use-Case Model

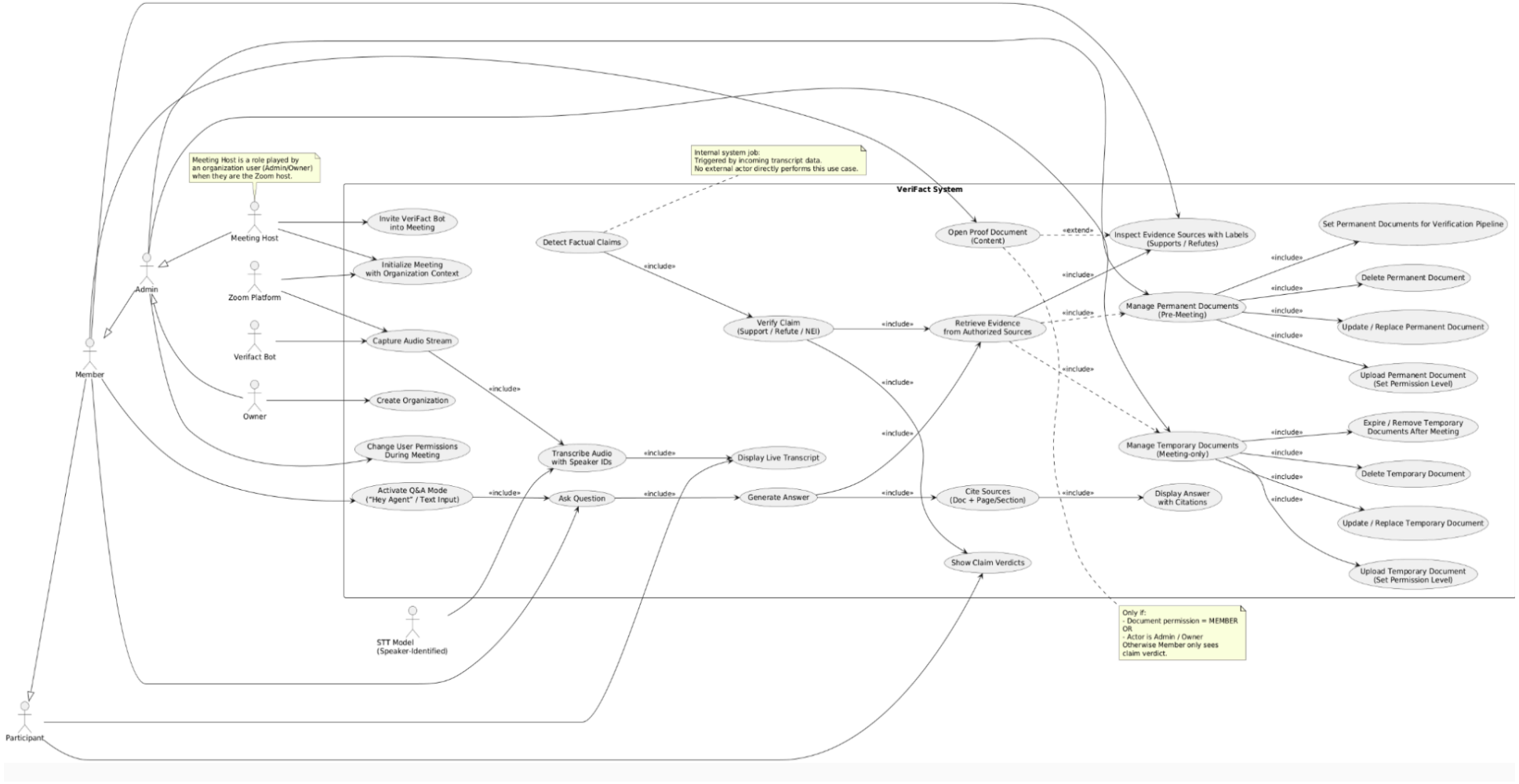


**Figure 1:** Use Case Diagram of VeriFact.

### 3.5.3 Object and Class Model

#### 3.5.3.1 Class Model

A high resolution .svg format of the class diagram can be found in the link below:
https://drive.google.com/file/d/1NjggHYE8C8LMguZkHEMAlnz9d8FwRoCm/view?usp=sharing

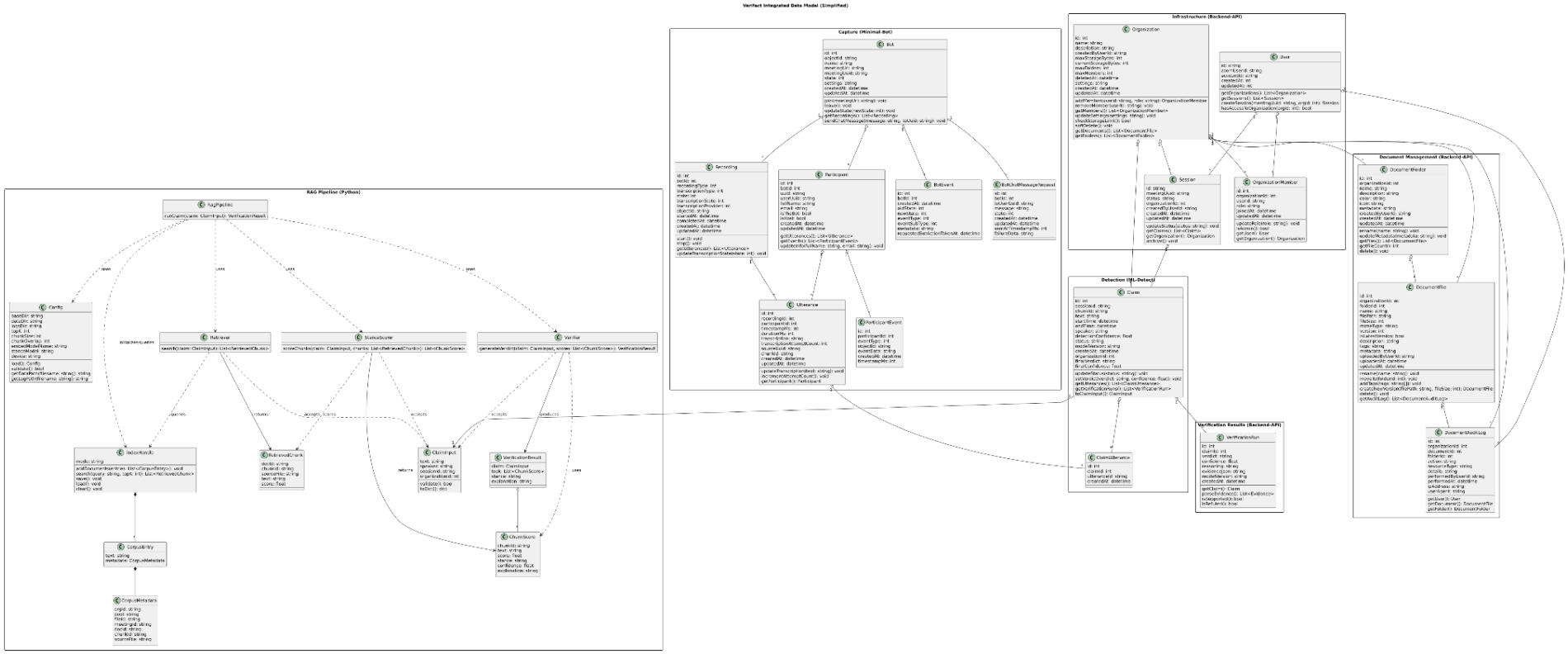

**Figure 2:** Class Diagram of VeriFact.

**Capture (Minimal-Bot) Package**

**Bot:** Represents a Zoom meeting bot instance that joins meetings, manages recordings, and coordinates participant interactions

**Recording:** Captures audio/video data from a bot session with transcription state tracking and provider information

**Participant:** Represents an individual in a Zoom meeting, tracking identity, role (host/attendee), and whether they are the bot itself

**ParticipantEvent:** Records discrete events (join, leave, reactions) for participants with timestamps and event metadata

**Utterance:** A single spoken statement from a participant, including transcription text, timing, duration, and linking to audio chunks

**BotEvent:** Logs state transitions and actions taken by the bot (joining, leaving, errors) with metadata

**BotChatMessageRequest:** Manages chat messages the bot sends to meeting participants, tracking delivery state and failures

**Detection (ML-Detect) Package**

**Claim:** A factual statement detected in meeting transcriptions that requires verification, with confidence scores and organizational context

**ClaimUtterance:** Junction entity linking claims to the specific utterances that contain them, enabling traceability to source audio

**Infrastructure (Backend-API) Package**

**User:** Represents a Zoom user account with authentication details, managing access to organizations and sessions

**Session:** A verification session tied to a specific Zoom meeting, tracking claims detected and verification status

**Organization:** Multi-tenant entity representing a company or team, managing members, storage quotas, and document access

**OrganizationMember:** Junction entity defining user membership and role (admin/member) within an organization

**Document Management (Backend-API) Package**

**DocumentFolder:** Organizes documents into a single-level folder structure with customizable color/icon for UI display

**DocumentFile:** Represents an uploaded file with versioning support, size limits, and metadata for searchability

**DocumentAuditLog:** Immutable log of all document operations (upload, delete, move, view) with user tracking and IP/user-agent capture

## Verification Results (Backend-API) Package

**VerificationRun:** Stores the outcome of verifying a claim against documents, including verdict (supported/refuted), confidence score, and evidence JSON

## RAG Pipeline (Python) Package

**Config:** Configuration parameters for the RAG system including model names, chunk sizes, retrieval settings, and hardware preferences

**CorpusMetadata:** Metadata attached to each document chunk, tracking organizational ownership, source file, and chunk identifiers

**CorpusEntry:** A text chunk from a document paired with its metadata, representing the atomic unit of the corpus

**IndexHandle:** Manages the vector index for semantic search, supporting document addition, querying, and persistence operations

**ClaimInput:** Standardized input format converting a detected claim into the structure required by the RAG pipeline

**RetrievedChunk:** A document chunk returned from semantic search with relevance score and source information

**ChunkScore:** Combines a retrieved chunk with stance classification (support/refute) and confidence from the stance model

**VerificationResult:** Final output aggregating scored chunks into an overall verdict and human-readable explanation

**RagPipeline:** Orchestrator coordinating the end-to-end verification workflow from claim input to final verdict

**Retriever:** Component responsible for semantic search against the document index, returning top-k relevant chunks

**StanceScorer:** Evaluates each retrieved chunk's relationship to the claim using a fine-tuned stance classification model

**Verifier:** Aggregates scored chunks to produce the final verdict and generate an explanation with evidence citations

## 3.5.3.2 Object Model

A high resolution .svg format of the object diagram can be found in the link below:
https://drive.google.com/file/d/1-BRR_oORBr7NS40Z10MGXqlj0tgwl9zb/view?usp=sharing

