Calls Craft

Project Team

M.Faizan ul Haq Sheikh 21I-1771
Abdullah bin Faisal 21I-1183
Mubashir Hassan 21I-1764

Session 2021-2025

Supervised by

Mr. Pir Sami Ullah Shah

Department of Computer Science

National University of Computer and Emerging Sciences
Islamabad, Pakistan

December, 2024

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Have you ever presented a customer a pitch with which you eventually were unsuccessful in convincing them to invest in and later figured out that the customer got distracted during the presentation or lost interest halfway through? Calls Craft is here to help you. Call Craft aids users in gathering up the customer's body language and helps them by delivering significant evaluation that aids in making their own pitch adjust to the customer's dynamic perspectives.

1.1 Existing Solutions

In a world full of different solutions and innovations, Callscraft is a unique solution specially designed for 'Sales Representative' to be able to increase their sales through analytics. Online Meeting portals like "Google Meet" or "Microsoft Teams" lack the feature of analyzing the facial and audio features of the users to show each other important analysis such as sentiments, attention metrics and more. Calls Craft combines the power of state of the art AI models with the Zoom SDK to create an all in one solution for sales representatives.

Table 1.1: Comparison of Existing Solutions

System Name	System Overview	System Limitations
Googel Meet	Online meeting platform, provides integration for a limited number of extensions.	Does not provide analytical feedback
Microsoft Teams	Online meeting platform.	Does not provide analytical feedback.

1.2 Problem Statement

In the world's current dynamic landscape, the foremost focus is to transform the way virtual sales presentations transpire. Most sales teams encounter problems in differentiating virtually in the changes of facial expressions and changes in the tone of clients that they would have understood in a physical, face-to-face meeting. Typical video calling platforms like Zoom are not capable of catching up on the change in behavior and tones as they lack the features; this incapability causes the sales team to lose sales, which they would have secured in face-to-face sales pitches.

Customer's body language plays a vital role in conversational settings. In a dialogue where the importance of attention to body language is not implemented, essential details are left out, which eventually creates misunderstandings and inefficient exchanges.

The lack of expertise in this case and the inability to address the switch in customers' body language creates challenges for the sales team to manage their strategy in real time, which ultimately shows the way to foregone opportunities, missed sales, and inefficient interactions. There is a need for an improved integrated solution to the challenge that aids sales teams with immediate insight and suggestions based on clients' behavioral conduct and sentiment evaluation to enhance their strategy and boost sales pitches to make them smoother and more productive.

1.3 Scope

In broader terms, the scope of the project under the proposal is to build a fully functional, detail-oriented, and AI-enhanced platform that integrates with existing popular video conferencing applications like Zoom as well as enhances our internal video conferencing. The platform would employ state-of-the-art behavior and emotion analytics to provide feedback and insights at the time of pitching to help sales teams improve their pitch and the way they engage with the client. The project has multiple different important modules that it shall cover including; integration with video conferencing platforms (our target platform being Zoom), a behavioral and Emotion Analytics Engine (considers Body Alignment, Gaze Direction, Facial Emotions, Speech Emotion, Pitch and Voice Pace to make judgments), Integration with Sales CRM Tool, Interactive Call Guidance System (a fully functional dashboard containing the analytics of the client), Task Management and finally Distraction Analytics (through attention monitoring)

1.4 Modules

The following is a list of modules we shall include in our project:

1.4.1 Call Front-end

The user engaging with the client and being able to see a detailed evaluation of the call they are having with them is the leading element of this project.

1.4.2 Transcript Generation

The users will have the option to access the real-time transcription of the dialogue they are having in the call.

1.4.3 Minutes of Meeting

The users will be able to download the synopsis of the transcription as a PDF file (or other formats)

1.4.4 Notepad

The user will be able to open their notepads and jot down significant information throughout the meeting.

1.4.5 Zoom Call

Utilizing the Zoom API, the meeting between the user and client will be mirrored on the web application for the user along with our entire dashboard.

1.4.6 CRM

CRM to better visualize and display data analytics for the user to review. Also, to maintain the history of all their clients' meetings for the last 30 minutes for efficient and quick usage.

1.5 Work Division

Name	Registration	Responsibility/ Module / Feature
Abdullah bin Faisal	21i-1183	Audio Models Exploration (Module 1.4.1) Frontend Design (Module 1.4.4)Notepad Feature (Module 1.4.6)CRM Integration Software Testing
Muhammad Faizan ul Haq Sheikh	21i-1771	Visual Model Exploration (Module 1.4.5) Zoom Integration with Visual Models (Module 1.4.5) Zoom Integration with Audio Models Server Side Integration
Mubashir Hassan	21i-1764	Textual Model Exploration (Module 1.4.2)Transcript Generation (Module 1.4.3)Minutes of Meeting Feature (Module 1.4.5) Zoom Integration with Textual Models Backend + Database

Table 1.2: Work Division

Chapter 2

Project Requirements

2.1 Use-case

The following includes the Use case diagram along with the High Level Use Case and Expanded Use Cases:

2.1.1 High-Level Use Cases

UC01: Signup	
Use Case	Sign Up
Actors	Sales Representative
Type	Primary
Description	The Sales Representative will sign up for an account on the web application with personal details such as email, password, and username.

Table 2.1: UC01: Signup Use Case

UC02: Login	
Use Case	Login
Actors	Sales Representative
Type	Primary
Description	The Sales Representative will login with their account on the web application with the personal details they provided to access the features and stored data.

Table 2.2: UC02: Login Use Case

UC03: Link Management	
Use Case	Generate Link
Actors	Sales Representative
Type	Primary
Description	The Sales Representative will either generate a meeting link to send the client for them to join the meeting or join a meeting through a link sent to them by their client.

Table 2.3: UC03: Link Management Use Case

UC04: Analytic Visuals	
Use Case	View Analytics
Actors	Sales Representative
Type	Primary
Description	The Sales Representative will view client analytics during the call.

Table 2.4: UC04: Analytic Visuals Use Case

UC05: Show Visualizations	
Use Case	Show Visualizations
Actors	Sales Representative
Type	Secondary
Description	The Sales Representative will be able to visualize the call data after the call.

Table 2.5: UC05: Show Visualizations Use Case

UC06: View Call Transcript	
Use Case	View Call Transcript
Actors	Sales Representative
Type	Secondary
Description	The Sales Representative reads the call transcript during and after the call.

Table 2.6: UC06: View Call Transcript Use Case

UC07: View Call Minutes of Meeting	
Use Case	View Call Minutes of Meeting
Actors	Sales Representative
Type	Secondary
Description	The Sales Representative reads the call Minutes of Meeting after the call. This is a summary of all the important points discussed in the meeting.

Table 2.7: UC07: View Call Minutes of Meeting Use Case

UC08: Notepad	
Use Case	Notepad
Actors	Sales Representative
Type	Secondary
Description	The Sales Representative can use a button to open a notepad to write and store notes during the call.

Table 2.8: UC08: Notepad Use Case

UC09: Read Suggestions	
Use Case	Read Suggestions
Actors	Sales Representative
Type	Primary
Description	The Sales Representative reads the suggestions prompted by the A.I. when certain thresholds of the client's attention span or focus analysis are reached.

Table 2.9: UC09: Read Suggestions Use Case

2.1.2 Expanded Use Cases

UC01: Sign-Up	
Use Case	Sign-Up
Scope	Sales Representative
Level	Primary
Primary Actor	Sales Representative
Stake Holders and Interests	Sales Representative
Preconditions	1) Sales representative must not already have an existing account. 2) The system is available and online. 3) User has internet access.
Postconditions	1) A new account is created and stored in the database. 2) A confirmation email is sent to the sales rep's email address. 3) The sales representative can now log in to the system.
Main Success Scenario	1) Sales Rep opens the sign-up page. 2) Sales Rep enters their username, email, and password. 3) System checks that the email is not already in the database. 4) System validates the entered data (e.g., password strength, email format). 5) System creates the new account. 6) System sends a confirmation email to the provided email address. 7) Sales Rep is informed that the account has been successfully created.
Extensions	3a) Email already exists in the database: System shows an error message and asks the Sales Rep to use a different email. 4a) Weak or invalid password: System prompts the Sales Rep to enter a stronger password. 4b) Invalid email format: System asks the Sales Rep to provide a valid email address. 6a) Confirmation email fails to send: System retries sending the email or prompts the Sales Rep to verify their email address.

Table 2.10: UC01: Sign-Up for Sales Representative

Use Case	Login
Scope	Sales Representative
Level	Primary
Primary Actor	Sales Representative
Stakeholders and Interests	Sales Representative
Preconditions	1) Sales representative has an existing account. 2) The system is available and online. 3) User has internet access.
Postconditions	1) The sales representative is successfully authenticated. 2) The system grants access to the sales representative's dashboard. 3) The system logs the login activity.
Main Success Scenario	1) Sales Rep opens the login page. 2) Sales Rep enters their email and password. 3) System checks if the email exists in the database. 4) System verifies that the password matches the email. 5) Sales Rep is successfully logged in and redirected to their dashboard.
Extensions	3a) Email does not exist in the database: System displays an error message asking the Sales Rep to sign up. 4a) Incorrect password: System shows an error message and prompts the Sales Rep to re-enter the password. 4b) Account is locked due to too many failed login attempts: System notifies the Sales Rep and offers a password recovery option.

Table 2.11: UC02: Login Use Case

Use Case	Link Management
Scope	Sales Representative
Level	Primary
Primary Actor	Sales Representative
Stakeholders and Interests	1) Sales Representative: Wants to effectively manage Zoom call links for meetings. 2) Client: Expects to receive the correct Zoom link and join the meeting seamlessly.
Preconditions	1) The sales representative and client have Zoom accounts. 2) The system is available and online. 3) Both parties have internet access.
Postconditions	1) A Zoom meeting link is successfully shared with the client. 2) The sales representative joins a client-sent Zoom call through a provided link. 3) The system logs meeting link generation and activity.
Main Success Scenario	1) Sales Rep opens the Zoom meeting management page. 2) Sales Rep generates a new Zoom link. 3) Sales Rep sends the Zoom link to the client via email or chat. 4) Client clicks the link to join the meeting. 5) Sales Rep joins the call as the host. OR 1) Sales Rep receives a Zoom link from a client. 2) Sales Rep clicks the link to join the call. 3) System verifies the link's validity. 4) Sales Rep successfully joins the call as a participant.
Extensions	3a) Invalid Zoom link sent by Sales Rep: System shows an error message and prompts Sales Rep to generate a new link. 4a) Client cannot join due to technical issues: Sales Rep resends the Zoom link or offers an alternate method to join. 3a) Invalid link received by Sales Rep: System shows an error message and prompts the Sales Rep to check the link.

Table 2.12: UC03: Link Management Use Case

Use Case	Analytic Visualizations
Scope	Sales Representative
Level	Primary
Primary Actor	Sales Representative
Stakeholders and Interests	1) Sales Representative: Wants real-time analysis of the client to enhance the sales pitch.
Preconditions	1) The client's camera must remain on during the video call. 2) Both the sales rep and client have a stable internet connection. 3) Call Craft's analytic tools are enabled and functioning.
Postconditions	1) The system successfully analyzes the client's video feed. 2) Real-time analysis is displayed next to the video call. 3) The sales representative uses the data to improve their sales strategy.
Main Success Scenario	1) Sales Rep initiates the video call through Call Craft. 2) Client joins the call with their camera turned on. 3) The system analyzes the client's body language, expressions, and other metrics. 4) The analysis is displayed next to the video feed in real-time. 5) Sales Rep uses the insights to adjust the pitch based on client reactions.
Extensions	3a) Client's camera is turned off: System prompts the client to turn on the camera for the analysis to begin. 3b) Video analysis tools encounter an error: System notifies the Sales Rep and attempts to reconnect the analytic feed. 4a) Analysis data not shown to Sales Rep due to network issues: System retries connection and notifies the Sales Rep.

Table 2.13: UC04: Analytic Visuals Use Case

Use Case	Show Visualizations
Scope	Sales Representative
Level	Optional
Primary Actor	Sales Representative
Stakeholders and Interests	1) Sales Representative: Wants to visualize data for better future decision-making.
Preconditions	1) Sales representative has had a prior call with the client.
Postconditions	1) Visualizations are successfully generated and displayed. 2) Sales rep uses the charts to support their sales efforts.
Main Success Scenario	1) Sales Rep attends a call with the client. 2) Analytics get stored in the database. 3) System processes the data and provides visual insights. 4) Sales Rep views the visualizations in the dashboard. 5) Sales Rep uses this information to prepare for future client meetings.
Extensions	3a) Data not stored correctly: System logs the issue and prompts Sales Rep to retry. 4a) Visualizations fail to load: System shows a message and suggests troubleshooting steps.

Table 2.14: UC05: Show Visualizations Use Case

UC06: Call Transcript	
Use Case	Call Transcript
Scope	Sales Representative
Level	Primary
Primary Actor	Sales Representative
Stakeholders and Interests	Sales Representative: Wants to keep track of the conversation for future reference.
Preconditions	1) Call is initiated between the sales representative and client. 2) Both participants' microphones are active during the call.
Postconditions	1) A complete transcript of the call is generated and available for review. 2) Transcript can be downloaded in multiple file formats (e.g., PDF, TXT) after the call ends.
Main Success Scenario	1) Sales Rep starts the video call with the client. 2) The system begins transcribing the conversation in real-time. 3) Sales Rep can view the transcript during the call. 4) After the call, the transcript is finalized and available for download. 5) Sales Rep downloads the transcript in a chosen file format (e.g., PDF, TXT).
Extensions	3a) Transcript stops due to network issues: System notifies the Sales Rep and attempts to restore real-time transcription. 4a) Download format unavailable: System notifies Sales Rep and suggests available formats for download.

Table 2.15: UC06: Call Transcript Use Case

UC07: Minutes of Meeting	
Use Case	Minutes of Meeting
Scope	Sales Representative
Level	Primary
Primary Actor	Sales Representative
Stakeholders and Interests	Sales Representative: Needs a summarized record of key points from the call.
Preconditions	Call between the sales representative and client is completed.
Postconditions	1) Minutes of Meeting (MoM) is generated summarizing the key points. 2) MoM file can be downloaded in multiple formats (e.g., PDF, DOCX).
Main Success Scenario	1) Sales Rep finishes the call with the client. 2) System automatically generates a Minutes of Meeting file. 3) MoM file summarizes important points discussed during the call. 4) Sales Rep reviews the MoM and selects a format for download (e.g., PDF, DOCX). 5) Sales Rep successfully downloads the MoM file.
Extensions	4a) MoM file generation fails due to network issues: System retries generating the file and notifies the Sales Rep. 4b) Download format not available: System suggests alternate available formats for the Sales Rep to choose from.

