



SECJ2203: Software Engineering

System Documentation (SD)

**QMS-Flow - Integrated Quality Management System for
Avialite Sdn. Bhd.**

Version 3.0

16 January 2026

Faculty of Computing

Prepared by: Tech TiTan

Revision Page

a. Overview

This system documentation covers the introduction, system requirements, personas, system features, launch phase, user story details, performance and design constraints.

b. Target Audience

Avialite management, auditors, engineers, administrator and course instructors.

c. Project Team Members

Member Name	Role	Task	Status
Rasyid	Team Leader/Analyst	System Features and Launch Phase	Complete
Rafiq	Project Manager	Performance & Design Cons	Complete
Hazim	QA Engineer	Introduction	Complete
Afiq Shahir	Developer	User Story	Complete
Afiq Irfan	Developer	Persona	Complete

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Muhammad Abdul Rasyid Bin Murad	Completed Chapter 1 & 2, Section 02	07/12/2025
2.0	Muhammad Afiq Irfan bin Zuraimi	Completed Section 3 until Section 7	4/1/2026
3.0	Muhammad Afiq Shahir Bin Abdul Rahim	Completed Section 8 and Appendix A: Traceability Matrix	16/1/2026

Table of Contents

1	Introduction	3
	1.1 Purpose	3
	1.2 Scope	4
	1.3 Definitions, Acronyms and Abbreviations	5
	1.4 References	6
	1.5 Overview	7
2	Specific Requirements	8
	2.1 Persona	9
	2.1.1 Persona 1 (Engineer)	
	2.1.2 Persona 2 (Manager)	
	2.1.3 Persona 3 (Admin)	
	2.1.4 Persona n (Auditor)	
	2.2 System Features	22
	2.3 Launch Phase	43
	2.4 User Story Details	45
	2.4.1 US001: User Story <User Story 1>	
	User Story description of US001	
	Activity Diagram of US001	
	System Sequence Diagram of US001	
	2.4.2 US002: User Story <User Story 2>	
	User Story description of US002	
	Activity Diagram of US002	
	System Sequence Diagram of US002	
	2.4.3 US003: User Story <User Story 3>	
	User Story description of US003	

		<p>Activity Diagram of US003</p> <p>System Sequence Diagram of US003</p> <p>2.4.4 US004: User Story <User Story 4></p> <p>User Story description of US003</p> <p>Activity Diagram of US003</p> <p>System Sequence Diagram of US00n3</p> <p>2.4.5 US005: User Story <User Story 5></p> <p>User Story description of US005</p> <p>Activity Diagram of US005</p> <p>System Sequence Diagram of US005</p> <p>2.4.6 US006: User Story <User Story 6></p> <p>User Story description of US006</p> <p>Activity Diagram of US006</p> <p>System Sequence Diagram of US006</p> <p>2.4.7 US007: User Story <User Story 7></p> <p>User Story description of US007</p> <p>Activity Diagram of US007</p> <p>System Sequence Diagram of US007</p> <p>2.4.8 US008: User Story <User Story 8></p> <p>User Story description of US008</p> <p>Activity Diagram of US008</p> <p>System Sequence Diagram of US008</p> <p>2.4.9 US009: User Story <User Story 9></p> <p>User Story description of US009</p> <p>Activity Diagram of US009</p> <p>System Sequence Diagram of US009</p> <p>2.4.10 US010: User Story <User Story 10></p>	
--	--	---	--

		User Story description of US010 Activity Diagram of US010 System Sequence Diagram of US010 2.4.11 US011: User Story <User Story 11> User Story description of US011 Activity Diagram of US011 System Sequence Diagram of US011 2.4.12 US012: User Story <User Story 12> User Story description of US0012 Activity Diagram of US0012 System Sequence Diagram of US0012 2.4.13 US013: User Story <User Story 13> User Story description of US013 Activity Diagram of US013 System Sequence Diagram of US013	
	2.5	Performance and Other Requirements	73
	2.6	Design Constraints	75
3	System Architectural Design		77
	3.1	Architectural Style and Rationale	77
	3.2	Component Model	80
4	Detailed Description of Components		81
	4.1	Complete Package Diagram	81
	4.2	Detailed Description	82
	4.2.1	P001: Audit Management Module Subsystem	82
	4.2.2	P002: Document & Change Management with KPI Dashboard Subsystem	85
	4.2.3	P003: Admin & System Support Module Subsystem	91

		4.2.4	P004: CAPA Management Module Subsystem	95	
		4.2.5	P005: Risk & Incident Management Module	100	
5	Data Design			103	
	5.1	Data Description			103
	5.2	Data Dictionary			105
6	User Interface Design			111	
	6.1	Overview of User Interface			111
	6.2	Screen Images			112
7	Requirements Matrix			129	
8	Test Cases			133	

1. Introduction

1.1 Purpose

This System Documentation (SD) / Software Requirements Specification (SRS) describes the functional and non-functional requirements, scope, interfaces, and constraints for **QMS-Flow, an Integrated Quality Management System** tailored for Avialite Sdn. Bhd. The purpose of this document is to provide a single, authoritative reference for stakeholders (Avialite management, engineers, and auditors), the development team (Tech TiTan), and course instructors to ensure that the system design and implementation satisfy the agreed requirements and meet ISO 9001:2015 compliance needs. Key deliverables defined by this document include the functional requirement set, user personas, use cases, performance and security requirements, and launch/sprint planning.

1.2 Scope

The QMS-Flow software product will be a web-based, centralized quality management platform that replaces manual and fragmented QMS processes at Avialite Sdn. Bhd. The system will:

- Provide centralized **Document Control** with versioning, metadata, approval workflows, and audit trails.
- Provide a **Traceability** capability linking customer complaints, CAPA records, training logs, and equipment calibration history.
- Support **CAPA (Corrective and Preventive Action)** lifecycle: create, assign, track, verify, and close CAPAs.
- Provide **Audit Management** tools: schedule audits, record findings, link findings to CAPAs, and track follow-ups.
- Manage **Training Records** and training assignments with completion tracking and linkage to CAPA or competence requirements.
- Manage **Equipment** registry and calibration schedules with alerts and history.
- Offer **Dashboards & KPI Reporting** for management oversight (CAPA closure rate, audit completion, training compliance, calibration status). Out of scope (initial release): full ERP/production control, manufacturing process automation, and third-party QMS vendor integrations beyond basic import/export (CSV/PDF) for legacy compatibility. The system is designed to integrate with Avialite's existing on-premise environment and preserve legacy workflows where possible to reduce disruption.

Primary benefits and objectives

- Improve document accuracy and prevent outdated SOP usage.
- Reduce compliance risk and speed corrective actions.
- Provide traceable audit evidence to support ISO 9001:2015 certification readiness.
- Increase operational visibility via real-time dashboards for management decision-making.

1.3 Definitions, Acronyms and Abbreviations

QMS	Quality Management System.
QMS-Flow	The name of the proposed Integrated Quality Management System for Avialite.
SOP	Standard Operating Procedure.
CAPA	Corrective and Preventive Action.
RBAC	Role-Based Access Control.
UAT	User Acceptance Testing
KPI	Key Performance Indicator.
RTO	Recovery Time Objective.
RPO	Recovery Point Objective.
ISO 9001:2015	International Standard for Quality Management Systems (clause references used throughout the design and requirements).

1.4 References

Primary source materials used to prepare this SRS:

- **Project 1 (Proposal)** - QMS-Flow Project Proposal (Tech TiTan / Avialite). This proposal provides background, a problem statement, proposed modules (Document Control, Traceability, Audit Management, CAPA, Training, and Equipment), and the project schedule referenced for scope and priorities.
- **Project SE SRS Template** - SECJ2203 SRS/SD template and earlier SRS draft for QMS-Flow.
- **Project SE SDD Template** - SECJ2203 SDD/SD template and earlier SDD draft for QMS-Flow.
- External standards (informative): ISO 9001:2015 (documented information and performance evaluation clauses) - referenced in the Requirements and Design Constraints sections.

1.5 Overview

The documentation is organized into 7 key chapters that define the system's lifecycle and technical architecture:

1. Introduction (Section 1) : Sets the foundation by defining the project's purpose, the scope of web-based centralized platform, and essential definitions and references.
2. Specific Requirements (Section 2) : Details the functional needs of the system through user personas and use cases. It outlines the six sprint launch phase and provides detailed descriptions of 13 primary use cases.
3. System Architectural Design (Section 3) : describes the layered architecture style chosen to ensure independent development of modules and localized system maintenance.
4. Detailed Description of Components (Section 4) : Provide a deep dive into the system's subsystems, including package diagrams, class diagrams, and sequence diagrams.
5. Data Design (Section 5) : Explains the data architecture, including description of database entities and a detailed data dictionary for attributes.
6. User Interface Design (Section 6) : Offers an overview of the web-based design, focusing on Role-Based Access Control (RBAC) and providing screen mockups for various user dashboards.
7. Requirement Matrix (Section 7) : Presents a traceability matrix that links subsystems and use cases to specific database entities and sequence diagrams, ensuring all requirements are accounted for in the design.

2.0 Specific Requirements

This section of the System Documentation (SD) explains the Software Requirements Specification (SRS) for the **QMS-Flow System**. It provides a clear and complete description of all system requirements to ensure that the design and development of the system meet the needs of Avialite Sdn. Bhd. The requirements help developers understand what the system must do, and they guide testers to check whether the system works correctly. These requirements cover both functional and non-functional parts of the system and support the main users, such as staff, supervisors, quality managers, and auditors.

This section also ensures that the system supports the goals of improving document control, tracking complaints, managing training, and ensuring ISO 9001:2015 compliance.

Key points in this section include:

1. Inputs and Outputs

- All inputs, such as user actions, uploaded documents, complaint reports, training updates, or calibration requests.
- All outputs such as notifications, approval status updates, complaint progress, training results, and calibration reminders.

2. System Function

- A breakdown of the main functions the system will perform, such as document approval, complaint handling, CAPA tracking, training management, and equipment calibration management
- These functions describe how the system responds to user inputs and how it produces results or updates.

3. User-Centric Requirement

- Requirements that can be seen or experienced by users such as staff, supervisors, QA managers, and auditors.
- This ensures the system is easy to use, clear, and helpful in daily work operations.

4. Compliance and Standards

- Requirements that help the system meet ISO 9001:2015 standards, including document control, traceability, audit trails, and proper record keeping.
- This ensures the system supports quality management and follows industry guidelines.

2.1 Persona

Different types of users with different roles and skills will use the QMS-Flow System. These include production engineer, admin, quality managers, and auditors. Most engineers have basic computer skills and need a simple system that they can understand easily. Admins and quality managers are more experienced users who require system features to plan documents, training, complaints, and equipment records. Auditors need to review information and check compliance. Analyzing these users with these different roles helps the development team design a system that matches their abilities to support their daily tasks. By knowing their needs, behavior, and challenges, the system can be made user-friendly, clear, and suitable for the work environment at Avialite Sdn. Bhd.

2.1.1 Persona 1 (Engineer)

The Engineer is central to maintaining ISO 9001-compliant operational records, daily registering new documents, updating equipment, tracking training, and initiating CAPA. Their simultaneous technical and documentation duties require a fast, accurate, and structured data entry system. To support productivity and compliance, the interface must be clear, consistent, and easy to use, minimizing complexity and manual errors.

2.1.1.1 Description

Name	Amir Hakim
Age	27
Role	Engineer (uses modules for Document Control, Training Records, CAPA, and Equipment Management)
Technical Skill	Moderate. Able to use computers for daily tasks, fill forms, upload files, and navigate structured modules. Not specialized in IT but learns new systems quickly.
Experience	3–5 years in engineering operations involving maintenance, production processes, and reporting non-conformances. Familiar with equipment handling, documentation, and safety compliance.
Motivations	Wants a fast and simple system to manage documents, track training status, update equipment records, and initiate CAPA without dealing with manual paperwork. Motivated by efficiency, compliance, and reducing repetitive manual tasks.
Goals	<ul style="list-style-type: none">• Maintain accurate equipment records and calibration dates.• Quickly initiate CAPA when problems arise.• Ensure training records for staff are updated and compliant.

Behaviours	<ul style="list-style-type: none"> • Upload documents immediately after preparing them. • checks equipment statuses regularly. • Follow clear system prompts and guided workflows.
Pain Points	<ul style="list-style-type: none"> • Hard to track which documents are the latest/approved version. • Equipment calibration dates are easily forgotten without notifications. • CAPA processes can be confusing if steps are not clearly guided.

2.1.1.2 User Need

The engineer needs a streamlined interface to upload and manage documents with version control, a guided CAPA initiation linked to complaints, and equipment calibration scheduling and status tracking.

2.1.1.3 User Stories

1. As an Engineer,

I want to upload SOPs with version control so that documents are traceable and current.

2. As an Engineer,

I want to initiate CAPA linked to complaints so that corrective actions are properly tracked.

3. As an Engineer,

I want to manage equipment calibration schedules so that compliance is maintained.

2.1.2 Persona 2 (Manager)

The manager oversees multi-departmental quality assurance, relying on accurate system data for informed decisions, approving documents, monitoring KPIs, and ensuring timely CAPA completion. Effective compliance evaluation requires real-time, up-to-date records and consistent document versions. The system's accuracy directly influences quality outcomes; thus, a structured, easily navigable system is vital for managerial oversight and maintaining ISO 9001 standards.

2.1.2.1 Description

Name	Farah Nadira
Age	35
Role	Quality Manager (oversees approvals, KPI monitoring, compliance, and CAPA oversight)
Technical Skill	Intermediate. Comfortable with dashboards, reports, approvals, and system navigation. Uses multiple QMS tools daily.
Experience	10–12 years in quality assurance, process improvement, audit preparation, and document control oversight.
Motivations	Wants accurate information for decision-making, smooth approval processes, and an organized system that reduces administrative workload.
Goals	<ul style="list-style-type: none">● Monitor KPIs in real time.● Ensure CAPA actions are completed on time.● Maintain compliance across engineering and production teams .
Behaviours	<ul style="list-style-type: none">● Reviews and approves documents promptly.● Cross-checks audit findings and assigns corrective actions.● Communicates with engineers about training & equipment

	status.
Pain Points	<ul style="list-style-type: none">• Overwhelmed by large volumes of documents awaiting approval.• Difficult to track CAPA progress without system reminders.• Manual coordination between departments leads to delays.

2.1.2.2 User Need

The manager needs a dashboard to monitor KPIs across CAPA, audits, training, and equipment. They also need document approval workflows with clear status updates and training assignments and progress tracking tools.

2.1.2.3 User Stories

1. As a Manager,

I want to approve or reject documents so that only validated SOPs are published.

2. As a Manager,

I want to assign training to employees so that competency gaps are addressed.

3. As a Manager,

I want to generate KPI reports so that I can evaluate departmental performance.

2.1.3 Persona 3 (Admin)

The admin maintains system integrity for documentation, training, and equipment tracking. Responsibilities include managing accounts, updating records, ensuring data accuracy, and providing technical support. Admins rely on clear, dependable workflows for system reliability, needing stability, precision, and efficient controls to keep information accurate, accessible, and compliant with ISO 9001. Inaccurate data or system issues severely impede their ability to maintain compliance-ready records.

2.1.3.1 Description

Name	Syed Tariq
Age	42
Role	System Administrator (manages system access, user accounts, and supports Traceability & CAPA module)
Technical Skill	Advanced. Skilled in system configuration, user management, troubleshooting, data validation, and maintaining digital records.
Experience	6–8 years in administrative and system management roles, familiar with QMS platforms, database maintenance, and digital workflows.
Motivations	wants a stable, error-free system that is easy to maintain. Motivated by minimizing user issues, ensuring data accuracy, and maintaining smooth system operations.
Goals	<ul style="list-style-type: none">● Manage user accounts and access permissions efficiently.● Ensure training and equipment records are accurate and updated.● Maintain system data integrity and prevent duplicate or incorrect entries.
Behaviours	<ul style="list-style-type: none">● Regularly updates training and equipment information.

	<ul style="list-style-type: none"> ● Monitors user activity for accuracy and compliance. ● Provides system support for staff who face technical issues.
Pain Points	<ul style="list-style-type: none"> ● High volume of update requests from engineers and managers. ● System downtime or lag affects administrative tasks. ● Difficulty managing large numbers of document or record changes during audit periods.

2.1.3.2 User Need

The Admin requires centralized training record management, CAPA assignment and progress tracking, and integration with audit and equipment modules.

2.1.3.3 User Stories

1. As an Admin,

I want to manage training records so that employee competencies are documented and updated.

2. As an Admin,

I want to monitor CAPA progress so that corrective actions are completed and traceable.

2.1.4 Persona 4 (Auditor)

The auditor is tasked with verifying the company's adherence to ISO 9001 requirements. Performing audits on a periodic basis (monthly, quarterly, or annually), their work hinges on accessing accurate, complete, and traceable documentation. They focus specifically on validating calibration and training records, assessing CAPA effectiveness, and scrutinizing overall paperwork quality. Consequently, the integrity and accuracy of system-generated documents are critical to their function.

2.1.4.1 Description

Name	Yi wen
Age	48
Role	Internal / External Quality Auditor
Technical Skill	High experience with digital QMS platforms, audit software, and compliance tools
Experience	More than 20 years performing internal and external audits in manufacturing and service sectors
Motivations	Responsible for reviewing whether the organization conforms with ISO 9001 requirements, performs scheduled audits and evaluates paperwork related to complaints , investigations, corrective actions (CAPA), calibration, training records, and management review outputs.
Goals	<ul style="list-style-type: none">• Review complaint and CAPA history with complete traceability• Before closing, confirm the effectiveness of the corrective measures.• Check that training, calibration, and management evaluations are appropriately documented• Detect gaps or unsafe practices early
Behaviours	<ul style="list-style-type: none">• Uses the system during monthly, quarterly, or yearly audits

	<ul style="list-style-type: none"> ● reviews CAPA timelines, evidence, and efficacy tests in-depth. ● Communicates follow-up remedial actions to the Quality Manager ● Cross-checks inquiry notes and attached documents
Pain Points	<ul style="list-style-type: none"> ● Records are often incomplete or stored in multiple places ● Hard to verify effectiveness when evidence is missing ● Calibration and training records are sometimes outdated

2.1.4.2 User Need

The auditor needs tools to schedule audits with scope and reminders, forms to record findings and trigger CAPA if needed, and access to complete CAPA and training for verification.

2.1.4.3 User Stories

1. As an Auditor,

I want to schedule internal and external audits so that compliance checks are planned and resources are allocated. (This traces to UC003: Schedule Audit)

2. As an Auditor,

I want to record audit findings and non-conformances so that CAPAs can be officially initiated. (This traces to UC004: Record Audit Findings)

2.2 System Features

QMS-Flow is designed as a hybrid Quality Management System (QMS) platform that integrates seamlessly into Avialite Sdn. Bhd.'s legacy workflows while replacing manual, fragmented processes with a centralized and automated solution. The system provides a closed-loop compliance cycle aligned with ISO 9001:2015 standards, ensuring that document control, traceability, CAPA, audits and management review are unified in one environment. The system features are illustrated in Figure 2.2.1 below. The detailed description of each module and function is tabulated in Table 2.2.1.

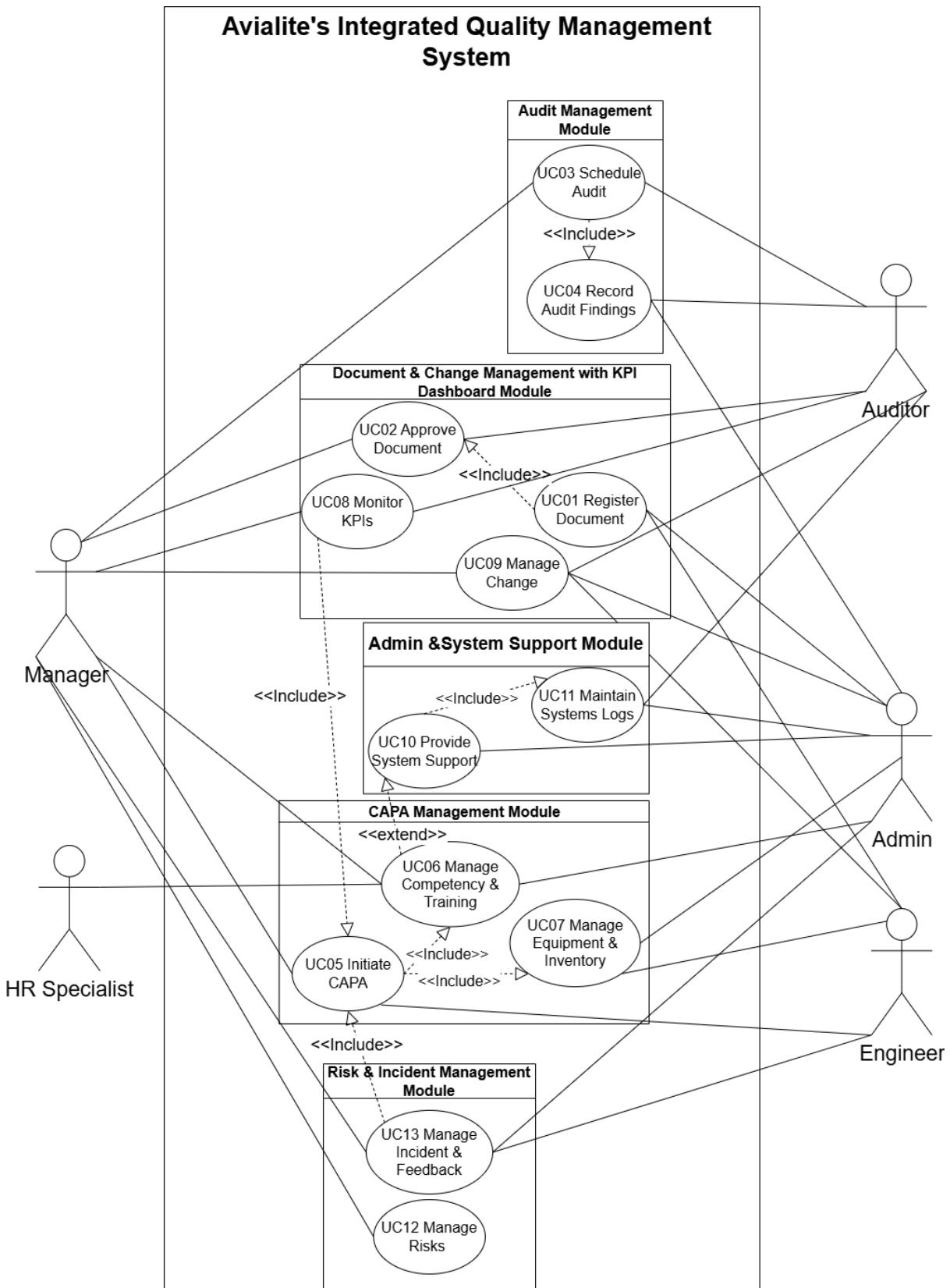


Figure 2.2.1: Use Case Diagram for QMS-FLow

Use Case	Function	Description
UC01	Register Document	Allows users to create and submit new documents into the system for review.
UC02	Approve Document	Enables managers or auditors to approve submitted documents, triggering version control and publication.
UC03	Schedule Audit	Plan and assign internal audits with dates and scope.
UC04	Record Audit Findings	Allows auditors to log findings during an audit, including observations and nonconformities.
UC05	Initiate CAPA	Starts corrective or preventive actions based on audit findings, incidents, or complaints.
UC06	Manage Competency & Training	Assesses staff skills and schedules training to close competency gaps
UC07	Manage Equipment & inventory	Tracks calibration, maintenance, and equipment readiness linked to CAPA.
UC08	Monitor KPIs	Enables managers to view and analyze key performance indicators related to quality objectives.
UC09	Manage Change	Handles change requests, reviews, and approvals to maintain controlled updates in processes or documents.
UC10	Provide System Support	Enables admins to respond to client or staff support requests, including technical issues or access problems.
UC11	Maintain System Logs	Automatically records system activities for traceability, including logins, approvals, and changes.
UC12	Manage Risks	Identify, assess, and mitigate risks that may impact quality or compliance.
UC13	Manage Incident & Feedback	Handle report incidents or customer complaints and track investigations and resolutions.

Table 2.2.1 : Description of Module and Functions for Integrated Quality Management System

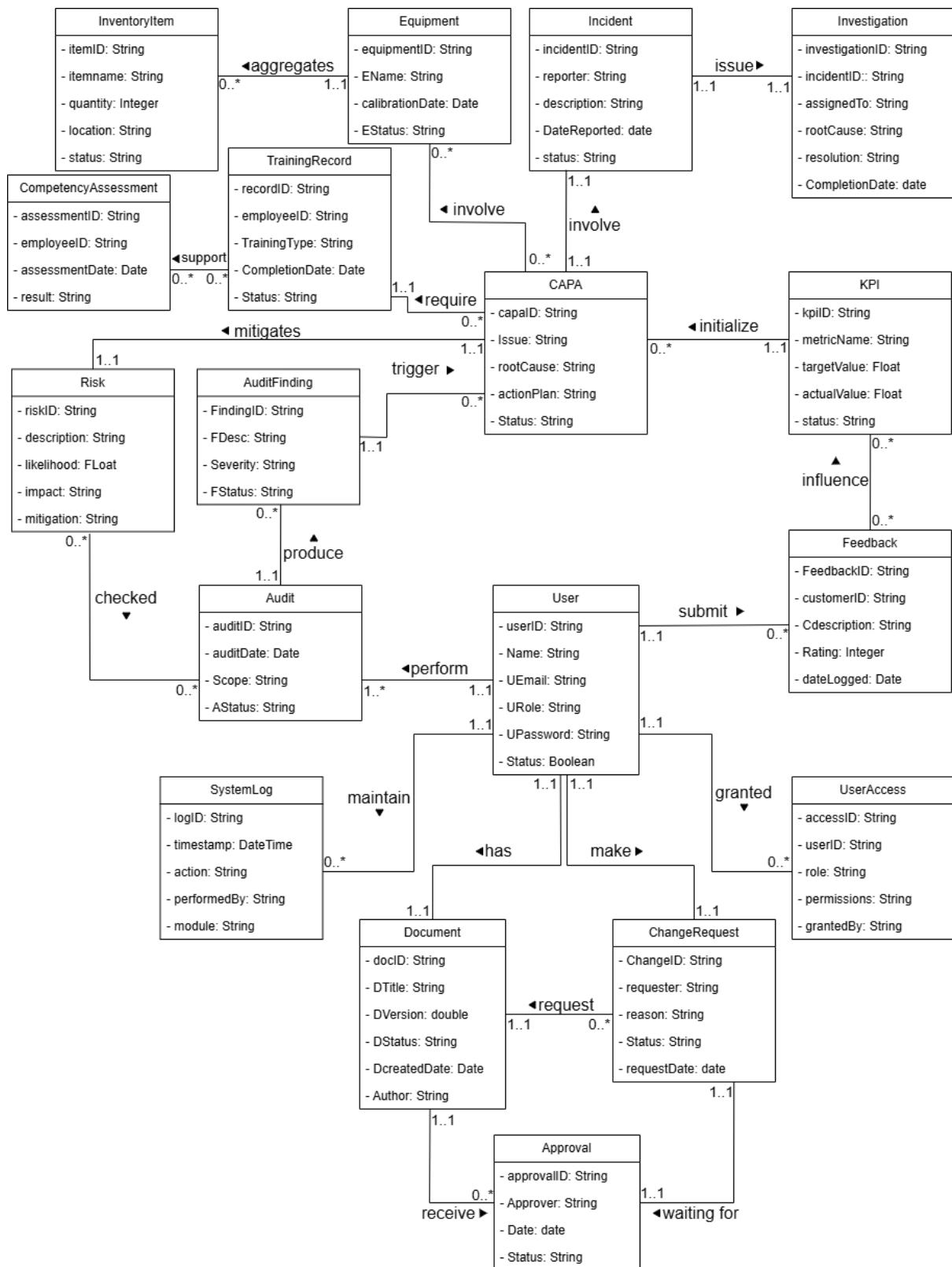


Figure 2.2.2: Domain Model for QMS-Flow

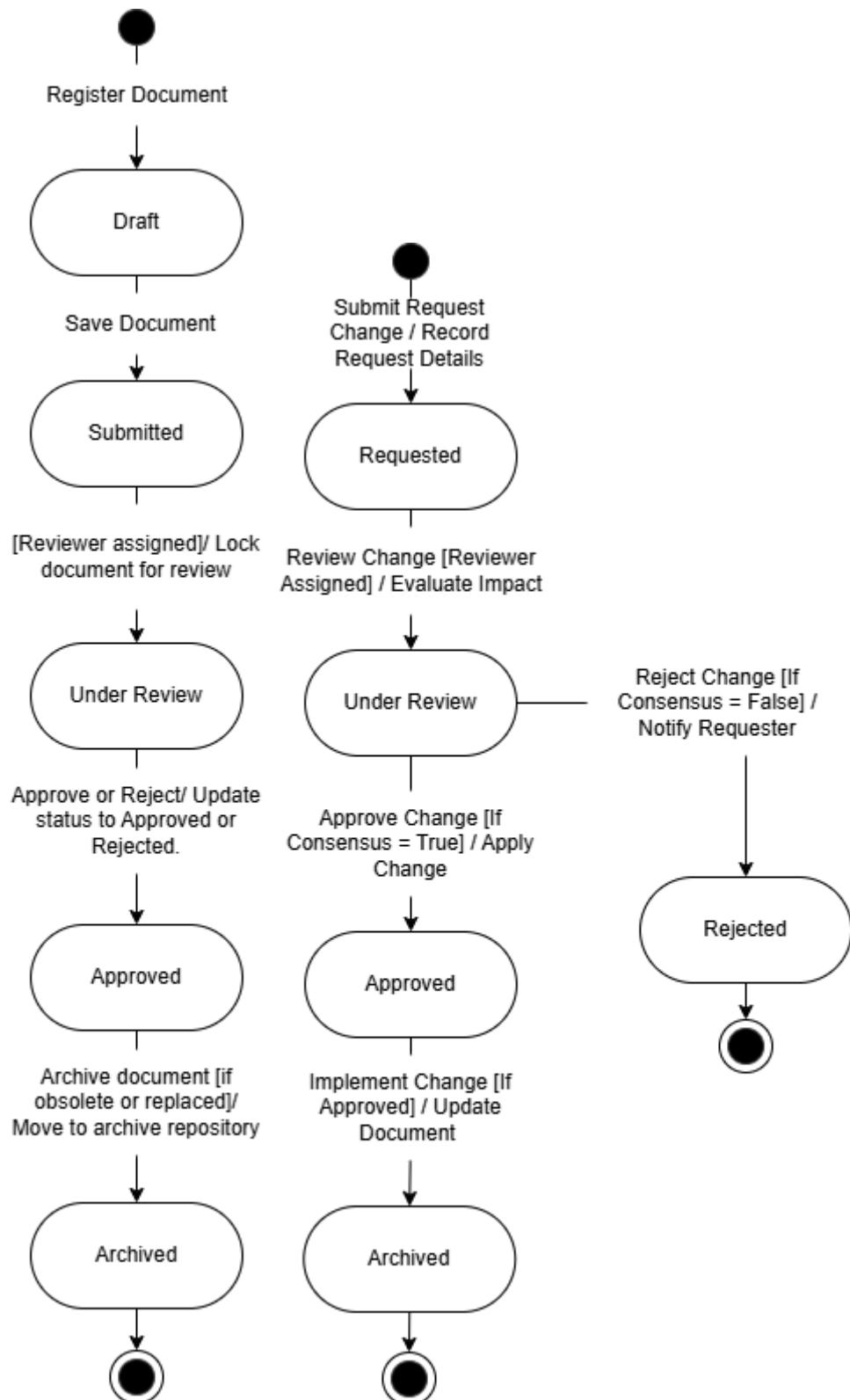


Figure 2.2.3: State Machine Diagram for Document and Change Management with KPI Dashboard Module

