



الجامعة السورية الخاصة  
SYRIAN PRIVATE UNIVERSITY

**Syrian Private University**

**Faculty Of Artifical Intelligence Engineering**

## **BI VOICE AGENT**

Prepared by

*Massa Mohammad Jamal Nasri*

*Ayman Naeem Alkotyfan*

Supervisors

Eng.Anas Abdualziz

Academic Year: 2025-2026



الجامعة السورية الخاصة  
SYRIAN PRIVATE UNIVERSITY

الجامعة السورية الخاصة

كلية هندسة الذكاء الصنعي

## مساعد صوتي لذكاء الأعمال

إعداد

ماسة محمد جمال نصرى  
أيمان نعيم القطيفان

اشراف

المهندس أنس عبد العزيز

السنة الدراسية 2026-2025

## Supervisor certification

I certify that the preparation of this project entitled BI Voice Agent, prepared by Ayman Al-kotyfan and Massa Nasri, was carried out under my supervision in the Department of Software Engineering and Artificial Intelligence as part of the requirements for obtaining a Bachelor's degree at the College of Artificial Intelligence Engineering.

| Name               | Role / Position     | Signature | Date               |
|--------------------|---------------------|-----------|--------------------|
| Eng. AnasAbdulaziz | Academic Supervisor |           | ____ / ____ / 2026 |

# Abstract

**BI Voice Agent** is an intelligent Business Intelligence (BI) system designed to enable users to interact with analytical databases using natural spoken language instead of traditional query interfaces. The primary objective of the system is to simplify data analysis for non-technical users by transforming voice-based questions into actionable insights in real time.

The system adopts a modular, service-oriented architecture that integrates Speech-to-Text, Natural Language Understanding, SQL generation, data processing, and visualization layers. User voice input is first transcribed using an automatic speech recognition module, after which the resulting text is analyzed to determine the intent and classify the query type. Analytical questions are then converted into structured SQL queries compatible with analytical databases, while non-analytical queries are filtered out early in the pipeline.

To support large-scale and real-time data analytics, BI Voice Agent relies on an ETL (Extract, Transform, Load) pipeline built on a distributed architecture. Data is ingested from heterogeneous sources, processed through streaming and transformation services, and stored in an analytical data warehouse optimized for fast querying. The system continuously monitors ETL execution states and metadata to ensure reliability, consistency, and accurate reporting.

The final results are presented to the user in the form of dynamic charts and dashboards, allowing seamless exploration of insights without requiring technical knowledge of SQL or BI tools. By combining voice interaction, data engineering, and intelligent query generation, BI Voice Agent provides an intuitive, scalable, and efficient solution for modern business intelligence applications.

## الملخص

**BI Voice Agent** هو نظام ذكاء أعمال (Business Intelligence) ذكي يهدف إلى تمكين المستخدمين من التفاعل مع قواعد البيانات التحليلية باستخدام اللغة الصوتية الطبيعية بدلاً من واجهات الاستعلام التقليدية. يهدف المشروع بشكل أساسي إلى تبسيط عملية تحليل البيانات، خاصة للمستخدمين غير التقنيين، من خلال تحويل الأسئلة الصوتية إلى نتائج تحليلية واضحة وبشكل فوري.

يعتمد النظام على بنية معمارية معيارية قائمة على الخدمات، حيث يتم دمج وحدات تحويل الصوت إلى نص، وفهم اللغة الطبيعية، وتوليد استعلامات SQL ، ومعالجة البيانات، وعرض النتائج. تبدأ العملية باستقبال صوت المستخدم وتحويله إلى نص، ثم تحليل النص لاستخراج نية السؤال وتحديد نوعه. في حال كان السؤال تحليليًا، يتم تحويله تلقائيًا إلى استعلام SQL متوافق مع قواعد البيانات التحليلية، بينما يتم إيقاف الأسئلة غير التحليلية في مراحل مبكرة من النظام.

لدعم تحليل البيانات الضخمة والزمن الحقيقي، يعتمد النظام على خط أنابيب ETL (الاستخلاص، التحويل، التحميل) مبني على بنية موزعة. يتم جلب البيانات من مصادر متعددة، معالجتها عبر خدمات بث وتحويل، ثم تحميلها إلى مستودع بيانات تحليلي عالي الأداء. كما يقوم النظام بتتبع حالات تنفيذ الـETL والبيانات الوصفية لضمان الاستقرار والدقة والموثوقية.

في المرحلة النهائية، يتم عرض النتائج للمستخدم على شكل رسوم بيانية ولوحات معلومات تفاعلية، مما يتيح استكشاف البيانات بسهولة دون الحاجة لمعرفة تقنية مسبقة بـ SQL أو أدوات ذكاء الأعمال. يجمع **BI Voice Agent** بين التفاعل الصوتي، وهندسة البيانات، والذكاء الاصطناعي ليقدم حلًا حديثاً، مرتناً، وقابلًا للتوسيع في مجال ذكاء الأعمال.

## TABLE OF CONTENTS

|   |           |
|---|-----------|
| Supervisor certification.....   | 3         |
| Abstact.....  | 4         |
| الملخص .....  | 5         |
| <b>Chapter 1 Introduction .....</b>   | <b>12</b> |
| <i>1. Introduction .....</i>  | 13        |
| <i>2. Problem Statement.....</i>  | 13        |
| <i>3. Project Objective.....</i>  | 14        |
| <i>4. Proposed System .....</i>   | 14        |
| <i>5. Report Organization .....</i>   | 15        |
| <i>6. Summary .....</i>   | 15        |
| <b>Chapter 2: Basic Concepts and Reference Study .....</b>                            | <b>16</b> |
| <b>I. Reference Study of Interactive AI Powered Business Intelligence Tools .....</b> | <b>17</b> |
| <i>1. Introduction .....</i>  | 17        |
| <i>2. Fundamental Concepts .....</i>  | 17        |
| <i>3. Literature Review for the system .....</i>                                      | 19        |
| <b>II. Comparative Analysis of AI-Powered Interactive BI Platforms .....</b>          | <b>26</b> |
| <i>4. Literature Review for Voice-Based Business Intelligence Systems .....</i>       | 28        |
| <b>Chapter 3 Project Management.....</b>  | <b>32</b> |
| <i>1. Introduction .....</i>  | 33        |
| <i>2. Project Charter .....</i>   | 33        |
| <i>3. Statement of Work (SOW) .....</i>   | 34        |
| 1. Project Description and Objectives.....  | 34        |
| 2. Project Scope .....  | 34        |
| 3. Project Goals.....   | 34        |
| <i>4. Project Requirements .....</i>  | 35        |
| <i>5. Assumptions .....</i>   | 35        |
| <i>6. Project Resources .....</i>   | 35        |
| Human Resources.....  | 35        |
| Technical Resources.....  | 35        |
| <i>7. Schedule .....</i>  | 35        |
| <i>8. Risk Management .....</i>   | 35        |
| <i>9. Summary .....</i>   | 36        |
| <b>Chapter 4: System Analysis .....</b>   | <b>37</b> |
| <i>1. Introduction .....</i>  | 38        |

|   |    |
|---|----|
| <b>2. Project Timeline .....</b>                                    | 38 |
| <b>3. Software Requirement specification document ( SRS ) .....</b> | 39 |
| <b>1. Introduction .....</b>  | 39 |
| <b>2. Overall Description .....</b>                                 | 43 |
| <b>3. System Features .....</b>                                     | 46 |
| <b>4. System Requirements .....</b>                                 | 53 |
| <b>5. Requirements modeling .....</b>                               | 56 |
| • <b>Sign Up .....</b>  | 56 |
| • <b>Email verification .....</b>                                   | 58 |
| • <b>Login.....</b>   | 60 |
| • <b>Logout.....</b>  | 62 |
| • <b>Manage Profile .....</b>                                       | 64 |
| • <b>Deactivate My Profile .....</b>                                | 66 |
| • <b>Edit Workspace Info.....</b>                                   | 68 |
| • <b>View Workspace Members List.....</b>                           | 70 |
| • <b>Invite Members .....</b>                                       | 71 |
| • <b>Assign Roles .....</b>   | 73 |
| • <b>Remove Members .....</b>                                       | 75 |
| • <b>Suspend Member.....</b>  | 77 |
| • <b>Accept Invitation .....</b>                                    | 79 |
| • <b>Record Voice and Generate Report .....</b>                     | 82 |
| • <b>Upload Voice and Generate Report .....</b>                     | 84 |
| • <b>View Dashboard Report.....</b>                                 | 86 |
| • <b>View Previous Reports.....</b>                                 | 87 |
| • <b>View Report Details .....</b>                                  | 88 |
| • <b>Edit SQL Query and Re-execute Rep.....</b>                     | 89 |
| • <b>Edit Transcription and Regenerate Report .....</b>             | 90 |
| • <b>Export Report as Image .....</b>                               | 91 |
| • <b>Delete Report .....</b>  | 92 |
| • <b>Upload Database and Execute ETL Pipeline .....</b>             | 93 |
| • <b>Delete Database from ClickHouse.....</b>                       | 95 |
| <b>6. System architecture .....</b>                                 | 97 |
| • <b>Class Diagram .....</b>  | 97 |
| • <b>ERD Diagram.....</b>   | 98 |

|  |     |
|--|-----|
| <i>7. Initial Test Cases</i> .....                   | 99  |
| <b>Chapter 5</b> .....                               | 120 |
| <b>System Design</b> .....                           | 120 |
| <i>1-Introduction</i> .....                          | 121 |
| <i>2-High-Level System Design</i> .....              | 121 |
| <i>3-Architecture Layers</i> .....                   | 123 |
| <i>4. Detailed design for system component</i> ..... | 126 |
| <i>4.1-Design class diagram</i> .....                | 126 |
| <b>Chapter 6</b> .....                               | 141 |
| <b>Practical Implementation</b> .....                | 141 |
| <i>6.1 Introduction</i> .....                        | 142 |
| <i>6.2 Used Tools</i> .....                          | 142 |
| <i>6.4 Summary</i> .....                             | 158 |
| <b>Chapter 7</b> .....                               | 159 |
| <b>Report Overview</b> .....                         | 159 |
| <i>7.1 Introduction</i> .....                        | 160 |
| <i>7.2 Report Structure and Purposes</i> .....       | 160 |
| <i>7.3 Summary</i> .....                             | 162 |
| <i>References</i> .....                              | 163 |

## List of Tables

|   |     |
|---|-----|
| Table 1 : Comparative Analysis of Reference Studies.....                                | 28  |
| Table 2 : Thematic Analysis of AI Techniques in Voice-Based Business Intelligence ..... | 30  |
| Table 3 : Roles and Responsibilities of Project Team Members.....                       | 34  |
| Table 4 : Risk Management Matrix.....   | 36  |
| Table 5 : Project Timeline and Sprint Planning.....                                     | 38  |
| Table 6 System Requirements Specification (SRS) .....                                   | 55  |
| Table 7 : Use Case Specification for "Sign Up".....                                     | 57  |
| Table 8 : Use Case Specification for "Email Verification" .....                         | 58  |
| Table 9 : Use Case Specification for "Login" .....                                      | 61  |
| Table 10 : Use Case Specification for "Logout" .....                                    | 62  |
| Table 11 : Use Case Specification for "Manage Profile" .....                            | 64  |
| Table 12 : Use Case Specification for "Deactive My Profile" .....                       | 66  |
| Table 13 : Use Case Specification for "Edit Workspace Info " .....                      | 68  |
| Table 14 : Use Case Specification for "View Workspace Member list" .....                | 70  |
| Table 15 : Use Case Specification for "Invite Member".....                              | 71  |
| Table 16 : Use Case Specification for "Assign Roles" .....                              | 73  |
| Table 17 : Use Case Specification for "Remove Members" .....                            | 75  |
| Table 18 : Use Case Specification for "Suspend Member" .....                            | 77  |
| Table 19 : Use Case Specification for "Accept Invitation" .....                         | 80  |
| Table 20 : Use Case Specification for "Record Voice".....                               | 82  |
| Table 21 : Use Case Specification for "Upload Voice" .....                              | 84  |
| Table 22 : Use Case Specification for "View Dashboard report" .....                     | 86  |
| Table 23 : Use Case Specification for "View Previous Reports" .....                     | 87  |
| Table 24 : Use Case Specification for "View Report Details" .....                       | 88  |
| Table 25 : Use Case Specification for "Edit SQL Query" .....                            | 89  |
| Table 26 : Use Case Specification for "Edit Transcription" .....                        | 90  |
| Table 27 : Use Case Specification for "Export Report" .....                             | 91  |
| Table 28 : Use Case Specification for "Delete Report" .....                             | 92  |
| Table 29 : Use Case Specification for "Upload Database".....                            | 93  |
| Table 30 : Use Case Specification for "Delete Database" .....                           | 95  |
| Table 31 : Test cases for User Registration .....                                       | 100 |
| Table 32 : Test Cases for User Login.....   | 101 |
| Table 33 : Test Cases for Email Verification.....                                       | 102 |
| Table 34 : Test Cases for Authentication and Token Management .....                     | 102 |
| Table 35 : Test Cases for User Management by Admin .....                                | 104 |
| Table 36 : Test Cases for Audit Logs and Activity Tracking.....                         | 104 |
| Table 37 : Test Cases for Accepting Workspace Invitations.....                          | 106 |
| Table 38 : Test Cases for Permission and Authorization Control.....                     | 107 |
| Table 39 : Test Cases for Role & Permission Enforcement (Advanced Scenarios) .....      | 107 |
| Table 40 : Test Cases for Member Suspension Management.....                             | 108 |
| Table 41 : Test Cases for Data Processing (ETL Pipeline).....                           | 110 |
| Table 42 : Test Cases for ETL Pipeline Execution (Detailed Stages) .....                | 111 |
| Table 43 : Test Cases for ClickHouse Load & Data Integrity .....                        | 112 |
| Table 44 : Test Cases for Metadata Management & Data Lineage .....                      | 113 |
| Table 45 : Test Cases for System Performance & Scalability .....                        | 114 |

|  |     |
|--|-----|
| Table 46 : Test Cases for Voice-to-SQL Processing & Intent Detection ..... | 115 |
| Table 47 : Test Cases for Chart Generation & Visualization.....            | 116 |
| Table 48 : Test Cases for Dashboard Creation & Management .....            | 117 |
| Table 49 : Test Cases for Account Deletion & Data Retention Policy .....   | 120 |

## List of Figures

|   |     |
|---|-----|
| Figure 1 : Gantt Charts.....  | 36  |
| Figure 2 : Sequence Diagram for "Sign Up" .....                     | 57  |
| Figure 3 : Sequence Diagram for "Email Verification".....           | 59  |
| Figure 4 : Sequence Diagram for "Log in" .....                      | 61  |
| Figure 5 : Sequence Diagram for "Logout" .....                      | 63  |
| Figure 6 : Sequence Diagram for "Manage Profile" .....              | 65  |
| Figure 7 : Sequence Diagram for "Deactivate My Profile" .....       | 67  |
| Figure 8 : Sequence Diagram for " Edit Workspace Info" .....        | 69  |
| Figure 9 : Sequence Diagram for " View Workspace members list"..... | 70  |
| Figure 10 : Sequence Diagram for " invite members" .....            | 72  |
| Figure 11 : Sequence Diagram for " Assign Roles" .....              | 74  |
| Figure 12 : Sequence Diagram for "Remove Member" .....              | 76  |
| Figure 13 : Sequence Diagram for "Suspend Member" .....             | 78  |
| Figure 14 : Sequence Diagram for " Accept invitation" .....         | 81  |
| Figure 15 : Sequence Diagram for "Record voice" .....               | 83  |
| Figure 16 : Sequence Diagram for " Upload Voice" .....              | 85  |
| Figure 17 : Sequence Diagram for "View Dashboard Reports".....      | 86  |
| Figure 18 : Sequence Diagram for "View Previous Reports" .....      | 87  |
| Figure 19 : Sequence Diagram for "View Reports Details" .....       | 88  |
| Figure 20 : Sequence Diagram for "Edit SQL" .....                   | 89  |
| Figure 21: Sequence Diagram for " Edit Transcription" .....         | 90  |
| Figure 22 : Sequence Diagram for " Export Report .....              | 91  |
| Figure 23: Sequence Diagram for " Delete Report" .....              | 92  |
| Figure 24 : Sequence Diagram for " Upload datanase" .....           | 94  |
| Figure 25 : Sequence Diagram for "Delete Database". ....            | 96  |
| Figure 26 : Class Diagram.....                                      | 97  |
| Figure 27 : ERD Diagram.....  | 98  |
| Figure 28 : System Architecture .....                               | 122 |
| Figure 29 : ETL Pipeline.....                                       | 125 |
| Figure 30 : Class diagram for user auth .....                       | 127 |
| Figure 31 : class diagram for User managements .....                | 130 |
| Figure 32 : class diagram for Workspace managements .....           | 132 |
| Figure 33 : class diagram for Database Managements .....            | 134 |
| Figure 34 : Class Diagram for AI Services.....                      | 137 |
| Figure 35 : class diagram for Voice Managements.....                | 140 |
| Figure 36 : Home Page Interface .....                               | 149 |
| Figure 37 : Login interface .....                                   | 149 |
| Figure 38 : sign up interface.....                                  | 150 |
| Figure 39 : Email verification in gmail.....                        | 150 |
| Figure 40 : Dashboard Manager .....                                 | 151 |
| Figure 41 : Workspace Settings .....                                | 151 |

|   |     |
|---|-----|
| Figure 42 : Profile settings .....            | 152 |
| Figure 43 : Detective account interface ..... | 152 |
| Figure 44 : invitation member interface ..... | 153 |
| Figure 45 : join workspace in gmail .....     | 153 |
| Figure 46 : dashboard Analysis .....          | 154 |
| Figure 47 : Member Settings .....             | 154 |
| Figure 48 : Suspend Member interface .....    | 155 |
| Figure 49 : change member role .....          | 155 |
| Figure 50 : Dashboard Executive .....         | 156 |
| Figure 51 : Upload database .....             | 156 |
| Figure 52 : Upload Voice.....                 | 157 |
| Figure 53 : Edit SQL .....                    | 157 |

# **Chapter 1 Introduction**

## ***1. Introduction***

This chapter provides a general introduction to the BI Voice Agent project and establishes the foundation for the rest of the report. It begins by presenting the overall context and motivation behind the project, highlighting the challenges associated with traditional Business Intelligence systems. The chapter then defines the problem addressed by the project and clearly states the main objectives to be achieved.

In addition, this chapter introduces the proposed system at a high level, explaining its main components and general workflow without going into technical details. Finally, the chapter outlines the structure of the report and describes the content of each subsequent chapter to guide the reader through the document.

## ***2. Problem Statement***

Despite the availability of powerful BI platforms, many organizations face difficulties in enabling users to effectively analyze data. The main challenges include the complexity of writing SQL queries, the need for technical expertise to interpret dashboards, and the lack of natural interaction with analytical systems. These limitations slow down decision-making processes and restrict access to data insights to a limited group of technical users.

Additionally, traditional BI systems do not provide seamless integration between voice-based interaction and real-time analytical querying. There is also a lack of intelligent pipelines that can automatically validate, process, and monitor analytical data while maintaining scalability and reliability.

Therefore, there is a need for a system that enables users to query analytical databases using voice commands, automatically interpret user intent, generate correct SQL queries, and present results in an understandable and visual form.

### **3. Project Objective**

The main objective of the **BI Voice Agent** project is to design and implement an intelligent voice-driven Business Intelligence system that simplifies data analysis and enhances accessibility for users of all technical levels.

The specific objectives of the project include:

- Enabling voice-based interaction with analytical databases.
- Automatically converting spoken questions into structured SQL queries.
- Classifying user questions to determine whether they are analytical or non-analytical.
- Integrating a scalable ETL pipeline for data ingestion, transformation, and loading.
- Providing real-time analytical results through visual dashboards and charts.
- Ensuring system scalability, reliability, and modularity.

### **4. Proposed System**

The proposed system is a multi-layered, service-oriented architecture that integrates voice processing, intelligent query generation, and data analytics. The system begins by capturing the user's voice input, which is converted into text using a speech recognition module. The text is then processed to extract intent and determine the type of query.

For analytical queries, the system generates SQL statements compatible with the analytical data warehouse. The data is processed through an ETL pipeline that ensures data consistency and quality. Finally, the query results are visualized using interactive charts and dashboards.

The architecture is designed to be modular, allowing each component to operate independently while maintaining seamless communication between services

## ***5. Report Organization***

The report is structured as follows:

- Chapter 1: Introduction
- Chapter 2: Basic Concepts and Reference Study
- Chapter 3: Project Management
- Chapter 4: System Analysis
- Chapter 5: System Design
- Chapter 6: Practical Implementation
- Chapter 7: Report Overview

## ***6. Summary***

This chapter presented an overview of the BI Voice Agent project, including the motivation behind the system, the problem it addresses, the project objectives, and the proposed solution. It also outlined the structure of the report and provided a foundation for the technical and analytical discussions in the following chapters.

## **Chapter 2: Basic Concepts and Reference Study**

# I. Reference Study of Interactive AI Powered Business Intelligence Tools

## *1. Introduction*

This chapter provides a foundation for understanding the **BI Voice Agent** system by exploring the fundamental concepts and reviewing related studies in the field of intelligent business analytics. It examines essential terminology, key principles, and core technologies such as Business Intelligence systems, voice-based interaction, natural language processing, and data analytics pipelines. The chapter aims to present an overview of AI-driven and natural language-based business intelligence platforms, along with insights from existing solutions and research studies. These references help identify current capabilities and limitations in the field and provide guidance for the design and development of the proposed BI Voice Agent system.

## *2. Fundamental Concepts*

This section introduces the core concepts required to understand the design and functionality of the BI Voice Agent system.

- **Business Intelligence (BI):**

Business Intelligence refers to technologies and processes used to collect, integrate, analyze, and visualize data to support decision-making. Traditional BI systems rely heavily on dashboards, predefined reports, and structured queries such as SQL.

- **Natural Language Querying:**

Natural Language Querying allows users to interact with data systems using human language instead of formal query syntax. This approach improves accessibility, especially for non-technical users, but introduces challenges such as ambiguity, context understanding, and intent recognition.

- **Voice-Based Interaction:**

Voice-based interfaces extend natural language querying by enabling spoken input. Such systems require accurate speech-to-text processing and must handle variations in pronunciation, accents, and query structure.

- **Natural Language to SQL (NL2SQL):**  
NL2SQL systems automatically translate user questions into structured SQL queries. These systems must correctly map user intent to database schemas, tables, and columns while ensuring query correctness and security.
- **ETL and Data Pipelines:**  
ETL (Extract, Transform, Load) pipelines are responsible for ingesting data from multiple sources, cleaning and transforming it, and loading it into analytical data stores. Scalable and monitored ETL pipelines are essential for real-time and enterprise-level analytics.
- **Analytical Data Warehouses:**  
Analytical databases are optimized for read-heavy workloads and large-scale aggregations. They support fast execution of complex analytical queries and are commonly used in modern BI systems.

Together, these concepts form the technical foundation upon which voice-driven and AI-powered business analytics systems are built.

- **Large Language Models (LLMs)**  
LLMs are advanced AI models trained on large-scale textual data to understand and generate human-like language. In BI Voice Agent, LLMs play a key role in interpreting user intent, converting natural language queries into structured representations, and assisting in SQL generation.
- **Apache Kafka (Data Streaming)**  
Kafka is a distributed event-streaming platform used for building scalable and fault-tolerant data pipelines. It enables asynchronous communication between ETL components, ensuring reliable data flow and system decoupling.
- **ClickHouse (Analytical Data Warehouse)**  
Is a high-performance column-oriented analytical database optimized for large-scale read-heavy queries. It supports fast aggregation and real-time analytics, making it suitable for BI applications.
- **Metabase (Data Visualization)**  
Is an open-source business intelligence tool used for visualizing analytical query results. It allows users to explore data through dashboards and charts without requiring SQL expertise.
- **LangChain / LangGraph (AI Orchestration)**

Are frameworks designed to manage and orchestrate interactions between large language models and external tools. They enable structured reasoning flows, multi-step decision-making, and modular AI pipelines.

### **3. Literature Review for the system**

#### 1-ThoughtSpot – Interactive AI-Powered Business Analytics Platform

##### *1.1 Platform Description*

**ThoughtSpot** is an AI-powered business analytics platform that allows users to query data through a simple, search-like interface. Employees can ask questions in natural language and instantly get visual or tabular answers without SQL skills.

It features **SpotIQ** for automatic pattern and anomaly detection, and **SearchIQ** for natural language and voice queries that reveal hidden insights in large datasets.

Powered by the in-memory **Falcon** engine, ThoughtSpot delivers fast, real-time analytics at enterprise scale, with **API** and **SDK** integrations that embed its capabilities into other business applications and dashboards.

##### *1.2 System Actors*

The platform serves multiple roles within an organization:

- **Non-technical users** (e.g., sales, marketing, and business managers) can access analytics easily through an intuitive search-based interface.
- **Data analysts and data engineers** prepare and model the data, design *Worksheets*, and define relationships between tables to ensure accurate results.
- **System administrators (IT Managers)** integrate ThoughtSpot with data sources and manage security and permissions, with support for identity systems like Okta and Azure AD.
- **Developers** use ThoughtSpot APIs to embed its analytical capabilities into other applications or to build custom interactive dashboards.

##### *1.3 Core Functional Requirements*

ThoughtSpot relies on several core functional requirements to ensure high performance and usability within enterprise environments:

1. **Data Integration:** Connects to various data sources such as cloud warehouses, databases, and Excel files.
2. **Data Modeling:** The technical team prepares structured data models (e.g., Star Schema) for efficient querying.
3. **Real-Time Queries:** Runs live queries on connected data without permanent storage, using in-memory caching for speed.
4. **High Performance:** Requires a fast data warehouse or cloud setup to utilize the Falcon engine effectively.
5. **Multi-Device Access:** Accessible via web and mobile interfaces for flexible, on-the-go analytics.
6. **Natural Interaction:** *SearchIQ* enables voice and natural language queries for intuitive exploration.
7. **APIs and SDKs:** Allows embedding analytics and extracting results programmatically into other systems.
8. **Deployment Flexibility:** Offered as a managed cloud service or on-premises installation (AWS, GCP, Azure).
9. **Security & Compliance:** Uses encrypted, VPN-secured connections and adheres to SOC 2, ISO 27001, and GDPR standards.

## 2-Databricks AI/BI Genie – Natural Language Business Intelligence Platform

### 2.1 Platform Description

*Databricks AI/BI Genie* is a feature within the Databricks platform that allows users to ask questions about their data in natural language within the **Lakehouse** environment. Genie interprets business questions, converts them automatically into SQL queries, and displays the results as interactive tables or charts. It is powered by **Large Language Models (LLMs)** customized to each organization's data and terminology. As a **compound AI system**, Genie can ask clarifying questions and generate accurate analyses, acting as a virtual business intelligence analyst that delivers instant insights.

### 2.2 System Actors

- **Non-technical business users:** Ask questions and receive instant insights without writing SQL queries.
- **Data analysts and engineers:** Configure the *Genie Space* and provide contextual definitions and examples to train the model.

- **System administrators:** Integrate Genie with data sources via **Unity Catalog**, manage security, and ensure sufficient computing resources.
- **Developers:** Embed Genie's capabilities into external applications using APIs.
- 

### *2.3 Core Functional Requirements*

1. **Structured data organization:** Data must be registered in Unity Catalog with clear descriptions and defined relationships.
2. **Rich metadata and context:** Add organizational terms, KPIs, and definitions within *Genie Space* to improve understanding.
3. **SQL Warehouse connection:** Execute read-only SQL queries without modifying source data.
4. **Adequate computing resources:** Ensure fast query execution and scalability for multiple users.
5. **Integration with LLMs:** Genie operates using managed models like Dolly or Azure OpenAI.
6. **Security and governance:** Enforces Unity Catalog policies for access control and data privacy.
7. **Performance monitoring:** Tracks user queries and feedback to continuously improve accuracy and reliability.

## 3-Microsoft Power BI – Interactive Business Intelligence Platform by Microsoft

### *3.1 Platform Description*

*Microsoft Power BI* is a leading cloud-based business intelligence platform that helps users collect, transform, and visualize data through interactive dashboards and reports. It includes **Power BI Desktop** for report creation and **Power BI Service** for online sharing.

With **Q&A** and **Copilot**, users can analyze data and generate insights using natural language, making analytics accessible even to non-technical users.

### *3.2 System Actors*

- **Admin:** Manages the workspace, security, and data connections.
- **Creator:** Builds models and reports using Power Query and DAX.

- **Contributor:** Edits and updates shared reports.
- **Viewer:** Consumes reports and dashboards for decision-making.
- **Stakeholders:** Define data requirements and use insights strategically.
- 

### *3.3 Core Functional Requirements*

1. **Data Connectivity:** Links to on-premises and cloud sources (SQL, Azure, Excel, etc.).
2. **Flexible Query Modes:** Supports Import and DirectQuery for real-time or cached data.
3. **Data Modeling:** Cleans and structures data via Power Query using *Star Schema*.
4. **In-memory Engine:** Uses VertiPaq for high-speed data compression and analytics.
5. **Secure Gateway:** Syncs on-premises data with cloud services safely.
6. **Licensing & Capacity:** Requires Pro or Premium licensing for enterprise performance.
7. **Integration:** Embeds reports via APIs and connects with Azure, Office 365, and Teams.
8. **Security:** Applies encryption and Row-Level Security (RLS) for controlled access.
9. **Performance:** Scales efficiently through optimized models and Premium capacity.

## 4-Vanna.AI – Open-Source Interactive Data Intelligence Framework

### *4.1 Platform Description*

*Vanna.AI* is an open-source framework for building AI agents that analyze data through natural language. It allows users to ask questions directly to a database without writing SQL, translating queries automatically using **Large Language Models (LLMs)**.

*Vanna* employs the **Retrieval-Augmented Generation (RAG)** approach, enhancing accuracy by providing contextual knowledge such as database schema, table descriptions, and relationships. It returns interactive results as tables or visualizations using libraries like *Plotly*, enabling a conversational and intuitive data analysis experience.

## 4.2 System Actors

- **End User:** Non-technical employee or analyst who interacts with the system through a chat interface to retrieve insights.
- **Developer/Data Engineer:** Integrates Vanna with databases and configures connections to AI models and data sources.
- **Data Expert:** Trains the model on database structure, adds examples, and improves accuracy through feedback.
- **System/Security Administrator:** Manages API keys, access permissions, and ensures queries are executed securely in read-only mode.

## 4.3 Core Functional Requirements

1. **Structured Database:** Supports relational and modern databases such as PostgreSQL, MySQL, SQLite, Snowflake, and BigQuery.
2. **Reliable Connectivity:** Requires accessible JDBC/SQL connections between Vanna and the target database.
3. **Well-Documented Schema:** Tables and fields should be clearly defined to enhance training and query accuracy.
4. **Python Environment:** Install the Vanna package and link it to both the database and the chosen language model.
5. **LLM Integration:** Compatible with GPT-4, Claude, or locally hosted open-source models.
6. **Vector Database (optional):** Used for RAG context retrieval via Pinecone, Chroma, or FAISS.
7. **Access Control & Security:** Operates with read-only permissions and user identity tokens for authorization checks.
8. **Flexible Interface:** Deployable as a web app (Flask/Streamlit) or integrated with tools like Slack or Teams.
9. **Cloud Option:** *Vanna Cloud* provides monitoring, query logging, and secure connectivity to enterprise data sources.

## 5. Sequel.sh – AI-Native Business Intelligence Platform

### *1-Platform Description*

*Sequel.sh* is a modern **AI-native BI platform** built from the ground up for natural interaction with data. It allows users to ask questions in plain English and instantly receive visual, data-driven answers—without needing SQL knowledge.

The platform translates questions into optimized SQL queries executed directly on live databases, ensuring real-time insights. Sequel automatically understands database schemas, enabling accurate interpretation without manual setup.

Its chat-style interface lets users explore, save, and share insights within collaborative **Workspaces**, with smart features like **automated insights** and interactive trend analysis.

### 5.1 System Actors

- **Business Users:** Product managers, marketers, and analysts who ask questions and interact with visual results through an intuitive chat interface.
- **Data Analysts and Experts:** Use the built-in SQL editor to review or fine-tune AI-generated queries for higher accuracy.
- **Data Engineers and Administrators:** Connect Sequel to company databases securely, define user roles, and manage permissions.
- **Security Teams:** Monitor access, encryption, and activity logs to ensure compliance and data privacy.

### 5.2 Core Functional Requirements

1. **Database Connectivity:** Supports PostgreSQL, MySQL, and SQLite for direct querying without data migration.
2. **Secure Connection:** Requires SSL-encrypted communication and read-only database credentials.
3. **Optimized Performance:** Databases should be indexed and tuned for instant, real-time query responses.

4. **Cloud Setup (SaaS):** Operates fully in the cloud with simple initial configuration and workspace creation.
5. **Workspace Collaboration:** Enables team environments with role-based access (Viewer / Member).
6. **Single Sign-On (SSO):** Supports OAuth or SAML for enterprise identity integration.
7. **Hybrid Deployment Option:** Can use a local agent within the client's cloud to keep data within secure boundaries.
8. **AI Model Integration:** Uses built-in language models to convert natural language to SQL automatically.
9. **Usage & Cost Management:** Tracks queries and data volume to manage subscription limits effectively

## II. Comparative Analysis of AI-Powered Interactive BI Platforms

| Criterion               | ThoughtSpot                                   | Databricks AI/BI Genie                          | Microsoft Power BI                      | Vanna.AI                             | Sequel.sh                               | BI Voice Agent (Our Project)                                     |
|-------------------------|---|---|---|--------------------------------------|---|--|
| <b>Platform Type</b>    | AI-driven search-based BI platform            | AI assistant inside Databricks Lakehouse        | Traditional BI with AI enhancements     | Open-source AI analytics framework   | AI-native BI for natural querying       | Voice-controlled BI assistant for speech-to-dashboard automation |
| <b>Main Purpose</b>     | Natural language search and instant analytics | Natural language analytics in Databricks        | Create and share dashboards and reports | Query databases via natural language | Provide instant AI-powered SQL insights | Convert voice input → SQL → dashboards automatically             |
| <b>Core Mechanism</b>   | NLP → SQL → Falcon engine                     | LLMs generate SQL with contextual understanding | Copilot Q&A + BI visualization          | LLM + RAG → SQL                      | English → optimized SQL on live data    | Whisper STT → LLM Text-to-SQL → ClickHouse → Metabase dashboards |
| <b>AI Capabilities</b>  | SpotIQ, SearchIQ                              | Context-aware LLMs (Compound AI)                | Copilot + Q&A                           | LLMs with RAG                        | Built-in LLM SQL translation            | Speech recognition + LLM SQL generation + auto visualization     |
| <b>Primary Users</b>    | Managers & non-technical users                | Analysts in Databricks                          | Analysts & employees                    | Developers & analysts                | Business users & engineers              | Managers, analysts, and business users who prefer voice commands |
| <b>Additional Roles</b> | Admins, devs, engineers                       | Domain experts, governors                       | Admin, Creator, Viewer                  | Developers, admins                   | Data engineers, security officers       | Manager, Data Analyst, System Admin (defined in our system)      |
| <b>Deployment Model</b> | Cloud / On-Prem                               | Databricks cloud                                | Cloud or Desktop                        | Local / Cloud                        | SaaS cloud or local agent               | Local deployment using Docker + Kafka + ClickHouse + Flask       |

|                                  |                         |                           |                          |                                      |                        |  |
|----------------------------------|-------------------------|---------------------------|--------------------------|--------------------------------------|------------------------|--|
| <b>Data Preparation</b>          | Worksheets, Star Schema | Genie Space configuration | Power Query, Star Schema | Metadata training, schema enrichment | Auto schema detection  | ETL pipeline (CSV → JSON → Kafka → ClickHouse) with cleaning & structuring |
| <b>Security &amp; Governance</b> | SSO, Encryption         | Unity Catalog             | Encryption, RLS          | Token identity checks                | SSL, activity logs     | Role-based access + workspace isolation + secure DB access                 |
| <b>Integration Capabilities</b>  | APIs & SDKs             | APIs for Teams            | Power BI Embedded        | Slack, Teams, Jupyter                | Export & integrations  | Kafka streaming, ClickHouse, Metabase, Flask APIs                          |
| <b>User Interface</b>            | Search bar + dashboards | Conversational UI         | Interactive dashboards   | Chat/web UI                          | Chat-like workspace    | Voice-based interface + dashboard viewer in Metabase                       |
| <b>Key Features</b>              | Instant insights        | Multi-agent reasoning     | Microsoft ecosystem      | Model-agnostic                       | Real-time collab       | Hands-free BI reporting, full automation, real-time voice-driven analysis  |
| <b>Strengths</b>                 | Fast & intuitive        | Enterprise precision      | Stable & scalable        | Flexible                             | Collaborative          | Accessibility, simplicity, minimal user effort, full automation            |
| <b>AI Operation Style</b>        | NLP + pattern mining    | Compound AI               | Copilot automation       | RAG agent                            | Real-time LLM          | Speech → NLP → SQL → Visualization pipeline                                |
| <b>Target Audience</b>           | Large enterprises       | Tech firms                | Enterprises & mid-size   | Developers                           | Startups               | Businesses wanting fast voice-driven insights without technical skills     |
| <b>Cloud Integration</b>         | AWS/Azure/GCP           | Databricks Cloud          | Azure                    | Any cloud/local                      | Sequel Cloud           | Docker-based local system (future: cloud deployment)                       |
| <b>Output Format</b>             | Dashboards              | SQL + visuals             | Reports, dashboards      | Tables, charts                       | Interactive dashboards | Auto-generated dashboards + downloadable reports                           |

*Table 1 : Comparative Analysis of Reference Studies*

## **4. Literature Review for Voice-Based Business Intelligence Systems**

### ***1. Abstract***

Voice-based Business Intelligence systems combine natural language processing, large language models, and analytical data platforms to enable intuitive access to data insights without requiring technical expertise. Recent studies and commercial platforms demonstrate the effectiveness of natural language and voice-driven analytics in reducing query complexity, improving decision-making speed, and increasing data accessibility across organizational roles.