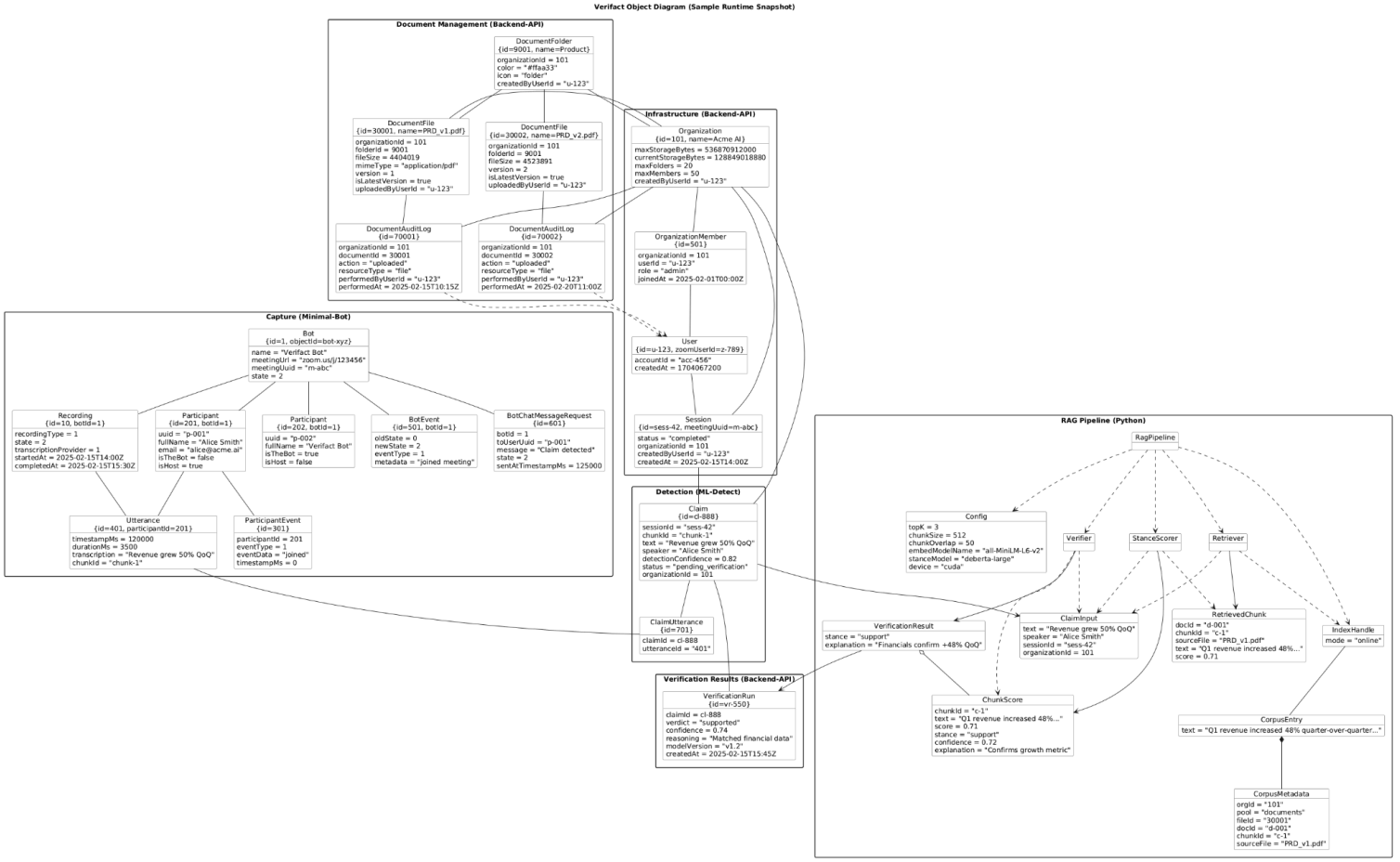

**Figure 3:** Object Diagram of VeriFact.A concrete instantiation showing a complete verification workflow for Organization 101 (Acme AI): the Verifact Bot joins Zoom meeting m-abc and captures utterances from participant Alice Smith (User u-123), the ML-Detect service identifies claim cl-888 ("Revenue grew 50% QoQ") with 0.82 detection confidence during session sess-42, the RAG pipeline retrieves relevant chunks from document files PRD_v1.pdf and PRD_v2.pdf stored in the Product folder (id=9001), the StanceScorer evaluates evidence with 0.72 stance confidence, and the Verifier generates VerificationRun vr-550 with a "supported" verdict at 0.74 confidence.

## 3.5.4 Dynamic Models

### 3.5.4.1 Sequence Diagrams

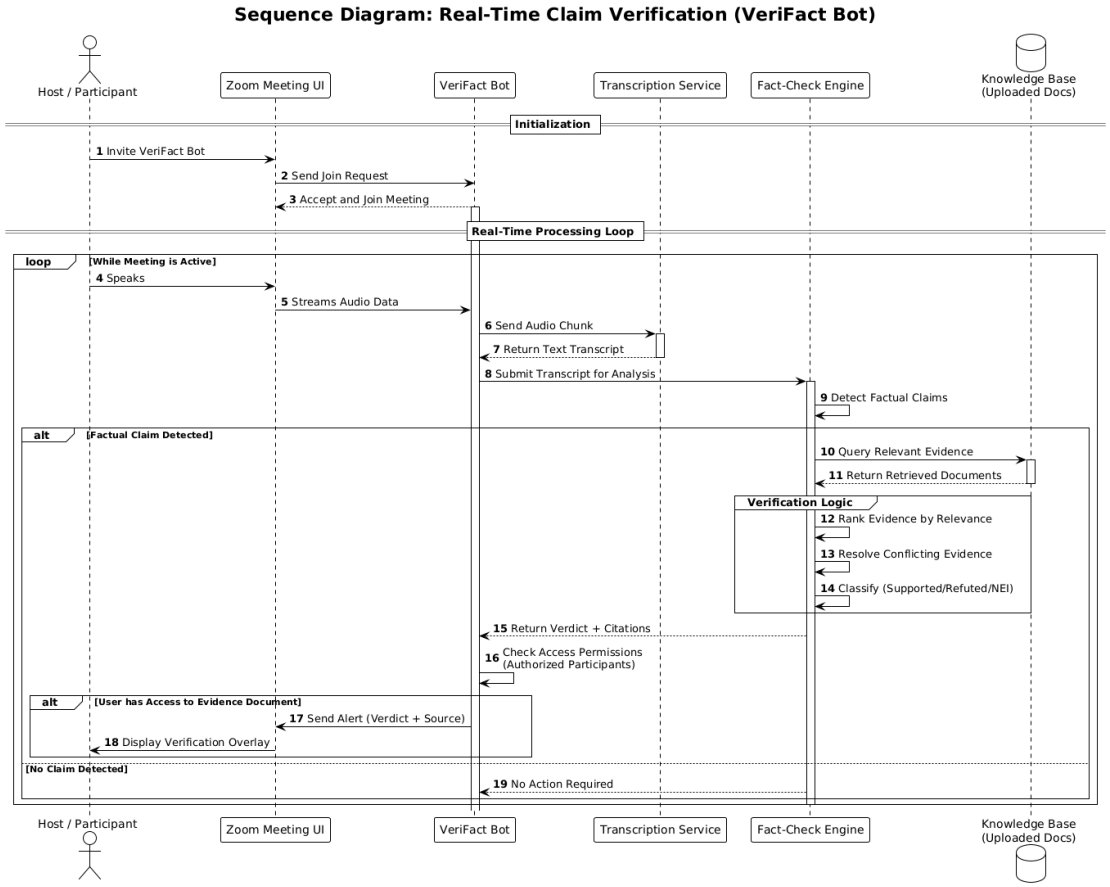


**Figure 4:** Sequence Diagram of VeriFact for real-time claim verification.

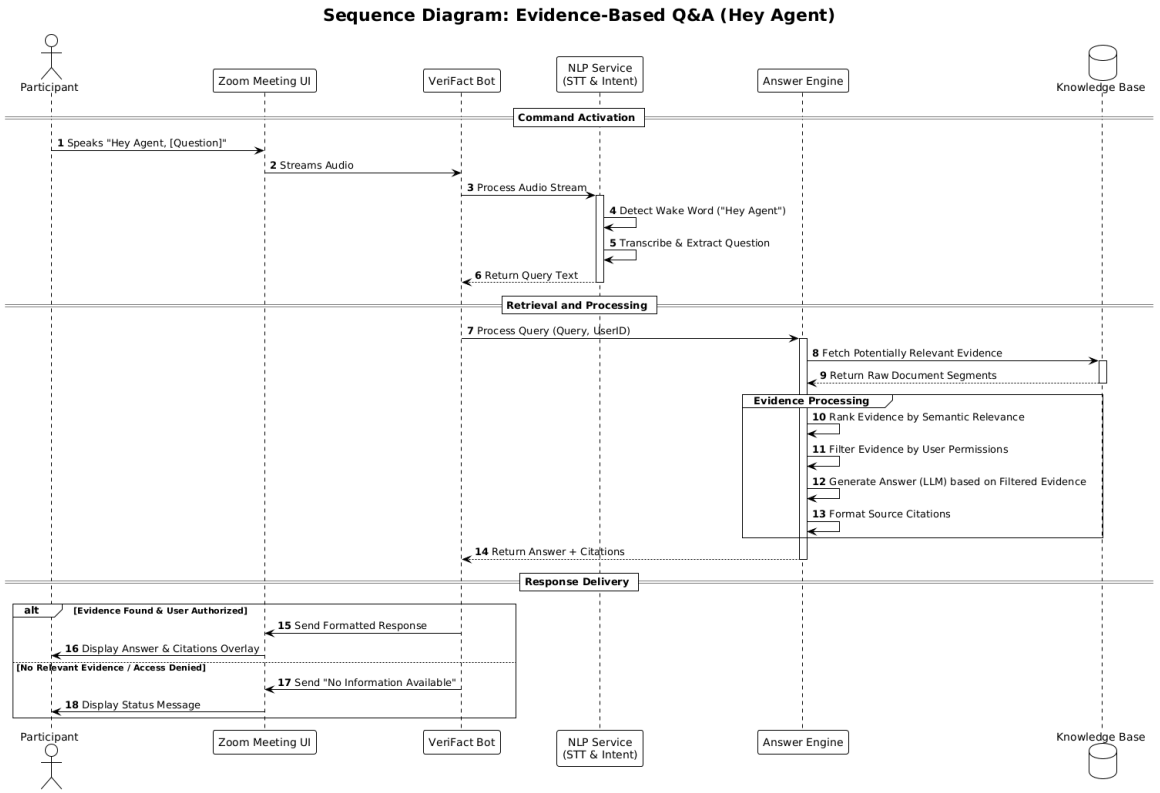


**Figure 5:** Sequence Diagram of VeriFact for Q&A using Agent.

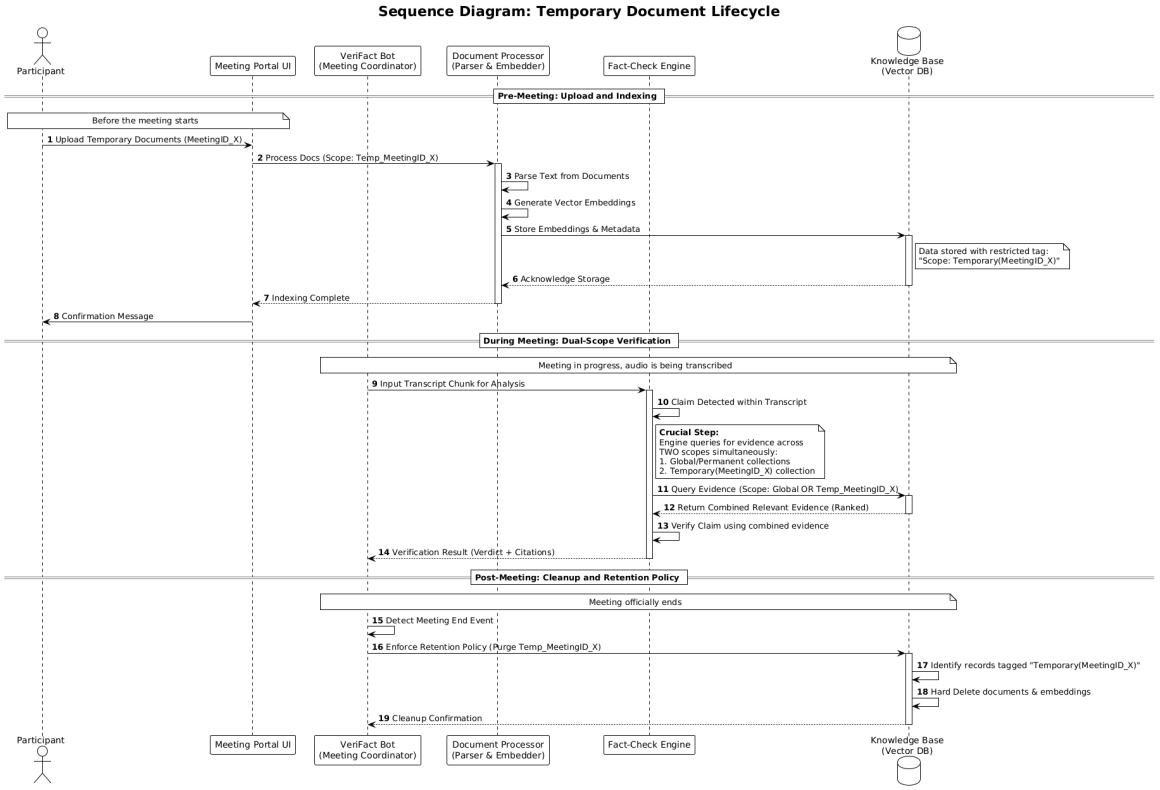


**Figure 6:** Sequence Diagram of VeriFact about temporary documents lifecycle.

## 3.5.5 User Interface

This section presents the user interface designs for our application which covers in-meeting views, lobby views, and post-meeting views. Both current implementations and future planned enhancements are documented.

