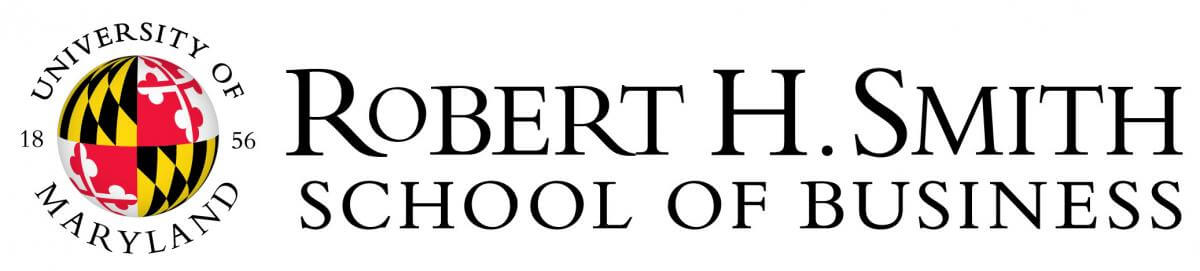
****

**Final System Design Phase Report**

**Client: Trace3**

****

**BUDT723**

**Business Process Analysis**

**Capstone Project**

Team Members of **Group 1** (under the guidance of *Dr. Paul Shapiro*):

* Farah Bushra (primary POC)
* Harshil Patel (secondary POC)
* Siddharth Khare
* Asheer Mogal
* Pranay Reddy Anthareddy
* Niharika Gupta
* Pradyum Mohta

Document last edited on: 04-28-2024

**TABLE OF CONTENTS:**

1. Executive Summary

2. Evaluation Letter from Client

3. Statement of Work

4. System Analysis Report

5. The system models of the Proposed system:

a) Data Flow Diagrams (DFDs)

b) Entity Relationship Diagrams (ERD)

c) The Synchronized System Models (The CRUD Matrix)

6. The Physical System Design:

a) The Candidate System Solutions Table

b) The Feasibility Analysis Matrix

c) The Physical Data Flow Diagrams

7. Input and Output Design

8. Implementation Plan

9. Lessons Learned

**EXECUTIVE SUMMARY**

This report contains the complete systems analysis and design proposal for the business process systems analysis for client - Trace3 (Meeting Management Tool). The project was completed by our team of information systems students from the R.H. Smith School of Business, University of Maryland, over the course of the 2024 spring semester.

Included in the following document are the detailed reports resulting from three project phases: the planning phase, the system analysis phase, and the final system design phase.

*Summary of Planning Phase*

During the planning phase, our team conducted preliminary research on the client, Trace3, and the industry in which it operates. The initial review of the business led to the identification of several opportunities for improvement, primarily revolving around the need for a centralized and automated system to manage meetings more efficiently.

These opportunities included streamlining the processes of scheduling, conducting, and analyzing meetings, facilitating communication and collaboration among participants and organizers, and providing valuable insights through data-driven analytics.

This portion of the project also contains the scope definition, which outlines our objective to research, design, and recommend a solution that would address the identified issues, while stopping short of actually building the system during this semester's course.

*Summary of Analysis Phase*

The system analysis phase involved industry-standard fact-finding techniques to derive logical system models of Trace3's current business processes regarding the following business functions:

*Meeting scheduling and coordination*

- Meeting data collection (recordings, transcripts, attendance)

- Meeting summarization and analysis

- Communication and collaboration (surveys, dashboards)

- User authentication and authorization

These business functions were modeled using data-flow diagramming techniques. The data stored in the current system (regardless of storage method) can be seen in the entity relationship diagram produced during this phase. It should be noted that the data-flow models constructed are technology-independent and only vary from the analysis phase to the proposal with respect to business process improvements.

*Summary of Final System Design Phase*

The main result of this phase is the proposed system solution, which is primarily determined as a result of the feasibility analysis matrix. Selecting a solution means choosing technology that will facilitate the logical processes modeled in our business process diagrams.

Our four candidate systems included:

1. Trace3 In-House Development (Custom Solution)

2. Commercial Off-The-Shelf (COTS) Solution

3. Outsourcing - Meeting Tracker Tool

4. Outsourcing - Meeting Management Tool

Details regarding the system proposal are found in the accompanying physical data-flow diagrams, and sample input/output screens are included in this portion of the project, as well as an implementation plan. Improvements were made to the data-flow diagrams to reflect the fact that the proposed solution integrates with various systems, such as Calendar and Meeting Applications, to automate data collection and streamline processes.

*Conclusions and Recommendations*

Based on the feasibility analysis matrix and other evaluation criteria, the recommended solution to Trace3 is Candidate 4: Outsourcing - Meeting Management Tool.

Our recommended solution meets all business requirements and is well-suited for the client in areas of evaluation such as operational feasibility, technical feasibility, economic feasibility, and schedule feasibility. Details of other candidate solutions can be found in the candidate solutions matrix. An implementation plan is also included in this report.

**EVALUATION LETTER from Trace3**The evaluation letter from the company (Trace3) will be submitted in the next deliverable. We have informed them about our request for the evaluation letter.   
  
Additionally, the Statement of work has been approved by the company as well. We have attached the related content below.

**STATEMENT OF WORK**

**CLIENT AND INDUSTRY BACKGROUND**

Trace3, founded in 2002, is a technology consulting company headquartered in Irvine, California, that provides IT strategy, solutions, and services to organizations around the globe. Trace3 has continued to work on delivering cutting-edge services to companies seeking to navigate the dynamic IT landscape and maintain a competitive edge.

With collaboration with top technology names like Cisco, Dell Technologies, and Microsoft, Trace3 has ensured it provides only the best-in-class solutions that are specifically tailored to their customers' needs. With a team of experienced consultants and engineers, Trace3 provides impactful solutions that drive growth and business value.

**PROBLEMS, OPPORTUNITIES AND DIRECTIVES**

**Challenges:**Some of the challenges that are associated are as follows:   
- Current system has a semi-optimal quality of resource allocation and meeting scheduling  
- An iterative feedback loop regarding meetings

- Keeping track of attendance of participants in meetings

- Datasets from Trace3 are required to tune our systems as per requirements. The data may be proprietary and there may be rules and regulations that need to be followed

- Clear communication of the agenda before the meeting begins

- Checking if the agenda of of the meeting was met

**Opportunities:**

- There is scope to make human resource allocation more efficient and streamlined. This will help in better time management

- Create a summary of each meeting to analyze the progress of the project.

- Make a more efficient system for Trace3 for managing their progress on different projects.

- Reminding participants about the scheduled meetings so they don't miss

- Avoiding unnecessary inclusion of employees who are involved in the high priority meets/tasks

**Directives:**

- The best design and development practices are required to be followed while creating the platform. This will also be in accordance with the compliance rules that Trace3 has set.

- The product design has to be according to the bylaws of the company

- All rules and regulations related to data-storing, record keeping and analysis have been followed according to what the company has set.

- Priorities will need to be given to a select feature of the platform that will be a union of the feasibility of the project team and requirements of Trace3.

**PROJECT SCOPE**

The project will develop an effective meeting assistance tool tailored for Trace3's strategic initiative meetings. This tool integrates meeting progress tracking, categorizing time by project, and generating insightful reports and dashboard for management review. Enhanced features will include action item reporting, meeting reminders, and integration with meet scheduling tools. Please find some of the key features:

* **Dashboard Analytics:** The tool will generate reports showing the meeting progress, meeting summary and meeting attendees; broken down by department, project, meeting, and attendee, offering comprehensive insights into resource utilization.
* **Meeting Summary Report:** As a primary requirement from Trace3, the tool will focus on accurately generating reports for each strategic meeting.
* **User Base:** The initial users of the system will include internal Strategic Initiatives (SI) team members, Project Managers (PMs), and Business Analysts (BAs), ensuring that the tool meets the specific needs of those directly involved in strategic project planning and execution.
* **Integration with Smartsheets:** The tool will be designed to integrate seamlessly with Smartsheets, allowing for the synchronization of meeting insights and action items with existing project management workflows.

**PROJECT OBJECTIVE**

Develop and implement a comprehensive meeting management tool for Trace3 by November 2024 (tentative), streamlining the entire meeting lifecycle from pre-meeting planning to post-meeting follow-up. The tool will include features such as scheduling, agenda creation (manual), document sharing, note-taking/summarizing/tracking unanswered questions(if time permits), task assignment, and analytics. Success will be measured by widespread adoption, reduced preparation time, positive user feedback on improving productivity and decision-making, and the use of analytics for data-driven insights. The tool will seamlessly integrate with Trace3's existing technology stack, including Smartsheets, Calendar and Zoom/Google meet.

**HIGH-LEVEL REQUIREMENTS**

1. Development of the core meeting analysis and utilization reporting tool/dashboard using AI.
2. The task process would be done in three sections: pre-meeting, in-meeting, and post-meeting actions.
3. The tool needs to be able to assist in meetings via features such as scheduling, task creation and assignment, resource sharing etc, as well as analyze the meeting proceedings and usage of resources.
4. Integration capabilities with existing project management software - Smartsheet.
5. Excludes complete overhaul of existing time management systems, and maintains data security and privacy

**PROJECT CONSTRAINTS**

Based on the constraints provided, here's how the project constraints may look from both the client side and the development team side:

Client-side constraints:

* **Limited Data Sharing due to Privacy**: The client may restrict the amount of data that can be shared externally due to privacy concerns. This constraint may impact the development team's access to certain data for testing or integration purposes.
* **Integration with Existing Systems**: The project requires integration with existing systems such as Smartsheets for scheduling meetings. This constraint means that the development team needs to ensure seamless integration while respecting the functionalities and limitations of the existing systems.
* **Adherence to Regulatory Standards and Compliance**: The project must adhere to regulatory standards and compliance requirements, which can impact the development process and dictate specific security and privacy measures that need to be implemented.

Team-side constraints:

* **Well-Defined Project Scope**: The development team must work within a well-defined project scope to ensure that all deliverables can be completed within the designated timeline of 6 months.
* **Budget-Conscious Resource and Technology Usage**: Resources and technologies will be thoughtfully used to keep the project within budget constraints. This means that the development team needs to prioritize cost-effective solutions and avoid unnecessary expenses.

**SYSTEM ANALYSIS REPORT**

**FACT-FINDING AND INFORMATION GATHERING PROCESS**

**Description of the fact findings and information-gathering techniques used in studying the current systems:**

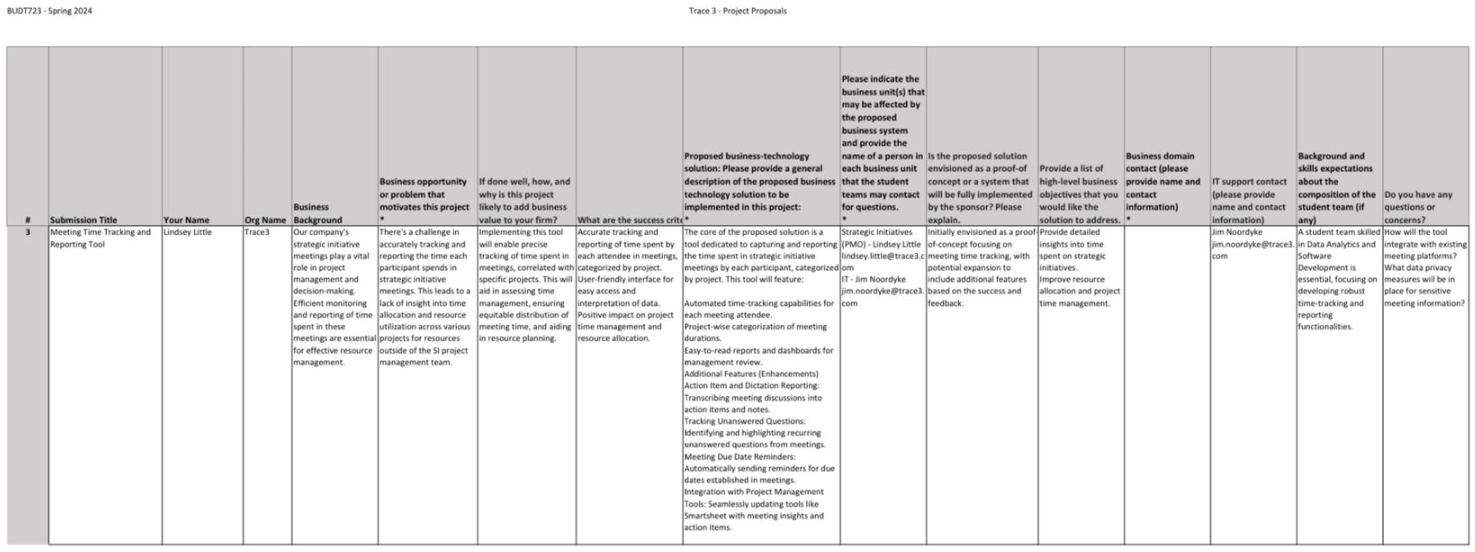
For studying the current information systems used by Trace3, we have used the following techniques - **Interviews & Observations**

1. Scheduling recurring meetings with the strategic initiative team at Trace3 (Ellen Edgington and Lindsey Little) on video conferencing. The meeting involved us asking questions about their current systems and where they would like to see improvements. This enabled us to ascertain the proper requirements of Trace3.

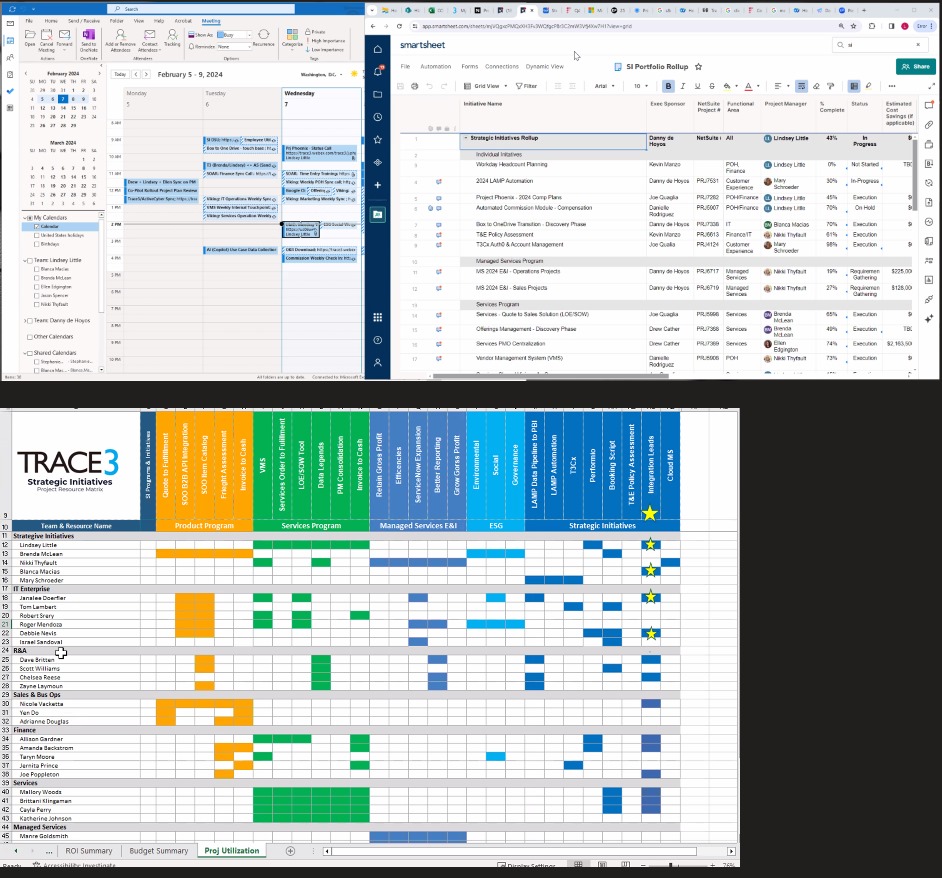
Profiles of the strategic initiative team:

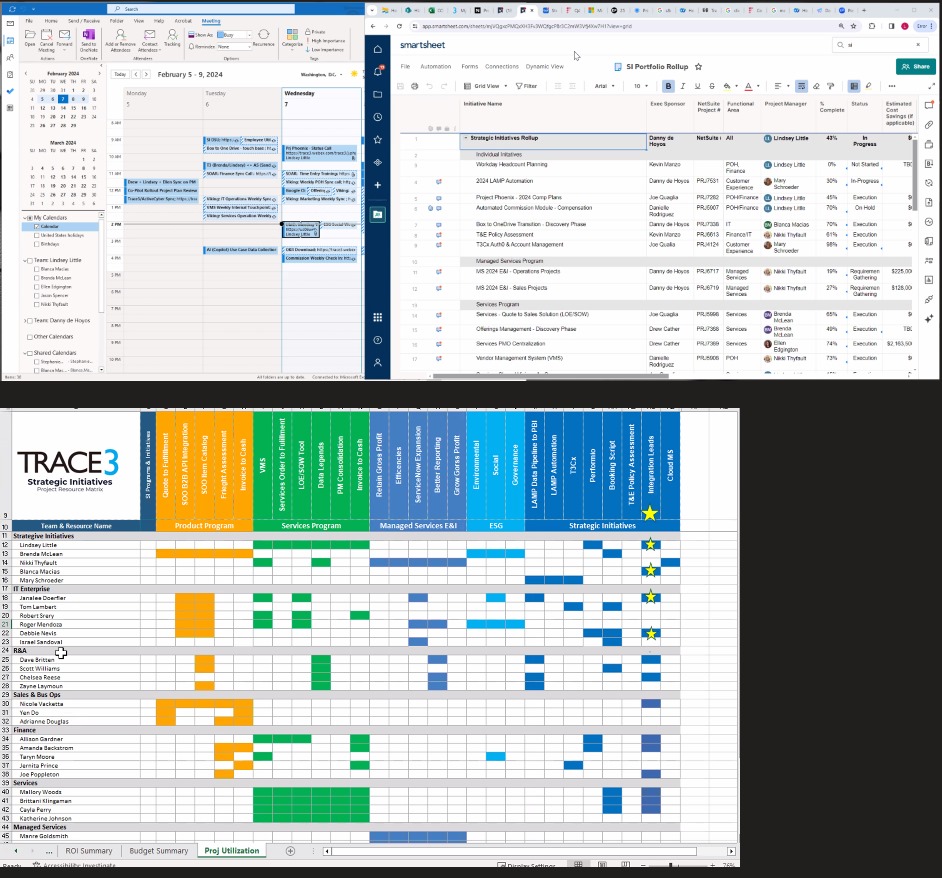
1. Lindsey Little - <https://www.linkedin.com/in/lindsey-jackson-little>
2. Ellen Eddington - <https://www.linkedin.com/in/ellen-edgington-psm-safe%C2%AE-5-certified-agilist-70a0495>
3. Getting the phase report drafts and related deliverables reviewed by Trace3. This step ensured alignment with Trace3's expectations, with iterative adjustments made based on their feedback.
4. Continuous communication established through email via our point of contact (Farah) with Trace3's strategic initiative team. This helped us keep track of any changing requirements. We have posted snippets of some of the information Trace3 has shared with us.
5. In addition to the continuous communication with Trace3, we also consulted with Prof. Shapiro to cross-check the drafts and other deliverables created by us.

Project Proposal shared by Trace3 before the start of the project:



An overview of the current system for scheduling meetings & tracking project progress:





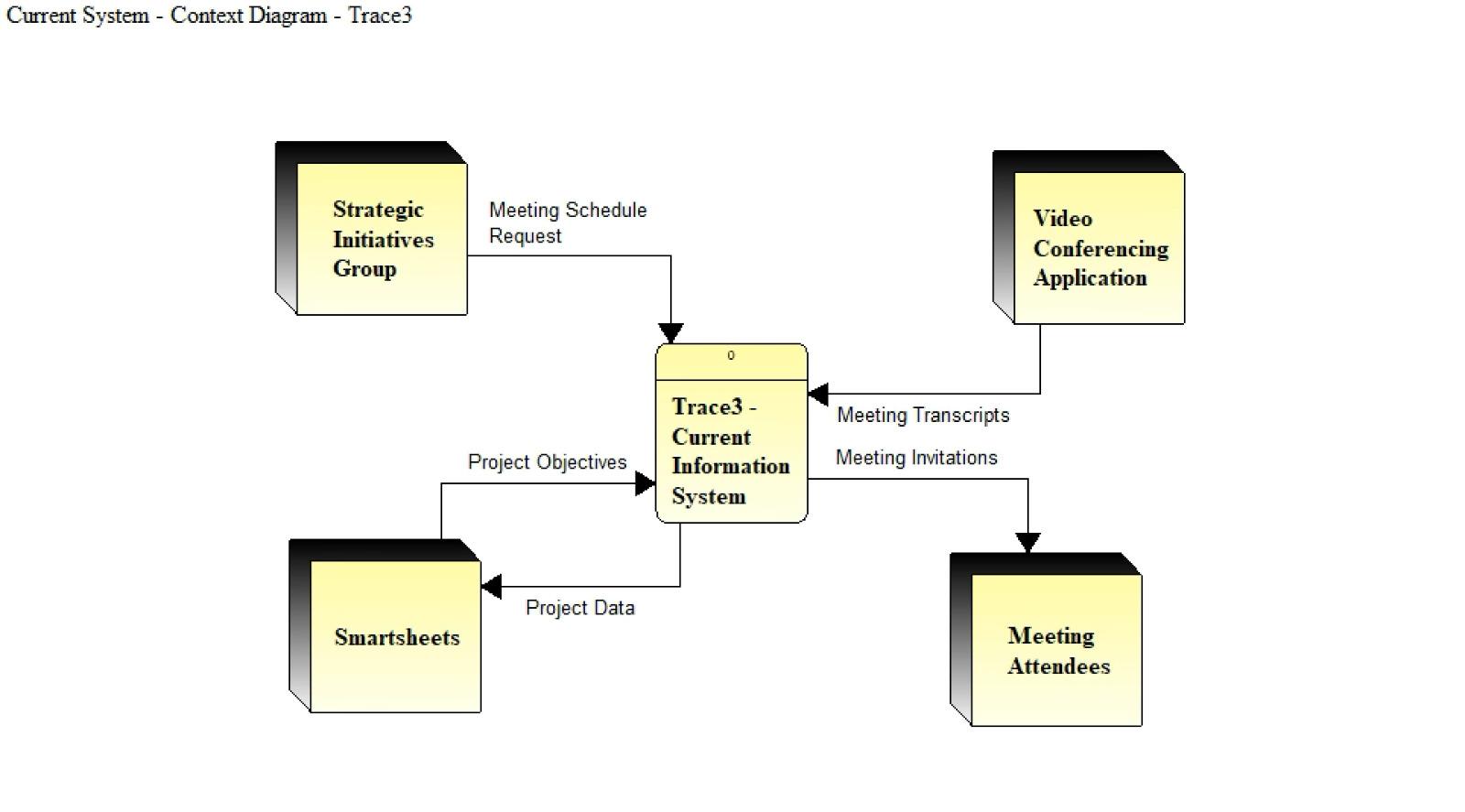
**DATA FLOW DIAGRAM**

**Visible Analyst Project Name:** GROUP1\_0501\_TRACE\_CURRENT SYSTEM

**Trace3 - Meeting Management Tool (Current)**

**Context Diagram**

*In the above Context Diagram, there are four external entities in the current meeting scheduling system for Trace3.*



**External entities are:**

* **Strategic Initiatives Group:** This external entity represents the group of project managers responsible for scheduling and organizing meetings. They initiate the meeting scheduling process by sending meeting schedule requests to the system.
* **Video Conferencing Application:** This external entity represents the video conferencing tool (such as Zoom) used to conduct virtual meetings. The system interacts with this application to exchange meeting details, conduct meetings, and receive meeting transcripts.
* **Smartsheets:** This external entity represents the project management and collaboration tool used by Trace3. The system receives project objectives from Smartsheets and sends project data back to it for seamless integration of meeting outcomes with ongoing project tracking.
* **Meeting Attendees:** This external entity represents the participants invited to the meetings. The system sends meeting invitations to the attendees, providing them with the necessary information to join the meetings.

*Trace3 Current Information System is the logical processing system for all data used in the meeting scheduling process at Trace3. Below are system inputs and outputs:*

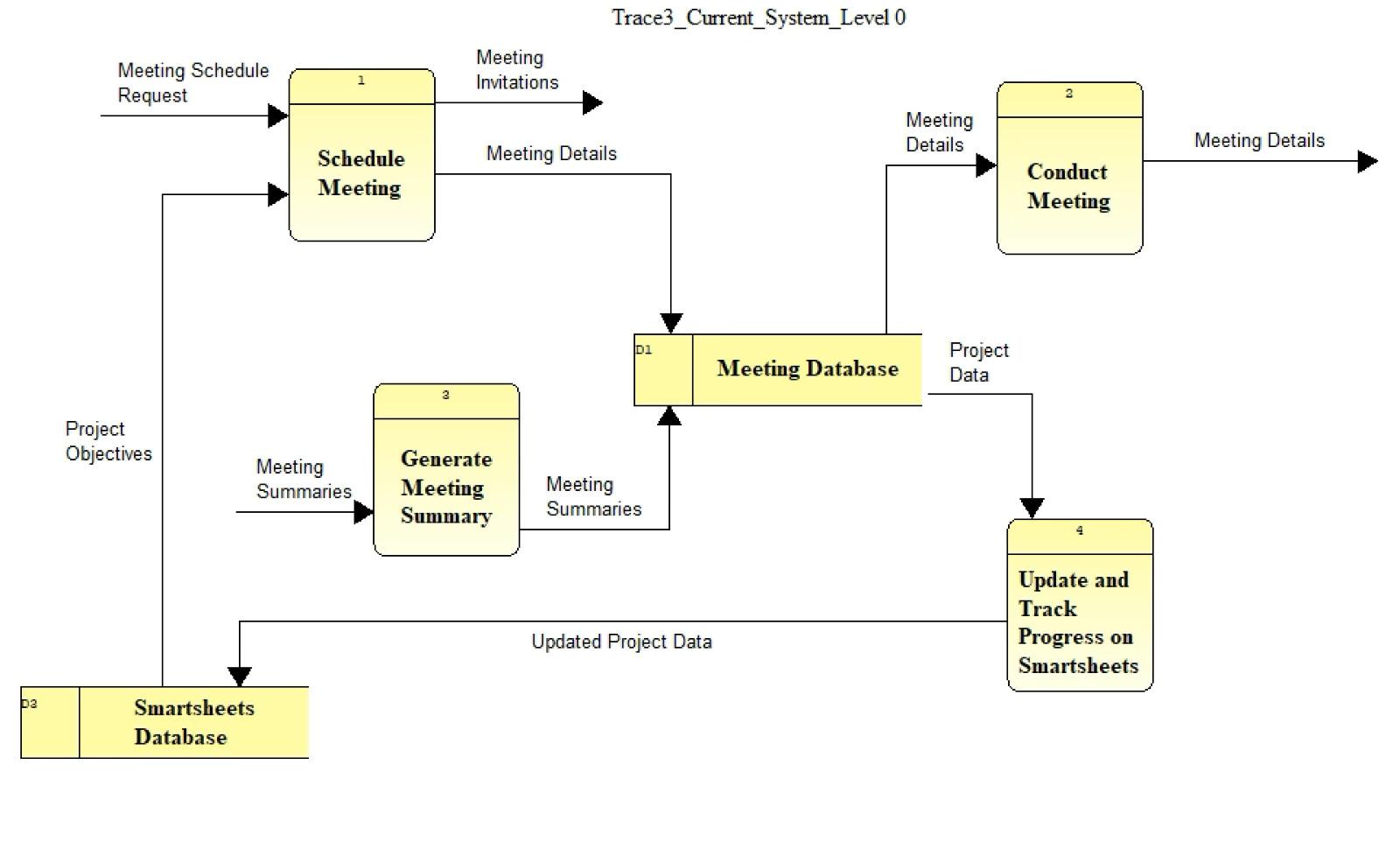
**System Inputs:**

* **From Strategic Initiatives Group:** 
  + Meeting Schedule Request (this request includes details such as the meeting agenda, date, time, and required attendees)
* **From Video Conferencing Application:**
  + Meeting Transcripts (the system receives text-based transcriptions of the meeting recordings)
* **From Smartsheets:**
  + Project Objectives (the system receives information related to projects, tasks, deadlines, and deliverables from Smartsheets)

**System Outputs:**

* **To Smartsheets:**
  + Project Data (the system sends meeting-related data, such as action items, decisions, and updates, to Smartsheets for synchronization)
* **To Meeting Attendees:**
  + Meeting Invitations (the system sends requests to attendees to join a scheduled meeting, providing them with the necessary details such as date, time, and joining instructions)

**Level 0 DFD**

*The Level 0 DFD expands on the context diagram, providing more details on the processes and data flows within the Trace3 Current Information System.*  
  


**Data Stores:**

* **Meeting Database (D1):** This data store holds all the relevant meeting information, including meeting details, attendees, and summaries. It serves as a central repository for all meeting-related data.
* **Smartsheets Database (D2):** This data store contains project-related information, such as project objectives, tasks, deadlines, and deliverables. It is used for tracking project progress and synchronizing meeting outcomes with project data.

**Processes:**

**1. Schedule Meeting:** This process manages the meeting scheduling aspect of the system. It receives meeting schedule requests from the Strategic Initiatives Group, stores meeting details in the Meeting Database, and sends out meeting invitations to attendees. The process involves the manual entry of meeting details into the system.

**Process Inputs:**

* From Strategic Initiatives Group:
  + Meeting Schedule Request
* From Smartsheet Database (data store):
  + Project Objectives (the Strategic Initiatives Group manually selects relevant project objectives when scheduling meetings)

**Process Outputs:**

* To Meeting Attendees:
  + Meeting Invitations
* To Meeting Database:
  + Meeting Details (invitations are sent via email or calendar invites)

**2. Conduct Meeting:** This process handles the actual meeting execution. It retrieves meeting details from the Meeting Database and interacts with the Video Conferencing Application to capture meeting data, such as recordings and transcripts. The system relies on the video conferencing tool's features for meeting capture.

**Process Inputs:**

* From Meeting Database (data store):
  + Meeting Details
* From Video Conferencing Application:
  + Meeting Transcripts
  + Meeting Attendance

**Process Outputs:**

* To Meeting Database (data store):
  + Meeting Data (transcripts, attendance, etc.)

**3. Generate Meeting Summary:** This process is responsible for creating meeting summaries and identifying action items and decisions. It uses the captured meeting data from the Meeting Database and the video conferencing tool's closed captioning feature to generate summaries.

**Process Inputs:**

* From Meeting Database (data store):
  + Meeting Data (transcripts, attendance, etc.)

**Process Outputs:**

* To Meeting Database (data store):
  + Meeting Summaries

**4. Update and Track Progress on Smartsheets:** This process ensures that project information remains up to date by synchronizing meeting outcomes and action items with the Smartsheets Database. It involves manual updates of project data based on the meeting summaries and decisions.