Existing solutions leverage a variety of AI techniques, including rule-based natural language interfaces, neural language models, and Large Language Models (LLMs), with performance evaluated through query accuracy, execution correctness, response latency, and user satisfaction. Analytical platforms report high effectiveness in structured environments; however, challenges remain in scalability, real-time data processing, intent disambiguation, and tight integration with ETL pipelines.

Reported systems typically achieve high query interpretation accuracy in controlled environments, but often depend on well-prepared schemas, extensive metadata, and pre-modeled datasets. These findings highlight the need for modular, scalable, and voice-first BI systems capable of handling real-time analytics and heterogeneous data sources—motivating the development of the proposed **BI Voice Agent**.

### ***2. Review of Existing Systems***

Several AI-powered business intelligence platforms have been proposed to support natural language and voice-based data exploration. This review analyzes five representative systems based on their architecture, system actors, and functional requirements.

#### **2.1 ThoughtSpot – Interactive AI-Powered Business Analytics**

ThoughtSpot enables search-based and voice-driven querying over enterprise datasets using natural language. It employs in-memory query acceleration and automated insight discovery through SpotIQ. While highly performant, the system relies on structured data models prepared by technical teams and operates primarily in enterprise-managed environments.

## **2.2 Databricks AI/BI Genie – Natural Language Analytics in Lakehouse**

Databricks Genie integrates natural language querying directly within the Lakehouse architecture. Powered by LLMs, it converts business questions into SQL queries and executes them on governed data using Unity Catalog. The system offers strong governance and scalability but depends heavily on cloud infrastructure and pre-configured metadata.

## **2.3 Microsoft Power BI – AI-Enhanced Business Intelligence**

Power BI provides natural language querying through Q&A and Copilot features, allowing users to interact with dashboards and reports using conversational language. Although widely adopted, its natural language capabilities are limited to predefined models and dashboards, and voice interaction is not fully native.

## **2.4 Vanna.AI – Open-Source NL2SQL Framework**

Vanna.AI is an open-source framework that converts natural language questions into SQL queries using LLMs and Retrieval-Augmented Generation (RAG). It provides flexibility and transparency but requires careful schema documentation and manual model training to achieve reliable performance.

## **2.5 Sequel.sh – AI-Native BI Platform**

Sequel.sh offers a chat-based interface that translates natural language queries into optimized SQL executed on live databases. It emphasizes simplicity and real-time analytics but operates mainly as a SaaS solution with limited customization for complex ETL workflows.

### **3. Thematic Analysis: Evolution of AI Techniques in Voice-Based BI**

This thematic analysis examines the dominant AI approaches used in voice-based and natural language business intelligence systems, focusing on their strengths, challenges, and reported effectiveness.

| Category                                  | Key Features   | Challenges   | Typical Usage  |
|---|--|--|--|
| Rule-Based and Search-Oriented Interfaces | Deterministic query mapping, predictable outputs     | Limited flexibility, poor handling of ambiguity                  | Early BI systems and basic natural language interfaces           |
| Traditional NL2SQL Systems                | Structured intent parsing, schema mapping            | Strong schema dependency, limited generalization                 | High-accuracy querying in constrained domains                    |
| LLM-Based BI Systems                      | Contextual understanding, flexible language handling | Hallucination risk, high computational cost, governance concerns | Advanced natural language analytics and improved user experience |
| Streaming and ETL-Integrated Analytics    | Real-time data availability, scalable pipelines      | Monitoring complexity, fault tolerance, metadata consistency     | Enterprise-grade BI architectures                                |
| Visualization-Centric BI Platforms        | Interactive dashboards, user-friendly interfaces     | Dependency on predefined models and queries                      | Decision support and executive analytics                         |

*Table 2 : Thematic Analysis of AI Techniques in Voice-Based Business Intelligence*

### **4. Summary of Findings**

The analysis reveals that:

- Most existing BI platforms rely on **pre-modeled data and structured schemas**
- LLM-based systems significantly improve **query flexibility and usability**
- Limited solutions provide **end-to-end voice interaction integrated with ETL**
- Real-time analytics and system observability remain key challenges
- Few systems offer **fully modular, service-oriented architectures**

These limitations highlight the need for a voice-first, modular, and scalable BI system that tightly integrates speech processing, intent analysis, SQL

generation, ETL monitoring, and visualization—addressed by the proposed **BI Voice Agent**.

### ***5. Positioning of the Proposed System***

Based on the reviewed literature and systems, **BI Voice Agent** distinguishes itself by:

- Providing **native voice-based interaction**
- Integrating **LLMs with structured reasoning (LangChain/LangGraph)**
- Supporting **real-time ETL pipelines (Kafka-based)**
- Utilizing a **high-performance analytical warehouse (ClickHouse)**
- Delivering insights through **dynamic visualization (Metabase)**

This positioning demonstrates how the proposed system addresses existing gaps while building upon proven concepts in AI-driven business intelligence.

# **Chapter 3 Project Management**

## **1. Introduction**

This chapter focuses on the project management aspects of the BI Voice Agent project, which play a crucial role in ensuring successful planning, execution, and delivery. It presents the key management documents and practices used to guide the project from initiation to completion.

The chapter covers the project charter, Statement of Work (SOW), roles and responsibilities, project resources, schedule, and risk management strategies. These elements provide a structured framework to control project scope, timeline, and quality while addressing potential risks that may arise during development.

## **2. Project Charter**

A project charter is a formal document that authorizes the initiation of a project and defines its high-level objectives, scope, and governance structure. It serves as a reference throughout the project lifecycle and supports informed decision-making.

Project Title: BI Voice Agent – Voice-Driven Business Intelligence System

Project Start Date: Oct 18, 2025

Projected Finish Date: Jan 1, 2026

Project Supervisors / Managers:

- Eng. Anas Abdulaziz

Project Objectives

- Design and develop a voice-based Business Intelligence system that allows users to query analytical data using natural language.
- Enable automatic conversion of voice queries into structured SQL queries.
- Integrate a scalable ETL pipeline for real-time data ingestion and analytics.
- Provide interactive visualizations and dashboards for data exploration.
- Improve accessibility of data analytics for non-technical users.

Approach

- Gather functional and non-functional requirements through analysis of BI use cases.
- Design a modular system architecture integrating voice processing, AI reasoning, and data analytics.
- Develop backend services, ETL components, and AI modules iteratively.
- Test system components individually and through end-to-end integration.
- Validate analytical results and visualization outputs.

## **Roles and Responsibilities**

| Name                | Role               | Responsibilities   |
|---------------------|--------------------|--|
| Eng. Anas Abdulaziz | Project Supervisor | Provide academic supervision, guide the project direction, and evaluate progress and deliverables. Oversee technical |

|                 |                              |   |
|-----------------|------------------------------|---|
|                 |                              | decisions, review system architecture, and support implementation challenges.   |
| Massa nasri     | AI & Data Sciences           | Prepare and preprocess datasets, support ETL validation, design the AI pipeline, integrate LLMs, implement ETL services, and develop SQL generation mechanisms. |
| Ayman Alkotyfan | Backend & Frontend Developer | Design frontend components, dashboards, and visualization interfaces, and implement backend logic and system integrations.                                      |

*Table 3 : Roles and Responsibilities of Project Team Members*

### **3. Statement of Work (SOW)**

The Statement of Work defines the scope, objectives, deliverables, and responsibilities of the BI Voice Agent project. It establishes a clear understanding of what will be developed and the criteria for successful completion.

#### **1. Project Description and Objectives**

The project aims to develop a voice-driven Business Intelligence system that enables users to interact with analytical databases using spoken language. The system supports intelligent query interpretation, real-time analytics, and visual reporting.

#### **2. Project Scope**

The BI Voice Agent focuses on:

- Voice-to-text processing.
- Natural language understanding and intent classification.
- Automatic SQL generation for analytical queries.
- ETL pipeline integration using streaming technologies.
- Data visualization through interactive dashboards.

#### **3. Project Goals**

- Simplify data analytics for non-technical users.
- Provide accurate and real-time analytical insights.
- Ensure system scalability and modularity.
- Enable seamless integration between AI and data engineering components.

## Deliverables

- Project plan and timeline.
- Software Requirements Specification (SRS).
- System architecture and design documentation.
- Functional BI Voice Agent system.
- Final project report.

## **4. Project Requirements**

### Technology and Tools

- Programming Languages: Python, JavaScript, SQL
- Backend Framework: Django / FastAPI
- Frontend: React
- Data Streaming: Apache Kafka
- Analytical Database: ClickHouse
- Visualization: Metabase
- AI Frameworks: LangChain, LangGraph
- LLMs: OpenAI-compatible or local large language models

## **5. Assumptions**

- Continuous availability of supervisors for feedback and evaluation.
- Stable development environment and infrastructure.
- Access to required software tools and datasets.
- Incremental feedback during development milestones.

## **6. Project Resources**

### **Human Resources**

- Academic supervisors
- AI and data engineering developers
- Frontend and integration support

### **Technical Resources**

- Local and cloud-based development environments
- Version control and collaboration tools
- Analytical databases and streaming platforms

## **7. Schedule**

- Project Start Date: Oct 18, 2025
- Project End Date: Jan 1, 2026

## **8. Risk Management**

| Risk Title                 | Risk Description                                  | Tracking Frequency | Impact | Mitigation Plan                                      |
|----------------------------|---|--------------------|--------|--|
| Limited team size          | Project progress depends on a small team          | Weekly             | High   | Cross-task collaboration and shared responsibilities |
| Complex system integration | Difficulty integrating AI, ETL, and BI components | Weekly             | High   | Incremental integration and early testing            |
| LLM output inconsistency   | AI-generated SQL may be incorrect or ambiguous    | Weekly             | High   | Validation layers and schema-aware prompting         |
| Data pipeline failures     | Streaming or ETL issues may affect analytics      | Weekly             | Medium | Monitoring, logging, and retry mechanisms            |
| Time constraints           | Tight academic deadlines                          | Weekly             | Medium | Task prioritization and milestone-based planning     |
| Visualization mismatch     | Results may not align with user expectations      | Weekly             | Medium | Iterative UI testing and feedback cycles             |

Table 4 : Risk Management Matrix

## 9. Gantt charts

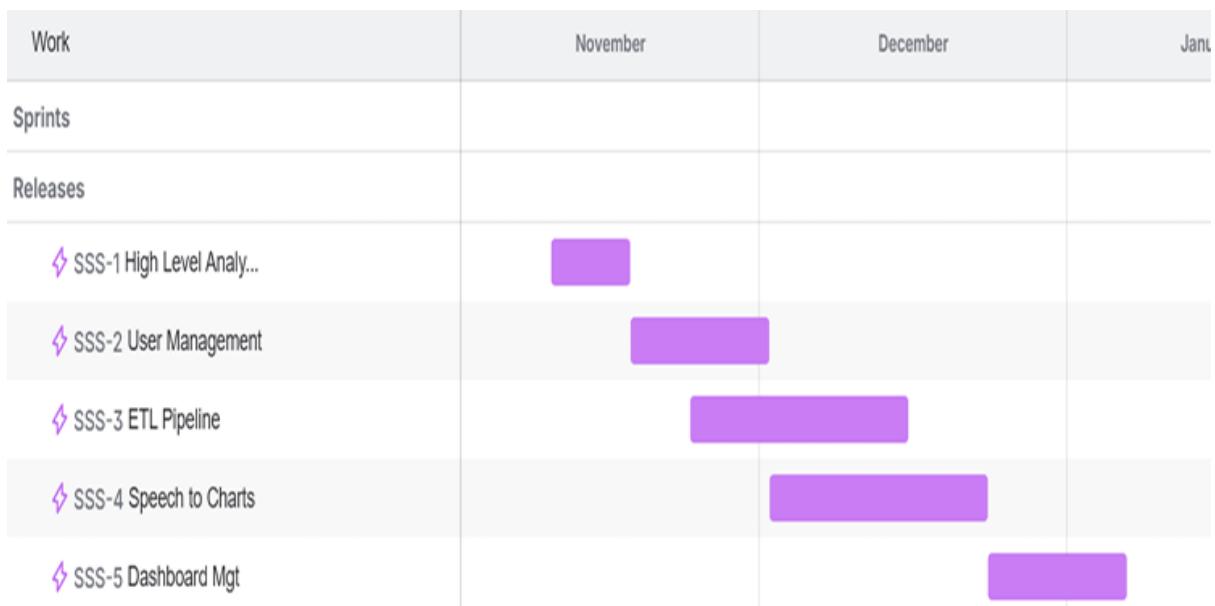


Figure 1 : Gantt Charts

## 10. Summary

This chapter presented the project management framework of the BI Voice Agent project, including the project charter, scope definition, roles, resources, schedule, and risk management plan. These management practices ensure structured development, controlled execution, and successful delivery of the project.

## **Chapter 4: System Analysis**

## **1. Introduction**

This chapter focuses on the detailed analysis of the **BI Voice Agent** system. It aims to define the system requirements, functionality, and operational context by analyzing user needs and system constraints. The chapter provides a clear understanding of how the proposed system will operate and how it addresses the challenges of traditional Business Intelligence systems.

The analysis includes an overview of the project timeline, sprint-based development approach, and the Software Requirements Specification (SRS). Functional and non-functional requirements are identified to ensure that the system meets performance, scalability, and usability objectives. This chapter serves as a foundation for the system design and implementation phases presented in the subsequent chapters.

## **2. Project Timeline**

The development of the **BI Voice Agent** project follows an iterative and incremental approach based on Agile methodology. The project is divided into five main sprints, each focusing on a specific subsystem or milestone. This approach allows continuous integration, early validation, and effective risk management throughout the development lifecycle.

| Sprint                  | Duration | Time Period     | Main Focus                                  |
|-------------------------|----------|-----------------|---|
| <b>Sprint 1 (SSS-1)</b> | 1 weeks  | 10 Nov – 17 Nov | High-Level Analysis and System Planning     |
| <b>Sprint 2 (SSS-2)</b> | 2 weeks  | 18 Nov – 2 Dec  | User Management and Access Control          |
| <b>Sprint 3 (SSS-3)</b> | 3 weeks  | 23 Nov – 15 Dec | ETL Pipeline and Data Ingestion             |
| <b>Sprint 4 (SSS-4)</b> | 3 weeks  | 3 Dec – 25 Dec  | Speech-to-Query and Chart Generation        |
| <b>Sprint 5 (SSS-5)</b> | 2 week   | 25 Dec – 10 Jan | Dashboard Management and System Integration |

*Table 5 : Project Timeline and Sprint Planning*

### **3. Software Requirement specification document ( SRS )**

#### **1. Introduction**

##### **1.1 Purpose**

The purpose of this document is to specify the software requirements for the BI Voice Agent platform. This document provides a detailed description of the system's functional requirements, system actors, and interaction scenarios, with a particular focus on Sprint 1, which covers the User Management domain. This SRS serves as a foundational reference for modeling system use cases, defining system behavior, and guiding the design and implementation phases of the BI Voice Agent platform.

##### **1.2 Project Scope**

The BI Voice Agent platform is designed to enable voice-driven interaction with Business Intelligence systems. It allows users to manage workspaces, authenticate securely, and collaborate within a shared analytical environment before accessing advanced BI capabilities such as voice-based querying and dashboard analytics.

This specification focuses on high-level requirements related to user authentication, workspace management, role-based access control, and collaboration features, which form the backbone of the platform.

###### **1.2.1 High-Level Requirements**

###### **• Authentication**

This requirement provides secure access to the BI Voice Agent platform. The system allows users to authenticate using valid credentials and ensures that only authorized users can access system functionalities. Authentication is mandatory for all user roles and serves as the entry point for interacting with dashboards, reports, and analytical features.

- **Profile Management**

The system allows authenticated users to view and manage their personal profile information. This includes updating basic details such as name, contact information, and account settings. Profile management ensures personalization while maintaining account security.

- **Workspace Management**

This functionality enables the Manager to create, configure, and manage workspaces. A workspace represents the logical boundary for dashboards, reports, members, and data sources. Workspace management ensures that analytical resources are organized and accessible only to authorized members.

- **Member Management**

The system allows the Manager to manage workspace members. This includes inviting users, assigning roles, and removing members when necessary.

Member management ensures controlled collaboration and enforces role-based access within the workspace.

- **Workspace Governance**

This requirement defines rules and policies that control how the workspace operates. It includes managing access permissions, enforcing role boundaries, and ensuring that shared dashboards and reports follow governance constraints defined by the Manager.

- **Upload Database**

The system enables the Manager to upload and register databases or datasets that will be used for analytical purposes. Uploaded data becomes available for querying, analysis, and visualization within the workspace, forming the foundation of BI operations.

- **Voice Query**

This functionality allows users to interact with the system using spoken language. Voice input is captured, converted into text, and processed to generate analytical queries. This feature enables non-technical users to explore data intuitively without writing SQL queries.

- **Analysis**

The system provides analytical capabilities that allow Analysts to explore data, apply aggregations, filters, and metrics, and derive insights. This requirement supports interactive analysis using both traditional and AI-assisted approaches.

- **Report Generation**

The system allows users to generate reports based on analytical results. Reports present structured insights in a readable format and can be accessed by authorized users for review and decision-making purposes.

- **Dashboard Management**

This requirement enables the Manager to create, edit, and manage dashboards. Dashboards provide a visual representation of analytical data and can include charts, tables, and KPIs. Managed dashboards can be shared with other workspace members.

- **Access Shared Dashboard**

The system allows authorized users to access dashboards shared within the workspace. This ensures that Analysts, Executives, and Dashboard Viewers can consume insights without modifying dashboard content

## 1.2.2 Actors

The BI Voice Agent system defines the following actors based on the high-level use case diagram:

### 1. Manager

The Manager is the owner of the workspace and holds full administrative control.

Responsibilities:

- Manage workspaces and workspace settings
- Upload databases

- Manage members and assign roles
- Create and manage dashboards
- Perform voice queries
- Access reports and analysis
- Manage personal profile and authentication

## **2. Analyst**

The Analyst is a workspace member responsible for analyzing data and generating insights.

Responsibilities:

- Perform data analysis
- Generate and view reports
- Access shared dashboards
- Authenticate and manage personal profile

## **3. Executive**

The Executive is a high-level consumer of insights with read-only analytical access.

Responsibilities:

- View reports
- Access shared dashboards
- Authenticate and manage personal profile

## **4. Report Viewer**

The Report Viewer focuses on consuming reports generated by the system.

Responsibilities:

- View reports
- Authenticate to access reporting features

## **5. Dashboard Viewer**

The Dashboard Viewer is responsible for viewing dashboards shared within the workspace.

Responsibilities:

- Access shared dashboards

- Authenticate to view dashboard content

## **6. Common User**

The Common User represents shared behavior across all authenticated users.

Responsibilities:

- Authenticate into the system
- Manage personal profile information

## **2. Overall Description**

### **2.1 Product Perspective**

The BI Voice Agent platform described in this document is a new, standalone system designed to enable voice-based interaction with Business Intelligence platforms. The system is built as a modular, service-oriented solution that integrates speech processing, AI-based reasoning, and analytical data platforms. The BI Voice Agent is not part of an existing product family, nor is it intended to replace a specific commercial BI tool. Instead, it complements traditional BI systems by providing an intuitive voice-driven interface that simplifies data exploration and decision-making for non-technical users.

### **2.2 Product Features**

#### **• User and Workspace Management**

- Create and manage workspaces.
- Invite users and assign roles (Manager, Analyst, Executive).
- Manage workspace members and permissions.

#### **• Authentication and Profile Management**

- Secure user authentication with role-based access.
- Email verification and session management.
- Profile viewing and editing for all users.

#### **• Voice-Based Querying**

- Allow users to submit analytical questions using voice input.
- Convert speech to text and process it for analysis.
- Enable natural language interaction without SQL knowledge.

- **Data Upload and Integration**

- Upload and register analytical databases.
- Prepare data for querying and visualization.

- **Data Analysis**

- Perform analytical operations such as filtering, aggregation, and trend analysis.
- Support interactive analysis for Analysts.

- **Report Generation**

- Generate structured analytical reports.
- Allow authorized users to view and consume reports.

- **Dashboard Management**

- Create and manage dashboards.
- Share dashboards with workspace members.

- **Access Shared Dashboards**

- Allow Executives and Viewers to access shared dashboards.
- Ensure read-only access for non-administrative users.

## 2.3 User Classes and Characteristics

This section describes the different classes of users interacting with the **BI Voice Agent** system, along with their main characteristics, responsibilities, and access levels. Each user class is defined based on its role within the workspace and the functionalities it can access.

### 2.3.1 Manager

The **Manager** is the primary administrative user and the owner of the workspace. This user class is responsible for configuring and governing the workspace environment and managing its members.

### **Characteristics:**

- Has full administrative privileges within the workspace.
- Possesses decision-making authority over data sources, dashboards, and access control.
- Typically represents a business owner, team leader, or BI administrator.

### **System Access:**

- Workspace management
- Member management and role assignment
- Database upload and integration
- Dashboard creation and management
- Voice-based querying
- Reports and analysis
- Authentication and profile management

### **2.3.2 Analyst**

The **Analyst** is a technical or semi-technical user focused on data exploration and insight generation. Analysts use the system to perform in-depth analysis without managing workspace settings.

### **Characteristics:**

- Skilled in data analysis and interpretation.
- Does not have administrative permissions.
- Works within the boundaries defined by the Manager.

### **System Access:**

- Data analysis features
- Report generation and viewing
- Access to shared dashboards
- Authentication and profile management

### **2.3.3 Executive**

The **Executive** is a high-level stakeholder who consumes insights for strategic decision-making. This user class focuses on reviewing results rather than performing analysis.

### **Characteristics:**

- Non-technical or business-oriented user.
- Requires quick access to summarized insights.
- Limited interaction with system configuration.

### **System Access:**

- View reports
- Access shared dashboards
- Authentication and profile management

#### **2.3.4 Report Viewer**

The **Report Viewer** is a user role dedicated to viewing and consuming reports generated by the system.

##### **Characteristics:**

- Read-only access to analytical reports.
- Does not perform analysis or data manipulation.
- Often represents external or internal stakeholders.

##### **System Access:**

- View reports
- Authentication

#### **2.3.5 Dashboard Viewer**

The **Dashboard Viewer** focuses on accessing and viewing dashboards shared within the workspace.

##### **Characteristics:**

- Read-only user.
- Uses dashboards for monitoring KPIs and performance indicators.
- No permission to modify dashboards or data.

##### **System Access:**

- Access shared dashboards
- Authentication

#### **2.3.6 Common User**

The **Common User** represents shared behavior across all authenticated users in the system.

##### **Characteristics:**

- Represents common system functionality available to all roles.
- Ensures consistent access to core services.

##### **System Access:**

- Authentication
- Profile management

### ***3. System Features***

## **3.1 Functional Requirements**

### **3.1.1 Authentication and Account Management**

#### **REQ-01: Sign Up**

The system shall allow any user to create a new account using a valid email address and password.

- **Actors:** Manager, Analyst, Executive
- **Category:** Authentication

#### **Sub-requirements:**

- **REQ-01.1:** The system shall display a registration form requesting name, email, and password.
- **REQ-01.2:** The system shall validate the email format and password strength.
- **REQ-01.3:** The system shall prevent account creation if the email already exists.
- **REQ-01.4:** If the user signs up as a Manager, the system shall automatically create a Workspace linked to the account.

#### **REQ-02: Email Verification**

The system shall verify the identity of newly registered users through an email verification process.

- **Actors:** All Users
- **Category:** Authentication

#### **Sub-requirements:**

- **REQ-02.1:** The system shall send a verification email upon successful registration.
- **REQ-02.2:** The system shall activate the account only after successful verification.
- **REQ-02.3:** The system shall prevent unverified users from logging in.
- **REQ-02.4:** The system shall allow resending the verification email if the link expires.

#### **REQ-03: Login**

The system shall allow users to log in using valid credentials.

- **Actors:** Manager, Analyst, Executive

- **Category:** Authentication

**Sub-requirements:**

- **REQ-03.1:** The system shall validate the entered email and password.
- **REQ-03.2:** The system shall ensure the account is verified and not suspended.
- **REQ-03.3:** The system shall create an authenticated session (JWT or session token).
- **REQ-03.4:** The system shall redirect users based on their role:
  - Manager → Workspace Dashboard
  - Analyst / Executive → Shared Dashboards

**REQ-04: Logout**

The system shall allow users to securely terminate their session.

- **Actors:** All Users
- **Category:** Authentication

**Sub-requirements:**

- **REQ-04.1:** The system shall invalidate the active session or authentication token.
- **REQ-04.2:** The system shall clear session data from cache or memory.
- **REQ-04.3:** The system shall redirect the user to the login page.

**3.1.2 Profile Management**

**REQ-05: Manage Profile**

The system shall allow users to view and update their personal profile information.

- **Actors:** All Users
- **Category:** User Management

**Sub-requirements:**

- **REQ-05.1:** The system shall display current profile information.
- **REQ-05.2:** The system shall allow editing personal data (name, photo, contact info).
- **REQ-05.3:** The system shall validate updated information before saving.
- **REQ-05.4:** The system shall confirm successful updates.

**REQ-06: Deactivate My Account**

The system shall allow users to deactivate their own account.

- **Actors:** All Users

- **Category:** User Management

**Sub-requirements:**

- **REQ-06.1:** The system shall request confirmation before deactivation.
- **REQ-06.2:** The system shall change account status to “Deactivated”.
- **REQ-06.3:** The system shall terminate all active sessions.
- **REQ-06.4:** Deactivated users shall not be able to log in.

### 3.1.3 Workspace Management

#### **REQ-07: Edit Workspace Information**

The system shall allow the Manager to update Workspace details.

- **Actors:** Manager
- **Category:** Workspace Management

**Sub-requirements:**

- **REQ-07.1:** The system shall display current Workspace information.
- **REQ-07.2:** The system shall allow editing Workspace name and description.
- **REQ-07.3:** The system shall validate updated data.
- **REQ-07.4:** The system shall save and apply changes immediately.

#### **REQ-08: View Workspace Members List**

The system shall allow users to view all members within the Workspace.

- **Actors:** Manager, Analyst, Executive
- **Category:** Workspace Management

**Sub-requirements:**

- **REQ-08.1:** The system shall retrieve Workspace members from the database.
- **REQ-08.2:** The system shall display member name, email, role, and status.
- **REQ-08.3:** Access shall be restricted to members of the Workspace only.

#### **REQ-09: Invite Members**

The system shall allow the Manager to invite new members via email.

- **Actors:** Manager
- **Category:** Workspace Management

**Sub-requirements:**

- **REQ-09.1:** The system shall allow the Manager to enter an email and select a role.

- **REQ-09.2:** The system shall validate email uniqueness.
- **REQ-09.3:** The system shall send an invitation email.
- **REQ-09.4:** The system shall track invitation status.

## **REQ-10: Assign Roles**

The system shall allow the Manager to assign or modify member roles.

- **Actors:** Manager
- **Category:** Access Control

### **Sub-requirements:**

- **REQ-10.1:** The system shall display available roles.
- **REQ-10.2:** The system shall apply role changes immediately.
- **REQ-10.3:** Updated permissions shall take effect without re-login when possible.

## **REQ-11: Manage Members**

The system shall allow the Manager to remove members from the Workspace.

- **Actors:** Manager
- **Category:** Workspace Management

### **Sub-requirements:**

- **REQ-11.1:** The system shall request confirmation before removal.
- **REQ-11.2:** The system shall revoke Workspace access.
- **REQ-11.3:** The system shall terminate active sessions of the removed member.

## **REQ-12: Suspend Member**

The system shall allow the Manager to temporarily suspend a member.

- **Actors:** Manager
- **Category:** Access Control

### **Sub-requirements:**

- **REQ-12.1:** The system shall mark the member as “Suspended”.
- **REQ-12.2:** Suspended members shall be prevented from logging in.
- **REQ-12.3:** Active sessions shall be invalidated immediately.

## **REQ-13: Accept Invitation**

The system shall allow invited users to join a Workspace.

- **Actors:** Analyst, Executive
- **Category:** Workspace Management

### **Sub-requirements:**

- **REQ-13.1:** The system shall validate the invitation token.
- **REQ-13.2:** New users shall be prompted to register before joining.
- **REQ-13.3:** Existing users shall join directly.
- **REQ-13.4:** The system shall assign the predefined role automatically.

## **4. Non-Functional Requirements**

This section describes the non-functional requirements of the **BI Voice Agent** system. These requirements define the quality attributes, performance constraints, and operational characteristics that ensure the system operates efficiently, securely, and reliably, while providing a high-quality user experience.

### **4.1 Performance Requirements**

- **NFR-01:** The system shall process voice input and return a response (SQL query or visualization) within an acceptable time frame.
- **NFR-02:** The system shall support concurrent users within a workspace without significant degradation in performance.
- **NFR-03:** The ETL pipeline shall ingest and process uploaded datasets efficiently, even for large-scale data sources.
- **NFR-04:** Analytical queries executed on ClickHouse shall return results with low latency suitable for interactive BI usage.

### **4.2 Scalability Requirements**

- **NFR-05:** The system shall be horizontally scalable to support an increasing number of users, workspaces, and datasets.
- **NFR-06:** The data ingestion and streaming components shall scale independently based on workload.
- **NFR-07:** The architecture shall support future expansion, such as adding new AI models or analytics components without major system redesign.

### **4.3 Security Requirements**

- **NFR-08:** The system shall enforce role-based access control (RBAC) to restrict functionality based on user roles (Manager, Analyst, Executive).

- **NFR-09:** All user authentication mechanisms shall follow secure standards, including encrypted password storage and secure session handling.
- **NFR-10:** Sensitive data, including credentials and tokens, shall be transmitted using secure communication protocols.
- **NFR-11:** Users shall only access data and dashboards associated with their authorized workspace.

#### **4.4 Reliability and Availability**

- **NFR-12:** The system shall maintain consistent operation during normal usage hours.
- **NFR-13:** In case of partial system failures (e.g., ETL service interruption), the system shall continue operating with graceful degradation.
- **NFR-14:** The system shall log errors and critical events to support monitoring and debugging.

#### **4.5 Usability Requirements**

- **NFR-15:** The system shall provide an intuitive and user-friendly interface suitable for both technical and non-technical users.
- **NFR-16:** Voice-based interaction shall be simple and require minimal user training.
- **NFR-17:** Dashboards and reports shall be presented in a clear and visually understandable manner.

#### **4.6 Maintainability Requirements**

- **NFR-18:** The system shall be modular, allowing individual components (ETL, AI modules, UI) to be maintained or upgraded independently.
- **NFR-19:** The codebase shall follow clear architectural and coding standards to facilitate future maintenance.
- **NFR-20:** System logs and monitoring data shall support issue diagnosis and system health tracking.

#### **4.7 Compatibility and Integration**

- **NFR-21:** The system shall integrate seamlessly with external BI tools such as Metabase.
- **NFR-22:** The system shall support integration with multiple database types through the ETL pipeline.
- **NFR-23:** The system shall be compatible with modern web browsers without requiring additional plugins.

#### **4.8 Portability**

- **NFR-24:** The system shall be deployable in different environments (development, testing, production).
- **NFR-25:** The system shall support containerized deployment to simplify setup and scalability.

## ***4. System Requirements***

| Req-ID  | Requirement Title   | Category        | Priority |
|---------|---|-----------------|----------|
| REQ-1.1 | The system shall allow users to create an account using email and password.         | Authentication  | High     |
| REQ-1.2 | The system shall verify newly registered users via an email verification link.      | Authentication  | High     |
| REQ-1.3 | The system shall allow users to log in using valid credentials.                     | Authentication  | High     |
| REQ-1.4 | The system shall allow users to securely log out and terminate their session.       | Authentication  | Medium   |
| REQ-1.5 | The system shall allow users to view and update their personal profile information. | User Management | Medium   |
| REQ-1.6 | The system shall allow users to deactivate their own account and revoke access.     | User Management | Medium   |

|         |   |                      |        |
|---------|---|----------------------|--------|
| REQ-2.1 | The system shall automatically create a workspace when a Manager registers.         | Workspace Management | High   |
| REQ-2.2 | The system shall allow the Manager to edit workspace information.                   | Workspace Management | High   |
| REQ-2.3 | The system shall allow all workspace members to view the list of workspace members. | Workspace Management | High   |
| REQ-2.4 | The system shall allow the Manager to invite Analyst and Executive users via email. | Workspace Management | High   |
| REQ-2.5 | The system shall allow invited users to accept workspace invitations.               | Workspace Management | High   |
| REQ-3.1 | The system shall allow the Manager to assign roles to workspace members.            | Access Control       | High   |
| REQ-3.2 | The system shall allow the Manager to modify member roles.                          | Access Control       | High   |
| REQ-3.3 | The system shall allow the Manager to remove members from the workspace.            | Workspace Management | High   |
| REQ-3.4 | The system shall allow the Manager to suspend workspace members.                    | Access Control       | Medium |
| REQ-4.1 | The system shall allow the Manager to upload structured databases or datasets.      | Data Management      | High   |
| REQ-4.2 | The system shall process uploaded data through an ETL pipeline.                     | Data Processing      | High   |
| REQ-4.3 | The system shall store processed data in an analytical database.                    | Data Management      | High   |
| REQ-4.4 | The system shall track ETL execution status and metadata.                           | Data Processing      | Medium |

|         |  |                      |        |
|---------|--|----------------------|--------|
| REQ-5.1 | The system shall allow users to submit analytical questions using voice input. | Voice Interaction    | High   |
| REQ-5.2 | The system shall convert voice input to text using speech-to-text processing.  | Voice Interaction    | High   |
| REQ-5.3 | The system shall analyze user queries to determine analytical intent.          | Analysis             | High   |
| REQ-5.4 | The system shall automatically generate SQL queries based on user intent.      | Analysis             | High   |
| REQ-5.5 | The system shall validate generated SQL queries before execution.              | Analysis             | Medium |
| REQ-6.1 | The system shall generate interactive dashboards based on analytical queries.  | Dashboard Management | High   |
| REQ-6.2 | The system shall allow the Manager to manage dashboards.                       | Dashboard Management | High   |
| REQ-6.3 | The system shall allow Analysts and Executives to access shared dashboards.    | Dashboard Access     | High   |
| REQ-6.4 | The system shall allow users to generate and view analytical reports.          | Reporting            | Medium |
| REQ-6.5 | The system shall allow users to export reports in common formats.              | Reporting            | Medium |

Table 6 System Requirements Specification (SRS)

## 5. Requirements modeling

**System features (use case specifications - Sequence Diagrams):**

- **Sign Up**

| Field                    | Description  |
|--------------------------|--|
| <b>Requirement ID</b>    | <b>R-01</b>  |
| <b>Requirement Name</b>  | Sign Up  |
| <b>Actors</b>            | Manager – Analyst – Executive  |
| <b>Preconditions</b>     | 1- The email address must not already exist in the system<br>2- The user must have a valid email address   |
| <b>Main Flow</b>         | 1. The user opens the Sign-Up page.<br>2. The system displays the registration form.<br>3. The user enters the required information: name, email, and password.<br>4. The system validates the input data (email format, password strength, required fields, etc.).<br>5. The system checks whether the email address already exists in the system.<br>6. The system creates a new user account.<br>7. If the user is a Manager, the system automatically creates a Workspace linked to their account.<br>8. The system sends a verification email to the user.<br>9. The system displays a confirmation message: "Your account has been created. Please check your email to complete the verification process." |
| <b>Alternative Flows</b> | <i>A1 - Email Already Exists</i><br>1. The user clicks the “Create Account” button.<br>2. The system detects that the email address is already registered.<br>3. The system displays an error message: "This email address is already in use."<br><i>A2 - Invalid Password</i><br>1. The user submits the registration form.<br>2. The system identifies that the password does not meet security requirements.<br>3. The system displays an error message: "The password does not meet the required criteria."  |

|                       |   |
|-----------------------|---|
| <b>Postconditions</b> | <ol style="list-style-type: none"> <li>1. A new account is successfully created (Manager, Analyst, or Executive).</li> <li>2. If the user is a Manager, a Workspace is automatically created.</li> <li>3. A verification email is sent to the user</li> </ol> |
|-----------------------|---|

Table 7 : Use Case Specification for "Sign Up"