Table 2.16: UC07: Minutes of Meeting Use Case

UC08: Notepad	
Use Case	Notepad
Scope	Sales Representative
Level	Optional
Primary Actor	Sales Representative
Stakeholders and Interests	Sales Representative: Wants a quick way to take notes during the call.
Preconditions	1) Sales representative is on a call with the client. 2) Sales Rep needs to note down thoughts or action items during the call.
Postconditions	1) Sales Rep successfully writes notes in the notepad. 2) Notes can be saved for future reference or downloaded after the call.
Main Success Scenario	1) Sales Rep clicks the 'Notepad' button during the call. 2) A small notepad interface opens next to the call screen. 3) Sales Rep types notes about the conversation or other thoughts. 4) Notes are automatically saved and can be reviewed after the call. 5) Sales Rep downloads the notes, if needed, after the call.
Extensions	4a) Notepad fails to open due to a system issue: System notifies the Sales Rep and attempts to reload the notepad feature. 5a) Notes download fails: System suggests retrying the download or saving the notes in an alternate format.

Table 2.17: UC08: Notepad Use Case

UC09: Read Suggestions	
Use Case	Read Suggestions
Scope	Sales Representative
Level	Primary
Primary Actor	Sales Representative
Stakeholders and Interests	Sales Representative
Preconditions	1) Sales representative and client are both online on the call. 2) Client has their camera on. 3) Client's Cognitive Resonance Score is below a set threshold.
Postconditions	1) The system prompts the sales rep with suggestive lines to help regain the client's focus. 2) If the cognitive resonance score improves, no more suggestions are prompted. 3) If the score drops, more suggestions are given.
Main Success Scenario	1) Sales Rep initiates a Zoom meeting with the client. 2) System analyzes the client's cognitive resonance score in real-time. 3) System monitors the score during the conversation. 4) Client's score drops below the threshold. 5) System prompts Sales Rep with suggestions to re-engage the client. 6) Sales Rep uses the suggestions. 7) System continues monitoring. 8) Client re-engages and the score rises above the threshold. 9) System records the interaction for post-meeting analysis.
Extensions	1a) Zoom meeting fails to start: System notifies Sales Rep. 3a) System fails to monitor score: Sales Rep continues without suggestions. 5a) System fails to generate suggestions: Sales Rep proceeds without AI guidance.

Table 2.18: UC09: Read Suggestions

UC10: Automatic Checklist	
Use Case	Automatic Checklist
Scope	Sales Representative
Level	Optional
Primary Actor	Sales Representative
Stakeholders and Interests	Sales Representative: Wants to ensure all key points are covered during the call.
Preconditions	1) Sales representative has set a checklist of key points before the call. 2) System's voice analysis tool is active and functioning to track the conversation.
Postconditions	1) Checklist items are automatically ticked off during the call. 2) Sales Rep reviews the checklist to ensure all points are covered.
Main Success Scenario	1) Sales Rep sets the main points/accomplishments to cover during the call. 2) Call starts and the system listens for discussions around the checklist items. 3) System automatically ticks off a point once it detects it has been covered in the conversation. 4) Sales Rep can view the checklist and confirm all points have been discussed. 5) Checklist is saved or exported at the end of the call.
Extensions	3a) System incorrectly ticks off a point: Sales Rep manually unticks the point and can mark it for later discussion. 5a) Export of checklist fails: System suggests an alternative format or retries the export process.

Table 2.19: UC10: Automatic Checklist Use Case

2.1.3 Use Case Diagram

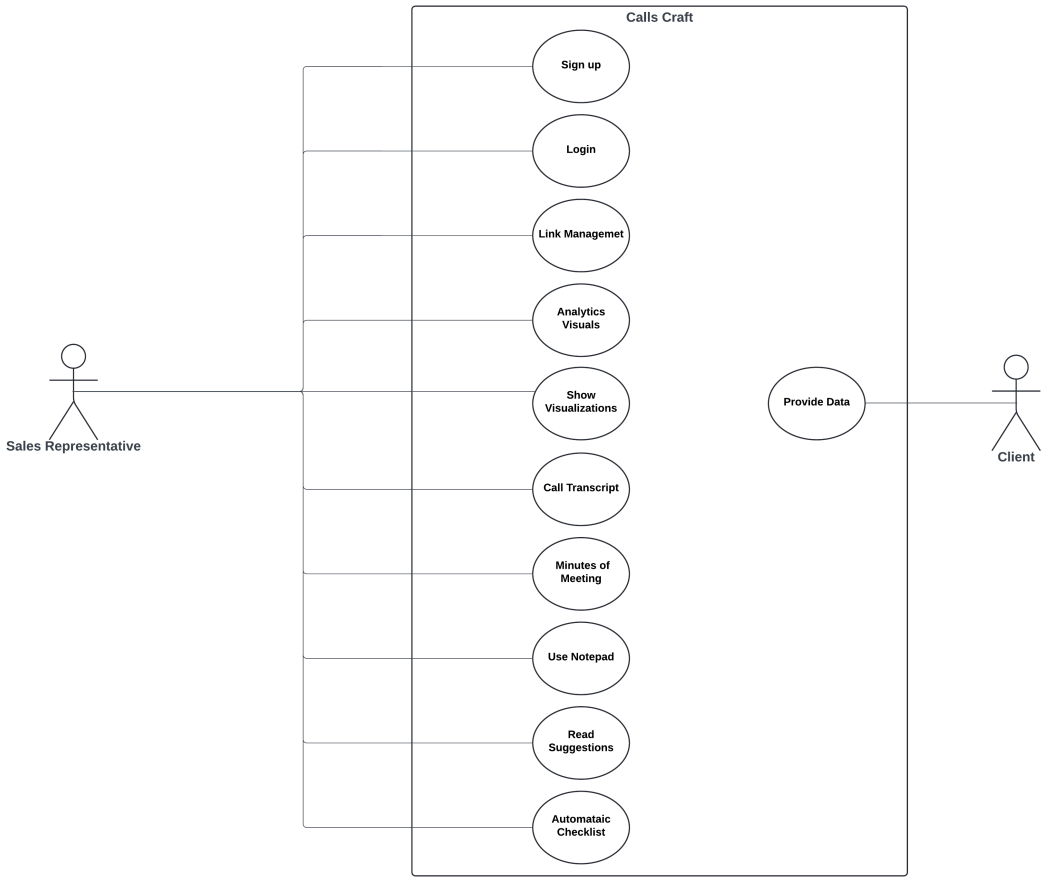

Figure 2.1: Use Case Diagram

2.2 Functional Requirements

2.2.1 Call Front-end

- The system should allow the user to have an interface to use Zoom API to communicate with the client.
- The system shall provide the user with an analytical analysis of facial features.[1][2][5][6]

2.2.2 Transcript Generation

- A real-time transcript of the whole dialogue between the user and client should be delivered to the user by the system.[3][4]
- A downloadable transcript file shall also be handed over to the user.

2.2.3 Minutes of Meeting

- A downloadable file of the transcript shall also be handed over to the user.

2.2.4 Notepad

- An interface for notes should be delivered in the call on the user side, and they should also be able to create personalized notes.

2.2.5 Document Visualizations

- The user will access the analytics of the call in the CRM.
- The user should be provided with the choice to download the visualizations created.

2.3 Non-Functional Requirements

This section specifies nonfunctional requirements:

2.3.1 Reliability

The system shall operate continuously during sales calls with an extremely **low** failure rate.

2.3.2 Usability

User Experience: An effortless user experience, with all features operating efficiently without delays during calls or data analysis to be delivered.

2.3.3 Performance

Response Time: Fast response times to be delivered for important interactions like call setup, data loading, and real-time feedback, securing effectual experience during sales calls.

Latency: The design should be in such a way that delivers **low latency** for real-time data processing during the client's sentiment and behavioral analysis, which also guarantees timely feedback for sales representatives.

2.3.4 Security

Password Hashing: All user passwords shall be stored as hashes using SHA-256 with added salt for enhanced security.

Data Privacy: The system will not store video or audio data but will only store the analytics generated from the ML models. This ensures enhanced **user privacy**.

Logging: An extensive logging system to be provided for all key actions, errors and user activities. Logs are to be maintained for auditing and troubleshooting reasons.

2.3.5 Compatibility

Major desktop-operating systems shall be easily adapted, and performance should be enhanced in these environments.

Chapter 3

System Overview

3.1 Architectural Design

For the architecture of our system, we chose to go with the Model View Controller (MVC) architecture. This is the most used architecture in the modern world due to its simplistic approach and the amount of impact it has during the whole development phase. The following is a general overview of the division of our system among the components of this architecture

Model: This part of the architecture includes the user information, call records, analytics, transcripts, interaction with the database, and external API access

View: This part of the architecture includes the parts of the web app that the user can see including the signup and login pages, and the call analytics front end. This part is only focused on the view part of the project and does not handle any sort of logic.

Controller: This part of the architecture includes the handling of all the inputs and their processing. It gets the data from the "Model" processes it and then proceeds to send it over to the "View" for the user to interact with it. Tasks like starting or ending calls through the Zoom API, processing analytics, and generating insights are all handles here.

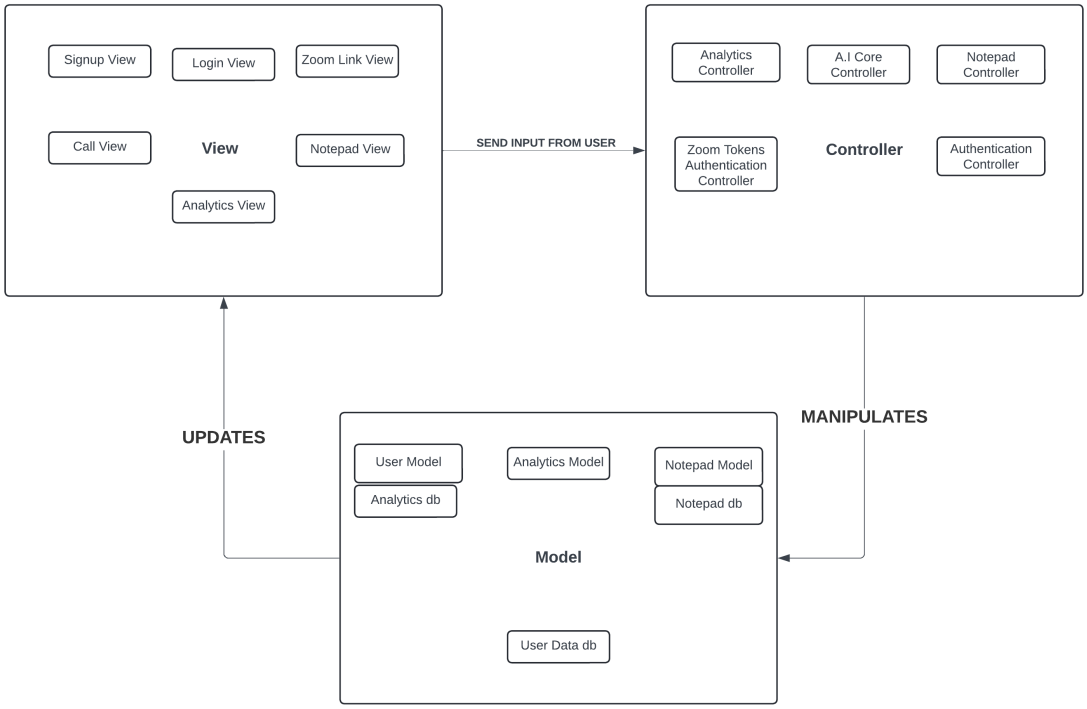

Figure 3.1: Model View Controller (MVC)

3.2 Data Design

The following Entity relationship diagram represents the important interconnectivity amongst multiple different entities in our project.
These range from the call itself, analytics, user (sales representative), CRM, Tasks, Client, and Transcripts. Each entity includes the important and most relevant attributes needed in order to make the system possible such as their IDs. The diagram signifies how the use can place a call and how each call can have a single client and analytics reading the client with other features such as transcripts, tasks and notepad.