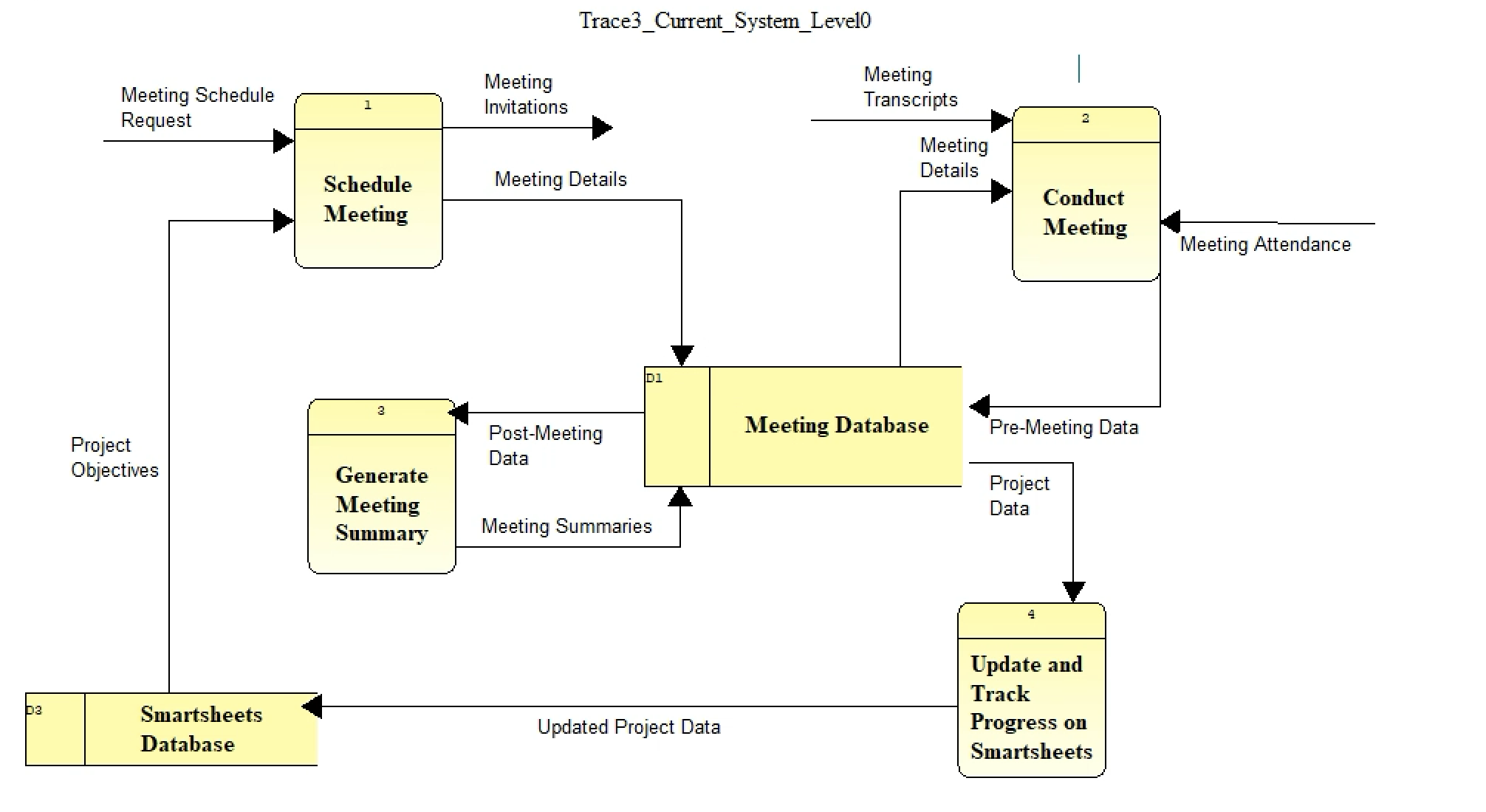
**Process Inputs:**

* From Meeting Database (data store):
  + Project Data

**Process Outputs:**

* To Smartsheets Database (data store):
  + Updated Project Data

**Level 1 DFD (Schedule Meeting Process)**

*We focus on the Schedule Meeting process from the Level Zero Diagram and expand it into three sub-processes. We also introduce a new data store, the Employee Availability Database, which is used to check attendee availability when scheduling meetings.*  
  


**Data Stores:**

* **Meeting Database (D1):** This data store holds the meeting details, attendees, and other relevant information. It serves as a central repository for all meeting-related data.
* **Employee Availability Database (D4):** This data store contains information about employee availability for meeting scheduling. It is used to check attendee availability when creating meeting requests.

**Sub-processes:**

**1.1 Create Meeting Request:** This sub-process initiates the meeting scheduling workflow by receiving meeting request details from the Strategic Initiatives Group, creating a meeting record, and sending the relevant information to the Check Attendee Availability and Send Meeting Invitations sub-processes.

**Sub-process Inputs:**

* From Strategic Initiatives Group:
  + Meeting Request Details

**Sub-process Outputs:**

* To Check Attendee Availability (sub-process):
  + Check Availability (the sub-process sends a request to check the availability of the specified attendees)
* To Send Meeting Invitations (sub-process):
  + Meeting Invitation Details (the sub-process provides the necessary information for sending out meeting invitations)
* To Meeting Database (Datastore):
  + Meeting Details (the sub-process creates a new record in the Meeting Database with the provided meeting details)

**1.2 Check Attendee Availability:** This sub-process receives the Check Availability request from the Create Meeting Request sub-process and retrieves employee availability information from the Employee Availability Database. It then sends the availability information back to the Create Meeting Request sub-process to ensure that the meeting is scheduled at a time when all required attendees are available.

**Sub-process Inputs:**

* From Create Meeting Request (sub-process)
  + Check Availability
* From Employee Availability Database (data store):
  + Employee Availability (the sub-process retrieves the availability information for the specified employees)

**1.3 Send Meeting Invitations:** This sub-process receives the Meeting Invitation Details from the Create Meeting Request sub-process and sends out meeting invitations to the selected attendees. The invitations include all the necessary information for attendees to join the meeting, such as date, time, location, and agenda.

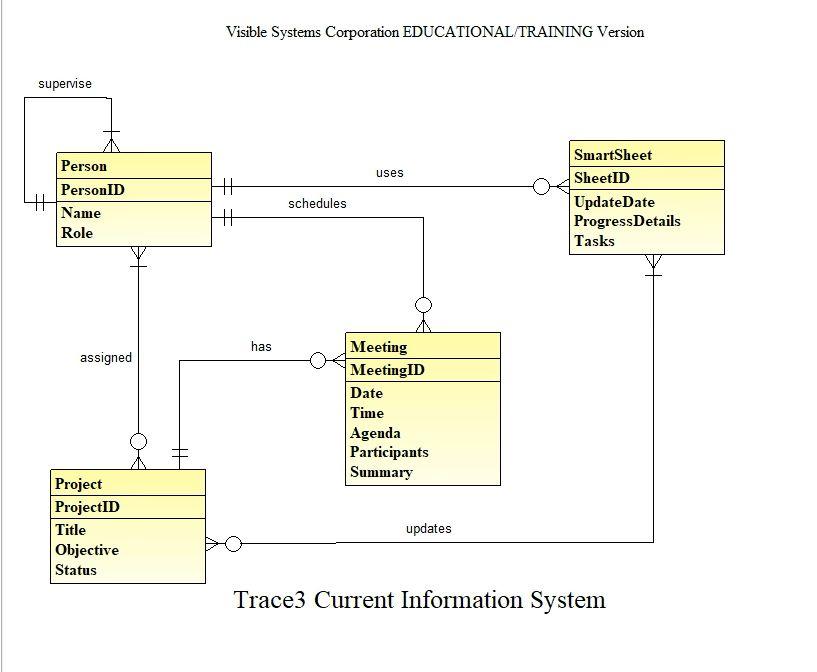
**Sub-process Inputs:**

* From Create Meeting Request (sub-process):
  + Meeting Invitation Details

**Sub-process Outputs:**

* To Meeting Attendees:
  + Meeting Invitations (the sub-process sends out the invitations to the specified attendees)

**ENTITY RELATIONSHIP DIAGRAM:**

(ERD - via Visible Analyst)  


**About the ERD:**   
  
The above ERD displays the relationships and entities in the current meeting management and scheduling system at Trace3. It is a manual solution that tries to integrate with external productivity tools like WebEx, Teams, and Smartsheets.

An organizer can request a meeting by checking Employee Availability on their scheduler (Outlook) to ensure attendee availability. Once scheduled, finalized meeting details are stored in the Meeting Database, and invitations are sent out accordingly.   
During meetings, the system utilizes Zoom’s closed captioning feature to generate captions of the whole meeting. The organizer of the meeting then manually updates the status of a project in Smartsheet.  
  
Description of the entities are:

1.⁠ **⁠Person**

- This entity represents an individual involved in the project and meeting system.

- Attributes:

- PersonID: A unique identifier for each person.

- Name: The full name of the person.

- Role: The role or position that the person holds within the organization.

- Relationships:

- Supervises: A one-to-many relationship indicating that a person may supervise none, one, or multiple other people.

- Uses: Indicates that a person uses the SmartSheet.

- Schedules: Shows that a person is responsible for scheduling meetings.

- Assigned: A one-to-many relationship, indicating a person can be assigned to none, one, or many projects.

2.⁠ **⁠Project**

- This entity captures information about specific projects within the organization.

- Attributes:

- ProjectID: A unique identifier for each project.

- Title: The official name or title of the project.

- Objective: A brief description of what the project aims to achieve.

- Status: The current state or phase of the project.

- Relationships:

- Assigned: Ties the project to one or more people.

- Has: Indicates that a project has associated meetings.

- Updates: A one-to-many relationship with SmartSheet, where project details are updated on the SmartSheet.

3.⁠ **⁠Meeting**

- Represents the meetings that are scheduled and held within the organization.

- Attributes:

- MeetingID: A unique identifier for each meeting.

- Date: The scheduled date for the meeting.

- Time: The scheduled time for the meeting.

- Agenda: A list of topics to be discussed in the meeting.

- Participants: A list of people who will attend the meeting.

- Summary: A brief overview or notes from the meeting.

- Relationships:

- Has: Linked to the Project, indicating that each project may have multiple meetings.

- Updates: Shows that meeting details are used to update the SmartSheet.

4.⁠ **⁠SmartSheet**

- An entity that likely represents a customized tool for tracking project progress, details, and tasks.

- Attributes:

- SheetID: A unique identifier for each SmartSheet.

- UpdateDate: The date when the sheet was last updated.

- ProgressDetails: Information regarding the progress of tasks and objectives.

- Tasks: A list of tasks that are part of the project or meeting.

- Relationships:

- Uses: Links back to the Person, indicating who updates the SmartSheet.

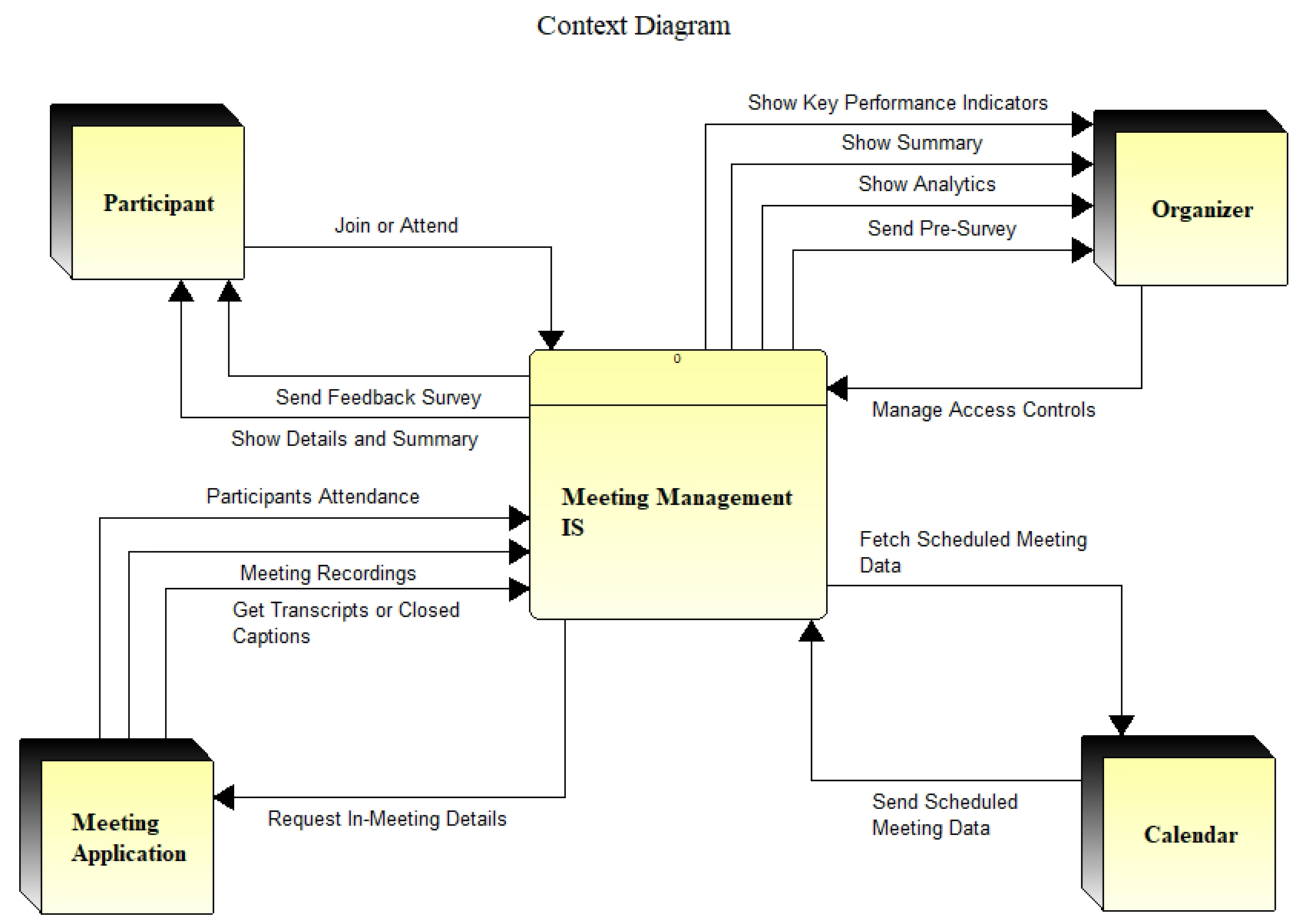
- Updates: Connected to both the Project and Meeting, showing that details from these entities are used to update the SmartSheet.

**THE SYSTEM MODEL(S) OF THE PROPOSED SYSTEM**

**DATA FLOW DIAGRAMS (DFD)**

**Context Diagram:**

The Context Diagram for the proposed Meeting Management Information System consists of four external entities: Participant, Organizer, Calendar, and Meeting Application. The Participant and Organizer interact with the central Meeting Management IS process by joining or attending meetings, managing access controls, showing analytics, sending pre-surveys, and fetching scheduled meeting data. The Meeting Management IS process communicates with the Calendar to fetch scheduled meeting data and send scheduled meeting data. It also interacts with the Meeting Application to request in-meeting details, collect meeting recordings, closed captions/transcripts, and participant attendance.

****

External entities are:

* **Participant:** This external entity represents any individual who is joining or attending a meeting managed by the Meeting Management IS. Participants can provide feedback through surveys after the meeting.
* **Organizer:** This external entity represents the individual or group responsible for organizing and managing the meetings within the Meeting Management IS. The organizer has control over access management, analytics, and pre-meeting surveys.
* **Meeting Application:** This external entity represents the software or platform used to conduct the virtual meetings, such as Zoom, Microsoft Teams, or Google Meet. The Meeting Management IS interacts with the Meeting Application to request in-meeting details and manage recordings.\
* **Calendar:** This external entity represents the calendaring system that stores and provides scheduled meeting data to the Meeting Management IS. This could be a standalone calendar application or an integrated calendar within the organization's productivity suite.