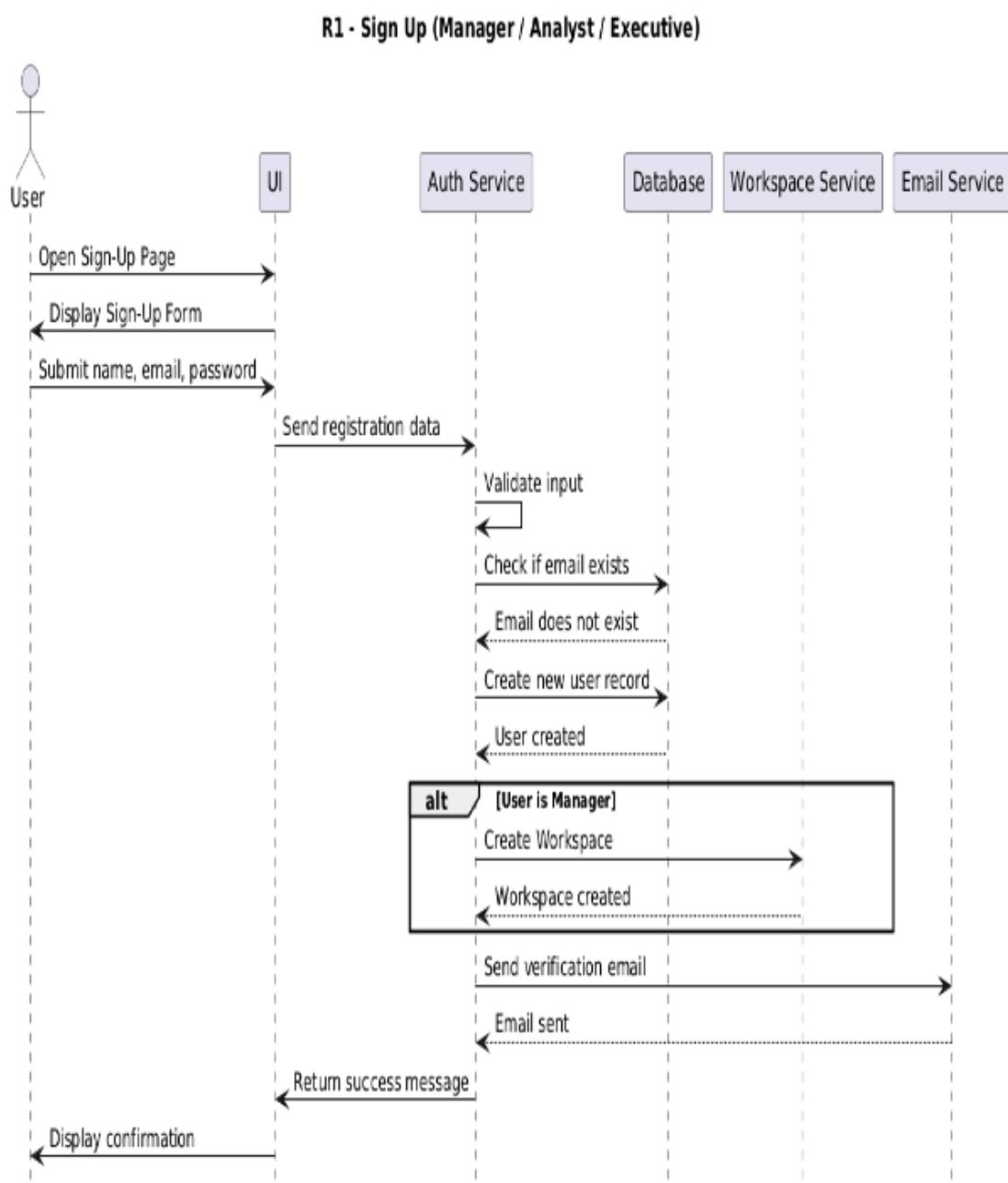


Figure 2 : Sequence Diagram for "Sign Up"

- **Email verification**

| <b>Field</b>             | <b>Description</b>  |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-02</b>   |
| <b>Requirement Name</b>  | Email verification  |
| <b>Actors</b>            | Manager – Analyst – Executive   |
| <b>Preconditions</b>     | 1. User has completed Sign Up<br>2. User account is in Unverified state   |
| <b>Main Flow</b>         | 1. User checks their email inbox<br>2. User opens the verification email<br>3. User clicks the verification link<br>4. System receives the verification request<br>5. System validates the verification token<br>6. System activates the user account<br>7. System displays: “Your account has been successfully verified. You can now log in.”                                 |
| <b>Alternative Flows</b> | <i>A1 - Expired Token</i><br>1. User clicks the link<br>2. System detects the token is expired<br>3. System displays: “Verification link expired.”<br>4. System offers “Resend verification email.”<br><i>A2 - Already Verified</i><br>1. User clicks the link<br>2. System detects the account is already activated<br>3. System displays: “Your account is already verified.” |
| <b>Postconditions</b>    | 1. User account is activated<br>2. User can log in to the system<br>3. Account status updated to Activated in the database  |

Table 8 : Use Case Specification for “Email Verification”

## R2 - Email Verification (Manager / Analyst / Executive)

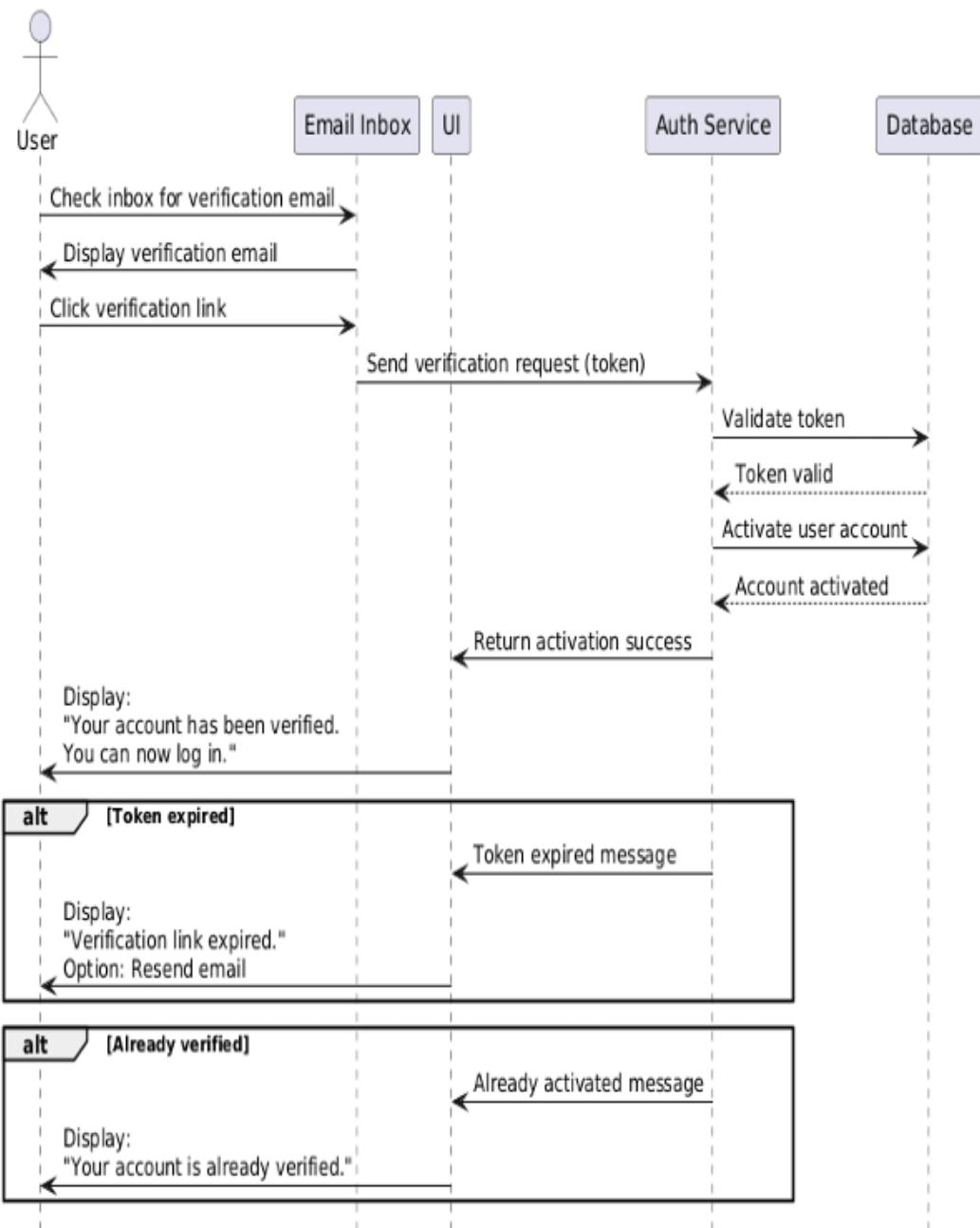


Figure 3 : Sequence Diagram for "Email Verification"

## • Login

| Field                    | Description   |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-03</b>   |
| <b>Requirement Name</b>  | Login   |
| <b>Actors</b>            | Manager – Analyst – Executive   |
| <b>Preconditions</b>     | <ol style="list-style-type: none"> <li>1. The user must already have a registered account in the system</li> <li>User</li> <li>2. The user's account must be verified via email</li> </ol>  |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. The user opens the Login page</li> <li>2. The system displays the login form</li> <li>3. The user enters their email and password</li> <li>4. The user clicks the “Login” button</li> <li>5. The system validates the provided credentials</li> <li>6. The system checks for a matching account in the database</li> <li>7. The system verifies that the account is activated and not suspended</li> <li>8. If the credentials are correct, the system creates a login session (Session or JWT Token)</li> <li>9. The system redirects the user to the appropriate landing page based on their role: <ul style="list-style-type: none"> <li>• Manager : Workspace Dashboard</li> <li>• Analyst / Executive : Shared Dashboard</li> </ul> </li> <li>10. The system displays a success message or directly redirects the user to their dashboard</li> </ol> |
| <b>Alternative Flows</b> | <p><i>A1 - Invalid Credentials</i></p> <ol style="list-style-type: none"> <li>1. The user enters an incorrect email or password</li> <li>2. The system displays:“Invalid login credentials.”</li> </ol> <p><i>A2 - Account Not Verified</i></p> <ol style="list-style-type: none"> <li>1. The user attempts to log in before email verification</li> <li>2. The system displays:“Please verify your email before logging in.”</li> </ol> <p><i>A3 - Account Suspended</i></p> <ol style="list-style-type: none"> <li>1. The system checks the user’s status</li> <li>2. The system detects that the account is suspended by the Manager</li> </ol>  |

|                       |  |
|-----------------------|--|
|                       | 3. The system displays: "Your account is suspended. You cannot log in."  |
| <b>Postconditions</b> | 4. User is successfully authenticated<br>5. A session (or JWT Token) is generated<br>6. User is redirected to the correct landing page according to their role |

Table 9 : Use Case Specification for "Login"

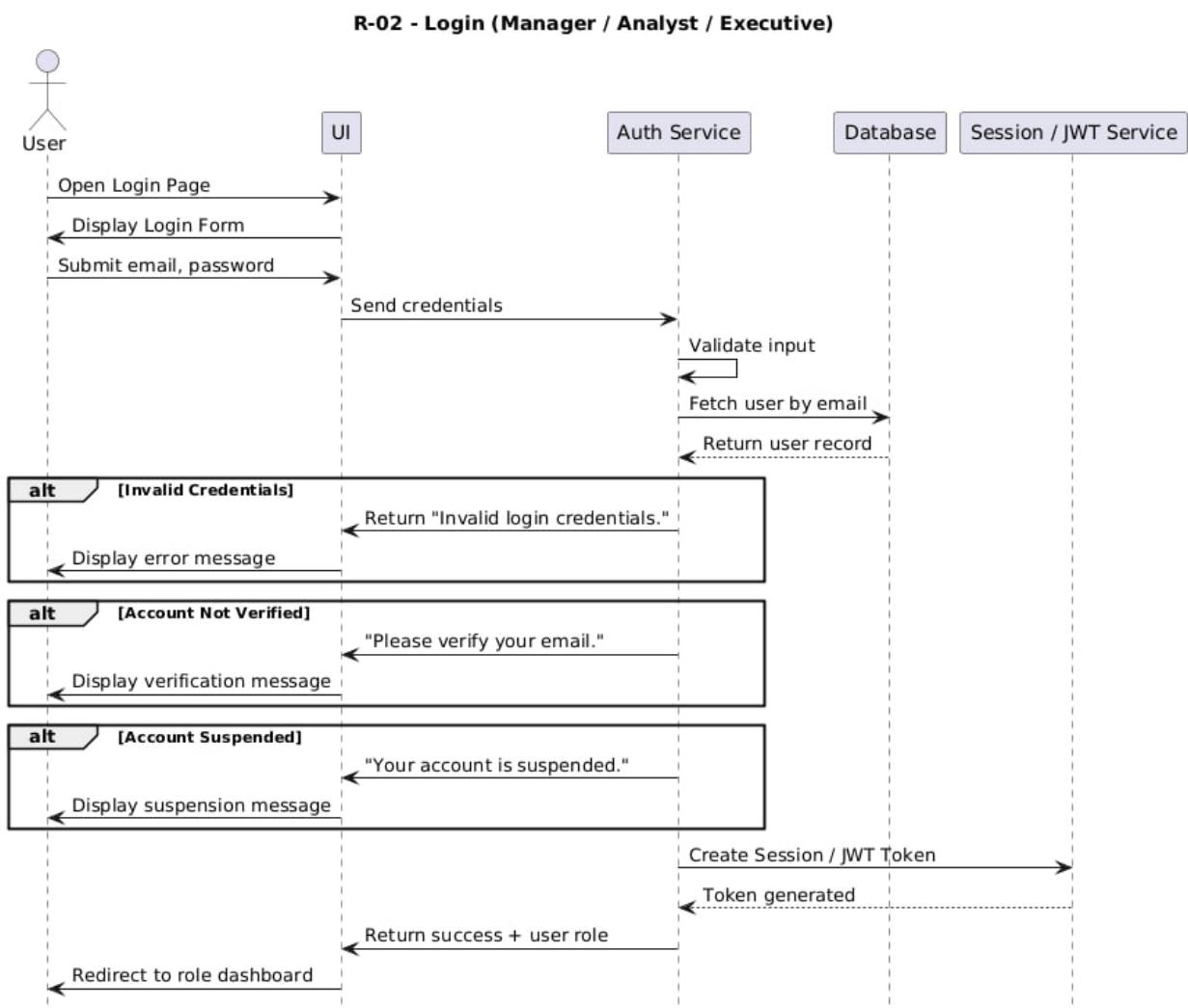


Figure 4 : Sequence Diagram for "Log in"

- **Logout**

| <b>Field</b>             | <b>Description</b>   |
|--------------------------|--|
| <b>Requirement ID</b>    | <b>R-04</b>  |
| <b>Requirement Name</b>  | Logout   |
| <b>Actors</b>            | Manager – Analyst – Executive  |
| <b>Preconditions</b>     | <ol style="list-style-type: none"> <li>1. The user must be currently logged in</li> <li>2. A valid session or authentication token must exist</li> </ol>   |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. The user clicks the “Logout” button</li> <li>2. The system receives the logout request</li> <li>3. The system invalidates the active session or authentication token</li> <li>4. The system clears any session-related data stored in memory or cache</li> <li>5. The system redirects the user to the login page or public home page</li> <li>6. The system displays a confirmation message: a. “You have been logged out successfully.”</li> </ol> |
| <b>Alternative Flows</b> | <p><i>A1 - Session Already Expired</i></p> <ol style="list-style-type: none"> <li>1. The user attempts to log out</li> <li>2. The system detects that the session has already expired</li> <li>3. The system redirects the user to the login page without showing an error message</li> </ol>  |
| <b>Postconditions</b>    | <ol style="list-style-type: none"> <li>1. The current user session is invalidated</li> <li>2. The user is no longer authenticated</li> <li>3. No protected actions can be performed until the user logs in again</li> </ol>  |

*Table 10 : Use Case Specification for “Logout”*

#### R-04 - Logout

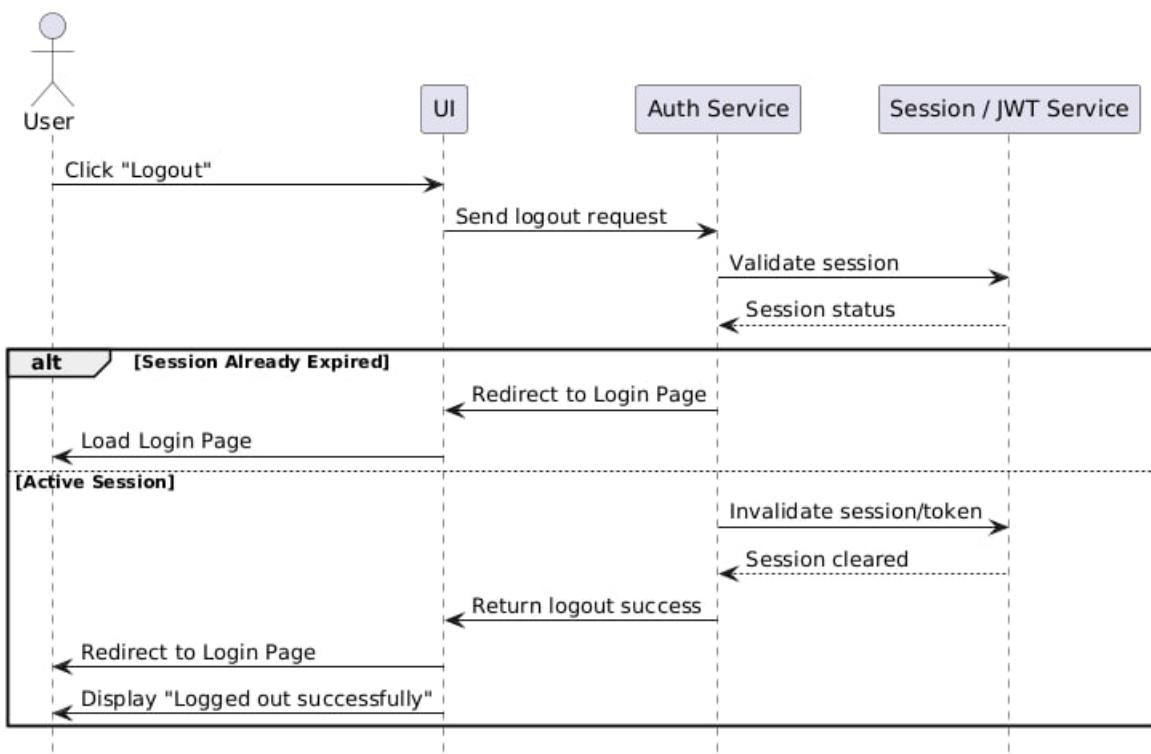


Figure 5 : Sequence Diagram for "Logout"

- **Manage Profile**

| <b>Field</b>             | <b>Description</b>   |
|--------------------------|--|
| <b>Requirement ID</b>    | <b>R-05</b>  |
| <b>Requirement Name</b>  | Manage Profile   |
| <b>Actors</b>            | Manager – Analyst – Executive  |
| <b>Preconditions</b>     | 1. User is logged in<br>2. Account is active   |
| <b>Main Flow</b>         | 1. User opens the Profile page<br>2. System retrieves and displays profile info<br>3. System shows editable fields (name, photo, phone...)<br>4. User updates fields and clicks Save<br>5. System validates updated data<br>6. System updates the database record<br>7. System shows success message |
| <b>Alternative Flows</b> | <i>A1 - Invalid Data</i><br>1. User clicks Save<br>2. System detects invalid fields<br>3. System displays: “Invalid profile information.”  |
| <b>Postconditions</b>    | 1. The current user session is invalidated<br>2. The user is no longer authenticated<br>3. No protected actions can be performed until the user logs in again  |

*Table 11 : Use Case Specification for “Manage Profile”*

### R-05 - Manage Profile

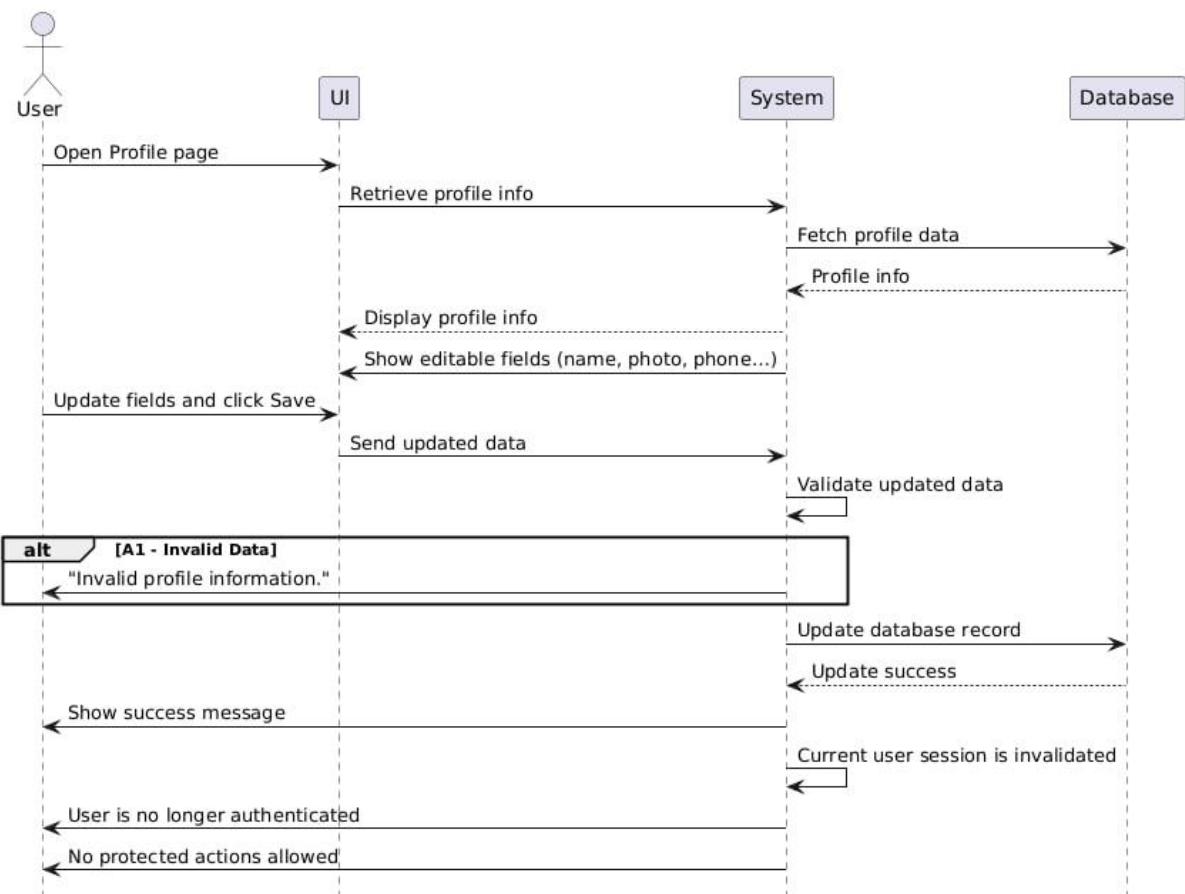


Figure 6 : Sequence Diagram for "Manage Profile"

- **Deactivate My Profile**

| Field                    | Description   |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-06</b>   |
| <b>Requirement Name</b>  | Deactivate My Profile   |
| <b>Actors</b>            | Manager – Analyst – Executive   |
| <b>Preconditions</b>     | 1. User is logged in  |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. User opens Account Settings</li> <li>2. System displays account management options including "Deactivate Account"</li> <li>3. User selects "Deactivate Account"</li> <li>4. System shows a confirmation warning</li> <li>5. User confirms the request</li> <li>6. System verifies the user's identity</li> <li>7. System sets account status to "Deactivated"</li> <li>8. System terminates active sessions</li> <li>9. System shows a success message</li> </ol> |
| <b>Alternative Flows</b> | <p>A1 - User Cancellation</p> <ol style="list-style-type: none"> <li>1. User clicks "Deactivate Account"</li> <li>2. System shows confirmation</li> <li>3. User selects "Cancel"</li> <li>4. System returns to settings unchanged</li> </ol>  |
| <b>Postconditions</b>    | <ol style="list-style-type: none"> <li>1. Account is deactivated</li> <li>2. User cannot log in unless reactivation is supported</li> <li>3. All sessions are terminated</li> </ol>   |

Table 12 : Use Case Specification for "Deactive My Profile"

### R-06 - Deactivate My Profile

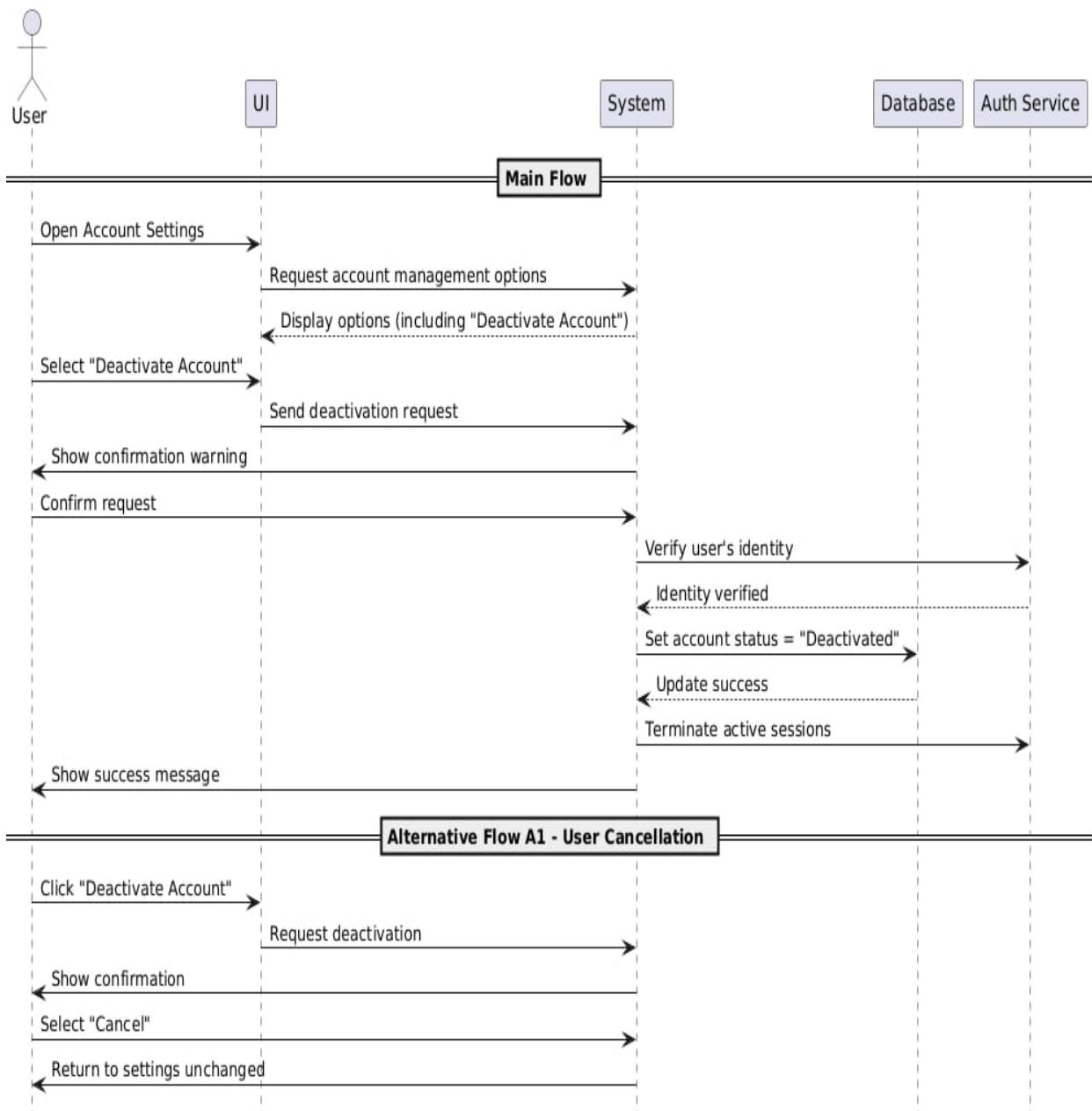


Figure 7 : Sequence Diagram for "Deactivate My Profile"

- **Edit Workspace Info**

| <b>Field</b>             | Description   |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-07</b>   |
| <b>Requirement Name</b>  | Edit Workspace Info   |
| <b>Actors</b>            | Manager   |
| <b>Preconditions</b>     | 1. manager is logged in   |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. Manager opens Workspace Settings</li> <li>2. System displays current workspace info</li> <li>3. Manager clicks “Edit Workspace Info”</li> <li>4. System shows editable fields</li> <li>5. Manager updates fields and clicks Save</li> <li>6. System validates updated data</li> <li>7. System updates the database</li> <li>8. System displays success message</li> </ol> |
| <b>Alternative Flows</b> | <p><i>A1 - Invalid Data</i></p> <ol style="list-style-type: none"> <li>1. Manager clicks Save</li> <li>2. System detects invalid data</li> <li>3. System shows: “Invalid input data.”</li> </ol>  |
| <b>Postconditions</b>    | <ol style="list-style-type: none"> <li>1. Workspace info updated</li> <li>2. Updated values displayed in manager UI</li> </ol>  |

*Table 13 : Use Case Specification for "Edit Workspace Info "*

## R-07 - Edit Workspace Info

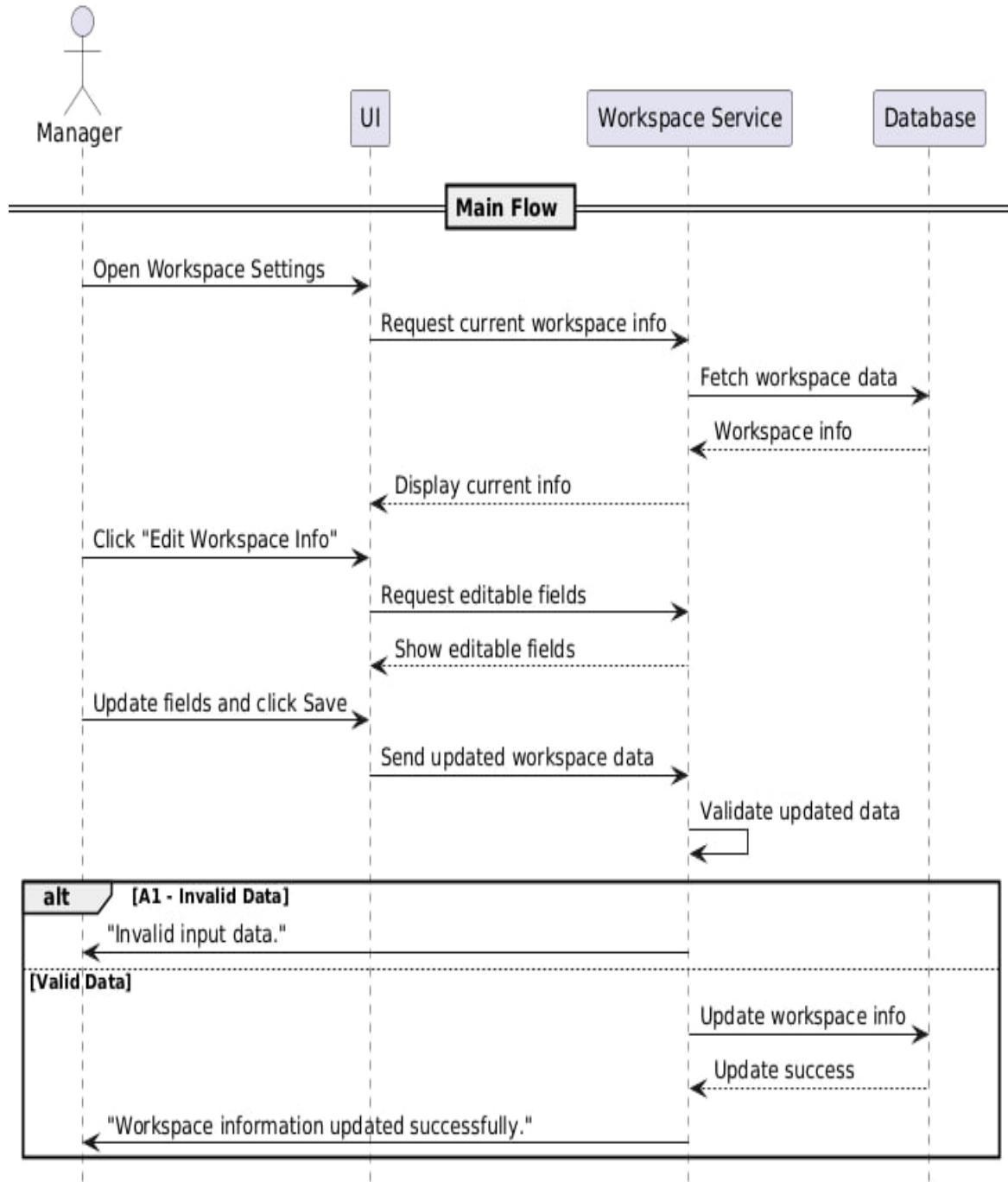


Figure 8 : Sequence Diagram for "Edit Workspace Info"

- **View Workspace Members List**

| Field                    | Description   |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-08</b>   |
| <b>Requirement Name</b>  | View Workspace Members List   |
| <b>Actors</b>            | Manager – Analyst – Executive   |
| <b>Preconditions</b>     | 1. user is logged in<br>2. User is a member of the current workspace  |
| <b>Main Flow</b>         | 1. User opens Workspace Members page<br>2. System verifies permission to view members<br>3. User stays on the same page<br>4. System retrieves workspace members from the database<br>5. System displays the list (name, email, role, status) |
| <b>Alternative Flows</b> |   |
| <b>Postconditions</b>    | 1. Members list successfully retrieved<br>2. User can view workspace members  |

Table 14 : Use Case Specification for "View Workspace Member list"

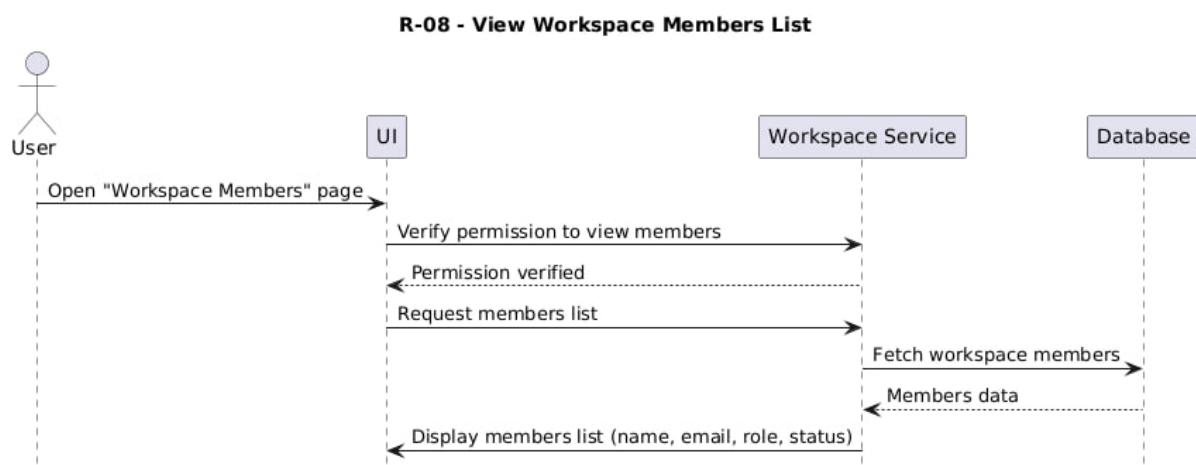


Figure 9 : Sequence Diagram for " View Workspace members list"

- **Invite Members**

| <b>Field</b>             | <b>Description</b>  |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-09</b>   |
| <b>Requirement Name</b>  | Invite Members  |
| <b>Actors</b>            | Manager   |
| <b>Preconditions</b>     | <ol style="list-style-type: none"> <li>1. manager is logged in</li> <li>2. New member's email is valid</li> </ol>   |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. Manager opens Members Management page</li> <li>2. System shows "Invite Member" option</li> <li>3. Manager clicks "Invite Member"</li> <li>4. System displays invitation form</li> <li>5. Manager enters email and selects role</li> <li>6. System validates email and role</li> <li>7. System creates invitation record</li> <li>8. System sends invitation email</li> <li>9. System shows success message</li> </ol>   |
| <b>Alternative Flows</b> | <p><i>A1 - Invalid Email:</i></p> <ol style="list-style-type: none"> <li>1. Manager enters invalid email.</li> <li>2. System rejects it.</li> <li>3. System shows: "Invalid email address."</li> </ol> <p><i>A2 - Already Invited</i></p> <ol style="list-style-type: none"> <li>1. Manager enters previously invited email</li> <li>2. System detects existing invitation</li> <li>3. System shows: "Invitation already sent."</li> </ol> <p><i>A3 - User Already Exists</i></p> <ol style="list-style-type: none"> <li>1. Manager enters email of existing user</li> <li>2. System detects existing account</li> <li>3. System shows: "User already exists."</li> </ol> |
| <b>Postconditions</b>    | <ol style="list-style-type: none"> <li>1. New invitation is created</li> <li>2. Invitation email is sent</li> </ol>   |

*Table 15 : Use Case Specification for "Invite Member"*

### R-09 - Invite Members

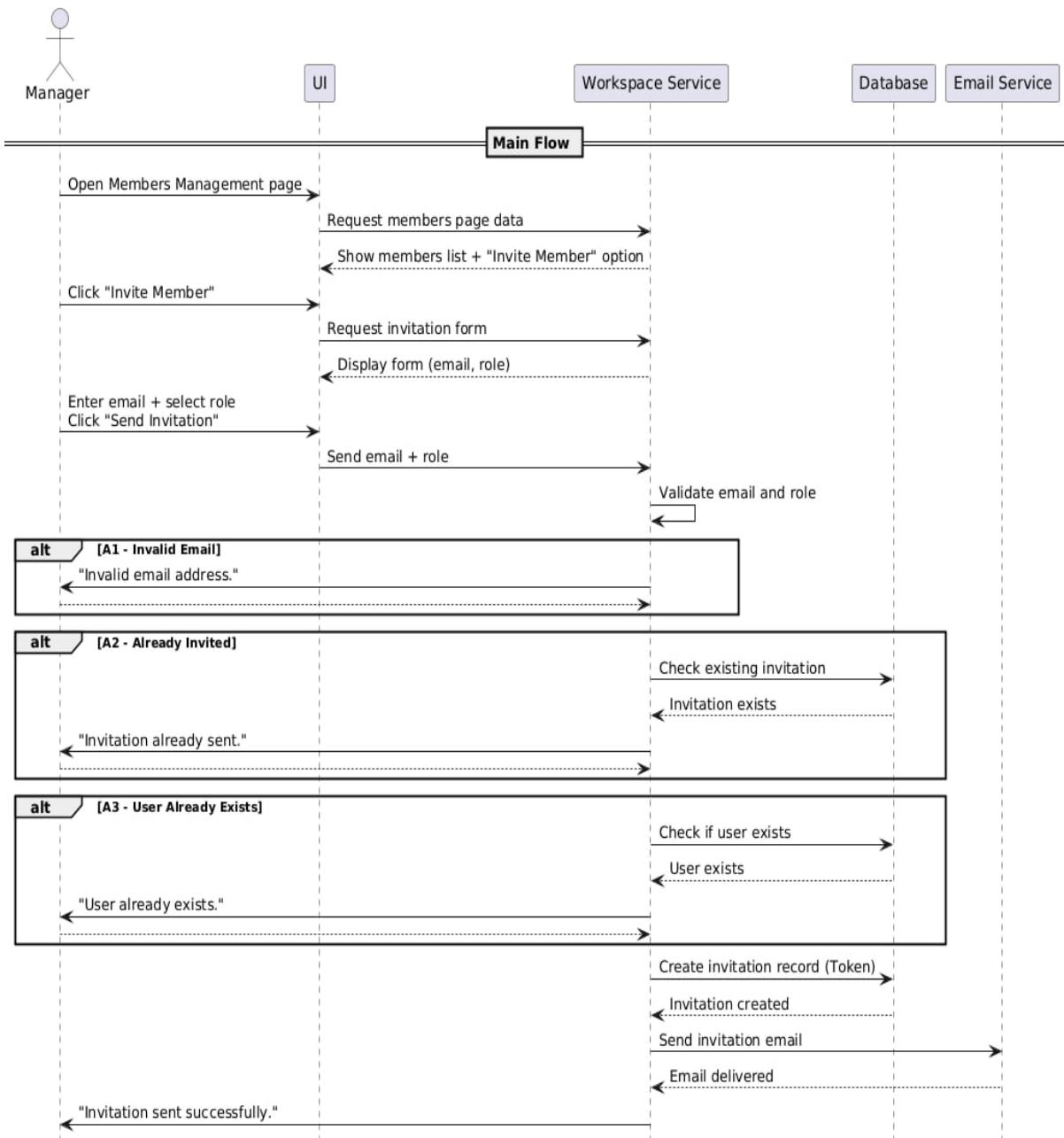


Figure 10 : Sequence Diagram for " invite members"