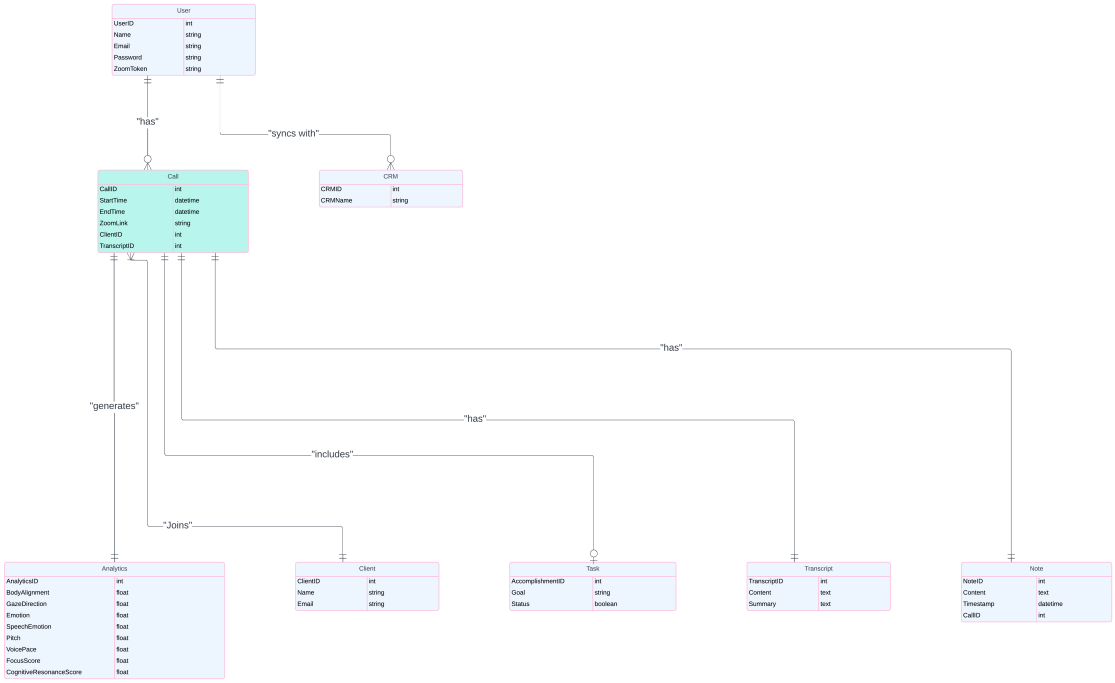

Figure 3.2: ERD

3.3 Domain Model

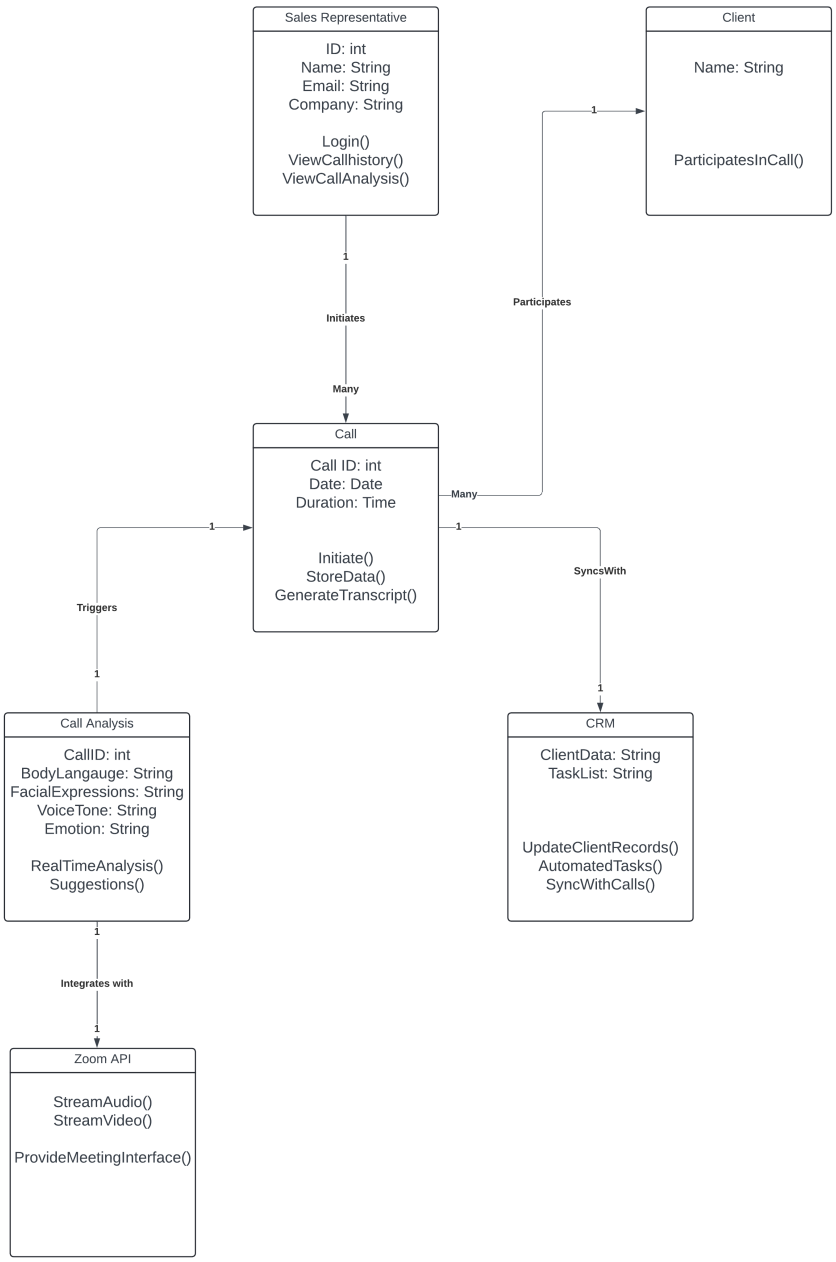

Figure 3.3: Domain Model Diagram

3.4 Design Models

3.4.1 Activity Diagrams:

An activity diagram is a type of Unified Modeling Language diagram that visually represents the workflow of a system or process. It illustrates the sequence of activities, decisions, and parallel actions in a system, helping to understand the dynamic behavior

3.4.1.1 Registration of Sales Representative:

Based on the flow of this diagram, when the user (which is the sales representative) wants to register or login on the system the user will initiate the software.An option will appear whether the user is new or already exists through which the user will decide whether to login to gain access or register as a new user.

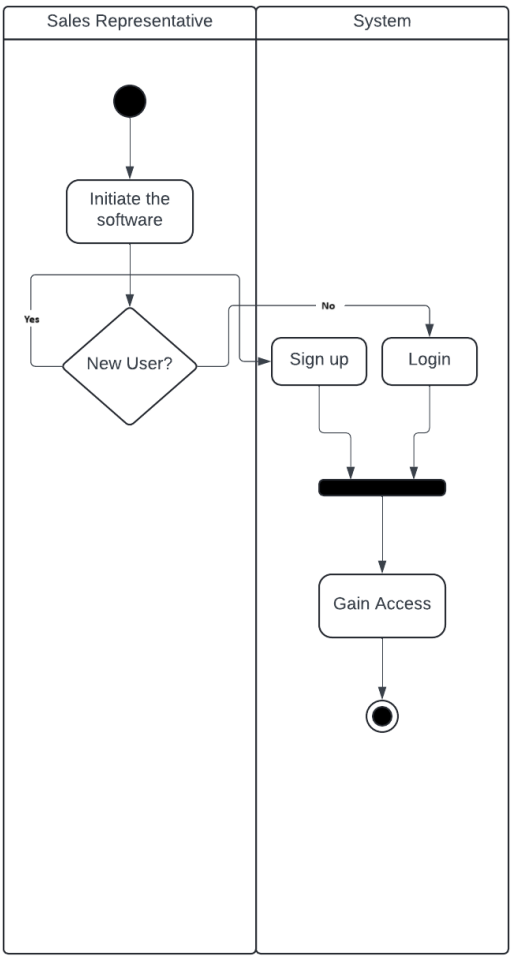

Figure 3.4: User Registration

3.4.1.2 Link Management:

Based on the flow of the diagram, when the user which is the sales representative initiates the software it will be asked whether it's a new user then sign up or else go with login option.Then further there will be two options either the clients shares the link of zoom meeting or user itself generates a link and send it to client for joining.Then the call will start and analysis will begin.

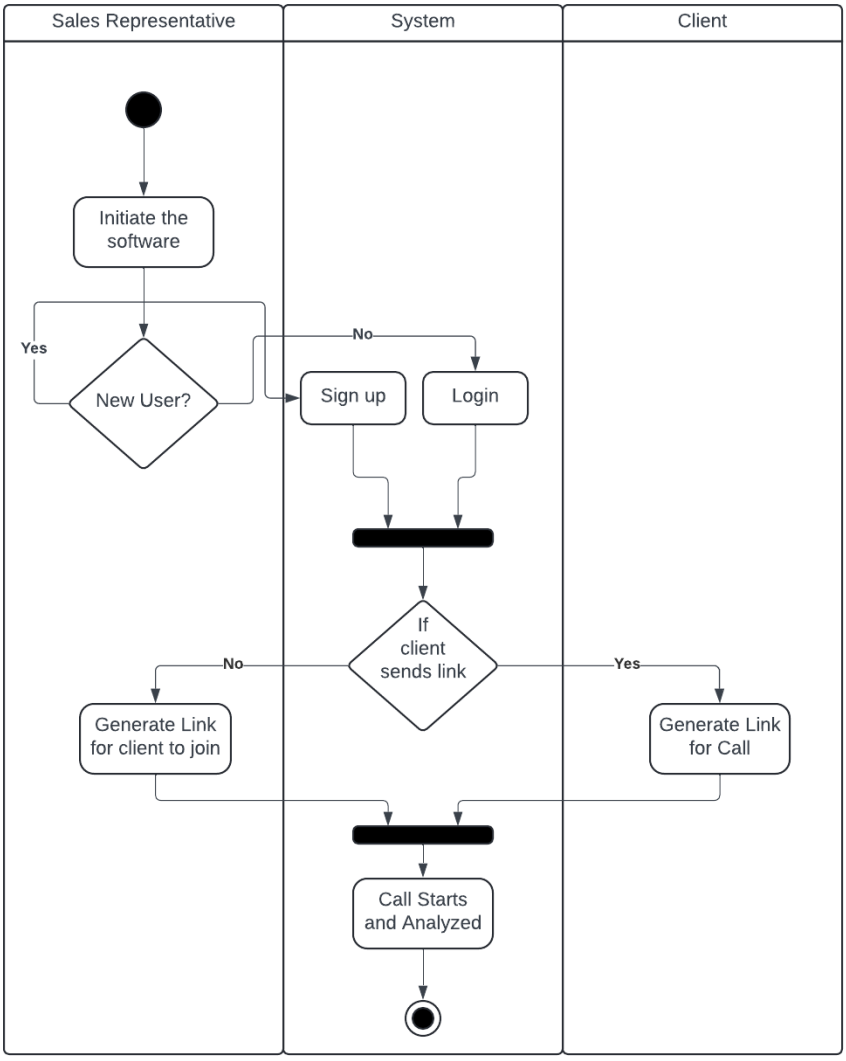

Figure 3.5: Link Management

3.4.1.3 Zoom SDK and Call Setup:

Based on the flow of the diagram, when the user which is the sales representative initiates the software it will be asked whether it's a new user then sign up or else go with login option. When the zoom call is initiates it will connect to Zoom's SDK and if it's a successful link then the sales representative will be notified and it will complete the call setup or otherwise it will notify the user to generate new link.

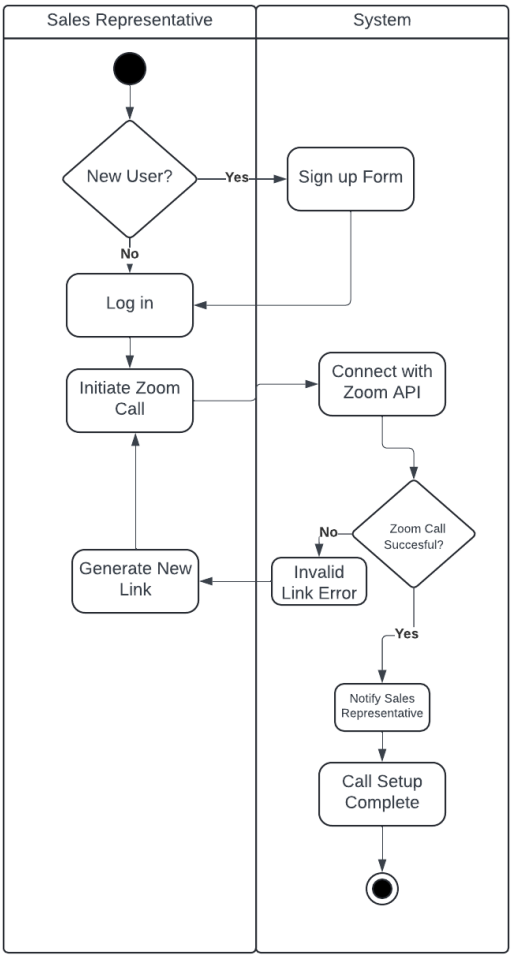

Figure 3.6: Zoom SDK and Call setup

3.4.1.4 On Call Analyses:

Based on the flow of the diagram, when the user which is the sales representative initiates the software it will be asked whether it's a new user then sign up or else go with login option. The user can then access the calls history and data related to it. The call will start and analyzed in which the user will be given two facilities to use notepad and read suggestions given by AI which will help user take crucial actions.

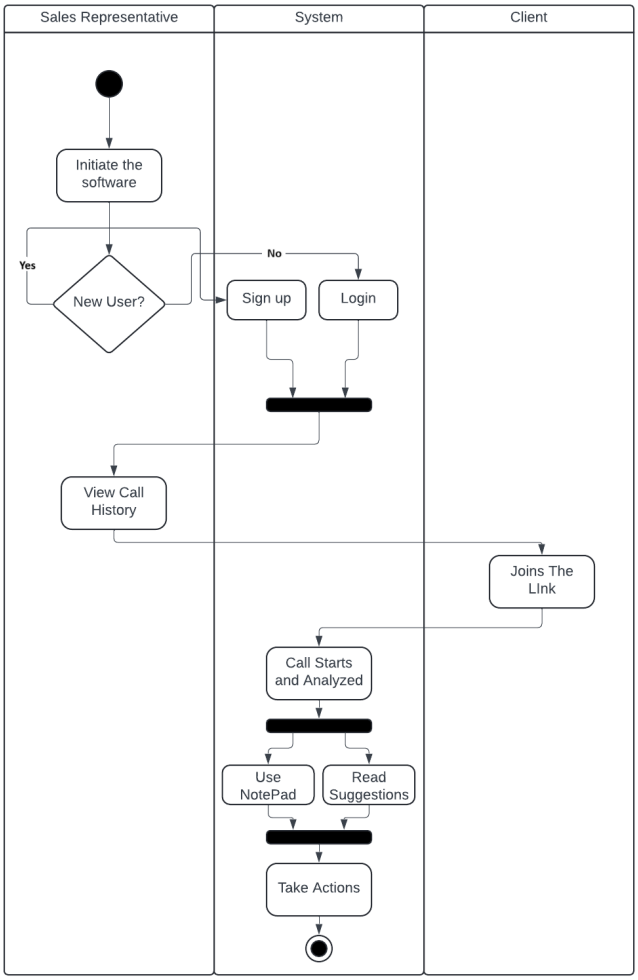

Figure 3.7: On Call Analyses

3.4.1.5 Data Analytics and Visualization:

Based on the flow of the diagram, when the user which is the sales representative initiates the software it will be asked whether it's a new user then sign up or else go with login option.The user can then access the calls history and data related to it. it will further give option to the user to check analytics and visualizations created for that call.

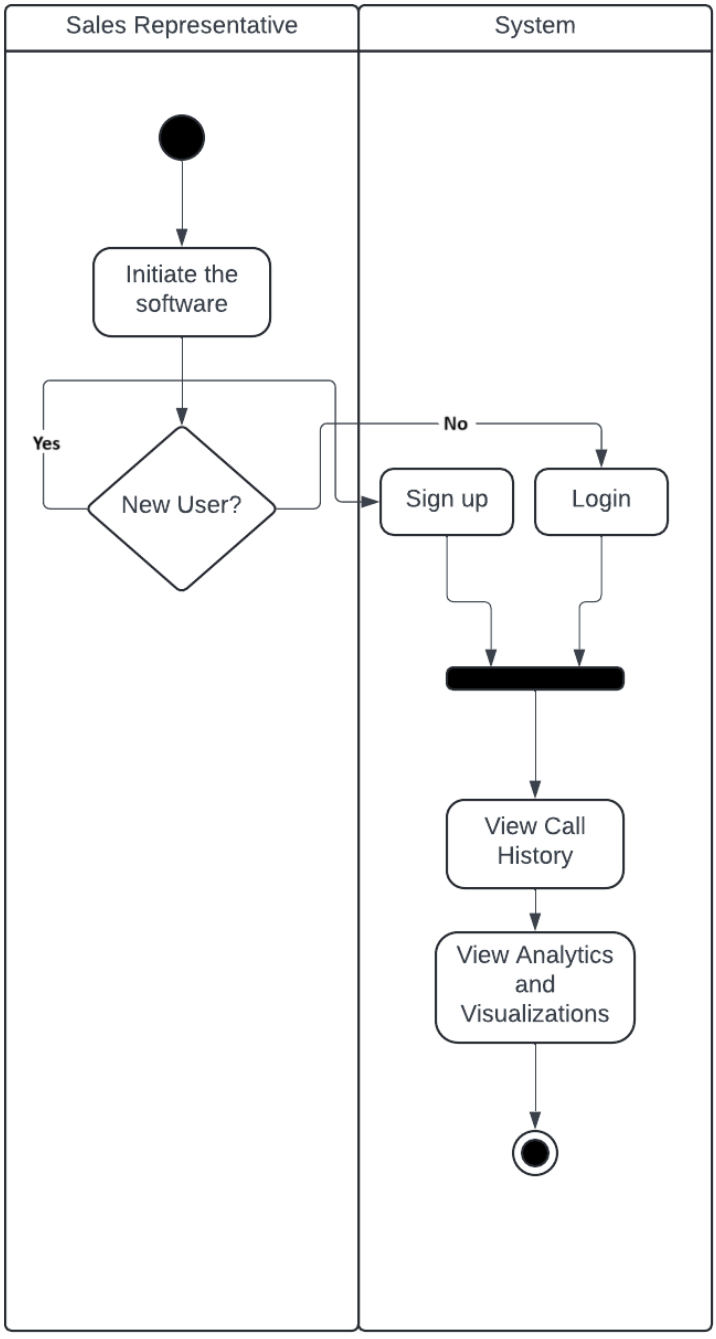

Figure 3.8: Data Analytics and Visualization

3.4.1.6 Real-Time AI Analysis of Client Behavior:

Based on the diagram the sale's representative will initiate the zoom call and the client will join it.The call's audio and video will be analyzed by AI.The focus will be on client's behavior and emotions and will provide real time feedback. if the client seems interested continue sale pitch otherwise provide suggestions to sales representative.when the call ends the crm will sync with the call data and transcript and meeting summary will be generated.

