VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY. PUNE

TY Sem – I : Project – I

TY – A 2025-26

Industry Project

Software Requirement Sheet

“Fluid Controls : Effective Information Flow System”

ROLL NO.	*PRN*	*NAME*
331011	22310396	ANISH SUJIT PATIL
331041	22311279	NIRAMAY SURESH BADGUJAR
331002	22310020	SHREYASH GAJANAN NALE
331006	22310205	DARSHAN DATTATRAY DHANVATE

Guided By:

Dr. Shalini Wankhade

Project Given by: Fluid Controls Ltd.

Project Title: Effective Information Flow System

---

1. Introduction

1.1 Purpose

This SRS describes the functional and non-functional requirements for the "Effective Information Flow System," which enables timely and accurate exchange of data and instructions among Admins, Supervisors, and Operators in a workforce environment.

Objective**:** "To achieve timely and accurate exchange of data and instructions by intelligently identifying and allocating the most effective manpower resources."

1.2 Scope

The system supports:

- User management & authentication
- Task creation, assignment, and status tracking
- Real-time notifications and optional chat
- Centralized document/instruction repository
- Manpower skill mapping and availability dashboard
- Audit logging and analytics with report export

1.3 Definitions, Acronyms, and Abbreviations

- **Admin**: System administrator with highest privileges
- **Supervisor**: Middle-tier user who assigns tasks
- **Operator**: End-user who executes tasks
- **JWT**: JSON Web Token
- **SRS**: Software Requirements Specification

1.4 References

- ISO/IEC/IEEE 29148:2018 Systems and Software Engineering — Life Cycle Processes — Requirements Engineering

---

2. Overall Description

2.1 Product Perspective

This is a web-based, modular application composed of:

- Frontend UI (React/Vue)

- REST & WebSocket backends
- Relational databases (e.g., MySQL/PostgreSQL)
- Optional cloud storage for files

2.2 User Classes and Characteristics

User Class	Description	Key Permissions
Admin	Manages users, roles, global settings	Create/Edit users, view/export all analytics
Supervisor	Assigns tasks, monitors team, uploads instructions	Create/Edit tasks, view dashboards, upload docs
Operator	Receives tasks, updates status, views instructions	View/complete tasks, comment, download documents

2.3 Operating Environment

- Modern web browsers (Chrome, Firefox, Edge)
- Server: Linux-based VM or container
- Database server: MySQL/PostgreSQL

2.4 Design & Implementation Constraints

- Must use JWT for stateless auth
- Role-based middleware on all protected endpoints
- Document storage must support versioning

2.5 Assumptions and Dependencies

- Email/SMS gateway available for password resets and alerts
- WebSocket service (e.g., Socket.IO) or Pusher for real-time events

---

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interface

- Forms for registration, login, profile, task creation, status update
- Dashboards: task list, manpower calendar, analytics charts

3.1.2 Hardware Interfaces

- No special hardware; runs on standard server/VM

3.1.3 Software Interfaces

- REST API endpoints (JSON)
- WebSocket channels for notifications/chat

- Email/SMS API

3.1.4 Communications Interfaces

- HTTPS for all traffic
- WebSockets over WSS

3.2 Functional Requirements

3.2.1 User Management & Authentication

- **FR-UM-1**: Admin can register Supervisors/Operators (name, email, role, skills).
- **FR-UM-2**: Users can log in with email/password; receive JWT on success.
- **FR-UM-3**: Role-based middleware restricts access per endpoint.
- **FR-UM-4**: Users can view/edit their profiles and upload a picture.
- **FR-UM-5**: Password change and password reset via secure email link.

3.2.2 Task Assignment

- **FR-TA-1**: Supervisors can create tasks (title, description, deadline, attachments, priority).
- **FR-TA-2**: Supervisors can assign tasks to one or more Operators based on skill tags.
- **FR-TA-3**: Tasks can be edited or reassigned.
- **FR-TA-4**: Operators and Supervisors can view/filter tasks by status.
- **FR-TA-5**: System sends alerts when deadlines approach.

3.2.3 Real-Time Notifications & Chat

- **FR-NT-1**: On task assignment or update, Operators receive real-time push via WebSocket.
- **FR-NT-2**: Instruction changes trigger broadcast alerts to relevant users.
- **FR-NT-3** (Optional): Users can send/receive chat messages in real time; stored in chat history.