**Meeting Management IS** is the central logical processing system for all data used in the operations of managing meetings. Below are system inputs and outputs:

**System Inputs:**

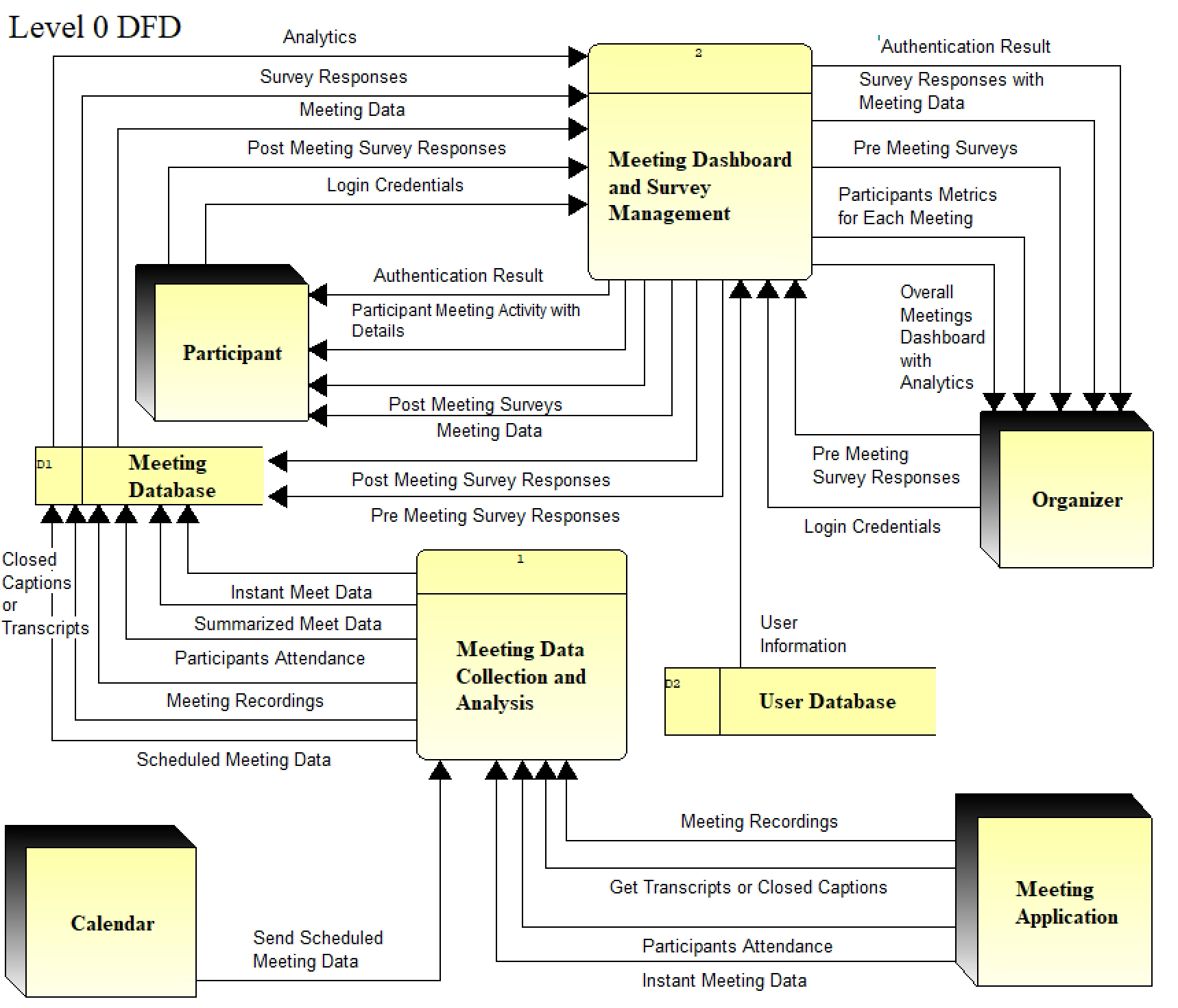
* From Participant:
  + Join or Attend Meeting
* From Organizer:
  + Manage Access Controls
* From Meeting Application:
  + Participants Attendance
  + Meeting Recordings
  + Get Transcripts or Closed Captions
* From Calendar:
  + Send Scheduled Meeting Data

**System Outputs:**

* To Participant:
  + Send Feedback Survey
  + Show Details and Summary
* To Organizer:
  + Show Key Performance Indicators
  + Show Summary
  + Show Analytics
  + Send Pre-Survey
* To Meeting Application:
  + Request In-Meeting Details
* To Calendar:
  + Fetch Scheduled Meeting Data

**Level 0 Data Flow Diagram:**

The Level 0 DFD is an explosion of the Meeting Management IS process from the Context Diagram. It reveals two main processes: "Meeting Data Collection and Analysis" and "Meeting Dashboard and Survey Management."



**Process 1: Meeting Data Collection and Analysis**

This process is responsible for collecting and analyzing meeting data from various sources, such as the Outlook Calendar, Meeting Application, and instant meetings. It also summarizes meetings and stores the analyzed data in the Meeting Database.

**Process Inputs:**

* From Outlook Calendar
  + Scheduled Meeting Data
* From Meeting Application
  + Meeting Recordings
  + Closed Captions/Transcripts
  + Participant Attendance
  + Instant Meeting Data

**Process Outputs:**

* To Meeting Database
  + Scheduled Meeting Data
  + Meeting Recordings
  + Closed Captions/Transcripts
  + Participant Attendance
  + Instant Meeting Data
  + Summarized Meeting Data

**Process 2: Meeting Dashboard and Survey Management**

This process is responsible for presenting meeting data to Admins/Organizers and Participants through dashboards and managing the survey mechanism. It retrieves data from the Meeting Database and sends surveys to gather feedback and information.

**Data Stores:**

* **Meeting Database:** This data store is responsible for storing all the collected, analyzed, and summarized meeting data, as well as survey responses.
* **User Database:** This data store is responsible for storing user information, such as email addresses and other relevant details.

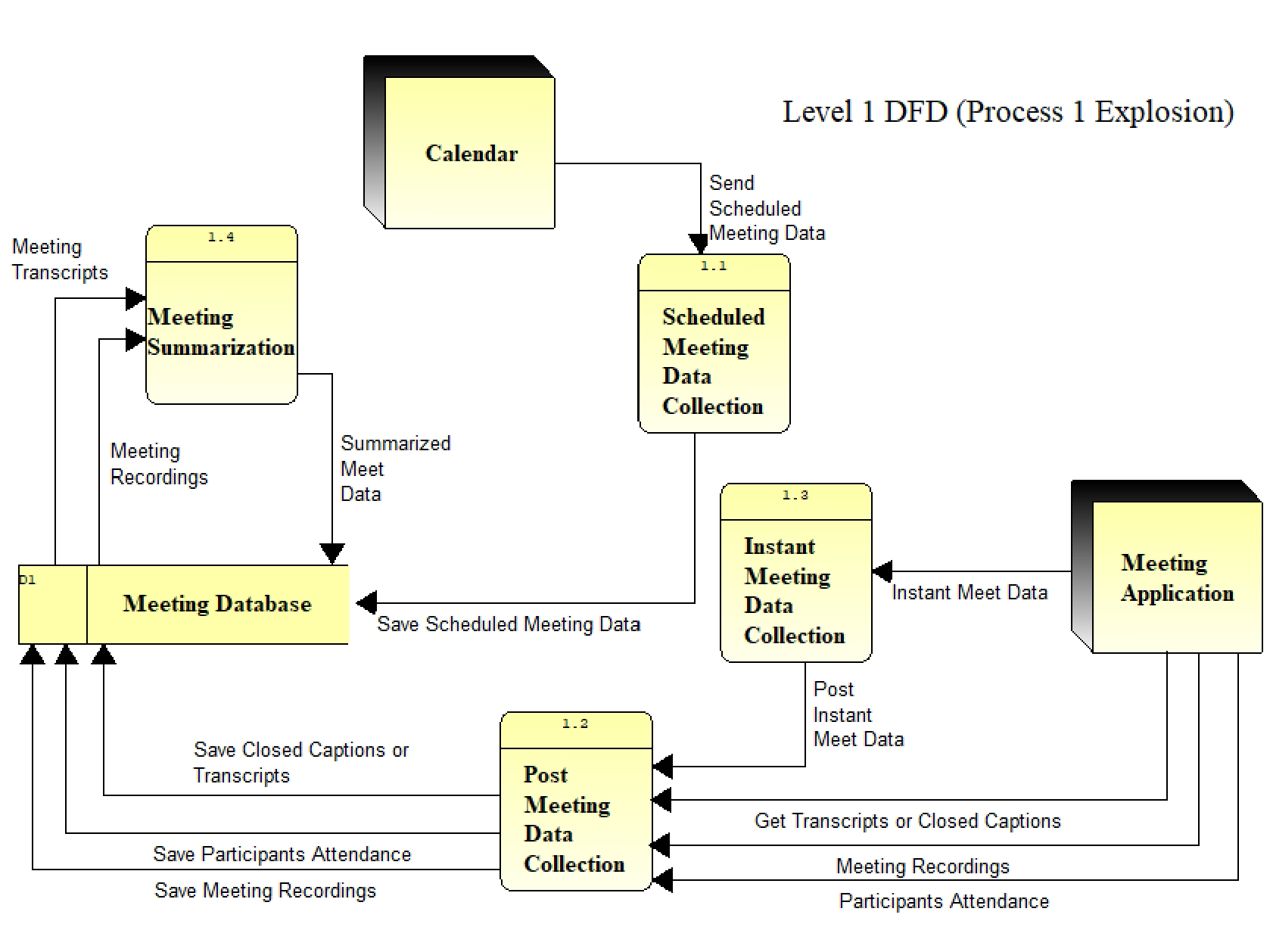
**Process Inputs:**

* From Meeting Database
  + Meeting Data
  + Analytics
  + Survey Responses
* From Admin/Organizer
  + Pre-Meeting Survey Responses
  + Login Credentials
* From Participant
  + Post-Meeting Survey Responses
  + Login Credentials
* From User Database
  + User Information

**Process Outputs:**

* To Admin/Organizer
  + Overall Meetings Dashboard with Analytics
  + Participant Metrics for Each Meeting
  + Pre-Meeting Surveys
  + Survey Responses with Meeting Data
  + Authentication Result
* To Participant
  + Participant's Meeting Activity and Details
  + Post-Meeting Surveys
  + Survey Responses with Meeting Data
  + Authentication Result
* To Meeting Database
  + Pre-Meeting Survey Responses
  + Post-Meeting Survey Responses

**Level 1 Process 1 (Meeting Data Collection and Analysis) Explosion DFD:**

****

This diagram explores the Meeting Data Collection and Analysis Process, which is responsible for collecting and analyzing meeting data from various sources. This process can be divided into four sub-processes:

**Process 1.1 Scheduled Meeting Data Collection:**

This process is responsible for fetching scheduled meeting data from the Outlook Calendar early in the morning using a scheduled task/service and saving it in the Meeting Database.

**Process Inputs:**

* From Outlook Calendar
  + Scheduled Meeting Data

**Process Outputs:**

* To Meeting Database
  + Scheduled Meeting Data

**Process 1.2 Post-Meeting Data Collection:**

This process is responsible for collecting meeting recordings, closed captions/transcripts, and participant attendance from the Meeting Application after the meeting concludes and storing the data in the Meeting Database.

**Process Inputs:**

* From Meeting Application
  + Meeting Recordings
  + Get Closed Captions or Transcripts
  + Participant Attendance
* From Instant Meeting Data Collection
  + Post Instant Meet Data

**Process Outputs:**

* To Meeting Database
  + Save Meeting Recordings
  + Save Closed Captions/Transcripts
  + Save Participant Attendance

**Process 1.3 Instant Meeting Data Collection:**

This process is responsible for verifying and collecting data for instant meetings that are not found on the calendar using the Meeting Application API and storing the data in the Meeting Database.

**Process Inputs:**

* From Meeting Application
  + Instant Meeting Data

**Process Outputs:**

* To Post Meeting Data Collection
  + Post Instant Meet Data

**Process 1.4 Meeting Summarization:**

This process is responsible for retrieving meeting data (transcripts or recordings) from the Meeting Database, summarizing meetings at the end of each day, and storing the summarized data back in the Meeting Database.

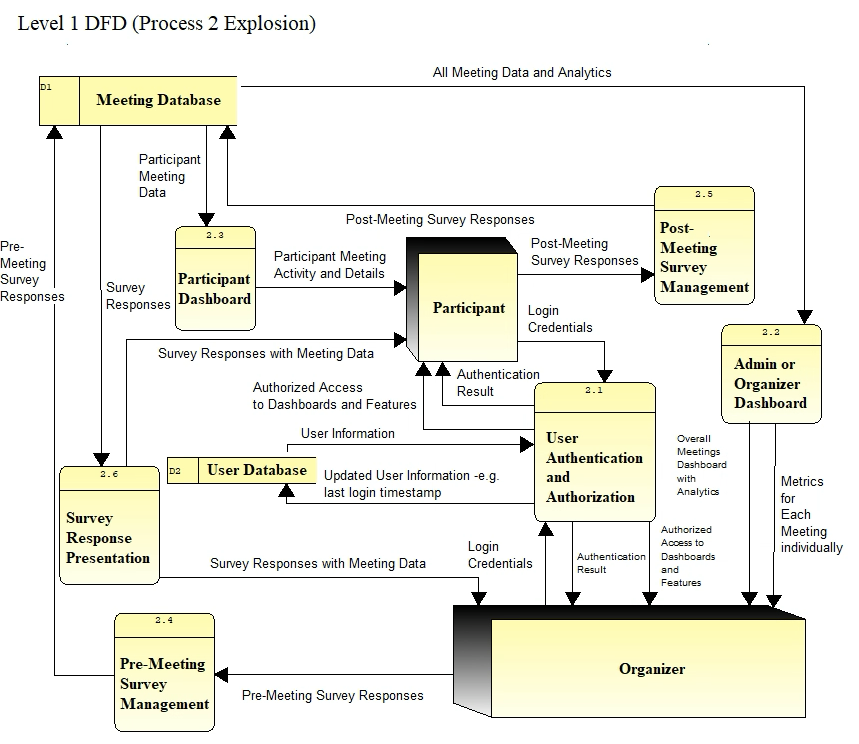
**Process Inputs:**

* From Meeting Database
  + Meeting Transcripts
  + Meeting Recordings

**Process Outputs:**

* To Meeting Database
  + Summarized Meeting Data

**Level 1 Process 2 (Meeting Dashboard and Survey Management) Explosion DFD:**

****

This diagram explores the Meeting Dashboard and Survey Management Process, which is responsible for presenting meeting data to Admins/Organizers and Participants through dashboards and managing the survey mechanism. This process can be divided into six sub-processes:

**Process 2.1 User Authentication and Authorization:**

This process is responsible for handling user login and logout functionality, authenticating users' credentials against the User Database, and authorizing user access to the appropriate dashboards and features based on their role.

**Process Inputs:**

* From Organizer
  + Login Credentials
* From Participant
  + Login Credentials
* From User Database
  + User Information

**Process Outputs:**

* To Organizer
  + Authentication Result
  + Authorized Access to Dashboards and Features
* To Participant
  + Authentication Result
  + Authorized Access to Dashboards and Features
* To User Database
  + Updated User Information (e.g., last login timestamp)

**Process 2.2 Admin/Organizer Dashboard:**

This process is responsible for retrieving all meeting data and analytics from the Meeting Database, showing the overall meetings dashboard with analytics to the Admins/Organizers, and presenting participant metrics for each meeting.

**Process Inputs:**

* From Meeting Database
  + All Meeting Data and Analytics

**Process Outputs:**

* To Organizer
  + Overall Meetings Dashboard with Analytics
  + Metrics for Each Meeting individually

**Process 2.3 Participant Dashboard:**

This process is responsible for retrieving the participant's meeting data from the Meeting Database and showing the participant's meeting activity and details, including invitees, attendance, agenda, summary, time, and platform.

**Process Inputs:**

* From Meeting Database
  + Participant's Meeting Data

**Process Outputs:**

* To Participant
  + Participant's Meeting Activity and Details

**Process 2.4 Pre-Meeting Survey Management:**

This process is responsible for sending pre-meeting surveys to Admins/Organizers to gather agenda and supporting documentation, receiving pre-meeting survey responses, and storing them in the Meeting Database.

**Process Inputs:**

* From Organizer
  + Pre-Meeting Survey Responses

**Process Outputs:**

* To Meeting Database
  + Pre-Meeting Survey Responses

**Process 2.5 Post-Meeting Survey Management:**

This process is responsible for sending post-meeting surveys to Participants to collect feedback and questions, receiving post-meeting survey responses, and storing them in the Meeting Database.

**Process Inputs:**

* From Participant
  + Post-Meeting Survey Responses

**Process Outputs:**

* To Meeting Database
  + Post-Meeting Survey Responses

**Process 2.6 Survey Response Presentation:**

This process is responsible for retrieving survey responses from the Meeting Database and presenting them to Admins/Organizers and Participants along with other meeting data.