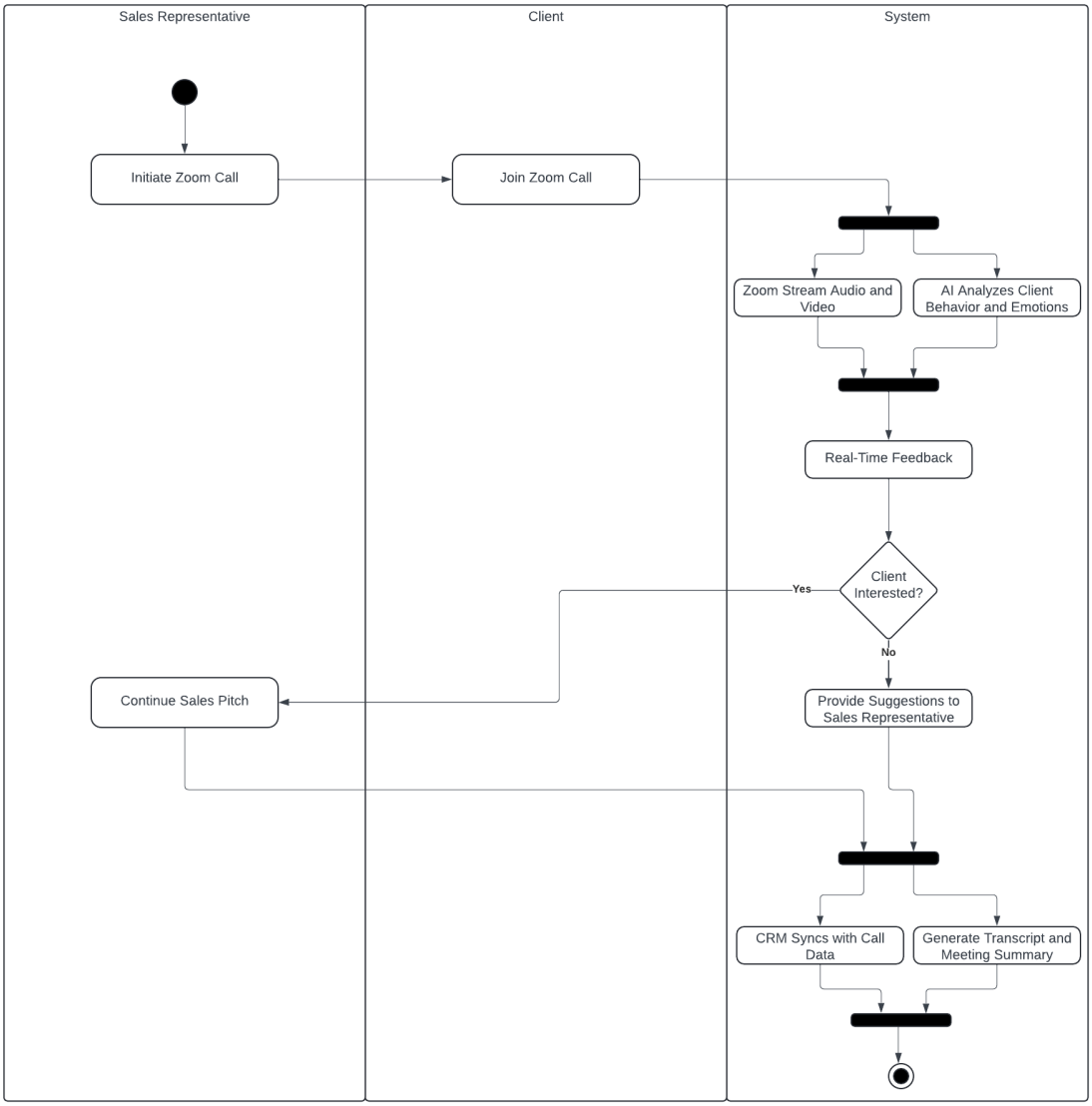

Figure 3.9: Real-Time AI Analysis of Client Behavior

3.4.1.7 Transcript Generation:

Based on the diagram the sales representative will end call.The system will save the recorded audio and video to generate transcript.It will then ask the user if they want summary or not of that call.Later will be stored in CRM.

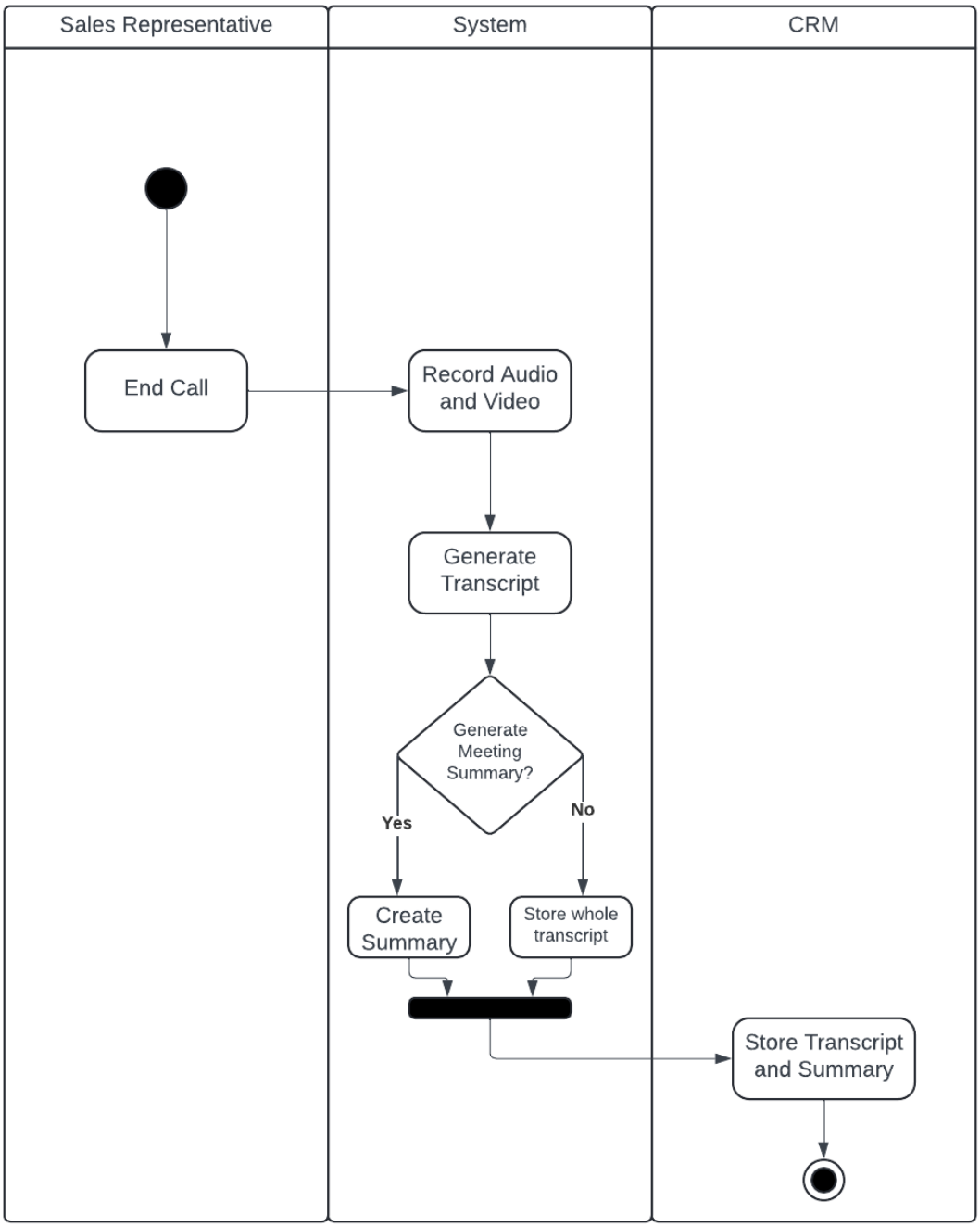

Figure 3.10: Transcript Generation

3.4.1.8 CRM Functionality:

Based on the diagram the sales representative will end call.The CRM will synchronise with call's data and update the clients record which will allow it to create visualizations

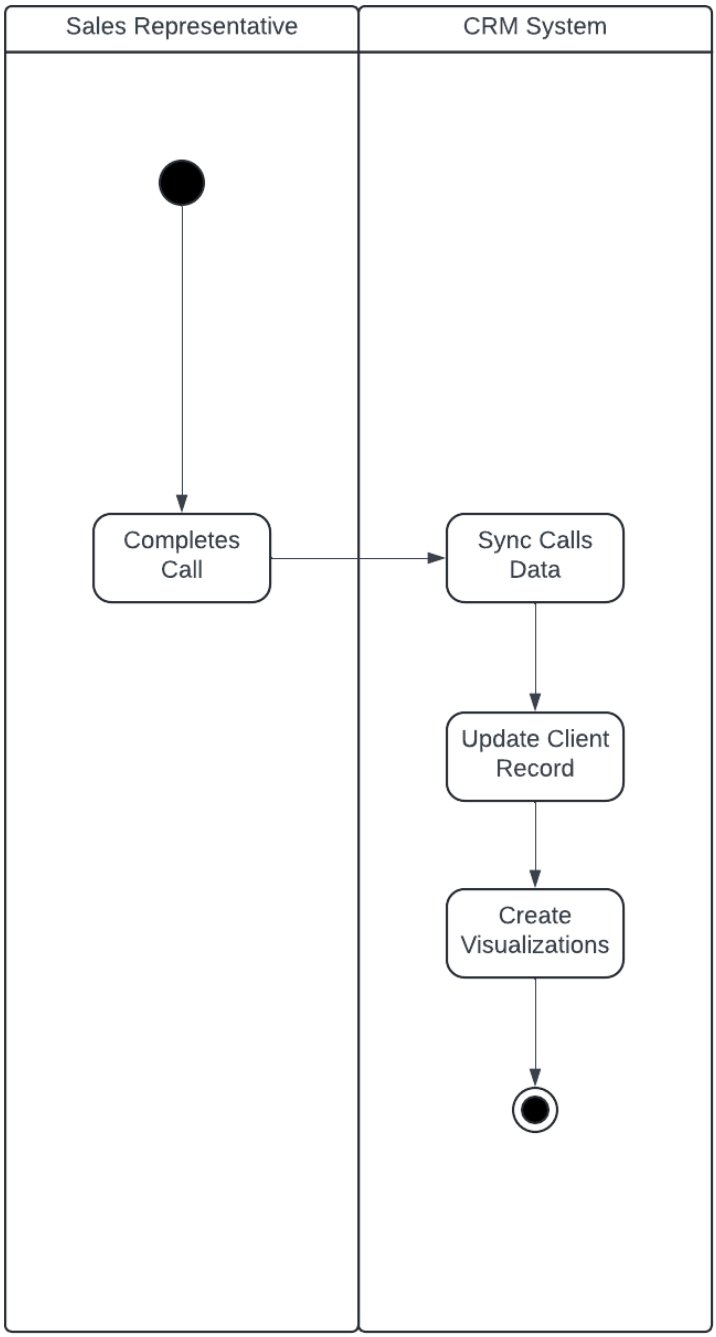

Figure 3.11: CRM Functionality

3.4.2 System Sequence Diagrams:

System sequence diagrams show user interactions with the system for each use case, focusing on external inputs and system responses. They illustrate how the system processes user inputs and highlight the core functionalities of the system.

3.4.2.1 Sign-Up:

Here the users input their details to create an account. The system validates the details and stores the user profile in the database.