3.2.4 Task Status Management

- **FR-TS-1**: Operators can mark tasks “In Progress,” “On Hold,” or “Completed,” optionally adding comments.
- **FR-TS-2**: Supervisors can view a live dashboard of task statuses with filters.
- **FR-TS-3**: Each task has a comment/thread for communication.

3.2.5 Centralized Instruction Repository

- **FR-CR-1**: Admin/Supervisor can upload documents (PDF/Image) with metadata (title, version, tags).
- **FR-CR-2**: System maintains version history; older versions retrievable.
- **FR-CR-3**: Users can search/filter by keyword, tag, or version.

- **FR-CR-4**: Role-based access to download or view.

3.2.6 Manpower Resource Mapping

- **FR-MP-1**: Operator profiles include skill tags and current availability status.
- **FR-MP-2**: Supervisor/Admin can update availability in real time.
- **FR-MP-3**: Visual dashboard (grid or calendar) displays manpower allocation.
- **FR-MP-4**: Users can search for available manpower by skill.

3.2.7 Audit Logs & Analytics

- **FR-AA-1**: System logs all user actions (task updates, uploads, status changes) with timestamp.
- **FR-AA-2**: Analytics engine computes task performance (completion rates, delays).
- **FR-AA-3**: Computes user performance metrics (workload, efficiency).
- **FR-AA-4**: Users can export reports (CSV/PDF).

3.3 Performance Requirements

- API response time ≤ 200 ms under normal load.
- WebSocket event delivery latency ≤ 100 ms.
- Full-text search queries return results < 500 ms.

3.4 Security Requirements

- Passwords hashed with bcrypt/Argon2.
- All endpoints protected by HTTPS.
- Role checks on every protected route.

3.5 Quality Attributes

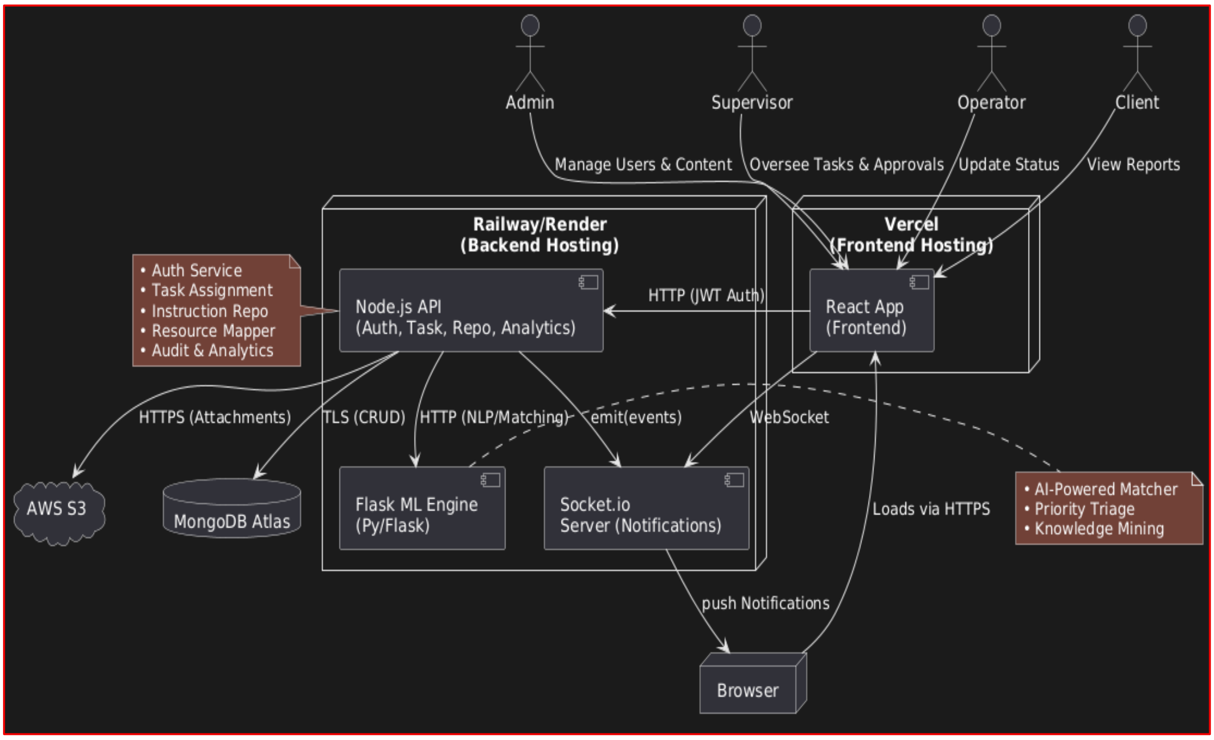
- **Scalability**: Horizontal scaling of WebSocket and REST services.
- **Reliability**: ≥ 99.5% uptime.
- **Maintainability**: Modular codebase with clear separation of concerns.