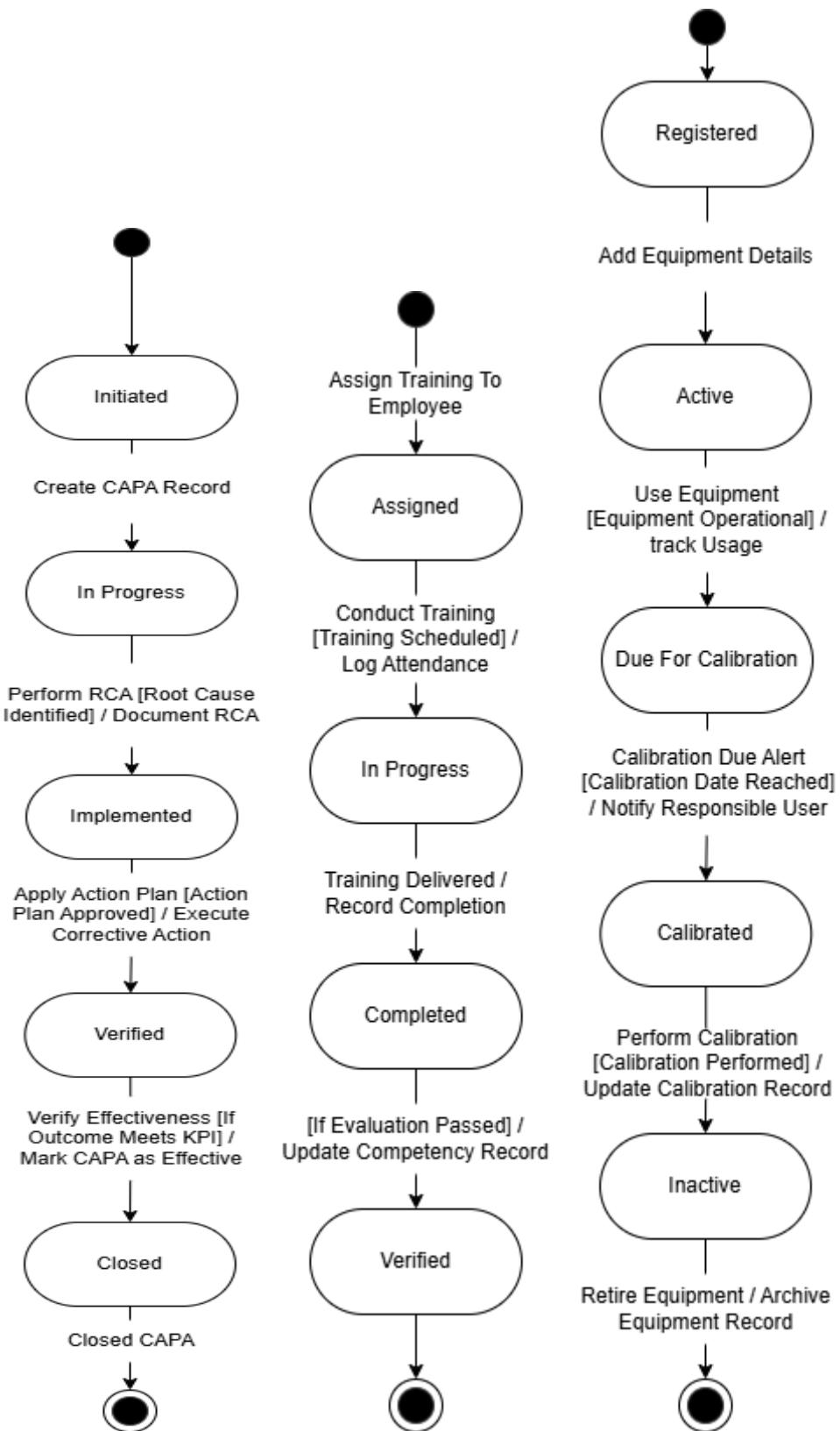


Figure 2.2.4: State Machine Diagram for CAPA Management Module

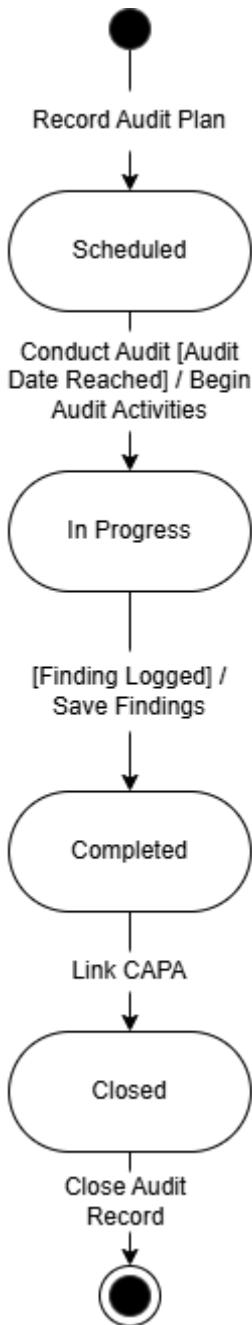


Figure 2.2.5: State Machine Diagram for Audit Management Module

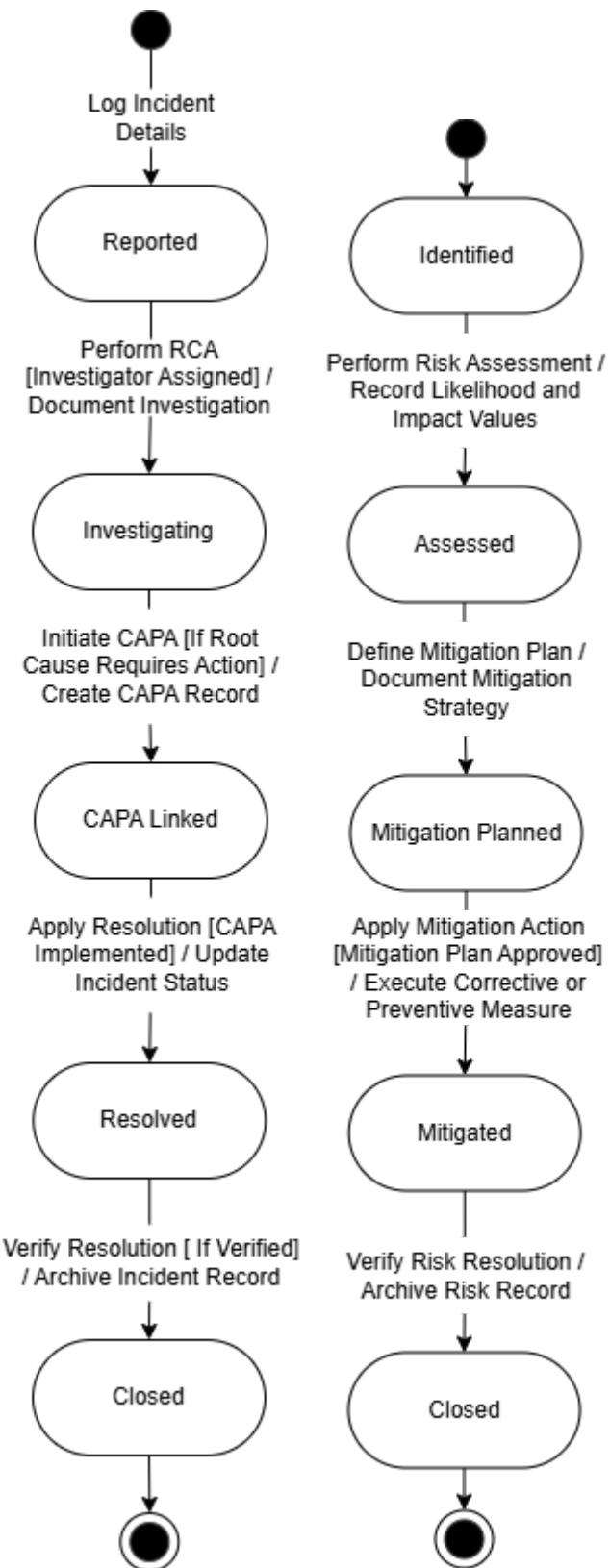


Figure 2.2.6: State Machine Diagram for Risk and Incident Module

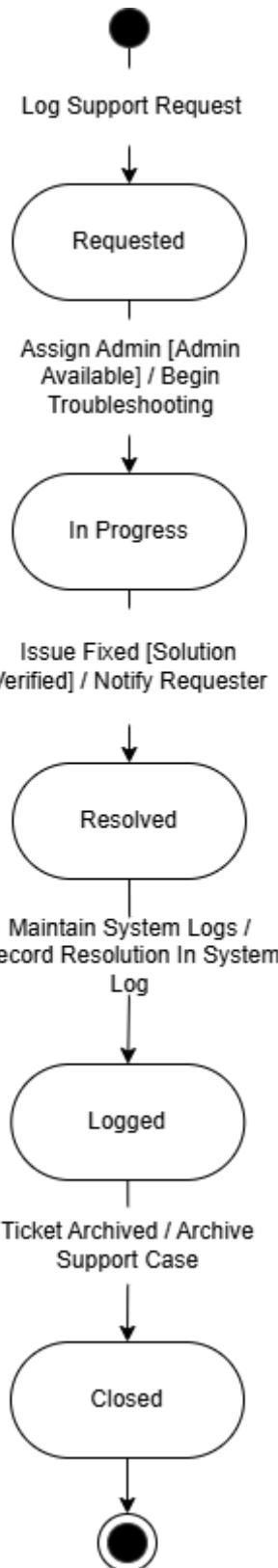


Figure 2.2.7: State Machine Diagram for Admin and System Support Module

Register Document
ID: UC01
Actor: Engineer, Admin
Preconditions: 1. The engineer logged in with document creation rights. 2. Document metadata defined.
Flow of events: 1. Engineer selects “Register Document.” 2. Engineer uploads the SOP or document. 3. System assigns a version number and metadata. 4. Include (Approve Document). 5. The document enters the “Pending Approval” state.
Postconditions: 1. Document stored in repository. 2. Awaiting approval.
Alternative Flows: 1. Whenever the upload fails, the system will prompt a retry. 2. If a duplicate is detected, the system will alert the user.

Table 2.2.2: Use Case Description for Register Document

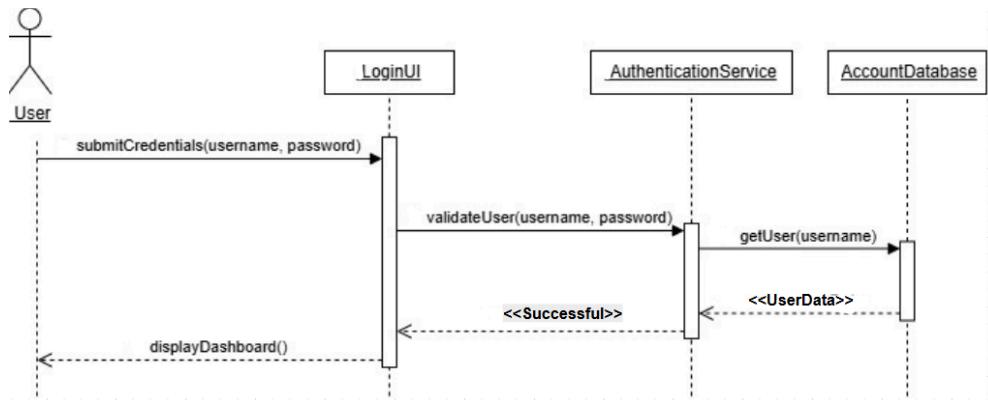


Figure 2.2.8: Sequence Diagram for Register Document

Approve Document
ID: UC02
Actors: Manager, Auditor
Preconditions: 1. Document exists in the “Pending Approval” state.
Flow of events: 1. Manager opens the approval request. 2. Manager reviews the document. 3. Manager selects Approves or rejects with comments. 4. System updates version control and publishes the approved SOP.
Postconditions: 1. The approved document is marked “Active” and traceable.
Alternate Flows: 1. Manager rejects the document and returns it to Engineer for revision.

Table 2.2.3: Use Case Description for Approve Document

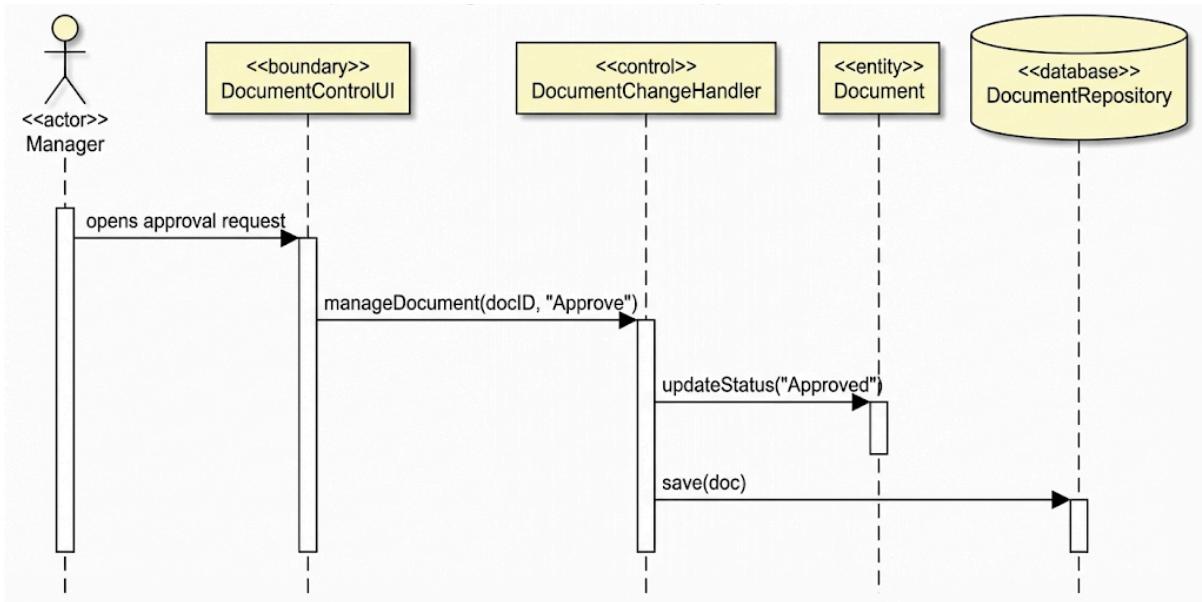


Figure 2.2.9: Sequence Diagram for Approve Document

Schedule Audit
ID: UC03
Actors: Auditor, Manager
Preconditions: 1. Auditor has access rights.
Flow of events: 1. Auditor selects “Schedule Audit.” 2. Include (Record Audit Findings). 3. Auditor defines the scope, date, and department. 4. System sends notifications to relevant staff. 5. Audit entry created in calendar.
Postconditions: 1. The audit schedule is stored and visible in the dashboard.
Alternate Flows: 1. Date conflicts or overlaps will make the system reschedule.

Table 2.2.4: Use Case Description for Schedule Audit

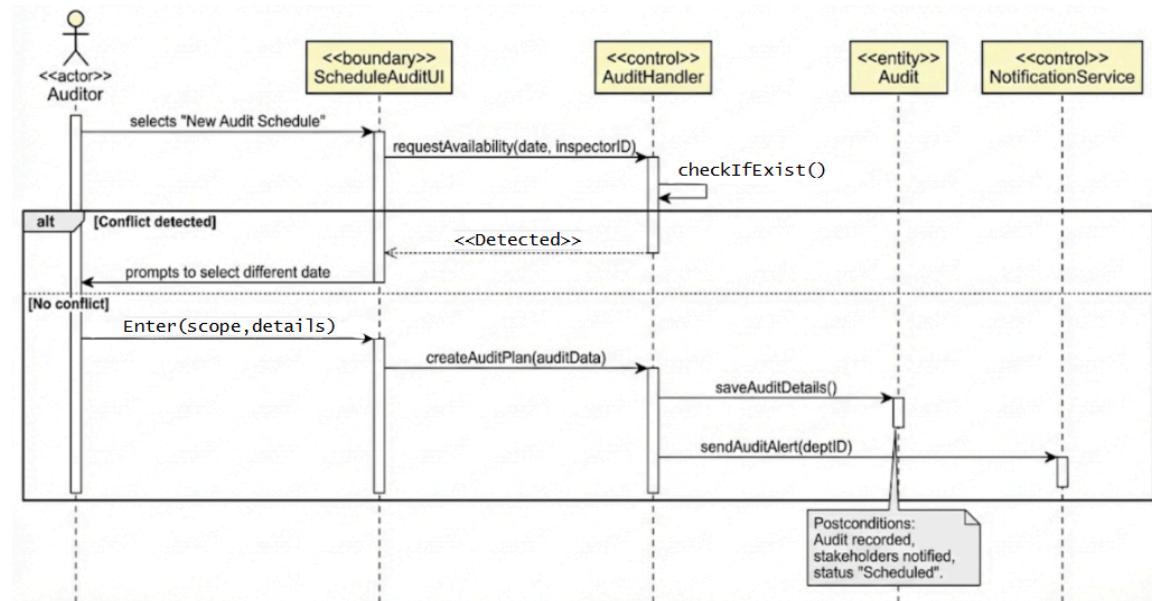


Figure 2.2.10: Sequence Diagram for Schedule Audit

Record Audit Findings
ID: UC04
Actors: Auditor, Admin
Preconditions: 1. Audit is scheduled and in progress.
Flow of events: 1. Auditor logs findings. 2. Auditor enters non-conformance details. 3. System links findings to CAPA module. 4. Notifications sent to Manager.
Postconditions: 1. Store findings and initiate CAPA if required.
Alternate Flows: 1. System prompts for correction if the entry is incomplete.

Table 2.2.5: Use Case Description for Record Audit Findings

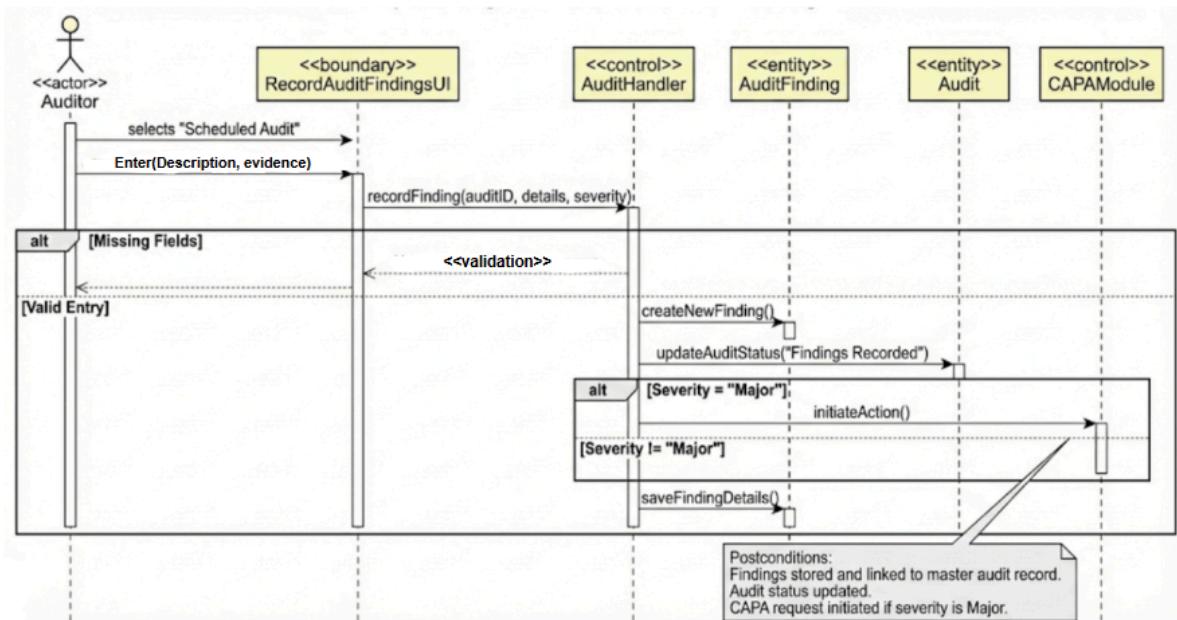


Figure 2.2.11: Sequence Diagram for Record Audit Findings

Initiate CAPA
ID: UC05
Actors: Engineer, Manager
Preconditions: 1. Receive complaint or incident occur or audit finding exists
Flow of events: 1. Engineer selects "Initiate CAPA." 2. Engineer enters the root cause, corrective plan, and responsible person. 3. Include (Manage Competency and Training). 4. Include (Manage Equipment and Inventory). 5. System validates CAPA form. 6. Manager reviews and approves CAPA assignment. 7. CAPA tracked until closure with status updates.
Postconditions: 1. CAPA record was created with traceability.
Alternate Flows: 1. Engineer will revise the plan if the CAPA is rejected.

Table 2.2.6: Use Case Description for Initiate CAPA

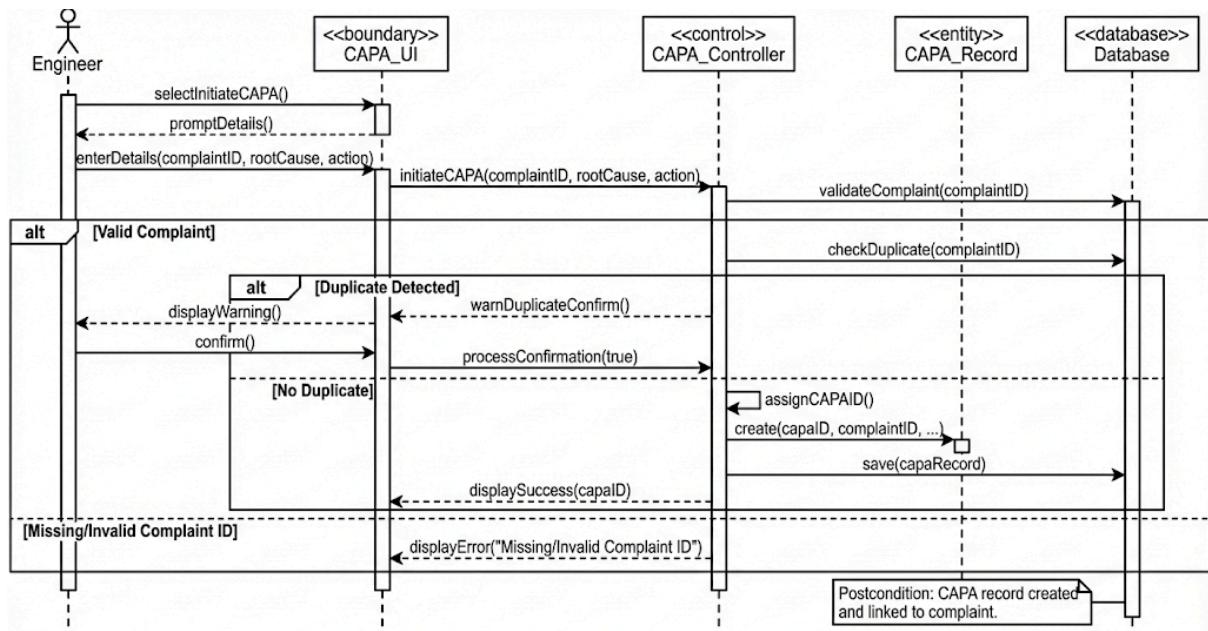


Figure 2.2.12: Sequence Diagram for Initiate CAPA

Manage Competency and Training
ID: UC06
<p>Actors: HR Specialist, Manager, Admin</p>
<p>Preconditions:</p> <ol style="list-style-type: none"> 1. Employee records exist.
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. Manager assigns training. 2. Manager selects the employee and training module. 3. HR Specialist will oversee the training. 4. System records assignment and completion status. 5. Training linked to competency gaps.
<p>Postconditions:</p> <ol style="list-style-type: none"> 1. Updated training record stored.
<p>Alternate Flows:</p> <ol style="list-style-type: none"> 1. Retraining is scheduled if the employee fails to complete the training.

Table 2.2.7: Use Case Description for Manage Competency and Training

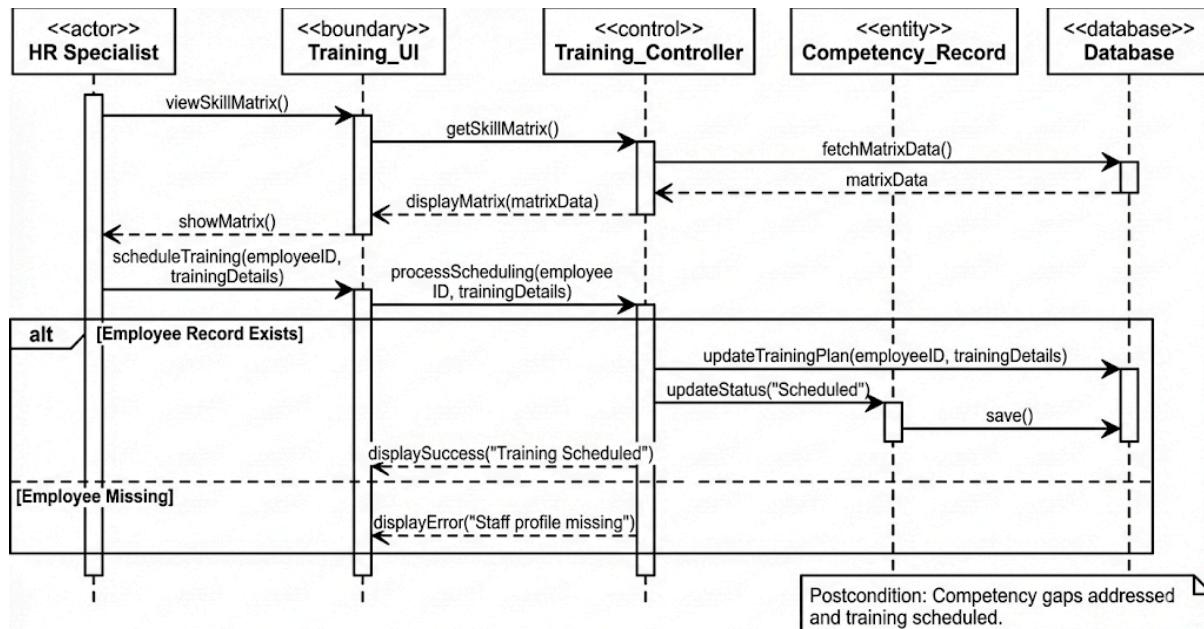


Figure 2.2.13: Sequence Diagram for Manage Competency and Training

Manage Equipment and Inventory
ID: UC07
Actors: Engineer, Admin
Preconditions: 1. Equipment registered in the system.
Flow of events: 1. Engineer updates calibration or maintenance. 2. Engineer enters the calibration date. 3. System sets reminder alerts. 4. Records are linked to CAPA if equipment fails.
Postconditions: 1. Equipment record updated.
Alternate Flows: 1. If calibration is overdue, the system will escalate the alert.

Table 2.2.8: Use Case Description for Manage Equipment and Inventory

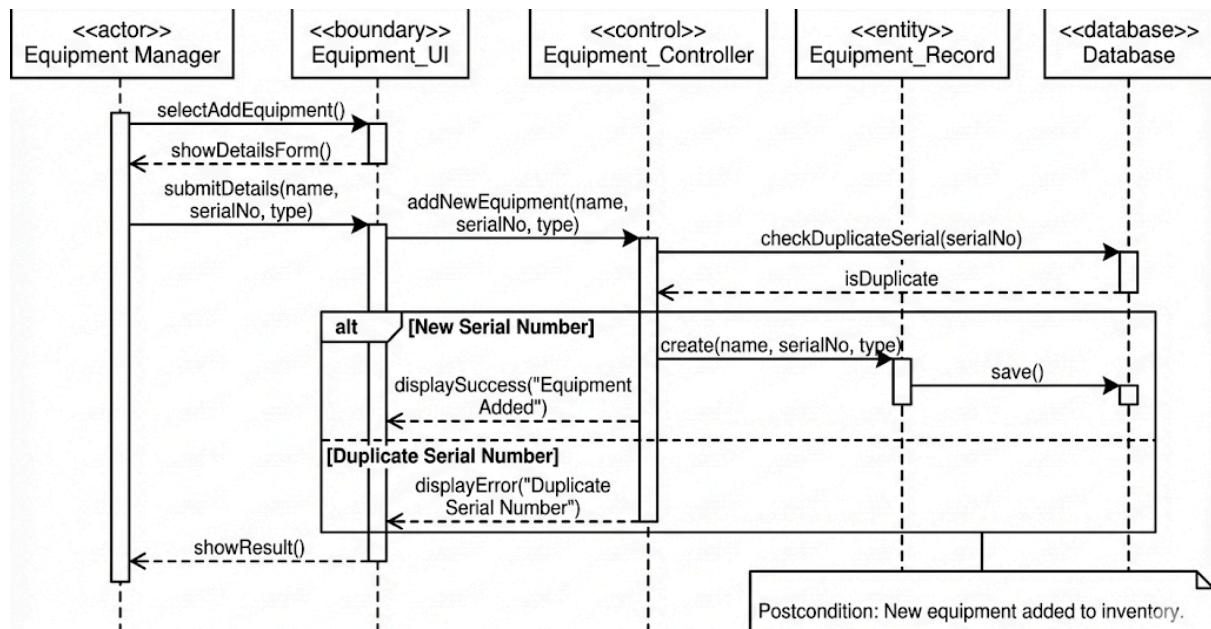


Figure 2.2.14: Sequence Diagram for Manage Equipment & Inventory

Monitor KPIs
ID: UC08
Actors: Manager, Auditor
Preconditions: 1. System has sufficient data to display KPIs.
Flow of events: 1. Manager opens the KPI dashboard. 2. Include (Initiate CAPA) 3. System aggregates CAPA closure rate, audit completion, and training compliance. 4. Manager views charts and reports.
Postconditions: 1. KPI report generated.
Alternate Flows: 1. The system will raise flags if the data is missing or incomplete.

Table 2.2.9: Use Case Description for Monitor KPIs

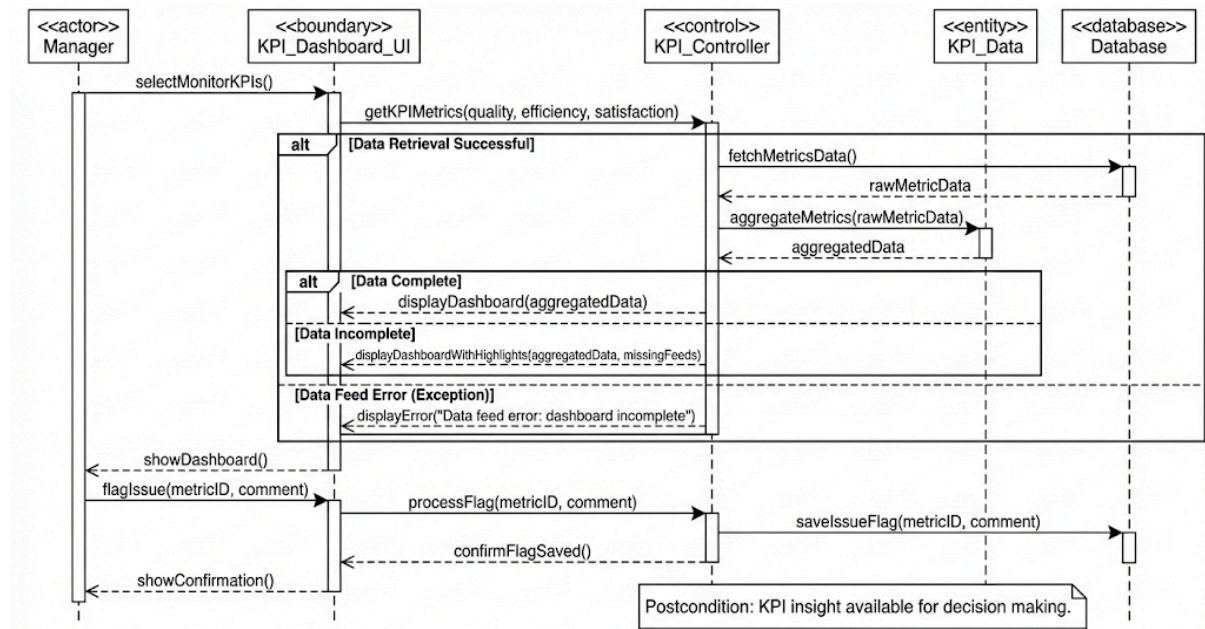


Figure 2.2.15: Sequence Diagram for Monitor KPIs

Manage Change
ID: UC09
<p>Actors: Manager, Engineer, Auditor, Admin</p>
<p>Preconditions:</p> <ol style="list-style-type: none"> 1. Change request submitted.
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. Manager reviews the change request. 2. Manager evaluates the impact. 3. Approves or rejects change. 4. System updates process/document version.
<p>Postconditions:</p> <ol style="list-style-type: none"> 1. Change log updated.
<p>Alternate Flows:</p> <ol style="list-style-type: none"> 1. The request will be archived if the change request is rejected.