### 3.5.5.1 In-Meeting UIs - Current Work

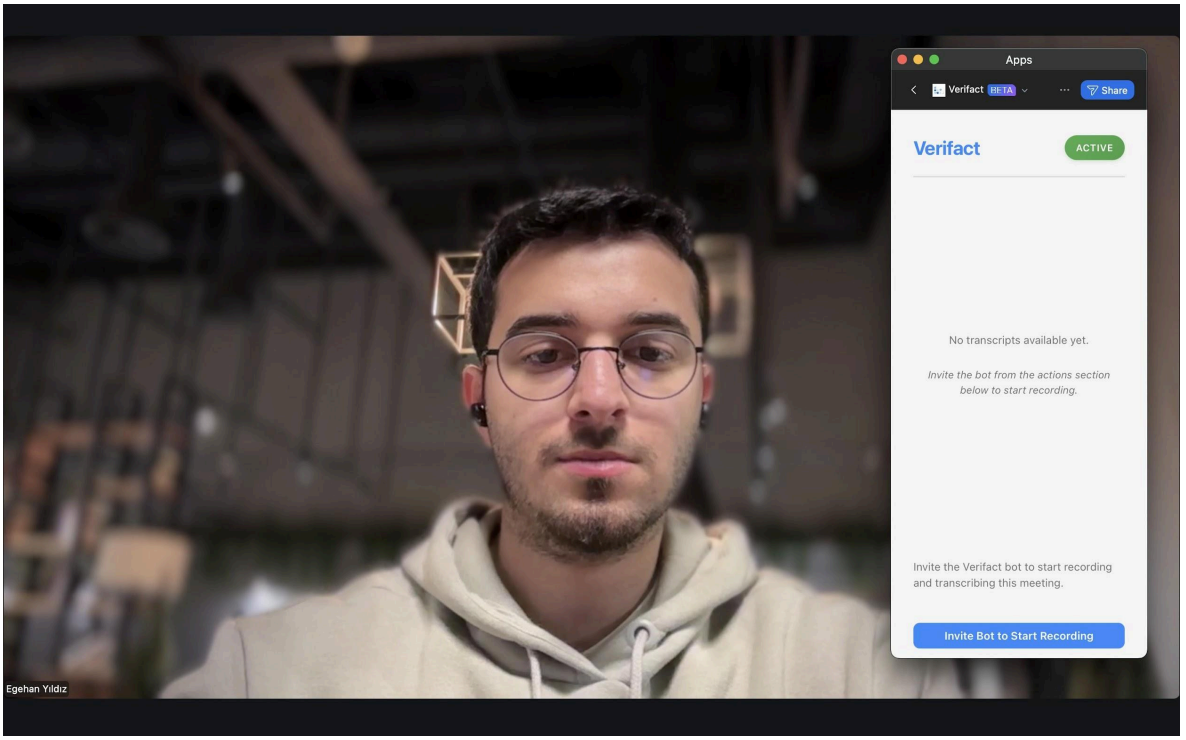


**Figure 6:** VeriFact Current Initial View

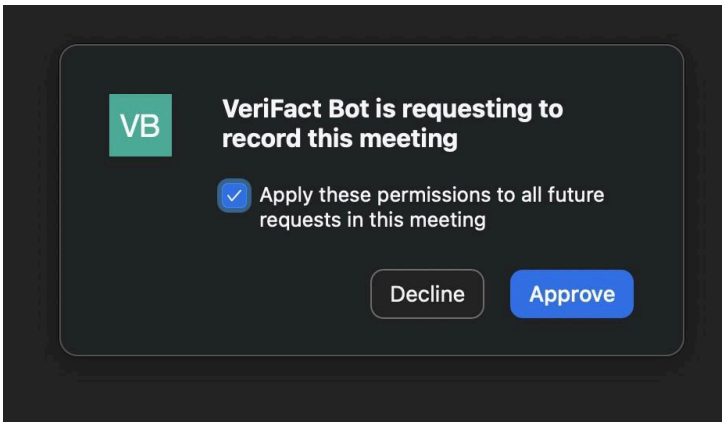


**Figure 7:** VeriFact Bot Request After Inviting the Bot

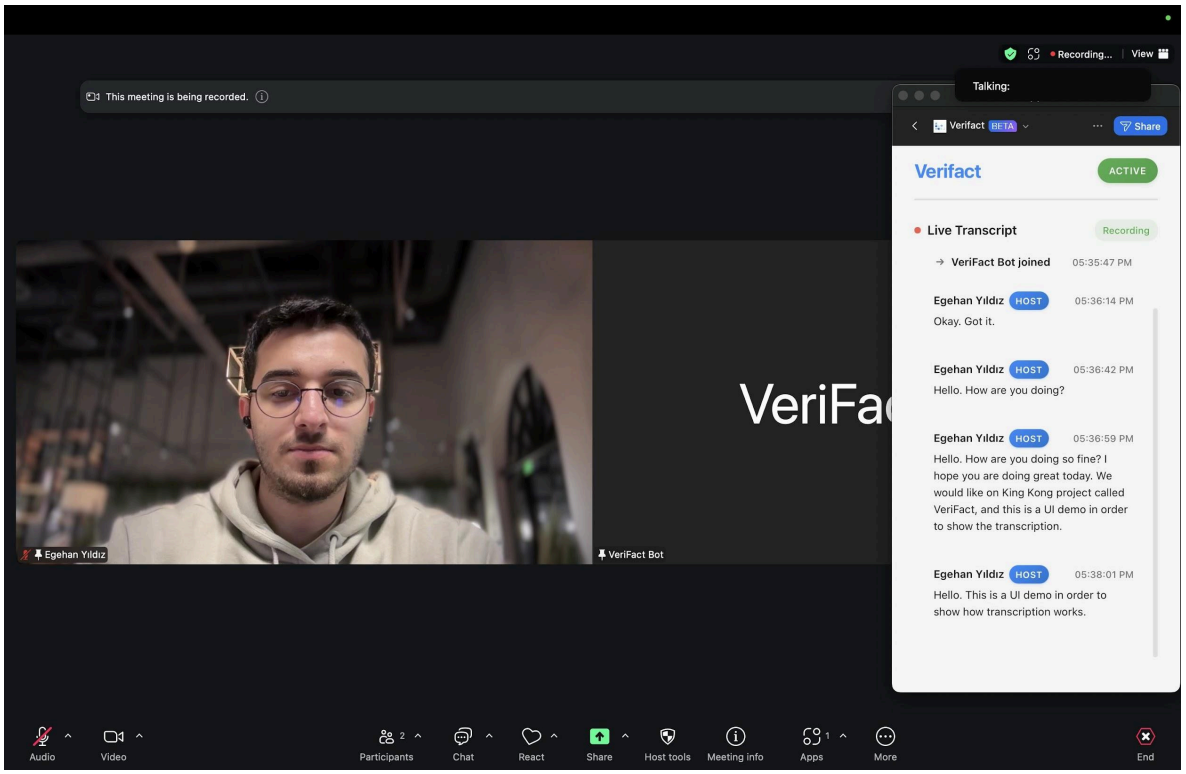


**Figure 8:** VeriFact In-meeting Transcription Example #1

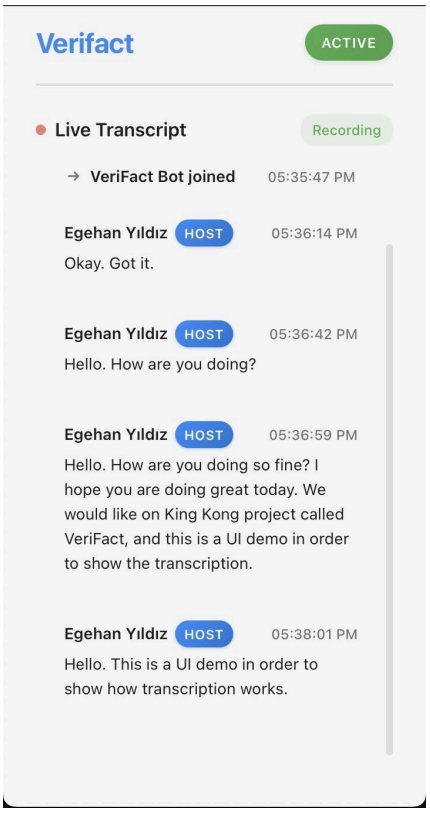


**Figure 9:** Transcription Example #1 Zoomed In

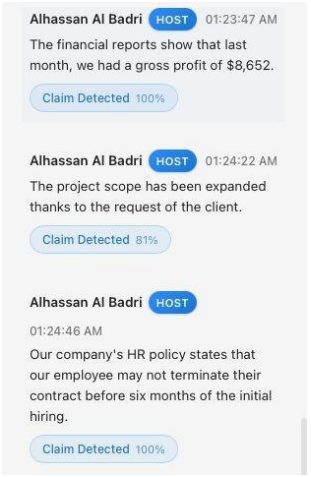


**Figure 10:** Transcription Example #2 Zoomed In

### 3.5.5.2 In-Meeting UIs - Future Work

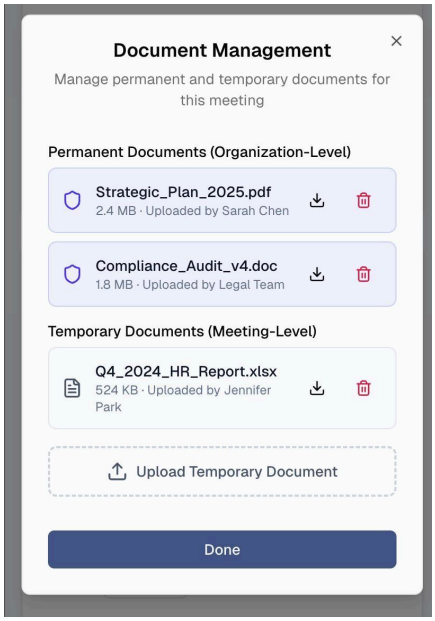


**Figure 11:** Document Management Pane Within Meeting

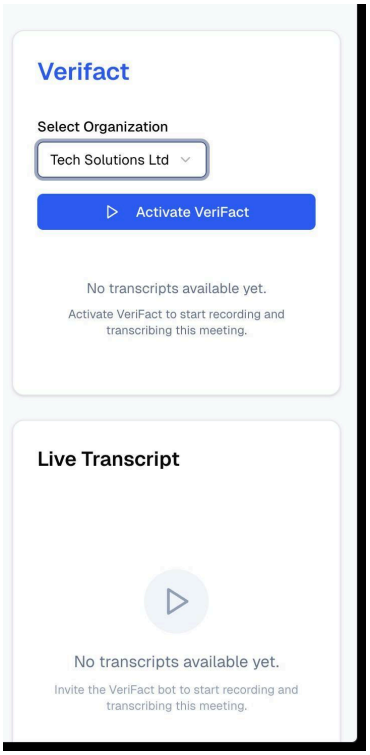


**Figure 12:** Advancement on Activate VeriFact Initial Screen

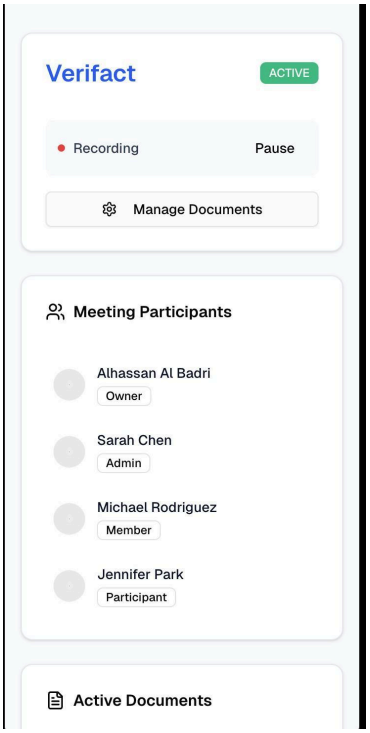


**Figure 13** General In-meeting View

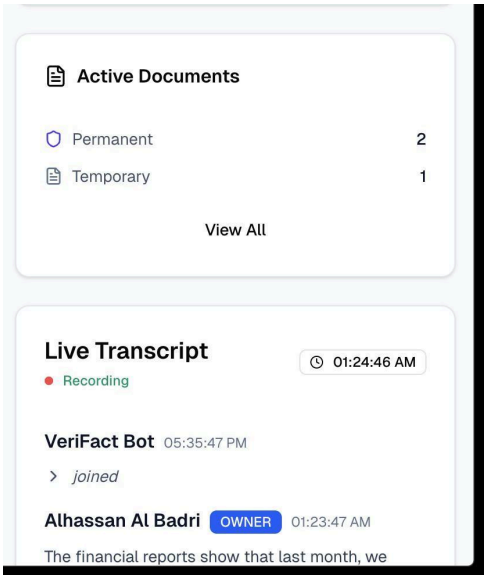


**Figure 14:** General In-meeting View with Activated Docs Pane Addition

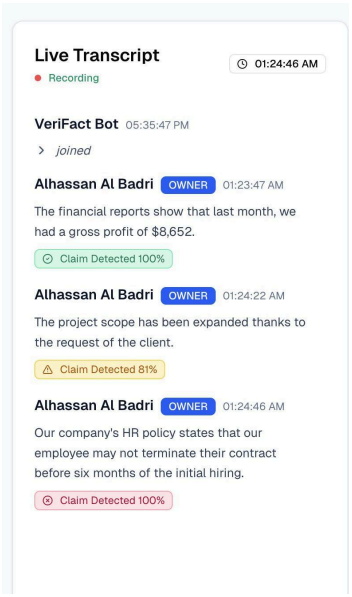


**Figure 15:** Claim detection scores and results view

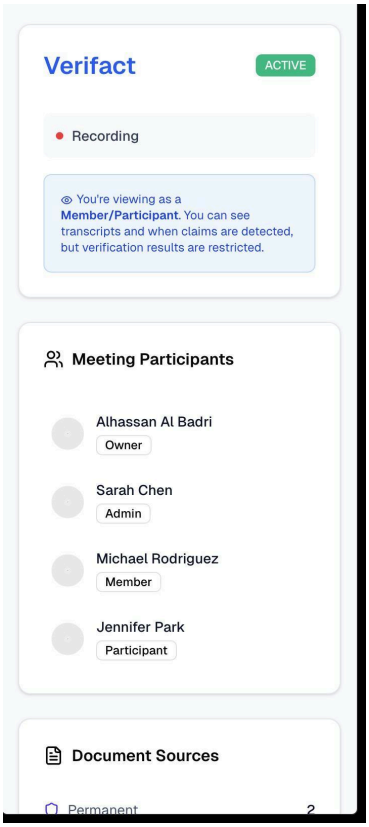


**Figure 16:** Member and Participant Screen with a Note Addition

### 3.5.5.3 LobbyView UIs - Current Work

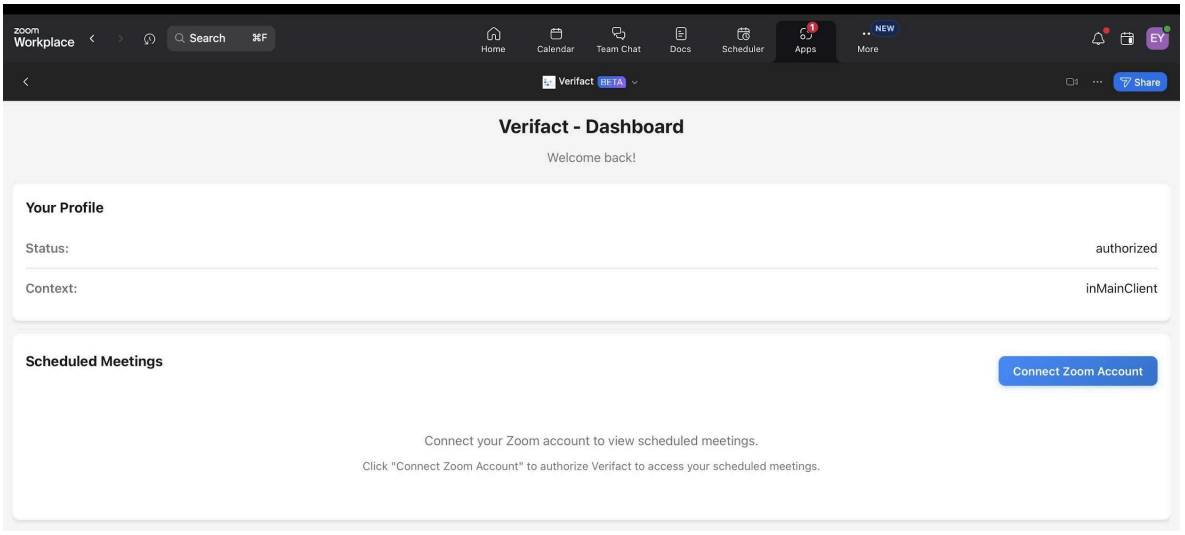


**Figure 17:** Lobby View Top Vertical Part

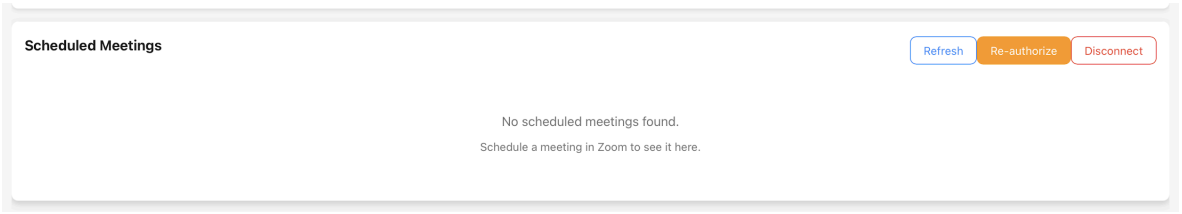


**Figure 18:** Scheduled Meeting Section After Connect Zoom Button

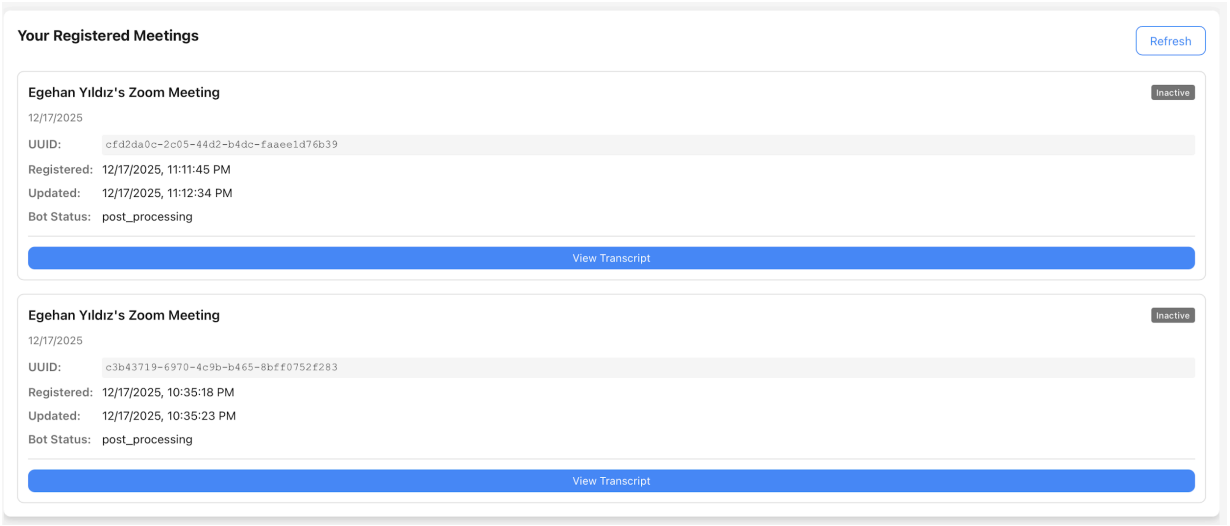


**Figure 19:** Preview Meetings and Their Available Transcripts View

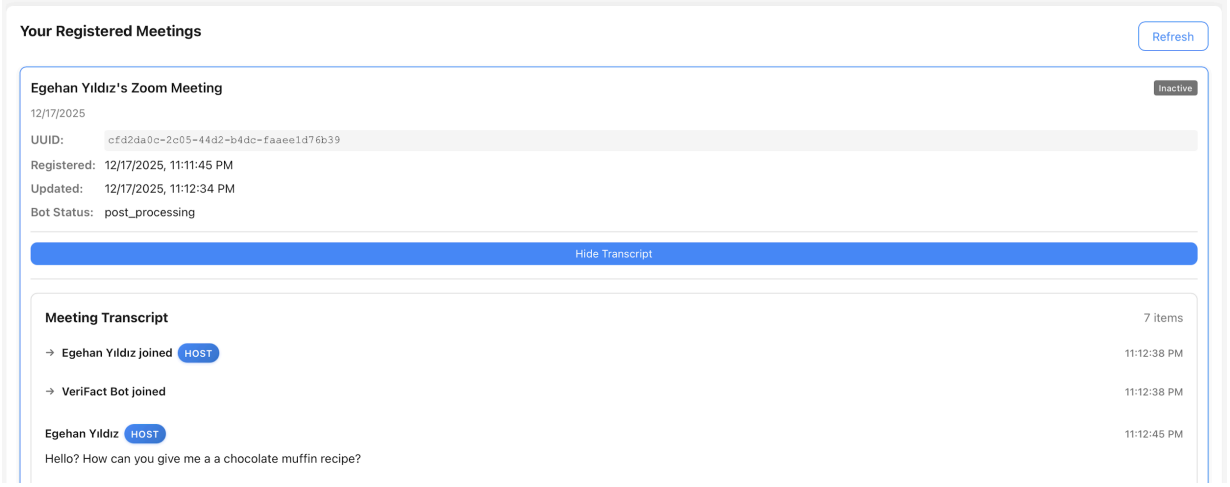


**Figure 20:** Preview Meetings and Transcripts Shown

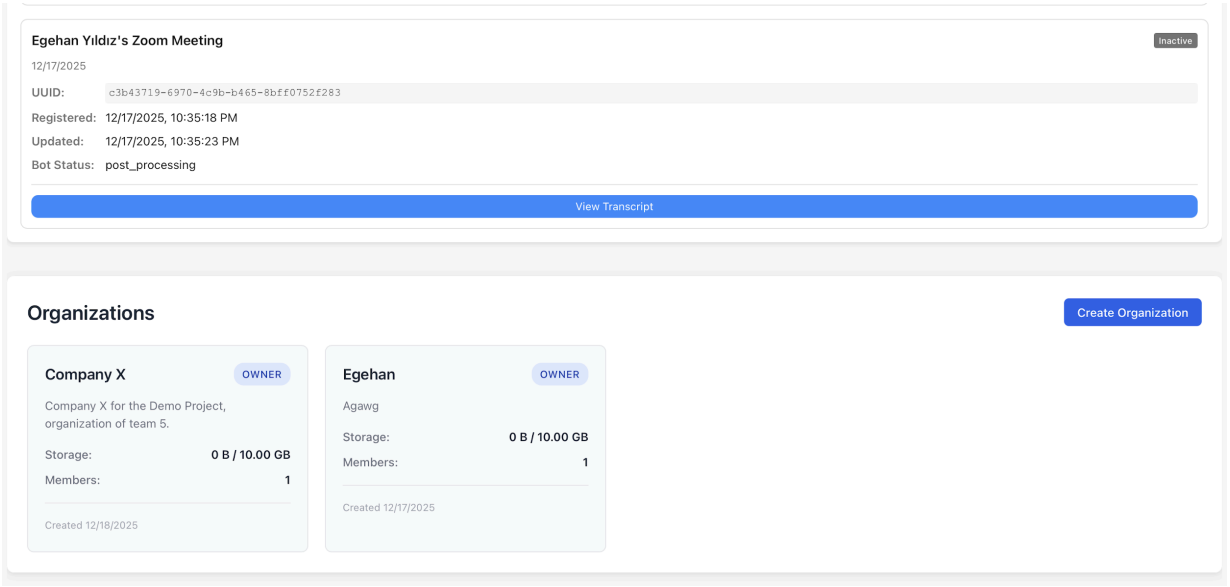


**Figure 21:** Organization Section within Lobby View

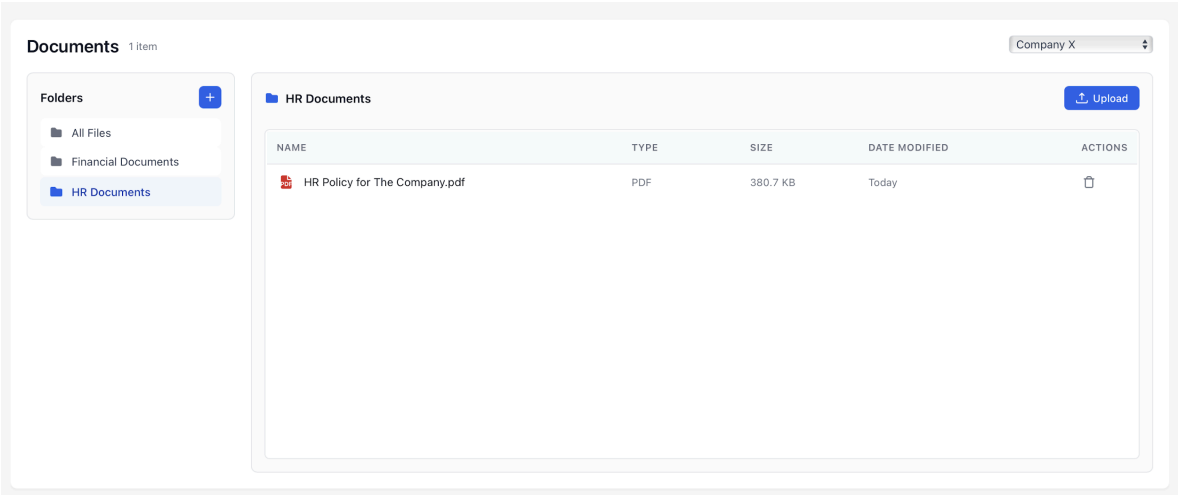


**Figure 22:** Documents Section Inside the Lobby View

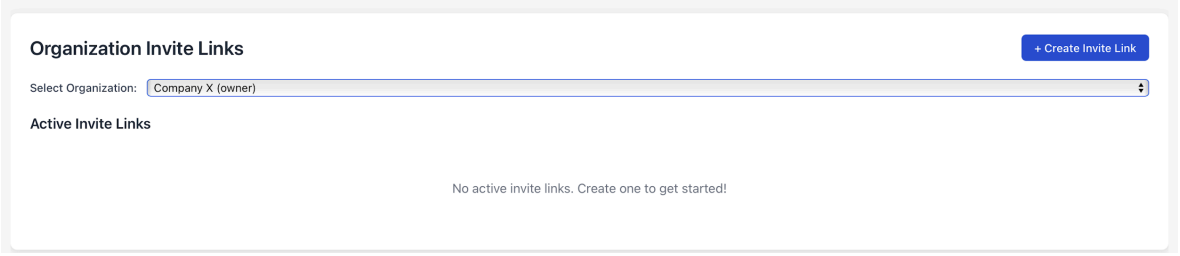


**Figure 23:** Organization Invite Link Creation Section

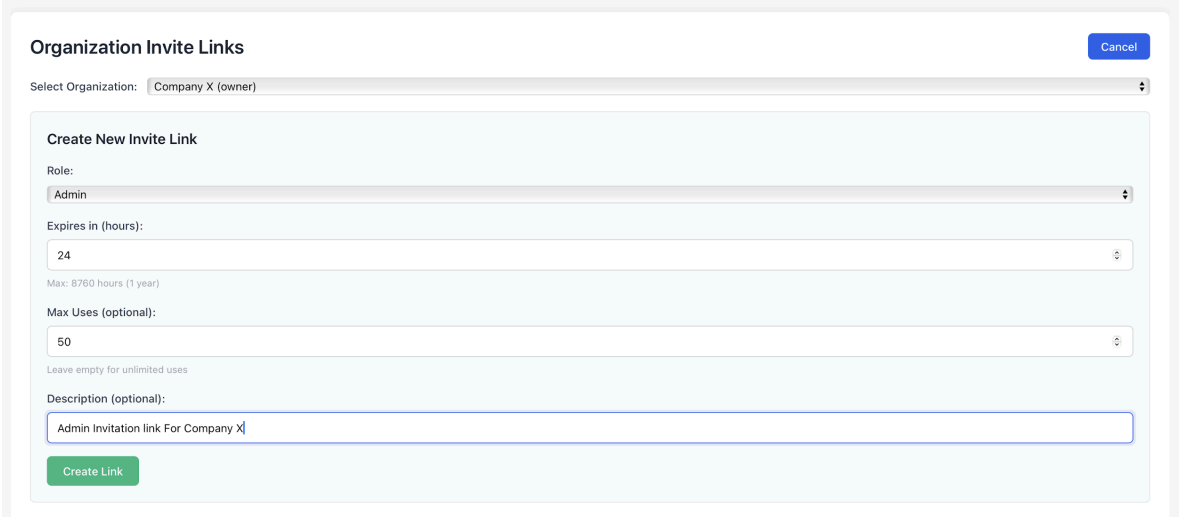


**Figure 24:** Organization Invite Link Creation in Progress

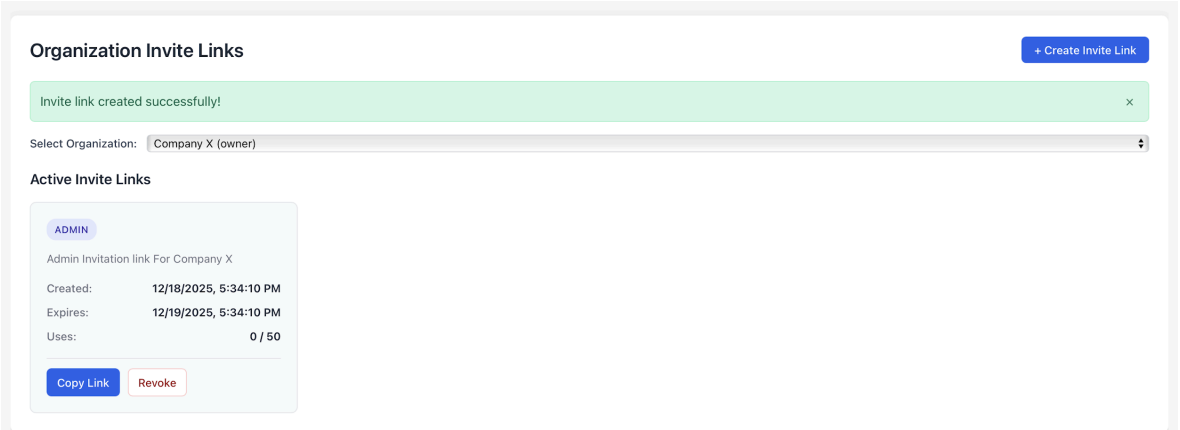


**Figure 25:** Organization Invite Link Created

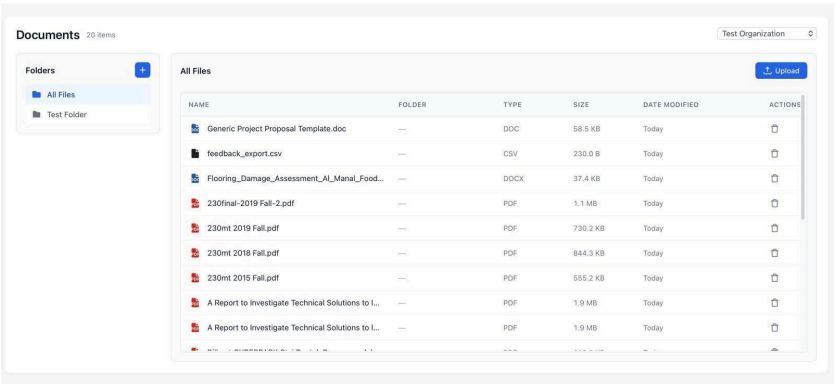