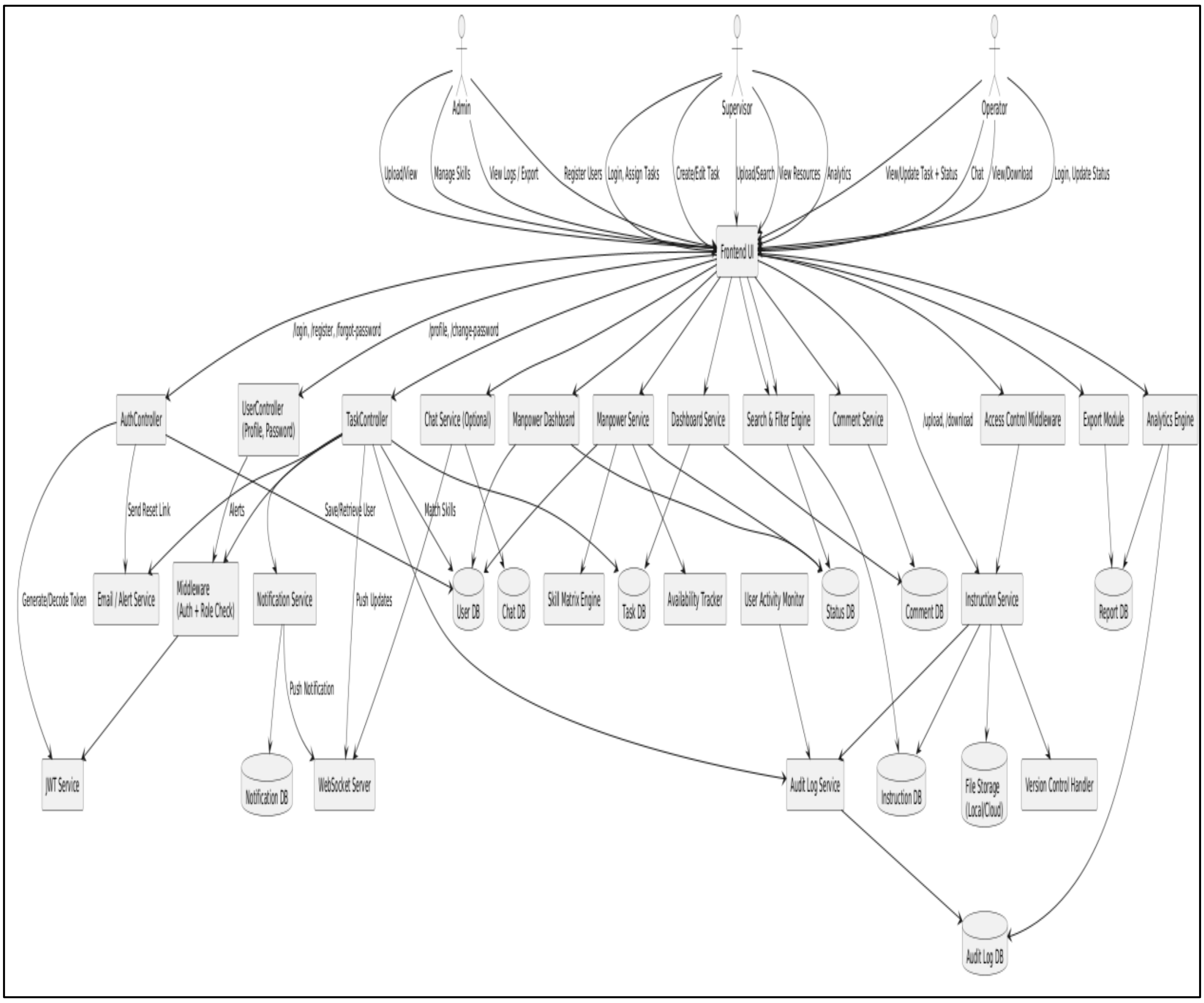
3.6 Design Constraints

- Use PlantUML for all architecture diagrams.
- Frontend in React (with Tailwind CSS), backend in Node.js (Express).