Table 2.2.10: Use Case Description for Manage Change

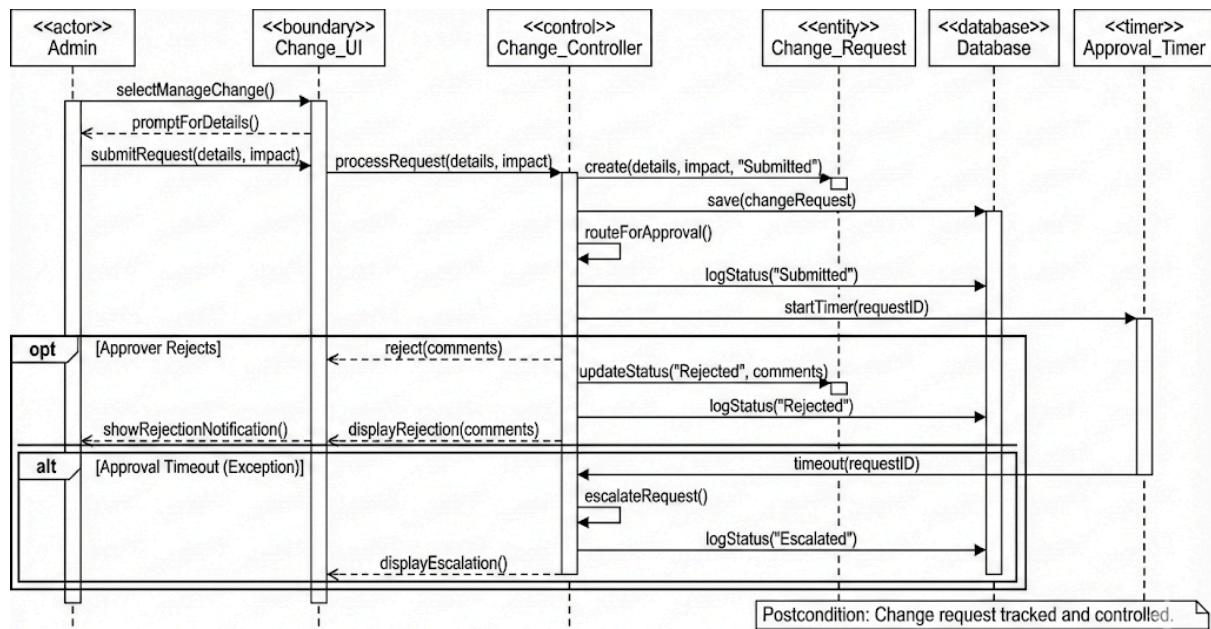


Figure 2.2.16: Sequence Diagram for Manage Change

Provide System Support
ID: UC10
Actors: Admin
Preconditions: 1. User reports the issue.
Flow of events: 1. Admin receives support request. 2. Admin reviews issue. 3. Provides a fix or escalates. 4. System logs support activity. 5. Include (Maintain System Logs).
Postconditions: 1. Support ticket closed.

Table 2.2.11: Use Case Description for Provide System Support

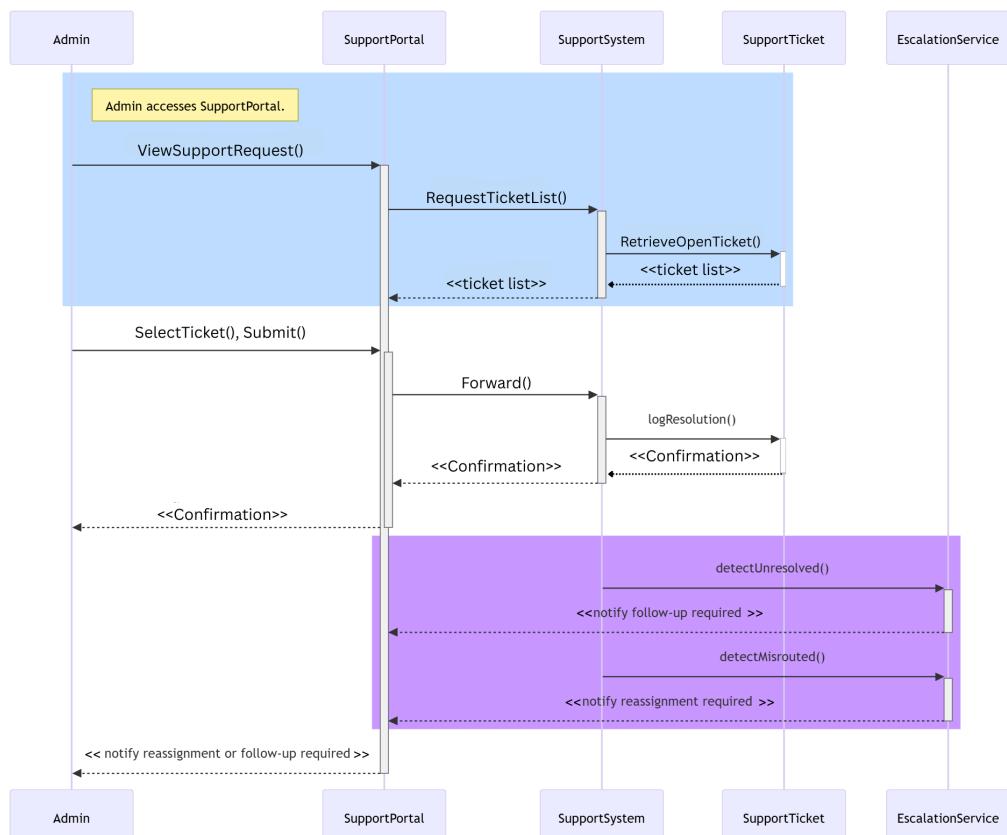


Figure 2.2.17: Sequence Diagram for Provide System Support

Maintain System Logs
ID: UC11
Actors: Admin, Auditor
Preconditions: 1. System operational.
Flow of events: 1. Any activity occurs. 2. System records logins, approvals, and CAPA actions. 3. Admin reviews logs.
Postconditions: 1. Logs stored securely.
Alternate Flows: 1. The system will restore a backup if there is a corruption in the logs.

Table 2.2.12: Use Case Description for Maintain System Logs

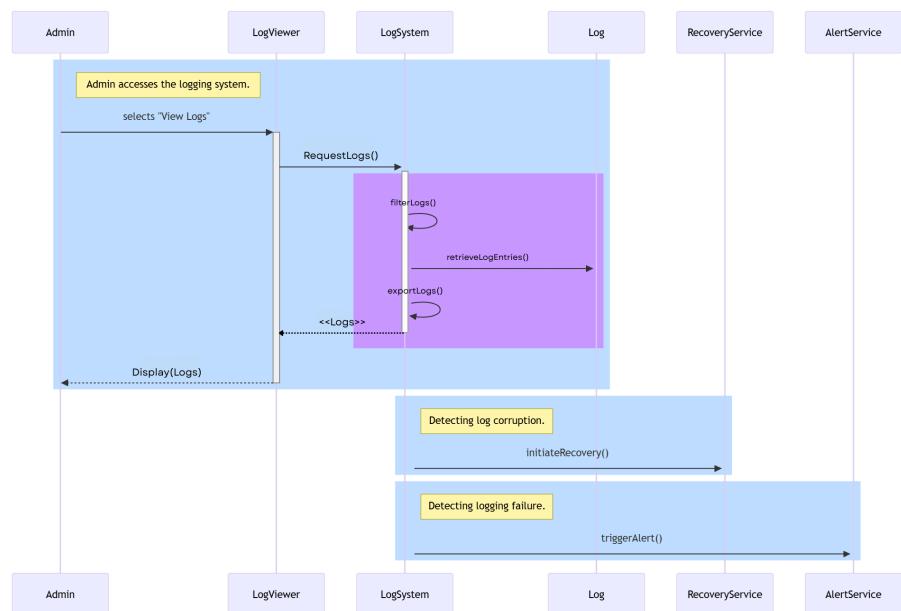


Figure 2.2.18: Sequence Diagram for Maintain System Logs

Manage Risks
ID: UC12
Actors: Manager
Preconditions: 1. Risk identified
Flow of events: 1. Manager enters the risk assessment. 2. Manager records likelihood and severity. 3. System generates a mitigation plan. 4. Manager reviews and decides whether to approve or reject.
Postconditions: 1. Risk record was updated.
Alternate Flows: 1. The rejected risks will be archived.

Table 2.2.13: Use Case Description for Manage Risks

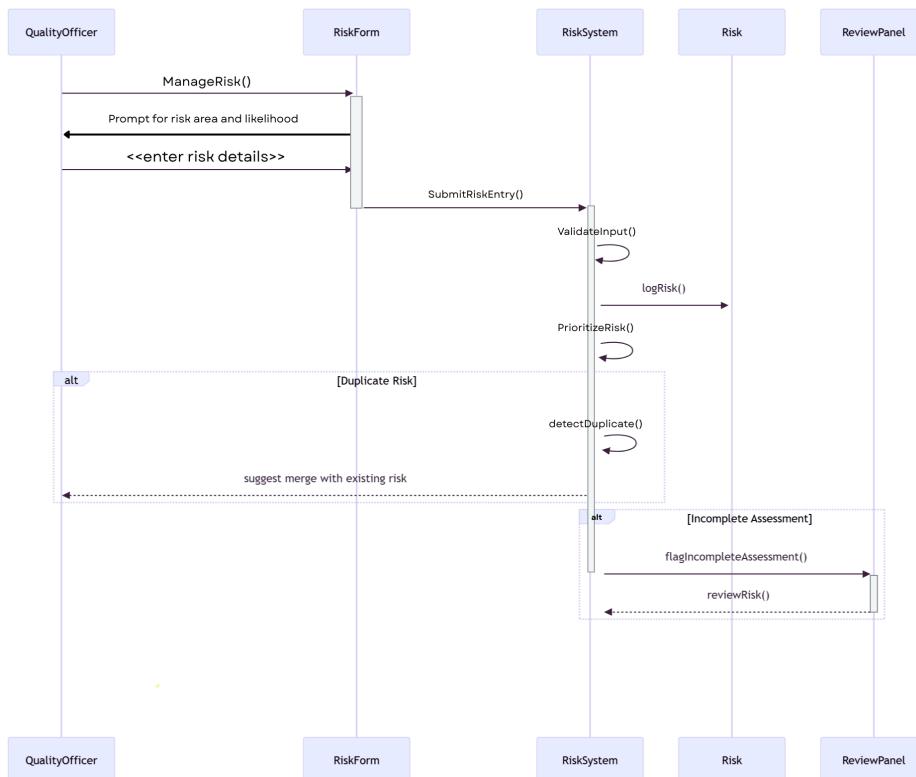


Figure 2.2.19: Sequence Diagram for Manage Risks

Manage Incident and Feedback	
ID: UC13	
Actors: Manager, Engineer, Admin	
Preconditions: 1. incident reported or feedback received.	
Flow of events: 1. Engineer logs the incident. 2. Engineer enters the description and reporter details. 3. System links the incident to CAPA if needed. 4. Include (Initiate CAPA). 5. Manager reviews the resolution.	
Postconditions: 1. The incident was closed with resolution traceability.	
Alternate Flows: 1. System will merge duplicate incidents.	

Table 2.2.13: Use Case Description for Manage Incident and Feedback

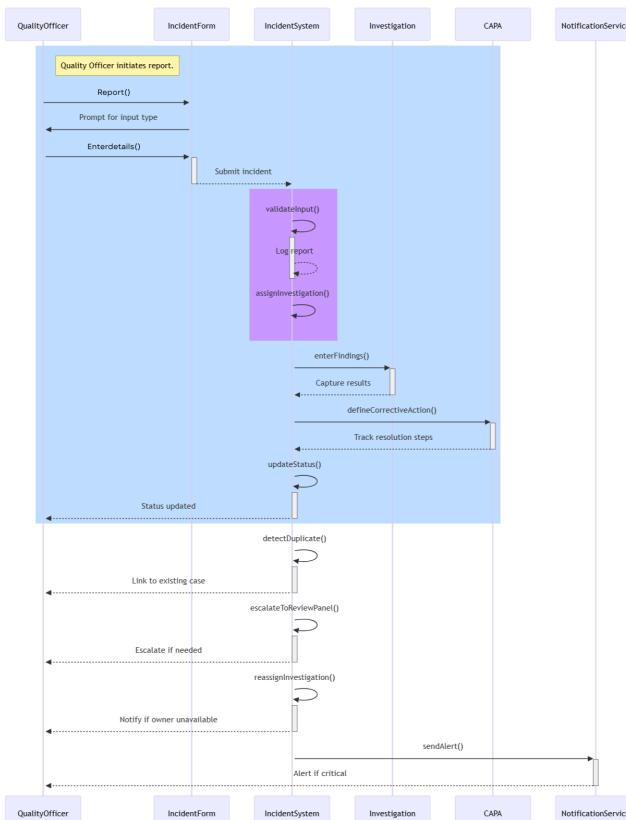


Figure 2.2.20: Sequence Diagram for Manage Incident & Feedback

2.3 Launch Phase

Sprint	User Story	Team Member Assigned
Sprint #1 UC01 (Register Document), UC02 (Approve Document) & UC09 (Manage Change) [Document Control Change Management Module]	As an engineer, I want to upload documents with version control so that SOPs are traceable. As a manager, I want to approve and reject documents so that only valid SOPs are published. As a manager, I want to manage change requests so that updates are controlled and documented.	Rasyid, Afiq Shahir
Sprint #2 UC03 (Schedule Audit), UC04 (Record Audit Findings) [Audit Management Module]	As an auditor, I want to schedule internal and external audits so that compliance checks are planned and resources are allocated. As an auditor, I want to record audit findings and non-conformances so that CAPAs can be officially initiated.	Afiq Irfan, Hazim
Sprint #3 UC06 (Manage Competency & Training) and UC07 (Manage Equipment & Inventory) [Training & Equipment Management Module]	As a manager, I want to assign training to employees so that competency gaps are addressed. As an engineer, I want to manage equipment calibration schedules so that compliance is maintained.	Rafiq, Hazim

	As an admin, I want to manage training records so that employee competencies are documented and updated.	
Sprint #4 UC10 (Provide System Support) and UC11 (Maintain System Logs) [Admin & System Support Module]	As an admin, I want to monitor CAPA progress so that corrective actions are completed and traceable.	Rasyid, Afiq Shahir
	As an admin, I want to maintain system logs so that activities are traceable and audit-ready.	
Sprint #5 UC12 (Manage Risks) & UC13 (Manage Incident & Feedback) [Risk & Incident Management Module]	As an engineer, I want to log incidents so that corrective actions are traceable.	Afiq Irfan, Afiq Shahir
	As a manager, I want to assess risks so that mitigation plans can be implemented.	
Sprint #6 UC05 (Initiate CAPA) & UC08 (Monitor KPIs) [Initiate CAPA & KPI Monitoring Module]	As an Engineer, I want to initiate CAPA linked to complaints so that corrective actions are properly tracked.	Rasyid, Rafiq
	As a manager, I want to generate KPI reports so that I can evaluate departmental performance.	

2.4 User Story Details

2.4.1 US001: User Story Register Document

User story: UC001 - Register Document	
ID:	US001
User Story Description	
As an Engineer I want to upload SOPs with version control So that documents are traceable and current	
Flow of events: <ol style="list-style-type: none">1. Engineer selects “Upload Document”2. System prompts for metadata3. Engineer uploads file4. System assigns Document ID and version5. Document stored as Draft	
Alternative flow : Invalid file format → error message Repository unavailable → error logged and notify admin	
Acceptance Criteria Precondition: Engineer logged in Postcondition: Document stored with metadata and status = Draft	
Exception flow: File exceed size limit: upload rejected, user prompted to retry	

Table 2.1: User Story Description for Register Document

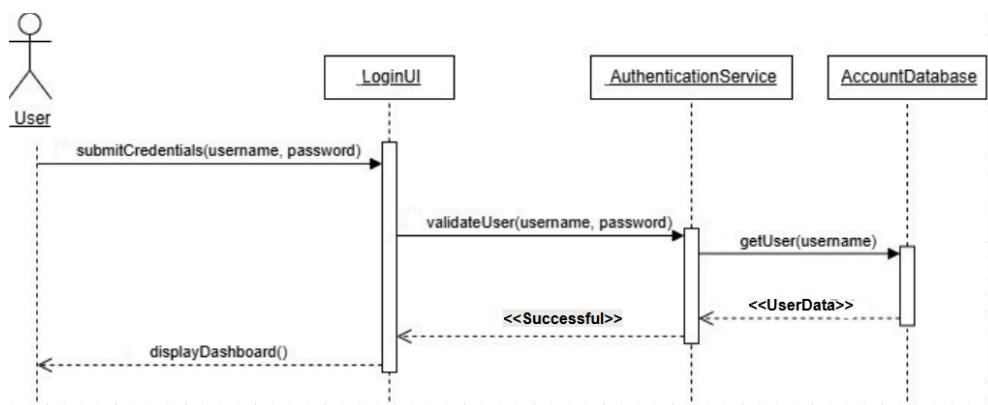


Figure 2.4.1.1: Sequence Diagram for Register Document

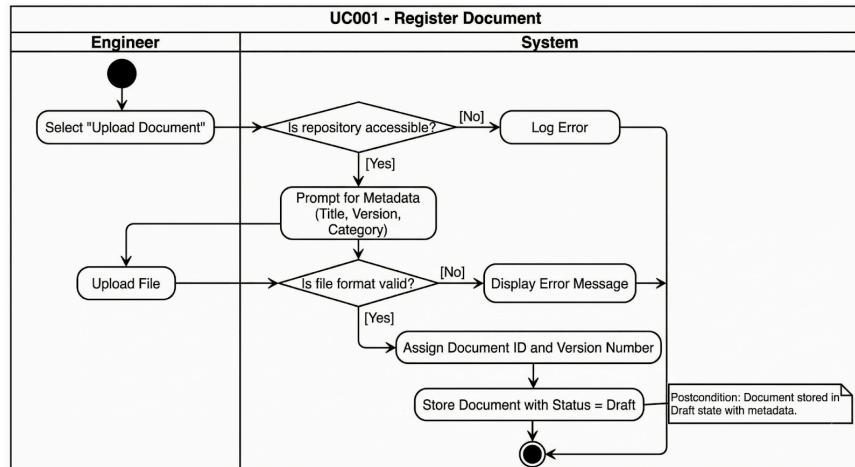


Figure 2.4.1.2: Activity Diagram for Register Document

2.1.1 US002: User Story Approve Document

User story: UC002 - Approve Document
ID: US002
<p>User Story Description</p> <p>As a Manager I want to approve or reject documents So that only validated SOPs are published.</p>
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. Manager review document 2. Manager selects approve or reject 3. System updates status
<p>Alternative flow n: Document rejected: returned to draft with reviewer comments</p>
<p>Acceptance Criteria Precondition: Document status = pendingApproval Postcondition: Status updated to Approved or Draft</p>
<p>Exception flow: Approval right missing: system denies access Document locked by another reviewer: approval deferred</p>

Table 2.4.2.1: User Story Description for Approve Document

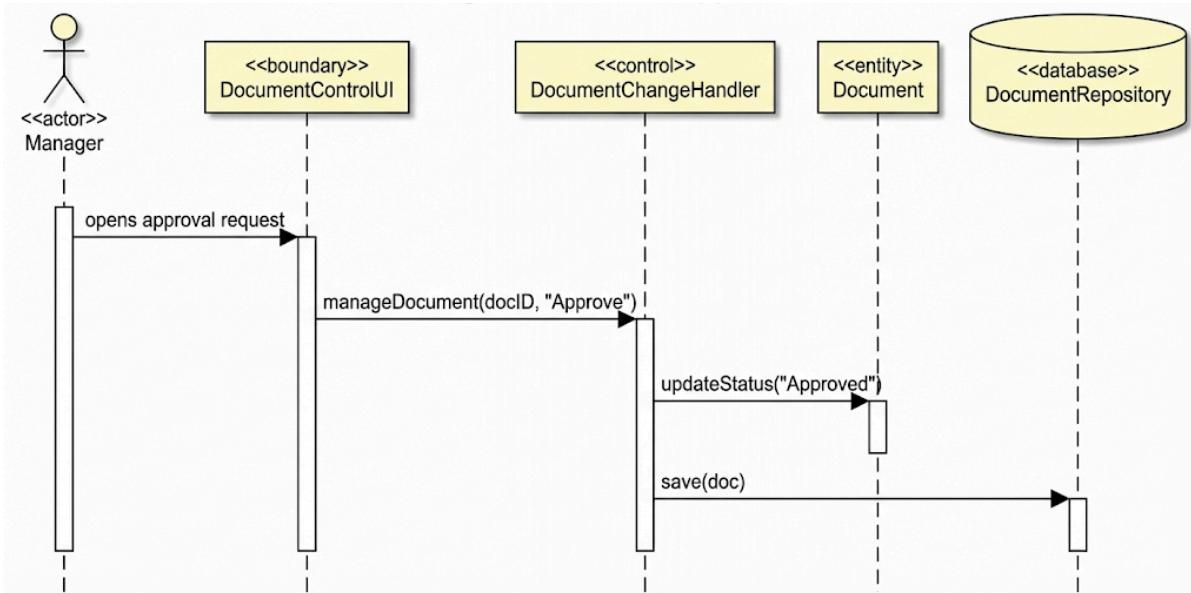


Figure 2.4.2.1: Sequence Diagram for Approve Document

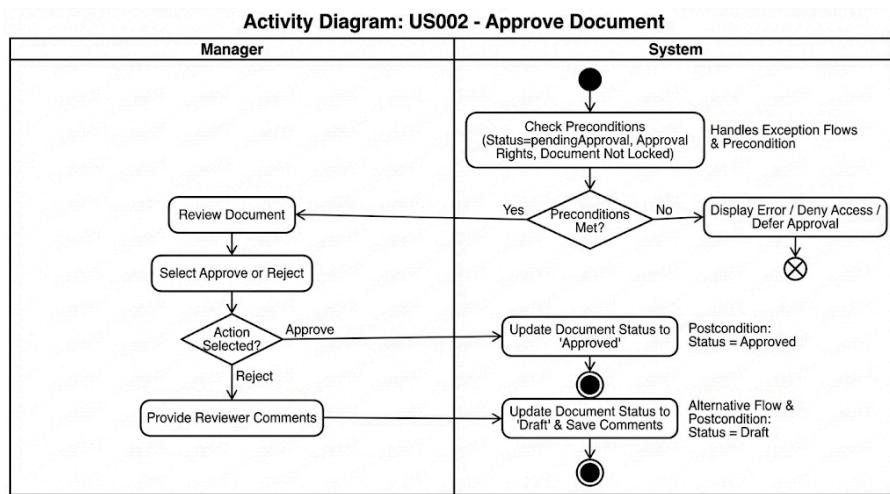


Figure 2.4.2.2: Activity Diagram for Approve Document

2.1.2 US003: User Story Schedule Audit

User story: UC003 - Schedule Audit	
ID: US003	
User Story Description	
<p>As an Auditor I want to schedule internal and external audits So that compliance checks are planned and resources are allocated</p>	
Flow of events: <ol style="list-style-type: none"> 1. Auditor selects "Schedule Audit" 2. System prompts for date, scope, auditor 3. Auditor enter details 4. System creates audit record 	
Alternative flow n:	
Acceptance Criteria Precondition: Auditor logged in Postcondition: Audit scheduled and visible in calendar with assigned auditor notified	
Exception flow: Calendar service down: scheduling fails, error logged Duplicate audit entry: system prompts for confirmation	

Table 2.4.3.1: User Story Description for Schedule Audit

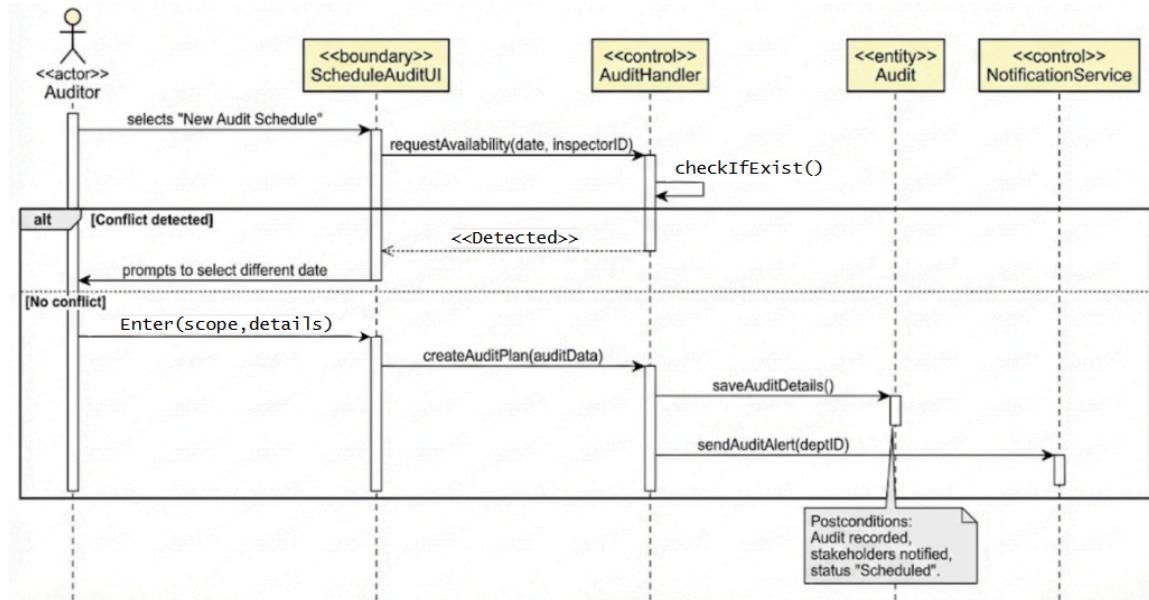


Figure 2.4.3.1: Sequence Diagram for Schedule Audit

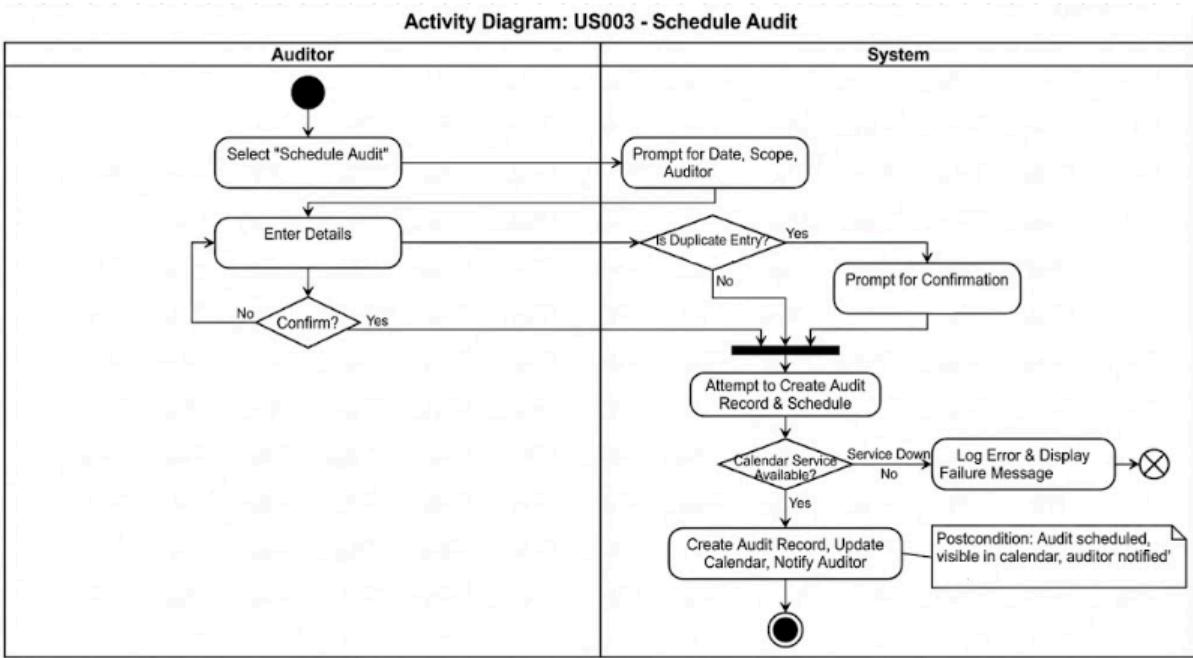


Figure 2.4.3.2: Activity Diagram for Schedule Audit

2.1.3 US004: User Story Record Audit Findings

User story: UC004 - Record Audit Findings	
ID: US004	
User Story Description	
<p>As an Auditor I want to record audit findings and non-conformances So that observation and nonconformities are documented for compliance </p>	
Flow of events: <ol style="list-style-type: none"> 1. Auditor records audit findings 2. System prompts for observations, evidence, and nonconformities 3. Auditor enters findings and attaches supporting files 4. System stores findings linked to audit ID 	
Alternative flow n: Incomplete entry → system requests missing details.	
Acceptance Criteria Precondition: Auditor logged in; audit session completed Postcondition: Audit findings recorded and accessible for CAPA or reporting	
Exception flow: File upload fails: retry required	

Table 2.4.4.1: User Story Description for Record Audit Findings

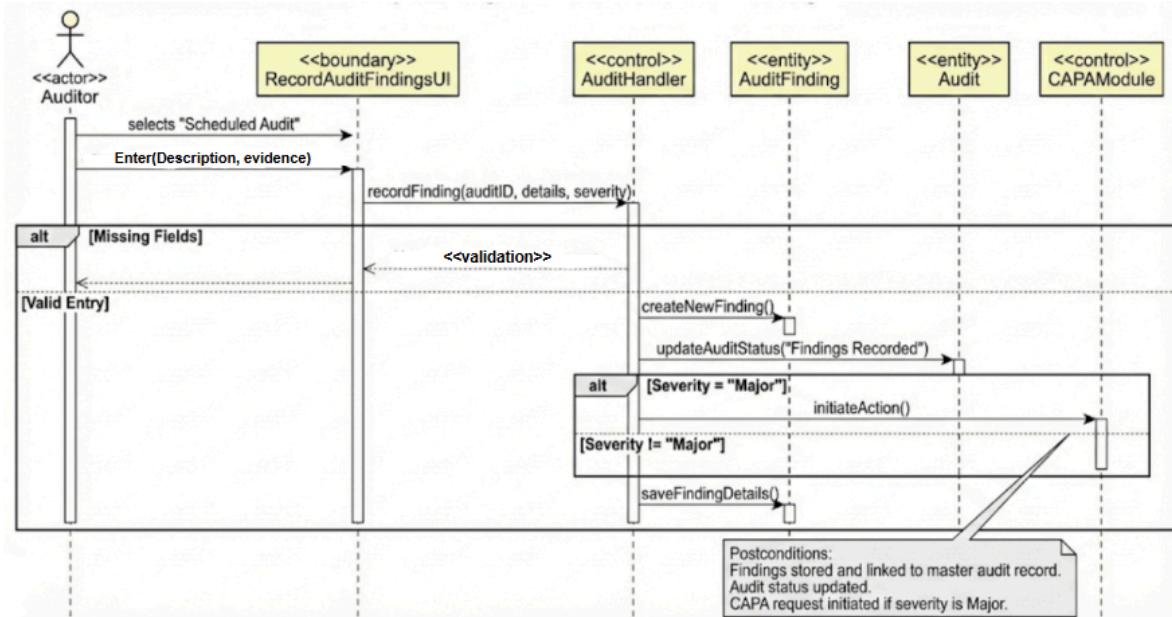


Figure 2.4.4.1: Sequence Diagram for Record Audit Findings

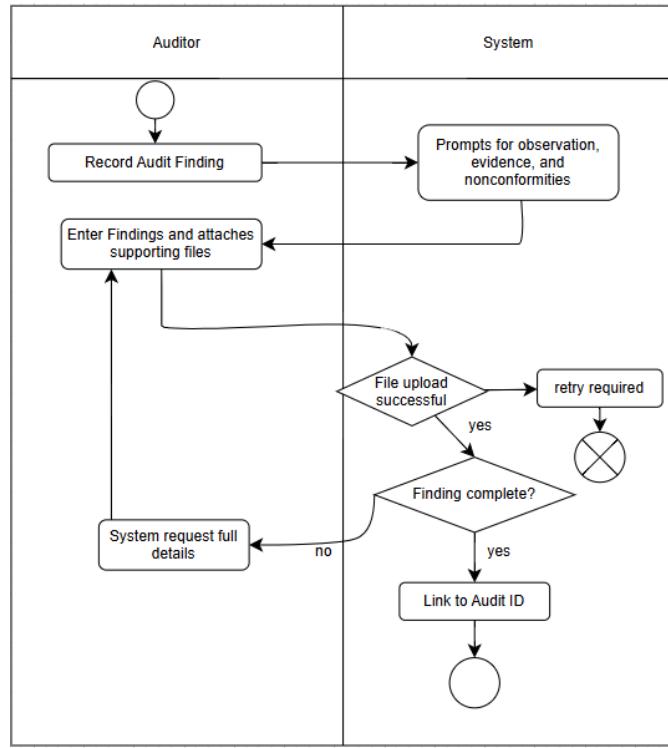


Figure 2.4.4.2: Activity Diagram for Record Audit Findings

2.4.5 US005: User Story Initiate CAPA

User story: UC005 - Initiate CAPA	
ID: US005	
User Story Description	
<p>As an Engineer I want to initiate CAPA linked to complaints So that corrective actions are properly tracked </p>	
Flow of events: <ol style="list-style-type: none"> 1. Engineer selects “Initiate CAPA” 2. System prompts for root cause and action 3. Engineer enters details 4. System assigns CAPA ID 	
Alternative flow n: Duplicate CAPA detected: system warn and request conformation	
Acceptance Criteria Precondition: Complaint logged Postcondition: CAPA record created and linked to complaint	
Exception flow: Missing complaint ID → error message	