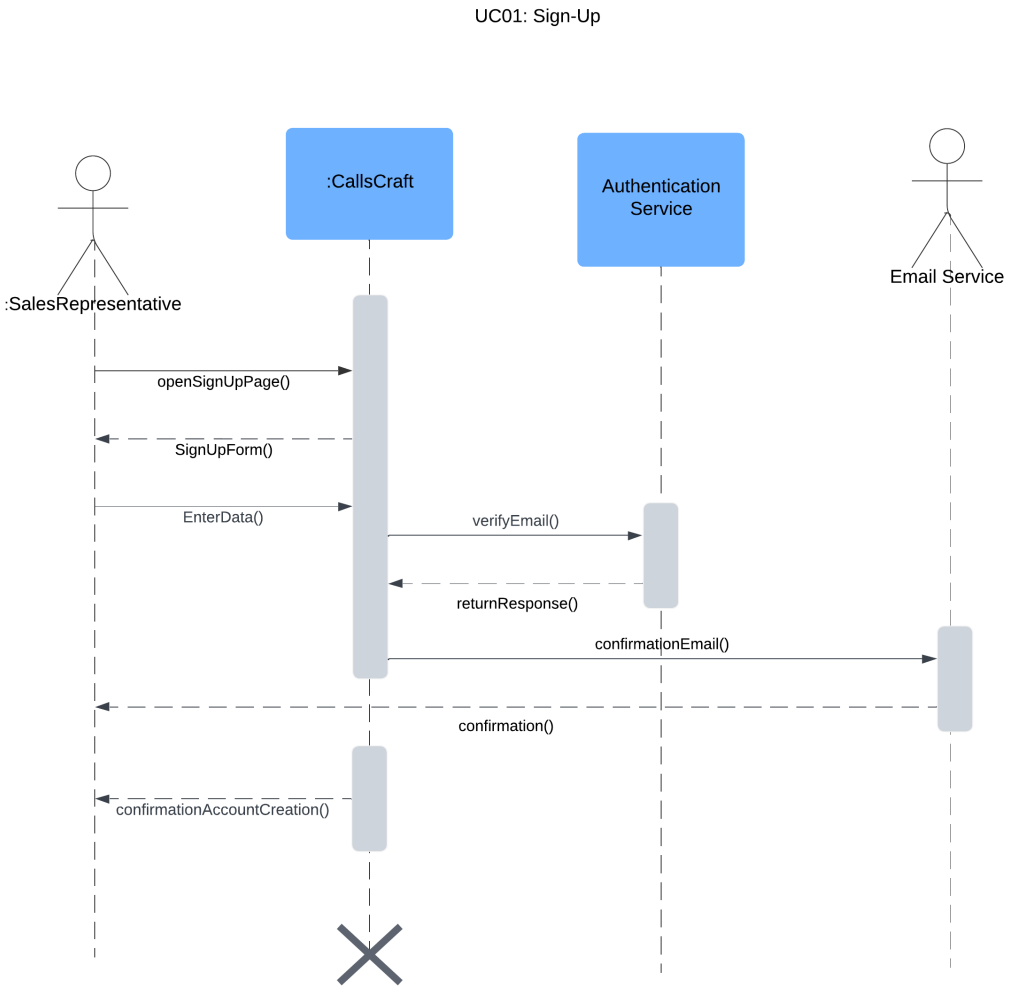

Figure 3.12: Sign-up

3.4.2.2 Login:

Here the credentials are verified. Upon success, the user gains access to their account.

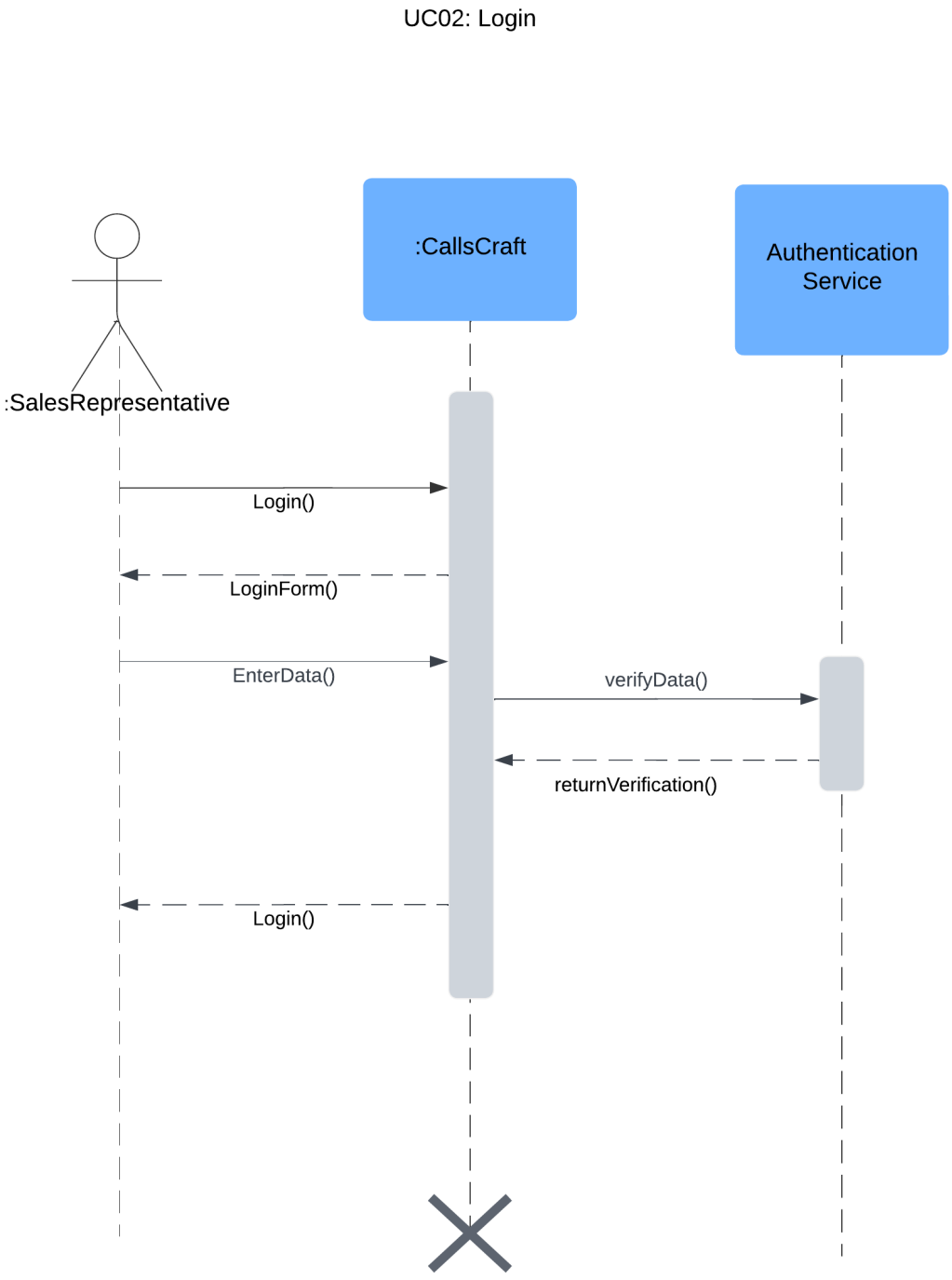

Figure 3.13: Login

3.4.2.3 Link Management:

This shows users create or join Zoom calls via CallsCraft. The system authenticates through the Zoom API and connects them to the call.

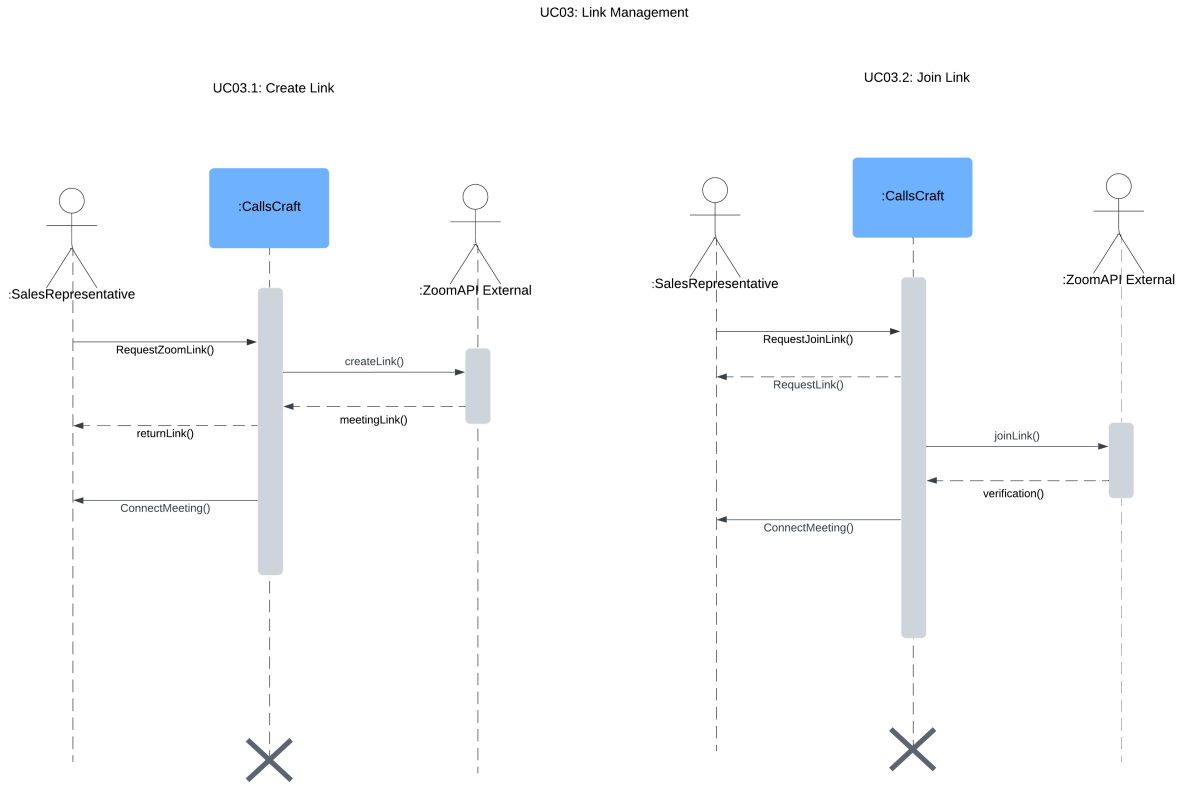

Figure 3.14: Link Management

3.4.2.4 Analytics Visualizations:

The system collects and visualizes analytics data from calls. These visualizations help users review metrics like body language and speech emotion.

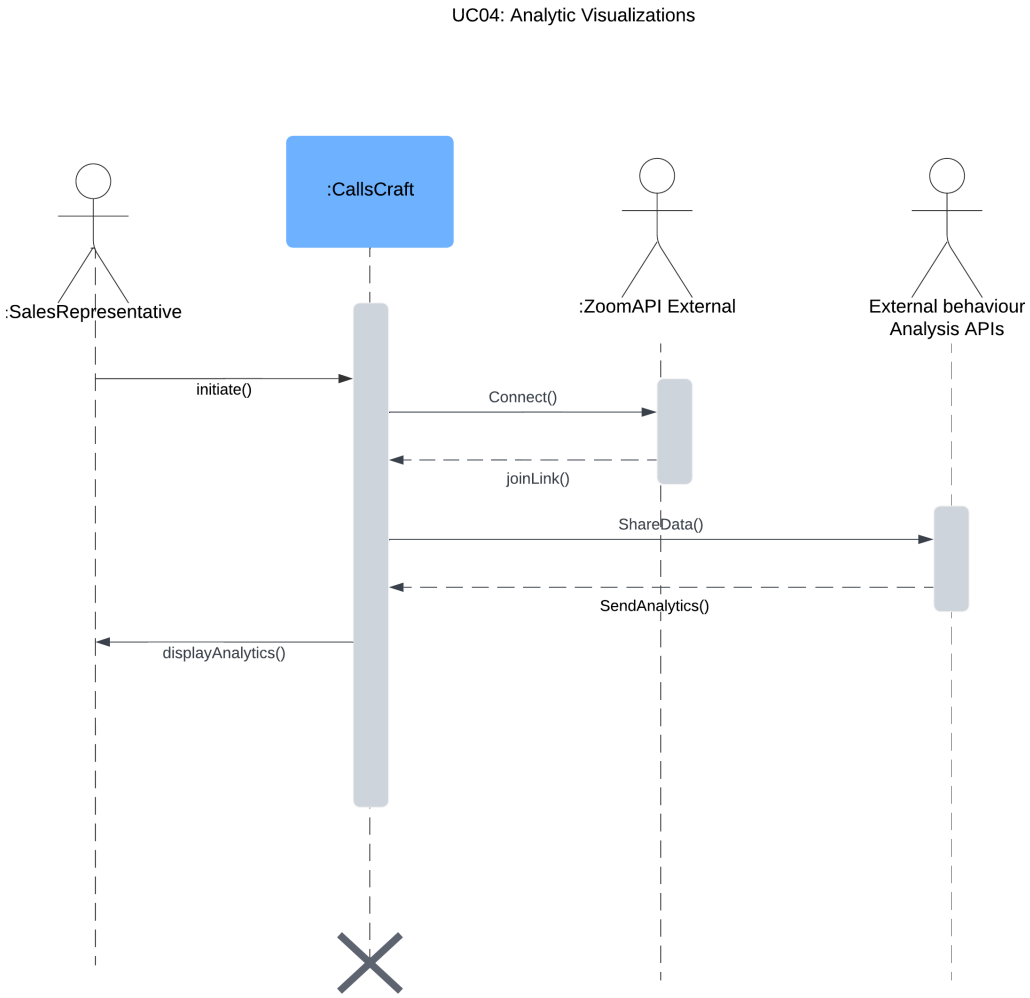

Figure 3.15: Analytics Visualizations

3.4.2.5 Display Visualizations:

This illustrates how users can view call analytics in charts and graphs.

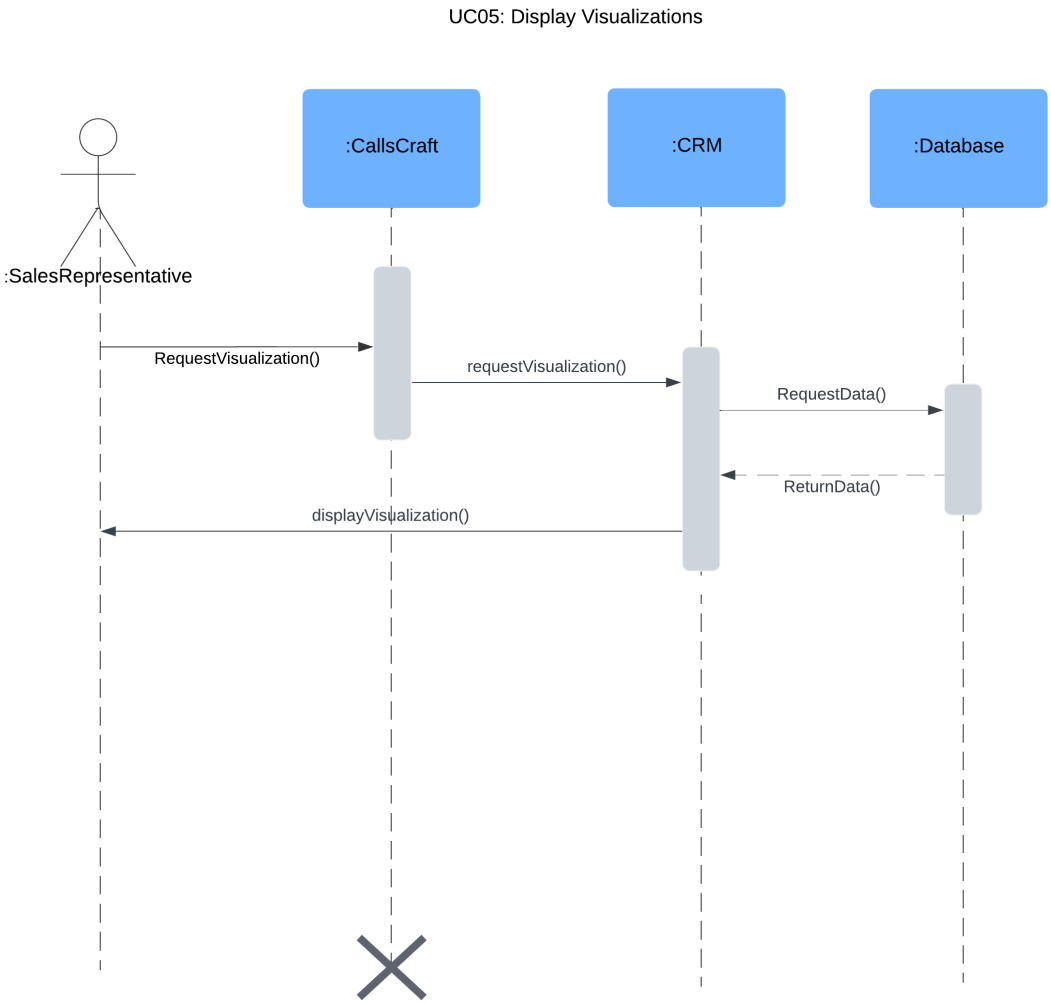

Figure 3.16: Display Visualizations

3.4.2.6 Call Transcript:

Generation of call transcripts after a meeting. The system processes the audio and provides a readable transcript for user review at the end of the call.