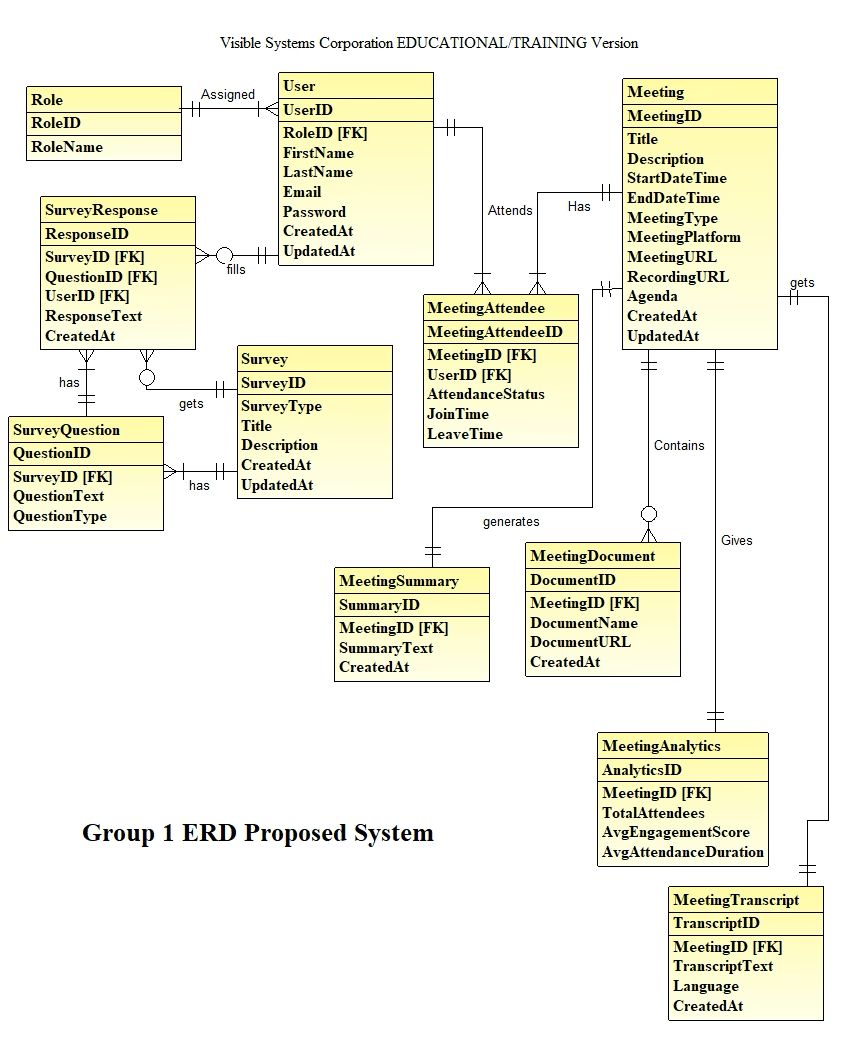
**Process Inputs:**

* From Meeting Database
  + Survey Responses

**Process Outputs:**

* To Organizer
  + Survey Responses with Meeting Data
* To Participant
  + Survey Responses with Meeting Data

**Entity Relationship Diagram**

****

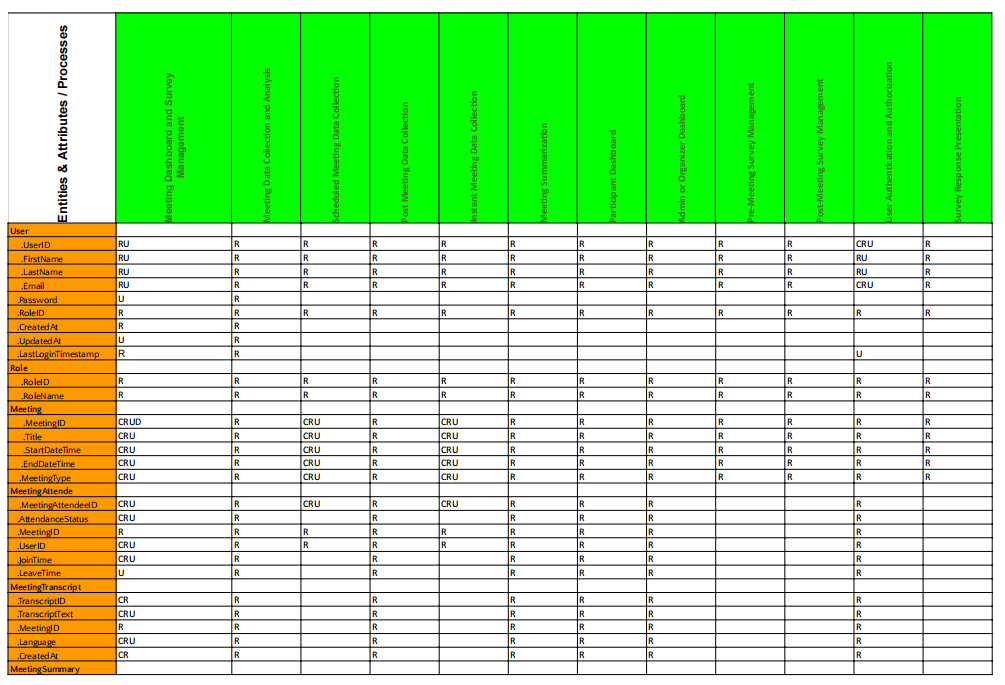
**Entities:**

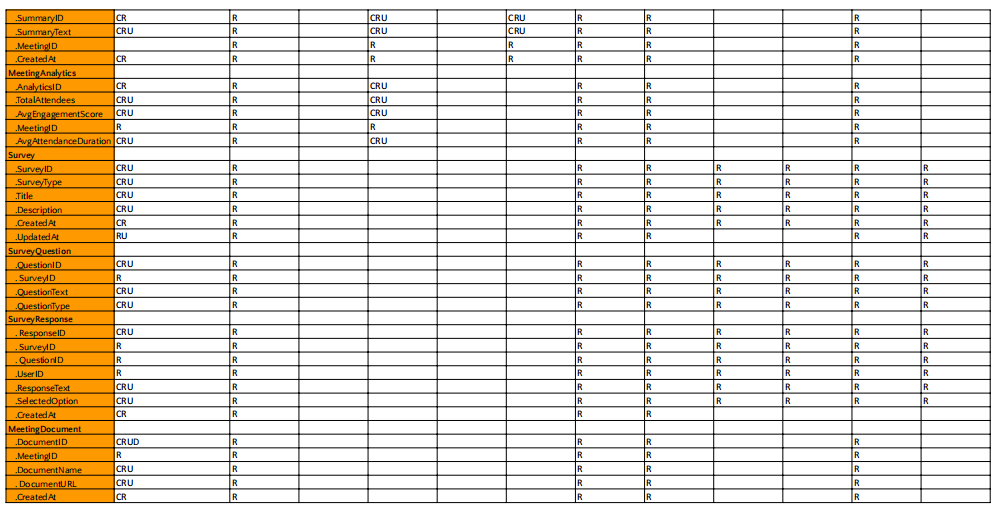
1. **User:** The User entity represents individuals who interact with the system, such as participants, organizers, and administrators.  
   1. Attributes:
      1. UserID (Primary Key): Uniquely identifies each user.
      2. FirstName: Stores the first name of the user.
      3. LastName: Stores the last name of the user.
      4. Email: Stores the email address of the user.
      5. Password: Stores the encrypted password of the user for authentication.
      6. Role ID(Foreign Key to Role): Indicates the role assigned to the user.
      7. CreatedAt: Stores the timestamp of when the user account was created.
      8. UpdatedAt: Stores the timestamp of when the user account was last updated.
2. **Role:** The Role entity represents the different roles that users can have within the system, such as Admin, Organizer, and Participant.  
   1. Attributes:
      1. RoleID (Primary Key): Uniquely identifies each role.
      2. RoleName: Stores the name of the role.
3. **Meeting:** The Meeting entity represents the meetings that are scheduled and conducted within the system.  
   1. Attributes:
      1. MeetingID (Primary Key): Uniquely identifies each meeting.
      2. Title: Stores the title or subject of the meeting.
      3. Description: Stores a brief description of the meeting.
      4. StartDateTime: Stores the start date and time of the meeting.
      5. EndDateTime: Stores the end date and time of the meeting.
      6. MeetingType: Indicates the type of meeting (e.g., Scheduled, Instant).
      7. MeetingPlatform: Indicates the platform used for the meeting (e.g., Zoom, WebEx, Teams).
      8. MeetingURL: Stores the URL or link to join the meeting.
      9. RecordingURL: Stores the URL or link to access the meeting recording.
      10. Agenda: Stores the agenda or key points to be discussed in the meeting.
      11. CreatedAt: Stores the timestamp of when the meeting was created.
      12. UpdatedAt: Stores the timestamp of when the meeting was last updated.
4. **MeetingAttendee:** The MeetingAttendee entity represents the association between users and the meetings they attend.
   1. Attributes:
      1. MeetingAttendeeID (Primary Key): Uniquely identifies each meeting attendee record.
      2. MeetingID (Foreign Key to Meeting): Indicates the meeting that the attendee is associated with.
      3. UserID (Foreign Key to User): Indicates the user who is attending the meeting.
      4. AttendanceStatus: Stores the attendance status of the user for the meeting (e.g., Attended, Absent).
      5. JoinTime: Stores the timestamp of when the user joined the meeting.
      6. LeaveTime: Stores the timestamp of when the user left the meeting.
5. **MeetingTranscript:** The MeetingTranscript entity represents the transcript or closed captions generated for a meeting.
   1. Attributes:
      1. TranscriptID (Primary Key): Uniquely identifies each meeting transcript.
      2. MeetingID (Foreign Key to Meeting): Indicates the meeting associated with the transcript.
      3. TranscriptText: Stores the text content of the transcript.
      4. Language: Stores the language of the transcript.
      5. CreatedAt: Stores the timestamp of when the transcript was created.
6. **MeetingSummary:** The MeetingSummary entity represents the summary generated for a meeting.  
     
     
     
     
   1. Attributes:
      1. SummaryID (Primary Key): Uniquely identifies each meeting summary.
      2. MeetingID (Foreign Key to Meeting): Indicates the meeting associated with the summary.
      3. SummaryText: Stores the text content of the summary.
      4. CreatedAt: Stores the timestamp of when the summary was created.
7. **MeetingAnalytics:** The MeetingAnalytics entity represents the analytics and metrics calculated for a meeting.
   1. Attributes:
      1. AnalyticsID (Primary Key): Uniquely identifies each set of meeting analytics.
      2. MeetingID (Foreign Key to Meeting): Indicates the meeting associated with the analytics.
      3. TotalAttendees: Stores the total number of attendees in the meeting.
      4. AvgEngagementScore: Stores the average engagement score of the meeting.
      5. AvgAttendanceDuration: Stores the average attendance duration of the meeting.
8. **Survey:** The Survey entity represents the surveys conducted within the system, such as pre-meeting and post-meeting surveys.
   1. Attributes:
      1. SurveyID (Primary Key): Uniquely identifies each survey.
      2. SurveyType: Indicates the type of survey (e.g., Pre-meeting, Post-meeting).
      3. Title: Stores the title or subject of the survey.
      4. Description: Stores a brief description of the survey.
      5. CreatedAt: Stores the timestamp of when the survey was created.
      6. UpdatedAt: Stores the timestamp of when the survey was last updated.
9. **SurveyQuestion:** The SurveyQuestion entity represents the questions associated with a survey.
   1. Attributes:
      1. QuestionID (Primary Key): Uniquely identifies each survey question.
      2. SurveyID (Foreign Key to Survey): Indicates the survey to which the question belongs.
      3. QuestionText: Stores the text of the question.
      4. QuestionType: Indicates the type of question (e.g., Multiple Choice, Open-ended).
10. **SurveyResponse:** The SurveyResponse entity represents the responses provided by users to survey questions.
    1. Attributes:
       1. ResponseID (Primary Key): Uniquely identifies each survey response.
       2. SurveyID (Foreign Key to Survey): Indicates the survey to which the response belongs.
       3. QuestionID (Foreign Key to SurveyQuestion): Indicates the question to which the response is associated.
       4. UserID (Foreign Key to User): Indicates the user who provided the response.
       5. ResponseText: Stores the text content of the response.
       6. CreatedAt: Stores the timestamp of when the response was created.
11. **MeetingDocument:** The MeetingDocument entity represents the documents associated with a meeting, such as agendas, presentations, or supporting materials.
    1. Attributes:
       1. DocumentID (Primary Key): Uniquely identifies each meeting document.
       2. MeetingID (Foreign Key to Meeting): Indicates the meeting to which the document belongs.
       3. DocumentName: Stores the name or title of the document.
       4. DocumentURL: Stores the URL or link to access the document.
       5. CreatedAt: Stores the timestamp of when the document was created.

**Relationships:**

1. User\_Attends\_Meeting (User (1) - (M) MeetingAttendee)
   1. Represents the relationship between users and the meetings they attend.
   2. A user can attend multiple meetings, and a meeting can have multiple users attending.
2. User\_fills\_To\_Survey (User (1) - (M) SurveyResponse)
   1. Represents the relationship between users and the survey responses they provide.
   2. A user can provide multiple survey responses, and a survey response belongs to a single user.
3. Role\_Assigned\_To\_User (Role (1) - (M) User)
   1. Represents the relationship between roles and users.
   2. A role can be assigned to multiple users, and a user is assigned a single role.
4. Meeting\_Has\_Attendees (Meeting (1) - (M) MeetingAttendee)
   1. Represents the relationship between meetings and their attendees.
   2. A meeting can have multiple attendees, and an attendee belongs to a single meeting.
5. Meeting\_gets\_Transcript (Meeting (1) - (1) MeetingTranscript)
   1. Represents the relationship between a meeting and its transcript.
   2. A meeting has one transcript, and a transcript belongs to a single meeting.
6. Meeting\_generates\_Summary (Meeting (1) - (1) MeetingSummary)
   1. Represents the relationship between a meeting and its summary.
   2. A meeting has one summary, and a summary belongs to a single meeting.
7. Meeting\_Gives\_Analytics (Meeting (1) - (1) MeetingAnalytics)
   1. Represents the relationship between a meeting and its analytics.
   2. A meeting has one set of analytics, and analytics belong to a single meeting.
8. Meeting\_Contains\_Documents (Meeting (1) - (M) MeetingDocument)
   1. Represents the relationship between meetings and their associated documents.
   2. A meeting can have multiple documents, and a document belongs to a single meeting.
9. Survey\_Has\_Questions (Survey (1) - (M) SurveyQuestion)
   1. Represents the relationship between surveys and their questions.
   2. A survey can have multiple questions, and a question belongs to a single survey.
10. Survey\_gets\_Responses (Survey (1) - (M) SurveyResponse)
    1. Represents the relationship between surveys and their responses.
    2. A survey can have multiple responses, and a response belongs to a single survey.
11. Question\_Has\_Responses (SurveyQuestion (1) - (M) SurveyResponse)
    1. Represents the relationship between survey questions and their responses.
    2. A survey question can have multiple responses, and a response belongs to a single survey question.

**SYNCHRONIZED SYSTEM MATRIX (CRUD)**





The comprehensive description below provides insights into how each process interacts with the system entities and performs CRUD operations to achieve its functionalities.

1. **Meeting Dashboard Process:**
   * Create (C): Not applicable for this process as it focuses on presenting data rather than creating it.
   * Read (R): The process reads data from entities like Meeting, MeetingAttendee, MeetingTranscript, MeetingSummary, MeetingAnalytics, and Survey. It gathers information about scheduled meetings, attendance, transcripts, summaries, analytics, and survey data.
   * Update (U): Not applicable for this process as it does not involve modifying existing data.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
2. **Survey Management Process:**
   * Create (C): The process creates surveys (Pre-meeting and Post-meeting) by defining survey questions, types, and schedules.
   * Read (R): It reads data from entities like Survey, SurveyQuestion, and SurveyResponse to retrieve survey details, questions, and responses.
   * Update (U): The process can update survey questions or schedules based on feedback or evolving meeting requirements.
   * Delete (D): Not applicable for this process as it does not involve deleting survey data.
3. **Meeting Data Collection & Analysis Process:**
   * Create (C): Not applicable for this process as it primarily focuses on reading and analyzing data.
   * Read (R): The process reads data from entities like Meeting, MeetingAttendee, MeetingTranscript, MeetingSummary, and MeetingAnalytics for analysis purposes.
   * Update (U): Not applicable for this process as it does not involve modifying existing data.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
4. **Scheduled Meeting Data Collection Process:**
   * Create (C): The process creates records for scheduled meetings in the Meeting and MeetingAttendee entities.
   * Read (R): It reads data from the Meeting and MeetingAttendee entities to retrieve information about scheduled meetings and attendees.
   * Update (U): The process can update scheduled meeting details if there are changes in schedules or meeting information.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
5. **Post Meeting Data Collection Process:**
   * Create (C): Not applicable for this process as it focuses on collecting data after meetings.
   * Read (R): The process reads data from entities like Meeting, MeetingAttendee, MeetingTranscript, MeetingSummary, and MeetingAnalytics for post-meeting analysis.
   * Update (U): Not applicable for this process as it does not involve modifying existing data.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
6. **Instant Meeting Data Collection Process:**
   * Create (C): The process creates records for instant meetings in the Meeting and MeetingAttendee entities.
   * Read (R): It reads data from the Meeting and MeetingAttendee entities to retrieve information about instant meetings and attendees.
   * Update (U): The process can update instant meeting details if there are changes in meeting information.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
7. **Meeting Summarization Process:**
   * Create (C): Not applicable for this process as it focuses on summarizing meetings rather than creating new records.
   * Read (R): The process reads meeting details, attendance, and transcripts to generate meeting summaries.
   * Update (U): The process can update existing meeting summaries if needed.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
8. **Participant Dashboard Process:**
   * Create (C): Not applicable for this process as it focuses on presenting data rather than creating or modifying it.
   * Read (R): The process reads data from various entities to provide participants with information about their associated meetings, attendance, transcripts, summaries, analytics, and survey data.
   * Update (U): Not applicable for this process as it does not involve modifying existing data.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
9. **Admin or Organizer Dashboard Process:**
   * Create (C): Not applicable for this process as it focuses on presenting data rather than creating or modifying it.
   * Read (R): The process reads data from various entities to provide admins or organizers with comprehensive information about all meetings, attendees, transcripts, summaries, analytics, survey data, and meeting documents.
   * Update (U): Not applicable for this process as it does not involve modifying existing data.
   * Delete (D): Not applicable for this process as it does not involve deleting data.
10. **Pre-Meeting Survey Management Process:**
    * Create (C): The process creates pre-meeting surveys by defining survey questions, types, and schedules.
    * Read (R): It reads data from entities like User, Role, Meeting, Survey, SurveyQuestion, and SurveyResponse to manage pre-meeting surveys and responses.
    * Update (U): The process can update pre-meeting survey questions or schedules based on feedback or evolving meeting requirements.
    * Delete (D): Not applicable for this process as it does not involve deleting survey data.
11. **Post-Meeting Survey Management Process:**
    * Create (C): Not applicable for this process as it focuses on managing post-meeting surveys rather than creating new records.
    * Read (R): The process reads data from entities like User, Role, Meeting, Survey, SurveyQuestion, and SurveyResponse to manage post-meeting surveys and responses.
    * Update (U): The process can update post-meeting survey questions or schedules based on feedback or evolving meeting requirements.
    * Delete (D): Not applicable for this process as it does not involve deleting survey data.
12. **User Authentication and Authorization Process:**
    * Create (C): The process creates new user records with user details such as UserID, FirstName, LastName, Email, Password, RoleID, CreatedAt, and UpdatedAt.
    * Read (R): It reads user and role information to authenticate users and determine their access levels and permissions within the system.
    * Update (U): The process can update user details such as names, emails, and last login timestamps.
    * Delete (D): Not applicable for this process as it does not involve deleting user data.