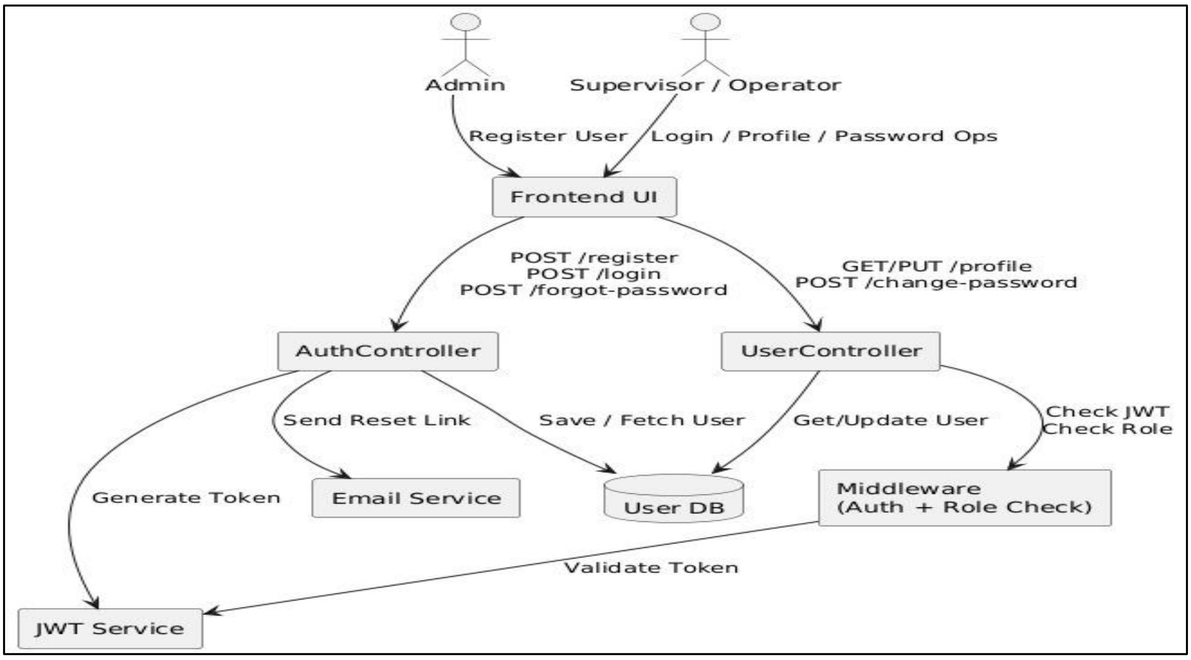
---

4. System Architecture:

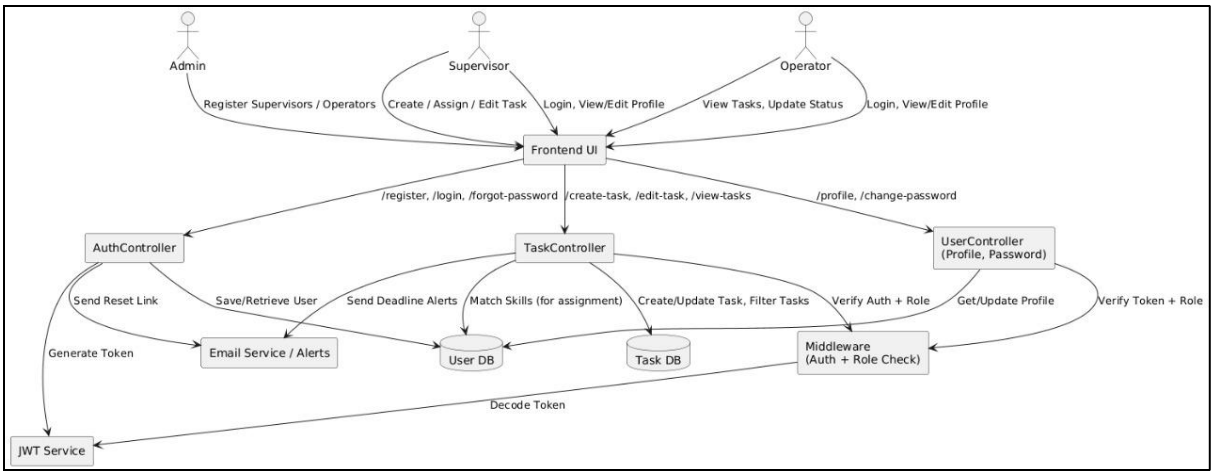




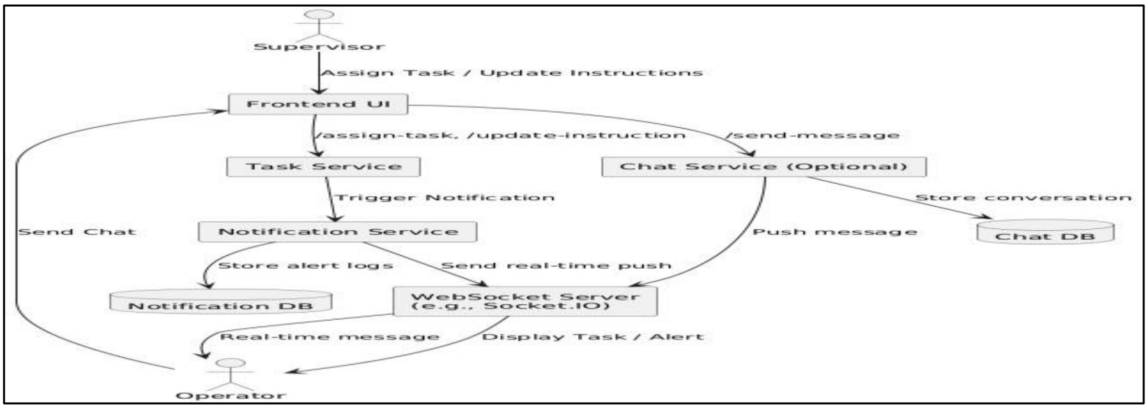
4.1 Module 1: User Management & Authentication module