- **Assign Roles**

| Field                    | Description  |
|--------------------------|--|
| <b>Requirement ID</b>    | <b>R-10</b>  |
| <b>Requirement Name</b>  | Assign Roles   |
| <b>Actors</b>            | Manager  |
| <b>Preconditions</b>     | 1. manager is logged in<br>2. Member exists in the workspace   |
| <b>Main Flow</b>         | 1. Manager opens Members Management page<br>2. System displays members and roles<br>3. Manager selects a member<br>4. System shows available actions including Change Role<br>5. Manager selects new role<br>6. System validates the role<br>7. System updates the role in the database<br>8. System shows success message |
| <b>Alternative Flows</b> | A1 - Member Not Found<br>1. Manager selects a member<br>2. System detects the member does not exist<br>3. System shows: "Member not found."  |
| <b>Postconditions</b>    | 1. Role is updated in the database<br>2. New role appears immediately in the UI  |

*Table 16 : Use Case Specification for "Assign Roles"*

### R-10 - Assign Roles

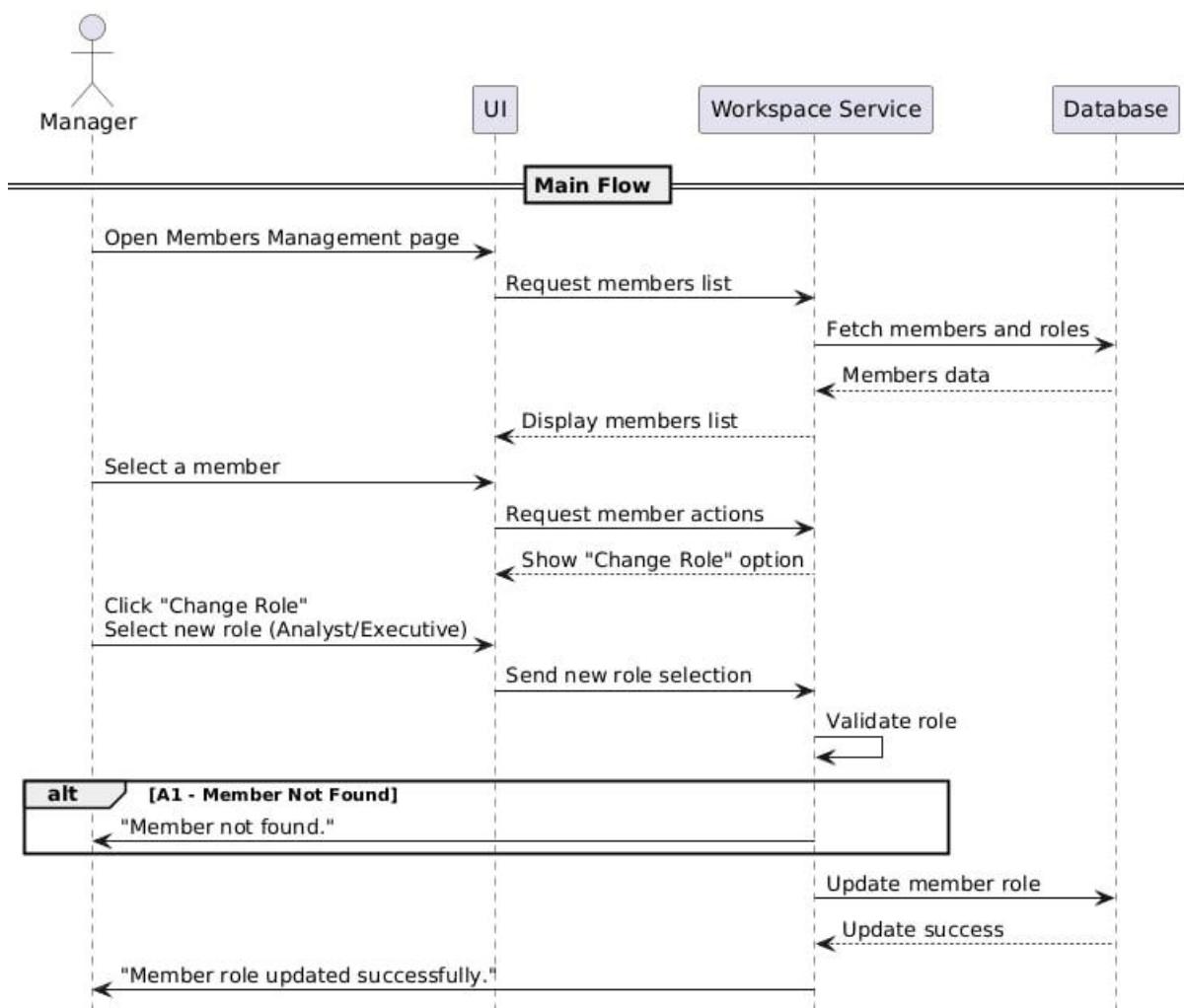


Figure 11 : Sequence Diagram for " Assign Roles"

- Remove Members

| Field             | Description   |
|-------------------|---|
| Requirement ID    | R-11  |
| Requirement Name  | Remove Members  |
| Actors            | Manager   |
| Preconditions     | 1. manager is logged in<br>2. Member exists in the workspace  |
| Main Flow         | 1. Manager opens Members Management page<br>2. System displays members with a Remove option<br>3. Manager selects a member and clicks “Remove”<br>4. System shows confirmation dialog<br>5. Manager confirms removal<br>6. System verifies permissions<br>7. System removes the member<br>8. System terminates active sessions<br>9. System shows success message |
| Alternative Flows | <i>A1 - Cancel Removal</i><br>1. System displays confirmation dialog<br>2. Manager clicks “Cancel”<br>3. System returns to Members Management with no changes   |
| Postconditions    | 1. Member is removed<br>2. Member loses workspace access  |

Table 17 : Use Case Specification for “Remove Members”

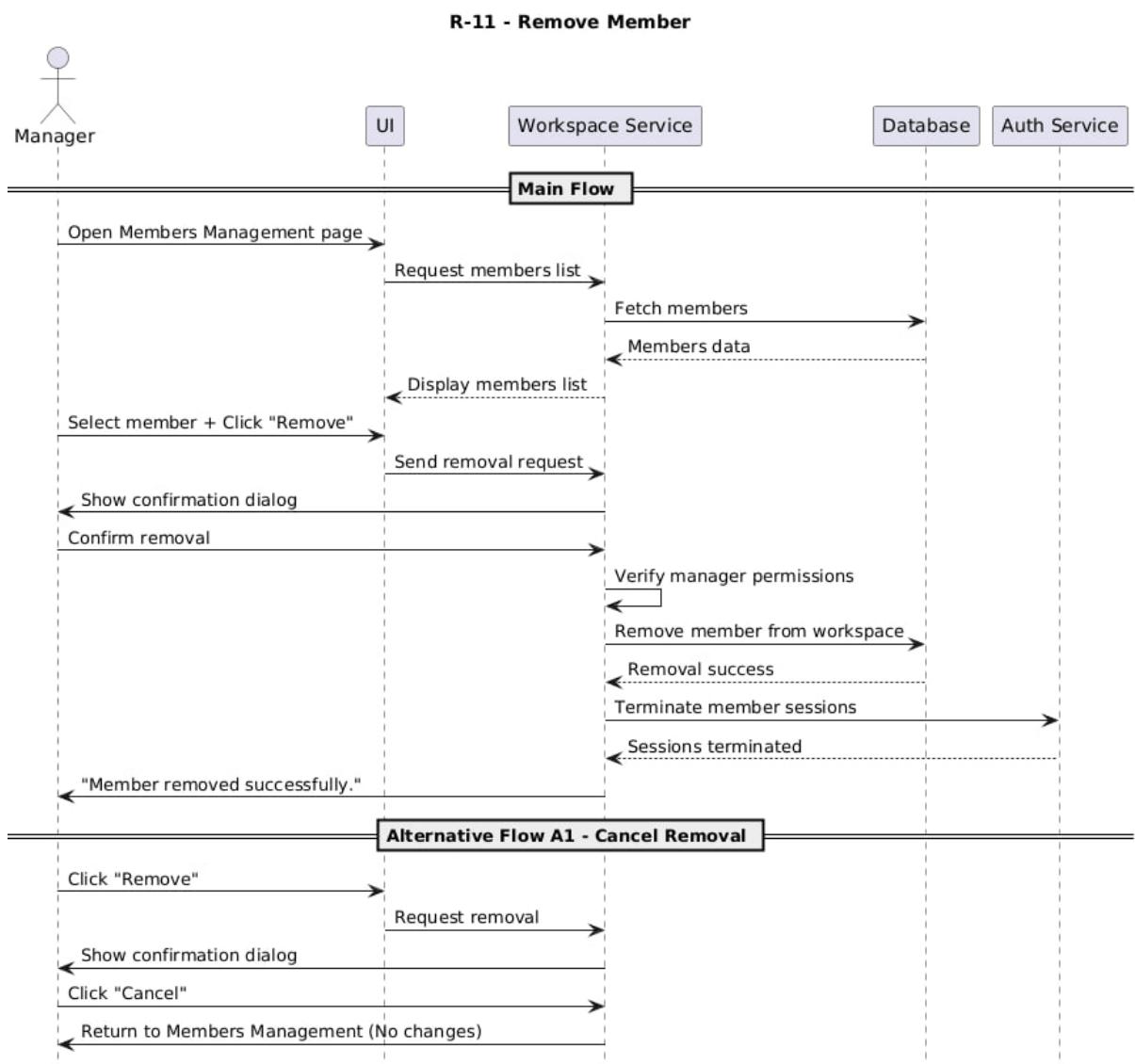


Figure 12 : Sequence Diagram for "Remove Member"

- **Suspend Member**

| <b>Field</b>             | <b>Description</b>  |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-12</b>   |
| <b>Requirement Name</b>  | Suspend Member  |
| <b>Actors</b>            | Manager   |
| <b>Preconditions</b>     | <ol style="list-style-type: none"> <li>1. manager is logged in</li> <li>2. Member exists in the workspace</li> </ol>  |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. Manager opens Members Management page</li> <li>2. System shows members with status (Active/Suspended)</li> <li>3. Manager selects a member and views details</li> <li>4. System shows “Suspend Member” option</li> <li>5. Manager clicks “Suspend Member”</li> <li>6. System shows confirmation with impact explanation</li> <li>7. Manager confirms</li> <li>8. System updates member status to Suspended</li> <li>9. System invalidates member's session</li> <li>10. System shows success message</li> </ol> |
| <b>Alternative Flows</b> | <p><i>A1 - Member Already Suspended</i></p> <ol style="list-style-type: none"> <li>1. Manager selects the member</li> <li>2. System detects status : Suspended</li> <li>3. System shows: “Member is already suspended.”</li> </ol>  |
| <b>Postconditions</b>    | <ol style="list-style-type: none"> <li>1. Member is suspended</li> <li>2. Member cannot log in to the workspace</li> <li>3. Active sessions are terminated</li> </ol>   |

*Table 18 : Use Case Specification for “Suspend Member”*

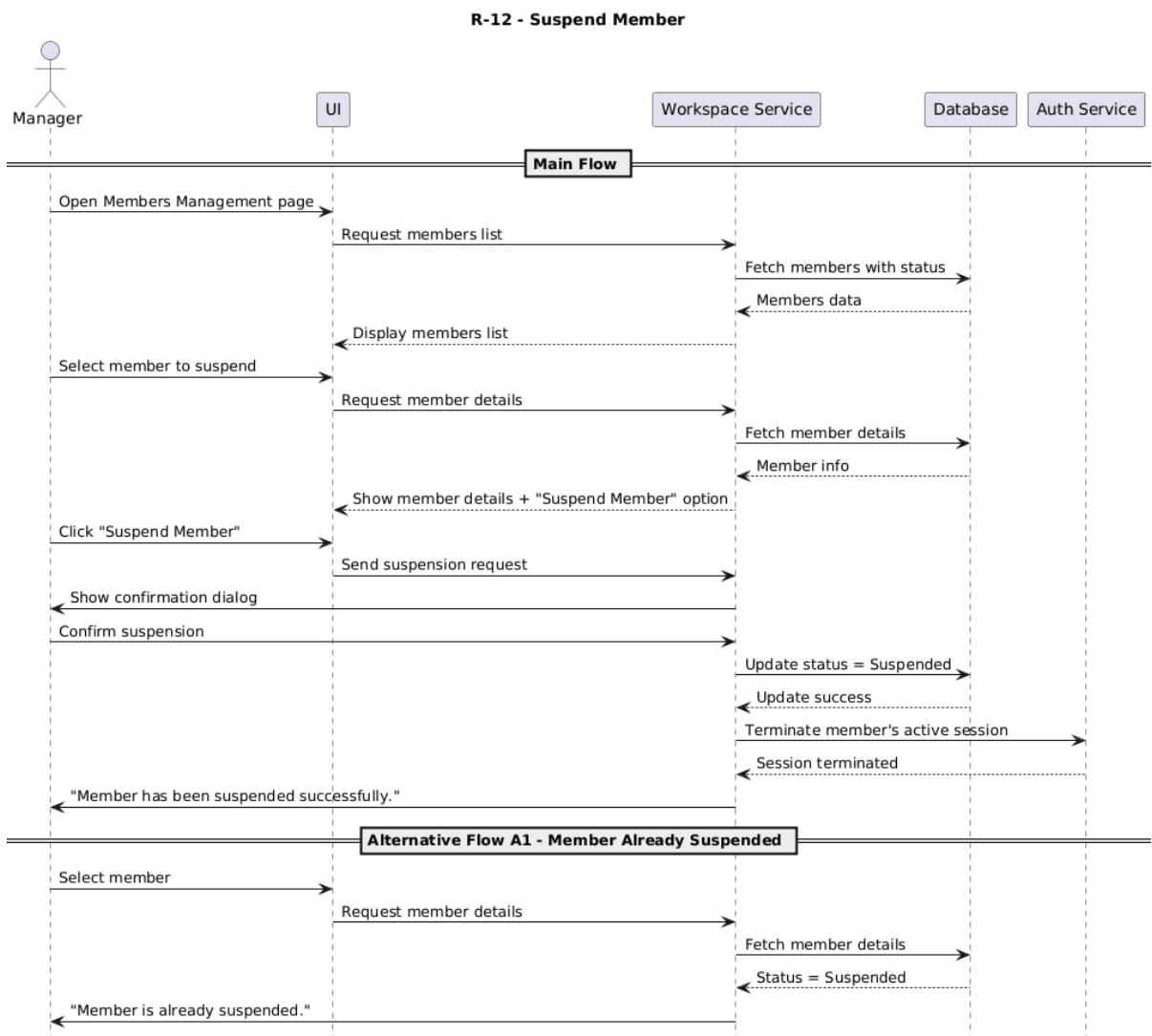


Figure 13 : Sequence Diagram for "Suspend Member"

- **Accept Invitation**

| <b>Field</b>             | <b>Description</b>  |
|--------------------------|---|
| <b>Requirement ID</b>    | <b>R-13</b>   |
| <b>Requirement Name</b>  | Accept Invitation   |
| <b>Actors</b>            | Analyst – Executive   |
| <b>Preconditions</b>     | <ol style="list-style-type: none"> <li>1. User has been invited by the manager</li> <li>2. User received the invitation email</li> <li>3. Invitation token is valid and unused</li> </ol>   |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. User opens invitation email</li> <li>2. System shows “Join Workspace” link</li> <li>3. User clicks the link</li> <li>4. System validates token</li> <li>5. System checks invitation validity and usage</li> <li>6. System displays one of two options: <ul style="list-style-type: none"> <li>• No account : Registration form</li> <li>• Existing account : Join button</li> </ul> </li> <li>7. User: <ul style="list-style-type: none"> <li>• New user : Register &amp; Join</li> <li>• Existing user : Join Now</li> </ul> </li> <li>8. System creates or links account</li> <li>9. System assigns the role</li> <li>10. System marks invitation as Used</li> <li>11. System displays success message</li> </ol> |
| <b>Alternative Flows</b> | <p><i>A1 - Invalid Link</i></p> <ol style="list-style-type: none"> <li>4. User clicks the link</li> <li>5. System checks invitation token</li> <li>6. Token is invalid</li> <li>7. System displays: “Invalid invitation link.”</li> </ol> <p><i>A2 - Invitation Expired</i></p> <ol style="list-style-type: none"> <li>1. User clicks the link</li> <li>2. System detects expired invitation</li> <li>3. System displays: “Invitation expired.”</li> </ol> <p><i>A3 - Invitation Already Used</i></p> <ol style="list-style-type: none"> <li>1. User opens the link</li> </ol>  |

|                       |   |
|-----------------------|---|
|                       | 2. System detects invitation status : Used<br>3. System displays: “This invitation was already used.” |
| <b>Postconditions</b> | 1. User is added to the workspace<br>2. Invitation is marked as Used<br>3. User has the assigned role |

*Table 19 : Use Case Specification for “Accept Invitation”*

### R-13 - Accept Invitation

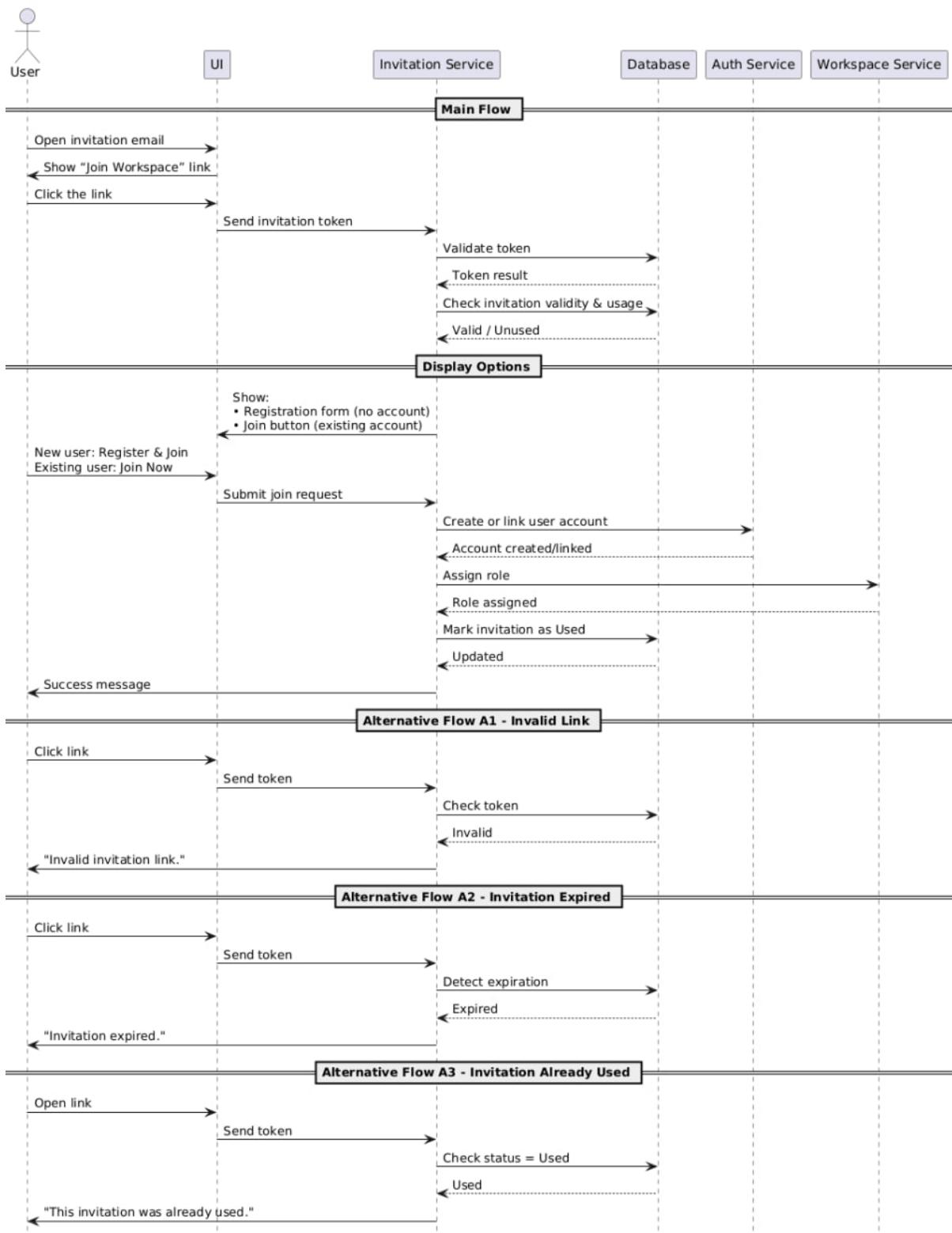


Figure 14 : Sequence Diagram for "Accept invitation"

- **Record Voice and Generate Report**

|                          |  |
|--------------------------|--|
| <b>Requirement ID</b>    | <b>R-14</b>  |
| <b>Requirement Name</b>  | Record Voice and Generate Report   |
| <b>Actors</b>            | Manager  |
| <b>Preconditions</b>     | 1. Manager is logged in<br>2. Microphone permission is granted   |
| <b>Main Flow</b>         | 1. Manager opens Voice Input page<br>2. System displays “Record Voice” option<br>3. Manager clicks “Record”<br>4. System records audio<br>5. Manager clicks “Stop”<br>6. System sends audio to Whisper model<br>7. System converts voice to text<br>8. System sends text to LLM for SQL generation<br>9. System generates SQL query<br>10. System executes SQL on ClickHouse<br>11. System sends results to Metabase<br>12. Metabase generates chart and dashboard<br>13. System saves report metadata<br>14. System displays updated dashboard to Manager |
| <b>Alternative Flows</b> | A1 – Voice Processing Failed<br>1. System shows error message  |
| <b>Postconditions</b>    | 1. Report is created<br>2. Report appears in Manager dashboard   |

*Table 20 : Use Case Specification for “Record Voice”*

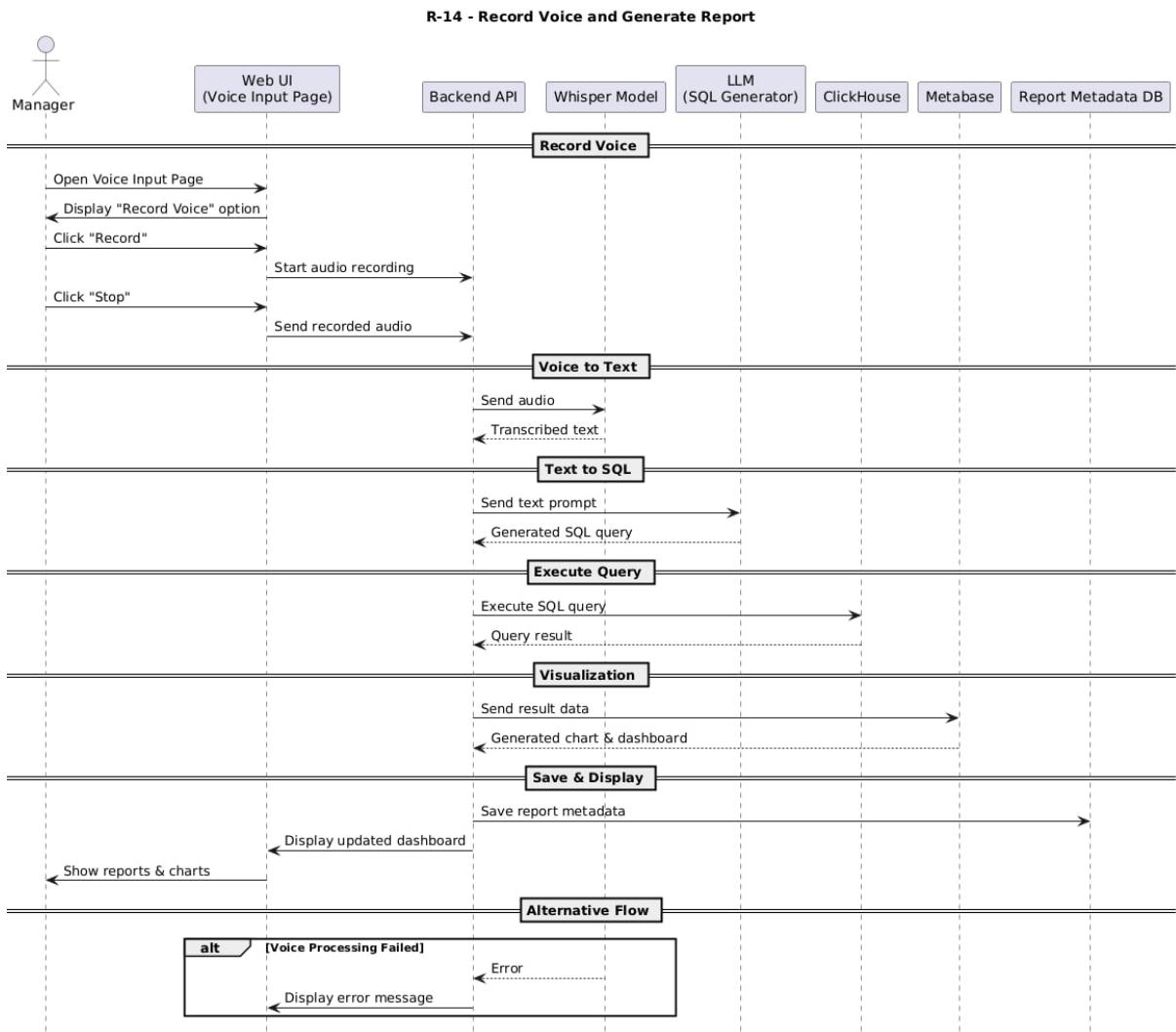


Figure 15 : Sequence Diagram for "Record voice"

- **Upload Voice and Generate Report**

|                          |  |
|--------------------------|--|
| <b>Requirement ID</b>    | R-15   |
| <b>Requirement Name</b>  | Upload Voice and Generate Report   |
| <b>Actors</b>            | Manager  |
| <b>Preconditions</b>     | 1. Manager is logged in<br>2. Audio file exists  |
| <b>Main Flow</b>         | 1. Manager opens Upload Voice page<br>2. System displays upload interface<br>3. Manager uploads audio file<br>4. System validates file format<br>5. System sends audio to Whisper model<br>6. System converts voice to text<br>7. System sends text to LLM for SQL generation<br>8. System generates SQL query<br>9. System executes query on ClickHouse<br>10. System sends result to Metabase<br>11. Metabase generates visualization<br>12. System saves report and updates dashboard |
| <b>Alternative Flows</b> | A1 – Invalid Audio File<br>1. System rejects upload  |
| <b>Postconditions</b>    | 1. Report is created<br>2. Report is visible on dashboard  |

*Table 21 : Use Case Specification for “Upload Voice”*

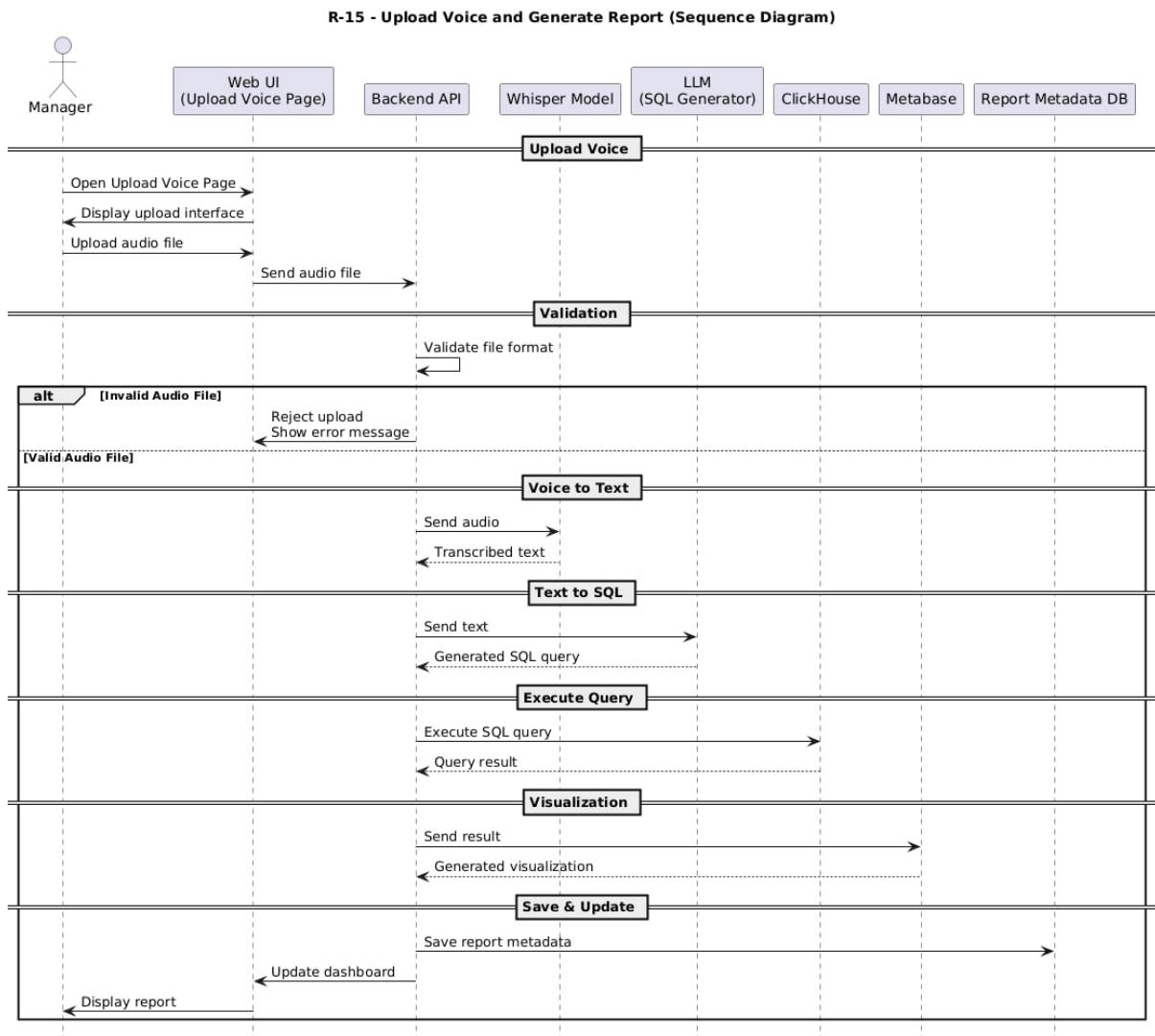


Figure 16 : Sequence Diagram for " Upload Voice"

- **View Dashboard Report**

|                          |   |
|--------------------------|---|
| <b>Requirement ID</b>    | R-16  |
| <b>Requirement Name</b>  | View Dashboard Reports  |
| <b>Actors</b>            | Manager, Analyst, Executive   |
| <b>Preconditions</b>     | 1. User is logged in  |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. User logs into the system</li> <li>2. System identifies user role</li> <li>3. System redirects user to dashboard page</li> <li>4. System retrieves available reports from Metabase</li> <li>5. System displays dashboards and visual reports</li> </ol> |
| <b>Alternative Flows</b> | <p>A1 – No Reports Available</p> <ol style="list-style-type: none"> <li>1. System displays empty dashboard message</li> </ol>   |
| <b>Postconditions</b>    | 1. Dashboard with reports is displayed to the user  |

Table 22 : Use Case Specification for "View Dashboard report"

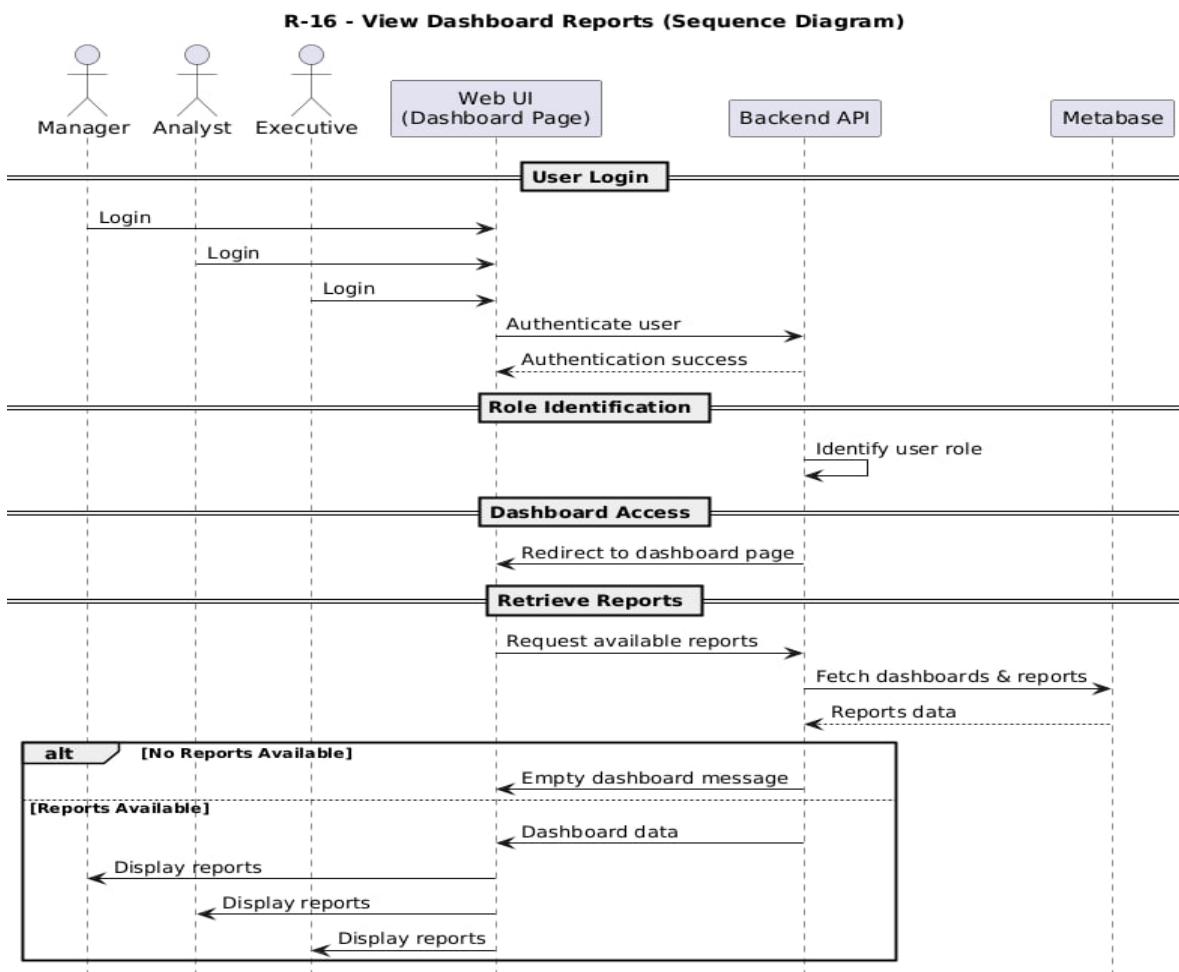


Figure 17 : Sequence Diagram for "View Dashboard Reports"

## • View Previous Reports

|                          |  |
|--------------------------|--|
| <b>Requirement ID</b>    | R-17   |
| <b>Requirement Name</b>  | View Previous Reports  |
| <b>Actors</b>            | Manager, Analyst, Executive  |
| <b>Preconditions</b>     | 1. Reports exist in the system   |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. User opens Reports section</li> <li>2. System retrieves report list based on user role</li> <li>3. System displays list of previous reports</li> <li>4. User selects a report</li> </ol> |
| <b>Alternative Flows</b> | A1 – Report Not Found<br><ol style="list-style-type: none"> <li>1. System displays error message</li> </ol>  |
| <b>Postconditions</b>    | 1. Selected report is opened   |

Table 23 : Use Case Specification for "View Previous Reports"