**PHYSICAL SYSTEM DESIGN**

**The Candidate System Solutions Table**

| **Characteristics** | **Description** | **Candidate 1**  **Trace3 in-house development**  **Custom solution** | **Candidate 2**  **COTS** | **Candidate 3**  **Outsourcing**  **Meeting Tracker Tool** | **Candidate 4**  **Outsourcing**  **Meeting Management Tool** |
| --- | --- | --- | --- | --- | --- |
| *Portion of System Computerized* | Brief Description of that portion of the system that would be computerized. | The entire system will be computerized. | Same as Candidate 1. | Same as Candidate 1. | Same as Candidate 1. |
| *Benefits* | Description of the business benefits that would be realized for this candidate. | High customization to meet organization's needs. | Established solution with proven track record. | Outsourcing allows focus on core business. | Same as Candidate 3. |
| *Servers and Workstations* | Description of the server and workstations needed to support this candidate. | Local office computers will run the software. The basic specs of a computer should be able to facilitate efficient operation of the tool. | Same as Candidate 1. | Same as Candidate 1. | Linux (8 core, 32GB RAM) |
| *Software Tools Needed* | Software tools needed to design and build the candidate. | Not applicable. | Same as Candidate 1. | Same as Candidate 1. | Whisper (OpenAI)  Supervisor (Deployment)  CRON Job (Task Scheduling)  nginx (Reverse Proxy & Serving) |
| *Application Software* | Description of the software to be purchased, built, accessed, or some combination of these techniques. | In-house development by Trace3. | COTS purchase by Trace3. | In-house development by UMD Smith (Outsourced by Trace3) | Same as Candidate 3. |
| *Method of Data Processing* | Generally some combination of online, batch, deferred batch, remote batch, and real time. | Real-time (dynamic scheduling). | Same as Candidate 1. | Same as Candidate 1. | Python |
| *Output Devices and Implications* | Description of output devices that would be used, special output requirements, and output considerations. | Standard computer device for output. | Same as Candidate 1. | Same as Candidate 1. | Dashboard |
| *Input Devices and Implications* | Description of: input methods to be used, input devices, special input requirements, and input considerations. | Standard computer device for input. | Same as Candidate 1. | Same as Candidate 1. | API Calls to Webex / Outlook Calendar |
| *Storage Devices and Implications* | Description of what data would be stored, what data would be accessed from existing stores, what storage media would be used, how much storage capacity would be needed, and how data would be organized. | All meeting data and user data, which is essential to the operation of the tool will be stored, backed up, processed and updated. The data will be stored on cloud. | Same as Candidate 1. | Same as Candidate 1. | MySQL/ PostgreSQL  Redis (Caching) |

**Feasibility Analysis Matrix**

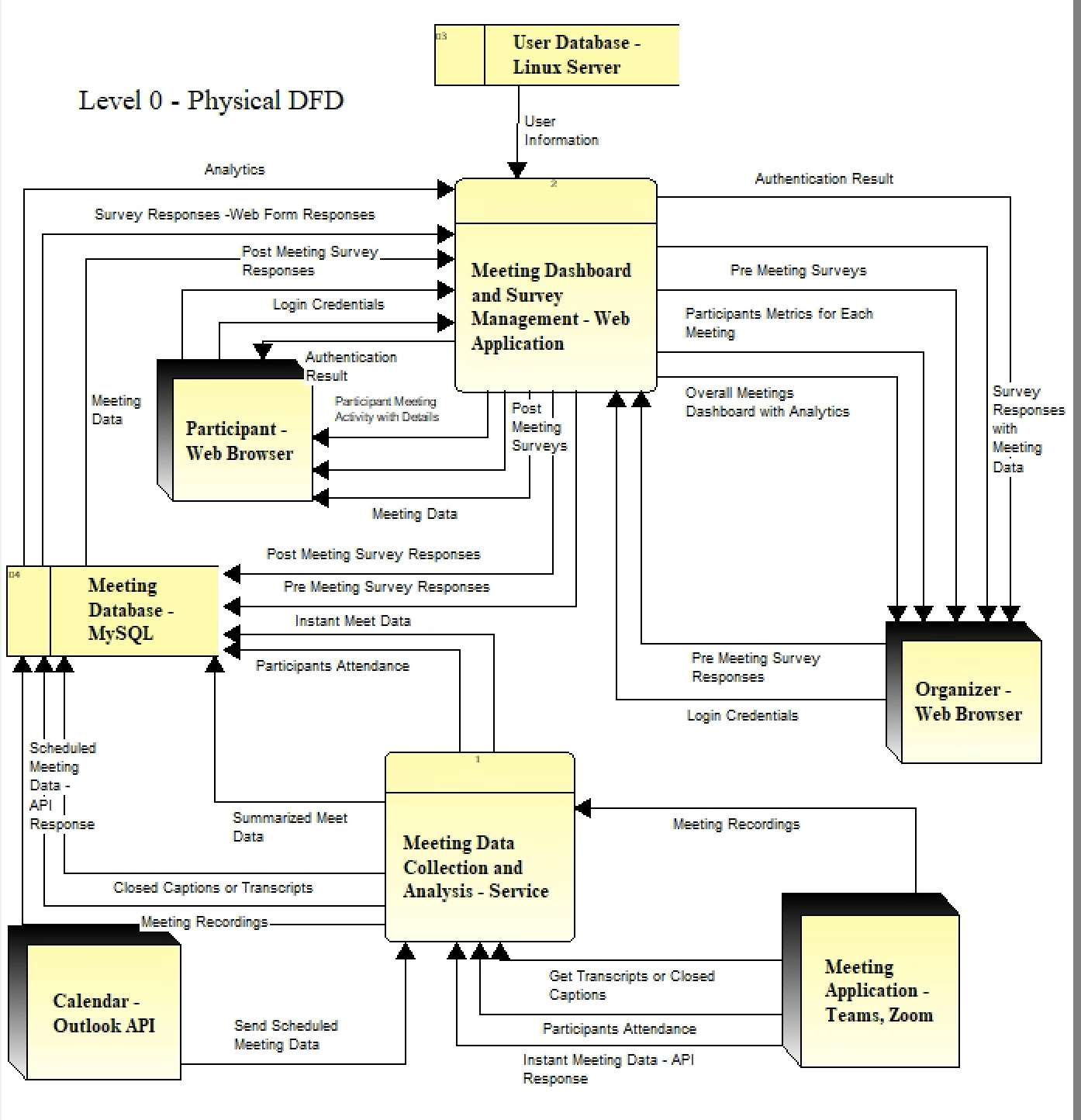
| **Feasibility Criteria** | **Description** | **Weight** | **Candidate 1**  **Trace3 in-house development**  **Custom solution** | **Candidate 2**  **COTS** | **Candidate 3**  **Outsourcing**  **Meeting Tracker Tool** | **Candidate 4**  **Outsourcing**  **Meeting Management Tool** |
| --- | --- | --- | --- | --- | --- | --- |
| ***Operational Feasibility*** | | | | | | |
| *Functionality* | A description of to what degree the candidate would benefit the organization and how well the system would work. | 25% | High customization to meet organization's needs.  SCORE = 95 | Established solution with proven track record.  SCORE = 80 | Outsourcing allows focus on core business.  SCORE = 65 | Outsourcing allows focus on core business.  SCORE = 85 |
| *Political* | A description of how well received this solution would be from the business owner, business employees, and the entire organizations perspective. | Likely well-received due to high customization potential.  SCORE = 90 | Well-received due to proven track record.  SCORE = 85 | Likely well-received due to outsourcing benefits.  SCORE = 75 | Likely well-received due to outsourcing benefits.  SCORE = 80 |
| ***Technical Feasibility*** | | | | | | |
| *Technology* | An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate. | 35% | All necessary technology needs to be developed / checked if already available.  SCORE = 75 | Technology is already available.  SCORE = 100 | Technology needs to be developed from scratch in this scenario.  SCORE = 60 | Technology needs to be developed from scratch in this scenario.  SCORE = 65 |
| *Expertise* | An assessment of the technical expertise needed to develop, operate, and maintain the candidate system. | Requires expertise in development.  SCORE = 60 | Level of expertise requirement is lower as compared to other candidates.  SCORE = 90 | Requires expertise in implementation.  SCORE = 70 | Requires expertise in implementation.  SCORE = 75 |
| ***Economic Feasibility*** | | | | | | |
| *Cost to develop (Or Buy)* | | 20% | Depends on in-house development costs.  $100,000 (estimate for internal development costs)  SCORE = 90 | Licensing costs.  $50,000 (estimate for licensing and setup)  SCORE = 90 | Outsourcing costs.  $80,000 (estimate for outsourcing dev costs)  SCORE = 75 | Outsourcing costs.  $75,000 (estimate for outsourcing dev costs)  SCORE = 75 |
| *Payback period (discounted)* | | 2 years | 1.5 years | 2.5 years | 2 years |
| *Net present value* | | $250,000 assuming a 10% discount rate over 5 years | $180,000 assuming a 10% discount rate over 5 years | $120,000 assuming a 10% discount rate over 5 years | $175,000 assuming a 10% discount rate over 5 years |
| ***Schedule Feasibility*** | | | | | | |
| *Schedule* | An assessment of how long the solution will take to design and implement. | 20% | > 4 months | < 1 month | < 2 months | ~ 3 months |
| ***Ranking*** | Based on scores of all feasibility criterion | 100% | 74.25 | 73.25 | 71.5 | 71.25 |

***Rationale for Weights:***

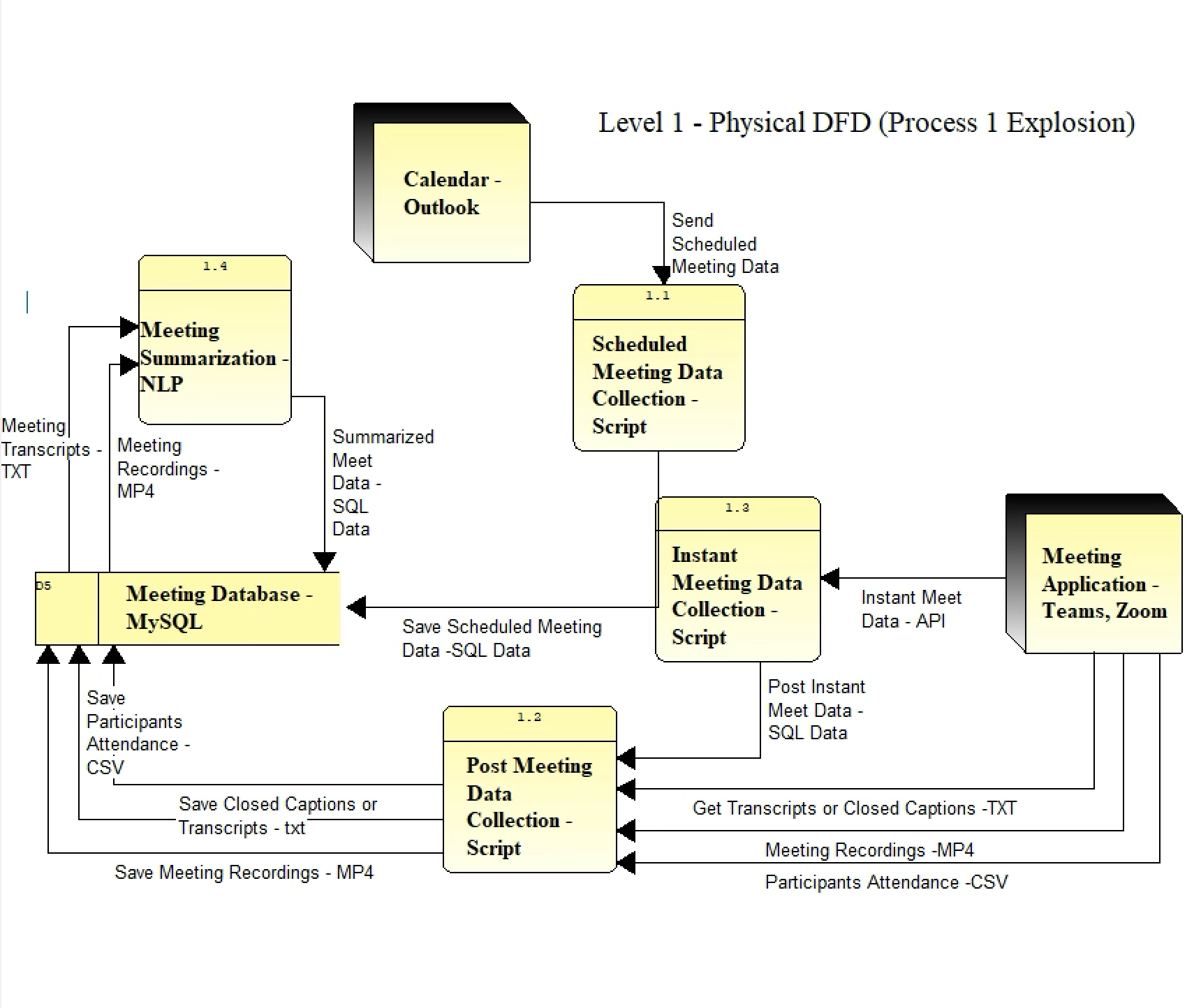
We rank the elements of feasibility in the following order, and list their associated weights and justifications:

1. Economic Feasibility | Weight: 20% : This category is weighted based on the kind of devices and data processing tools which will be required for the project.
2. Operational Feasibility | Weight: 25% : This category is second on the list of importance to the degree to which the candidate would benefit the organization as well as the degree to which the candidate will be accepted by the organization, based on the requirements & the intricacies of the business problem.
3. Technical Feasibility | Weight: 35% : technical feasibility will have significant influence over the system proposal since the system architecture needs to be mature enough to be well understood, used, and modified for future needs by Trace3 employees.
4. Schedule Feasibility | Weight: 20% : The timeline for the project has been decided well in advance, resulting in the appropriate weight for this type of feasibility.