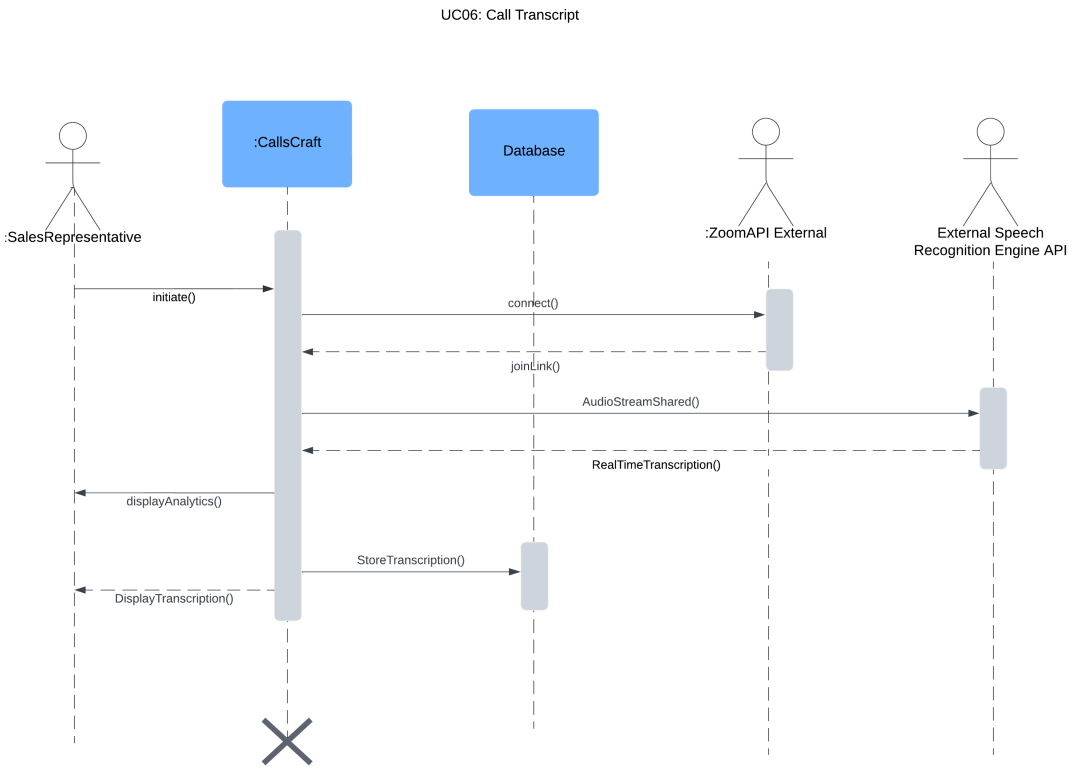

Figure 3.17: Call Transcript

3.4.2.7 Minutes of Meeting:

automatic generation of meeting minutes from the call transcript. Key points are summarized.

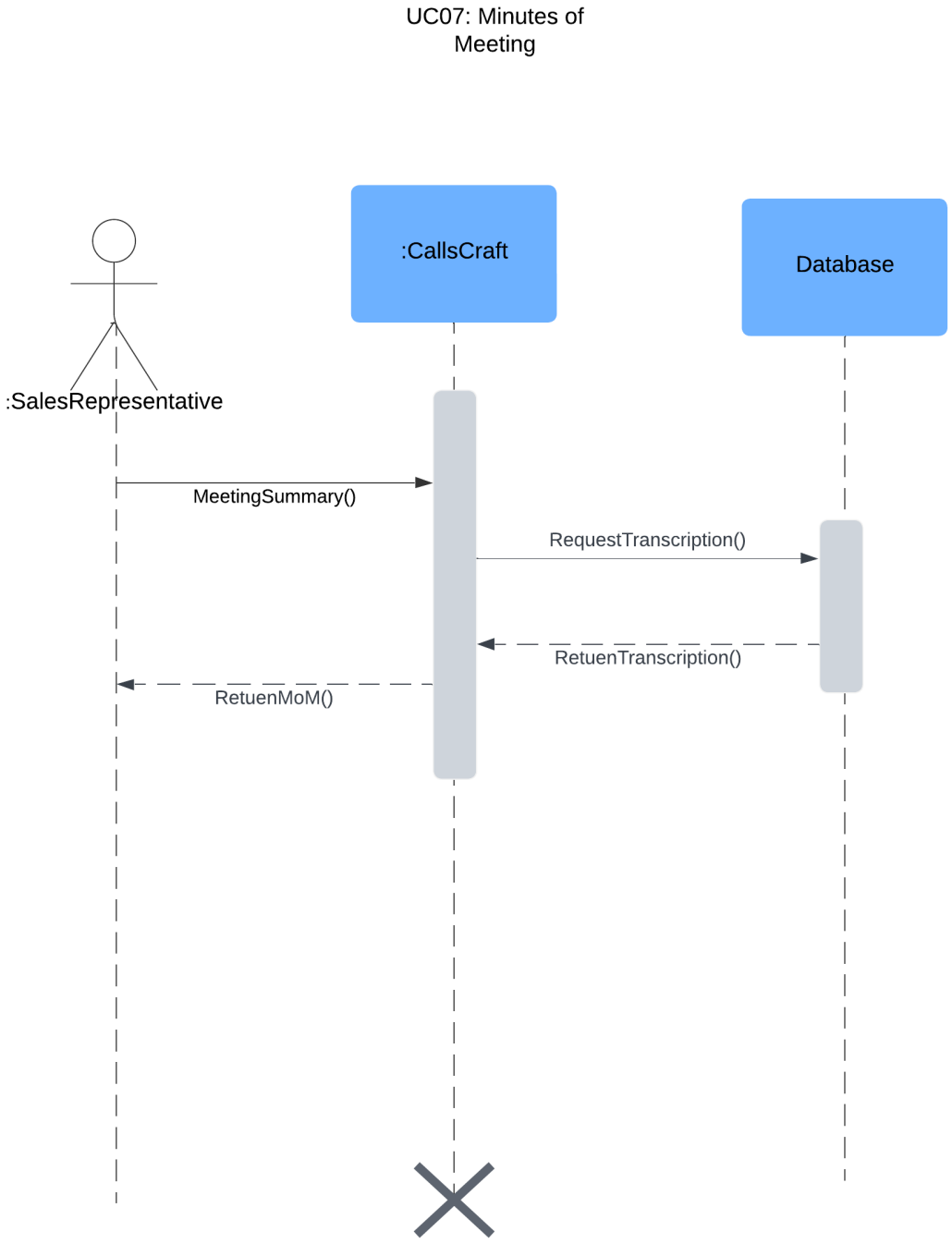

Figure 3.18: Minutes of Meeting

3.4.2.8 Notepad:

Users can take notes of important things during call.

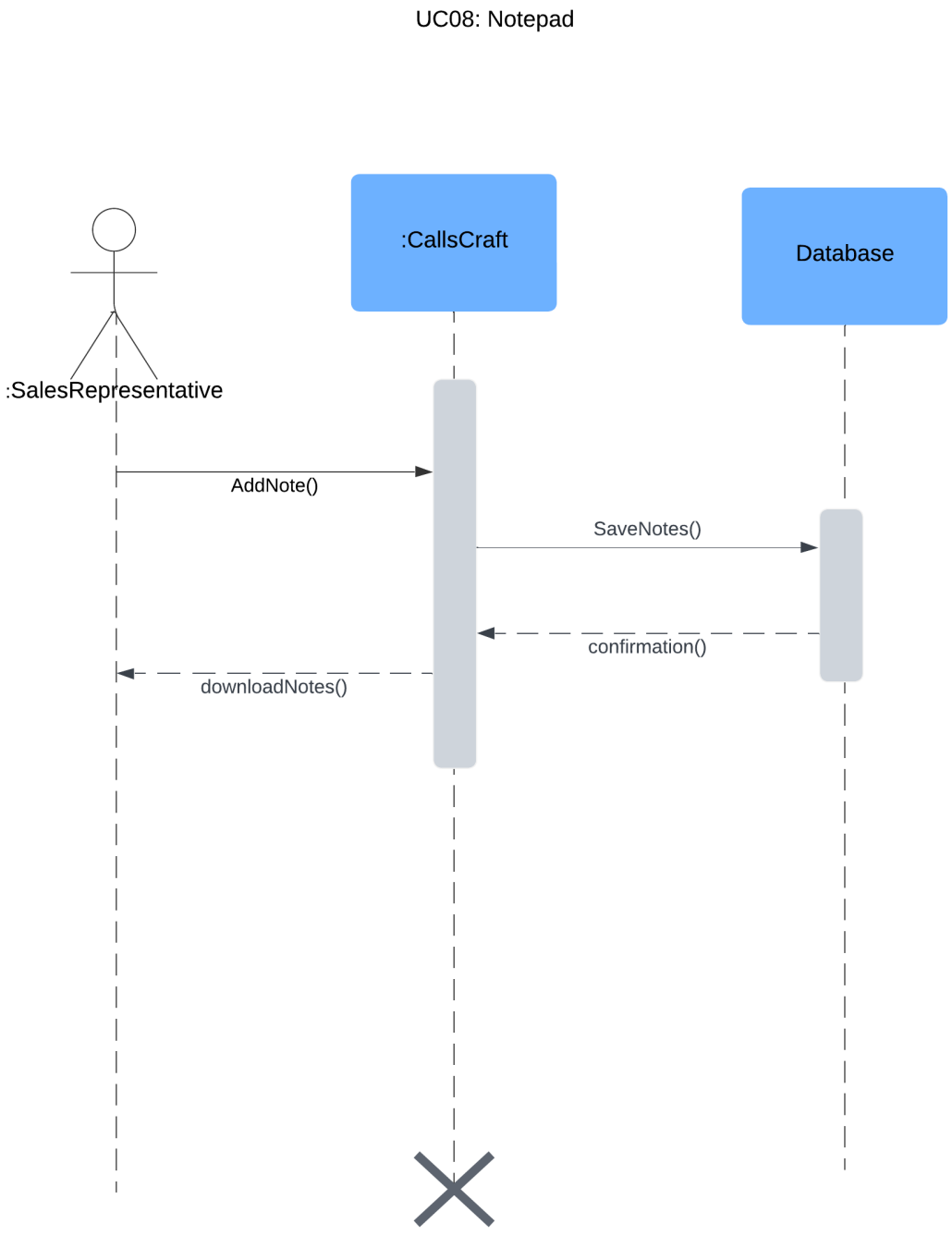

Figure 3.19: Notepad

3.4.2.9 Read Suggestions:

The system offers real-time suggestions to the representative during a call. These suggestions help maintain client's focus, improving engagement.

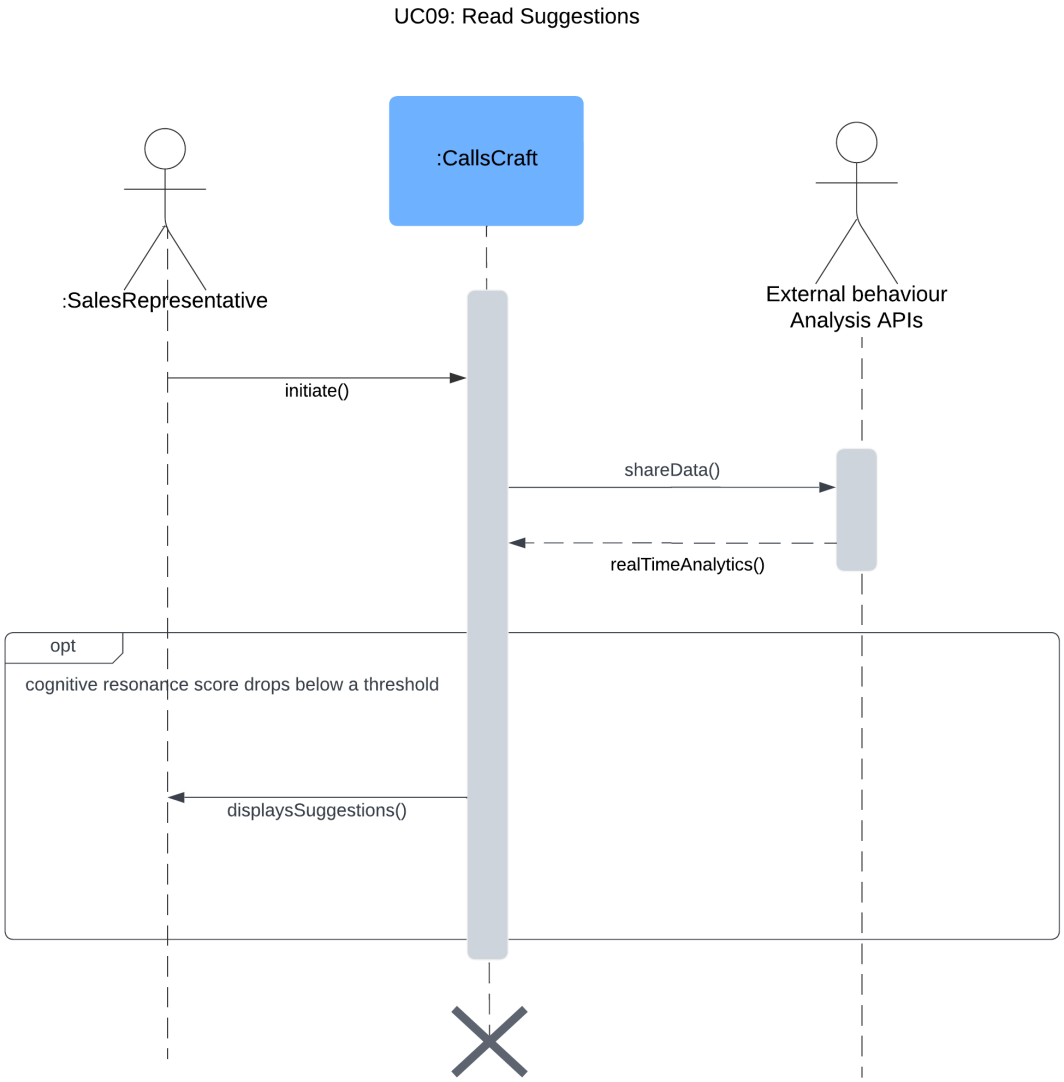

Figure 3.20: Read Suggestions

3.4.2.10 Automatic Checklist:

The system keeping record of call objectives with an automatic checklist. It marks goals as completed during the call to ensure all the tasks are completed.