**Figure 26:** Documents Part Filled with Few Documents for RAG Purposes

### 3.5.5.4 LobbyView UIs - Future Work

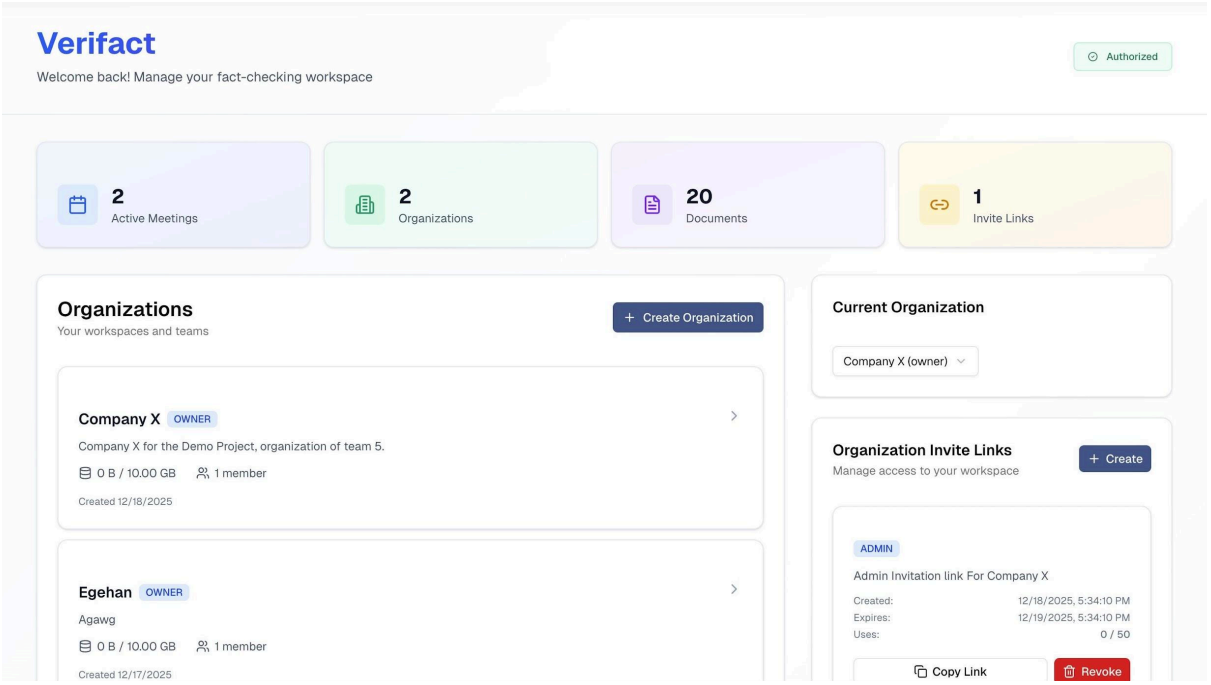


**Figure 27:** Possible Further Advancements #1 on Lobby View

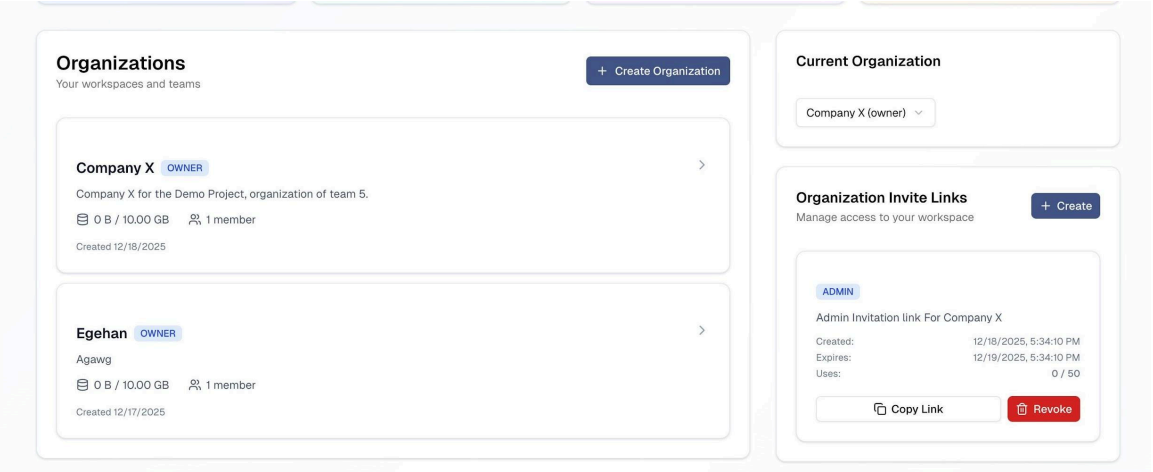


**Figure 28:** Possible Further Advancements #2 on Lobby View

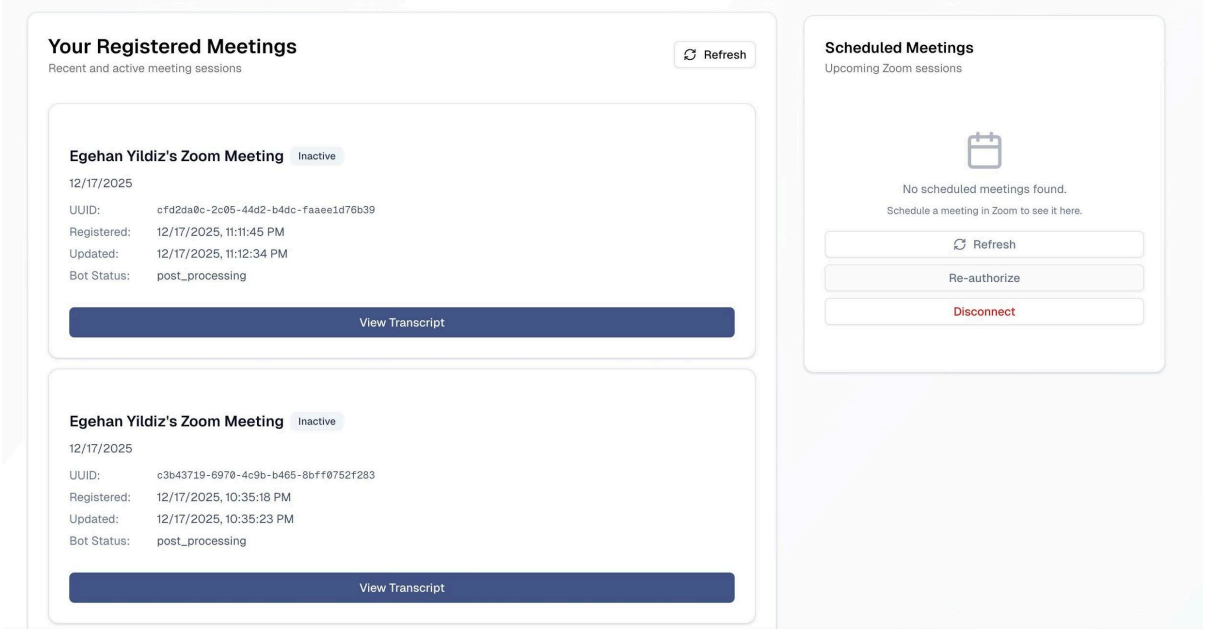


**Figure 29:** Possible Further Advancements #3 on Lobby View

### 3.5.5.5 Post-Meeting UIs - Future Work

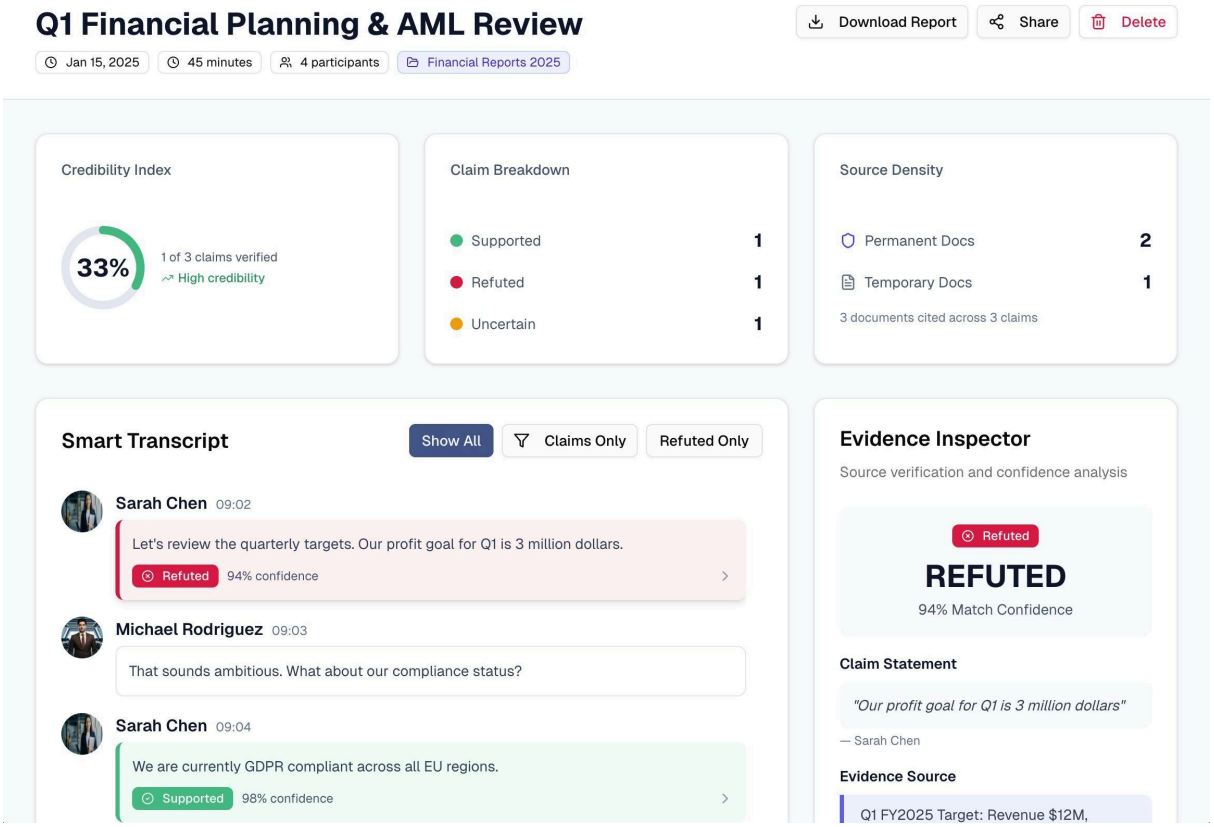


**Figure 30:** Admin View of Post-Meeting Section within VeriFact #1 (will be integrated into Lobby View)

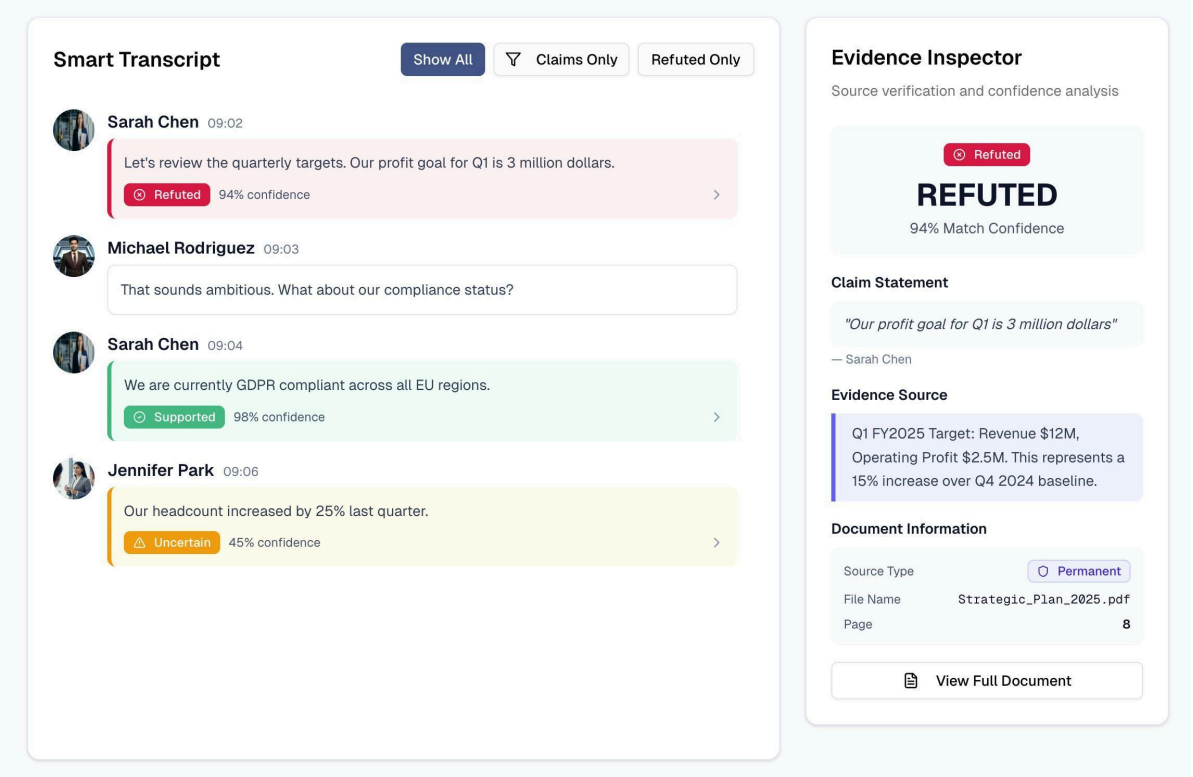


**Figure 31:** Admin View of Post-Meeting Section within VeriFact #2 (will be integrated into Lobby View)

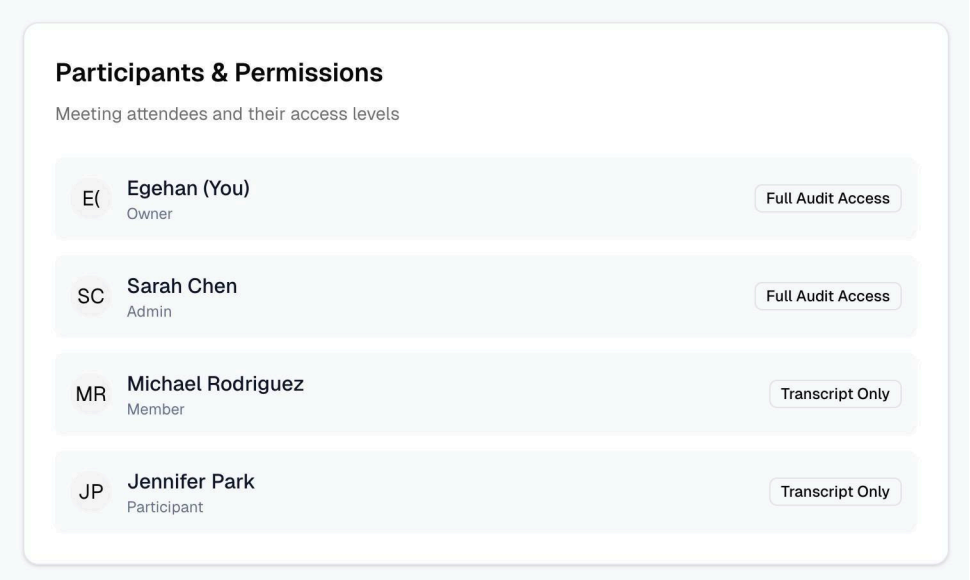


**Figure 32:** Participants and Their Permissions Documented After the Meeting has Ended.

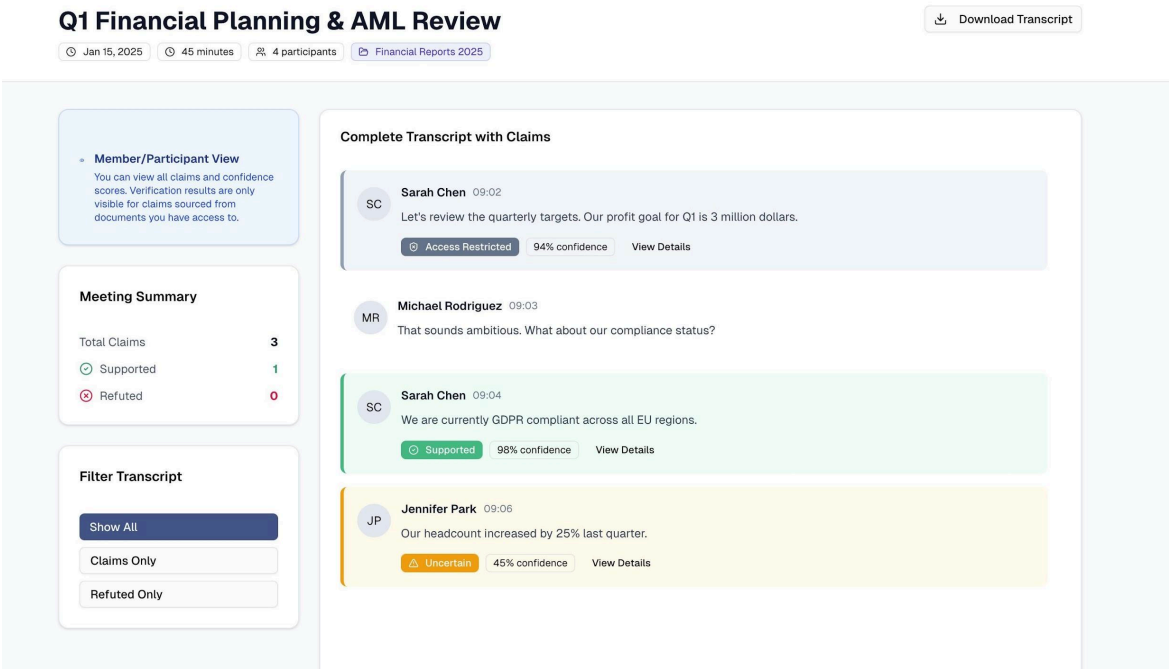


**Figure 33:** Post-Meeting Screen for Members/Participants (with an indicator of “Access Restriction

# 4 Other Analysis Elements

## 4.1 Consideration of Various Factors in Engineering Design

### 4.1.1 Constraints

The design and implementation of VeriFact are governed by a set of technical, ethical, legal, and practical constraints. Since the system operates in real time during live meetings, performance and latency constraints directly influence model selection, system architecture, and pipeline design. All verification outputs must be produced quickly while remaining interpretable and reproducible, particularly in sensitive domains such as healthcare and law.

Privacy and regulatory constraints, including data protection regulations such as GDPR, play a central role in shaping the architecture. VeriFact follows a data minimization approach, storing only what is necessary, enforcing encryption at rest and in transit, and ensuring that temporary meeting data is removed after sessions. These constraints limit data retention strategies and prohibit secondary use of data without explicit consent.