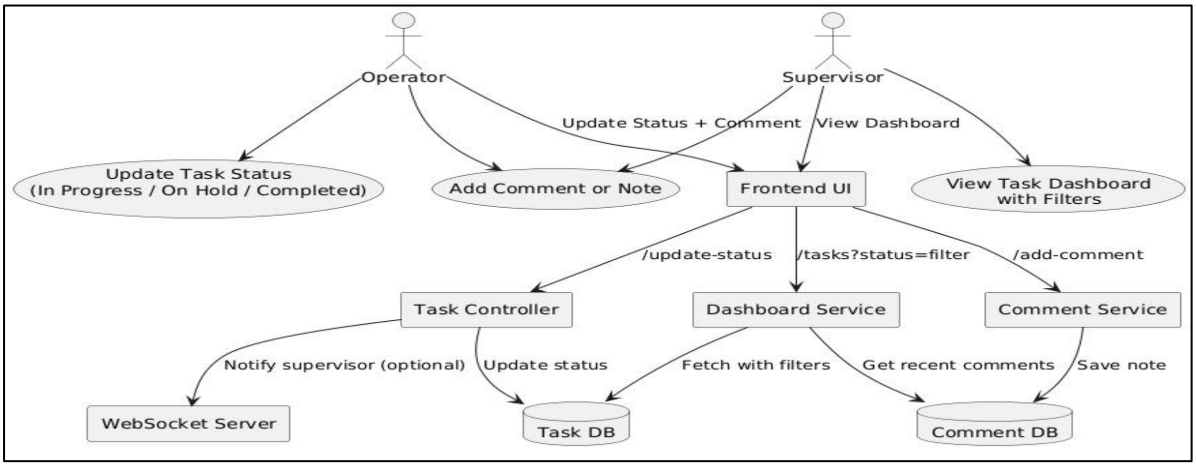
4.2 Module 2 : Task Assignment Module



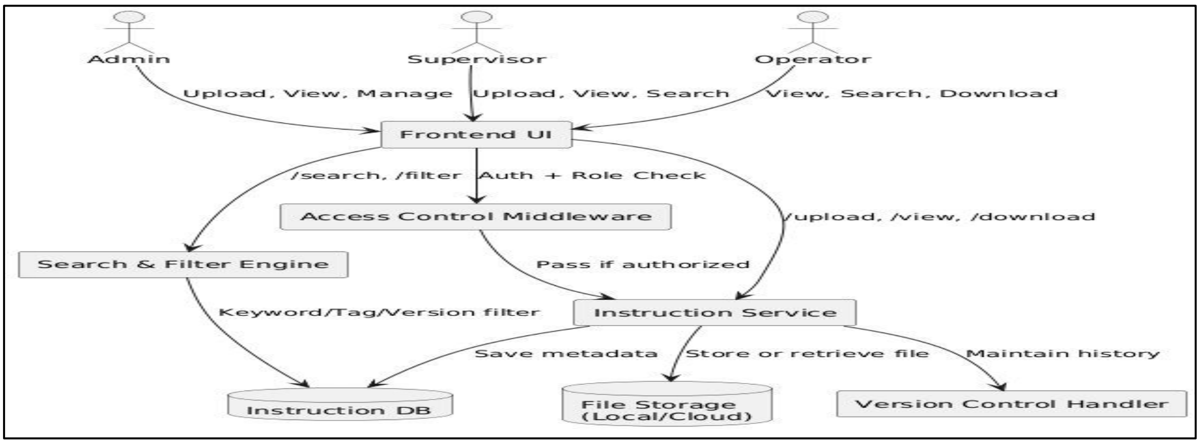
4.3 Module 3 : Real Time Notification module



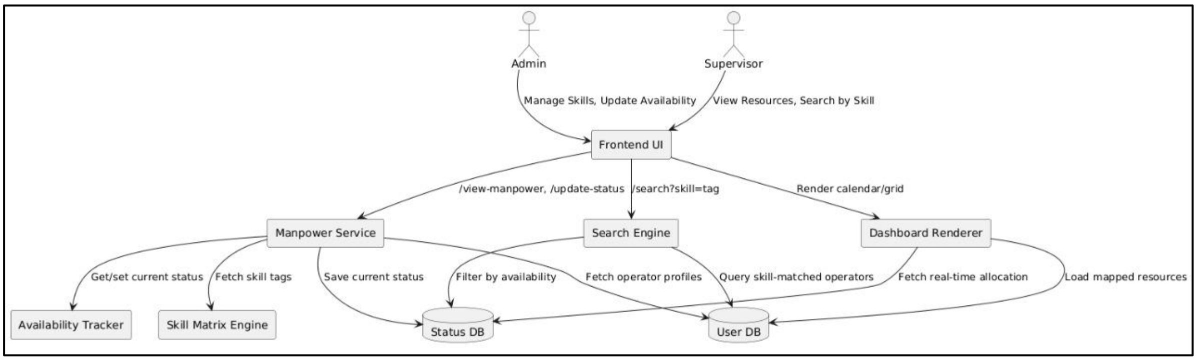
4.4 Module 4 :Task Status Management



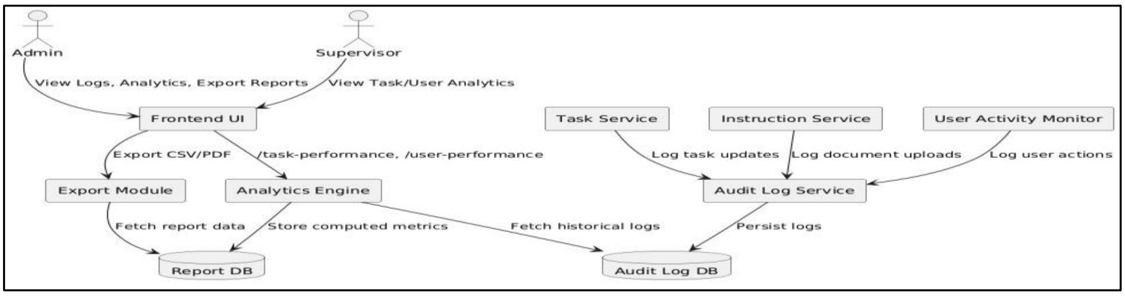
4.5 Module 5: Centralized Instruction Repository – Component



4.6 Module 6 : Manpower Resource Mapping



4.7 Module 7 : Audit Logs & Analytics



5. Appendices

5.1 Glossary

- **Task**: A work item assigned to Operators.
- **Instruction**: Reference document or guideline uploaded to the repository.

5.2 Document Revision History

Date	Version	Description	Author
2025-07-25	1.0	Initial draft	Anish Patil

Thank You!