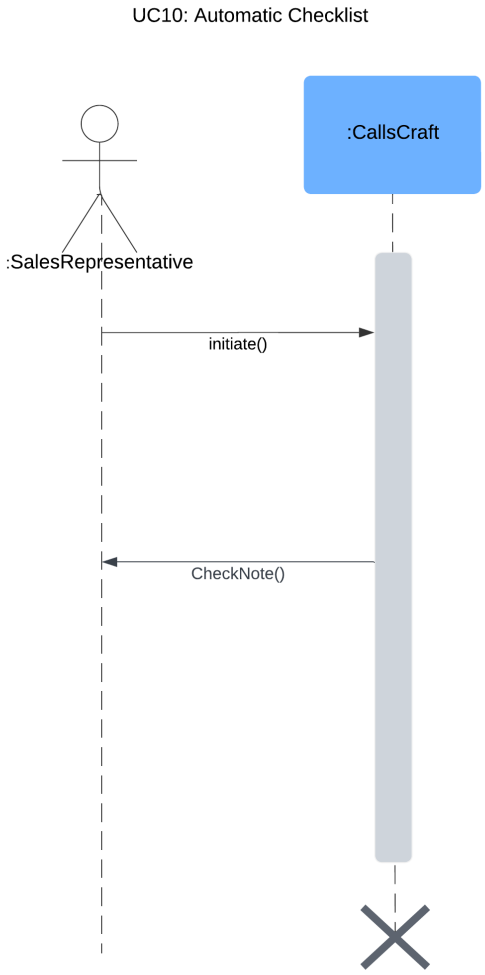

Figure 3.21: Automatic Checklist

3.4.3 Data Flow Diagram:

The Data Flow Diagram shows the overall working of the whole system. The context level diagram shows a top view of all the major parts while the Level 1 diagrams displays the inner working in a more in depth way of how the data is being processed by which components.

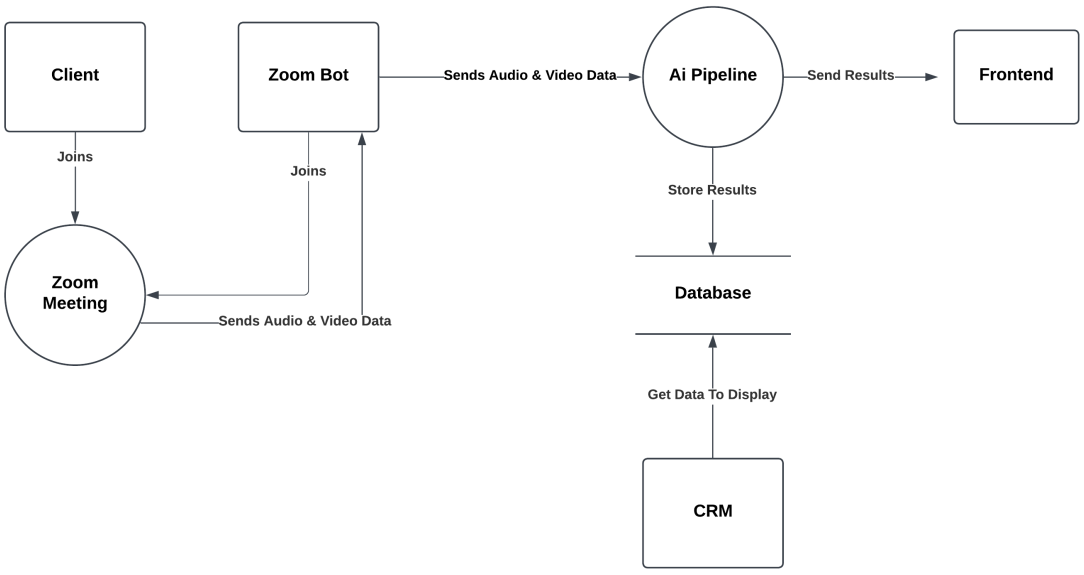

Figure 3.22: Context Level Diagram

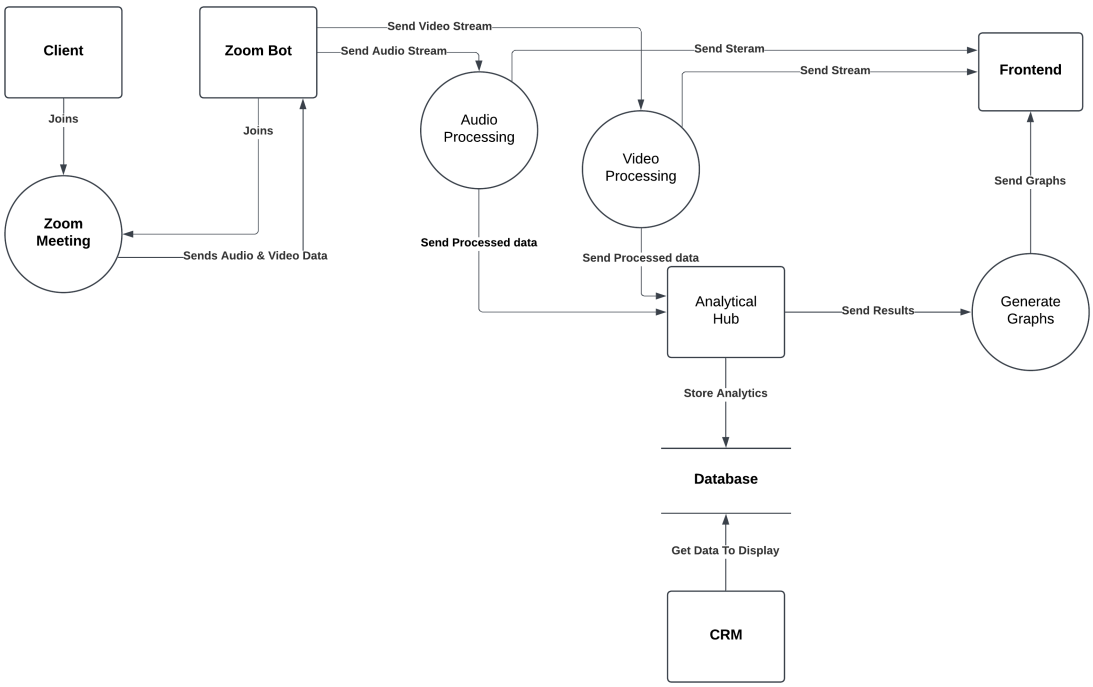

Figure 3.23: Level 1 Diagram

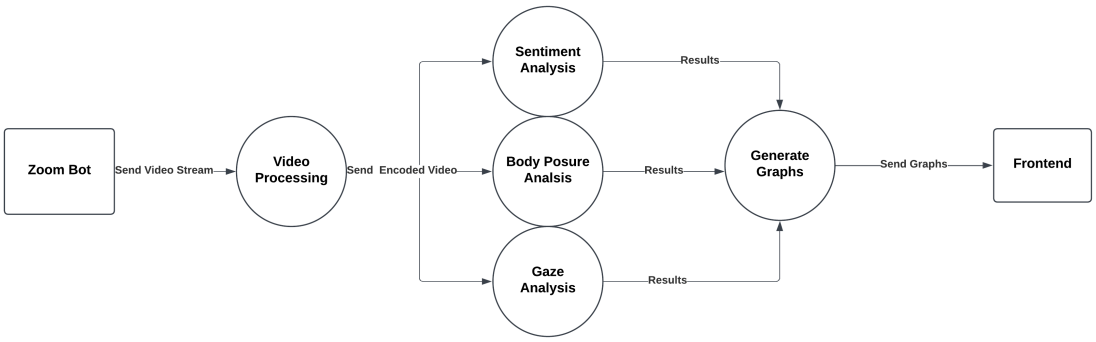

Figure 3.24: Lower Level Diagram for Video Processing

Figure 3.25: Lower Level Diagram for Audio Processing

Chapter 4

Implementation and Testing

4.1 Algorithm Design

The following are the pseudocode for the algorithms used in the project

Algorithm 1 Body Posture Alignment Detection

Require: Video Stream Input (Frames from webcam)
Ensure: Posture Classification (Aligned / Misaligned), Angle between Shoulders

```
Initialize MediaPipe Pose module
Initialize OpenCV for object capture
Set ANGLE_THRESHOLD to 5 degrees
while Video Stream is Open do
    Read frame from the video capture
    if Frame is not read successfully then
        Continue to next frame
    end if
    Flip and convert the frame to RGB
    Process the frame with MediaPipe Pose for landmark detection
    if Pose Landmarks are detected then
        Extract landmarks for Left Shoulder and Right Shoulder
        Calculate the pixel coordinates for Left and Right Shoulders
        Calculate the angle between the shoulders using trigonometry:
        Compute delta_x and delta_y between Left and Right Shoulder coordinates
        Compute angle in radians using atan2 function
        Convert the angle to degrees
          Normalize the angle to range [-180, 180) degrees
        Calculate the minimal angle difference from -180 degrees (aligned position)
        Print the angle and angle difference for debugging
        Classify the posture based on angle difference:
        if angle difference is less than ANGLE_THRESHOLD then
            Set posture as "Aligned"
            Set color as Green (0, 255, 0)
        else
            Set posture as "Misaligned"
        end if
        Display posture and angle difference on the frame
    else
        Return "No pose detected" message
    end if
end while
```

Algorithm 2 Eye Gaze Estimation and Classification

Require: Video Stream Input (Frames from webcam)
Ensure: Gaze Direction (Up, Down, Left, Right, Center) with Iris Position Ratios

while Video Stream is Open **do**
    Read a frame from the video capture
    Convert the image from BGR to RGB
    Detect facial landmarks using MediaPipe
    if Face is detected **then**
        Extract facial landmarks for the face
        Extract eye and iris landmarks based on configuration
        Calculate iris center for left and right eyes using the iris landmarks
        Compute the horizontal iris position ratio for both eyes:
            Compute the ratio between iris center and eye corners (left-right)
        Compute the vertical iris position ratio for both eyes:
            Compute the ratio between iris center and eyelids (top-bottom)
        Compute the average horizontal iris ratio for both eyes
        Compute the average vertical iris ratio for both eyes
        Return the average horizontal and vertical iris ratios for calibration
        Define gaze thresholds for horizontal and vertical directions:
            GAZE_RATIO_THRESHOLD = (0.40, 0.60)
            VERTICAL_GAZE_RATIO_THRESHOLD = (0.40, 0.60)
        Classify horizontal gaze direction based on average ratio:
        if average ratio is within gaze threshold **then**
            Set horizontal gaze direction as "Center" average ratio is less than threshold
            Set horizontal gaze direction as "Left"
        else
            Set horizontal gaze direction as "Right"
        end if
        Classify vertical gaze direction based on average vertical ratio:
        if average vertical ratio is within gaze threshold **then**
            Set vertical gaze direction as "Center" average vertical ratio is less than threshold
            Set vertical gaze direction as "Up"
        else
            Set vertical gaze direction as "Down"
        end if
        Combine vertical and horizontal gaze directions into final gaze direction
    else
        Return message "No face detected" on the frame
    end if
end while

Algorithm 3 Audio Transcription and Analysis

Require: User settings (model type, microphone settings, time limits)
Ensure: Transcription, WPM, speech pace, and emotion classification

```
Read and store model type, microphone settings, and time limits.
Set energy threshold and adjust for ambient noise.
Load model based on user's input; adjust for English-only transcription if required.
Create a transcription list and variables for WPM, phrase timing, and completion flags.
while program is running do
    if audio data in queue then
        Process and Transcribe Audio:
        if time since last phrase > timeout then
            Mark current phrase as complete.
        end if
        Convert audio for transcription.
        Use Whisper to transcribe audio.
        Calculate WPM:
        if first transcription then
            Record start time.
        end if
        Count words and calculate elapsed time to compute WPM.
        Classify Speech Pace:
        if WPM > 160 then
            Set pace to "fast".
        else if WPM < 100 then
            Set pace to "slow".
        else
            Set pace to "normal".
        end if
        Detect Emotion:
        Pass text to emotion model (e.g., EmoRoBERTa) and retrieve emotion.
        Update Transcription List:
        if phrase complete then
            Add to list.
        else
            Update latest entry.
        end if
        Display Results:
        Print transcription, emotion, speech pace, and WPM.
    else
        Wait briefly for new audio data.
    end if
end while
```

Algorithm 4 LLM Integration for Emotion-based Feedback

Require: Local LLM model (e.g., Llama), custom prompt template for sales feedback
Ensure: One-word suggestions based on detected emotion

Initialize local LLM model (`ChatOllama`) with specified version.
Define custom `PromptTemplate`:
    Input variable: `emotion`
    Template: "You are a sales representative. Based on the client's detected mood, provide a one-word suggestion for engagement."
Create an `LLMChain` using the prompt template and LLM.
Initialize a `feedback_cache` to store suggestions for detected emotions.
function GENERATEFEEDBACK(emotion)
    if `emotion` in `feedback_cache` **then**
        Return cached feedback.
    else
        Run `LLMChain` with input emotion.
        Cache the feedback for future use.
        Return feedback.
    end if
end function

Algorithm 5 Real-Time Emotion Detection using DeepFace

Require: Camera stream, frame processing interval
Ensure: Detected dominant emotion

Initialize video capture (`cv2.VideoCapture`) and set resolution to 640x480.
Configure frame processing interval (e.g., every 5th frame).
while camera stream is active **do**
    Capture frame from video feed.
    if frame capture fails **then**
        Exit loop.
    end if
    if frame count modulo processing interval $\neq 0$ **then**
        Display frame and wait for user input.
        if key pressed is `q` **then**
            Exit loop.
        end if
        Continue to next frame.
    end if
    Analyze frame using `DeepFace.analyze` with emotion detection.
    Retrieve dominant emotion from analysis results.
    Generate Sales Feedback:
    Call `GenerateFeedback` with detected emotion.
    Display emotion and feedback on the frame.
      Exception
    Log error message.
    Display updated frame in fullscreen.
end while
Release camera resources and close all display windows.

4.2 External APIs/SDKs

Describe the third-party APIs/SDKs used in the project implementation in the following table. Few examples of APIs are provided in the table.