Social and organizational constraints also affect system behavior. As an automated verification tool, VeriFact may influence meeting dynamics and user interactions. To address this, the system avoids surveillance-oriented designs and presents verification results with supporting evidence excerpts rather than authoritative judgments. Economic constraints further restrict the use of expensive cloud infrastructure and large-scale GPU experimentation, encouraging lightweight models and efficient, scalable components. More details available in Table 18.

### 4.1.2 Standards

The VeriFact project follows established software engineering, security, AI, and usability standards to ensure correctness, reliability, and compliance throughout design and implementation.

- IEEE 830 / IEEE 29148 (Requirements Engineering): Used to structure functional and non-functional requirements so that they are complete, testable, and traceable across the system lifecycle [5].
- IEEE 1016 (Software Design Descriptions): Applied to document system architecture, modules, interfaces, and data flows in a consistent and industry-recognized format [6, 7].
- UML 2.5.1 (Unified Modeling Language): Used for use case, component, and sequence diagrams to ensure standardized system modeling [8].
- IEEE 829 / IEEE 29119 (Software Test Documentation): Referenced to structure test plans and reports and link requirements to verifiable test cases [9].
- GDPR (General Data Protection Regulation): Guides user consent handling, data retention policies, and deletion of temporary meeting data.
- ISO/IEC 27001 (Information Security Management): Used to manage risks related to document storage, user data, and backend access control [10].
- ISO/IEC 27018 (Protection of PII in Clouds): Applied to ensure safe handling of personally identifiable information in cloud environments [11].
- OWASP ASVS (Application Security Verification Standard): Referenced for securing APIs through authentication, input validation, and protection against common web vulnerabilities [12].
- ISO/IEC 23894 (AI Risk Management): Used to identify and mitigate AI-related risks such as bias and misclassification [13].

- NIST AI Risk Management Framework: Guides transparency, reliability, and human oversight practices in AI-driven verification workflows [14].
- ML Reproducibility Best Practices: Ensures reproducible model training through versioned datasets, pipelines, and model checkpoints [15].