Table 2.4.5.1: User Story Description for Initiate CAPA

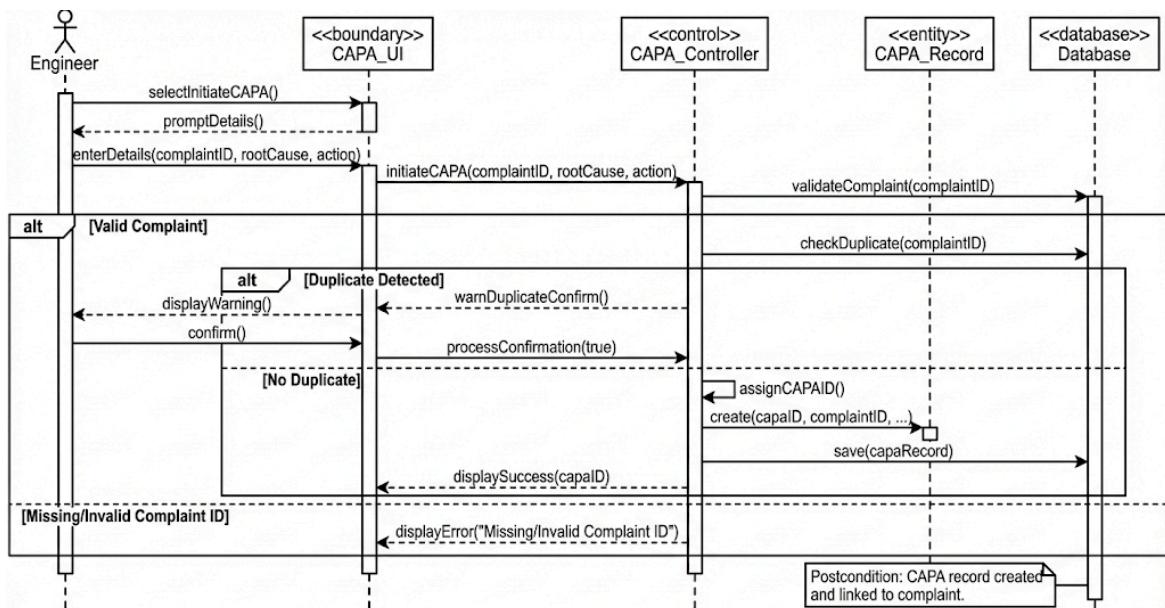


Figure 2.4.5.1: Sequence Diagram for Initiate CAPA

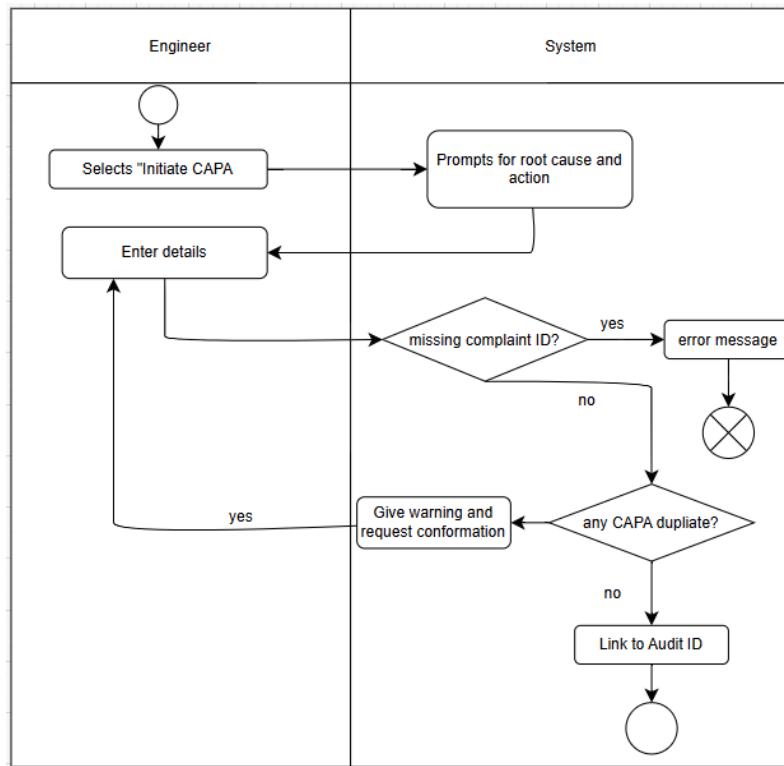


Figure 2.4.5.2: Activity Diagram for Initiate CAPA

2.4.6 US006: User Story Manage Competency and Training

User story: UC006 - Manage Competency and Training	
ID: US006	
User Story Description	
<p>As a HR specialist I want i want to manage competency and training So that competency gaps are addressed</p>	
Flow of events:	
<ol style="list-style-type: none"> 1. Manager assigns training 2. System displays staff skill matrix 3. Specialist assesses skills and identifies gaps 4. Specialist schedules training 5. System updates training plan and competency records 	
Alternative flow n:	
Acceptance Criteria	
<p>Precondition: Employee record exists Postcondition: Competency gaps addressed and training scheduled</p>	
Exception flow:	
Staff profile missing: error message	

Table 2.4.6.1: User Story Description for Manage Competency and Training

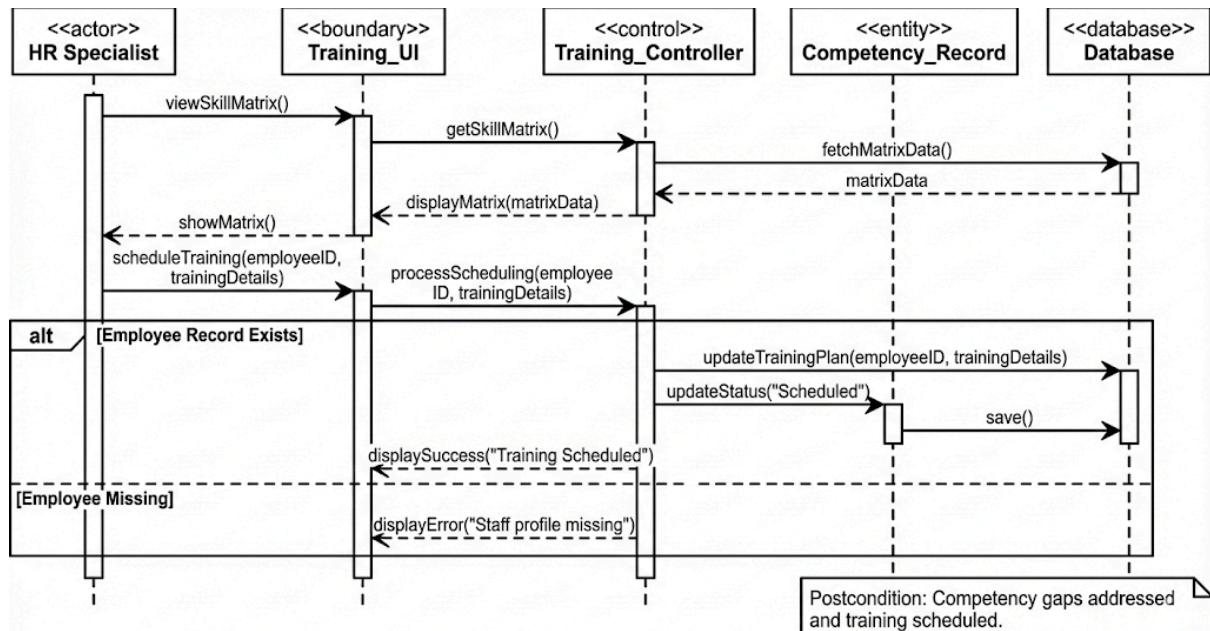


Figure 2.4.6.1: Sequence Diagram for Manage Competency and Training

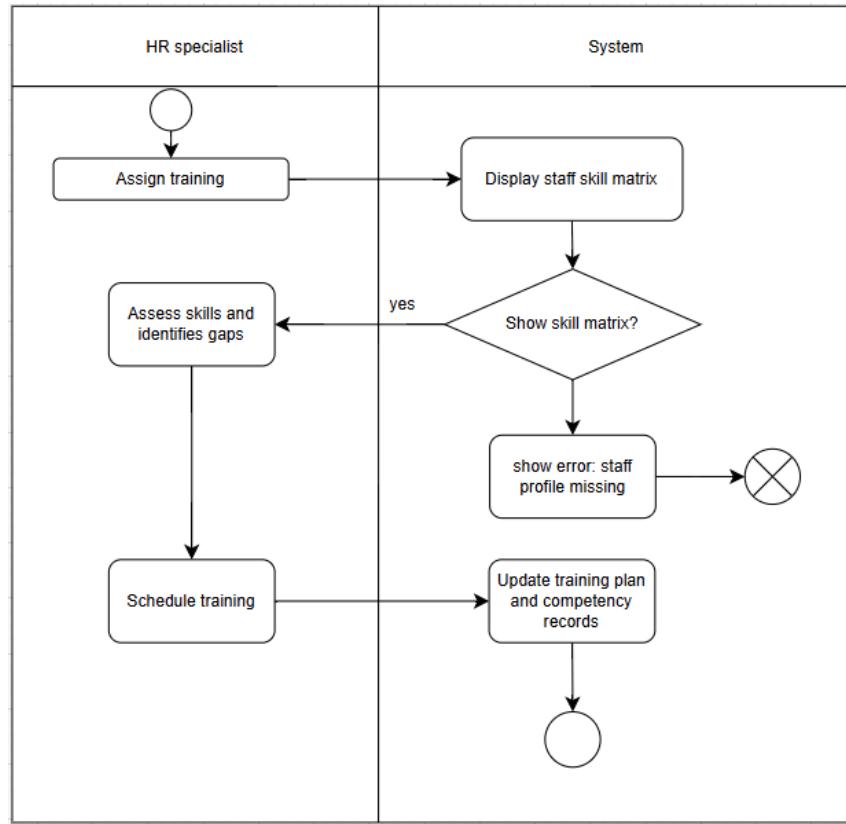


Figure 2.4.6.2: Activity Diagram for Manage Competency and Training

2.4.7 US007: User Story Manage Equipment & Inventory

User story: UC007 - Manage Equipment & Inventory	
ID: US007	
User Story Description	
<p>As an Engineer I want manage equipment and inventory So that calibration, maintenance, and readiness are tracked</p>	
Flow of events: <ol style="list-style-type: none"> 1. Engineer selects “Manage Equipment” 2. System displays equipment list and inventory. 3. Engineer updates calibration, maintenance, or readiness status 4. System logs updates and sets readiness flag 	
Alternative flow n: Equipment not found: system prompts for registration	
Acceptance Criteria Precondition: Engineer logged in; equipment registered Postcondition: Equipment and inventory status updated	
Exception flow: Calibration overdue: alert triggered	

Table 2.4.7.1: User Story Description for Manage Equipment & Inventory

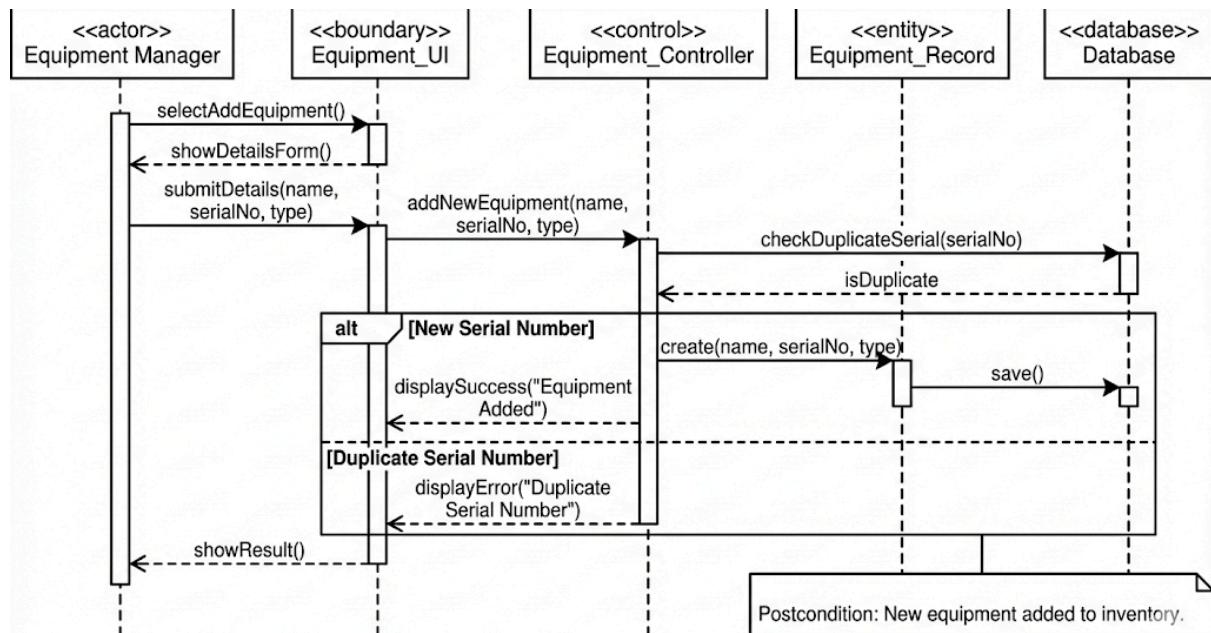


Figure 2.4.7.1: Sequence Diagram for Manage Equipment & Inventory

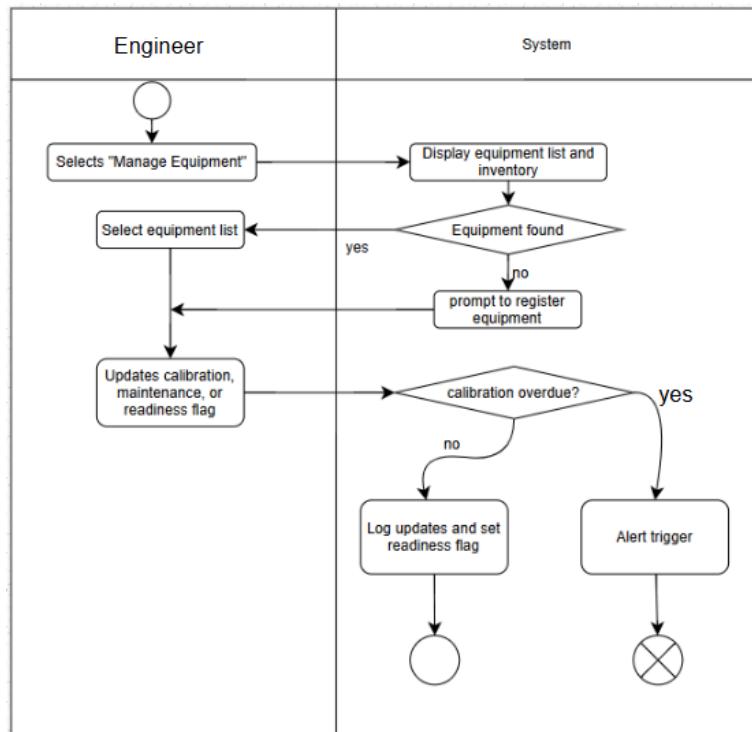


Figure 2.4.7.2: Activity Diagram for Manage Equipment & Inventory

2.4.8 US008: User Story Monitor KPIs

User story: UC008 - Monitor KPIs	
ID: US009	
User Story Description	
<p>As a Manager I want to monitor KPIs So that I can evaluate departmental performance.</p>	
Flow of events: <ol style="list-style-type: none"> Manager selects “Monitor KPIs” System displays metrics(quality, efficiency, satisfaction) Manager reviews trends and flags issues 	
Alternative flow n: KPI data incomplete: system highlights missing feeds	
Acceptance Criteria Precondition: Manager logged in;KPI dashboard accessible Postcondition: KPI insight available for decision making	
Exception flow: Data feed error: dashboard incomplete	

Table 2.4.8.1: User Story Description for Monitor KPIs

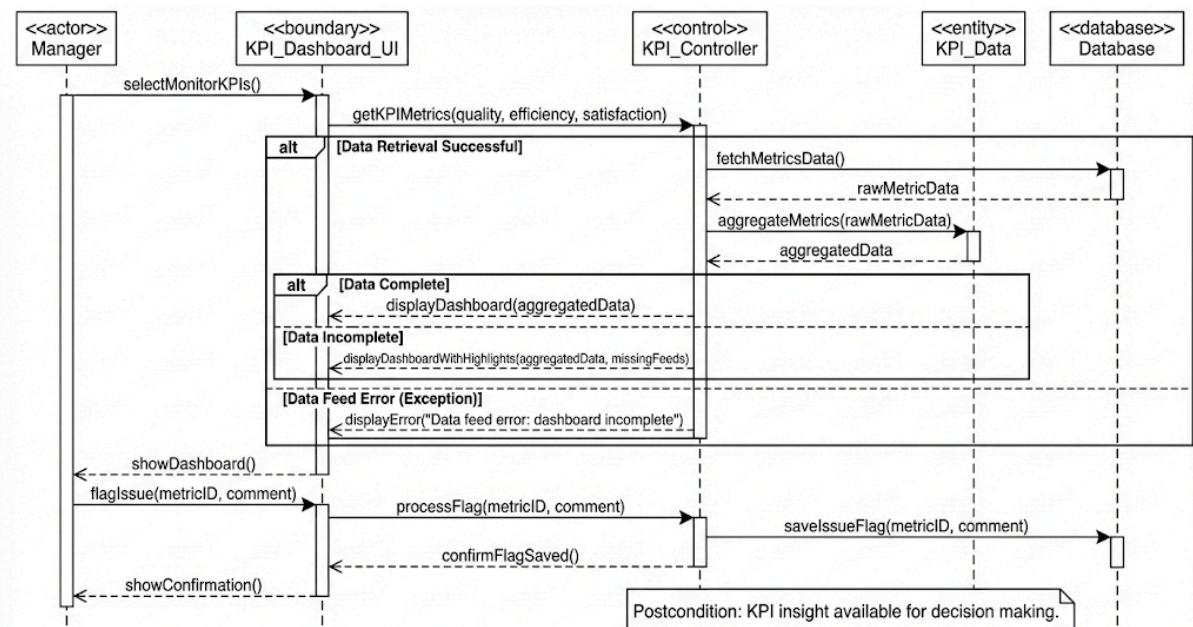


Figure 2.4.8.1: Sequence Diagram for Monitor KPIs

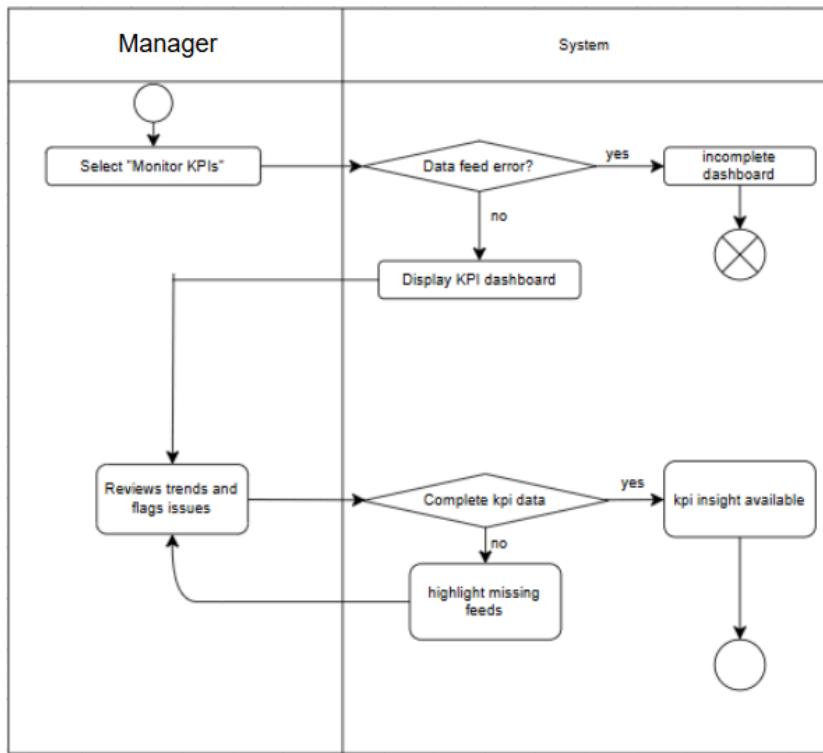


Figure 2.4.8.2: Activity Diagram for Monitor KPIs

2.4.9 US009: User Story Manage Change

User story: UC009 - Manage Change	
ID: US009	
User Story Description	
<p>As an Admin I want to manage change request So that updates are controlled and documented</p>	
Flow of events: <ol style="list-style-type: none"> 1. Admin selects “manage change.” 2. System prompts for change details and impact 3. Admin submits request 4. System routes for approval and logs status 	
Alternative flow n: Change rejected: returned with comments	
Acceptance Criteria Precondition: Support request received from a client or staff member Postcondition: Change request tracked and controlled	
Exception flow: Approval delayed: escalation triggered	

Table 2.4.9.1: User Story Description for Manage Change

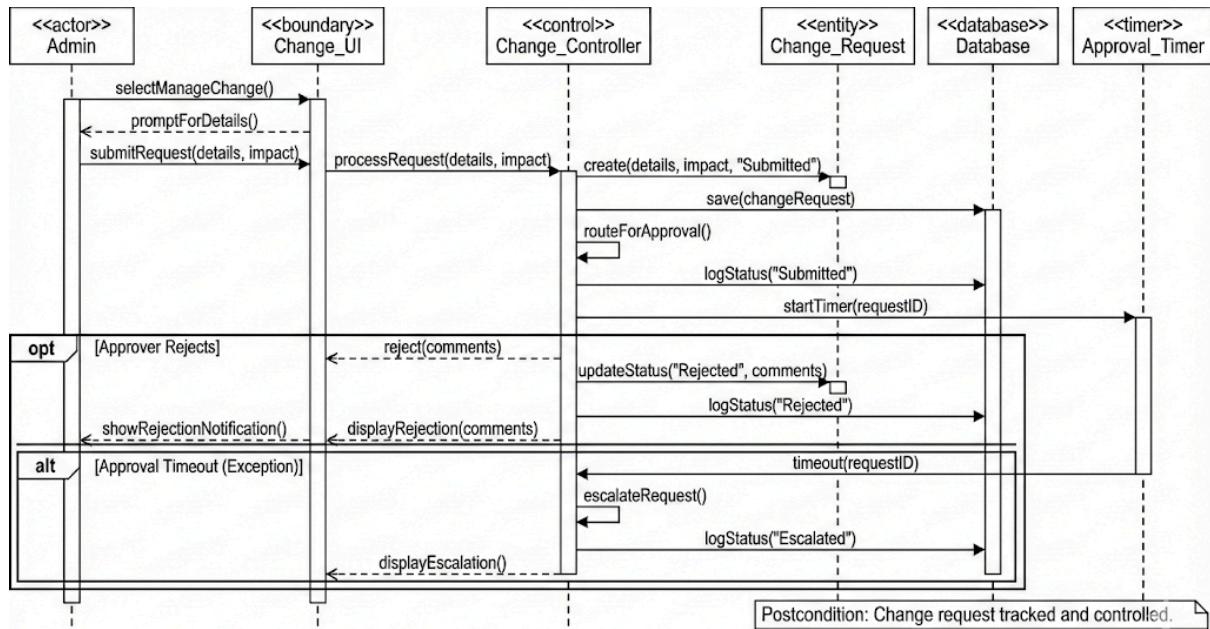


Figure 2.4.9.1: Sequence Diagram for Manage Change

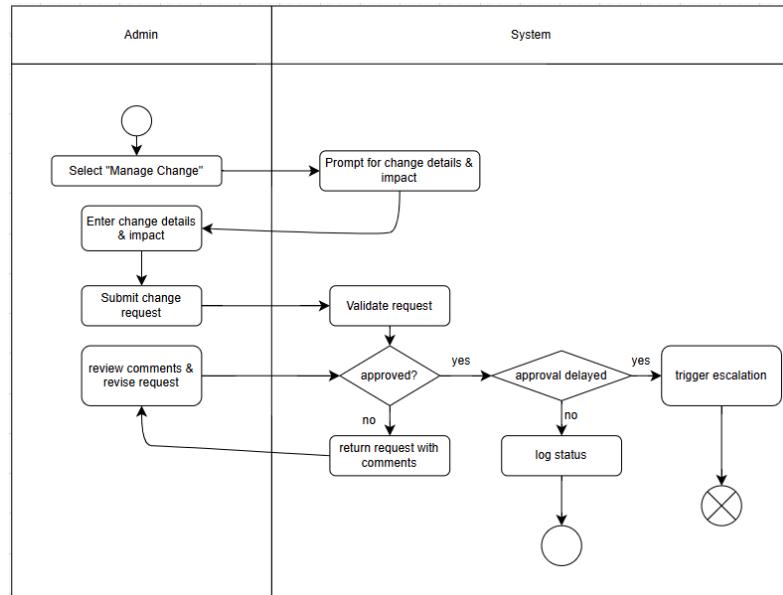


Figure 2.4.9.2: Activity Diagram for Manage Change

2.4.10 US010: User Story Provide System Support

User story: UC010 - Provide System Support
ID: US010
User Story Description
<p>As an Admin I want to provide system support So that client and staff issues are resolved quickly</p>
Flow of events: <ol style="list-style-type: none">1. Navigate to Support Dashboard.2. Click on Ticket ID.3. Enter text in "Resolution" field.4. Click "Update Status to resolved" button.5. Check "Resolved" tab in dashboard.
Alternative flow 1: <ol style="list-style-type: none">1. Issue unresolved: flagged for follow up
Acceptance Criteria Precondition: Admin logged in; support portal active Postcondition: Support ticket resolved or escalated
Exception flow: Ticket misrouted: reassignment required

Table 2.4.10.1: User Story Description for Provide System Support

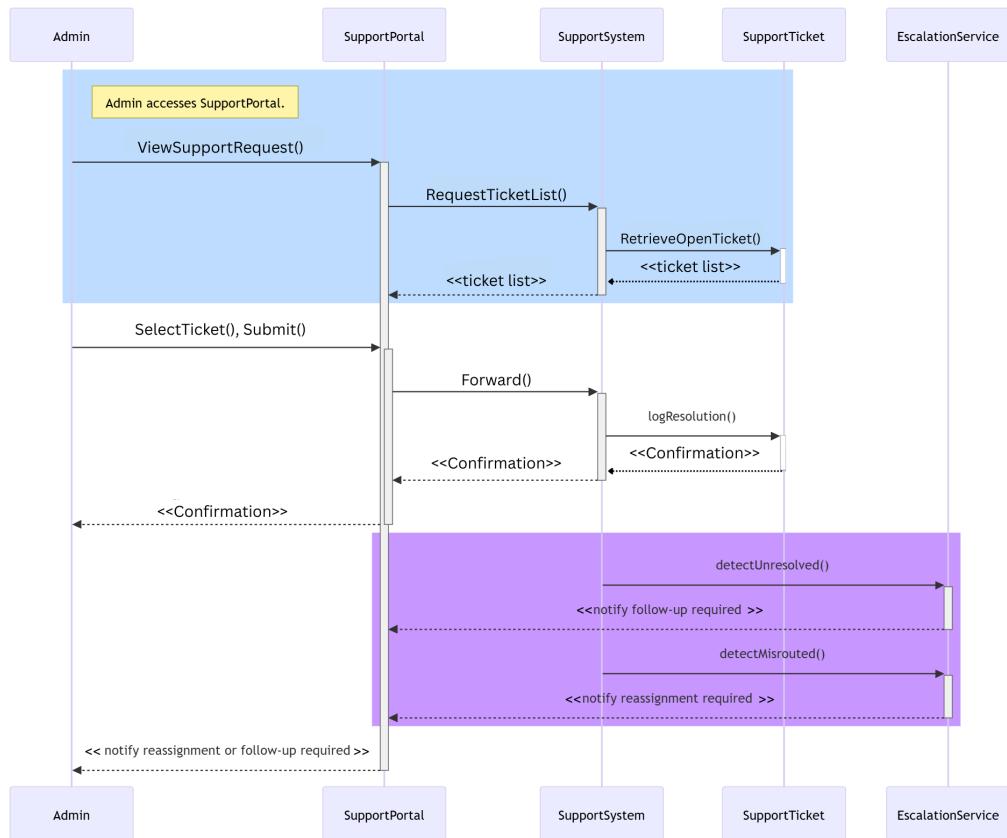


Figure 2.4.10.1: Sequence Diagram for Provide System Support

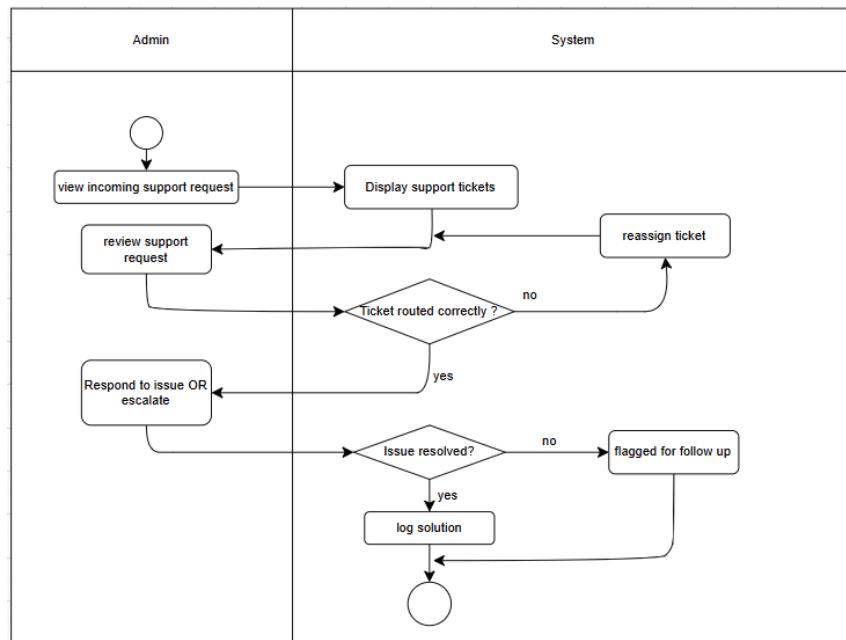


Figure 2.4.102: Activity Diagram for Provide System Support

2.4.11 US011: User Story Maintain System logs

User story: UC011 - Maintain System Logs
ID: US011
User Story Description
<p>As an Admin I want to maintain system logs So that activities are traceable for accountability</p>
Flow of events: <ol style="list-style-type: none">1. System automatically records activities (logins, approvals, changes)2. Admin selects “view logs”3. Admin filters and exports logs
Alternative flow 1: Log corruption detected: system initiates recovery
Acceptance Criteria Precondition: Admin logged in; logging service active Postcondition: Logs available for audit and traceability
Exception flow: Logging service down: alert triggered

Table 2.4.11.1: User Story Description for Maintain System Logs

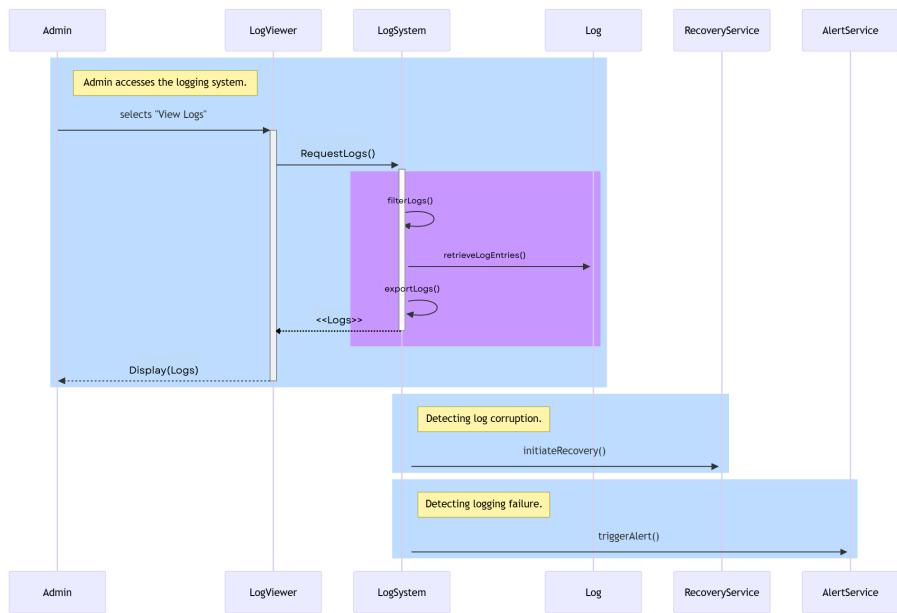


Figure 2.4.11.1: Sequence Diagram for Maintain System Logs

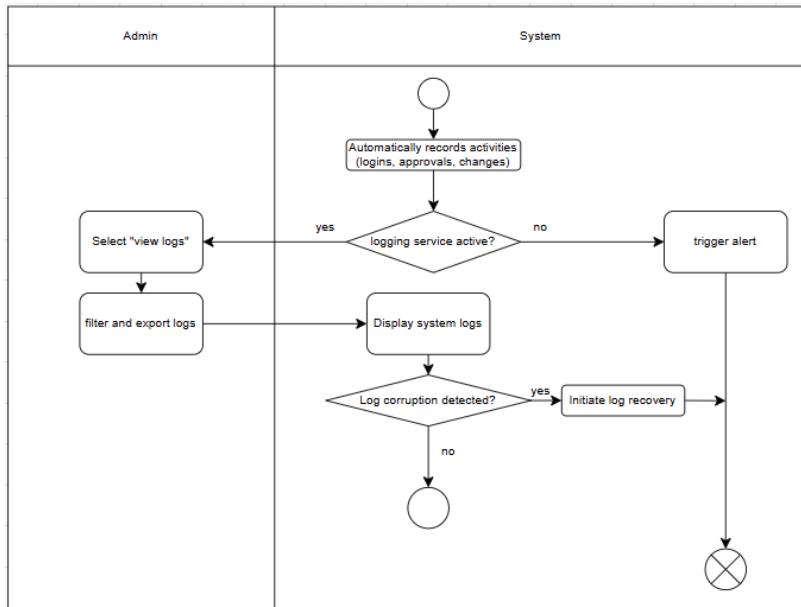


Figure 2.4.11.2: Activity Diagram for Maintain System Logs

2.4.12 US012: User Story Manage Risks

User story: UC012 - Manage Risks
ID: US012
User Story Description
<p>As a Manager I want to manage risks So that potential issues are identified, assessed, and mitigated</p>
Flow of events: <ol style="list-style-type: none">1. Manager selects “manage Risks.”2. System prompts for risks area and likelihood3. Manager submits risk entry4. System logs and prioritize risk
Alternative flow n: Duplicate risk entry: system suggests merge
Acceptance Criteria Precondition: Quality Officer logged in; process data accessible Postcondition: Risk identified, assessed, and tracked.
Exception flow: Risk assessment incomplete: flagged for review