API	Description	Purpose of usage	API endpoint/function/class used
Ollama	Locally running LLMs	To locally run LLMs for suggestive feature	query$_{p}$*rompt*
Groqcloud	Running LLMs on cloud	To run LLMs on cloud for suggestive features	client.chat.completions
Emoroberta	Emotion Detection	To detect client emotions throughout the conversation	j-hartmann/emotion-english-distilroberta-base

SDK	Description	Purpose of usage	API endpoint/function/class used
ZOOM-Meeting-SDK	A meeting SDK to make a headless bot	To access raw data including audio and video data through the headless bot	—
ZOOM-Auth-SDK	An authentication SDK to make JWT tokens	Using the JWT token we authenticate the	—
ZOOM-web-SDK	A react supported frontend SDK	A react supported SDK that lets us integrate the zoom meeting stream into the react frontend	—

4.3 Testing Details

4.3.1 Unit Testing

Table 4.1: Unit Test Case for Body Alignment

Test Case ID	Test Objective	Precondition	Steps	Test Data	Expected Results	Pass/Fail
TC001	Verify that the angle between the shoulders and the angle difference are correctly calculated when the user is aligned.	User is sitting with shoulders aligned (perfect posture).	1. Start the video capture. 2. Run the posture detection logic. 3. Check the angle between the shoulders and the angle difference between the calculated value and the expected aligned value (-180 degrees). 4. Verify the posture classification result.	User with perfectly aligned shoulders.	- Angle between shoulders: approximately -180 degrees. - Angle difference: 0.00 degrees. - Posture classification: "Aligned" (within the threshold of 5 degrees).	Pass
TC002	Verify that the angle between the shoulders and the angle difference are calculated correctly when the user has a slight misalignment.	User is sitting with a slight misalignment (shoulders not perfectly aligned).	1. Start the video capture. 2. Run the posture detection logic. 3. Check the angle between the shoulders and the angle difference between the calculated value and the expected aligned value (-180 degrees). 4. Verify the posture classification result.	User with slight misalignment (angle between shoulders between -170 and -180 degrees).	- Angle between shoulders: between -170 and -180 degrees. - Angle difference: between 0.00 and 5.00 degrees. - Posture classification: "Aligned" (within the threshold of 5 degrees).	Pass
TC003	Verify that the angle difference calculation works correctly when the user is significantly misaligned.	User is sitting with a significant misalignment (shoulders significantly off-center).	1. Start the video capture. 2. Run the posture detection logic. 3. Check the angle between the shoulders and the angle difference between the calculated value and the expected aligned value (-180 degrees). 4. Verify the posture classification result.	User with significant misalignment (angle between shoulders between -150 and -170 degrees).	- Angle between shoulders: between -150 and -170 degrees. - Angle difference: greater than 5 degrees. - Posture classification: "Misaligned".	Pass
TC004	Verify that the system correctly identifies the misalignment when the user is extremely misaligned (greater than 20 degrees).	User is sitting with extreme misalignment (shoulders significantly off-center).	1. Start the video capture. 2. Run the posture detection logic. 3. Check the angle between the shoulders and the angle difference between the calculated value and the expected aligned value (-180 degrees). 4. Verify the posture classification result.	User with extreme misalignment (angle between shoulders between -140 and -160 degrees).	- Angle between shoulders: between -140 to -179 degrees. - Angle difference: greater than 10 degrees. - Posture classification: "Misaligned".	Pass
TC005	Verify that the posture detection works when no pose is detected.	No user in the camera frame or an invalid frame (e.g., frame with no visible person).	1. Start the video capture. 2. Run the posture detection logic. 3. Check if the system detects any pose landmarks. 4. Verify the system's response when no pose is detected.	No user in the frame.	- The system should detect "No pose detected". - The system should display "No pose detected" and not attempt to calculate angles or classify posture.	Pass

Table 4.2: Unit Test Case for Gaze Tracking

Test Case ID	Test Objective	Precondition	Steps	Test Data	Expected Results	Pass/Fail
TC001	Verify that the iris center is correctly calculated.	Valid iris landmarks are provided as input.	1. Define example iris landmarks (indices of iris points on the eye). 2. Define a list of general facial landmarks. 3. Call `get_iris_center` with the defined iris landmarks and facial landmarks. 4. Verify the result by comparing the calculated iris center.	- Iris Landmarks: [36, 37, 38, 39, 40, 41] (indices of iris points). - Landmarks: { 36: [100, 200], 37: [110, 210], 38: [120, 220], 39: [130, 230], 40: [140, 240], 41: [150, 250] }	Iris center: [125.0, 225.0]	Pass
TC002	Verify that the iris position ratio is correctly computed horizontally.	Valid eye landmarks and calculated iris center.	1. Define the eye landmarks (left and right corners of the eye). 2. Define the iris center. 3. Call computeIrisPositionRatio with eye landmarks and iris center. 4. Verify the calculated ratio.	- Eye Landmarks: [[50, 180], [150, 180]] (left and right corners of the eye). - Iris Center: [100.0, 185.0]	Horizontal ratio: 0.5 (iris center is halfway between the eye corners).	Pass
TC003	Verify that the iris position ratio is correctly computed vertically.	Valid top and bottom eyelid landmarks and calculated iris center.	1. Define the top and bottom eyelid landmarks. 2. Define the iris center. 3. computeIrisPoisitonVertical with the top and bottom eyelid landmarks and iris center. 4. Verify the calculated vertical ratio.	- Eye Top Landmark: [100, 170] (top eyelid). - Eye Bottom Landmark: [100, 200] (bottom eyelid). - Iris Center: [100.0, 185.0]	Vertical ratio: 0.5 (iris center is halfway between the eyelids).	Pass
TC004	Verify that the Euclidean distance between two points is calculated correctly.	Two points are provided as input.	1. Define two points in the 2D space. 2. computeDistance with the two points. 3. Verify the calculated distance.	- Point 1: [100, 100]. - Point 2: [150, 150].	Distance: $\sqrt{(150-100)^2+(150-100)^2} = \sqrt{2500+2500} = \sqrt{5000} \approx 70.71$	Pass
TC005	Verify that minimal angle difference is computed correctly.	Two angles are provided as input.	1. Define two angles. 2. Call minimalAngleDifference with the two angles. 3. Verify the calculated minimal angle difference.	- Angle 1: 170. - Angle 2: -170.	Angle difference: 20 (minimal difference is wrapped around).	Pass

Table 4.3: Unit Test Case for Argument Parsing

Test Case ID	Test Objective	Precondition	Steps	Test Data	Expected Results	Pass/Fail
TC001	Verify valid argument parsing	Valid command line arguments for all parameters.	1. Call `parse_arguments` with valid arguments. 2. Verify the parsed arguments and their values.	Arguments: ['–model', 'medium', '–energy_threshold', '1000', '–record_timeout', '2', '–phrase_timeout', '3']	Parsed Arguments: { 'model': 'medium', 'energy_threshold': 1000, 'record_timeout': 2, 'phrase_timeout': 3, 'non_english': False }	Pass
TC002	Verify invalid model argument handling	Invalid model name provided.	1. Call `parse_arguments` with invalid model argument. 2. Verify that the system exits due to the invalid model.	Arguments: `['-model', 'invalid']`	SystemExit (exit triggered for invalid model)	Pass
TC003	Verify invalid energy threshold argument handling	Non-numeric energy threshold provided.	1. Call `parse_arguments` with invalid energy threshold. 2. Verify that the system exits due to the invalid threshold.	Arguments: `['-energy_threshold', 'abc']`	SystemExit (exit triggered for invalid threshold)	Pass

Table 4.4: Unit Test Case for Words Per Minute Calculation

Test Case ID	Test Objective	Precondition	Steps	Test Data	Expected Results	Pass/Fail
TC001	Verify correct WPM calculation for standard case	Words spoken = 150, Time in seconds = 60	1. Call `calculate_wpm` with 150 words and 60 seconds. 2. Verify the calculated WPM value.	Words spoken: 150, Time in seconds: 60	WPM: 150.0	Pass
TC002	Verify correct WPM calculation for zero words spoken	Words spoken = 0, Time in seconds = 10	1. Call `calculate_wpm` with 0 words and 10 seconds. 2. Verify the calculated WPM value.	Words spoken: 0, Time in seconds: 10	WPM: 0.0	Pass
TC003	Verify correct WPM calculation for high WPM case	Words spoken = 100, Time in seconds = 0.1	1. Call `calculate_wpm` with 100 words and 0.1 seconds. 2. Verify the calculated WPM value.	Words spoken: 100, Time in seconds: 0.1	WPM: 60000.0	Pass
TC004	Verify correct WPM calculation for normal case	Words spoken = 50, Time in seconds = 30	1. Call `calculate_wpm` with 50 words and 30 seconds. 2. Verify the calculated WPM value.	Words spoken: 50, Time in seconds: 30	WPM: 100.0	Pass
TC05	Verify correct WPM calculation for edge case with very high WPM	Words spoken = 1000, Time in seconds = 0.001	1. Call `calculate_wpm` with 1000 words and 0.001 seconds. 2. Verify the calculated WPM value is greater than 1000000.	Words spoken: 1000, Time in seconds: 0.001	WPM: greater than 1000000.0	Pass
TC006	Verify correct WPM calculation for zero time case	Words spoken = 100, Time in seconds = 0	1. Call `calculate_wpm` with 100 words and 0 seconds. 2. Verify the calculated WPM value.	Words spoken: 100, Time in seconds: 0	WPM: 0.0	Pass

Table 4.5: Unit Test Case for Speaking Pace Detection

Test Case ID	Test Objective	Precondition	Steps	Test Data	Expected Results	Pass/Fail
TC001	Verify correct classification of fast pace	WPM value above fast threshold (170)	1. Call `classify_pace` with a WPM of 170.0. 2. Verify that the speech is classified as fast.	WPM: 170.0	Pace: fast	Pass
TC002	Verify correct classification of slow pace	WPM value below slow threshold (90)	1. Call `classify_pace` with a WPM of 90.0. 2. Verify that the speech is classified as slow.	WPM: 90.0	Pace: slow	Pass
TC003	Verify correct classification of normal pace	WPM value in normal range (120)	1. Call `classify_pace` with a WPM of 120.0. 2. Verify that the speech is classified as normal.	WPM: 120.0	Pace: normal	Pass

Table 4.6: Unit Test Case for Emotion Detection

Test Case ID	Test Objective	Precondition	Steps	Test Data	Expected Results	Pass/Fail
TC001	Verify camera initialization	Mock `cv2.VideoCapture` to simulate successful initialization	1. Patch `cv2.VideoCapture` to simulate a working camera. 2. Initialize `EmotionDetection`. 3. Call `cap.isOpened()` to verify the camera status.	Mocked `cv2.VideoCapture`	Camera is initialized successfully (`isOpened()` returns `True`).	Pass
TC002	Verify emotion detection from frames	Mock `DeepFace.analyze` to return a specific emotion	1. Patch `DeepFace.analyze` to return `"happy"`. 2. Initialize `EmotionDetection`. 3. Call the feedback generator with the detected emotion.	Mocked `DeepFace.analyze: 'happy'`	Detected emotion is `"happy"`. Feedback is a non-empty string (e.g., "smile").	Pass
TC003	Verify frame capture from the camera	Mock `cv2.VideoCapture` to return frames	1. Patch `cv2.VideoCapture` to return `(True, "frame_data")`. 2. Initialize `EmotionDetection`. 3. Call `cap.read()` and verify the output.	Mocked `cv2.VideoCapture` returns `(True, "frame_data")`.	Frame capture is successful (`ret` is `True`). Captured frame is `"frame_data"`.	Pass
TC004	Verify integration between emotion detection and feedback generation	Mock `DeepFace.analyze` and `generate_feedback`	1. Patch `DeepFace.analyze` to return `'happy'`. 2. Patch `generate_feedback` to return `'smile'`. 3. Initialize `EmotionDetection` and process the frame.	Mocked `DeepFace.analyze: 'happy'` Mocked feedback: `'smile'`	Detected emotion is `"happy"`. Generated feedback is `"smile"`.	Pass
TC005	Verify frame skipping mechanism	Mock `cv2.VideoCapture` and set `frame_count` to simulate skip condition	1. Patch `cv2.VideoCapture` to simulate frame capture. 2. Set `frame_count` to 4. 3. Call `cap.read()` and verify no processing occurs.	Mocked `cv2.VideoCapture`	Frame is skipped.	Pass

Table 4.7: Unit Test Case for LLM Feedback

Test Case ID	Test Objective	Precondition	Steps	Test Data	Expected Results	Pass/Fail
TC001	Verify feedback generation for a valid emotion	Emotion "happy" is not in the feedback cache	1. Initialize `LLMFeedback`. 2. Call `generate_feedback` with the emotion `"happy"`. 3. Verify the feedback.	Emotion: `"happy"`	Feedback is a non-empty string (e.g., "smile")	Pass
TC002	Verify feedback generation for a cached emotion	Emotion "sad" exists in the feedback cache with feedback "listen"	1. Initialize `LLMFeedback`. 2. Manually set `feedback_cache["sad"] = "listen"`. 3. Call `generate_feedback` with the emotion `"sad"`. 4. Verify the cached feedback.	Emotion: `"sad"`	Feedback is `"listen"`	Pass
TC003	Verify feedback generation for an empty emotion string	No precondition	1. Initialize `LLMFeedback`. 2. Call `generate_feedback` with an empty string `""`. 3. Verify the feedback.	Emotion: `""`	Feedback is a fallback message (e.g., "neutral")	Pass
TC004	Verify feedback generation for an invalid emotion	No precondition	1. Initialize `LLMFeedback`. 2. Call `generate_feedback` with an invalid emotion `"invalid_emotion"`. 3. Verify the feedback.	Emotion: `"invalid_emotion"`	Feedback is a fallback message (e.g., "neutral")	Pass
TC005	Verify feedback generation for multiple emotions	No precondition	1. Initialize `LLMFeedback`. 2. Call `generate_feedback` for each emotion in `["happy", "sad", "angry", "excited"]`. 3. Verify the feedback for each emotion.	Emotions: `["happy", "sad", "angry", "excited"]`	Feedback is generated for all emotions as non-empty strings (e.g., "smile", "listen")	Pass

Bibliography

[1] Google AI Edge. Mediapipe: A framework for building perception pipelines. GitHub repository, 2024.

[2] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, et al. Mediapipe: A framework for perceiving and processing reality. In *Proc. of the 3rd Workshop on Computer Vision for AR/VR at IEEE CVPR*, 2019.

[3] OpenAI. Whisper tiny: A general-purpose speech recognition model. Hugging Face repository, 2024.

[4] A. Radford et al. Whisper: A general-purpose speech recognition model. OpenAI, 2022.

[5] S. Serengil. Deepface: A unified face recognition framework. GitHub repository, 2024.

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014.