- TLS 1.2+ (Encrypted Communication): Ensures secure communication between the Zoom bot, backend services, and storage layers.
- AES-256 (Encryption at Rest): Used to encrypt stored documents, embeddings, and metadata [4].
- S3-Compatible Object Storage Conventions: Used to enforce lifecycle rules for document expiration and retention.
- ISO 9241-210 (Human-Centered Design): Followed to reduce cognitive load and ensure fact-checking alerts support, rather than disrupt, meeting interactions [16, 17].
- Zoom App Framework UI Consistency Standards: Ensures the interface integrates seamlessly with native Zoom controls and user expectations.

### 4.2 Risks and Alternatives

Table 17: Risks

| Risk Summary | Likelihood | Effect on the project | B Plan Summary |
|---|---|---|---|
| Real-time latency exceeds acceptable limits | Medium | Delayed claim verification may reduce visibility and usability during live meetings. | Balance between a light-enough model that provides results sufficiently correct. In addition, caching document embeddings will be considered. |
| Claim misclassification (false positive / false negative) | Medium | Incorrect verification results may reduce the user's trust in the system. This affects the quality | Clearly present confidence scores and evidence excerpts. Ensuring that uncertain cases are reflected as NEI. |

| | | | |
|---|---|---|---|
| | | of decision making based on verdicts. | |
| Zoom SDK [3] or API Limitations Change | Low | Core functionalities such as audio capture or in-meeting UI integration may be changed by the Zoom team. | Closely follow Zoom SDK deprecation notices, updates, and developer notice emails. We will maintain a modular integration layer to easily adapt to any changes made. |

### 4.3 Project Plan

Table 18: Factors that can affect analysis and design.

| | Effect level | Effect |
|---|---|---|
| Public health | High | Misleading verification outputs may influence clinical discussions or policy discussions. This requires a cautious and detailed view of verification results, for it to be verifiable and reproducible. |