Table 2.4.12.1: User Story Description for Manage Risks

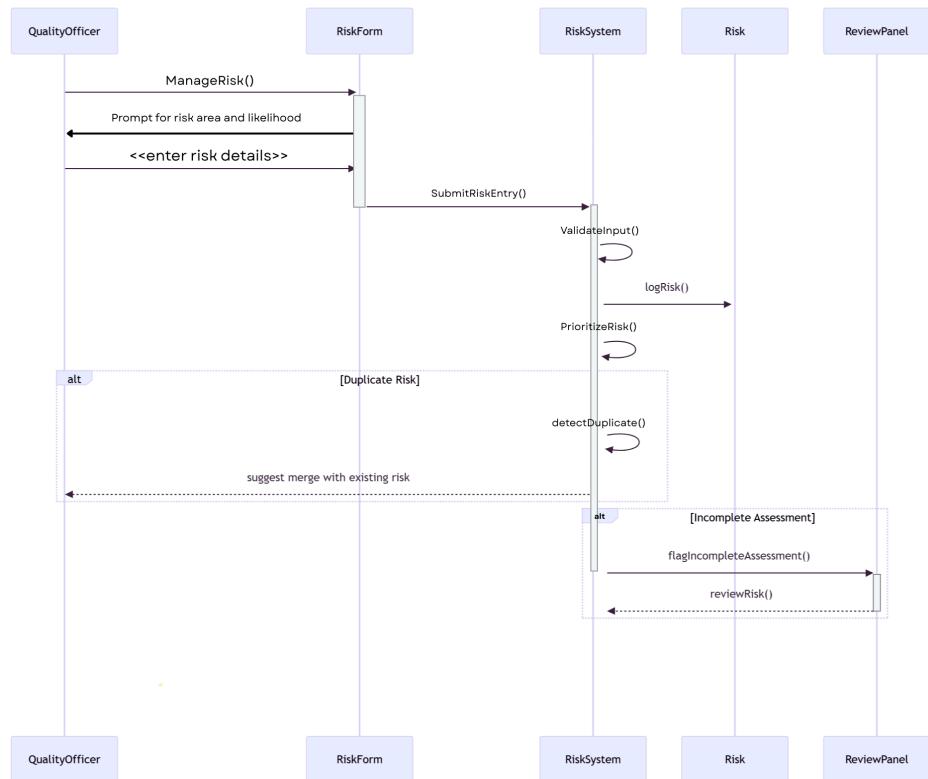


Figure 2.4.12.1: Sequence Diagram for Manage Risks

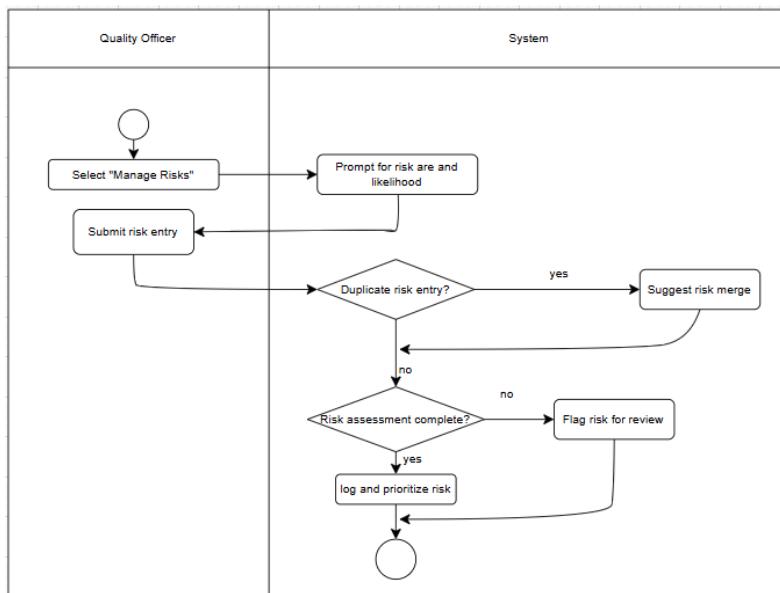


Figure 2.4.12.2: Activity Diagram for Manage Risks

2.4.13 US013: User Story Manage Incident & Feedback

User story: UC013 - Manage Incident & Feedback
ID: US013
<p>User Story Description</p> <p>As a Manager I want to manage Incidents and feedback So that complaints and nonconformities are resolved</p>
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. Manager selects “Report incident/Feedback” 2. System prompts for input type(incident, complaint, suggestion) 3. User enters details including date, location, description, and severity 4. System validates input and logs the report 5. System assigns investigation owner and sets status = Open 6. Assigned manager conducts investigation and enters findings 7. Corrective actions are defined and tracked 8. System updates status to resolved or escalated
<p>Alternative flow n: Duplicate report detected: system links to existing case</p>
<p>Acceptance Criteria Precondition: Quality Officer logged in; incident or feedback reported Postcondition: Incident/Feedback recorded, investigated, and resolution status updated</p>
<p>Exception flow: Root cause unclear: system escalates to review panel Investigation owner unavailable: reassignment triggered Critical Incident flagged: immediate notification sent to management</p>

Table 2.4.13.1: User Story Description for Manage Incident & Feedback

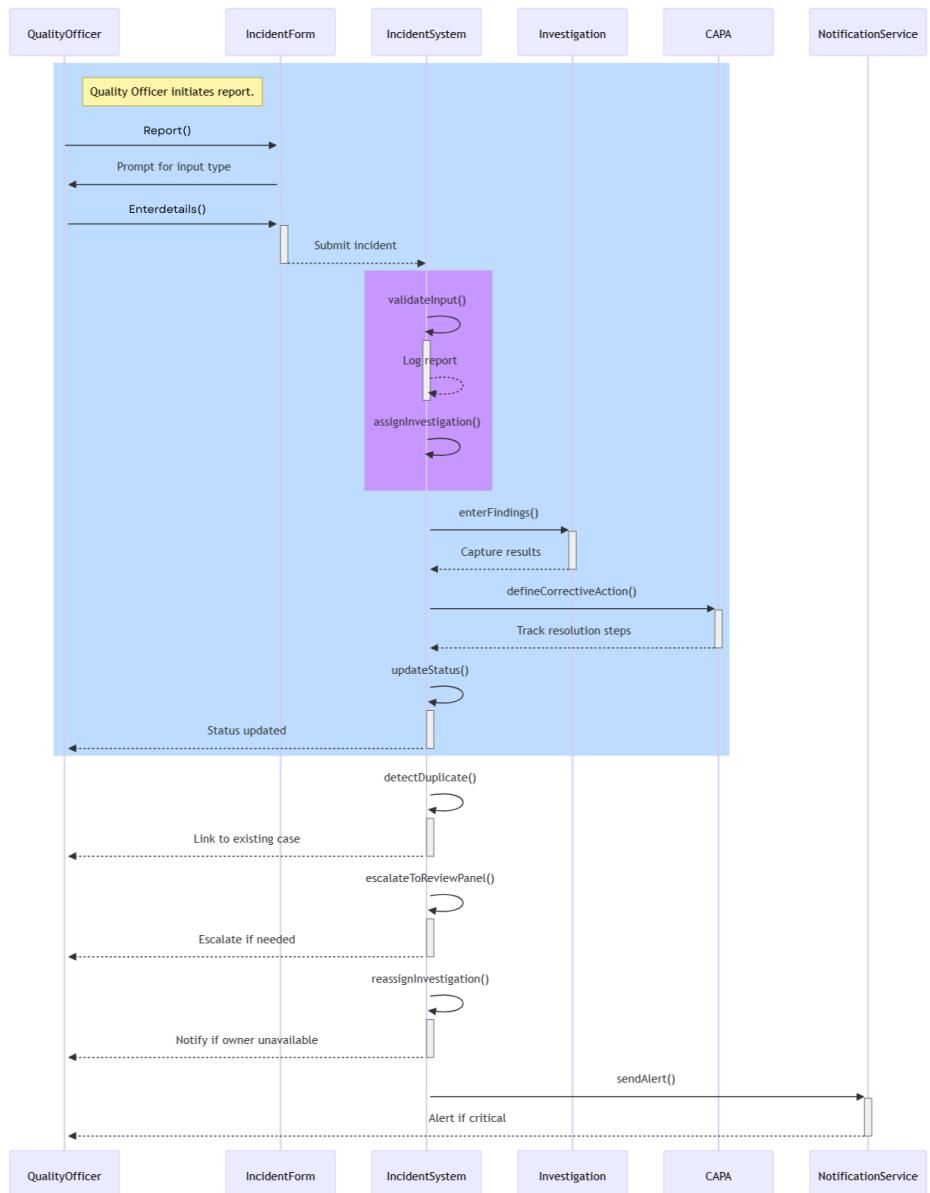


Figure 2.4.13.1: Sequence Diagram for Manage Incident & Feedback

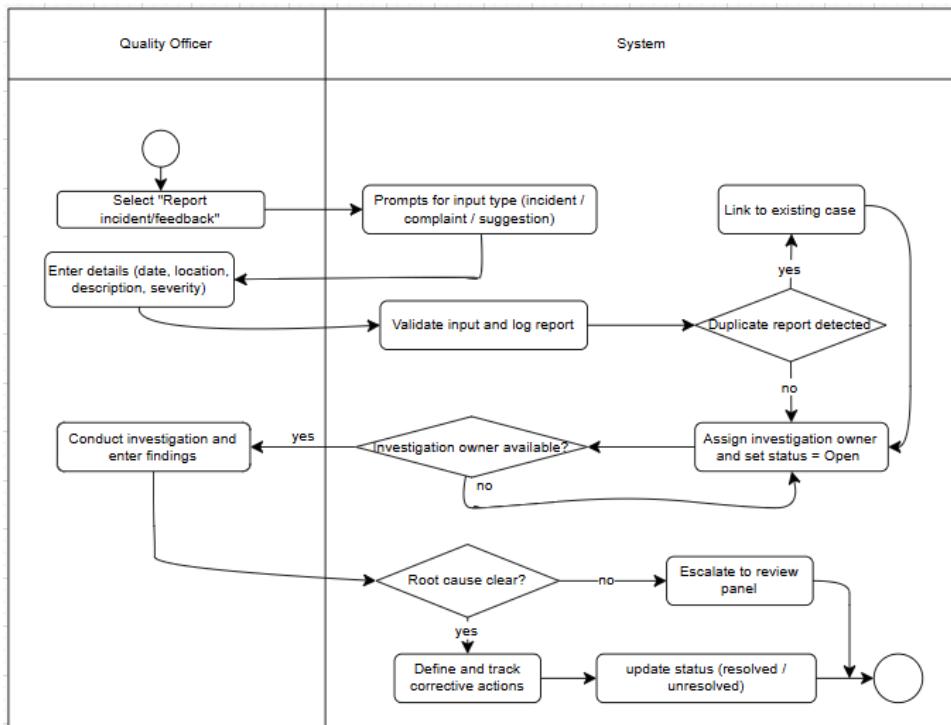


Figure 2.4.13.2: Activity Diagram for Manage Incident & Feedback

2.5 Performance and Other Requirements

This section details the non-functional requirements (NFRs) necessary to ensure QMS-Flow operates effectively within Avialite's environment.

2.5.1 Performance Requirements

- Response Time:
 - The system must load dashboard pages within 2 seconds and complete document search queries within 3 seconds over the internal network.
- Throughput:
 - The system must support simultaneous access by at least 50 concurrent users (management, engineers, and auditors) without performance degradation.
- Capacity:
 - The database must support the storage of up to 10,000 document versions and 5 years of historical audit data without archival.
- Availability:
 - The system must maintain 99.9% availability during business hours (8:00 AM – 6:00 PM) to ensure uninterrupted access to SOPs and audit logs.

2.5.2 Security Requirements

- Access Control:
 - The system must implement strictly Role-Based Access Control (RBAC). Only users with the "Manager" role can approve documents, and only "Auditors" can close audit findings.
- Data Integrity:
 - The system must maintain an immutable Audit Trail for all document changes, capturing the User ID, Timestamp, and Action (Create, Edit, Approve, Delete).

- Encryption:
 - All sensitive data (employee competency records and proprietary technical drawings) must be encrypted at rest (AES-256) and in transit (TLS 1.2+).

2.5.3 Reliability and Recoverability

- Data Backups:
 - The system must perform automated incremental backups daily and full backups weekly.
- Recovery Time:
 - In the event of a system failure, the system must be recoverable within 4 hours (RTO) with a maximum data loss of 1 hour (RPO).

2.6 Design Constraints

This section outlines the limitations and standards that strictly govern the design and development of QMS-Flow.

2.6.1 Regulatory and Compliance Constraints

- ISO Compliance:
 - The system workflow must strictly adhere to ISO 9001:2015 clauses regarding "Documented Information" (Clause 7.5) and "Performance Evaluation" (Clause 9).
- Auditability:
 - The system architecture must prevent the deletion of any finalized audit records or approved CAPA reports to satisfy external regulatory audits.

2.6.2 Hardware and Software Constraints

- Browser Compatibility:
 - The web interface must be fully functional on the latest stable versions of Google Chrome and Microsoft Edge, as these are the standard browsers used at Avialite Sdn. Bhd.
- Legacy Integration:
 - The system must be able to export reports in .CSV and .PDF formats to ensure compatibility with legacy reporting tools used by upper management.
- Hosting:
 - The system must be deployable on the existing Avialite on-premise Windows Server environment (or specified Cloud Provider) with limited internet bandwidth for external access.

2.6.3 User Interface Constraints

- Mobile Accessibility:

- The "Approve Document" and "Record Audit Findings" modules must be responsive and usable on tablet devices (iPad/Android) to facilitate on-site factory audits.
- Language:
 - The user interface must be provided in English, as it is the operational language of Avialite Sdn. Bhd

3 System Architectural Design

3.1 Architecture Style and Rationale

The architectural style chosen for the development of the QMS-Flow system is the **Layered Architecture**. This architectural style arranges the system into a set of logical layers where each layer is responsible for a specific group of functionalities and provides services to the layer above it. The lowest layer represents the core services of the system, such as data storage and access, while higher layers focus on application logic and user interaction. This structure allows the system to be developed in an organized manner.

The layered architecture is suitable for QMS-Flow because it is a web-based system that must support multiple quality management modules such as Training Management, Corrective Action (CAPA), Equipment and Device Management, and Management Review. By separating the system into layers, each QMS module can be developed, maintained, and upgraded independently without affecting other parts of the system. This approach supports incremental development, allowing new modules or features to be added over time as organizational requirements evolve.

Another important reason for selecting this architectural style is its ability to localize changes within the system. When changes and modifications are made to one layer, only the adjacent layers are affected. This reduces the risk of system-wide errors and improves maintainability. For QMS-Flow, this is especially important as ISO 9001:2015 requirements may change or require updates to processes and documentation.

Moreover, layered architecture supports multi-level access control and security, which is important for QMS-Flow since there are different user roles. Security checks and authorization rules can be enforced in appropriate layers so that users only access functions permitted by their role. Thus, the best choice for the QMS-Flow system is a layered structure.

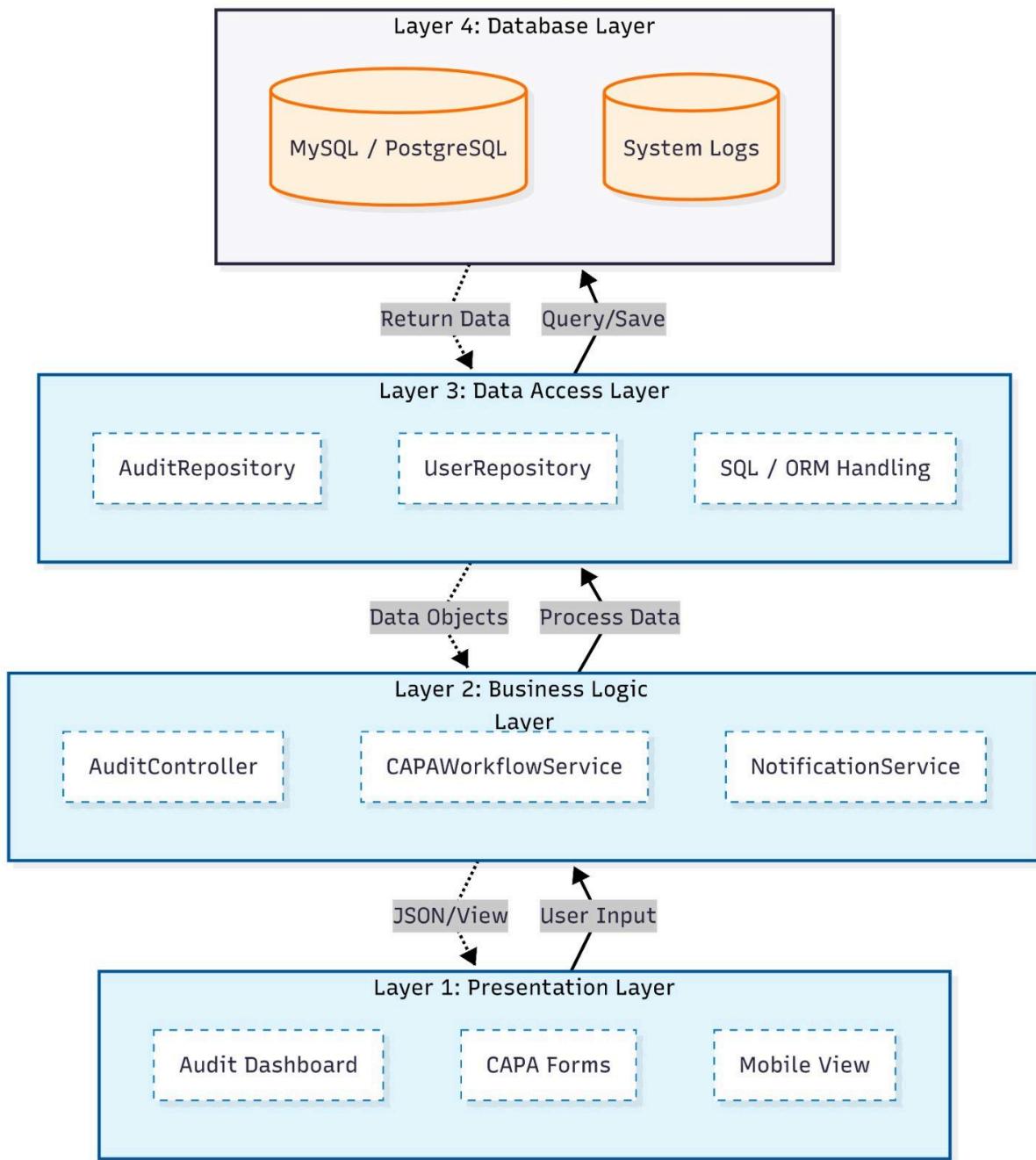


Figure 3.1: Layered Architecture Diagram of QMS Flow

3.2 Component Model

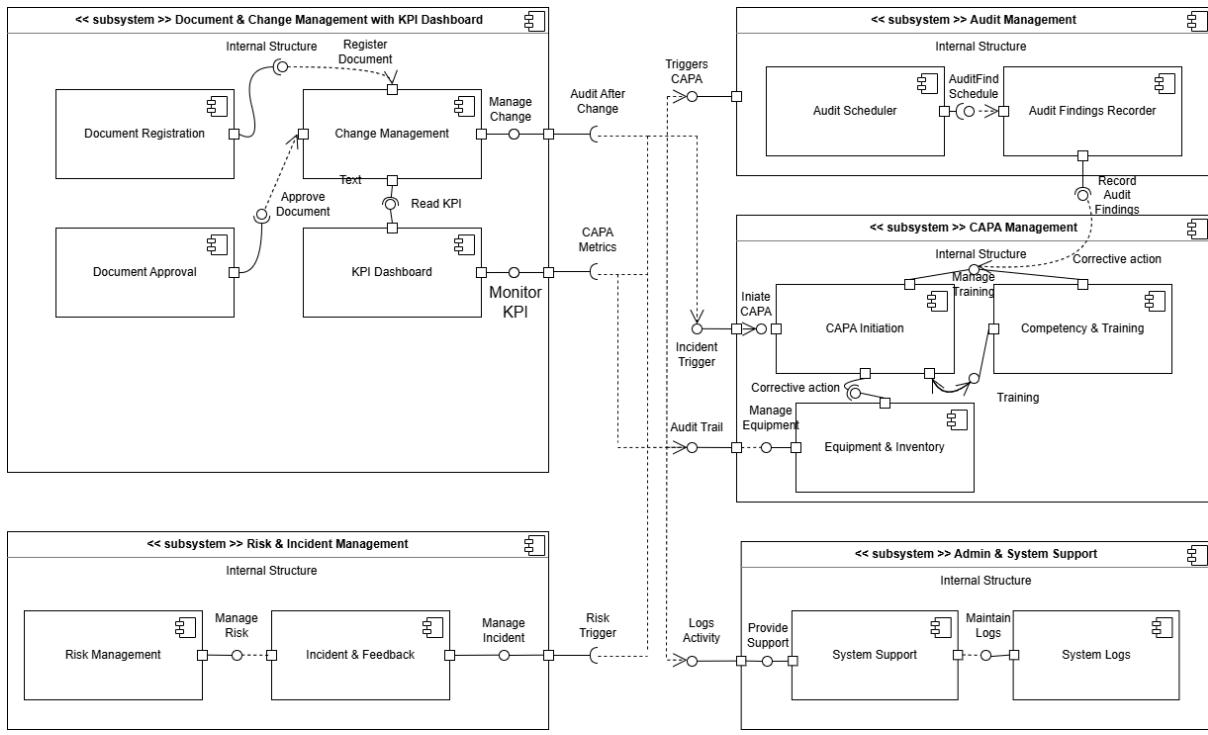


Figure 3.2: Component Diagram of QMS Flow

4 Detailed Description of Components

4.1 Complete Package Diagram

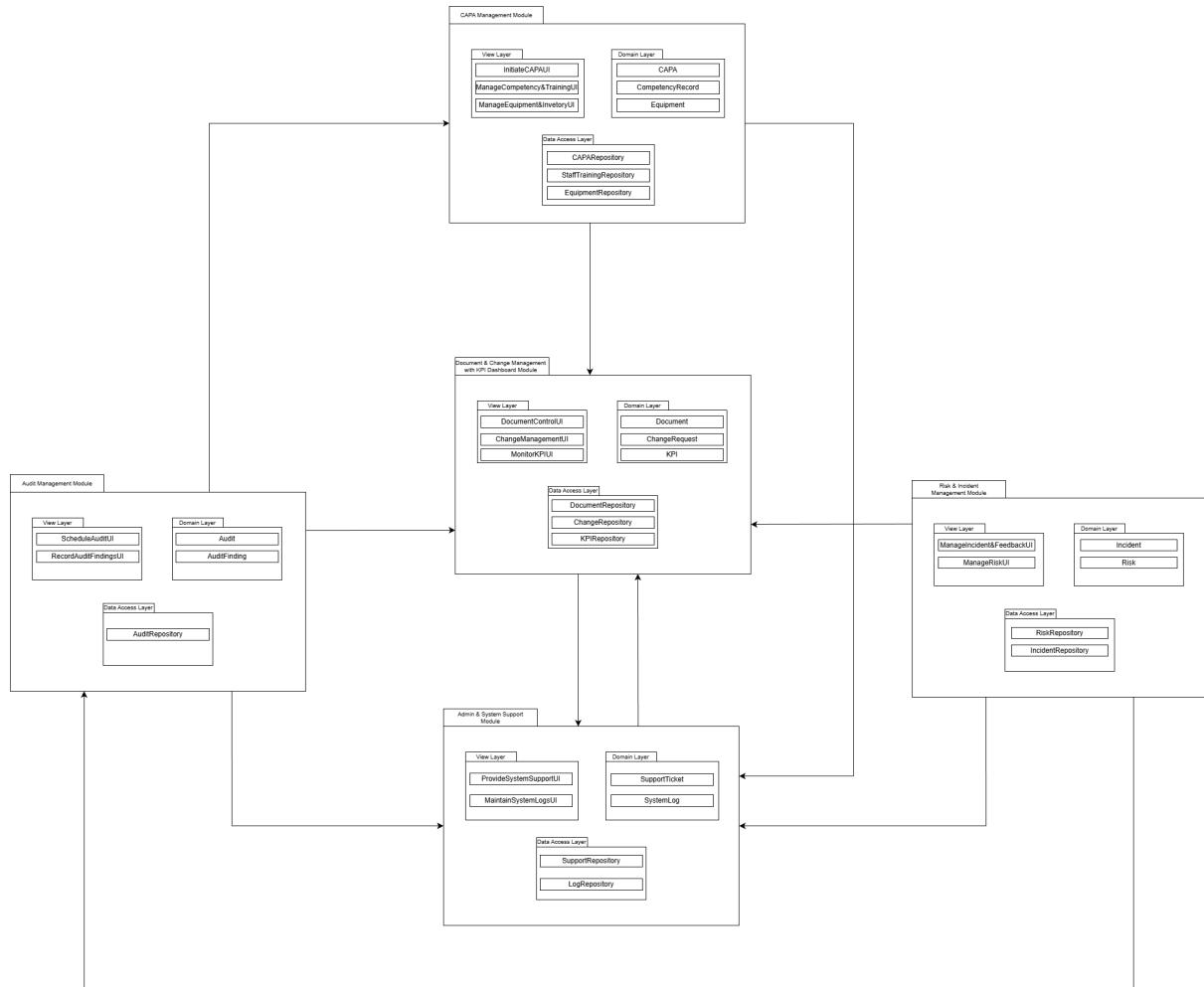


Figure 4.1: Package Diagram for QMS Flow

4.2 Detailed Description

4.2.1 P001: Audit Management Module Subsystem

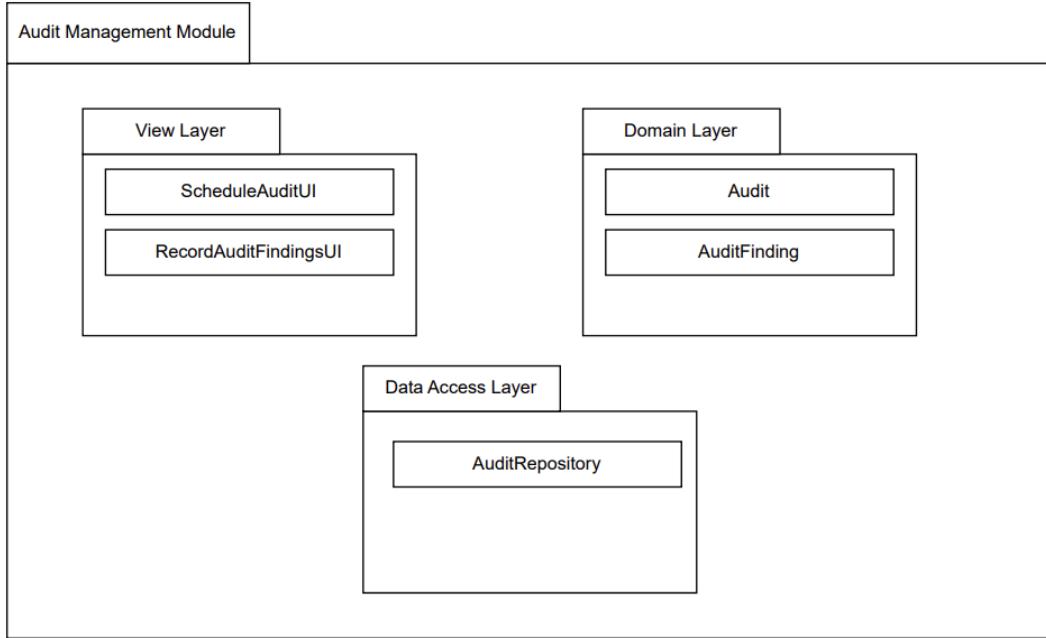


Figure 4.2.1: Package Diagram for Audit Management Module Subsystem

4.2.1.1 Class Diagram

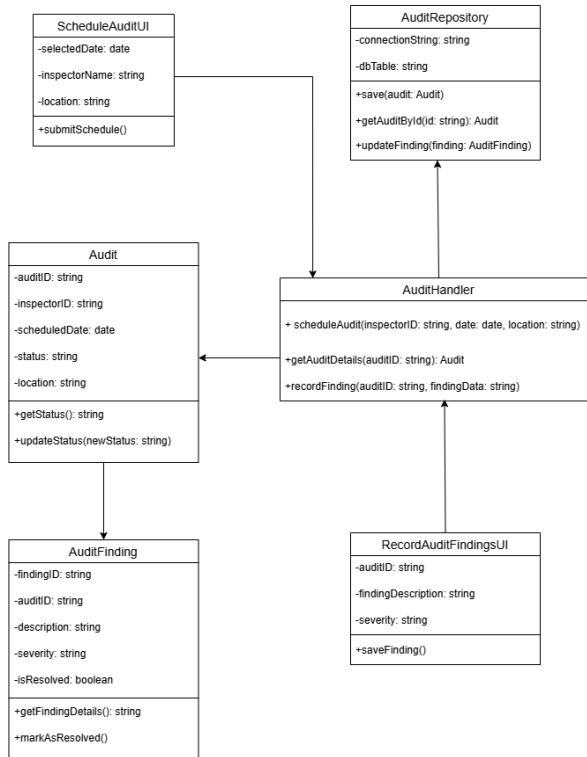


Figure 4.2.2: Class Diagram for Audit Management Module Subsystem

Entity Name	Audit
Method Name	ScheduleAuditUI
Input	Audit Date, Scope, Assigned Auditor(s)
Output	Confirmed Audit Schedule
Algorithm	<ol style="list-style-type: none"> 1. Start 2. Manager selects "Schedule Audit" 3. Input audit parameters (dates, scope) and assign Auditors 4. Validate auditor availability 5. Save schedule to Audit Repository 6. End

Entity Name	AuditFinding
Method Name	RecordAuditFindingsUI
Input	Audit Evidence, Non-conformity Details
Output	Stored Audit Report / Triggered CAPA
Algorithm	<ol style="list-style-type: none"> 1. Start 2. Auditor selects active audit 3. Log evidence and specific findings 4. If critical finding exists, trigger UC05: Initiate CAPA 5. Update Audit status to "Completed" 6. End