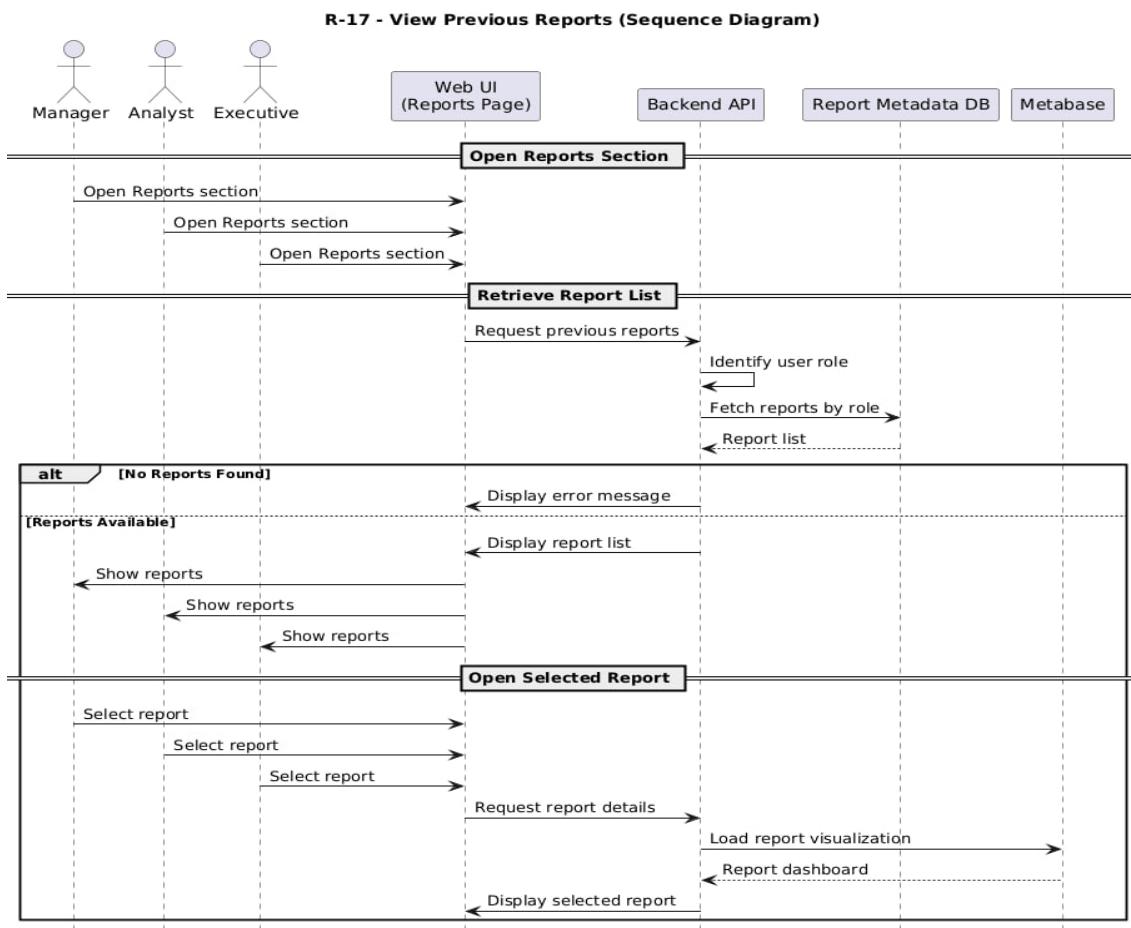


Figure 18 : Sequence Diagram for "View Previous Reports"

## • View Report Details

|                          |   |
|--------------------------|---|
| <b>Requirement ID</b>    | R-18  |
| <b>Requirement Name</b>  | View Report Details   |
| <b>Actors</b>            | Manager, Analyst, Executive   |
| <b>Preconditions</b>     | 1. Report exists  |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. User opens a specific report</li> <li>2. System displays report visualization (charts)</li> <li>3. System displays report metadata (date, source, title)</li> </ol> |
| <b>Alternative Flows</b> | <p>A1 – Visualization Not Available</p> <ol style="list-style-type: none"> <li>1. System displays warning message</li> </ol>  |
| <b>Postconditions</b>    | 1. Report details are visible   |

Table 24 : Use Case Specification for "View Report Details"

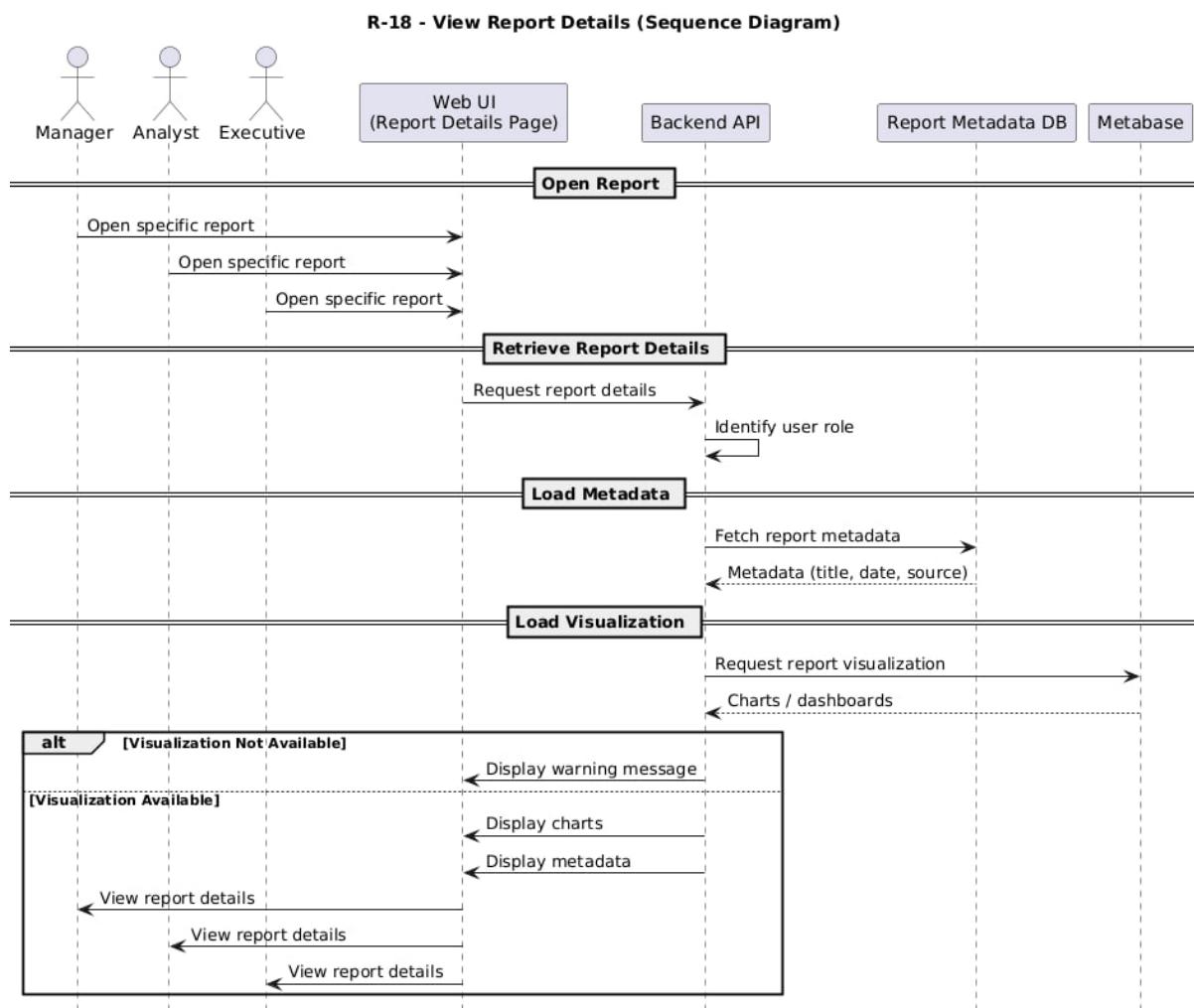


Figure 19 : Sequence Diagram for "View Reports Details"

- Edit SQL Query and Re-execute Rep

|                          |   |
|--------------------------|---|
| <b>Requirement ID</b>    | R-19  |
| <b>Requirement Name</b>  | Edit SQL Query and Re-execute Report  |
| <b>Actors</b>            | Analyst   |
| <b>Preconditions</b>     | 1. SQL query exists   |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. Analyst edits SQL query</li> <li>2. Analyst clicks “Execute”</li> <li>3. System validates SQL</li> <li>4. System executes query on ClickHouse</li> <li>5. System updates result in Metabase</li> <li>6. System refreshes dashboard</li> </ol> |
| <b>Alternative Flows</b> | A1 – SQL Error<br>1. System shows error details   |
| <b>Postconditions</b>    | 1. Report is updated  |

Table 25 : Use Case Specification for “Edit SQL Query”

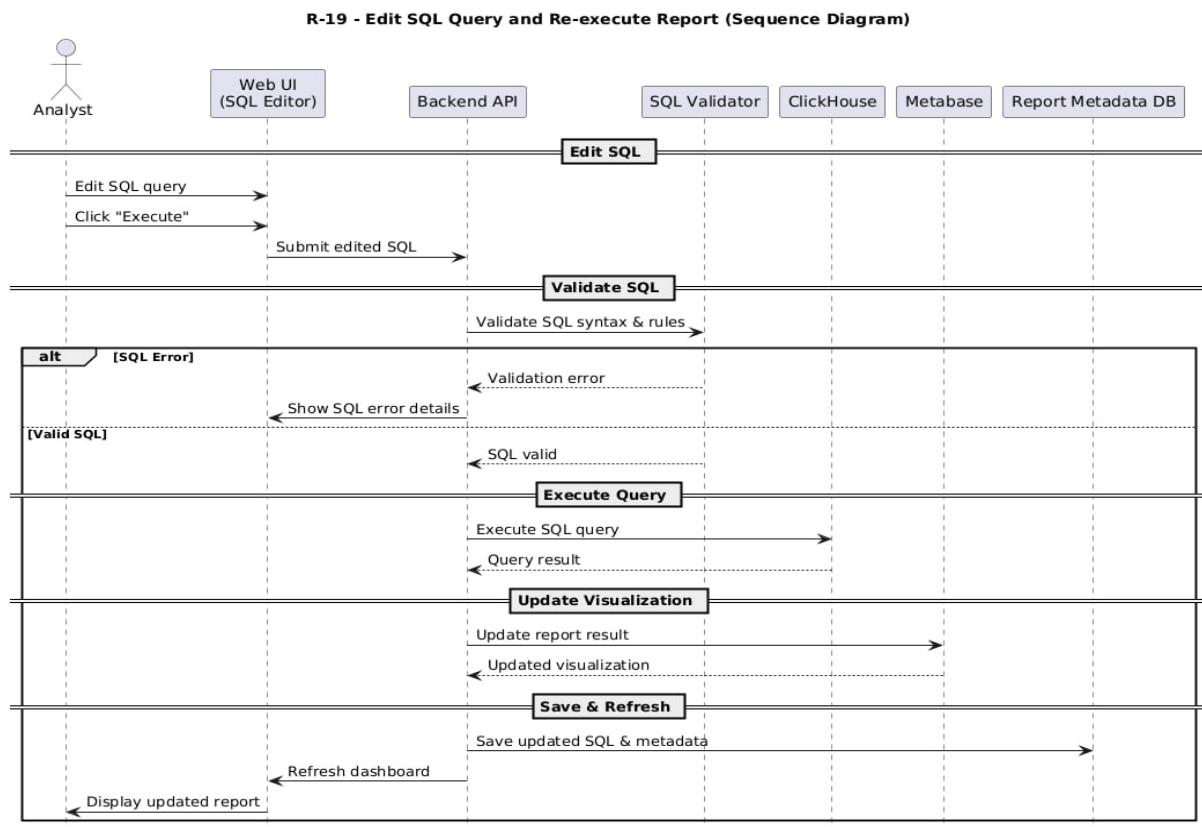


Figure 20 : Sequence Diagram for "Edit SQL"

- **Edit Transcription and Regenerate Report**

|                          |   |
|--------------------------|---|
| <b>Requirement ID</b>    | R-20  |
| <b>Requirement Name</b>  | Edit Transcription and Regenerate Report  |
| <b>Actors</b>            | Analyst   |
| <b>Preconditions</b>     | 1. Transcription exists   |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. Analyst edits transcription text</li> <li>2. Analyst clicks “Regenerate”</li> <li>3. System sends updated text to LLM</li> <li>4. System generates new SQL query</li> <li>5. System executes query on ClickHouse</li> <li>6. System updates Metabase visualization</li> </ol> |
| <b>Alternative Flows</b> | <p>A1 – Intent Not Clear</p> <ol style="list-style-type: none"> <li>1. System asks for clarification</li> </ol>   |
| <b>Postconditions</b>    | 1. Updated report is saved  |

Table 26 : Use Case Specification for “Edit Transcription”

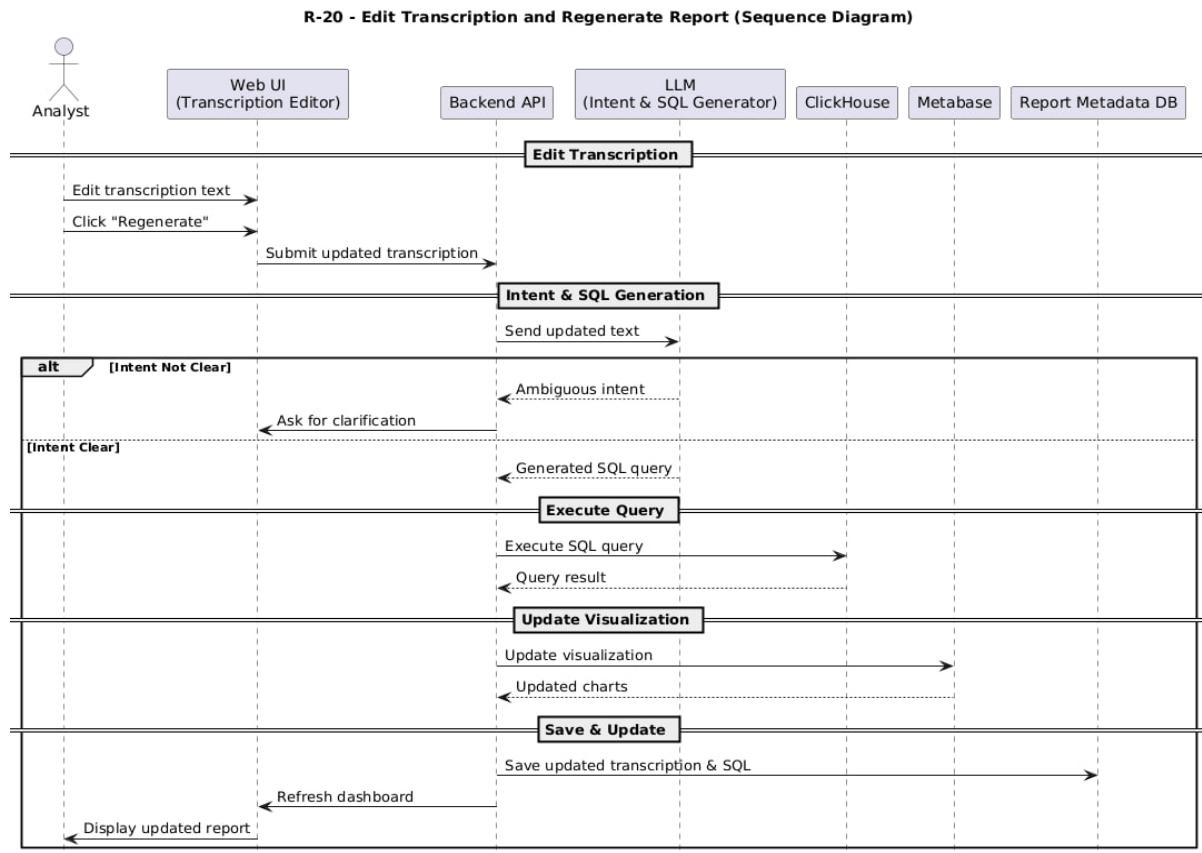


Figure 21: Sequence Diagram for "Edit Transcription"

## • Export Report as Image

|                          |  |
|--------------------------|--|
| <b>Requirement ID</b>    | <b>R-21</b>  |
| <b>Requirement Name</b>  | Export Report as Image   |
| <b>Actors</b>            | Manager, Analyst, Executive  |
| <b>Preconditions</b>     | 1. Report exists   |
| <b>Main Flow</b>         | 1. User selects “Export as Image” option<br>2. System requests image export from Metabase<br>3. Metabase generates report image<br>4. System downloads image file to user device |
| <b>Alternative Flows</b> | A1 – Export Failed<br>1. System displays error message   |
| <b>Postconditions</b>    | 1. Report image is saved locally   |

Table 27 : Use Case Specification for “Export Report”

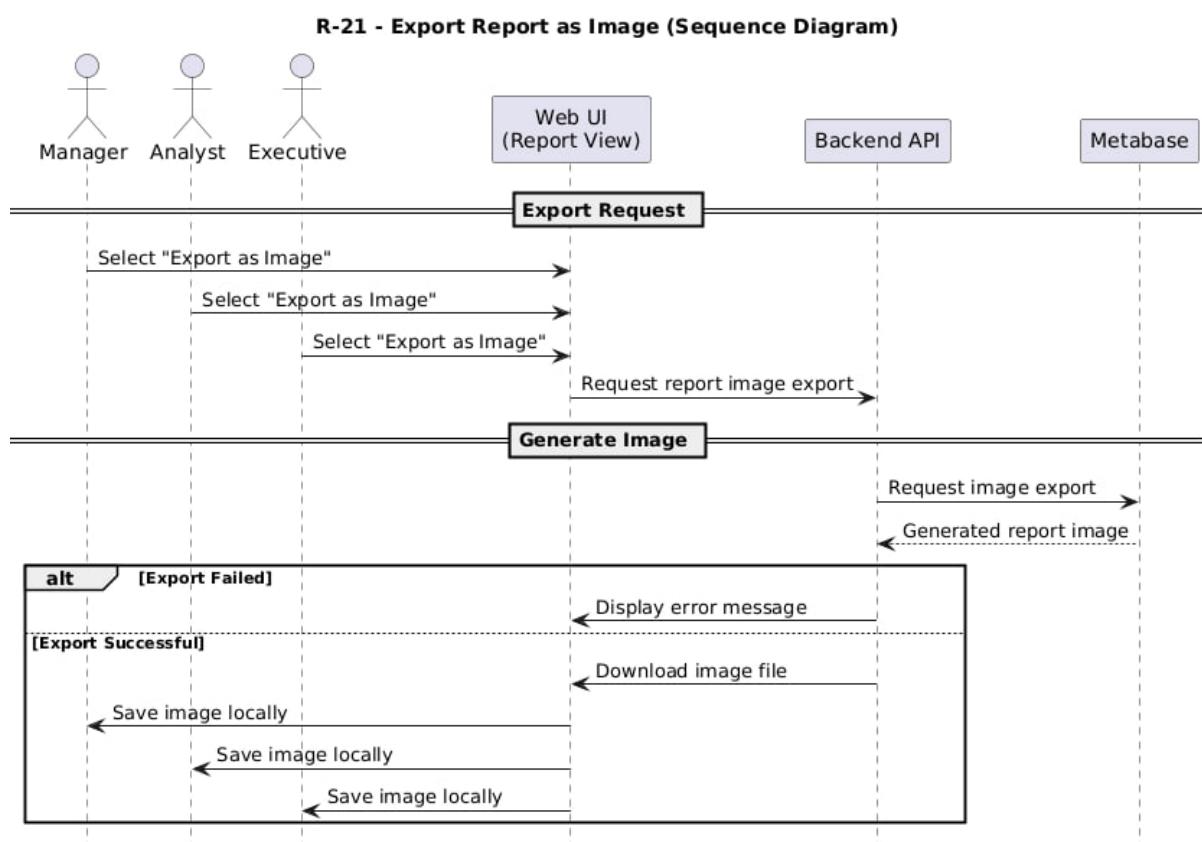


Figure 22 : Sequence Diagram for " Export Report"

- Delete Report

|                          |  |
|--------------------------|--|
| <b>Requirement ID</b>    | R-22   |
| <b>Requirement Name</b>  | Delete Report  |
| <b>Actors</b>            | Manager  |
| <b>Preconditions</b>     | 1. Report exists   |
| <b>Main Flow</b>         | 1. Manager selects report<br>2. System displays confirmation dialog<br>3. Manager confirms deletion<br>4. System deletes report from Metabase and metadata store |
| <b>Alternative Flows</b> | A1 – Deletion Cancelled<br>1. System keeps report  |
| <b>Postconditions</b>    | 1. Report is deleted   |

Table 28 : Use Case Specification for "Delete Report"

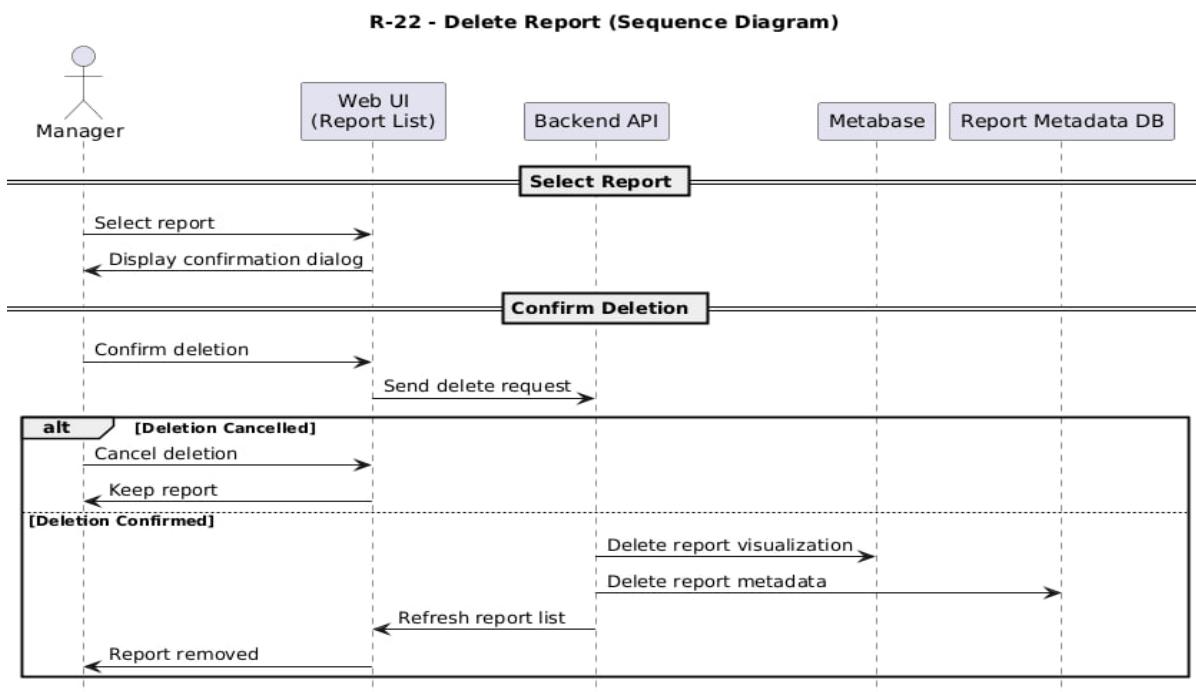


Figure 23: Sequence Diagram for "Delete Report"

- **Upload Database and Execute ETL Pipeline**

|                          |  |
|--------------------------|--|
| <b>Requirement ID</b>    | <b>R-23</b>  |
| <b>Requirement Name</b>  | Upload Database and Execute ETL Pipeline   |
| <b>Actors</b>            | Manager  |
| <b>Preconditions</b>     | 1. Manager is logged in<br>2. Database file exists   |
| <b>Main Flow</b>         | 1. Manager opens Database Management page<br>2. System displays database upload interface<br>3. Manager uploads database file<br>4. System validates file format and schema<br>5. System starts ETL process<br>6. System extracts data from uploaded database (Extract)<br>7. System cleans and transforms data (Transform)<br>8. System loads processed data into ClickHouse (Load)<br>9. System verifies data loading success<br>10. System updates metadata and available datasets<br>11. System displays successful ETL completion message |
| <b>Alternative Flows</b> | A1 – ETL Failure<br>1. System detects ETL error<br>2. System logs error details<br>3. System displays failure message  |
| <b>Postconditions</b>    | 1. Database is loaded into ClickHouse<br>2. Data is available for reporting<br>3. ETL status is stored   |

*Table 29 : Use Case Specification for “Upload Database”*

R-23 - Upload Database and Execute ETL Pipeline (Sequence Diagram)

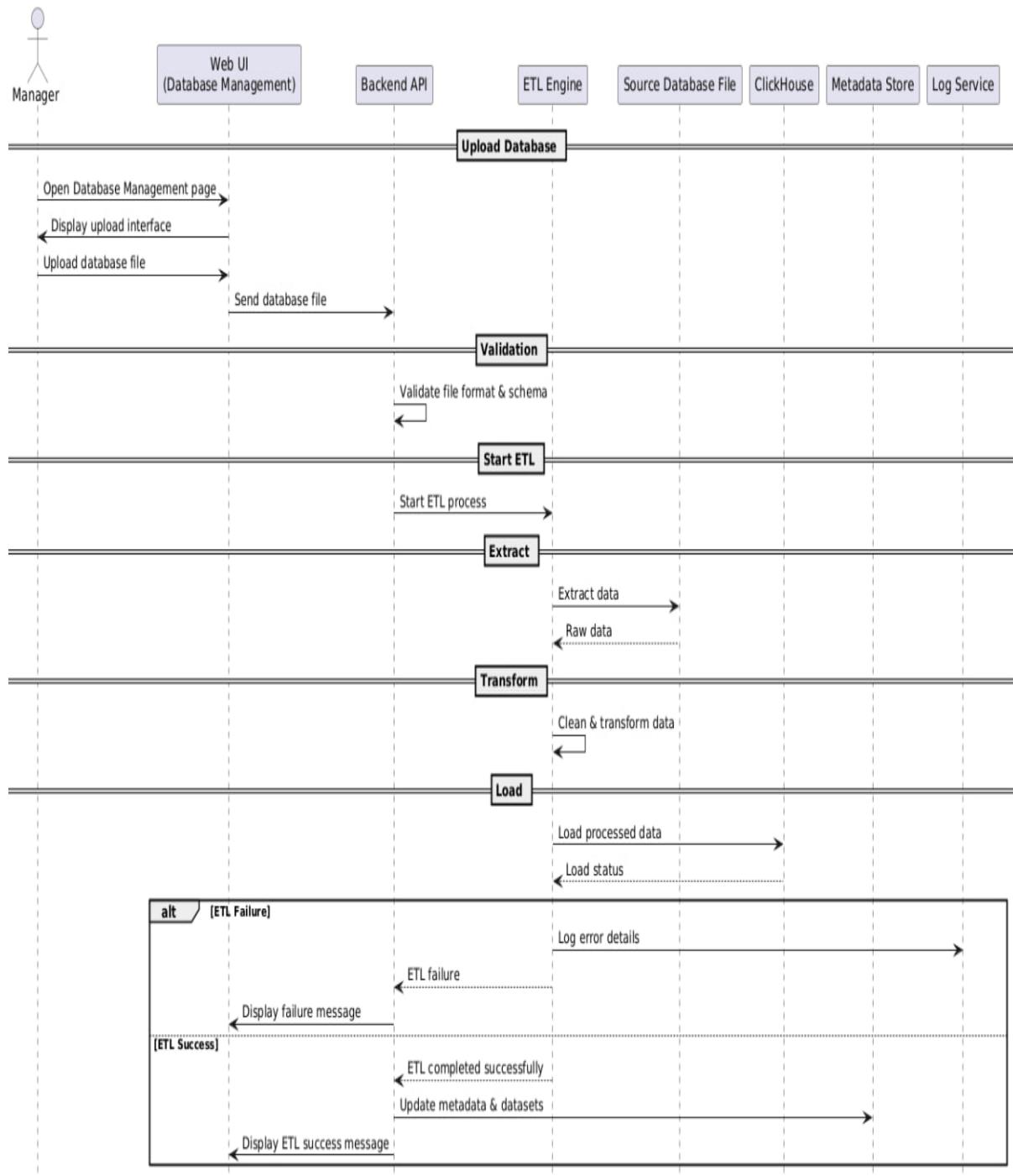


Figure 24 : Sequence Diagram for " Upload database"

- **Delete Database from ClickHouse**

|                          |  |
|--------------------------|--|
| <b>Requirement ID</b>    | R-24   |
| <b>Requirement Name</b>  | Delete Database from ClickHouse  |
| <b>Actors</b>            | Manager  |
| <b>Preconditions</b>     | <ol style="list-style-type: none"> <li>1. Manager is logged in</li> <li>2. Database exists in ClickHouse</li> </ol>  |
| <b>Main Flow</b>         | <ol style="list-style-type: none"> <li>1. Manager opens Database Management page</li> <li>2. System displays list of databases</li> <li>3. Manager selects a database to delete</li> <li>4. System displays warning message explaining impact</li> <li>5. Manager confirms deletion</li> <li>6. System connects to ClickHouse</li> <li>7. System executes database deletion command</li> <li>8. System verifies deletion success</li> <li>9. System removes related metadata and reports</li> <li>10. System displays success message</li> </ol> |
| <b>Alternative Flows</b> | <p>A1 – Deletion Failed</p> <ol style="list-style-type: none"> <li>1. System detects deletion error</li> <li>2. System displays error message</li> </ol>   |
| <b>Postconditions</b>    | <ol style="list-style-type: none"> <li>1. Database is removed from ClickHouse</li> <li>2. Related reports become unavailable</li> <li>3. Metadata is updated</li> </ol>  |

*Table 30 : Use Case Specification for "Delete Database"*

### R-24 - Delete Database from ClickHouse (Sequence Diagram)

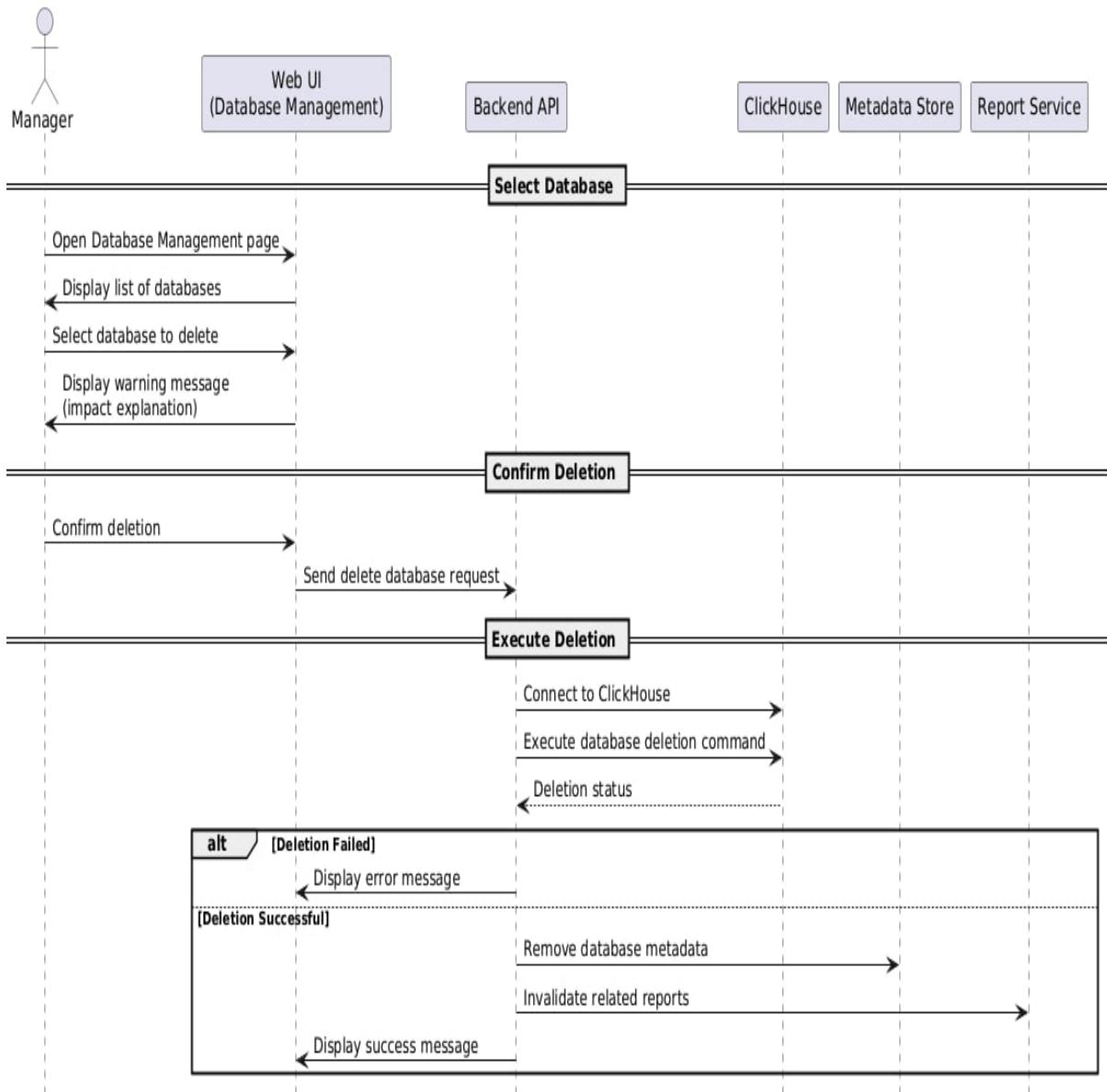


Figure 25 : Sequence Diagram for "Delete Database"

## 6. System architecture

- Class Diagram

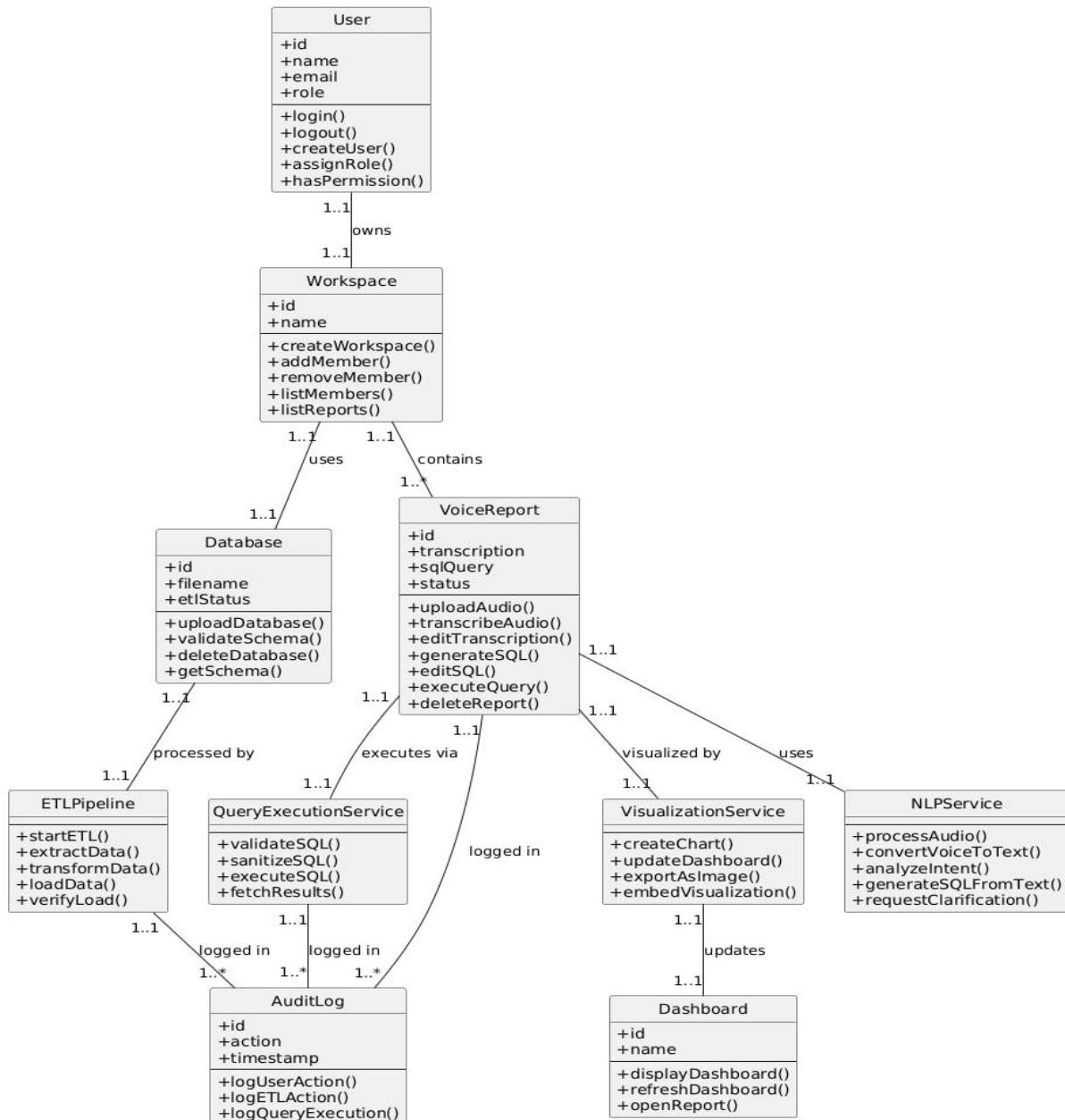


Figure 26 : Class Diagram

## • ERD Diagram

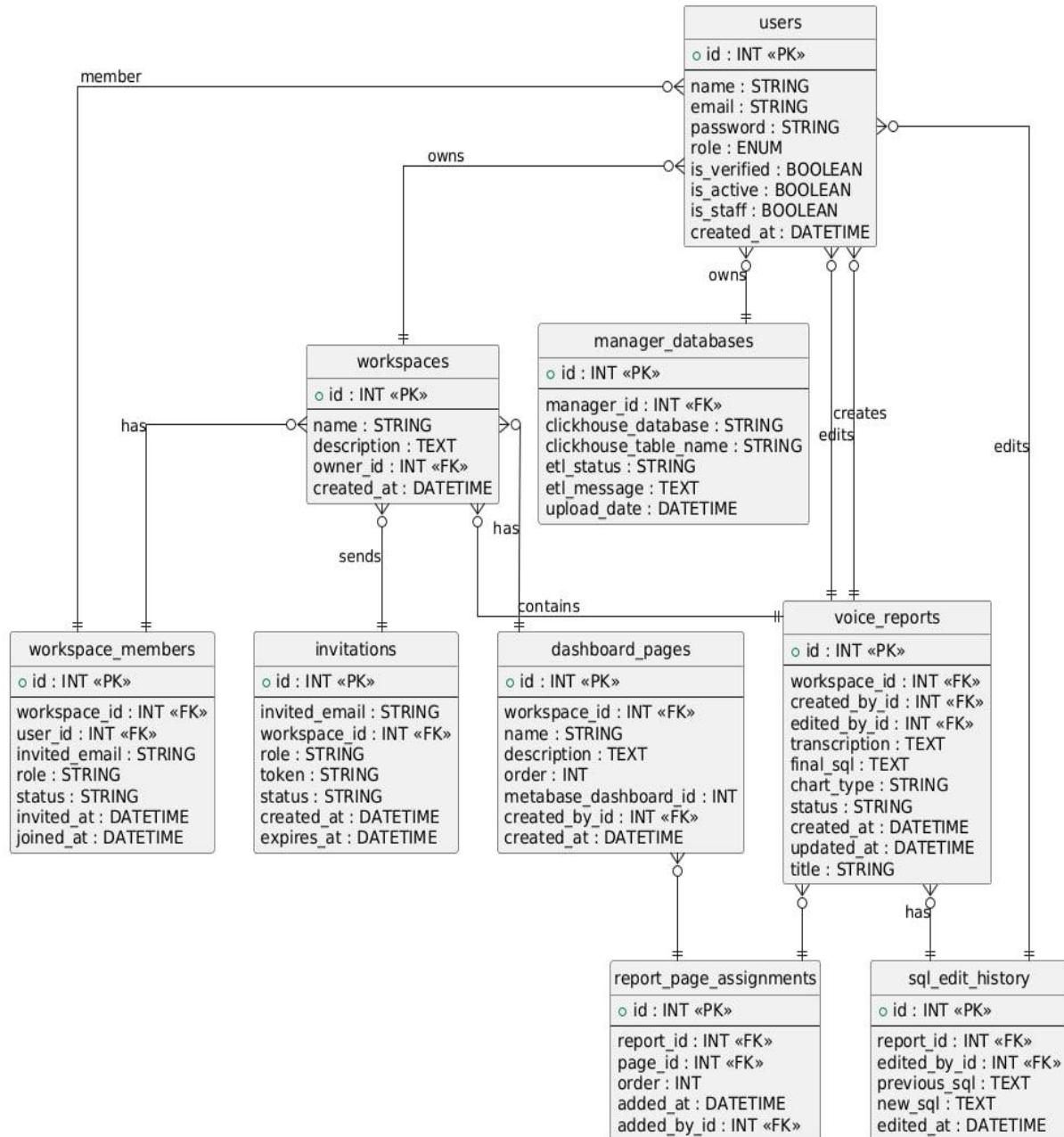


Figure 27 : ERD Diagram

## 7. Initial Test Cases

- Test cases for User Registration

| Test Case ID | Requirement Name  | Description                                      | Preconditions                                   | Test Steps   | Expected Result  | Test Type | Priority |
|--------------|-------------------|--|---|--|--|-----------|----------|
| TC-REG-001   | User Registration | Valid user registration with all required fields | No user account exists with provided email      | 1. Navigate to signup page<br>2. Enter valid name (min 3 chars)<br>3. Enter unique email<br>4. Enter password (min 8 chars with uppercase/lowercase/number)<br>5. Select role (Manager/Analyst/Executive)<br>6. Click Register | User account created successfully; Verification email sent; User redirected to email verification page; Account status is unverified | Positive  | High     |
| TC-REG-002   | User Registration | Registration with duplicate email address        | User account already exists with provided email | 1. Navigate to signup page<br>2. Enter name<br>3. Enter existing email<br>4. Enter password<br>5. Select role<br>6. Click Register   | Registration fails; Error message displayed: "Email already exists"; User remains on signup page; No duplicate account created       | Negative  | High     |
| TC-REG-003   | User Registration | Registration with invalid email format           | No user account exists                          | 1. Navigate to signup page<br>2. Enter valid name<br>3. Enter invalid email (missing @ or domain)<br>4. Enter password<br>5. Select role<br>6. Click Register  | Registration fails; Error message: "Invalid email format"; Email field highlighted with validation error                             | Negative  | High     |
| TC-REG-004   | User Registration | Registration with weak password                  | No user account exists                          | 1. Navigate to signup page<br>2. Enter valid name<br>3. Enter unique email   | Registration fails; Error message:   | Negative  | High     |

|                   |                   |  |                        |  |   |          |      |
|-------------------|-------------------|--|------------------------|--|---|----------|------|
|                   |                   |  |                        | 4. Enter weak password (less than 8 chars or missing uppercase/number)<br>5. Select role<br>6. Click Register  | “Password must be at least 8 characters with uppercase, lowercase, and number”; Password field highlighted  |          |      |
| <b>TC-REG-005</b> | User Registration | Manager auto-creates workspace on signup | No user account exists | 1. Navigate to signup page<br>2. Enter valid name<br>3. Enter unique email<br>4. Enter valid password<br>5. Select role as Manager<br>6. Click Register<br>7. Verify email | User account created successfully; Workspace automatically created; User assigned as workspace owner; Workspace name defaults to “{UserName}'s Workspace”; User can access workspace after verification | Positive | High |