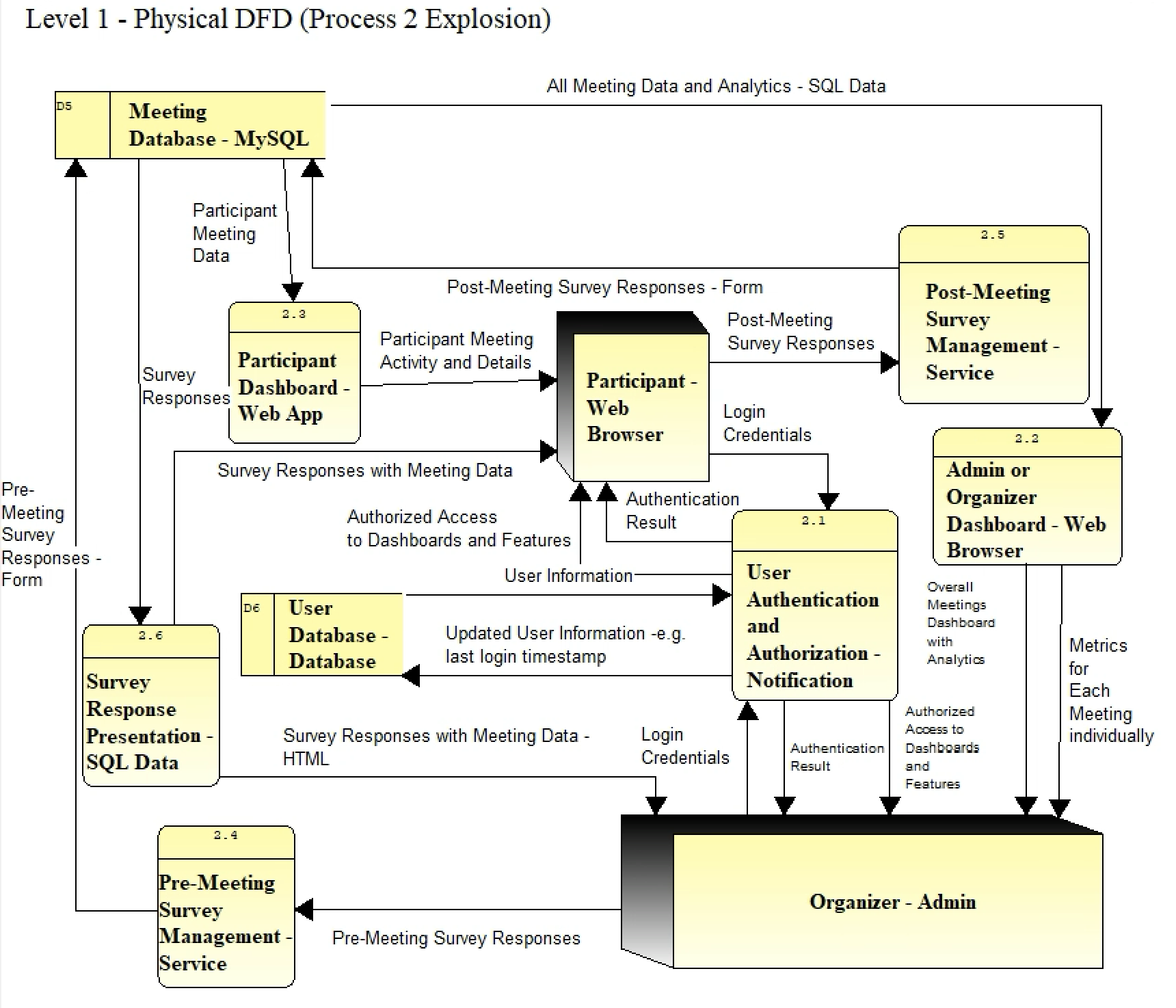
**Physical Data Flow Diagram**

**Level 0 - Physical Data Flow Diagram:**  
  
  
  
The Level 0 Physical DFD consists of two main processes: Meeting Data Collection and Analysis, and Meeting Dashboard and Survey Management. The Meeting Data Collection and Analysis process interacts with the Outlook Calendar to retrieve scheduled meeting data using an API. It also communicates with the Meeting Application (Zoom, Webex, or Teams) to collect post-meeting data such as closed captions, transcripts, participants' attendance, and meeting recordings. This data is stored in the Meeting Database (MySQL).

The Meeting Dashboard and Survey Management process retrieves data from the Meeting Database and presents it to the Participants and Organizers through a web application. Participants can view their meeting activity and details, while Organizers have access to overall meeting dashboards with analytics. The Survey Management component sends pre-meeting surveys to Organizers and post-meeting surveys to Participants, storing the responses in the Meeting Database. The User Database (Linux Server) is used for user authentication and authorization purposes, ensuring secure access to the system.

**Level 1 - Process 1 - Physical Data Flow Diagram:**   
  
  
The Level 1 - Process 1 Physical DFD focuses on four sub-processes: Scheduled Meeting Data Collection, Post Meeting Data Collection, Instant Meeting Data Collection, and Meeting Summarization. The Scheduled Meeting Data Collection sub-process retrieves scheduled meeting data from the Outlook Calendar using an API and stores it in the Meeting Database.   
The Post Meeting Data Collection sub-process interacts with the Meeting Application to collect post-meeting data such as closed captions, transcripts, participants' attendance, and meeting recordings. This data is then stored in the Meeting Database.

The Instant Meeting Data Collection sub-process uses APIs of the Meeting Application to identify meetings that occurred instantly without prior scheduling. It collects data for these meetings and stores it in the Meeting Database. The Meeting Summarization sub-process utilizes the meeting transcripts or recordings stored in the Meeting Database to generate meeting summaries. If transcripts are not available, it uses speech-to-text tools like Whisper by OpenAI to convert the audio into text before summarizing the meeting. This process is run at the end of each day using a scheduled task or service.

**Level 1 - Process 2 - Physical Data Flow Diagram:**  
  
  
The Level 1 - Process 2 Physical DFD focuses on the Meeting Dashboard and Survey Management process. It consists of five sub-processes: Participant Dashboard, Admin/Organizer Dashboard, User Authentication and Authorization, Pre-Meeting Survey Management, and Post-Meeting Survey Management. The Participant Dashboard sub-process retrieves relevant meeting data from the Meeting Database and presents it to the Participants through a web application. Participants can view their meeting activity, including details such as invitees, attendance, agenda, and summary.

The Admin/Organizer Dashboard sub-process provides Organizers with access to overall meeting dashboards containing analytics and metrics derived from all the meetings stored in the Meeting Database. The User Authentication and Authorization sub-process interacts with the User Database to verify user credentials and determine their access rights to the system. The Pre-Meeting Survey Management sub-process sends pre-meeting surveys to Organizers, allowing them to provide information such as the meeting agenda and supporting documentation. The responses are stored in the Meeting Database.

The Post-Meeting Survey Management sub-process sends post-meeting surveys to Participants, collecting their feedback on various aspects of the meeting. The responses are stored in the Meeting Database and can be viewed by both Organizers and Participants.

All in all, these Physical DFDs provide a detailed view of how the Meeting Management System would be implemented using physical technologies and components, showcasing the flow of data and the interactions between various parts of the new proposed system.

**INPUT AND OUTPUT DESIGN**

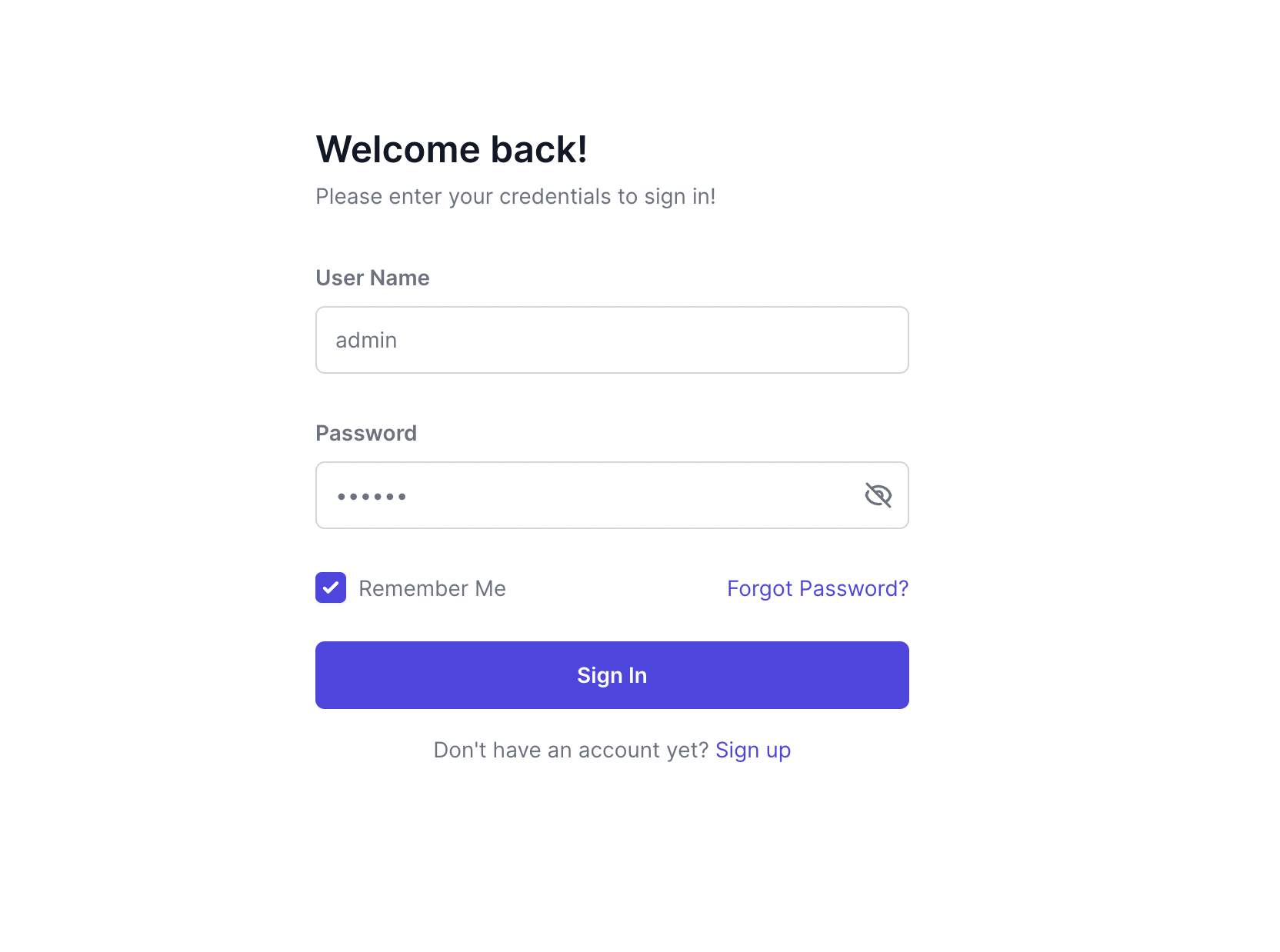
*Inputs*

*System Inputs:*

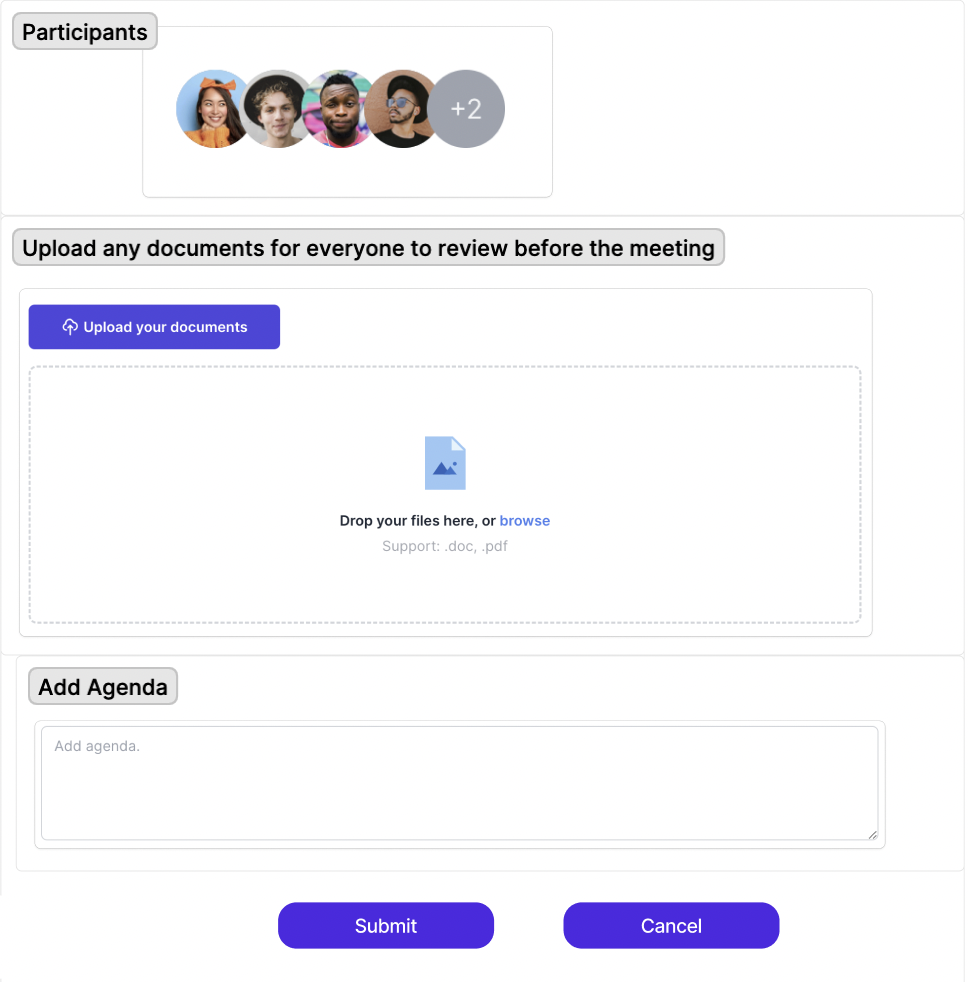
1. We will import scheduled meeting information from Microsoft Outlook using recurring cron jobs.
2. After the meeting is over we will fetch data via API from the Video Conferencing application.

*User Inputs*

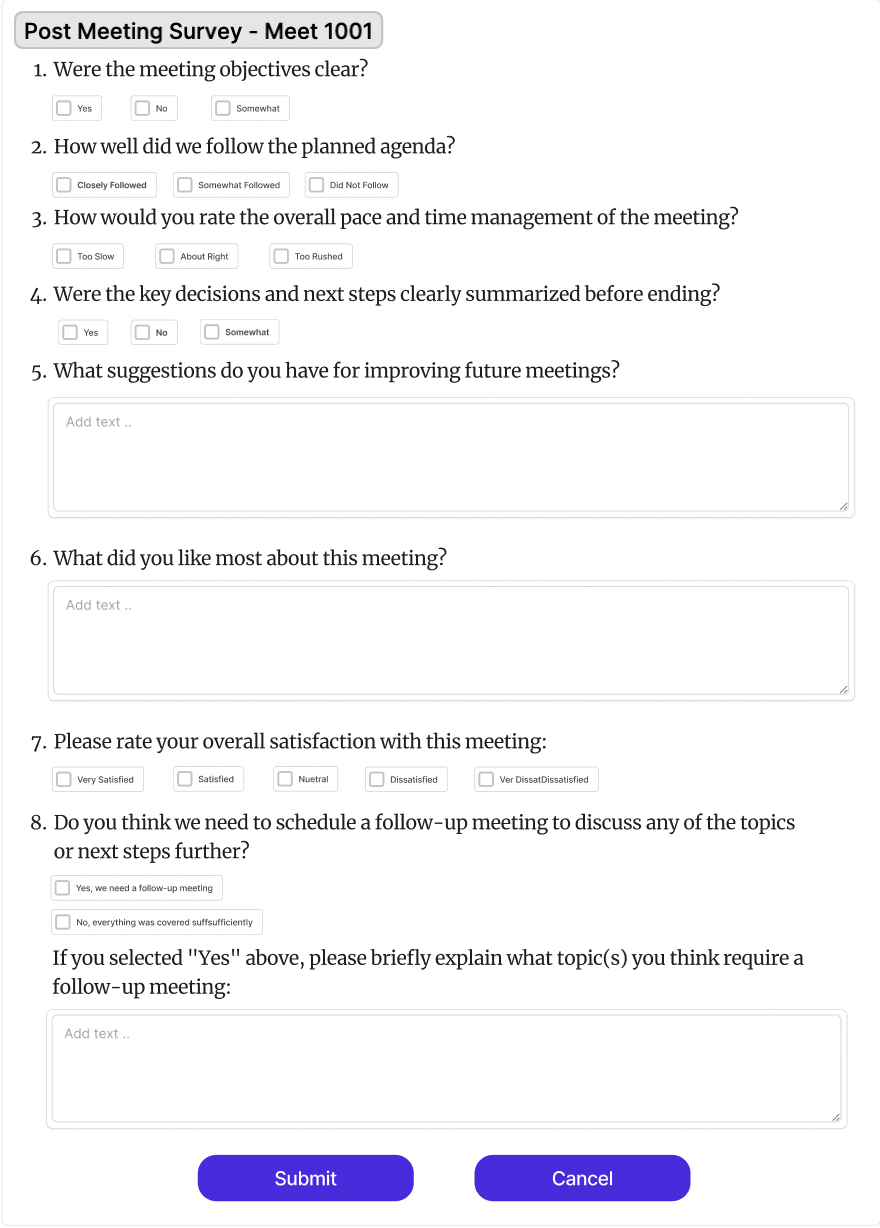
This image depicts a user login screen for the meeting management system. It has fields for the user to enter their email address and password. There are also options to select the user's role, such as administrator, organizer, or participant. The "Login" button allows the user to access the system after entering their credentials. Additionally, there is a link for users to click if they forgot their password.