4.2.1.2 Sequence Diagram

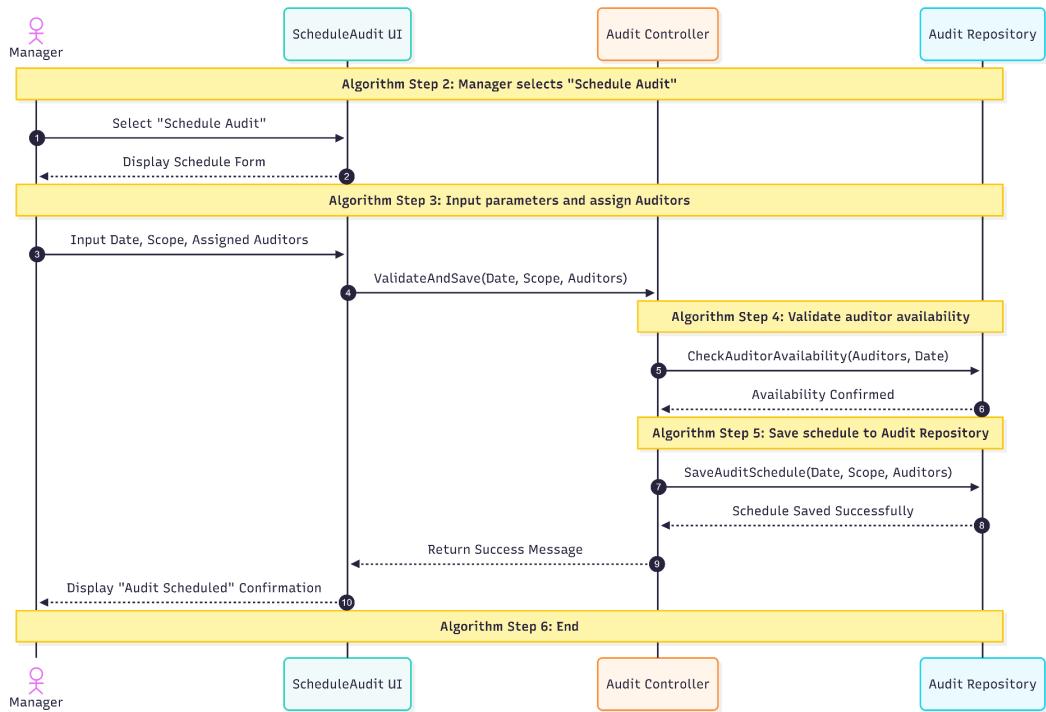
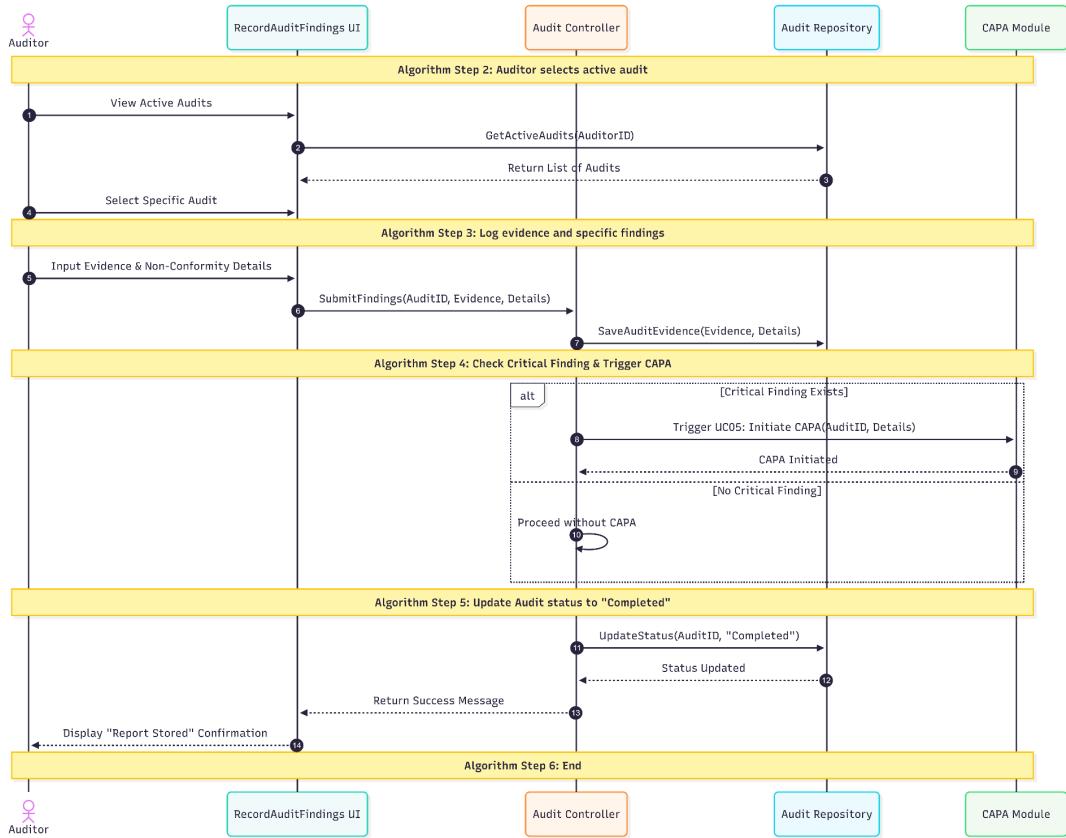


Figure 4.2.3: sequence diagram for ScheduleAuditUI



4.2.4: sequence diagram for RecordAuditFindingsUI

4.2.2 P002: Document & Change Management with KPI Dashboard Module Subsystem

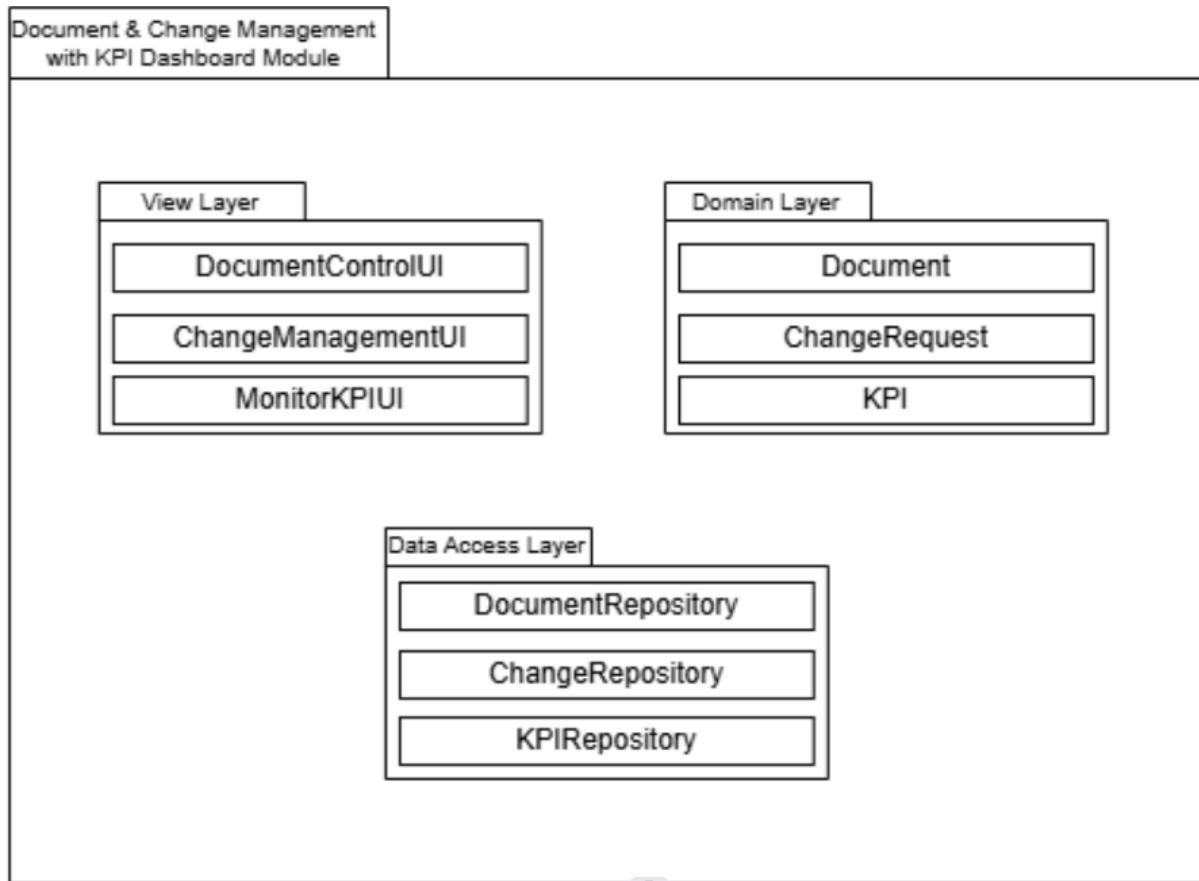


Figure 4.3.1: Package Diagram for Document & Change Management with KPI Dashboard Module Subsystem

4.2.2.1 Class Diagram

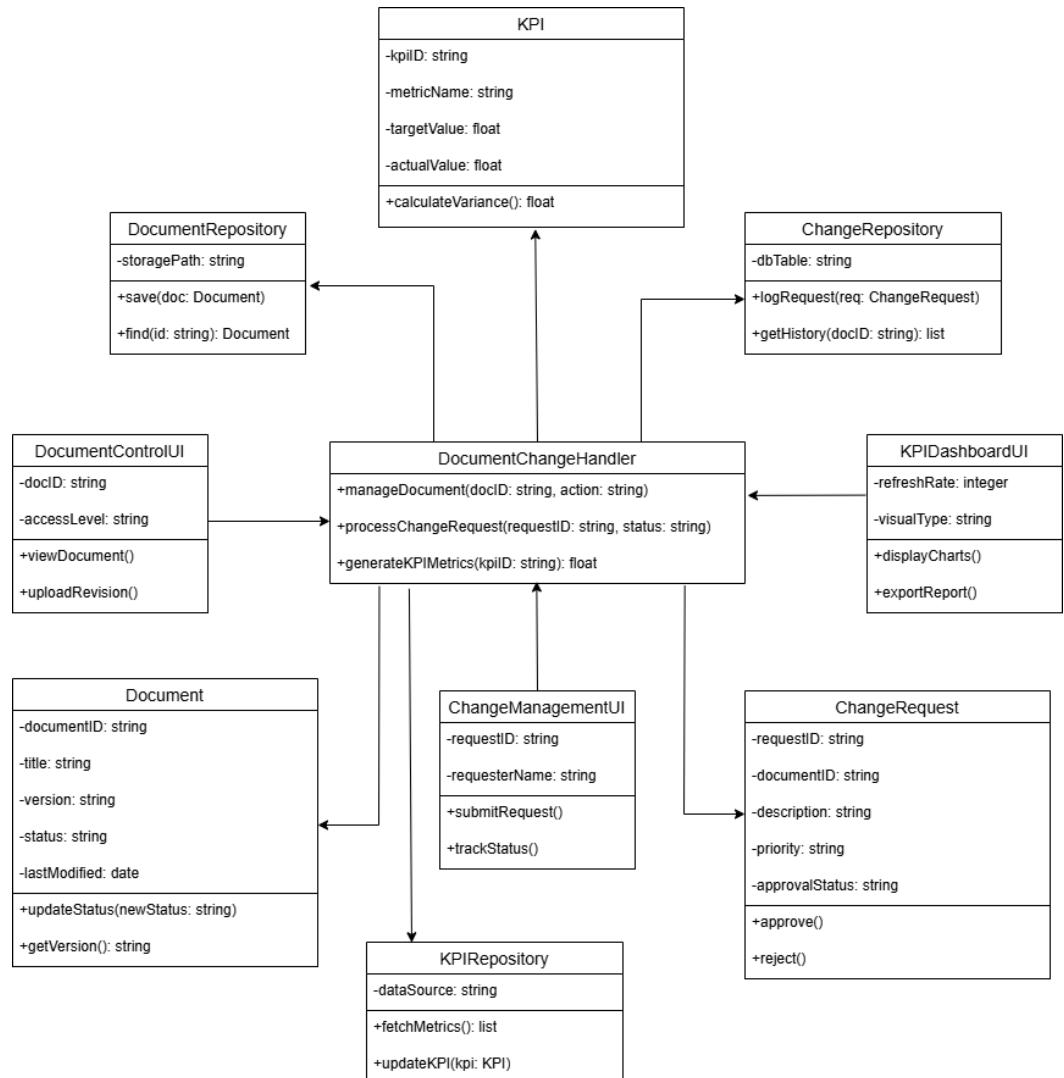


Figure 4.3.2: class diagram for Document & Change Management with KPI Dashboard
Module Subsystem

Entity Name	Document
Method Name	DocumentControlUI
Input	Document File, Metadata
Output	Formal Approved Document for Traceability
Algorithm	<p>1. Start</p> <p>2. Upload & Metadata: Staff captures new document file and enters metadata (Title, Version, Owner) for traceability.</p> <p>3. Validation: System checks for duplicate IDs and assigns "Draft" status.</p> <p>4. Review Queue: Document is routed to the Manager or Auditor for review.</p> <p>5. Decision Logic: Manager reviews files against compliance requirements to ensure they are formally approved before use.</p> <p>6. Approval: If requirements are met, the system updates status to "Approved" and makes it available for use.</p> <p>7. Rejection: If requirements are not met, system records comments and returns the file to "Draft" for revision.</p> <p>8. Log Update: Every status change is automatically recorded in the System Logs (UC11) for accountability.</p> <p>9. End</p>

Entity Name	KPIReport
Method Name	MonitorKPIUI
Input	Performance Data from all modules
Output	Real-time Performance Dashboards
Algorithm	<ol style="list-style-type: none"> 1. Start 2. System aggregates data from Audit, CAPA, and Training 3. Calculate metrics against targets 4. Display on Manager's dashboard 5. If KPI < target, trigger UC05: Initiate CAPA 6. End

Entity Name	ChangeRequest
Method Name	ChangeManagementUI
Input	Proposed Change, Rationale
Output	Process/Document Update
Algorithm	<ol style="list-style-type: none"> 1. Start 2. Staff proposes process/document changes 3. Manager/Admin reviews proposal 4. If approved, route to UC02 for final document approval 5. Update change logs 6. End

4.2.2.2 Sequence Diagram

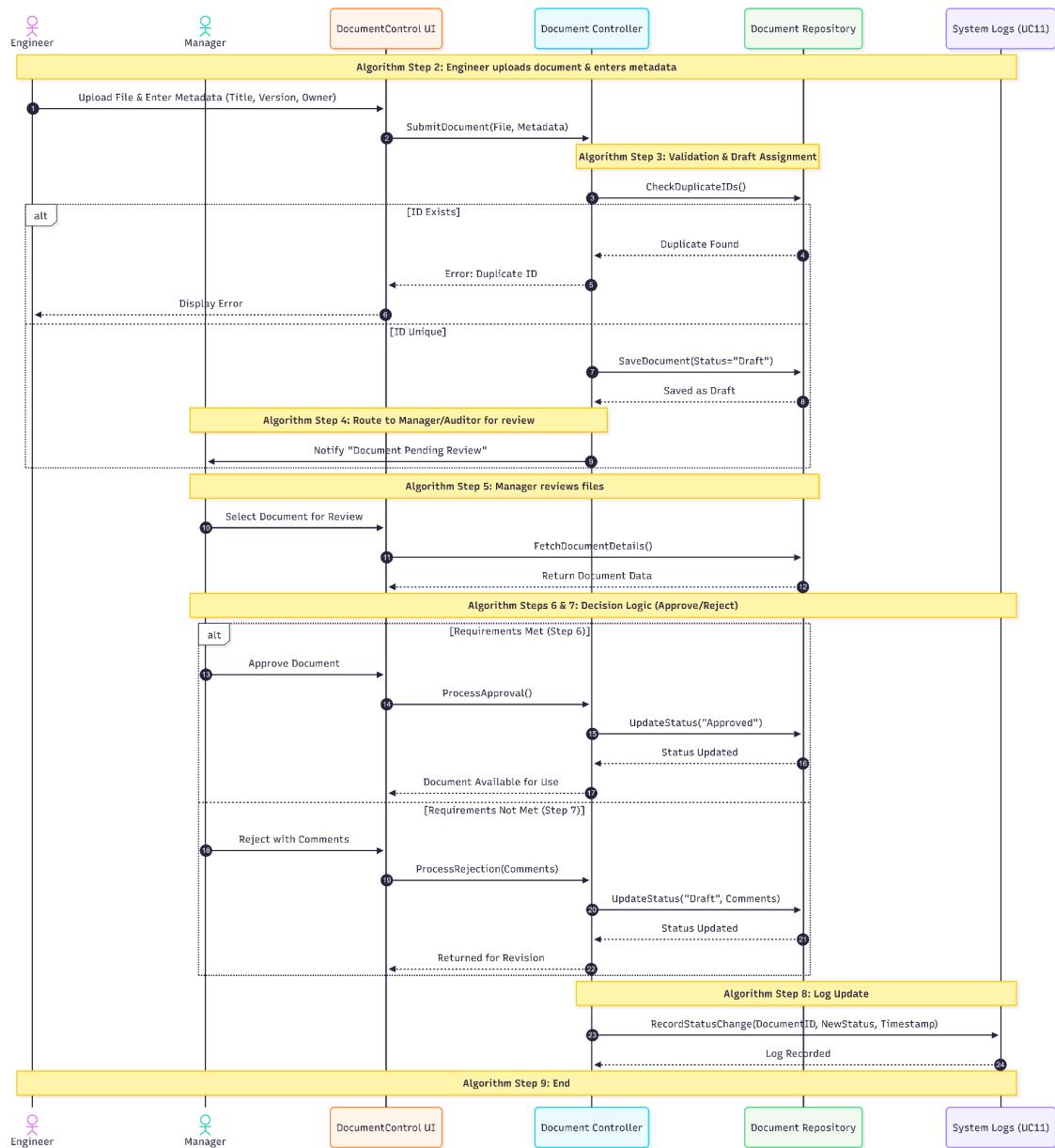


figure 4.3.3: sequence diagram for DocumentControlUI

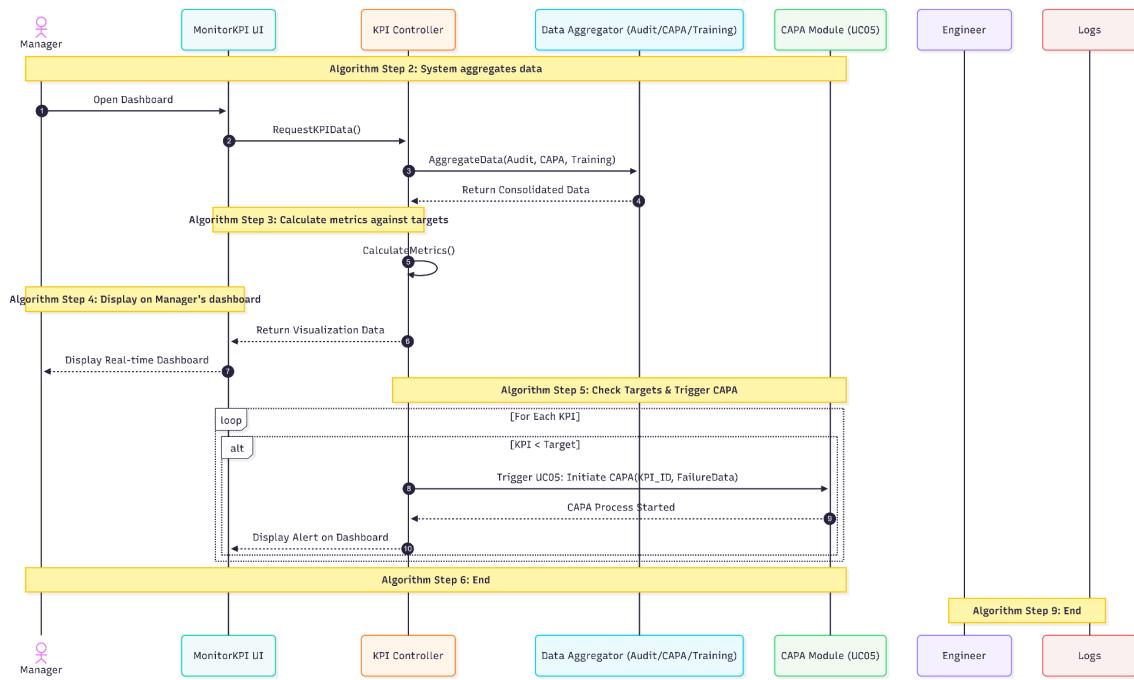


figure 4.3.4: sequence diagram for MonitorKPIUI

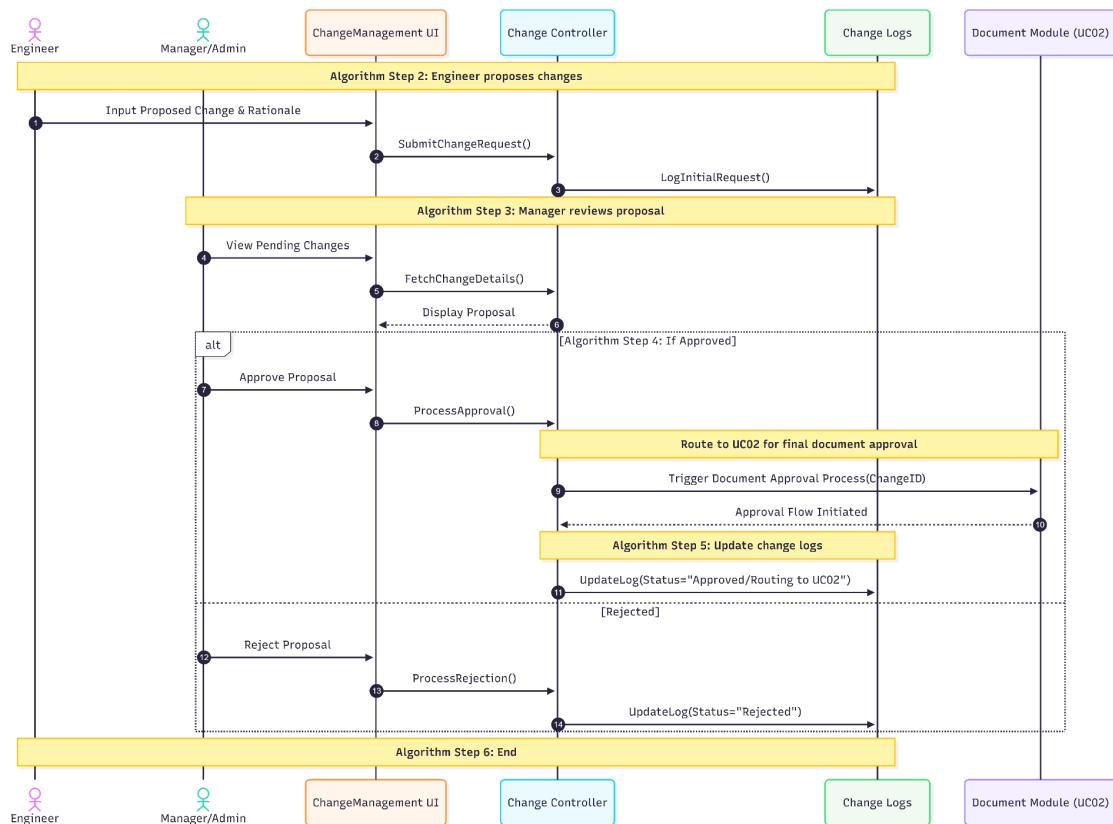


figure 4.3.5: sequence diagram for ChangeManagementUI

4.2.3 P003: Admin & System support Module Subsystem

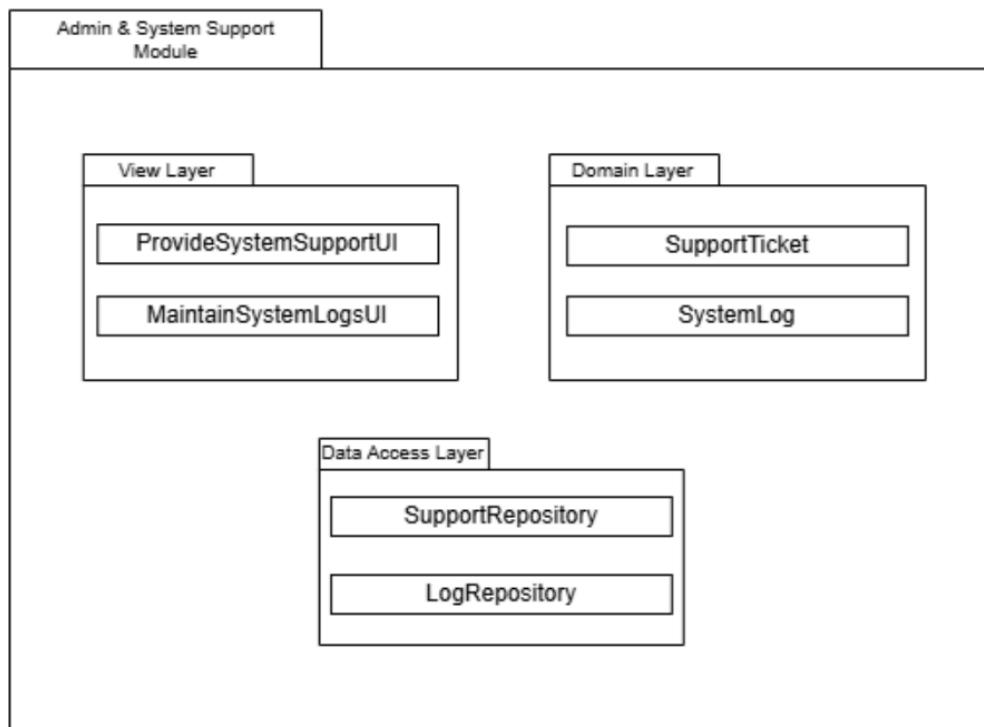


Figure 4.4.1: Package Diagram for Admin & System support Module Subsystem

4.2.3.1 Class Diagram

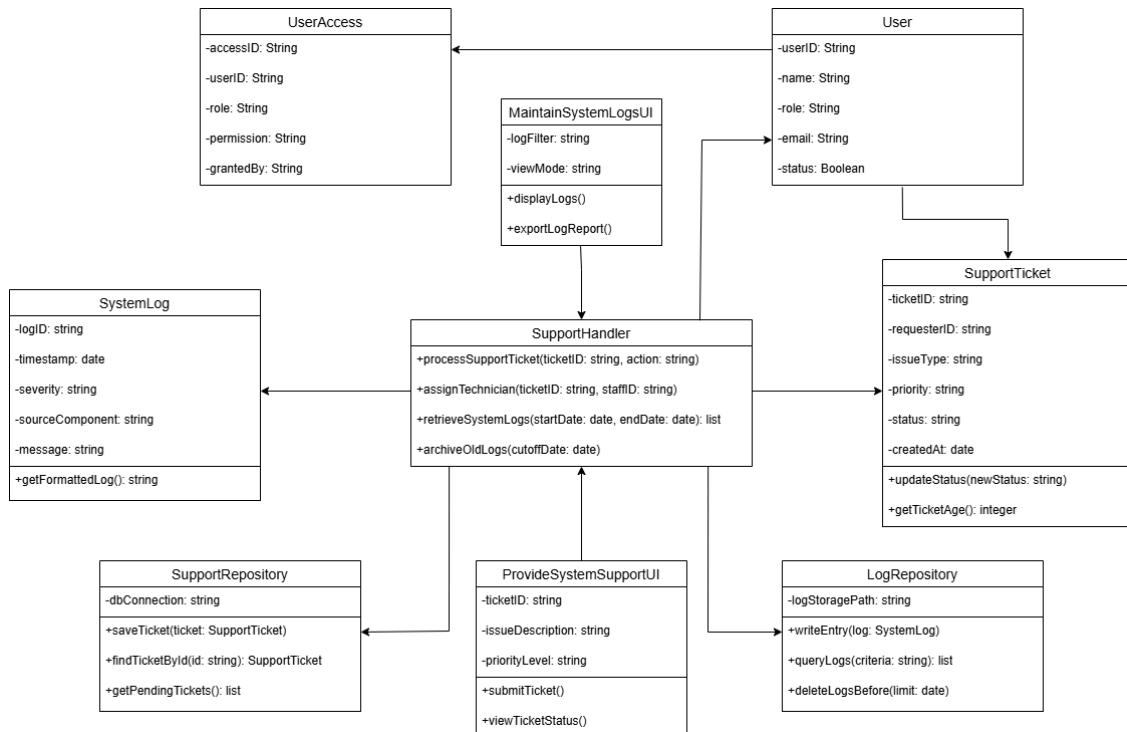


Figure 4.4.2: Class Diagram For Admin & System support Module Subsystem

Entity Name	SupportTicket
Method Name	ProvideSystemSupportUI
Input	Inquiry Type, Issue Description
Output	Resolution to Inquiry
Algorithm	<ol style="list-style-type: none"> 1. Start 2. User submits support request 3. Admin reviews and provides technical support 4. Update ticket to "Resolved" 5. End

Entity Name	SystemLog
--------------------	-----------

Method Name	MaintainSystemsLogsUI
Input	System Transactions, User ID
Output	Audit Trail for Accountability
Algorithm	<ol style="list-style-type: none"> 1. Start 2. Automatically capture every user action across modules 3. Store timestamped activity data 4. Provide log access for Admin review 5. End