Table 31 : Test cases for User Registration

- Test Cases for User Login

| Test Case ID       | Requirement Name | Description  | Preconditions                         | Test Steps  | Expected Result  | Test Type | Priority |
|--------------------|------------------|--|---------------------------------------|---|--|-----------|----------|
| <b>TC-PASS-001</b> | Password Reset   | Request password reset with valid registered email | User account exists and is verified   | 1. Navigate to Forgot Password page<br>2. Enter registered email<br>3. Click Reset Password   | Password reset email sent successfully; Confirmation message displayed | Positive  | High     |
| <b>TC-PASS-002</b> | Password Reset   | Request password reset with non-existent email     | No account exists with provided email | 1. Navigate to Forgot Password page<br>2. Enter non-existent email<br>3. Click Reset Password | Request rejected; Error message: “Email not found”; No email sent      | Negative  | High     |
| <b>TC-PASS-003</b> | Password Reset   | Reset password using valid token                   | Valid reset token received via email  | 1. Open reset password link from email  | Password updated successfully; User redirected to login page; Old      | Positive  | High     |

|                    |                |   |                             |   |   |          |      |
|--------------------|----------------|---|-----------------------------|---|---|----------|------|
|                    |                |   |                             | 2. Enter new valid password<br>3. Confirm password<br>4. Submit                 | password invalidated  |          |      |
| <b>TC-PASS-004</b> | Password Reset | Reset password using expired or invalid token | Token is expired or invalid | 1. Open reset link with expired/invalid token<br>2. Attempt to set new password | Reset fails; Error message: "Invalid or expired token"; User prompted to request new reset link | Negative | High |
| <b>TC-PASS-005</b> | Password Reset | Reset password with weak new password         | Valid reset token exists    | 1. Open reset password link<br>2. Enter weak password<br>3. Submit              | Reset fails; Password validation error shown; Password not updated                              | Negative | High |

Table 32 : Test Cases for User Login

- Test Cases for Email Verification

| Test Case ID        | Requirement Name   | Description                                      | Preconditions   | Test Steps   | Expected Result   | Test Type | Priority |
|---------------------|--------------------|--|---|--|---|-----------|----------|
| <b>TC-EMAIL-001</b> | Email Verification | Valid email verification with correct token      | User registered but not verified; Valid verification token sent | 1. Receive verification email<br>2. Click verification link with token<br>3. System verifies token | Email verification successful; User is_verified = True; User can now login; Confirmation message displayed                              | Positive  | High     |
| <b>TC-EMAIL-002</b> | Email Verification | Email verification with expired token            | Verification token has expired (>24 hours)                      | 1. Receive verification email<br>2. Wait more than 24 hours<br>3. Click verification link          | Verification fails; Error message: "Verification link expired"; User redirected to request new verification; Account remains unverified | Negative  | High     |
| <b>TC-EMAIL-003</b> | Email Verification | Email verification with invalid token            | Verification token is malformed or tampered                     | 1. Receive verification email<br>2. Modify token in URL<br>3. Click modified link                  | Verification fails; Error message: "Invalid verification link"; User redirected to signup/login; No verification occurs                 | Negative  | High     |
| <b>TC-EMAIL-004</b> | Email Verification | Email verification with already verified account | User account is already verified                                | 1. User already verified<br>2. Click verification link again                                       | Verification succeeds; Message displayed: "Your account is already verified"; No duplicate verification performed                       | Positive  | High     |
| <b>TC-EMAIL-005</b> | Email Verification | Email verification updates                       | Invited user verifies email                                     | 1. Invited user receives verification email  | Email verified; WorkspaceMember status changes from   | Positive  | High     |

|  |  |                         |  |   |  |  |  |
|--|--|-------------------------|--|---|--|--|--|
|  |  | Workspace Member status |  | 2. Click verification link<br>3. Email verified | pending_acceptance to active; joined_at timestamp set; User gains workspace access |  |  |
|--|--|-------------------------|--|---|--|--|--|

Table 33 : Test Cases for Email Verification

- Test Cases for Authentication and Token Management

| Test Case ID          | Requirement Name | Description                              | Preconditions  | Test Steps   | Expected Result  | Test Type | Priority |
|-----------------------|------------------|--|--|--|--|-----------|----------|
| <b>TC-LOG OUT-001</b> | User Logout      | Successful user logout                   | User is logged in with valid access and refresh tokens | 1. User clicks Logout<br>2. System sends logout request with refresh token | Logout successful; Refresh token blacklisted; User session terminated; User redirected to login page     | Positive  | High     |
| <b>TC-LOGOUT-002</b>  | User Logout      | Logout with already expired access token | Access token expired; Refresh token still valid        | 1. Access token expires<br>2. User clicks Logout                           | Logout succeeds; Refresh token blacklisted; No error displayed; User redirected to login                 | Positive  | Medium   |
| <b>TC-LOGOUT-003</b>  | User Logout      | Logout with invalid refresh token        | Refresh token is invalid or tampered                   | 1. User attempts logout with invalid refresh token                         | Logout fails; Error message: "Invalid token"; No token blacklisted; User prompted to re-login            | Negative  | High     |
| <b>TC-LOGOUT-004</b>  | User Logout      | Logout from multiple active sessions     | User logged in on multiple devices                     | 1. User logs out from one device   | Logout affects only current session; Other sessions remain active unless global logout is enabled        | Positive  | Medium   |
| <b>TC-LOGOUT-005</b>  | User Logout      | Logout clears client-side session data   | User is logged in                                      | 1. User clicks Logout  | Access token removed from client storage; Refresh token removed; No authenticated data remains on client | Positive  | High     |

Table 34 : Test Cases for Authentication and Token Management

- Test Cases for User Management by Admin

| Test Case ID        | Requirement Name         | Description                       | Preconditions                                    | Test Steps  | Expected Result  | Test Type | Priority |
|---------------------|--------------------------|-----------------------------------|--|---|--|-----------|----------|
| <b>TC-ADMIN-001</b> | User Management by Admin | Admin creates a new user account  | Admin is logged in and has required permissions  | 1. Login as Admin<br>2. Navigate to User Management<br>3. Click “Create User”<br>4. Enter user details (name, email, role)<br>5. Click Save | User account created successfully; Invitation/verification email sent; User appears in user list with correct role | Positive  | High     |
| <b>TC-ADMIN-002</b> | User Management by Admin | Admin edits existing user details | User account exists                              | 1. Login as Admin<br>2. Navigate to User Management<br>3. Select an existing user<br>4. Modify user details (name/role)<br>5. Click Save    | User details updated successfully; Changes reflected immediately; Update logged in audit trail                     | Positive  | High     |
| <b>TC-ADMIN-003</b> | User Management by Admin | Admin deactivates a user account  | User account exists and is active                | 1. Login as Admin<br>2. Navigate to User Management<br>3. Select user<br>4. Click Deactivate  | User account deactivated; User cannot login; Status updated to inactive; Action logged in audit log                | Positive  | High     |
| <b>TC-ADMIN-004</b> | User Management by Admin | Deactivated user attempts login   | User account is deactivated                      | 1. Navigate to login page<br>2. Enter deactivated user credentials  | Login fails; Error message: “Your account has been deactivated”; No tokens issued                                  | Negative  | High     |
| <b>TC-ADMIN-005</b> | User Management by Admin | Admin deletes a user account      | User account exists; Admin has delete permission | 1. Login as Admin<br>2. Navigate to User Management<br>3. Select user<br>4. Click Delete  | User account deleted successfully; User removed from system; Related sessions invalidated;                         | Positive  | High     |

|  |  |  |  |                     |                              |  |  |
|--|--|--|--|---------------------|------------------------------|--|--|
|  |  |  |  | 5. Confirm deletion | Action recorded in audit log |  |  |
|--|--|--|--|---------------------|------------------------------|--|--|

Table 35 : Test Cases for User Management by Admin

- Test Cases for Audit Logs and Activity Tracking

| Test Case ID  | Requirement Name               | Description                                | Precondition s                      | Test Steps   | Expected Result   | Test Type | Priority |
|---------------|--------------------------------|--|-------------------------------------|--|---|-----------|----------|
| TC-AUD IT-001 | Audit Logs & Activity Tracking | Log user login activity                    | User account exists and is verified | 1. Login with valid credentials  | Login action recorded in audit log; Log includes user_id, action=LOGIN, timestamp, IP address | Positive  | High     |
| TC-AUD IT-002 | Audit Logs & Activity Tracking | Log failed login attempt                   | User account exists                 | 1. Navigate to login page<br>2. Enter valid email<br>3. Enter incorrect password<br>4. Click Login | Failed login attempt recorded;<br>Action=FAILED_LOGIN; Timestamp and IP stored                | Positive  | High     |
| TC-AUD IT-003 | Audit Logs & Activity Tracking | Log user invitation action                 | Manager invites a new user          | 1. Login as Manager<br>2. Send invitation to new user  | Invitation action logged;<br>Log includes inviter_id, invitee_email, role, workspace_id       | Positive  | High     |
| TC-AUD IT-004 | Audit Logs & Activity Tracking | Log workspace update action                | Manager updates workspace settings  | 1. Login as Manager<br>2. Update workspace name or description                                     | Workspace update recorded in audit log;<br>Old and new values stored; Timestamp recorded      | Positive  | High     |
| TC-AUD IT-005 | Audit Logs & Activity Tracking | Admin views audit logs                     | Admin is logged in with permission  | 1. Login as Admin<br>2. Navigate to Audit Logs page  | Audit logs displayed successfully; Logs are sortable and filterable by date, user, and action | Positive  | High     |
| TC-AUD IT-006 | Audit Logs & Activity Tracking | Unauthorized user cannot access audit logs | User is Analyst or Executive        | 1. Login as Analyst/Executive<br>2. Attempt to access Audit Logs                                   | Access denied; Error message: "Insufficient permissions"; No logs displayed                   | Negative  | High     |

Table 36 : Test Cases for Audit Logs and Activity Tracking

- Test Cases for Accepting Workspace Invitations

| Test Case ID   | Requirement Name            | Description                               | Preconditions   | Test Steps  | Expected Result   | Test Type | Priority |
|----------------|-----------------------------|---|---|---|---|-----------|----------|
| TC-ACCEP-T-001 | Accept Workspace Invitation | Existing user accepts invitation          | User has existing account; Valid invitation token received  | 1. Receive invitation email<br>2. Click acceptance link with token<br>3. System validates token<br>4. User logs in                                    | Invitation accepted; WorkspaceMember status changes to active; User added to workspace; Role assigned; User can access workspace after login                | Positive  | High     |
| TC-ACCEP-T-002 | Accept Workspace Invitation | New user accepts invitation and registers | User does not have account; Valid invitation token received | 1. Receive invitation email<br>2. Click acceptance link with token<br>3. System redirects to signup<br>4. User registers with provided email and role | Invitation accepted; User account created; WorkspaceMember created with status=active; User can access workspace; Email and role pre-filled from invitation | Positive  | High     |
| TC-ACCEP-T-003 | Accept Workspace Invitation | Invitation acceptance with expired token  | Invitation token has expired (>48 hours)                    | 1. Receive invitation email<br>2. Wait more than 48 hours<br>3. Click acceptance link   | Acceptance fails; Error message: "Invitation link expired"; User redirected to request new invitation; No workspace access granted                          | Negative  | High     |
| TC-ACCEP-T-004 | Accept Workspace Invitation | Invitation acceptance with invalid token  | Invitation token is malformed or tampered                   | 1. Receive invitation email<br>2. Modify token in URL<br>3. Click modified link   | Acceptance fails; Error message: "Invalid invitation link"; User redirected; No workspace access granted  | Negative  | High     |
| TC-ACCEP-T-005 | Accept Workspace Invitation | Cannot accept already                     | Invitation status is already accepted                       | 1. Accept invitation<br>2. Attempt to accept same invitation again  | Acceptance fails; Error message: "Invitation already accepted"; User remains in workspace; No   | Negative  | High     |

|  |  |                     |  |  |                              |  |  |
|--|--|---------------------|--|--|------------------------------|--|--|
|  |  | accepted invitation |  |  | duplicate membership created |  |  |
|--|--|---------------------|--|--|------------------------------|--|--|

Table 37 : Test Cases for Accepting Workspace Invitations

- Test Cases for Permission and Authorization Control

| Test Case ID | Requirement Name                   | Description                            | Preconditions                                   | Test Steps   | Expected Result  | Test Type | Priority |
|--------------|------------------------------------|--|---|--|--|-----------|----------|
| TC-PER M-001 | Permission & Authorization Control | Manager can view all workspace members | Manager is verified and workspace has members   | 1. Login as Manager<br>2. Navigate to Members page<br>3. View member list  | All members displayed (accepted, pending, invited); Manager sees all statuses; Member details include name, email, role, status; List is complete and accurate | Positive  | High     |
| TC-PER M-002 | Permission & Authorization Control | Analyst can only view active members   | Analyst is verified member of workspace         | 1. Login as Analyst<br>2. Navigate to Members page<br>3. View member list  | Only active members displayed; Pending and invited members hidden; No member management actions available; Read-only access                                    | Positive  | High     |
| TC-PER M-003 | Permission & Authorization Control | Manager can assign roles to members    | Manager is verified; Member exists in workspace | 1. Login as Manager<br>2. Navigate to Members<br>3. Select member<br>4. Click “Change Role”<br>5. Select new role (Analyst/Executive)<br>6. Click Save | Role updated successfully; Role updated in User and WorkspaceMember; Changes reflected immediately; Audit log recorded; Member notified                        | Positive  | High     |
| TC-PER M-004 | Permission & Authorization Control | Analyst cannot assign roles            | User is Analyst                                 | 1. Login as Analyst<br>2. Navigate to Members<br>3. Attempt to change another member’s role  | Access denied; Role change option hidden or disabled; Error message: “Only managers can assign roles”; No changes applied                                      | Negative  | High     |

|                     |                                    |                            |  |  |  |          |      |
|---------------------|------------------------------------|----------------------------|--|--|--|----------|------|
| <b>TC-PER-M-005</b> | Permission & Authorization Control | Manager cannot demote self | Manager is the only manager in workspace | 1. Login as Manager<br>2. Navigate to Members<br>3. Select own profile<br>4. Attempt to change own role to Analyst | Role change blocked;<br>Error message: "Cannot change your own role";<br>Manager role remains unchanged; Workspace must always have at least one manager | Negative | High |
|---------------------|------------------------------------|----------------------------|--|--|--|----------|------|

Table 38 : Test Cases for Permission and Authorization Control

- Test Cases for Role & Permission Enforcement (Advanced Scenarios)

| Test Case ID       | Requirement Name              | Description                                      | Preconditions                             | Test Steps  | Expected Result  | Test Type | Priority |
|--------------------|-------------------------------|--|---|---|--|-----------|----------|
| <b>TC-PERM-006</b> | Role & Permission Enforcement | Executive has read-only access to workspace data | Executive is verified member of workspace | 1. Login as Executive<br>2. Navigate to Members page<br>3. Attempt to edit member details | Members list visible;<br>No edit/delete actions available; Read-only access enforced                             | Positive  | High     |
| <b>TC-PERM-007</b> | Role & Permission Enforcement | Analyst cannot invite new members                | Analyst is verified member of workspace   | 1. Login as Analyst<br>2. Navigate to Members<br>3. Attempt to invite new member          | Access denied; Invite button hidden or disabled; Error message displayed   | Negative  | High     |
| <b>TC-PERM-008</b> | Role & Permission Enforcement | Executive cannot invite new members              | Executive is verified member of workspace | 1. Login as Executive<br>2. Navigate to Members<br>3. Attempt to invite new member        | Access denied; Invite functionality unavailable; No invitation sent  | Negative  | High     |
| <b>TC-PERM-009</b> | Role & Permission Enforcement | Manager can remove workspace members             | Manager is verified; Target member exists | 1. Login as Manager<br>2. Navigate to Members<br>3. Select member<br>4. Click Remove      | Member removed successfully;<br>WorkspaceMember marked as removed;<br>User loses workspace access; Action logged | Positive  | High     |
| <b>TC-PERM-010</b> | Role & Permission Enforcement | Analyst cannot remove workspace members          | Analyst is verified member                | 1. Login as Analyst<br>2. Navigate to Members<br>3. Attempt to remove a member            | Access denied; Remove option hidden; Error message displayed; No changes applied                                 | Negative  | High     |

Table 39 : Test Cases for Role & Permission Enforcement (Advanced Scenarios)

- Test Cases for Member Suspension Management

| Test Case ID    | Requirement Name           | Description                         | Preconditions                            | Test Steps   | Expected Result   | Test Type | Priority |
|-----------------|----------------------------|-------------------------------------|--|--|---|-----------|----------|
| TC-SUSPE ND-001 | Suspend & Unsuspend Member | Manager suspends active member      | Manager is verified; Member is active    | 1. Login as Manager<br>2. Navigate to Members<br>3. Select member<br>4. Click “Suspend”              | Member suspended; is_active = False; Member cannot login; Member cannot access workspace; Status shows “suspended”                      | Positive  | High     |
| TC-SUSPE ND-002 | Suspend & Unsuspend Member | Suspended member cannot login       | Member is suspended (is_active = False)  | 1. Attempt to login with suspended member credentials<br>2. Enter correct password<br>3. Click Login | Login fails; Error message: “Your account has been suspended”; No tokens issued; Member cannot access system                            | Negative  | High     |
| TC-SUSPE ND-003 | Suspend & Unsuspend Member | Manager unsuspends suspended member | Member is suspended                      | 1. Login as Manager<br>2. Navigate to Members<br>3. Select suspended member<br>4. Click “Unsuspend”  | Member unsuspended; is_active = True; Member can login again; Workspace access restored; Status shows “active”                          | Positive  | High     |
| TC-SUSPE ND-004 | Suspend & Unsuspend Member | Analyst cannot suspend members      | User is Analyst                          | 1. Login as Analyst<br>2. Navigate to Members<br>3. Attempt to suspend another member                | Access denied; Suspend button hidden or disabled; Error message: “Only managers can suspend members”; No suspension occurs              | Negative  | High     |
| TC-SUSPE ND-005 | Suspend & Unsuspend Member | Manager cannot suspend self         | Manager is the only manager in workspace | 1. Login as Manager<br>2. Navigate to Members<br>3. Select own profile<br>4. Click “Suspend”         | Suspension blocked; Error message: “You cannot suspend yourself”; Manager remains active; Workspace retains at least one active manager | Negative  | High     |

Table 40 : Test Cases for Member Suspension Management

- Test Cases for Data Processing (ETL Pipeline)

| Test Case ID | Requirement Name               | Description  | Preconditions  | Test Steps  | Expected Result  | Test Type | Priority |
|--------------|--------------------------------|--|--|---|--|-----------|----------|
| TC-ETL-001   | Data Processing (ETL Pipeline) | Successful ETL pipeline execution on uploaded file | Valid CSV file uploaded; ETL services running                        | 1. Upload valid CSV file<br>2. Trigger ETL process automatically<br>3. Monitor ETL status | ETL pipeline executes successfully; Data extracted, transformed, and loaded; Status changes to “completed”; Records inserted into database | Positive  | High     |
| TC-ETL-002   | Data Processing (ETL Pipeline) | ETL fails due to invalid data values               | CSV uploaded with invalid data (e.g., non-numeric in numeric column) | 1. Upload CSV with invalid values<br>2. Trigger ETL process                               | ETL fails; Error logged with details; Status set to “failed”; Invalid rows reported; No partial load committed                             | Negative  | High     |
| TC-ETL-003   | Data Processing (ETL Pipeline) | ETL retry after fixing data errors                 | Previous ETL attempt failed  | 1. Fix CSV file<br>2. Re-upload corrected file<br>3. Re-run ETL                           | ETL completes successfully; Status updated to “completed”; Corrected data loaded; Previous failed attempt archived                         | Positive  | High     |
| TC-ETL-004   | Data Processing (ETL Pipeline) | ETL handles empty CSV file                         | CSV file contains headers only, no data rows                         | 1. Upload empty CSV file<br>2. Trigger ETL process  | ETL completes with warnings; No records loaded; Warning message displayed; System remains stable   | Negative  | Medium   |

|                   |                                |                                      |                              |   |   |          |      |
|-------------------|--------------------------------|--------------------------------------|------------------------------|---|---|----------|------|
| <b>TC-ETL-005</b> | Data Processing (ETL Pipeline) | Unauthorized user cannot trigger ETL | User is Analyst or Executive | 1. Login as Analyst/Executive<br>2. Attempt to trigger ETL on uploaded file | Access denied; Error message: “Only managers can process data”; ETL not started | Negative | High |
|-------------------|--------------------------------|--------------------------------------|------------------------------|---|---|----------|------|

Table 41 : Test Cases for Data Processing (ETL Pipeline)

- Test Cases for ETL Pipeline Execution (Detailed Stages)

| Test Case ID      | Requirement Name       | Description                             | Preconditions                                     | Test Steps  | Expected Result  | Test Type | Priority |
|-------------------|------------------------|---|---|---|--|-----------|----------|
| <b>TC-ETL-001</b> | ETL Pipeline Execution | Successful ETL pipeline execution       | Valid CSV file uploaded; All ETL services running | 1. Upload valid CSV file<br>2. Trigger ETL pipeline<br>3. Monitor pipeline stages (Detector → Transformer → Loader) | Pipeline completes successfully; Status set to “completed”; Data loaded into ClickHouse; Execution logs stored; User notified                | Positive  | High     |
| <b>TC-ETL-002</b> | ETL Pipeline Execution | ETL pipeline fails at Detector stage    | CSV file has unrecognizable schema                | 1. Upload CSV with invalid/unknown schema<br>2. Monitor pipeline<br>3. Check Detector service logs                  | Pipeline fails at Detector; Error message: “Schema detection failed”; Status = “failed”; Error details logged; User notified; No data loaded | Negative  | High     |
| <b>TC-ETL-003</b> | ETL Pipeline Execution | ETL pipeline fails at Transformer stage | Data transformation rules fail                    | 1. Upload CSV with data that fails transformation<br>2. Monitor pipeline<br>3. Check Transformer service logs       | Pipeline fails at Transformer; Error message: “Data transformation failed”; Status = “failed”; Specific transformation error logged;         | Negative  | High     |

|                   |                        |   |                                   |  |   |          |      |
|-------------------|------------------------|---|-----------------------------------|--|---|----------|------|
|                   |                        |   |                                   |  | Data not loaded;<br>User can retry  |          |      |
| <b>TC-ETL-004</b> | ETL Pipeline Execution | ETL pipeline fails at Loader stage          | ClickHouse connection unavailable | 1. Upload valid CSV<br>2. Stop ClickHouse service<br>3. Monitor pipeline<br>4. Check Loader service logs           | Pipeline fails at Loader; Error message: “ClickHouse connection failed”; Status = “failed”; Retry mechanism triggered; Data queued for retry; User notified | Negative | High |
| <b>TC-ETL-005</b> | ETL Pipeline Execution | ETL pipeline with large dataset (10M+ rows) | Valid CSV with 10M+ rows uploaded | 1. Upload large CSV file<br>2. Monitor pipeline execution<br>3. Check memory usage<br>4. Verify data in ClickHouse | Pipeline completes successfully; All rows loaded; Execution time logged (expected 5–15 min); No memory leaks; Data integrity verified                       | Positive | High |

Table 42 : Test Cases for ETL Pipeline Execution (Detailed Stages)

- Test Cases for ClickHouse Load & Data Integrity

| Test Case ID     | Requirement Name                 | Description                                | Preconditions                                   | Test Steps  | Expected Result  | Test Type | Priority |
|------------------|----------------------------------|--|---|---|--|-----------|----------|
| <b>TC-CH-001</b> | ClickHouse Load & Data Integrity | Successful data load into ClickHouse table | ETL completed successfully; Target table exists | 1. Complete ETL pipeline<br>2. Loader inserts data into ClickHouse<br>3. Query target table | Data loaded successfully; Row count matches transformed data; No errors during insert; Load metrics recorded     | Positive  | High     |
| <b>TC-CH-002</b> | ClickHouse Load & Data Integrity | Data type mismatch during ClickHouse load  | Transformed data has incompatible data types    | 1. Complete transformation<br>2. Attempt to load data into ClickHouse                       | Load fails; Error message: “Data type mismatch”; Loader stops operation; Error logged; No partial load committed | Negative  | High     |
| <b>TC-CH-003</b> | ClickHouse Load & Data Integrity | Partial failure handling                   | Network interruption during load                | 1. Start data load<br>2. Interrupt network  | Load rolled back or resumed based on strategy; No duplicate  | Negative  | High     |

|                  |                                  |   |                                 |  |  |          |      |
|------------------|----------------------------------|---|---------------------------------|--|--|----------|------|
|                  |                                  | during batch insert                                 |                                 | 3. Resume connection   | rows; Data consistency preserved; Retry logged   |          |      |
| <b>TC-CH-004</b> | ClickHouse Load & Data Integrity | Verify primary key or unique constraint enforcement | Table has defined unique key    | 1. Load data with duplicate keys<br>2. Query table                               | Duplicate rows rejected or handled according to config; Constraint violations logged; Data integrity maintained        | Positive | High |
| <b>TC-CH-005</b> | ClickHouse Load & Data Integrity | Large-scale data load performance                   | Large transformed dataset ready | 1. Load large dataset into ClickHouse<br>2. Measure load time and resource usage | Load completes within acceptable time; CPU and memory usage within thresholds; No timeouts; Performance metrics logged | Positive | High |

Table 43 : Test Cases for ClickHouse Load & Data Integrity

- Test Cases for Metadata Management & Data Lineage

| Test Case ID       | Requirement Name                   | Description                             | Preconditions                               | Test Steps  | Expected Result   | Test Type | Priority |
|--------------------|------------------------------------|---|---|---|---|-----------|----------|
| <b>TC-META-001</b> | Metadata Management & Data Lineage | Store metadata for uploaded data source | CSV file uploaded successfully              | 1. Upload CSV file<br>2. Complete upload process<br>3. Check metadata service                           | Metadata stored successfully; Includes source_name, file_id, uploader, upload_time, file_size, schema version             | Positive  | High     |
| <b>TC-META-002</b> | Metadata Management & Data Lineage | Track schema evolution across versions  | Multiple versions of same data source exist | 1. Upload initial CSV (v1)<br>2. Upload updated CSV with schema change (v2)<br>3. View metadata history | Schema versions tracked correctly; Version numbers incremented; Changes recorded; Previous versions preserved             | Positive  | High     |
| <b>TC-META-003</b> | Metadata Management & Data Lineage | Maintain end-to-end data lineage        | ETL pipeline completed                      | 1. Upload CSV<br>2. Run ETL pipeline<br>3. Inspect lineage graph  | Lineage recorded from source → detector → transformer → loader → ClickHouse; Each stage timestamped; Traceability ensured | Positive  | High     |
| <b>TC-META-004</b> | Metadata Management & Data Lineage | Metadata consistency after ETL failure  | ETL pipeline fails at any stage             | 1. Upload CSV<br>2. Trigger ETL<br>3. Force pipeline failure  | Metadata reflects failed status; Failure stage recorded; No inconsistent or orphan metadata entries                       | Negative  | High     |
| <b>TC-META-005</b> | Metadata Management & Data Lineage | Unauthorized user cannot view metadata  | User is Analyst or Executive                | 1. Login as Analyst/Executive<br>2. Attempt to access metadata page                                     | Access denied; Error message: “Insufficient permissions”; Metadata not exposed  | Negative  | High     |

*Table 44 : Test Cases for Metadata Management & Data Lineage*

- Test Cases for System Performance & Scalability

| Test Case ID | Requirement Name                 | Description                                      | Preconditions                             | Test Steps  | Expected Result   | Test Type | Priority |
|--------------|----------------------------------|--|---|---|---|-----------|----------|
| TC-PERF-001  | System Performance & Scalability | System handles high number of concurrent uploads | System deployed; Multiple users available | 1. Simulate 50+ users uploading CSV files concurrently<br>2. Monitor system performance<br>3. Check upload completion     | All uploads processed successfully; No system crash; Acceptable response times; Queueing handled correctly; No data loss  | Positive  | High     |
| TC-PERF-002  | System Performance & Scalability | ETL pipeline performance under heavy load        | Large datasets uploaded concurrently      | 1. Upload multiple large CSV files<br>2. Trigger ETL pipelines in parallel<br>3. Monitor CPU, memory, and processing time | ETL pipelines complete within acceptable SLA; Resource usage within limits; No memory leaks; ETL queues balanced          | Positive  | High     |
| TC-PERF-003  | System Performance & Scalability | Query performance with concurrent BI queries     | ClickHouse contains large datasets        | 1. Execute multiple aggregation queries concurrently<br>2. Monitor query latency  | Queries return correct results; Average latency within acceptable threshold; No query failures; System remains responsive | Positive  | High     |

|                    |                                  |  |                                   |   |   |          |        |
|--------------------|----------------------------------|--|-----------------------------------|---|---|----------|--------|
| <b>TC-PERF-004</b> | System Performance & Scalability | System scalability with horizontal scaling | Additional worker nodes available | 1. Add new ETL worker node<br>2. Trigger multiple pipelines<br>3. Compare throughput before and after scaling | Throughput increases after scaling; Load distributed evenly; No configuration issues; System scales horizontally                                | Positive | Medium |
| <b>TC-PERF-005</b> | System Performance & Scalability | Stress test beyond system capacity         | Load exceeds designed limits      | 1. Simulate extreme load beyond capacity<br>2. Monitor system behavior  | System degrades gracefully; Requests throttled or queued; No data corruption; Clear error messages returned; System recovers after load reduced | Negative | High   |

Table 45 : Test Cases for System Performance & Scalability

- Test Cases for Voice-to-SQL Processing & Intent Detection

| Test Case ID       | Requirement Name                           | Description   | Preconditions  | Test Steps  | Expected Result  | Test Type | Priority |
|--------------------|--|---|--|---|--|-----------|----------|
| <b>TC-VSQL-001</b> | Voice-to-SQL Processing & Intent Detection | Correct intent classification for analytical question     | Transcribed text available                                 | 1. Upload audio asking analytical question<br>2. Transcription completed<br>3. NLP module analyzes intent     | Intent correctly classified as <i>analytical</i> ; SQL generation triggered; Intent confidence score recorded        | Positive  | High     |
| <b>TC-VSQL-002</b> | Voice-to-SQL Processing & Intent Detection | Correct intent classification for conversational question | Transcribed text available                                 | 1. Upload audio asking conversational question<br>2. Transcription completed<br>3. NLP module analyzes intent | Intent classified as <i>conversational</i> ; No SQL generated; Textual response prepared; Status updated accordingly | Positive  | High     |
| <b>TC-VSQL-003</b> | Voice-to-SQL Processing & Intent Detection | SQL generation for valid analytical query                 | Intent classified as analytical; Schema metadata available | 1. Transcribe audio<br>2. Detect intent<br>3. Generate SQL query  | SQL generated successfully; Query syntactically valid; Uses correct table and columns; SQL stored with report        | Positive  | High     |
| <b>TC-VSQL-004</b> | Voice-to-SQL Processing                    | SQL generation failure due to                             | Transcribed text ambiguous                                 | 1. Upload ambiguous analytical question   | SQL generation skipped; Error message: “Ambiguous  | Negative  | High     |

|                    |  |                                 |                                |   |   |          |      |
|--------------------|--|---------------------------------|--------------------------------|---|---|----------|------|
|                    | & Intent Detection                         | ambiguous intent                |                                | 2. Attempt SQL generation               | query”; User prompted for clarification; No execution performed                           |          |      |
| <b>TC-VSQL-005</b> | Voice-to-SQL Processing & Intent Detection | SQL validation before execution | SQL generated from voice query | 1. Generate SQL<br>2. Run SQL validator | SQL validated successfully; Injection-safe; Compatible with ClickHouse; Execution allowed | Positive | High |

Table 46 : Test Cases for Voice-to-SQL Processing & Intent Detection

- Test Cases for Chart Generation & Visualization

| Test Case ID        | Requirement Name                 | Description                                    | Preconditions   | Test Steps   | Expected Result  | Test Type | Priority |
|---------------------|----------------------------------|--|---|--|--|-----------|----------|
| <b>TC-CHART-001</b> | Chart Generation & Visualization | Automatic chart type recommendation            | SQL query executed successfully; Result set available | 1. Execute SQL query<br>2. Navigate to Visualization tab | System recommends appropriate chart type (bar/line/pie) based on result schema; Recommendation confidence shown; User can accept or change | Positive  | High     |
| <b>TC-CHART-002</b> | Chart Generation & Visualization | Generate bar chart for categorical aggregation | SQL result contains category + metric                 | 1. Execute aggregation query<br>2. Select Bar Chart      | Bar chart generated correctly; Axes labeled; Data matches query results; Chart renders without errors                                      | Positive  | High     |
| <b>TC-CHART-003</b> | Chart Generation & Visualization | Generate line chart for time-series data       | SQL result contains date/time + metric                | 1. Execute time-based query<br>2. Select Line Chart      | Line chart rendered correctly; Time axis ordered; No missing points; Chart interactive (hover/zoom)  | Positive  | High     |
| <b>TC-CHART-004</b> | Chart Generation & Visualization | Handle incompatible chart selection            | SQL result incompatible with selected chart           | 1. Execute query<br>2. Select incompatible chart type    | Chart generation blocked; Error message: “Selected chart type is not compatible with data”; User prompted to choose another type           | Negative  | High     |

|                     |                                  |   |                             |   |  |          |        |
|---------------------|----------------------------------|---|-----------------------------|---|--|----------|--------|
| <b>TC-CHART-005</b> | Chart Generation & Visualization | Chart performance with large result set | Query returns large dataset | 1. Execute query with large result<br>2. Generate chart | Chart renders within acceptable time; Data sampling or pagination applied; UI remains responsive; No browser crash | Positive | Medium |
|---------------------|----------------------------------|---|-----------------------------|---|--|----------|--------|