**

This image illustrates a user interface for scheduling a new meeting. It includes fields for the user to input various details about the meeting. There are text fields for the meeting title, date, time, and location. The user can also select the meeting type from a dropdown menu (e.g., team meeting, client meeting, project review). Additionally, there are fields to enter the meeting agenda and choose the meeting attendees from a list or by entering their email addresses. At the bottom, there are buttons to "Schedule" the meeting or "Cancel" the scheduling process.

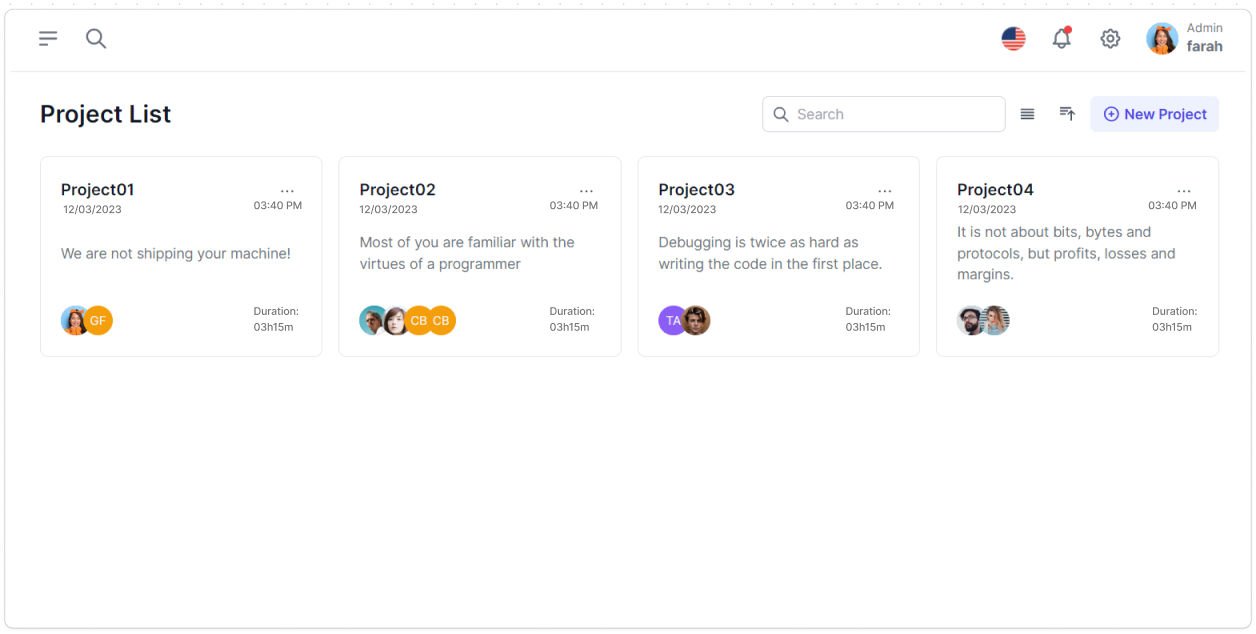
**

The image shows a user interface for a post-meeting survey. It allows participants to provide feedback on the meeting's effectiveness, rating various aspects such as the clarity of the agenda, productivity, and time management. Participants can also offer suggestions for improvement and indicate their overall satisfaction with the meeting. The "Submit" button at the bottom allows users to send their responses, while the "Cancel" button discards the feedback. This survey interface enables the collection of valuable insights to enhance future meetings.

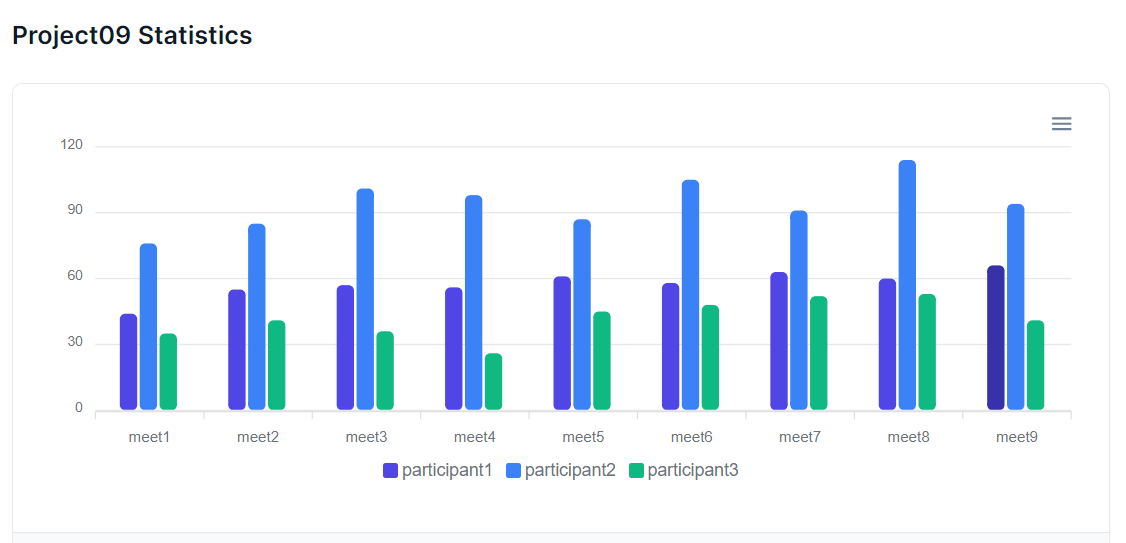
**

*Outputs*

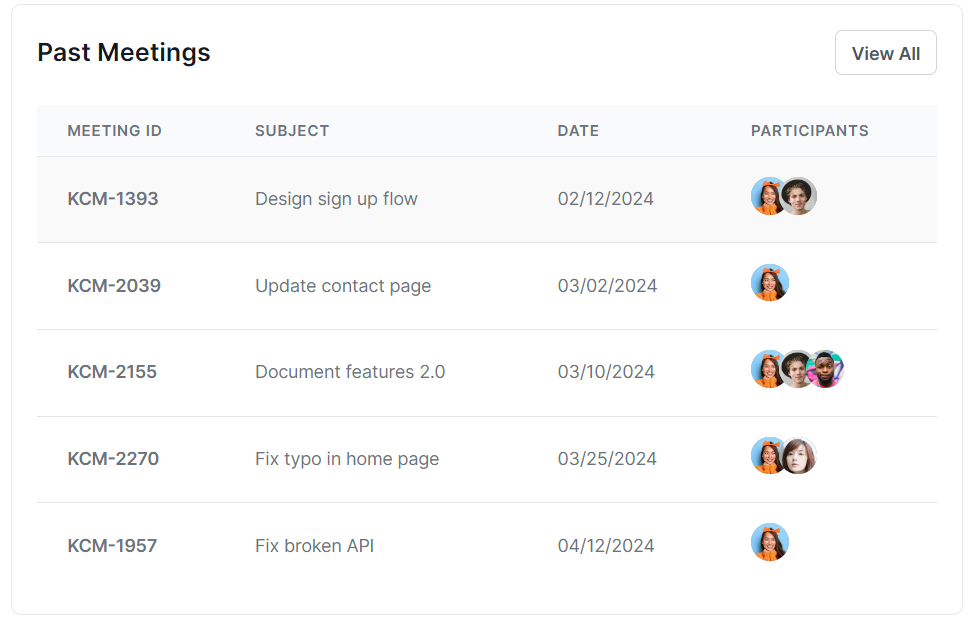
This image illustrates a user interface that displays a list of projects. The projects are organized in a table format, with columns for the project name, start date, end date, status, and a link to view more details. Users can browse through the list to find specific projects and click on the "View Details" link to access additional information about each project. This interface provides a centralized view of all projects and their key details.



The image depicts a dashboard presenting various meeting statistics. The dashboard includes visual elements such as graphs, charts, and key performance indicators (KPIs) to provide an overview of meeting-related data. The statistics may include the total number of meetings, average meeting duration, attendance rate, and the distribution of meeting types. This dashboard allows users to quickly grasp important insights and trends related to meetings within the organization.

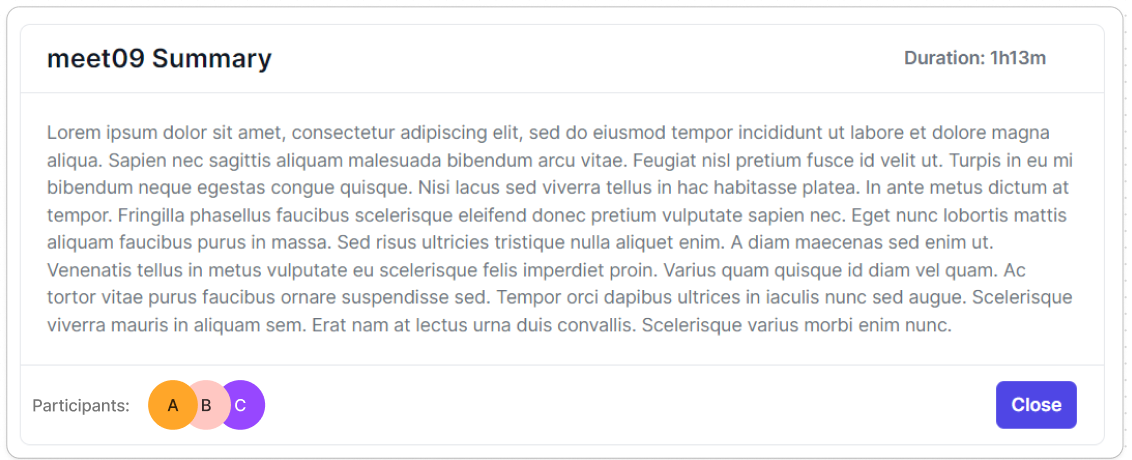


This image shows a user interface displaying a list of past meetings. The meetings are presented in a table format, with columns for the meeting title, date, time, organizer, and a link to view the meeting summary. Users can scroll through the list to find specific meetings and click on the "View Summary" link to access more detailed information about each meeting. The interface provides an easy way to browse and retrieve past meeting records.



The image shows a user interface presenting a summary of a specific meeting. The summary includes the meeting title, date, time, location, and organizer. Below that, there is a section displaying the meeting attendees, along with their names and roles. The main part of the summary consists of a detailed description of the meeting, including the agenda items discussed, decisions made, action items assigned, and any other relevant notes. This interface allows users to quickly review the key points and outcomes of a particular meeting.

These output designs showcase different aspects of the meeting management system, providing users with access to past meeting records, meeting statistics, project lists, and detailed meeting summaries. They offer a comprehensive view of the system's reporting and information retrieval capabilities, enabling users to effectively manage and analyze meeting-related data.



**IMPLEMENTATION PLAN**

We will use an abrupt cutover process to implement our new management tool. We have chosen this method because there is no existing system for this purpose. There will be a point where Trace3 will stop using their original workflow and use the updated new system. Despite the rapid nature of the implementation, we utilize the relevant safeguards to protect Trace3's proprietary data and ensure business continuity.  
  
**1. Preparatory Phase:**   
  
Procurement of necessary software APIs for development. We will also create different cron jobs to import existing projects and meeting information.We are planning to use Whisper AI to convert speech to text to generate summary of meetings. We will use the Whisper package available in python.

- <https://platform.openai.com/docs/guides/speech-to-text>

We will be using dummy data provided by Trace3 itself to configure our systems to fit the data. In addition to the above, we’ll be using hardware components like linux servers, NGINX for reverse proxy and a supervisor that will be running the scheduled tasks. All required software, hardware suites will be arranged to ensure necessary development. Development phase will also be carried out which will involve integrating the frontend, backend and server parts of the project.   
  
**2. Installation, Configuration and Testing:**

After development of the platform is completed, it will be deployed on the host server. This will involve setting up Linux servers to host the development environment, installing and configuring cronjob, whisper software suites, configuring NGINX as a reverse proxy to handle incoming requests, setting up the supervisor to manage and monitor scheduled tasks. Integrating the OpenAI Speech-to-Text API and Cron Jobs into our development environment.

Configuring the system according to the data provided by Trace3 to ensure compatibility with their systems. Additionally, we will conduct alpha testing to ensure that all components are functioning as expected. This includes testing the functionality of software suites, verifying the Speech-to-Text conversion performance, etc.

**3. Data Migration:**   
  
Once the installation, configuration, and initial testing are completed, the focus will shift to data migration. This part involves transferring existing meeting data from Trace3's current systems to our newly developed platform. Scheduled meeting data extracted from Outlook calendars will be migrated and stored in the meeting database. Additionally, any supplementary meeting data obtained through APIs will be verified and seamlessly integrated into our system. Frameworks will be invoked to ensure the accuracy and integrity of the transferred data, enabling uninterrupted continuity of meeting management operations.

**4. Conduct Beta Testing:**   
  
Following the successful migration of data, the system will undergo comprehensive beta testing to validate its functionality and performance in a real-world environment. This testing phase involves inviting a select group of users, including administrators, organizers, and participants, to interact with the system and provide feedback. Testing scenarios will include various use cases, such as scheduling meetings, accessing meeting summaries, and analyzing meeting analytics. Any identified issues or discrepancies will be addressed in-time to refine the system and enhance its overall usability and effectiveness.  
  
**5. Bring the New System Online, and Drop the Old System:**  
  
With beta testing completed and any necessary adjustments made, the final step is to bring the new meeting management system online and retire the old system. This process involves deploying the system on production servers, configuring it for live operation, and conducting final checks to ensure everything is functioning optimally.   
Once verified, users will transition from the old system to the new system seamlessly. Any remaining data or functionalities from the old system will be migrated or integrated into the new system as needed. Finally, the old system will be discontinued, marking the successful transition to the upgraded meeting management solution, which is being built by our team.

**LESSONS LEARNED**

1. Utilizing tools like Visible Analyst to build DFDs and ERDs not only helped us organize our thoughts but also made complex system interactions more understandable to everyone involved. This reinforced the value of visual aids in making technical information accessible and clear.
2. This project reminded us of the importance of effective project management and the power of a collaborative team. Navigating through the complexities of this project, we learned to value each team member's contributions and the impact of organized coordination on meeting our project goals.
3. The evolving scope of the project, driven by ongoing discussions with Trace3, taught us to remain flexible and responsive. We've come to appreciate the ability to adapt to changes as a crucial skill that keeps our work relevant and on point.
4. The fact-finding techniques, such as discussions with Trace3's strategic initiative team and observations of their current processes, proved invaluable in understanding their current system and requirements. We learned that thorough requirements gathering is the foundation for a successful system design.
5. Through our interactions with Trace3, we gained insights into their business processes and pain points, which helped us tailor the system design to their specific needs. This emphasized the importance of actively listening to the client and asking the right questions.
6. Evaluating and selecting the appropriate technologies for the proposed system solution was a critical task. We considered factors such as feasibility, scalability, compatibility with Trace3's existing systems, and the team's expertise.
7. We learned that careful technology selection is essential for project success. It involves striking a balance between meeting the client's requirements and considering the technical constraints and capabilities of the development team.
8. We encountered some challenges in coordinating work and resolving conflicts, but through open discussions and a focus on common goals, we were able to overcome these obstacles and maintain a positive team dynamic.

—-end report—-