4.2.3.2 Sequence Diagram

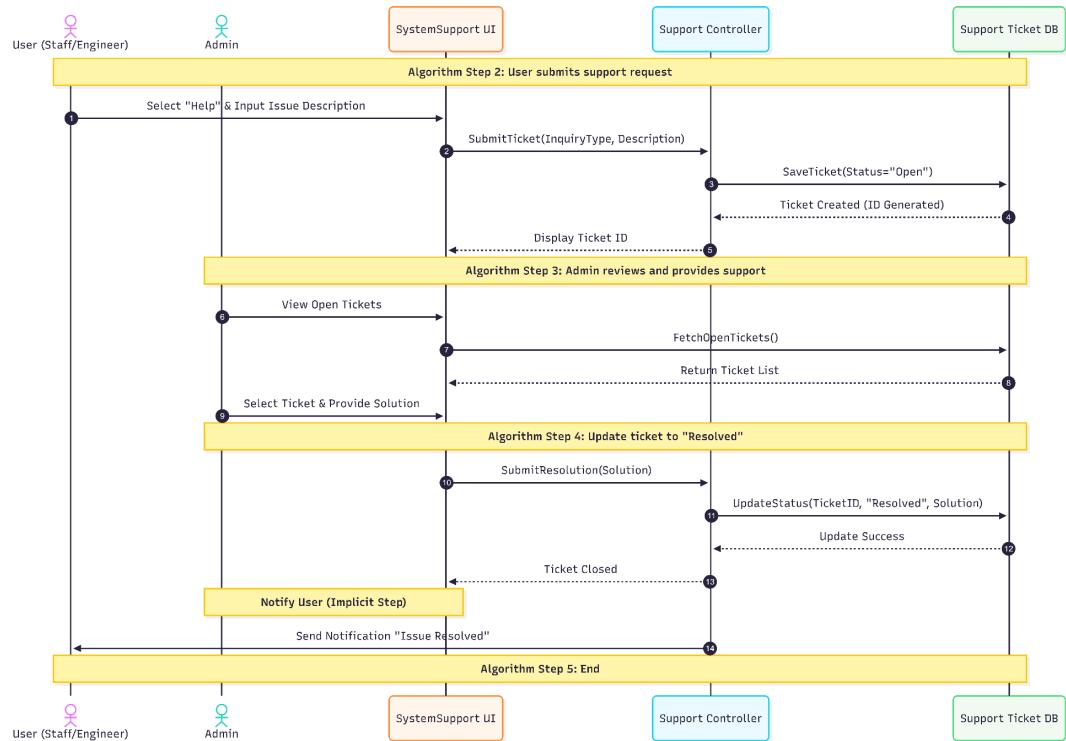


Figure 4.4.3: Sequence Diagram For ProvideSystemSupportUI

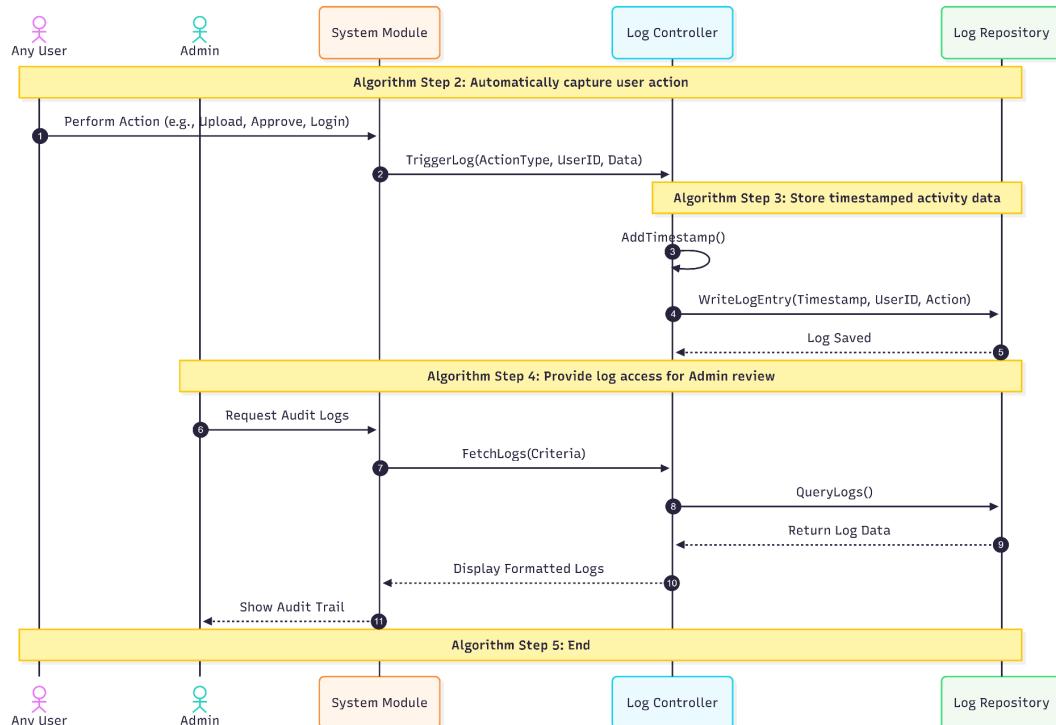


Figure 4.4.4: Sequence Diagram For MaintainSystemsLogsUI

4.2.4 P004: CAPA Management Module Subsystem

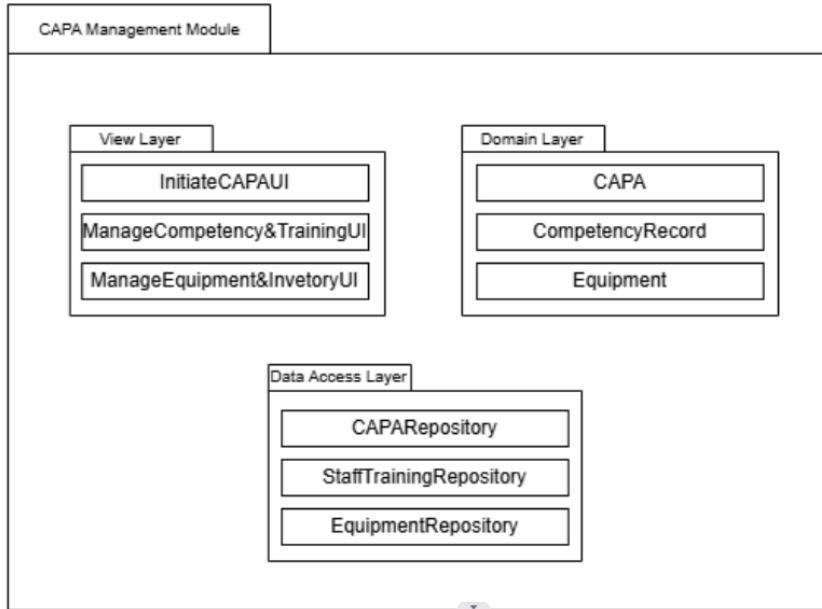


Figure 4.4.1: Package Diagram for CAPA Management Module Subsystem

4.2.4.1 Class Diagram

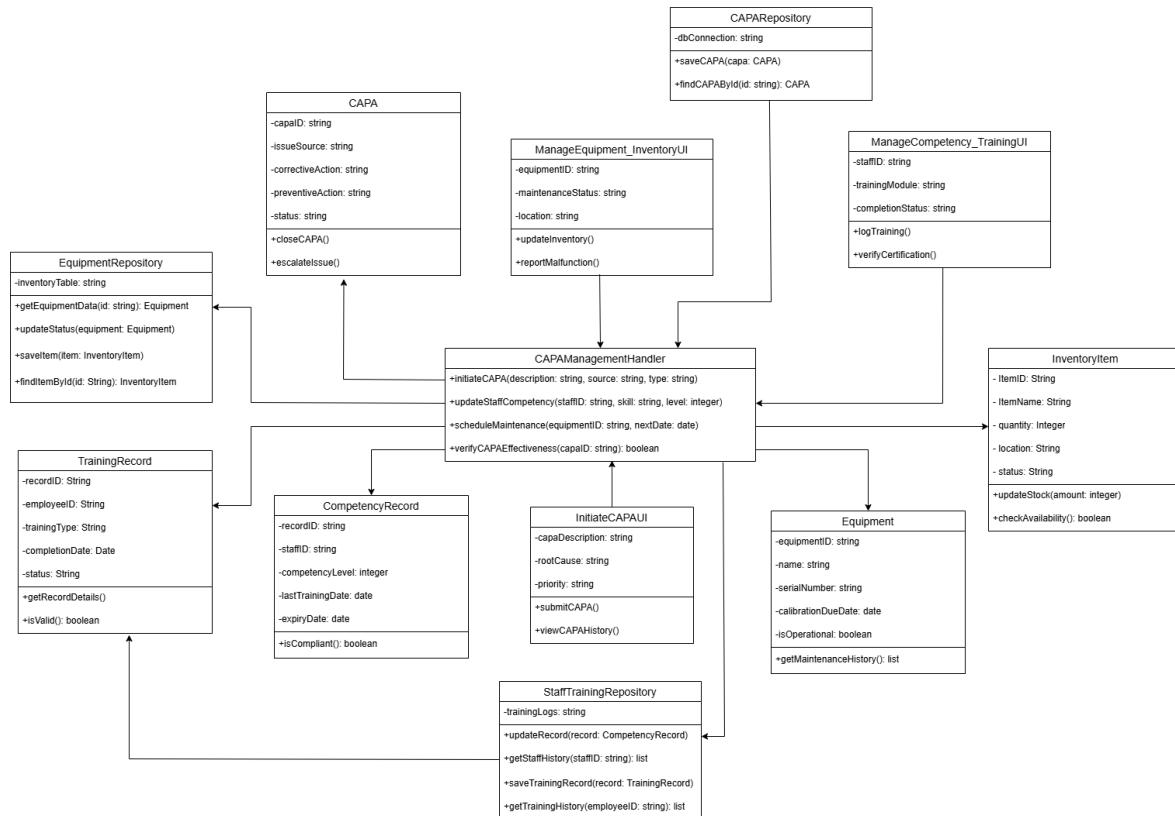


Figure 4.4.2: Class Diagram for Management Module Subsystem

Entity Name	CAPA
Method Name	IniateCAPAUI
Input	Source Issue (Audit/Incident/KPI), Root Cause
Output	Open Corrective Action Plan
Algorithm	<ol style="list-style-type: none"> 1. Start 2. Receive trigger from Audit, KPI, or Risk 3. Log issue and perform root cause analysis 4. Assign action owner and deadline 5. End

Entity Name	CompetencyRecord
Method Name	ManageCompetency&TrainingUI
Input	Staff ID, Training Requirements, Evaluation Results
Output	Updated Staff Competency Log
Algorithm	<ol style="list-style-type: none"> 1. Start 2. HR/Manager identifies training gap (often from CAPA) 3. Schedule training session 4. Record attendance and evaluation score 5. Update staff competency status 6. End

Entity Name	Equipment
Method Name	ManageEquipment&InventoryUI
Input	Asset ID, Maintenance Schedule, Calibration Data
Output	Equipment Readiness Status
Algorithm	<ol style="list-style-type: none"> 1. Start 2. System checks calibration schedules 3. Engineer performs maintenance/calibration 4. Log results and update "Ready" status 5. End

4.2.4.2 Sequence Diagram

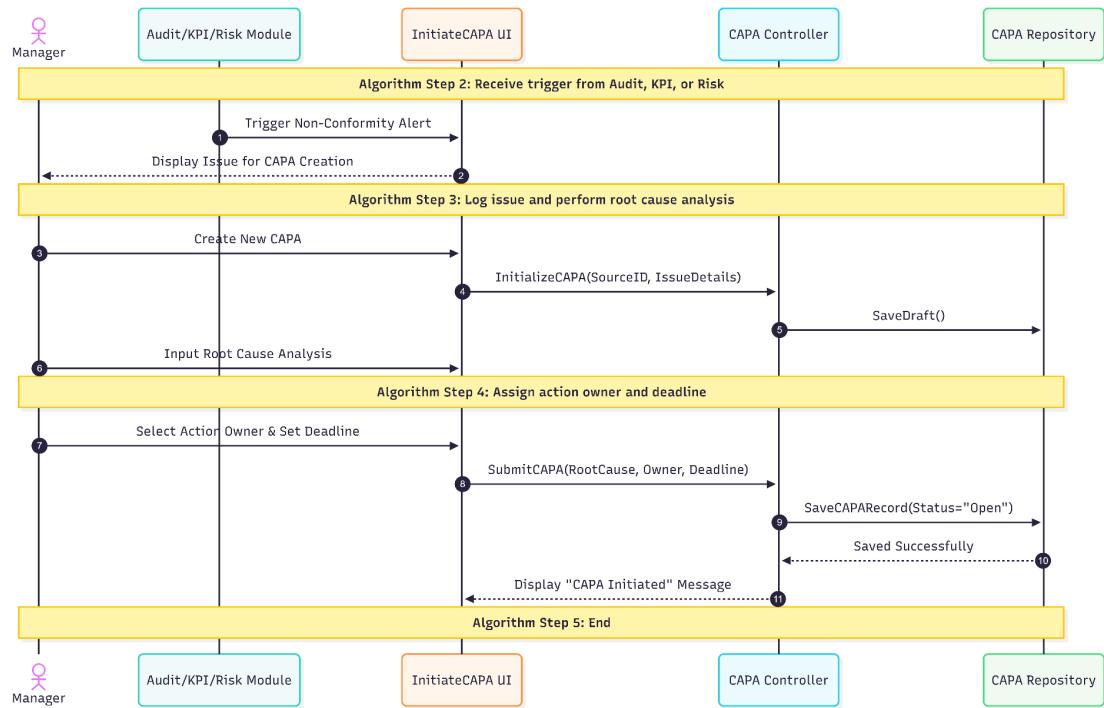


Figure 4.4.3: sequence Diagram for IniateCAPAUUI

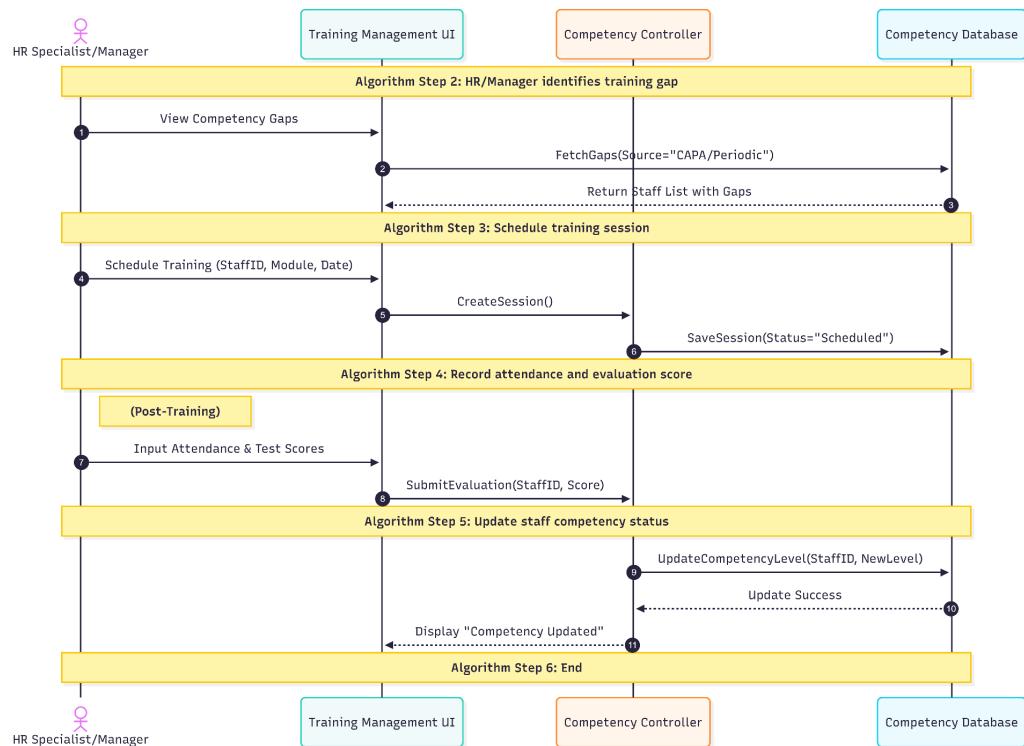


Figure 4.4.4: sequence Diagram for ManageCompetency&TrainingUI

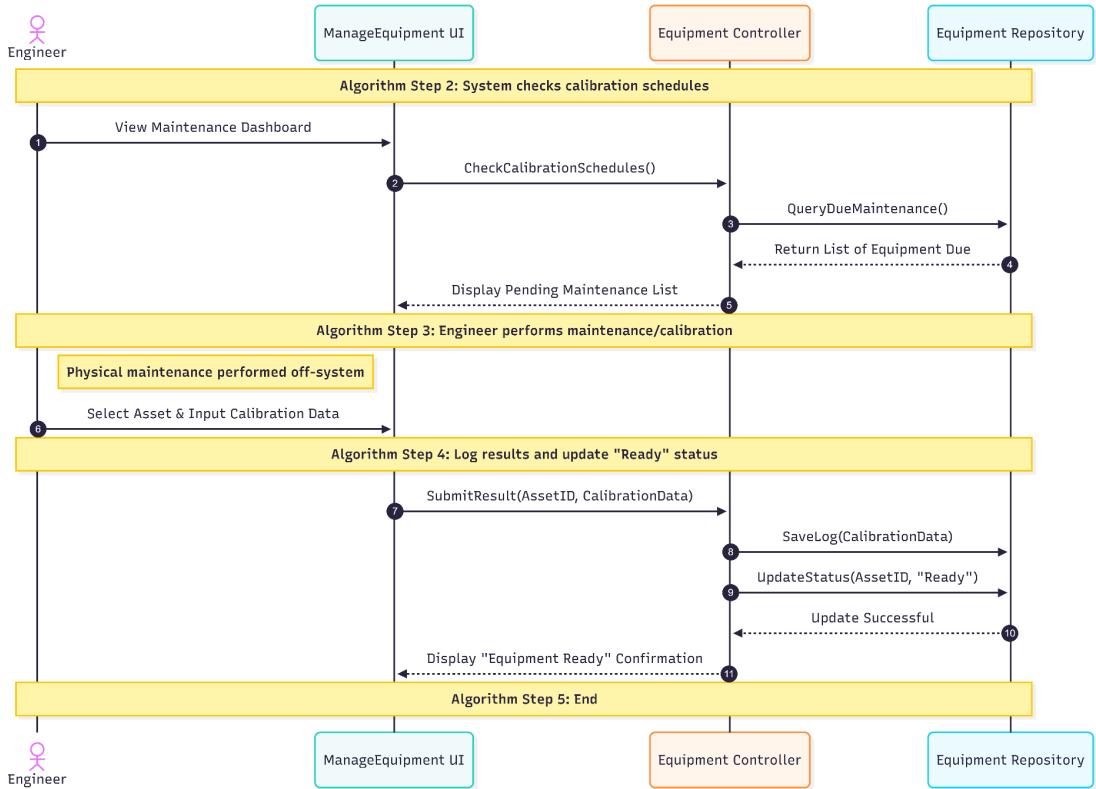


Figure 4.4.5: sequence Diagram for ManageEquipment&InventoryUI

4.2.5 P005: Risk & Incident Management Module

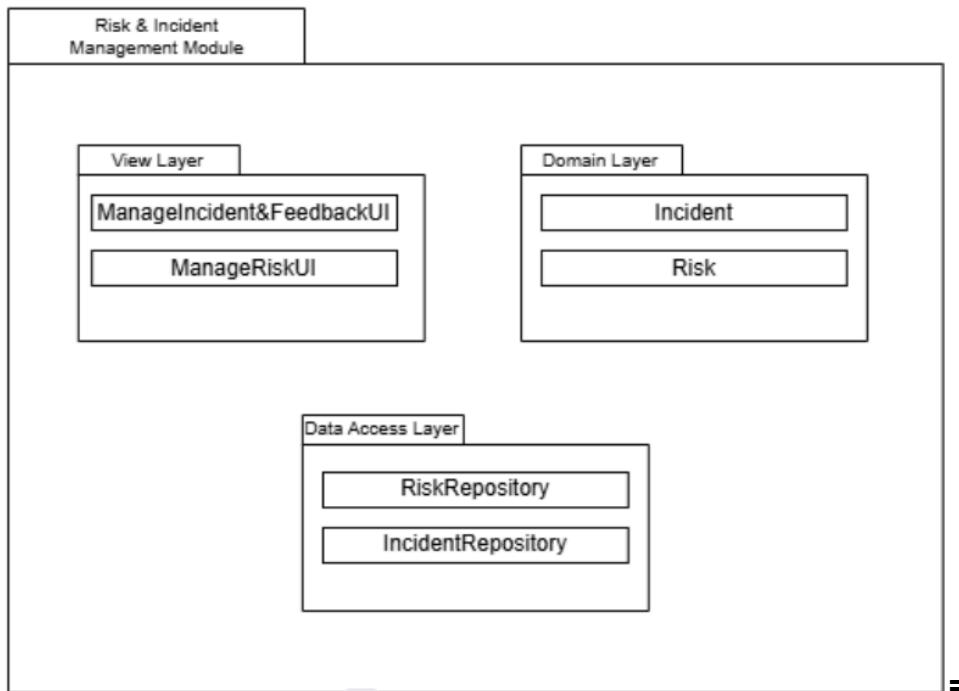


Figure 4.5.1: Package Diagram for Risk & Incident Management Module Subsystem

4.2.5.1 Class Diagram

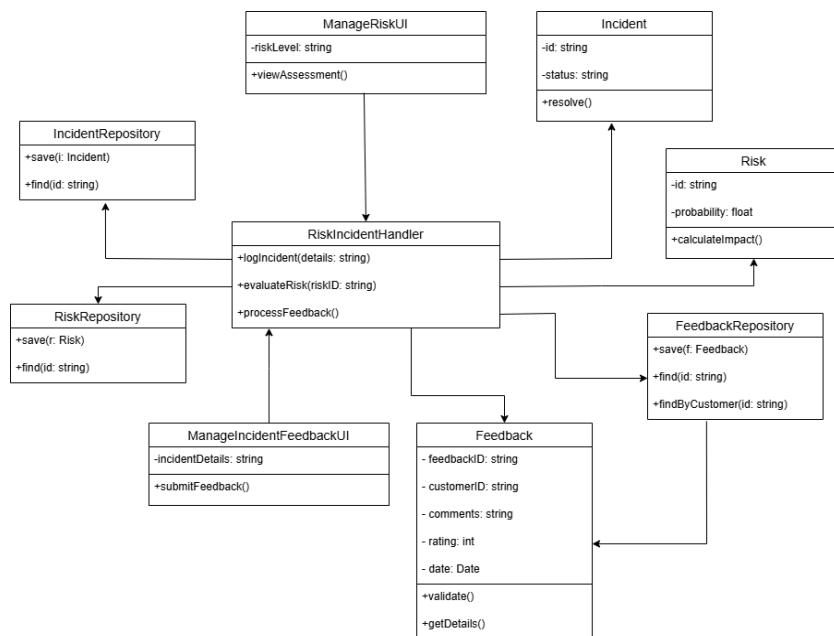


Figure 4.5.2: Class Diagram for Risk & Incident Management Module Subsystem

Entity Name	RiskProfile
Method Name	ManageRiskUI
Input	Potential Risk, Likelihood, Impact
Output	Risk Register & Mitigation Plan
Algorithm	<ol style="list-style-type: none"> 1. Start 2. Identify potential process/compliance risks 3. Score risk priority (Likelihood x Impact) 4. Define and track mitigation actions 5. End

Entity Name	IncidentReport
Method Name	ManageIncident&FeedbackUI
Input	Customer Feedback, Incident Description, Evidence
Output	Logged Incident/Complaint Record
Algorithm	<ol style="list-style-type: none"> 1. Start 2. Capture feedback or report nonconformity 3. Classify severity 4. Route to UC05: Initiate CAPA if corrective action is needed 5. End

4.2.5.2 Sequence Diagram

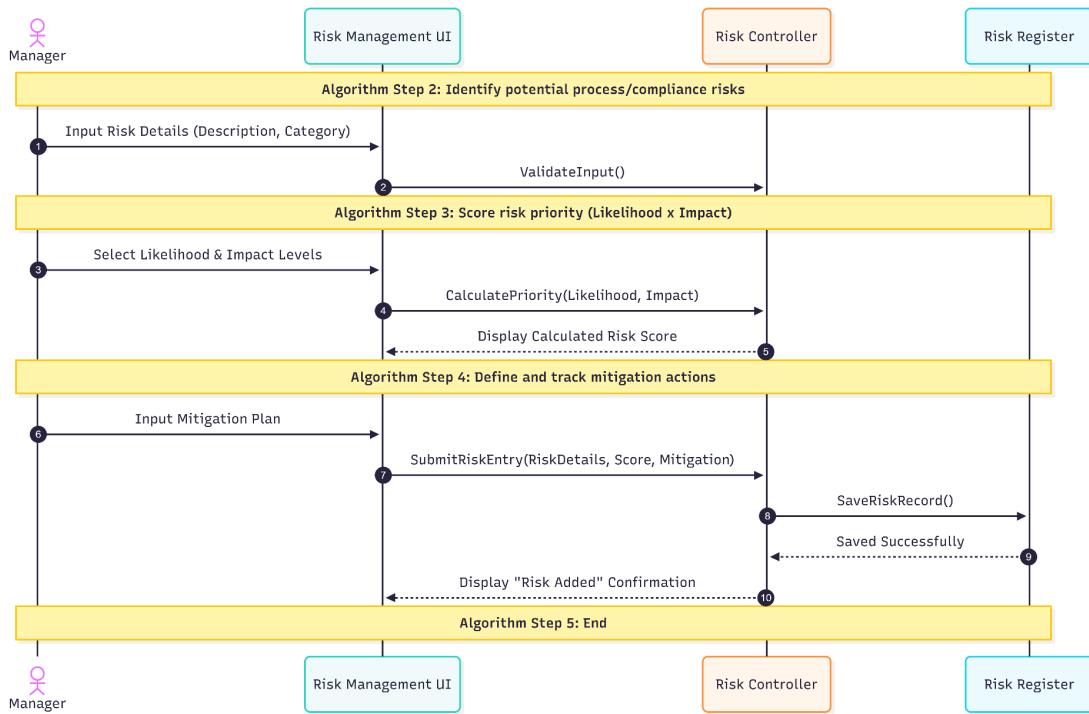


Figure 4.5.3: Class Diagram for ManageRiskUI

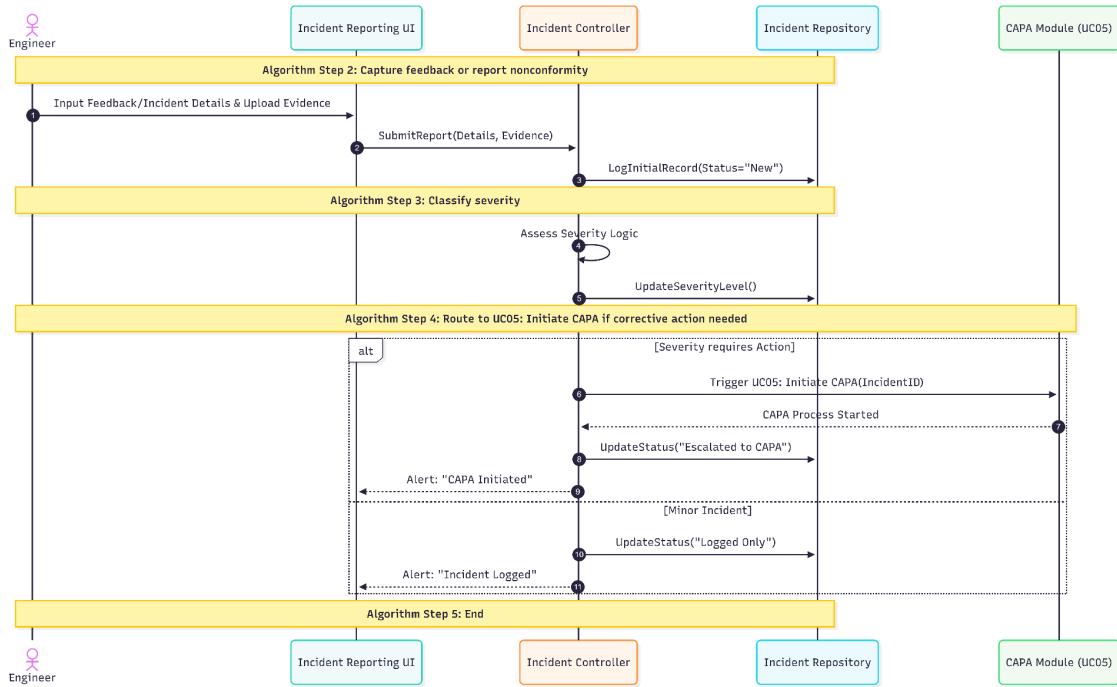


Figure 4.5.4: Class Diagram for ManageIncident&FeedbackUI

5 Data Design

5.1 Data Description

The major data or system entities are stored in a relational database named QMSFlowDB and processed and organized into 18 entities as listed in Table 5.1. Data is organized into normalized tables to reduce redundancy and maintain consistency.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1.	Approval	Stores approval records for documents and change requests.
2.	Audit	Stores audit schedules, scope, and assigned auditors.
3.	AuditFinding	Stores findings and nonconformities identified during audits.
4.	CAPA	Stores corrective and preventive action records linked to findings, incidents, or risks.
5.	ChangeRequest	Stores requests for document or process changes.
6.	CompetencyAssessment	Stores employee competency evaluation results.
7.	Document	Stores SOPs, policies, and controlled documents with versioning and approval workflows.
8.	Equipment	Stores equipment records, calibration dates, and status.
9.	Feedback	Stores customer complaints, suggestions, and satisfaction ratings.
10.	Incident	Stores reported incidents and nonconformities.

11.	Inventory	Represents items tracked in the system, linked to equipment and CAPA when issues arise.
12.	Investigation	Stores investigations into incidents, including root cause and resolution.
13.	KPI	Stores performance metrics and results
14.	Risk	Stores identified risks, likelihood, impact, and mitigation plans.
15.	SystemLog	Stores system activity logs for accountability and troubleshooting.
16.	TrainingRecord	Stores employee training records and completion status.
17.	User	Stores system users (engineers, managers, auditors, customers).
18.	UserAccess	Stores user roles and permissions granted by admins.

5.2 Data Dictionary

5.2.1 Entity: <Approval>

Attribute Name	Type	Description
approvalID	String	Unique approval record
documentID	String	Links to document
approverID	String	User ID of approver
approvalDate	Date	Date of approval
status	String	Pending/Approved/Rejected

5.2.2 Entity: <Audit>

Attribute Name	Type	Description
auditID	String	Unique audit record
inspectorID	String	Identifies which qualified inspector
scheduleDate	Date	Date when audit is scheduled
scope	String	Scope covered
auditorID	String	Linked to User (auditor)
status	String	Planned/In Progress/Closed

5.2.3 Entity: <AuditFinding>

Attribute Name	Type	Description
findingID	String	Unique finding record
auditID	String	Linked to Audit
description	String	Details of finding
severity	String	Minor/Major/Critical
isResolved	boolean	Check

5.2.4 Entity: <CAPA>

Attribute Name	Type	Description
capaID	String	Unique CAPA record
IssueSource	String	origin or trigger of the CAPA
correctiveAction	String	description of the immediate steps taken to address
preventiveAction	String	description of the long-term measures implemented
status	String	Raised/Implemented/Closed

5.2.5 Entity: <ChangeRequest>

Attribute Name	Type	Description
description	String	A detailed explanation
documentID	String	Linked to Document
requesterID	String	Linked to User
priority	String	The level of urgency assigned
approvalStatus	String	Requested/Approved/Rejected/Implemented
requestDate	Date	Date submitted

5.2.6 Entity: <CompetencyAssessment>

Attribute Name	Type	Description
assessmentID	String	Unique assessment record
employeeID	String	Linked to User
assessmentDate	Date	Date of assessment
result	String	Pass/Fail/Needs Improvement

5.2.7 Entity: <Document>

Attribute Name	Type	Description
documentID	String	Unique document Identifier
title	String	Document title
version	string	Version number
status	String	Draft/Approved/Archived
lastModified	Date	Date modified

5.2.8 Entity: <Equipment>

Attribute Name	Type	Description
equipmentID	String	Unique equipment record
name	String	Equipment name
serialNumber	String	The unique manufacturer-provided identification number
calibrationDueDate	String	scheduled date by which the equipment must be recalibrated
isOperational	boolean	status flag indicating if the equipment is currently functional

5.2.9 Entity: <Feedback>

Attribute Name	Type	Description
feedbackID	String	Unique feedback record
customerID	String	Linked to User (customer)
comments	String	Feedback text
rating	integer	Satisfaction rating (1-5)
date	Date	Submission date

5.2.10 Entity: <Incident>

Attribute Name	Type	Description
id	String	Unique incident record
status	String	Reported/Investigating/Closed

5.2.11 Entity: <InventoryItem>

Attribute Name	Type	Description
ItemID	String	Unique Item Record
ItemName	String	Name Of Item
quantity	Integer	Stock Quantity
location	String	Storage Location
status	String	Available/Consumed

5.2.12 Entity: <Investigation>

Attribute Name	Type	Description
investigationID	String	Unique investigation record
incidentID	String	Linked to incident
assignedTo	String	Linked to User
rootCause	String	Root cause analysis
resolution	String	Resolution details
completionDate	Date	Date completed

5.2.13 Entity: <KPI>

Attribute Name	Type	Description
kpiID	String	Unique KPI record
metricName	String	KPI name
targetValue	Double	Target threshold
actualValue	Double	Actual measured value

5.2.14 Entity: <Risk>

Attribute Name	Type	Description
id	String	Unique risk record
probability	float	Probability value

5.2.15 Entity: <SystemLog>

Attribute Name	Type	Description
logID	String	Unique log entry
timeStamp	Date	Log time
severity	String	classification of the log entry importance
sourceComponent	String	Linked to User/Admin
message	String	text description providing specific details

5.2.16 Entity: <TrainingRecord>

Attribute Name	Type	Description
recordID	String	Unique training record
employeeID	String	Linked to User
trainingType	String	Training category
completionDate	Date	Completion date
status	String	Completed/Pending

5.2.17 Entity: <User>

Attribute Name	Type	Description
userID	String	Unique user identifier
name	String	Full name
role	String	Admin/Manager/Auditor/Engineer/Customer
email	String	Contact email
status	Boolean	Active/Inactive

5.2.18 Entity: <UserAccess>

Attribute Name	Type	Description
accessID	String	Unique access record
userID	String	Linked to User
role	String	Role assigned
permission	String	Access rights
grantedBy	String	Linked to User (Admin)

6 User Interface Design

6.1 Overview of User Interface

The interface for QMS-FLOW is aimed at providing a simple, secure, and user-friendly environment to support quality management according to ISO 9001:2015. The interface is designed to ensure that the system is clear and consistent to enable the factory personnel to move easily through the system without much training. The system has a dynamic web-based design. For desktop users, there is a persisting sidebar to navigate through, and for mobile users such as floor operators, there is a simplified version of the interface designed for handheld interaction.

Key Design Features:

- Role-Based Access Control (RBAC): The user's role determines how the interface changes dynamically and guarantees security:
 - Engineers: Get access to a simple dashboard for tracking training status and recording complaints.
 - Managers: Examine team competency matrices, equipment lists, and departmental metrics.
 - Admins: Access approval processes, strategic analytics, and complete system controls.
 - Auditors: Access a restricted "Read-Only" mode for verification Purposes.
- Visual Status Indicators: The system indicates status using a standard "Traffic Light" color scheme to facilitate prompt decision-making:
 - Green: Indicates compliance or completed tasks ("Certified", "Resolved", "Active").
 - Orange/Amber: Indicates warnings or upcoming deadlines ("Due Soon", "Pending").
 - Red: Highlights critical failures or expired items ("Overdue", "Non-Compliant").
- Consistent Navigation: A standardized sidebar menu with module links (such as Dashboard, CAPA Management, and Reports) is present on the left side of every desktop screen. This guarantees that users are always aware of their location within the system.
- Interactive Dashboards: Users are shown relevant dashboards with charts and summary cards after logging in. These are more than just static displays and

they enable users to promptly recognize and take action on critical activities, such awaiting approvals.

- Data Integrity & Feedback: Every action has clear feedback from the system. Sensitive operations (such as audit trails) are monitored in thorough logs for security purposes, and certain views (such as the Auditor Dashboard) have conspicuous banners to denote limited access modes.

6.2 Screen Images

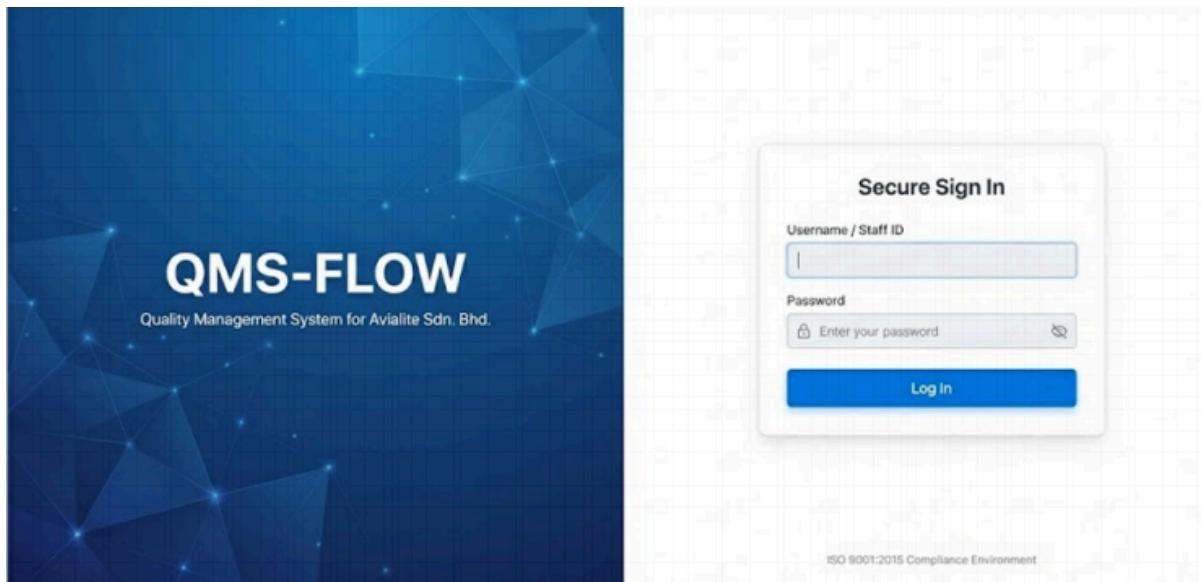


Figure 6.1 : Overview of Login Page

Description: a secure authentication screen that requires a password and a distinct username or staff ID in order to access the system.