Table 47 : Test Cases for Chart Generation & Visualization

- Test Cases for Dashboard Creation & Management

| Test Case ID       | Requirement Name                | Description                           | Preconditions                                   | Test Steps  | Expected Result  | Test Type | Priority |
|--------------------|---------------------------------|---------------------------------------|---|---|--|-----------|----------|
| <b>TC-DASH-001</b> | Dashboard Creation & Management | Create dashboard from executed report | Report executed successfully; Results available | 1. Login as Manager<br>2. Open executed report<br>3. Click “Add to Dashboard”<br>4. Enter dashboard name<br>5. Save | Dashboard created successfully; Chart added to dashboard; Dashboard visible in dashboard list; Ownership assigned to creator | Positive  | High     |
| <b>TC-DASH-002</b> | Dashboard Creation & Management | Add multiple charts to same dashboard | Dashboard exists; Multiple reports executed     | 1. Login as Manager<br>2. Open another executed report<br>3. Add chart to existing dashboard                        | Chart added successfully; Layout auto-adjusted; All charts render correctly  | Positive  | High     |
| <b>TC-DASH-003</b> | Dashboard Creation & Management | Edit dashboard layout                 | Dashboard exists with charts                    | 1. Login as Manager<br>2. Open dashboard<br>3. Rearrange or resize charts<br>4. Save layout                         | Layout updated successfully; New layout persisted; Charts render correctly after reload                                      | Positive  | Medium   |
| <b>TC-DASH-004</b> | Dashboard Creation & Management | Analyst cannot modify dashboard       | User is Analyst                                 | 1. Login as Analyst<br>2. Open dashboard<br>3. Attempt to add/remove charts or edit layout                          | Access denied; Edit controls hidden or disabled; Error message displayed; Dashboard unchanged                                | Negative  | High     |
| <b>TC-DASH-005</b> | Dashboard Creation & Management | Delete dashboard                      | Dashboard exists; User is Manager               | 1. Login as Manager<br>2. Open dashboard<br>3. Click “Delete Dashboard”   | Dashboard deleted successfully; Removed from list; Charts not deleted from reports; Action logged                            | Positive  | High     |

|  |  |  |  |                     |  |  |  |
|--|--|--|--|---------------------|--|--|--|
|  |  |  |  | 4. Confirm deletion |  |  |  |
|--|--|--|--|---------------------|--|--|--|

Table 48 : Test Cases for Dashboard Creation & Management

- Test Cases for Dashboard Sharing & Access Control

| Test Case ID | Requirement Name                   | Description                                     | Preconditions                     | Test Steps  | Expected Result  | Test Type | Priority |
|--------------|------------------------------------|---|-----------------------------------|---|--|-----------|----------|
| TC-SHARE-001 | Dashboard Sharing & Access Control | Manager shares dashboard with workspace members | Dashboard exists; User is Manager | 1. Login as Manager<br>2. Open dashboard<br>3. Click “Share”<br>4. Select workspace members<br>5. Set access level (view-only)<br>6. Save | Dashboard shared successfully; Selected members can view dashboard; Share permissions stored; Members notified | Positive  | High     |
| TC-SHARE-002 | Dashboard Sharing & Access Control | Analyst can view shared dashboard               | Dashboard shared with Analyst     | 1. Login as Analyst<br>2. Navigate to Dashboards  | Shared dashboard visible; Dashboard loads correctly; Charts rendered; View-only access enforced                | Positive  | High     |
| TC-SHARE-003 | Dashboard Sharing & Access Control | Analyst cannot edit shared dashboard            | Dashboard shared as view-only     | 1. Login as Analyst<br>2. Open shared dashboard<br>3. Attempt to edit layout or charts  | Edit controls hidden or disabled; Error message displayed; Dashboard remains unchanged                         | Negative  | High     |
| TC-SHARE-004 | Dashboard Sharing & Access Control | Unauthorized user cannot access dashboard       | User not member of workspace      | 1. Attempt to access dashboard URL directly   | Access denied; Error message: “Unauthorized access”; No dashboard data exposed; Attempt logged                 | Security  | High     |
| TC-SHARE-005 | Dashboard Sharing & Access Control | Revoke dashboard sharing                        | Dashboard shared previously       | 1. Login as Manager<br>2. Open dashboard sharing settings<br>3. Remove member access  | Access revoked successfully; Member no longer sees dashboard; Revocation logged; No cached access remains      | Positive  | High     |

- Test Cases for Notification & Messaging Management

| Test Case ID | Requirement Name                    | Description   | Preconditions                | Test Steps   | Expected Result  | Test Type | Priority |
|--------------|-------------------------------------|---|------------------------------|--|--|-----------|----------|
| TC-NOTIF-001 | Notification & Messaging Management | Receive notification on workspace invitation          | User invited to workspace    | 1. Invite user to workspace<br>2. User logs in<br>3. Check notifications panel | Notification received successfully;<br>Notification shows inviter name, workspace name, timestamp; Status marked as unread | Positive  | High     |
| TC-NOTIF-002 | Notification & Messaging Management | Mark notification as read                             | User has unread notification | 1. Login<br>2. Open notifications panel<br>3. Click notification               | Notification marked as read; Visual indicator updated; Read timestamp stored   | Positive  | Medium   |
| TC-NOTIF-003 | Notification & Messaging Management | Notification generated on role change                 | Manager changes member role  | 1. Login as Manager<br>2. Change role of member                                | Notification sent to affected member;<br>Notification includes old role and new role;<br>Action logged                     | Positive  | High     |
| TC-NOTIF-004 | Notification & Messaging Management | Notification generated on report execution completion | Report execution finishes    | 1. Execute report<br>2. Wait for completion                                    | Notification sent to report owner; Includes execution status and duration; Link to report provided                         | Positive  | Medium   |
| TC-NOTIF-005 | Notification & Messaging Management | Unauthorized user cannot send notifications           | User is Analyst/Executive    | 1. Login as Analyst/Executive<br>2. Attempt to send system notification        | Access denied; Error message displayed; No notification sent   | Negative  | High     |

- Test Cases for Account Deletion & Data Retention Policy

| Test Case ID | Requirement Name                  | Description                                       | Preconditions                              | Test Steps  | Expected Result   | Test Type | Priority |
|--------------|-----------------------------------|---|--|---|---|-----------|----------|
| TC-DEL-001   | Account Deletion & Data Retention | Admin permanently deletes user account            | Account is deactivated; User exists        | 1. Login as Admin<br>2. Navigate to User Management<br>3. Select deactivated user<br>4. Click “Delete Account”<br>5. Confirm deletion | Account deleted permanently; User record removed; Login impossible; Deletion logged; No orphan references     | Positive  | High     |
| TC-DEL-002   | Account Deletion & Data Retention | Deleted user cannot authenticate                  | User account deleted                       | 1. Attempt login with deleted user credentials  | Login fails; Error message: “Account does not exist”; No tokens issued; Authentication blocked                | Negative  | High     |
| TC-DEL-003   | Account Deletion & Data Retention | User data retention after account deletion        | User has historical reports and audit logs | 1. Delete user account<br>2. Check reports, audit logs, and ETL history   | Reports and audit logs preserved according to retention policy; User ID anonymized; Data integrity maintained | Positive  | High     |
| TC-DEL-004   | Account Deletion & Data Retention | Prevent deletion of last active workspace manager | User is last active manager in workspace   | 1. Login as Admin<br>2. Attempt to delete last manager account  | Deletion blocked; Error message: “Workspace must have at least one active manager”; Account preserved         | Negative  | High     |
| TC-DEL-005   | Account Deletion & Data Retention | Unauthorized user cannot delete account           | User is Analyst or Executive               | 1. Login as Analyst/Executive<br>2. Attempt to access account deletion endpoint   | Access denied; Error message displayed; No deletion performed; Attempt logged                                 | Security  | High     |

*Table 49 : Test Cases for Account Deletion & Data Retention Policy*

## **Chapter 5**

## **System Design**

## ***1-Introduction***

- The purpose of this section is to provide a comprehensive description of the full system architecture of the BI Voice Agent platform. The platform is designed to convert voice input into SQL queries, execute analytical operations on the data using ClickHouse, and present the results through interactive dashboards
- This section outlines the core system components, architectural layers, microservices, external integrations, and the end-to-end data flow across the entire system

## ***2-High-Level System Design***

The platform is built using a Microservices architecture, divided into several functional domains. It includes a front-end interface, an API gateway, independent service servers, an ETL pipeline powered by Kafka, multiple data storage layers, and external analytics tools such as Metabase

## • Microservices

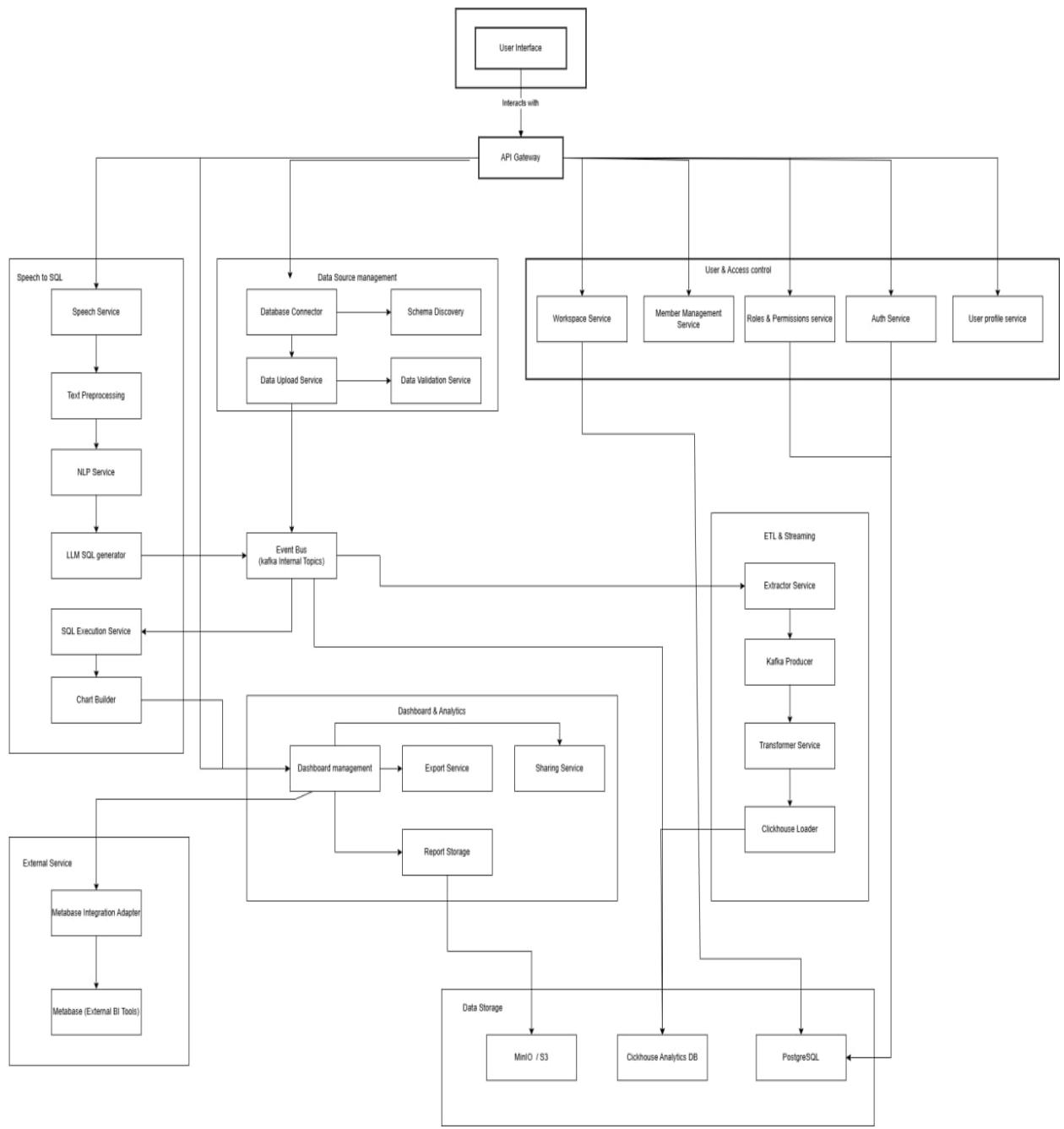


Figure 28 : System Architecture

## **3-Architecture Layers**

### **1-Client**

This layer includes the Web User Interface (Web UI)

Users interact with the system through this interface by performing actions such as logging in, uploading databases, submitting voice queries, viewing analytical results, and creating dashboards

### **2- API Gateway**

The API Gateway serves as the single entry point for all incoming requests

It handles routing, basic validation, and request forwarding. It also isolates the front-end from the internal microservices, ensuring a clean separation of concerns and secure communication

### **3- Microservices Layer**

#### - Speech-to-SQL Domain

- **Speech Service:** Converts voice input into text using Whisper
- **Text Preprocessing:** Cleans, normalizes, and prepares the text
- **NLP Service:** Analyzes user intent and extracts semantic context
- **LLM SQL Generator:** Generates an SQL query using a Large Language Model
- **SQL Execution Service:** Executes the generated SQL query on ClickHouse
- **Chart Builder:** Produces initial charts and tables from the query results

#### - Data Source Management Domain

- **Database Connector:** Connects to the user's external database
- **Data Upload Service:** Receives and processes uploaded CSV/SQL files
- **Schema Discovery:** Identifies tables, fields, and relationships

- **Data Validation Service:** Validates data formats, consistency, and correctness

- User & Access Control Domain

- **Auth Service:** Handles login, authentication, and session management
- **User Profile Service:** Manages user profile information
- **Roles & Permissions Service:** Controls roles, access levels, and authorization
- **Workspace Service:** Creates and manages user workspaces
- **Member Management Service:** Manages invitations, role assignments, and member updates

- Dashboard & Analytics Domain

- **Dashboard Management:** Creates and edits dashboards, and manages chart placement
- **Export Service:** Exports reports as PDF, Excel, or image files
- **Sharing Service:** Manages report sharing with Workspace members
- **Report Storage:** Stores generated charts, tables, and analytical outputs

- ETL & Streaming Domain

- **Extractor Service:** Reads and extracts raw input data
- **Kafka Producer:** Sends extracted data into Kafka topics
- **Transformer Service:** Cleans, structures, and transforms data in transit
- **ClickHouse Loader:** Loads processed data into the ClickHouse database

- Data Storage

- **ClickHouse Analytics DB** — the primary analytical data warehouse
- **PostgreSQL** — stores users, roles, invitations, dashboards, and metadata
- **MinIO / S3** — stores uploaded files, generated results, and assets

- External Integrations Layer

- Metabase Integration Adapter
- Metabase (External BI Tool)

- **ETL Pipeline**

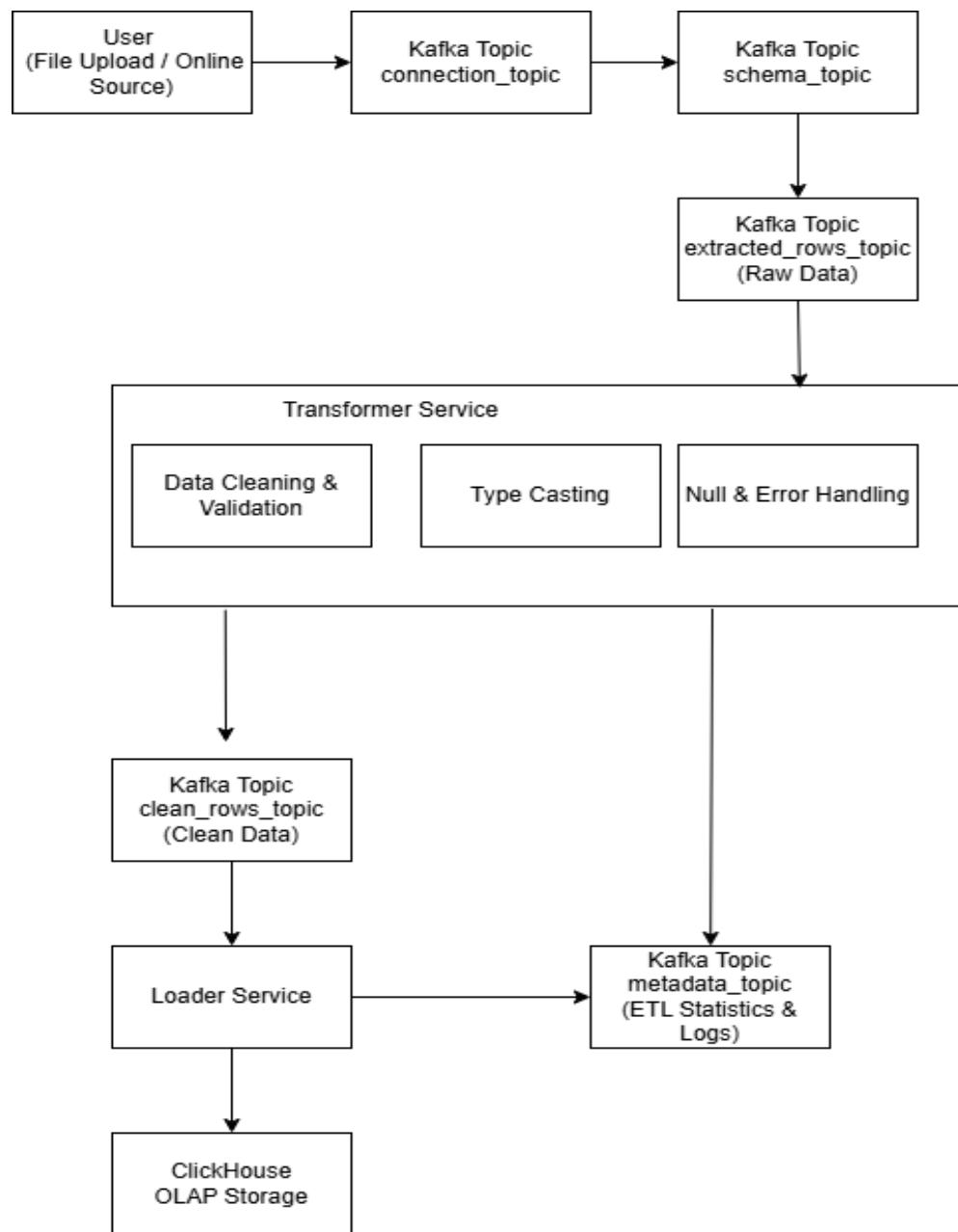


Figure 29 : ETL Pipeline

## **4. Detailed design for system component**

In this section we will dive into the detailed design for each component of the system, also mentions the design principles and patterns used to build a strong software system that fulfills its requirements.

### **4.1-Design class diagram**

#### **4.1.1 - User Authentication**

This class diagram illustrates the relationships and responsibilities between the components of the User Authentication module in the system. This module is responsible for managing user accounts and handling authentication-related operations such as registration, login, logout, and email verification. It includes the following primary entities:

- **UserRepository:**

Handles all database operations related to user management, including creating users, retrieving user data, updating user information, and deactivating user accounts.

- **AuthService:**

Provides core authentication services and business logic. It uses the UserRepository to perform user-related operations such as user registration, login, logout, and email verification.

- **BaseAuthView:**

A base class for all authentication-related views. It provides access to the authentication services and ensures a unified structure for authentication endpoints.

SignUpView, LoginView, LogoutView, and EmailVerificationView:

Represent the API endpoints responsible for user registration, login, logout, and email verification, respectively. Each view handles HTTP requests and delegates authentication logic to the service layer.

## Relationships

### 1. Dependency:

AuthService depends on UserRepository for all user data operations.

BaseAuthView depends on AuthService to access authentication functionality.

### 2. Inheritance:

SignUpView, LoginView, LogoutView, and EmailVerificationView inherit from BaseAuthView, allowing them to reuse common authentication behavior and maintain consistency across the authentication module.

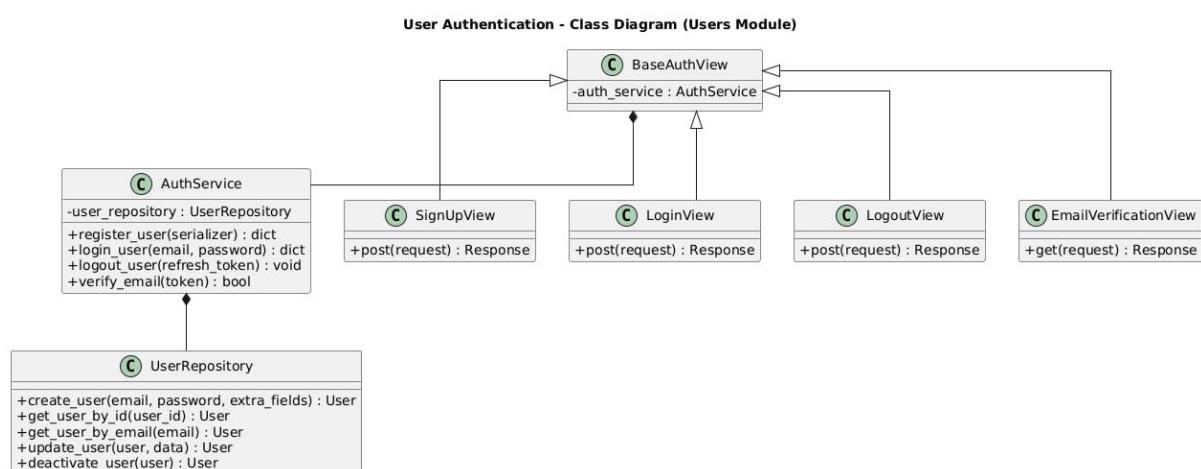


Figure 30 : Class diagram for user auth

## 4.1.2 - User Management

This class diagram illustrates the relationships and responsibilities between the components of the User Management module in the system. This module is responsible for managing users within a workspace, including creating users, updating their information, managing roles, activating or suspending accounts, and deleting users. It includes the following primary entities:

- **UserManagementService:**

Handles the core business logic related to user management. It coordinates user-related operations by interacting with both user and workspace member repositories.

- **UserRepository:**

Handles database operations related to user entities, such as retrieving users, creating new users, updating user data, deactivating users, and deleting user accounts.

- **WorkspaceMemberRepository:**

Manages workspace membership data, including assigning roles to users within a workspace, updating member status, and removing users from a workspace.

- **BaseUserManagementView:**

A base class for all user management-related views. It provides access to user management services and ensures a consistent structure for user management endpoints.

`UserListView`, `UserCreateView`, `UserUpdateView`, `UserSuspendView`, `UserActivateView`, and `UserDeleteView`:

Represent API endpoints for listing users, creating new users, updating user information, suspending users, activating suspended users, and deleting users, respectively.

## Relationships

### 1. Dependency:

`UserManagementService` depends on `UserRepository` and `WorkspaceMemberRepository` to perform all user and workspace-related data operations.

`BaseUserManagementView` depends on `UserManagementService` to access user management functionality.

### 2. Inheritance:

`UserListView`, `UserCreateView`, `UserUpdateView`, `UserSuspendView`, `UserActivateView`, and `UserDeleteView` inherit from `BaseUserManagementView`, allowing them to reuse common logic and maintain consistency across user management endpoints.

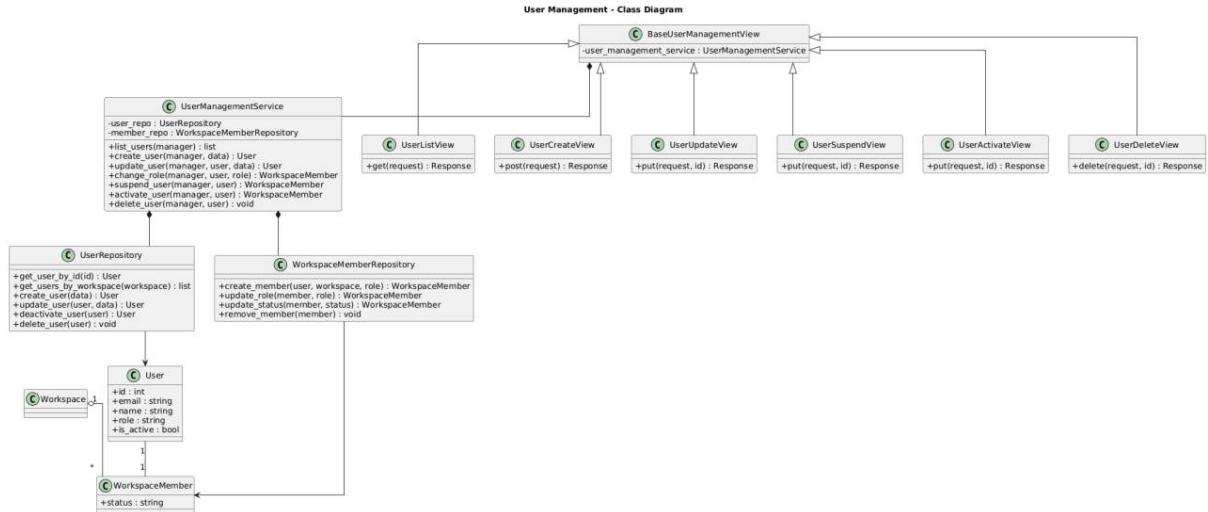


Figure 31 : class diagram for User managements

#### 4.1.3 - Workspace Management

This class diagram illustrates the relationships and responsibilities between the components of the Workspace Management module in the system. This module is responsible for managing workspaces and their members, including updating workspace information, handling member roles, sending invitations, accepting invitations, and controlling member status within a workspace. It includes the following primary entities:

- **WorkspaceService:**

Handles the core business logic related to workspace management. It coordinates workspace operations and member management by interacting with workspace and invitation repositories.

- **WorkspaceRepository:**

Handles database operations related to workspaces, such as retrieving workspace information, updating workspace data, managing workspace members, and removing users from a workspace.

- **InvitationRepository:**

Manages invitation-related data, including creating invitations, retrieving invitations by token, and expiring invitations.

- **InvitationService:**

Provides invitation-specific business logic, including inviting new members to a workspace and accepting workspace invitations using secure tokens.

- **BaseWorkspaceView:**

A base class for all workspace-related views. It provides access to workspace services and ensures a unified structure for workspace management endpoints.

**WorkspaceUpdateView**, **WorkspaceMembersView**, **InvitationView**, **RoleAssignmentView**, **MemberManageView**, **MemberSuspendView**, **MemberUnsuspendView**, and **AcceptInvitationView**:

Represent API endpoints for updating workspace details, listing workspace members, inviting users, assigning roles, managing members, suspending or unsuspending members, and accepting workspace invitations, respectively.

## Relationships

### 1. Dependency:

**WorkspaceService** depends on **WorkspaceRepository** and **InvitationRepository** to perform workspace and member-related data operations.

**InvitationService** depends on **InvitationRepository** to manage invitation lifecycle operations.

**BaseWorkspaceView** depends on **WorkspaceService** to access workspace management functionality.

**AcceptInvitationView** depends on **InvitationService** to handle invitation acceptance.

### 2. Inheritance:

`WorkspaceUpdateView`, `WorkspaceMembersView`, `InvitationView`, `RoleAssignmentView`, `MemberManageView`, `MemberSuspendView`, and `MemberUnsuspendView` inherit from `BaseWorkspaceView`, ensuring consistent behavior and reuse of shared workspace logic across all workspace management endpoints.

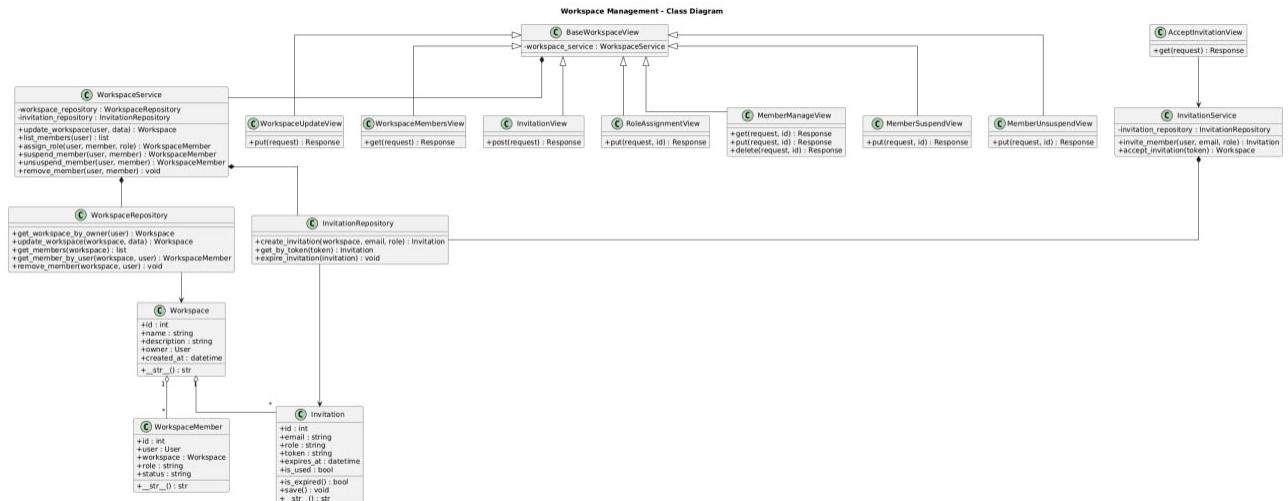


Figure 32 : class diagram for Workspace managements

#### 4.1.4 - Database Management

This class diagram illustrates the relationships and responsibilities between the components of the Database Management module in the system. This module is responsible for handling user-uploaded databases, managing their lifecycle, interacting with the analytical database engine, and providing database preview and status functionalities. It includes the following primary entities:

- **DatabaseService:**

Handles the core business logic related to database management. It coordinates database upload, replacement, deletion, preview generation, and status checking by interacting with the database repository and the analytical database client.

- **DatabaseRepository:**

Handles database-related persistence operations, including creating database records, retrieving databases by workspace, updating database status and metadata, and deleting databases.

- ClickHouseClient:

Provides an abstraction layer for interacting with the ClickHouse analytical database engine. It is responsible for executing queries, retrieving table previews, schemas, row counts, and checking table existence.

- DatabaseUtils:

Provides utility functions related to database processing, such as cleaning database data and formatting file sizes for display purposes.

- BaseDatabaseView:

A base class for all database-related views. It provides access to database services and ensures a unified structure for database management endpoints.

DatabaseUploadView, DatabaseDetailView, DatabasePreviewView, DatabaseStatusView, and DatabaseHealthCheckView:

Represent API endpoints for uploading databases, retrieving or deleting database details, previewing database content, checking database status, and performing database health checks, respectively.

## Relationships

### 1. Dependency:

DatabaseService depends on DatabaseRepository to perform database-related persistence operations.

DatabaseService depends on ClickHouseClient to interact with the analytical database engine.

BaseDatabaseView depends on DatabaseService to access database management functionality.

DatabaseService utilizes DatabaseUtils for auxiliary database processing tasks.

## 2. Inheritance:

DatabaseUploadView, DatabaseDetailView, DatabasePreviewView, DatabaseStatusView, and DatabaseHealthCheckView inherit from BaseDatabaseView, ensuring consistency and reuse of shared logic across all database management endpoints.

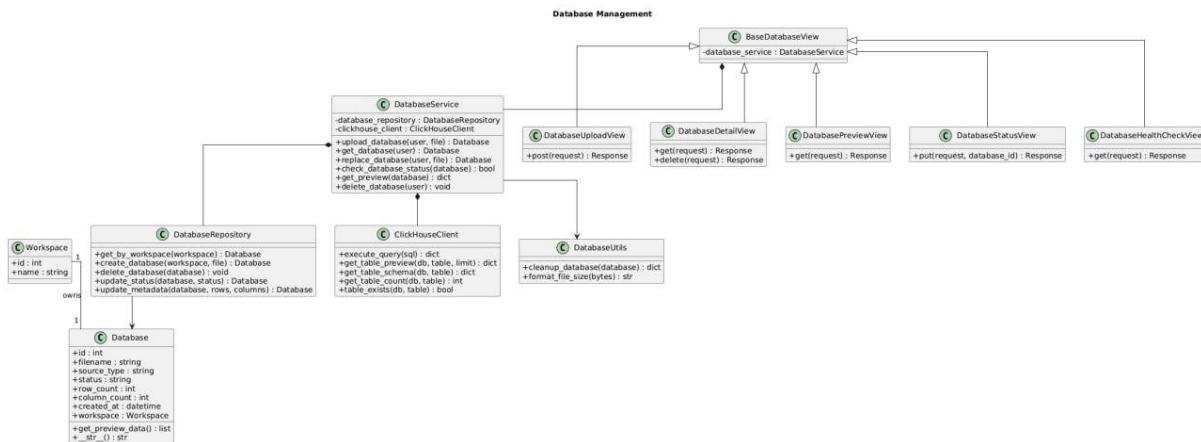


Figure 33 : class diagram for Database Managements

### 4.1.5 - AI Services

This class diagram illustrates the relationships and responsibilities between the components of the AI Services module in the system. This module is responsible for processing voice-based analytical requests, transforming spoken input into structured queries, and ensuring the correctness and security of generated SQL queries. It includes the following primary entities:

- AIProcessingService:

Handles the core AI processing pipeline. It orchestrates the entire flow of voice request processing, starting from audio transcription and ending with validated SQL query generation. It integrates multiple AI and validation components to ensure accurate and secure query generation.

- WhisperService:

Responsible for transcribing audio input into text using a speech-to-text model.

- QuestionClassifier:

Classifies the transcribed text to identify the type of analytical question being asked.

- IntentService:

Extracts structured analytical intent from the transcribed text, including metrics, dimensions, and filters.

- IntentValidator:

Validates the extracted intent semantically, checks schema compatibility, and ensures SQL executability.

- SQLCompiler:

Compiles validated intent into executable SQL queries while applying appropriate casting and formatting rules.

- SQLGuard:

Performs security validation on generated SQL queries to prevent unsafe or unauthorized query execution.

- VoiceRequestRepository:

Handles persistence operations for voice requests, including storing audio files, transcriptions, intents, generated SQL queries, and request status updates.

- AIServiceView:

Represents the API endpoint responsible for receiving voice-based requests and triggering the AI processing workflow.

## Relationships

## 1. Dependency:

AIProcessingService depends on WhisperService, QuestionClassifier, IntentService, IntentValidator, SQLCompiler, and SQLGuard to execute the AI processing pipeline.

AIProcessingService depends on VoiceRequestRepository to store and update voice request data throughout processing stages.

AIServiceView depends on AIProcessingService to initiate voice request processing.

## 2. Association and Ownership:

A User with a manager role owns voice requests and is allowed to initiate voice-based analytical queries within a Workspace.

A Workspace contains multiple VoiceRequest entities representing historical voice queries and their results

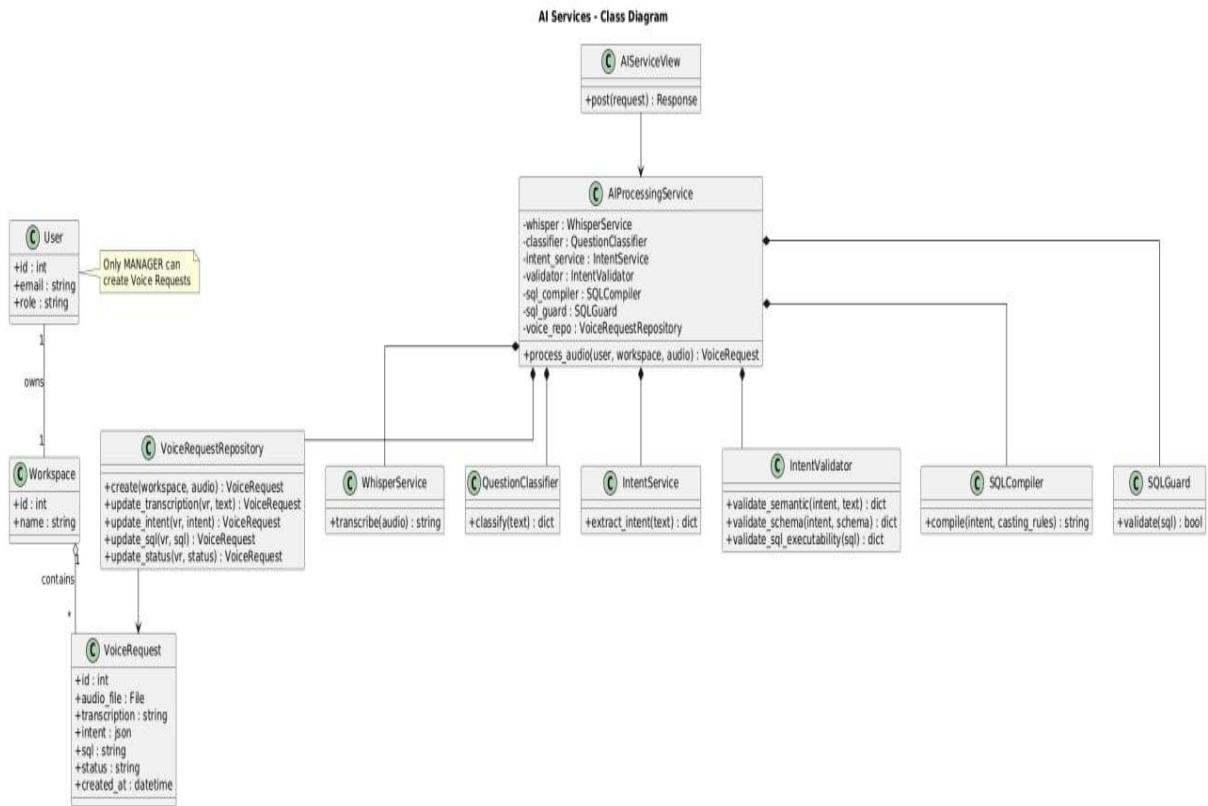


Figure 34 : Class Diagram for AI Services

#### 4.1.6 - Voice Reports

This class diagram illustrates the relationships and responsibilities between the components of the Voice Reports module in the system. This module is responsible for managing voice-based analytical reports, starting from audio upload and processing, passing through query execution and validation, and ending with report visualization and dashboard integration. It includes the following primary entities:

- **VoiceReportService:**

Handles the core business logic related to voice reports. It coordinates audio upload, AI processing, SQL execution, report management, and integration with dashboards and visualization services.