| Public safety | High | In legal and regulatory contexts, incorrect verification outputs could lead to unlawful decisions. To mitigate this, the system ensures to provide evidence excerpts and use RAG to minimize hallucinations. |
| Public welfare | Medium | Organizational trust and meeting outcomes may be affected. False positives could disrupt collaboration or reduce confidence in using the software. |
| Global factors | Medium | Regulations such as GDPR, regarding data processing and retention, are constraints that influence the architecture choices |

| | | |
|---|---|---|
| | | related to encryption and data minimization (store what you need style approach). |
| Cultural factors | Low | Cultural factors are not too considerable, especially since VeriFact is only planned to support English. |
| Social factors | Medium | The presence of VeriFact, being an automatic verification system, may alter the interaction between participants and presenters. Therefore, VeriFact's goal is to avoid surveillance and help provide more transparent information. |
| Environmental factors | Low | Since we plan to host our infrastructure on the cloud at a later stage, the environmental and energy consumption can be considered. We will ensure our models are not too overpowered for our use cases, as well as having our system scale independently to not waste any resources. |
| Economic factors | Medium | Project budget limitations restrict use of expensive GPU compute and large-scale experimentation. |

Table 19: List of work packages

| WP# | Work package title | Leader | Members involved |
|---|---|---|---|
| WP1 | Requirements Finalization & System Integration Design | Alhassan Raad Jassim Al-badri | Orhun Ege Çelik, Egehan Yıldız |

| | | | |
|---|---|---|---|
| WP2 | Zoom Integration & Audio Ingestion Pipeline | Orhun Ege Çelik | Alhassan Raad Jassim Al-badri, Eray İşçi |
| WP3 | Claim Detection & Verification Pipeline Integration | Egehan Yıldız | İrem Damla Karagöz |
| WP4 | Document Management & Organization Infrastructure | Eray İşçi | Orhun Ege Çelik |
| WP5 | Frontend UI & Real-Time Visualization | İrem Damla Karagöz | Alhassan Raad Jassim Al-badri |
| WP6 | Testing, Evaluation & Finalization | Egehan Yıldız | All Members |