- **VoiceReportRepository:**

Handles persistence operations for voice reports, including creating reports, updating transcription, intent, SQL queries, report status, and retrieving reports by workspace.

- **AIServiceClient:**

Acts as a client interface to the AI Services module, responsible for processing audio input and returning structured analytical results.

- **ClickHouseExecutor:**

Executes validated SQL queries against the analytical database engine and returns query results.

- **SQLGuard:**

Ensures the security and validity of SQL queries before execution.

- **MetabaseService:**

Integrates the system with Metabase to create analytical questions, dashboards, and associate reports with visual dashboards.

- **JWTEmbeddingService:**

Generates secure embedded URLs for dashboards, allowing controlled access to visual reports.

- **SQLEditHistoryRepository:**

Stores the history of SQL edits applied to voice reports, enabling traceability and auditability of query modifications.

- **BaseVoiceReportView:**

A base class for all voice report-related views. It provides access to voice report services and ensures a consistent structure for report-related endpoints.

`VoiceUploadView`, `QueryExecuteView`, `SQLEditView`, `ReportListView`, `ReportDetailView`, and `WorkspaceDashboardView`:

Represent API endpoints for uploading voice input, executing queries, editing SQL queries, listing reports, viewing or deleting report details, and displaying workspace dashboards, respectively.

## Relationships

### 1. Dependency:

`VoiceReportService` depends on `VoiceReportRepository`, `SQLEditHistoryRepository`, `AIServiceClient`, `ClickHouseExecutor`, `SQLGuard`, `MetabaseService`, and `JWTEmbeddingService` to perform report processing, execution, validation, and visualization tasks.

`BaseVoiceReportView` depends on `VoiceReportService` to access voice report functionality.

### 2. Association and Ownership:

A User creates and manages voice reports within a Workspace.

A Workspace contains multiple `VoiceReport` entities representing generated analytical reports.

Each `VoiceReport` may have multiple SQL edit history records to track modifications over time.

### 3. Inheritance:

`VoiceUploadView`, `QueryExecuteView`, `SQLEditView`, `ReportListView`, `ReportDetailView`, and `WorkspaceDashboardView` inherit from `BaseVoiceReportView`, ensuring reuse of common logic and consistency across all voice report endpoints.

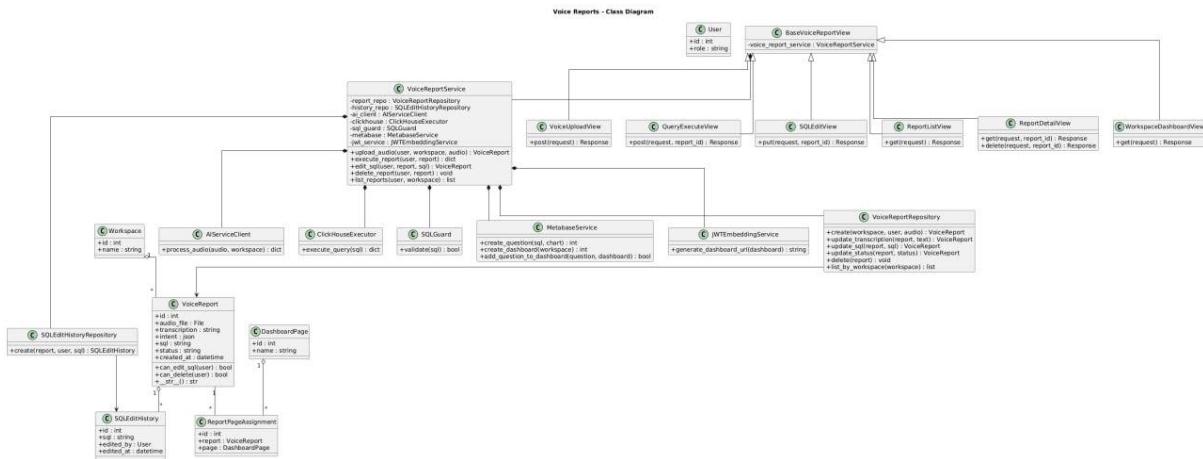


Figure 35 : class diagram for Voice Managements

# **Chapter 6**

## **Practical Implementation**

## ***6.1 Introduction***

In this chapter, we present the practical implementation of the proposed system, highlighting the technologies, tools, and frameworks used during development. This chapter focuses on how the system was built in practice, covering backend and frontend implementation, database management, artificial intelligence components, and infrastructure tools. In addition, system interfaces are demonstrated, and the chapter concludes with an overview of testing procedures used to verify system functionality across different scenarios.

## ***6.2 Used Tools***

### ***6.2.1 Django & Django REST Framework***

Django is a high-level Python web framework that enables rapid development and clean, pragmatic design. It provides built-in features such as authentication, ORM, and administrative interfaces, which significantly reduce development complexity.

In this project, Django is used as the main backend framework for implementing the system's core logic, user management, workspace management, database handling, and voice report services.

Additionally, Django REST Framework (DRF) is used to build RESTful APIs. DRF integrates seamlessly with Django's models, views, and URL routing, making it ideal for developing scalable and secure APIs. It provides serializers, authentication mechanisms, permission control, and standardized API responses, which are essential for frontend-backend communication.

### ***6.2.2 React***

React is a popular JavaScript library for building user interfaces, developed by Facebook. It enables the creation of reusable UI components and efficient rendering through its virtual DOM mechanism.

In this project, React is used to implement the frontend application, providing a dynamic and interactive user interface. React facilitates smooth navigation between pages, real-time UI updates, and modular component-based design. Tools such as React Router are used for client-side routing, while state management is handled using lightweight libraries to maintain authentication and workspace states.

### **6.2.2 React**

React is a popular JavaScript library for building user interfaces, developed by Facebook. It enables the creation of reusable UI components and efficient rendering through its virtual DOM mechanism.

In this project, React is used to implement the frontend application, providing a dynamic and interactive user interface. React facilitates smooth navigation between pages, real-time UI updates, and modular component-based design. Tools such as React Router are used for client-side routing, while state management is handled using lightweight libraries to maintain authentication and workspace states.

### **6.2.3 PostgreSQL Database**

PostgreSQL is an open-source relational database management system known for its robustness, reliability, and advanced features.

In this system, PostgreSQL is used as the primary relational database, storing core application data such as users, workspaces, voice reports, database metadata, and permissions. The Django ORM is used to interact with PostgreSQL, providing an abstraction layer that simplifies database operations while maintaining data integrity and consistency.

#### **6.2.4 ClickHouse**

ClickHouse is a high-performance, column-oriented database designed for analytical workloads and large-scale data processing.

In this project, ClickHouse is used as the analytical database engine. It stores processed data generated by the ETL pipeline and serves as the execution engine for analytical SQL queries generated from voice inputs. ClickHouse is accessed using both HTTP and native protocols, depending on the use case, ensuring efficient query execution and batch data loading.

#### **6.2.5 Visual Studio Code (VS Code)**

Visual Studio Code is a lightweight yet powerful source code editor that provides features such as syntax highlighting, IntelliSense, debugging tools, and Git integration.

VS Code is used as the primary development environment for the entire project, including frontend, backend, AI services, and ETL components. Its extensibility and rich plugin ecosystem enhance productivity and code quality.

#### **6.2.6 Bitbucket**

Bitbucket is a Git-based source code repository hosting platform that supports version control and collaborative development.

In this project, Bitbucket is used to manage the source code repository, track changes, and maintain development history. It facilitates collaboration between team members and ensures proper versioning of the system throughout the development lifecycle.

## ***6.3 AI Used Technologies and Tools***

### **6.3.1 OpenAI Whisper**

OpenAI Whisper is an automatic speech recognition (ASR) model capable of transcribing speech into text with high accuracy.

In this system, Whisper is used as the speech-to-text component, converting voice inputs submitted by users into textual form. The transcribed text serves as the initial input for subsequent natural language processing and intent extraction stages.

### **6.3.2 Large Language Models (LLMs) via OpenRouter**

Large Language Models are used to extract analytical intent and generate structured SQL queries from natural language input.

The system integrates LLMs through the OpenRouter API, enabling flexible model selection and scalable inference. These models are responsible for understanding user questions, identifying metrics and filters, and producing SQL queries suitable for analytical execution.

### **6.3.3 LangChain and LangGraph**

LangChain is a framework designed for building applications powered by large language models. It provides abstractions for prompt management, chaining logic, and model interaction.

LangGraph extends this approach by enabling graph-based workflows for reasoning and decision-making. In this project, LangChain and LangGraph are used to implement the reasoning pipeline, allowing structured processing of user queries, classification of question types, and routing between analytical and conversational flows.

### **6.3.4 Pydantic**

Pydantic is a data validation library that uses Python type annotations to enforce data correctness.

In this system, Pydantic is used to validate intent schemas, ensuring that extracted intents, metrics, dimensions, and filters conform to predefined structures before SQL generation and execution. This validation step improves system reliability and reduces runtime errors.

### **6.3.5 Ollama and Local Language Models**

Ollama is a local inference platform used to run large language models on-premise without relying on external cloud services. In this project, Ollama is utilized to host and execute local language models for intent extraction and SQL generation.

Using local models provides several advantages, including improved data privacy, reduced latency, and full control over model behavior. This approach ensures that sensitive analytical queries and business data are processed locally without being sent to third-party services.

## **6.4 ETL Pipeline Technologies**

To support large-scale data ingestion and processing, the system includes an ETL pipeline built using modern distributed tools.

Apache Kafka is used as a message broker to enable asynchronous communication between ETL services.

Pandas and OpenPyXL are used for data extraction, transformation, and file processing.

SurrealDB is used to store metadata, logs, and audit information related to ETL operations.

ClickHouse serves as the final destination for processed analytical data.

This architecture ensures scalability, fault tolerance, and efficient data processing.

- Apache Kafka

Apache Kafka is a distributed event streaming platform used as the backbone of the ETL pipeline. In this system, Kafka enables asynchronous and decoupled communication between ETL services, allowing data to flow efficiently through extraction, transformation, and loading stages.

Kafka topics are used to transfer structured messages between services, ensuring scalability, fault tolerance, and reliable data processing across the ETL pipeline.

- SurrealDB

SurrealDB is a modern, cloud-native database used for storing ETL metadata, logs, and audit information. In this project, SurrealDB records pipeline execution details, transformation metadata, and processing status.

This separation between analytical data (stored in ClickHouse) and metadata (stored in SurrealDB) improves system organization, traceability, and monitoring capabilities.

## ***6.5 Infrastructure and DevOps Tools***

The system uses Docker and Docker Compose to containerize and orchestrate backend services, ETL components, Kafka, ClickHouse, and SurrealDB. Environment variables are managed using .env files to ensure secure and flexible configuration across environments.

This setup simplifies deployment, improves reproducibility, and enables local and distributed development environments.

## **6.6 Testing**

Testing is conducted using pytest and pytest-django, which provide a robust testing framework for Django applications. Test cases are designed to validate API endpoints, authentication flows, database operations, and AI processing logic.

These tests ensure that the system behaves correctly under different scenarios and that core functionalities operate as expected.

## **6.7 Data Visualization and Reporting**

- Metabase

Metabase is an open-source business intelligence and data visualization platform used to present analytical results generated by the system. In this project, Metabase is integrated to create interactive dashboards and visual reports based on SQL queries generated from voice input.

The system programmatically creates questions and dashboards in Metabase and embeds them securely into the application using JWT-based embedding. This integration enables users to view analytical insights in a clear and interactive manner without directly accessing the Metabase interface.

## 6.3 System interfaces

- Home page



Figure 36 : Home Page Interface

- Log in

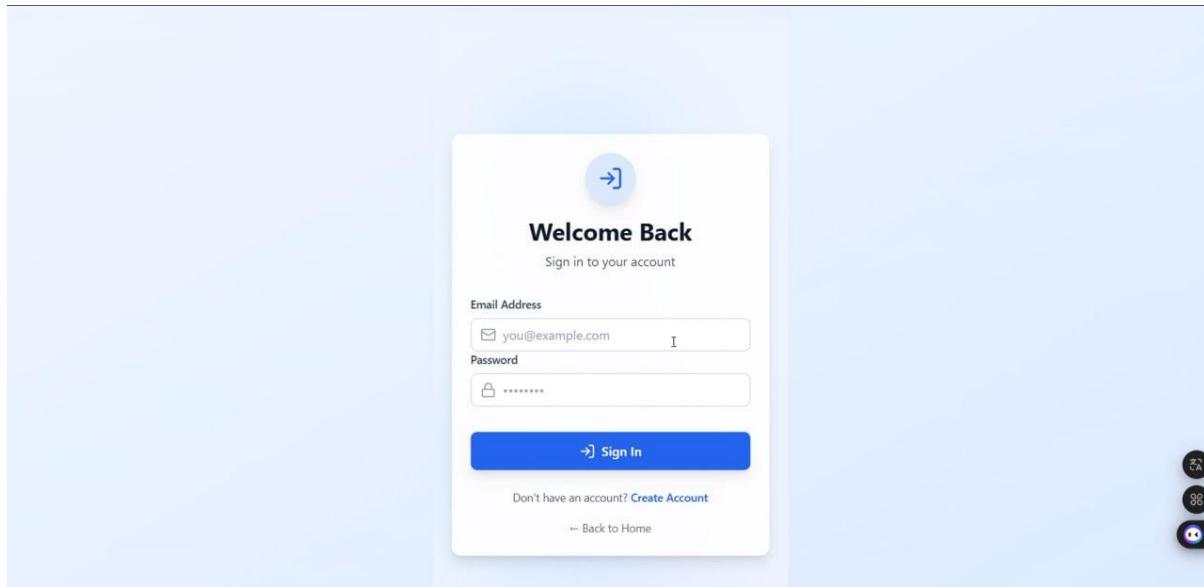


Figure 37 : Login interface

- **Sign up**

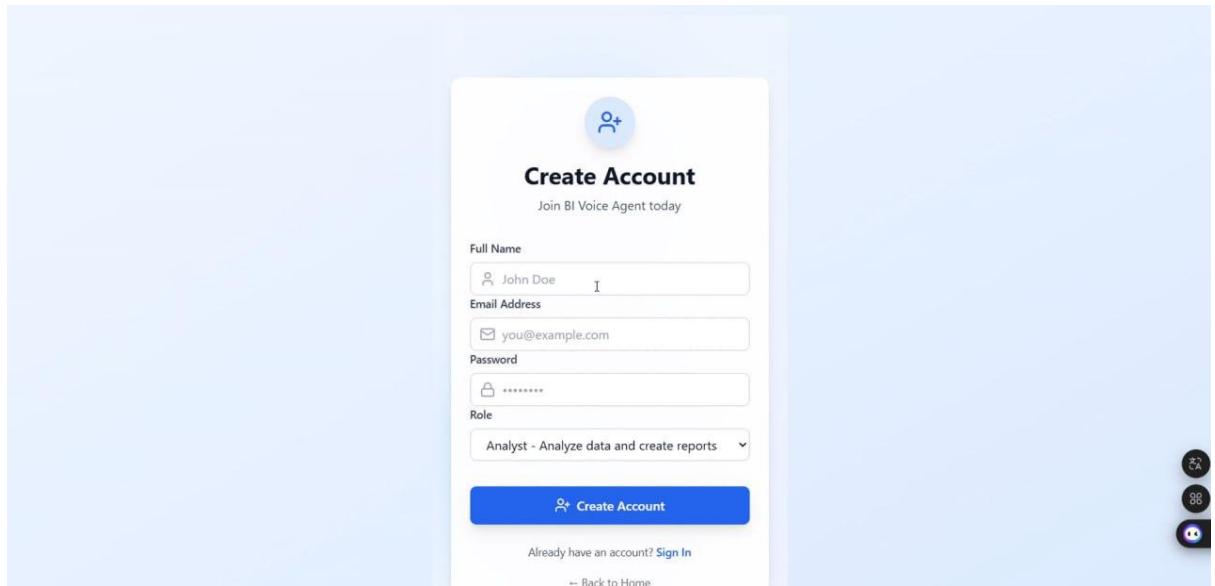


Figure 38 : sign up interface

- **Link for verification email**

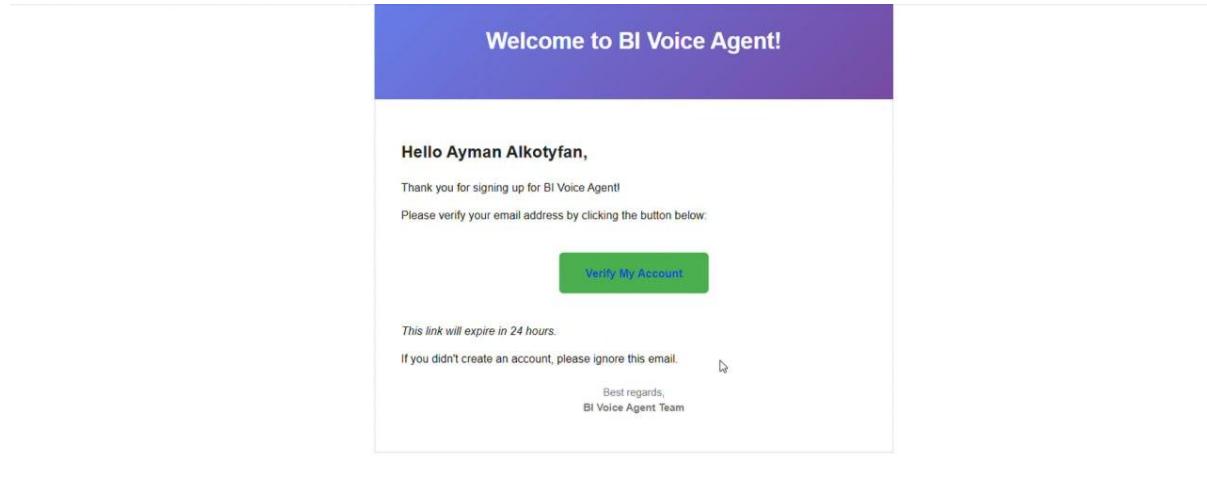


Figure 39 : Email verification in gmail

## • Dashboard manager

Figure 40 : Dashboard Manager

## • Workspace settings

Figure 41 : Workspace Settings

## • Profile settings

Welcome, Ayman Alkotyfan MANAGER

**Profile Settings**  
Manage your account information

**Profile Information**

Full Name  
John Doe

Email Address  
you@example.com

**Reset Changes** **Save Changes**

**Account Information**

Role Your account type manager

Email Status Verification status Verified

Account Status Active or suspended Active

Figure 42 : Profile settings

## • Detective account

Welcome, Ayman naeem MANAGER

**Reset Changes** **Save Changes**

**Account Information**

Role Your account type manager

Email Status Verification status Verified

Account Status Active or suspended Active

**Danger Zone**

**⚠ Deactivate Account**  
Once you deactivate your account, you will lose access to all workspaces and your data. This action can be reversed by contacting support.

**⚠ Deactivate Account**

Figure 43 : Detective account interface

## • invitation member

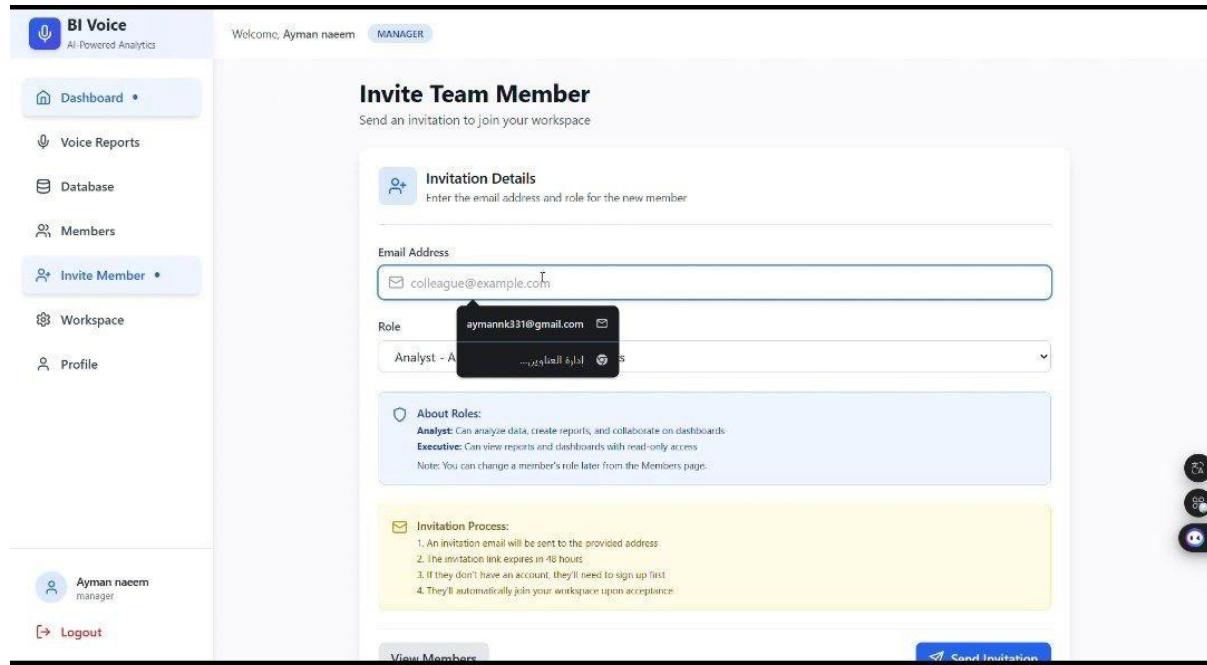


Figure 44 : invitation member interface

## • Join workspace in gmail

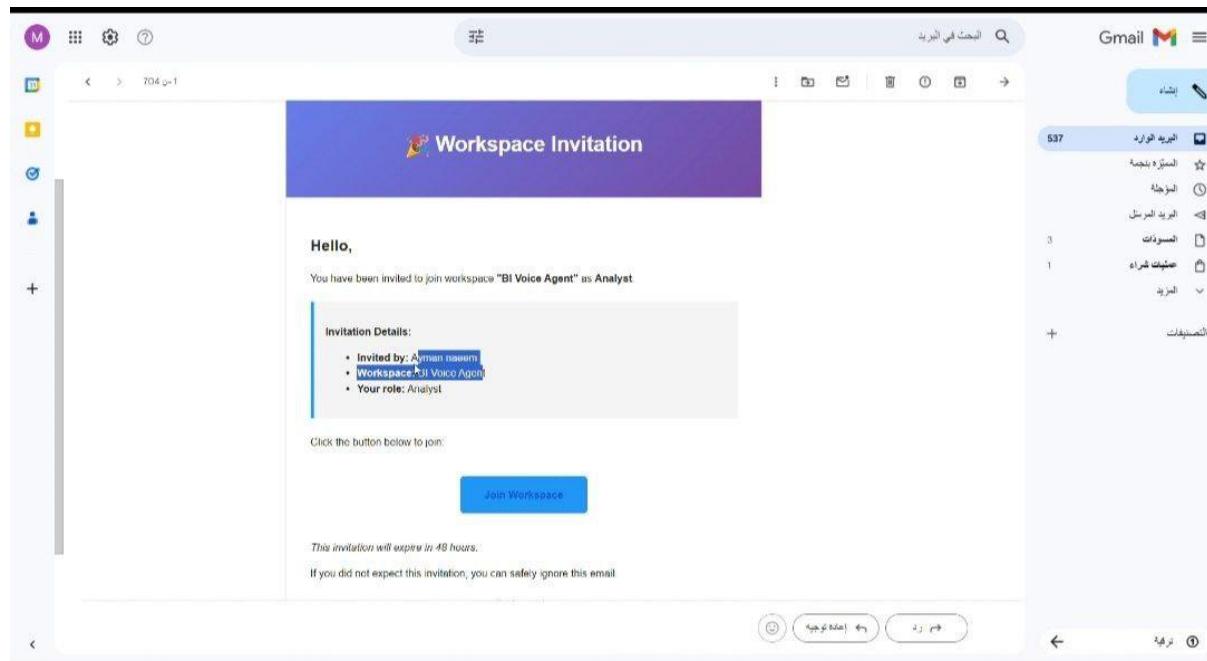


Figure 45 : join workspace in gmail

## • Dashboard analysis

Figure 46 : dashboard Analysis

## • Member settings

Figure 47 : Member Settings

- **Suspend member**

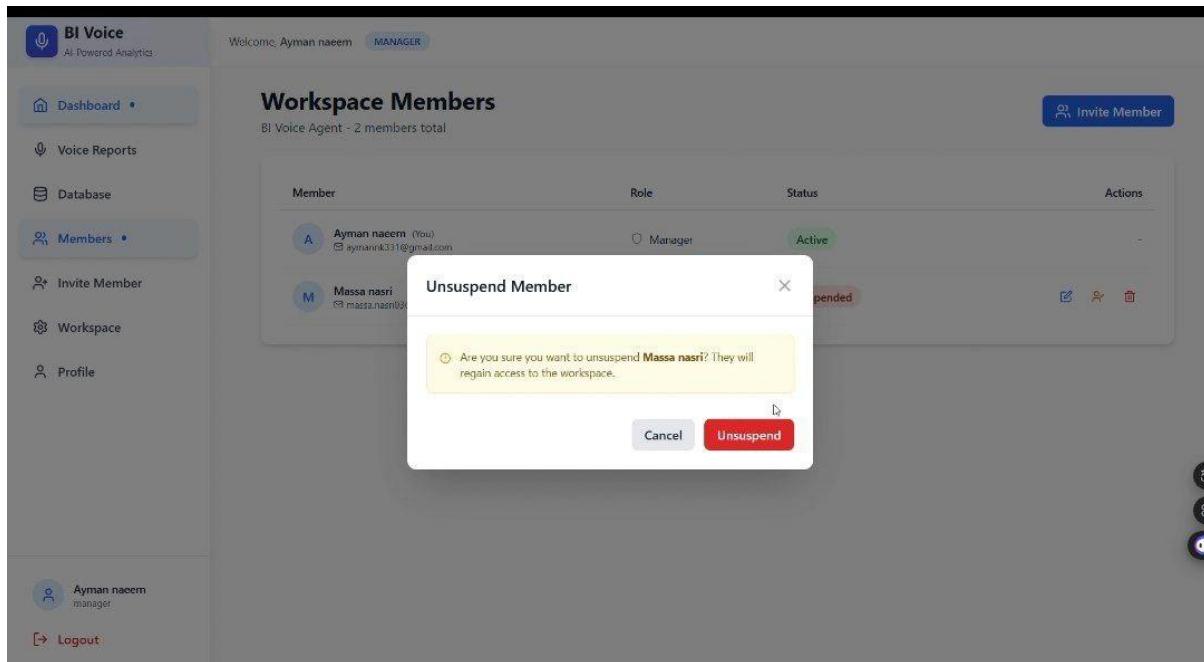


Figure 48 : Suspend Member interface

- **Change member role**

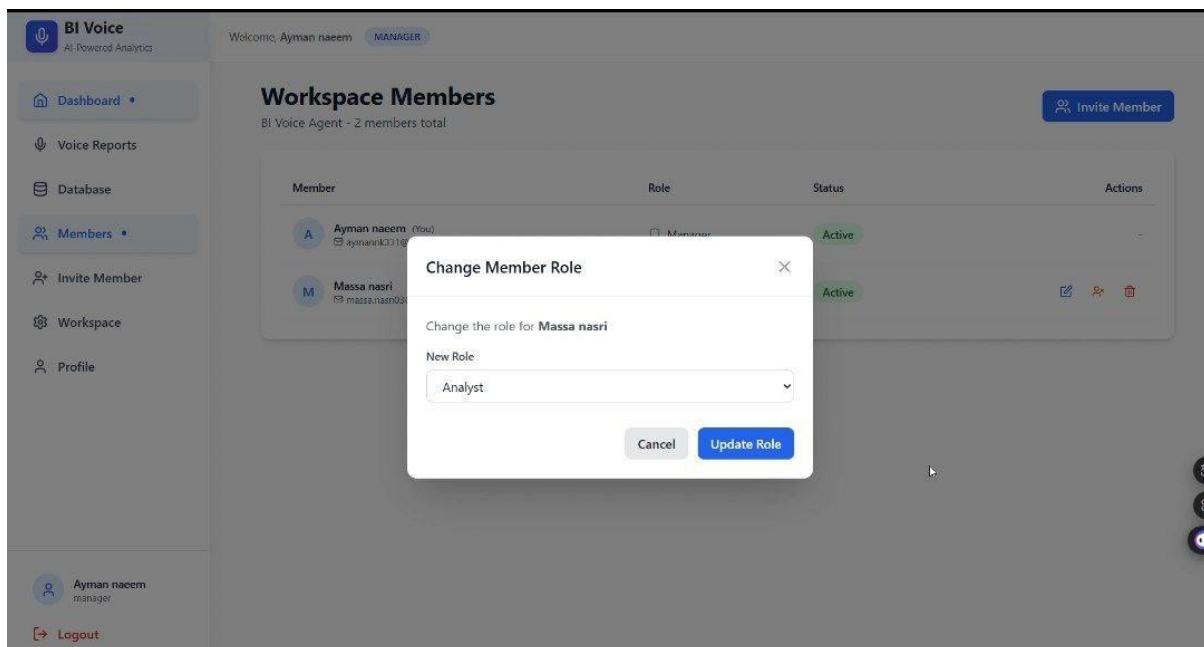


Figure 49 : change member role

## • Dashboard Executive

The screenshot shows the BI Voice Executive Dashboard interface. At the top, there's a navigation bar with 'BI Voice' and 'Welcome, Massa nasri EXECUTIVE'. On the left, a sidebar includes 'Dashboard', 'Analytics', 'Members', 'Profile', and user information for 'Massa nasri executive'. A central green header bar says 'Executive Dashboard' and 'Comprehensive analytics and insights at a glance'. Below it, three cards show 'Total Reports 0', 'Completed 0', and 'Total Rows 0'. In the center, a large yellow circle with an exclamation mark contains the text 'Dashboard Not Available' and the message 'No dashboard available yet. Reports need to be created first.' A green button labeled 'View Analytics Page' is below this. At the bottom right, a URL 'localhost:5173/dashboard/analytics' is visible.

Figure 50 : Dashboard Executive

## • Upload database

The screenshot shows the BI Voice Database Management page. At the top, there's a navigation bar with 'BI Voice' and 'Welcome, Ayman naeem MANAGER'. On the left, a sidebar includes 'Dashboard', 'Voice Reports', 'Database \*', 'Members', 'Invite Member', 'Workspace', 'Profile', and user information for 'Ayman naeem manager'. A success message 'Database uploaded successfully and ETL processing started' is displayed in a black box. The main area has a title 'Database Management' with the sub-instruction 'Upload and manage your data source'. It features a 'Database Information' section with details: 'Filename: states\_all.csv', 'File Size: 287.47 KB', 'Rows: -', 'Columns: -', 'Upload Date: 29 11:29:40 2026/1/1', and 'Status: Processing'. Below this, a progress bar with a sunburst icon and the text 'Processing Your Data' is shown, along with the message 'Your database is being processed through the ETL pipeline. This may take a few moments.' At the bottom right, there's a red 'Delete Database' button.

Figure 51 : Upload database

## • Upload voice

Welcome, Ayman naem / MANAGER

**Database Management**

Upload and manage your data source

**Database Information**

- Filename: states\_all.csv
- File Size: 287.47 KB
- Rows
- Columns
- Upload Date: 29 11:29:40 2026/1/
- Status: Processing

**Processing Your Data**

Your database is being processed through the ETL pipeline. This may take a few moments.

Ayman naem / manager

Logout

Database uploaded successfully and ETL processing started

Delete Database

Figure 52 : Upload Voice

## • SQL editor

Welcome, Massa nasri / ANALYST

**SQL Editor**

Edit and optimize SQL queries for voice reports

**All Reports**

Report #4 completed  
Show average total revenue by year

**Report #4** completed

Show average total revenue by year

**SQL Query**

```
SELECT * FROM (SELECT YEAR, IF(isNaN(avgIf(toFloat64OrNull(TOTAL_REVENU), toFloat64OrNull(TOTAL_REVENU) IS NOT NULL)), 0, avgIf(toFloat64OrNull(TOTAL_REVENU), toFloat64OrNull(TOTAL_REVENU) IS NOT NULL)) AS avg_total_revenue FROM all.states_all.csv WHERE toFloat64OrNull(TOTAL_REVENU) IS NOT NULL GROUP BY YEAR) WHERE avg_total_revenue != 0;
```

Save Changes Execute

**Execution Results**

| Rows | Time  | Chart |
|------|-------|-------|
| 12   | 236ms | Line  |

Massa nasri / analyst

Logout

Figure 53 : Edit SQL

## ***6.4 Summary***

This chapter presented the practical implementation of the BI Voice Agent system, detailing the technologies and tools used across all layers of the application. The system combines modern web technologies, advanced AI models, distributed data processing, and scalable infrastructure to deliver an intelligent voice-based business intelligence platform.

# **Chapter 7**

## **Report Overview**

## ***7.1 Introduction***

This chapter presents an overview of the entire report and explains the purpose and content of each chapter. The main goal of this chapter is to guide the reader through the logical organization of the report and clarify how each chapter contributes to the development, analysis, and implementation of the proposed AI-based BI Voice Agent system. The chapter ensures a clear understanding of the report flow, starting from problem identification and theoretical background, moving through system analysis and design, and ending with implementation and evaluation.

## ***7.2 Report Structure and Purposes***

- **Chapter 1: Introduction**

This chapter introduces the project background and motivation, defines the problem statement addressed by the BI Voice Agent system, and outlines the project objectives. It also provides a general overview of the proposed solution, highlighting the role of artificial intelligence in enabling voice-based business intelligence analysis, and presents the structure of the report.

- **Chapter 2: Fundamental Concepts and Literature Review**

This chapter discusses the fundamental concepts related to the project, including Business Intelligence (BI), voice processing systems, speech-to-text technologies, natural language processing (NLP), and data analytics. It also reviews related studies and existing systems that combine AI, voice interfaces, and BI tools, providing a comparative analysis and identifying gaps that the proposed system aims to address.

- Chapter 3: Project Management

This chapter focuses on the project management aspects of the BI Voice Agent. It includes the project charter, scope of work (SOW), project timeline illustrated through a Gantt chart, and risk management analysis. This chapter ensures that the project is planned, monitored, and executed in a structured and controlled manner.

- Chapter 4: System Analysis

This chapter presents the system analysis phase, starting with the overall project timeline and followed by the Software Requirements Specification (SRS). It details functional and non-functional requirements, use case descriptions, initial test cases, and the initial Requirement Traceability Matrix (RTM), ensuring a clear and comprehensive understanding of system requirements.

- Chapter 5: System Design

This chapter describes the system design of the BI Voice Agent in detail. It includes the overall system architecture, data flow between components, and the design of core modules such as voice input processing, AI reasoning, database interaction, and visualization integration. This chapter serves as a blueprint for implementing the system.

- Chapter 6: Practical Implementation

This chapter covers the practical implementation of the system, including the tools, frameworks, and technologies used. It explains the implementation of AI components (such as speech-to-text, NLP processing, and SQL generation), system interfaces, and integration with databases and visualization tools. It also presents test execution

results and the final Requirement Traceability Matrix (RTM), demonstrating the consistency between requirements and implementation.

### ***7.3 Summary***

This chapter provides a clear guide to understanding the organization of the report and the role of each chapter in documenting the development of the BI Voice Agent system. By following this structured approach, the report ensures a logical progression from conceptual foundations and system analysis to design, implementation, and evaluation, resulting in a comprehensive and well-documented project.

## ***References***

### **[1] ThoughtSpot.**

ThoughtSpot Official Website. Available at:

<https://www.thoughtspot.com/>

### **[2] Databricks.**

Databricks AI/BI Genie. Available at:

<https://www.databricks.com/product/ai-bi/genie>

### **[3] Microsoft.**

Microsoft Power BI – Business Intelligence Tool. Available at:

<https://www.microsoft.com/en-us/power-platform/products/power-bi>

### **[4] Vanna AI.**

Vanna.AI Official Website. Available at: <https://vanna.ai/>

### **[5] Sequel.sh.**

Sequel.sh – Natural Language to SQL Platform. Available at:

<https://sequel.sh/>

### **[1] OpenRouter.**

OpenRouter – Unified API for Large Language Models. Available at:

<https://openrouter.ai/>

### **[2] OpenAI.**

ChatGPT: Large Language Model for Natural Language Understanding and Generation. Available at: <https://chat.openai.com/>

### **[3] OpenAI.**

Whisper: Speech-to-Text Model for Automatic Speech Recognition.  
Available at:  
<https://openai.com/research/whisper>

### **[4] ClickHouse.**

ClickHouse – Column-Oriented Database Management System.  
Available at:  
<https://clickhouse.com/>

### **[5] Metabase.**

Metabase – Open Source Business Intelligence and Analytics Platform. Available at:  
<https://www.metabase.com/>

### **[6] Apache Kafka.**

Apache Kafka – Distributed Event Streaming Platform. Available at:  
<https://kafka.apache.org/>

### **[7] Python.**

Python Programming Language – Official Documentation. Available at:  
<https://www.python.org/>

## [8] Django.

Django Web Framework – Official Documentation. Available at:  
<https://www.djangoproject.com/>