Table 20: Complete details of work packages

| **WP 1:** *Requirements Finalization & System Integration Design* | | | |
|---|---|---|---|
| **Start date:** Week 1 **End date:** Week 3 | | | |
| **Leader:** | Alhassan Raad Jassim Al-badri | **Members involved:** | Orhun Ege Çelik, Egehan Yıldız |
| **Objectives:** The objective of this work package is to finalize system requirements and align existing components into a coherent end-to-end architecture. Since several core features are already implemented, this package focuses on refining interfaces, defining pipeline boundaries, and resolving integration assumptions. It also ensures consistency between functional requirements, ethical constraints, and system behavior. | | | |
| **Tasks:**<br>**Task 1.1: Requirements Refinement** – Review and finalize functional and non-functional requirements based on the current implementation status.<br>**Task 1.2: Pipeline Definition** – Define the final data flow between Zoom integration, STT, claim detection, verification, and UI layers.<br>**Task 1.3: Integration Planning** – Identify missing links between existing components and define integration milestones. | | | |

| | | | |
|---|---|---|---|
| **Deliverables**<br>**D1.1:** Finalized Requirements Specification<br>**D1.2:** Updated System Architecture Diagram | | | |
| **WP 2:** *Zoom Integration & Audio Ingestion Pipeline* | | | |
| **Start date:** *Week 3* **End date:** *Week 6* | | | |
| **Leader:** | Orhun Ege Çelik | **Members involved:** | Eray İşçi |
| **Objectives:** This work package focuses on stabilizing and extending the existing Zoom integration. The goal is to ensure reliable meeting joining, audio capture, and event handling under real-time conditions. Emphasis is placed on robustness, speaker identification, and clean audio streaming to downstream components. | | | |
| **Tasks:**<br>**Task 2.1: Zoom Bot Stabilization** – Improve reliability of bot join/leave behavior and meeting lifecycle handling.<br>**Task 2.2: Audio Stream Normalization** – Ensure captured audio is consistently formatted and streamed to the STT module.<br>**Task 2.3: Event Handling** – Capture participant and meeting events for synchronization with the pipeline. | | | |
| **Deliverables**<br>**D2.1:** Stable Zoom Bot Integration<br>**D2.2:** Audio Ingestion Interface | | | |
| **WP 3:** *Claim Detection & Verification Pipeline Integration* | | | |
| **Start date:** *Week 6* **End date:** *Week 8* | | | |
| **Leader:** | Egehan Yıldız | **Members involved:** | İrem Damla Karagöz |
| **Objectives:** This work package integrates the already available claim detection and claim verification components into a single real-time pipeline. While claim verification exists as an MVP, the focus here is on connecting it to live transcripts, evidence retrieval, and result streaming. Performance and latency constraints are central concerns. | | | |
| **Tasks:** | | | |

| | | | |
|---|---|---|---|
| **Task 3.1: Claim Detection Integration** – Connect the claim detection model to live STT output.<br>**Task 3.2: Verification Pipeline Wiring** – Integrate the existing claim verification MVP into the main pipeline.<br>**Task 3.3: Latency Optimization** – Optimize pipeline execution to meet real-time constraints. | | | |
| **Deliverables**<br>***D3.1:*** Integrated Claim Detection Module<br>***D3.2:*** End-to-End Claim Verification Pipeline | | | |
| **WP 4:** *Document Management & Organization Infrastructure* | | | |
| **Start date:** *Week 8* **End date:** *Week 12* | | | |
| **Leader:** | Eray İşçi | **Members involved:** | Orhun Ege Çelik |
| **Objectives:** This work package extends the partially implemented organization and document management flow. The goal is to transition from local storage to a structured, secure storage model and enable document usage within the verification pipeline. Access control and organization-level separation are key priorities. | | | |
| **Tasks:**<br>**Task 4.1: Organization Flow Completion** – Finalize organization creation, invitations, and role handling.<br>**Task 4.2: Document Storage Integration** – Replace local storage with a scalable object storage solution (e.g., S3-compatible).<br>**Task 4.3: Document Indexing** – Enable document embedding and indexing for retrieval during verification. | | | |
| **Deliverables**<br>**D4.1:** Organization & Permission Management Module<br>**D4.2:** Secure Document Storage & Indexing System | | | |
| **WP 5:** *Frontend UI & Real-Time Visualization* | | | |
| **Start date:** *Week 12* **End date:** *Week 14* | | | |
| **Leader:** | İrem Damla Karagöz | **Members involved:** | Alhassan Raad Jassim Al-badri |

| | | | |
|---|---|---|---|
| **Objectives:** This work package focuses on presenting system outputs to users in a clear and non-intrusive manner. The UI will display live transcripts, detected claims, verification results, and evidence while respecting permission constraints. The design prioritizes usability within the Zoom side panel. | | | |
| **Tasks:**<br>**Task 5.1: Transcript & Claim Visualization** – Render live transcripts and detected claims in real time.<br>**Task 5.2: Verification Result Display** – Show verification status, confidence, and evidence snippets.<br>**Task 5.3: Q&A Interface Integration** – Integrate the in-meeting Q&A interaction flow. | | | |
| **Deliverables**<br>***D5.1:*** Zoom-Embedded Frontend Interface<br>***D5.2:*** Real-Time Visualization Components | | | |
| **WP 6:** *Testing, Evaluation, Iteration & Finalization* | | | |
| **Start date:** *Week 14* **End date:** *Week 18* | | | |
| **Leader:** | Egehan Yıldız | **Members involved:** | All Members |
| **Objectives:** The objective of this work package is to validate the system against functional, performance, and ethical requirements. It includes testing under realistic meeting scenarios, measuring latency and accuracy, and preparing the final documentation and demonstration. | | | |
| **Tasks:**<br>**Task 6.1: Functional & Integration Testing** – Verify correctness of end-to-end workflows.<br>**Task 6.2: Performance Evaluation** – Measure latency, throughput, and reliability.<br>**Task 6.3: Documentation & Presentation** – Prepare final reports and project presentation. | | | |
| **Deliverables**<br>**D6.1:** Test & Evaluation Report<br>**D6.2:** Final Project Report and Demo | | | |

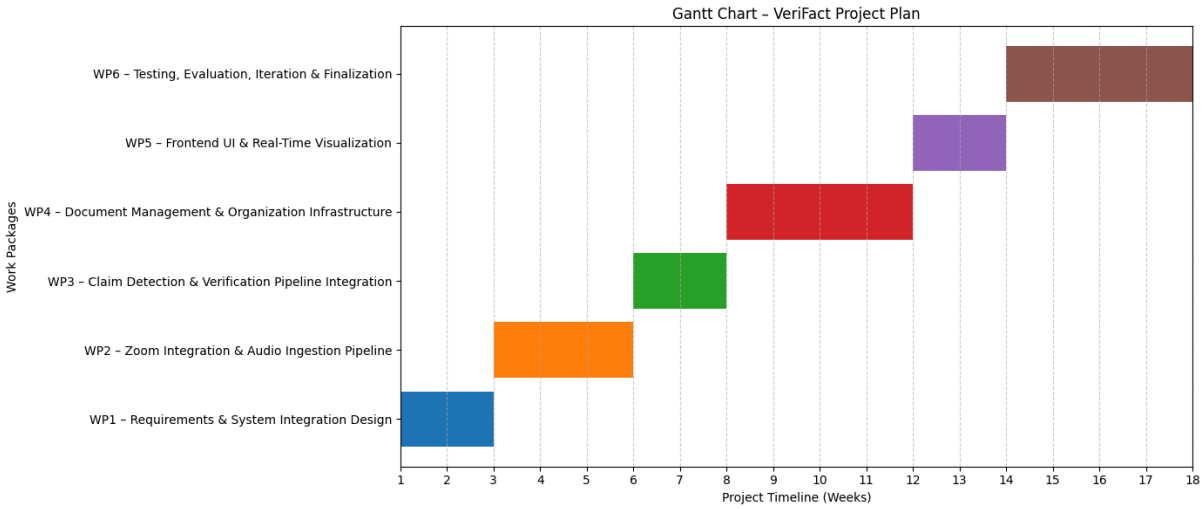


**Figure 34:** Gantt Chart of VeriFact.

## 4.4 Ensuring Proper Teamwork

Effective teamwork in the VeriFact project is ensured through a clear division of roles and responsibilities aligned with defined work packages. Each work package has an assigned leader responsible for coordination, technical decisions, and progress tracking, while other members contribute to implementation and review tasks. Regular team meetings are planned to synchronize development efforts, discuss integration challenges, and review milestones. Version control systems and shared documentation are used to support collaboration, trace changes, and prevent integration conflicts. This structured approach ensures accountability while maintaining flexibility for cross-support when needed.

## 4.5 Ethics and Professional Responsibilities

The team acknowledges its professional responsibility to design and implement VeriFact in an ethical, transparent, and responsible manner. Since the system processes live speech and potentially sensitive documents, strict attention is given to user consent, privacy, and data protection principles. Verification results are presented as probabilistic assistance rather than authoritative judgments to avoid misuse or overreliance. The system is explicitly designed to prevent surveillance, manipulation, or unfair evaluation of individuals, aligning with professional computing ethics and applicable data protection regulations.

### 4.6 Planning for New Knowledge and Learning Strategies

The project involves technical domains such as real-time systems, machine learning–based claim verification, and platform-specific integrations that require continuous learning. Team members plan to acquire new knowledge through targeted literature review, experimentation with open-source tools, and incremental prototyping. Learning is integrated into the development process by validating concepts through small-scale implementations before full integration. Knowledge sharing within the team is encouraged through internal discussions and documentation to ensure that acquired expertise benefits the entire project rather than individual components.

## 5 Glossary

**Claim Detection Model:** A machine learning model that determines whether a sentence should be considered a claim. Our system uses a lightweight local model to perform this analysis in real time.

**Confidence Score**: A probability value that reflects the model's certainty in its prediction. This score is used to filter out low-confidence outputs.

**Data Retention Policy:** Rules that specify how long audio, transcripts, and embeddings remain available in memory.

**Embedding:** A numerical vector that represents the meaning of text. These vectors help the system perform semantic search during evidence retrieval.

**Encryption (AES-256):** An encryption algorithm used to encrypt stored documents, embeddings, and metadata at rest, where stored (e.g., in S3-compatible object storage).

**Evidence Retrieval Module:** The part of the system that searches a curated dataset to locate information relevant to a detected claim. It relies on embeddings and similarity search.

**Pipeline:** The full sequence of operations that data passes through. This includes audio streaming, transcription, segmentation, claim detection, retrieval, scoring, and presentation in the user interface.

**RAG (Retrieval-Augmented Generation):** A method that enriches model outputs with information retrieved from a knowledge source. In this project, retrieval is used to find relevant evidence for each detected claim.

**Zoom Bot:** An automated participant in a Zoom meeting that receives audio, processes it, and sends results to the user interface.

**Zoom SDK (Software Development Kit):** The toolkit that enables integration with Zoom and access to audio streams and meeting events [3].

## 6 References

**[1]** “Automated live fact-checking,” ClaimBuster, https://idir.uta.edu/claimbuster/ (accessed Nov. 27, 2025).

**[2]** J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, “Fever dataset,” Fact Extraction and VERification, https://fever.ai/dataset/fever.html (accessed Nov. 27, 2025).

**[3]**“Class ZoomSdk,” ZoomSdk | @zoom/appssdk - v0.16.36, https://appssdk.zoom.us/classes/ZoomSdk.ZoomSdk.html (accessed Nov. 27, 2025).

**[4]** National Institute of Standards and Technology (NIST), “Advanced Encryption Standard (AES),” FIPS PUB 197, Nov. 2001.

**[5]** IEEE Standard 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE, 1998.

**[6]** ISO/IEC/IEEE 29148:2018, *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*, 2018.

**[7]** IEEE Standard 1016-2009, *IEEE Standard for Information Technology — Systems Design — Software Design Descriptions*, IEEE, 2009.

**[8]** Object Management Group (OMG), *Unified Modeling Language (UML) Specification, Version 2.5.1*, Dec. 2017. [Online]. Available: https://www.omg.org/spec/UML/2.5.1/

**[9]** IEEE Standard 829-2008, *IEEE Standard for Software and System Test Documentation*, IEEE, 2008.

**[10]** ISO/IEC/IEEE 29119-1:2013, *Software and Systems Engineering — Software Testing — Part 1: Concepts and Definitions*, 2013.

**[11]** ISO/IEC 27018:2019, *Information Technology — Security Techniques — Code of Practice for Protection of Personally Identifiable Information (PII) in Public Clouds Acting as PII Processors*, 2019.

**[12]** OWASP Foundation, *OWASP Application Security Verification Standard (ASVS) Version 4.0.3*, OWASP, 2021. [Online]. Available: https://owasp.org/ASVS/

**[13]** ISO/IEC 23894:2023, *Information Technology — Artificial Intelligence — Guidance on Risk Management*, 2023.

**[14]** National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST, Jan. 2023.

**[15]** P. Pineau *et al.*, “Improving reproducibility in machine learning research,” *arXiv preprint arXiv:2003.12206*, 2020.

**[16]** T. Gebru *et al.*, “Datasheets for datasets,” *Communications of the ACM*, vol. 64, no. 12, pp. 86–92, 2021.

**[17]** ISO, *ISO 9241-210:2019 — Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*. International Organization for Standardization, Geneva, Switzerland, 2019.