Purpose: to tightly regulate access and guarantee that only individuals with the proper authorization can access the system.

The screenshot shows the QMS-FLOW dashboard for a 'Production Operator'. The left sidebar has a dark blue background with the 'QMS-FLOW' logo at the top. Below it are five menu items: 'Dashboard' (selected), 'Log Complaint', 'SOP Library', 'My Training', and 'Settings'. The main content area has a white background. At the top right, it says 'Logged in as: Production Operator' with a small profile icon of a person wearing a hard hat. Below that is a 'Quick Actions' section with two buttons: a blue one labeled '+ Create New Complaint Report' and a grey one labeled 'View Latest SOPs'. Underneath is a table titled 'My Recent Submissions' with columns: Date, Complaint ID, Issue Summary, and Status. The table lists five entries from April 16 to April 24, 2024. The status for the first entry is 'Pending' (orange button), while the others are 'Resolved' (green button). At the bottom of the table, it says 'Showing 1 to 5 of 25 entries' and has navigation buttons for 'Prev', 'Next', and 'Next >'. The overall interface is clean and modern.

Date	Complaint ID	Issue Summary	Status
April 24, 2024	CMP-00487	Machine misalignment causing defects	Pending
April 22, 2024	CMP-00472	Quality issue with raw materials	Resolved
April 21, 2024	CMP-00468	Calibration issue with measuring equipment	Resolved
April 18, 2024	CMP-00455	Incorrect documentation in SOP-103	Resolved
April 16, 2024	CMP-00441	Packaging defects observed	Resolved

Figure 6.2: Overview of Engineer Dashboard

Description: The primary hub of engineers, which includes “Quick Action” as generating reports and a table that displays the status of “My Recent Submissions”.

Purpose: To enable employees to monitor the status of their own submissions and promptly report problems.

The screenshot shows the QMS-FLOW application interface. On the left is a dark blue sidebar with the title 'QMS-FLOW' at the top. Below it are several menu items: 'Dashboard', 'Log Complaint', 'SOP Library', a highlighted 'My Training' button, and 'Settings'. The main content area has a white background and features the title 'My Competency Profile: Operator A' at the top. Below this, there are two status boxes: one for 'Job Role: Machine Operator' and another for 'Compliance Status: Action Required'. A large section titled 'Assigned Training Modules' contains a table with three rows. The columns are 'Module Name', 'Validity/Expiry', 'Status', and 'Action'. The first row for 'Fire Safety Protocols' shows 'Valid until Oct 2026' and 'Certified' with a 'Download Cert' button. The second row for 'Heavy Machinery Handling' shows 'Expiring in 5 Days' and 'Due Soon' with a 'Start E-Learning' button. The third row for 'Hygiene Standard v2.0' shows 'Expired Yesterday' and 'Non-Compliant' with a 'Retake Test' button.

Module Name	Validity/Expiry	Status	Action
Fire Safety Protocols	Valid until Oct 2026	Certified	[Download Cert]
Heavy Machinery Handling	Expiring in 5 Days	Due Soon	[Start E-Learning]
Hygiene Standard v2.0	Expired Yesterday	Non-Compliant	[Retake Test]

Figure 6.3: Overview of Engineer “My Training” Page

Description: shows the competency profile of the logged-in user, including assigned modules with "Certified," "Expiring," or "Non-Compliant" status indicators.

Purpose: to assist staff members in keeping an eye on their own credentials and acting before certifications expire.

The screenshot displays the 'Log New Complaint' interface within the QMS-FLOW application. The left sidebar features a dark blue background with a network-like pattern and includes links for Dashboard, Log Complaint (which is highlighted in blue), SOP Library, My Training, and Settings. The main content area shows the 'Log New Complaint' form. At the top, a breadcrumb navigation indicates: Dashboard > Customer Complaints > New Record. The form is divided into three steps: 1. Identification (selected), 2. Issue Details, and 3. Evidence Upload. The 'Identification' step contains fields for Complaint Source (set to 'Customer Email'), Product Batch Number (containing several asterisks and a green checkmark), Date of Occurrence (with a calendar icon), and Initial Description (with placeholder text 'Briefly brikef description of complaint.'). At the bottom right of the form are 'Cancel' and 'Next Step' buttons.

Figure 6.4: Overview of Engineer “Log Complaint” Page

Description: The complaint source, batch number, date, and a description of the problems are all included in this recommended form for reporting non-conformances.

Purpose: to ensure that all relevant information is recorded for research by standardizing data collection.

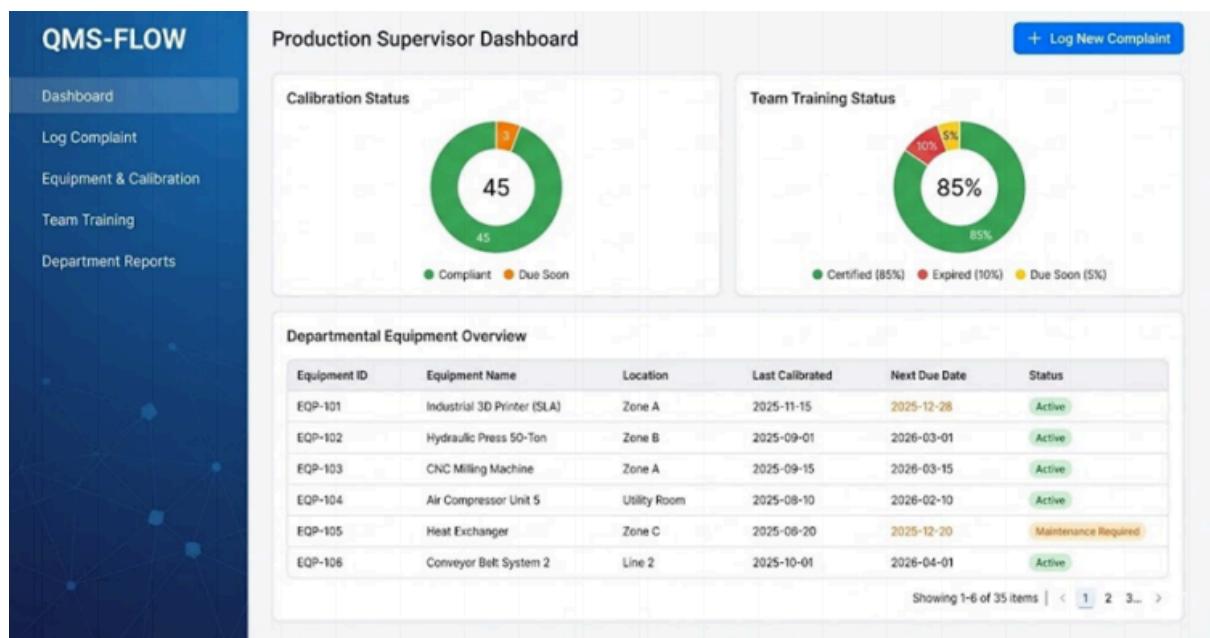


Figure 6.5: Overview of Manager Dashboard

Description: A list of equipment with "Due Soon" notifications comes after a management view that displays departmental data via charts for training and calibration.

Purpose: to give supervisors a high-level summary on the state of the department and impending deadlines.

Date	Meeting Type	Topic	Attendees	Action Items	Minutes File
2025-12-28	Shift Handover	Night Shift Target Handoff	12 Staff	Pending (2)	
2025-12-20	Toolbox Talk	Machine Guarding Safety	20 Staff	Closed	
2025-12-15	Quality Briefing	Defect Reduction Plan	8 Staff	In Progress	
2025-12-10	Shift Handover	Day Shift Handover Notes	15 Staff	Closed	
2025-12-05	Safety Talk	PPE Compliance Review	18 Staff	Closed	
2025-11-28	Quality Briefing	New SOP Introduction	22 Staff	Pending (1)	
2025-11-20	Shift Handover	Weekend Production Plan	10 Staff	Closed	

Figure 6.6: Overview of Manager “Department Reports” Page

Description: A digital diary that keeps track of internal communications like "Shift Handover" and "Safety Talks," along with meeting minutes and attendance statistics.

Purpose: to maintain continuity across shifts and produce a traceable record of operational briefings.

All Assets	Due < 30 Days	Out of Service	Calibration Overdue			
Asset ID	Equipment Name	Serial No.	Last Calibrated	Next Due	Status	Certificate
EQP-101	Industrial 3D Printer	SN-9982	2025-01-15	2026-01-15	Valid	
EQP-105	Heat Exchanger Unit	SN-4421	2024-12-20	2025-12-20	Expiring	
EQP-109	Conveyor Motor B	SN-1102	2024-06-01	2025-06-01	Non-Compliant	
EQP-112	CNC Lathe Machine X1	SN-8875	2025-01-20	2026-01-20	Valid	
EQP-118	Digital Caliper Set	SN-3321	2024-07-15	2025-07-15	Valid	
EQP-121	Air Compressor 500L	SN-1144	2024-11-30	2025-05-30	Valid	
EQP-125	Hydraulic Press 20T	SN-5582	2023-12-01	2024-12-01	Non-Compliant	
EQP-130	Oven (High Temp)	SN-9902	2025-02-01	2026-02-01	Valid	
EQP-134	Microscope (Optical)	SN-6678	2024-12-28	2025-12-28	Expiring	
EQP-141	Tensile Tester	SN-7721	2025-01-10	2026-01-10	Valid	

Showing 1-10 of 250 items | < 1 2 3 ... >

Figure 6.7: Overview of Manager “Equipment & Calibration” Page

Description: a thorough inventory of all departmental assets, with calibration dates and status badges such as "Valid," "Expiring," or "Non-Compliant" displayed.

Purpose: to keep track of maintenance schedules in order to avoid using uncalibrated equipment.

Staff ID	Name	Role	Assigned Training	Completion Date	Expiry Date	Status
OP-001	Ali Ahmad	Operator	Safety Lvl 1	2025-10-10	2026-10-10	Certified
OP-004	Sarah Lee	Packer	Hygiene Protocols	2023-05-20	2025-05-20	Expired Action Required
OP-005	Raj Kumar	Technician	Machine Maintenance	2025-11-01	2026-11-01	Certified
OP-008	Mei Ling	Quality Inspector	QMS Auditing Basics	2024-12-15	2026-12-15	Certified
OP-012	David Chen	Forklift Driver	Advanced Forklift Safety	2024-08-01	2025-08-01	Expired Action Required
OP-015	Fatima Hassan	Assembly Line	ESD Control	2025-09-30	2026-09-30	Certified
OP-019	Kenji Tanaka	Supervisor	Leadership in Safety	2023-11-20	2025-11-20	Expired Action Required
OP-022	Elena Petrova	Operator	Chemical Handling	2025-01-10	2026-01-10	Certified

Showing 1-8 of 20 items | < 1 2 3 >

Figure 6.8: Overview of Manager “Team Training Matrix” Page

Description: Each team member's training status is listed in a matrix, with expired certifications highlighted in red for prompt attention.

Purpose: to determine skill gaps and guarantee the workforce's continued competence and compliance.



Figure 6.9: Overview of Admin / Manager Dashboard

Description: Important KPIs like "Open CAPAs" and "Compliance Score," as well as a list of urgent approval activities, are displayed in the system command center.

Purpose: to prioritize high-risk items that need management action and to offer strategic monitoring.

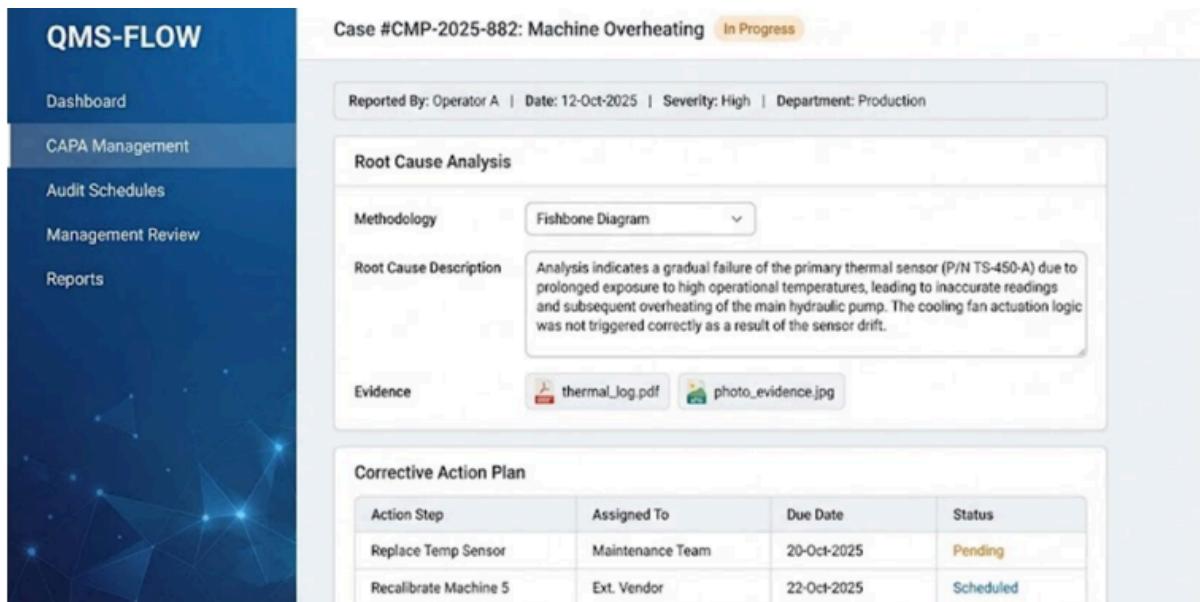


Figure 6.10: Overview of Admin / Manager “CAPA Management” Page

Description: An investigative management workflow screen containing a "Corrective Action Plan" with tasks and a "Root Cause Analysis" section.

Purpose: to record the examination and fixing of non-conformances in accordance with ISO guidelines.

The screenshot shows the QMS-FLOW interface for management review meetings. On the left, a sidebar menu includes 'Dashboard', 'CAPA Management', 'Audit Schedules', 'Management Review' (which is highlighted in blue), and 'Reports'. The main content area is titled 'Management Review Meetings' and features a button '+ Schedule New Meeting'. Below this, a section for the 'Next Scheduled Review: Q4 2025' displays the date (15-Dec-2025), organizer (Quality Manager), and status (Agenda Finalized). An 'Agenda Preview' lists four items: 1. Review of Quality Policy & Objectives ... 2. Customer Satisfaction Survey Results ... 3. Audit Findings Summary ... 4. Resource Allocation Review ... With buttons 'Edit Agenda' and 'View Minutes'. A 'Past Meeting History' table lists five completed reviews from 2024 to 2025, each with a 'View Minutes' button.

Date	Meeting Name	Status	Actions
15-Sep-2025	Q3 2025 Review	Completed	View Minutes
15-Jun-2025	Q2 2025 Review	Completed	View Minutes
15-Mar-2025	Q1 2025 Review	Completed	View Minutes
15-Dec-2024	Q4 2024 Review	Completed	View Minutes
15-Sep-2024	Q3 2024 Review	Completed	View Minutes

Figure 6.11: Overview of Admin / Manager “Management Review” Page

Description: An executive review scheduling interface that shows the agenda for the upcoming meeting and a list of previous reviews that have been finished.

Purpose: to lead sessions for strategic planning and ongoing improvement.



Figure 6.12: Overview of Admin / Manager “General Reports & SOPs” Page

Description: a unified view with the "SOP Library" with version control at the bottom and operational analytics charts at the top.

Purpose: to track versions of system documentation and examine trends in quality.

The screenshot shows a split-screen interface for 'System Records & Analytics'. On the left, a sidebar menu lists 'QMS-FLOW' with options: Dashboard, CAPA Management, Audit Schedules, Management Review, and Reports (which is highlighted). Below the menu is a decorative background graphic of a network of blue dots connected by lines.

The main content area has two tables:

- Equipment Calibration Status** (Left Table):

ID	Name	Last Calibrated	Next Due	Status
EQP-101	3D Printer	2024-12-28	2025-12-28	Active
EQP-105	Heat Exchanger Unit	2024-12-20	2025-12-20	Active
EQP-109	Conveyor B	2023-06-01	2024-06-01	Overdue ⚠
EQP-112	CNC Lathe Machine X1	2025-01-20	2026-01-20	Active
EQP-118	Digital Caliper Set	2024-07-15	2025-07-15	Active
EQP-121	Air Compressor 500L	2024-05-30	2025-05-30	Active
EQP-125	Hydraulic Press 20T	2023-12-01	2024-12-01	Overdue ⚠
- Staff Training Matrix** (Right Table):

Staff Name	Role	Certification	Expiry Date	Status
Ali Ahmad	Operator	Safety Lvl 1	2026-10-10	Valid
Sarah Lee	Packer	Hygiene Protocols	2025-05-20	Expired ⚠
Raj Kumar	Technician	Machine Maintenance	2026-11-01	Valid
Mei Ling	Quality Inspector	QMS Auditing Basics	2026-12-15	Valid
David Chen	Forklift Driver	Advanced Forklift Safety	2025-08-01	Expired ⚠
Fatima Hassan	Assembly Line	ESD Control	2026-09-30	Valid
Kenji Tanaka	Supervisor	Leadership in Safety	2025-11-20	Expired ⚠

Figure 6.13: Overview of Admin / Manager “Resource Compliance” Page

Description: A split-screen display that simultaneously tracks "Staff Training" and "Equipment Calibration" to identify compliance issues throughout the factory

Purpose: to check compliance of people's training and equipment calibration from a single perspective.

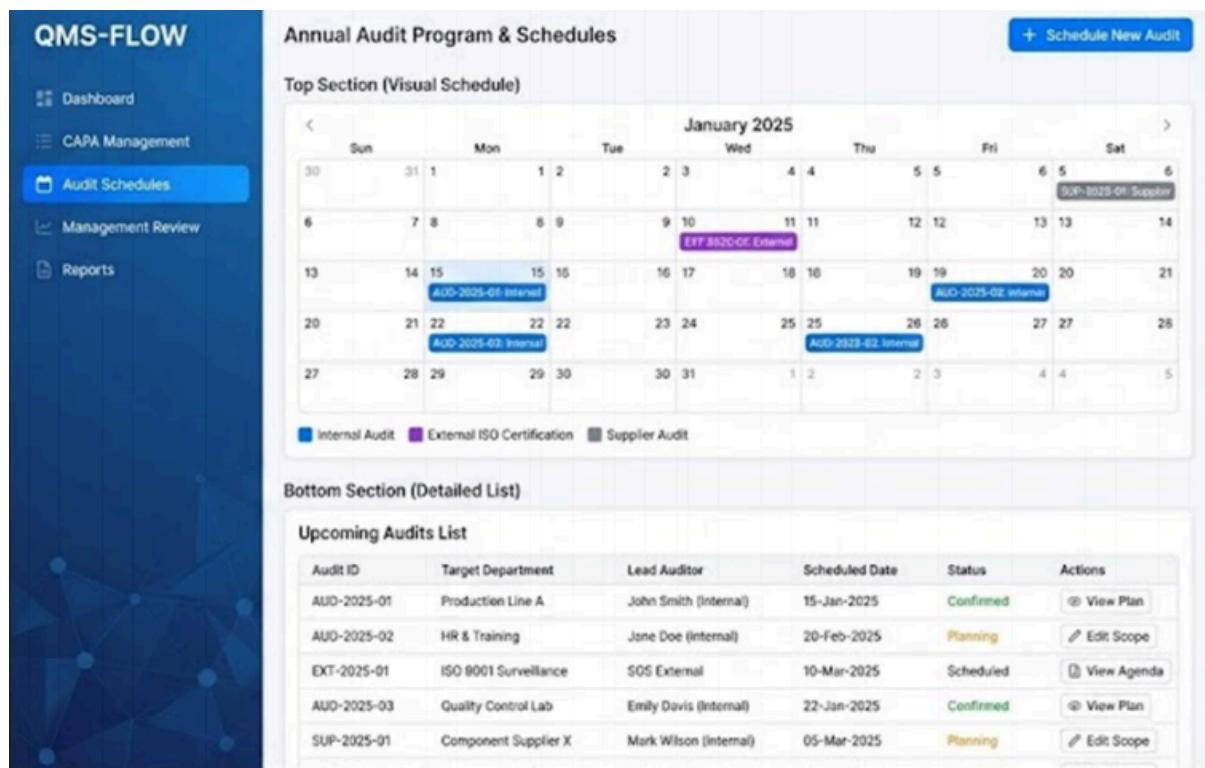


Figure 6.14: Overview of Admin / Manager “Audit Schedules” Page

Description: a planning screen with a comprehensive list of upcoming internal and external audits as well as a visual calendar for the annual audit program.

Purpose: to guarantee that the audit program is organized and carried out on time.

The screenshot shows the QMS-FLOW Auditor Dashboard. On the left, there is a sidebar with the title "QMS-FLOW" and a blue header bar containing the word "Dashboard". Below the sidebar, there are several menu items: "Complaint History (View)", "CAPA Records (View)", "Training Matrices (View)", and "Audit Trails". The main content area has a yellow banner at the top stating "VIEW ONLY MODE: You have read-only access for verification purposes." Below this, there is a section titled "Recent System Activity Log for Verification" which contains a table of recent actions. The table has columns for "Timestamp", "User Role", "Action Taken", "Record ID", and "Action". The data in the table is as follows:

Timestamp	User Role	Action Taken	Record ID	Action
2025-10-27 10:15 AM	Supervisor	Updated Calibration Record	EQP-101	View Details
2025-10-27 09:45 AM	Quality Manager	Approved CAPA Plan	CAPA-2025-001	View Details
2025-10-26 04:30 PM	Production Operator	Submitted New Complaint	CMP-00492	View Details
2025-10-26 03:15 PM	Training Coordinator	Added New SOP to Library	SOP-105	View Details

Figure 6.15: Overview of Auditor Dashboard

Description: a limited view with a "Read-Only" banner and completed with necessary sidebar for easy access to view necessary document and modifications and shows quick access to recent system activity log

Purpose: to provide easy access and history of recent system activity log for verification

The screenshot shows the QMS-FLOW application interface. On the left, there is a sidebar with various menu items: Dashboard, Complaint History (View), CAPA Records (View), Training Matrices (View), and Audit Trails. The 'Audit Trails' item is currently selected. The main content area is titled 'System Audit Trail & Change Log' and includes a 'READ-ONLY MODE: EXTERNAL AUDITOR' banner. Below the banner are several filter options: 'Filter by User' (dropdown menu showing 'All Users', 'Supervisor_A', 'Admin_Main', 'Admin_Main', 'Operator_Z'), 'Filter by Date Range' (with 'Start Date' and 'End Date' fields), 'Filter by Record ID' (with 'Enter ID...' field), and 'Event Type (Create/Edit/Delete)' (dropdown menu). A large table below these filters displays a list of audit events. The table columns are: Timestamp, User ID, Action, Record Affected, Field Changed, Old Value, and New Value. The data in the table is as follows:

Timestamp	User ID	Action	Record Affected	Field Changed	Old Value	New Value
2025-12-28 14:30:05	Supervisor_A	Update	EQP-105	Next Due Date	2025-12-20	2026-12-20
2025-12-28 10:15:22	Admin_Main	Approval	SOP-SAFE-101	Status	Pending	Published
2025-12-27 09:05:00	Operator_Z	Login Failed	System Access	-	-	Invalid Password
2025-12-27 08:45:10	Admin_Main	Create	USR-205	User Account	-	New User Created
2025-12-26 16:20:35	Quality_Mgr	Delete	DOC-OLD-99	Document	Archived	Document Deleted
2025-12-26 15:10:05	Supervisor_B	Update	TRN-REC-55	Completion Date	2025-12-25	2025-12-26
2025-12-26 11:05:44	System_Auto	System Event	Backup	Daily Backup	Pending	Completed
2025-12-26 09:30:21	Auditor_Ext	View	CAPA-2025-08	CAPA Record	-	Record Viewed
2025-12-25 14:00:00	Admin_Main	Update	SYS-CONFIG	Timeout Setting	30 min	60 min
2025-12-25 10:00:00	Operator_Y	Login Success	System Access	-	-	Session Started

Showing 1-10 of 500+ entries | < 1 2 3 4 5 ... >

Figure 6.16: Overview of Auditor “Audit Trails” Page

Description: a limited view with a "Read-Only" banner and a complete "System Audit Trail" log that documents each user action and data modification.

Purpose: to show data integrity and make it possible for external auditors to confirm system history without making any modifications.

QMS-FLOW

Archived Record - Modification Disabled

CAPA Record #CMP-882: Machine Overheating

CLOSED

Root Cause Analysis

Fishbone Diagram Analysis

Analysis identified primary root cause as failure of thermal sensor (P/N TS-450-A) due to prolonged exposure to high operational temperatures, leading to inaccurate readings and delayed cooling system activation. Contributory factor included outdated preventive maintenance schedule for sensor replacement.

Corrective & Preventive Action Plan

Action ID	Action Description	Assigned To	Due Date	Completion Date	Status	Action
ACT-001	Replace Thermal Sensor (TS-450-A) on Machine 5	Maintenance Team	2025-10-20	2025-10-18	Completed	View Log
ACT-002	Update Preventive Maintenance Schedule for Sensors	Engineering Dept	2025-10-25	2025-10-24	Completed	View Log
ACT-003	Conduct Training on New Maintenance Procedures	Training Coordinator	2025-11-01	2025-10-30	Completed	View Log
ACT-004	Verify Effectiveness of Corrective Actions	Quality Assurance	2025-11-15	2025-11-14	Effective	View Log

Auditor Verification

I have verified this record conforms to ISO 9001 standards.

Mark as Verified

Figure 6.17: Overview of Auditor “CAPA Records” Page

Description: An "Auditor Verification" checkbox is located at the bottom of a locked view of a closed record with alteration disabled.

Purpose: To enable auditors to safely examine and confirm old data.

7 Requirements Matrix

Below is the Traceability Matrix, which is utilized as a traceability tool for the QMS FLOW System as a whole. The traceability matrix links the high-level subsystems in the QMS FLOW System (P001 to P005), as well as their use cases (UC001 to UC013), to particular elements of design, such as Sequence Diagrams and Entities. All functional requirements must, therefore, be backed by corresponding elements within these traceability matrices, thus validating the system's consistency and completeness in terms of design.

The sequence diagrams for each use case vs. corresponding classes (entities) are presented in Table 7.1.

Table 7.1: Requirements Traceability Matrix (Use Cases vs. Entities)

Persona, Use Case, Sequence Diagram	Do cu me nt	Ap pr ov al / Ch an ge	Au dit / Fin din g	C A P A	Tr ai ni ng	Equ ipm ent	Risk / Incid ent	KPI	Syst em Log	User / Ac ce ss
P002, UC001, SD001: Register Document	X								X	X

P002, UC002, SD002: Approve Document	X	X							X	X
P001, UC003, SD003: Schedule Audit			X						X	X
P001, UC004, SD004: Record Audit Findings			X						X	X
P004, UC005, SD005: Initiate CAPA			X	X			X		X	X
P004, UC006, SD006: Manage Competency				X	X				X	X

P004, UC007, SD007: Manage Equipment						X			X	X
P002, UC008, SD008: Monitor KPIs			X	X	X	X		X	X	
P002, UC009, SD009: Manage Change	X	X							X	X
P003, UC010, SD010: Provide System Support									X	X
P003, UC011, SD011: Maintain System Logs									X	X

P005, UC012, SD012: Manage Risks				X			X		X	X
P005, UC013, SD013: Manage Incident & Feedback				X			X		X	X

8 Test Cases

8.1 TC001: Test <Document & Change Management with KPI Dashboard Module> Subsystem 2:

<Approve Document (UC02)>

- (a) TC001_01: Test <Approve Document (SD02 - NF)>
- (b) TC001_02: Test <Approve Document (SD02 - AF)>
- (c) TC001_03: Test <Approve Document (SD02 - EF)>

8.1.1 TC001_01: Test <Approve Document (SD02 - NF)>

Test Case ID	TC003_01	Test Case Description	Verify that a Manager can successfully approve a pending document.	
Created By	Hazim	Reviewed By	Rasyid	Version
QA Tester's Log	Initial test of approval workflow. All managerial permissions verified and functioning correctly.			
Tester's Name			Date Tested	Test Case (Pass/Fail/Not)
S#	Prerequisites:		S#	Test Data
1	User logged in as Manager		1	Username: rasyid_mgr
2	Document status is "Pending Approval".		2	Password: P@ssword123
			3	Doc_ID: SOP-2026-005
			4	Action: Approve
Test Scenario	Verify that a manager can successfully approve a document.			
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Manager logs in and navigates to "Review Queue"	List of pending documents is displayed		
2	Select a document and click "Approve"	System prompts for confirmation		
3	Confirm approval	Status updates to "Approved" and version is locked		

8.1.2 TC001_02: Test <Approve Document (SD02 - AF)>

Test Case ID	TC003_02	Test Case Description	Verify behavior when a document is rejected	
Created By	Hazim	Reviewed By	Rasyid	Version
QA Tester's Log	Verified that rejection comments are mandatory. System successfully routed the document back to the author.			
Tester's Name		Date Tested		Test Case (Pass/Fail/Not)
S#	Prerequisites:		S#	Test Data
1	document is open for review		1	Action: Reject
			2	Comments: "Missing section 4.2 regarding safety protocols."
Test Scenario	Verify behavior when a document is rejected			
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Manager selects a document and clicks "Reject"	System prompts for rejection comments		
2	Enter comments and confirm	Status returns to "Draft", comments sent to owner		

8.1.3 TC001_03: Test <Approve Document (SD02 - EF)>

Test Case ID	TC003_03	Test Case Description	Verify behavior when approval rights are missing		
Created By	Hazim	Reviewed By	Rasyid	Version	1.2
QA Tester's Log	Tested security boundary. System correctly restricted the 'Approve' button from the Engineer UI.				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	system is displaying the Document List.		1	Username: hazim_eng	
			2	Password: P@ssword123	
			3	Target: Approval_Button	
Test Scenario	Verify on entering valid userid and password, the customer can login				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	User without manager role attempts to approve	System denies access and displays error			

8.2 TC002: Test <Document & Change Management with KPI Dashboard

Module> Subsystem 2: <Register Document (UC01)>

- (a) TC003_01: Test <Register Document (SD01 - NF)>
- (b) TC003_02: Test <Register Document (SD01 - AF)>
- (c) TC003_03: Test <Register Document (SD01 - EF)>

8.2.1 TC002_01: Test <Register Document (SD01 - NF)>

Test Case ID	TC003_01	Test Case Description	Test the Normal Flow of Document Registration		
Created By	Hazim	Reviewed By	Rasyid	Version	1
QA Tester's Log	Standard upload test. Verified that PDF metadata is correctly mapped to the database schema.				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Engineer is logged in with document creation rights		1	File: Safety_Manual_V1.pdf	
			2	Title: "Standard Safety Ops"	
			3	Owner: Engineering Dept	
Test Scenario	Verify that an engineer can upload a new SOP successfully.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Select "Register Document" and upload SOP file	File is uploaded to the repository			
2	Enter metadata (Title, Version, Owner)	Metadata is saved and versioned			
3	Submit for review	Status becomes "Pending Approval"			

8.2.2 TC002_02: Test <Register Document (SD01 - AF)>

Test Case ID	TC003_02	Test Case Description	Verify behavior when a duplicate ID is detected		
Created By	Hazim	Reviewed By	Rasyid	Version	1.1
QA Tester's Log	Duplicate ID test. System successfully flagged existing ID and prevented database conflict				
Tester's Name		Date Tested		Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	system database already contains a document with a specific unique ID		1	Doc_ID	
Test Scenario	Verify that an engineer can upload a new SOP successfully.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Enter metadata with an existing Document ID	System alerts user of duplicate			

8.2.3 TC002_03: Test <Register Document (SD01 - EF)>

Test Case ID	TC003_03	Test Case Description	Verify behavior when the upload fails		
Created By	Hazim	Reviewed By	Rasyid	Version	1.2
QA Tester's Log	Input validation test. Verified that executable and script files are blocked by the security filter.				
Tester's Name		Date Tested		Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Engineer is logged in with document creation rights		1	File: System_Update.exe	
			2	File: Photo_Draft.png	
Test Scenario	Verify that an engineer can upload a new SOP successfully.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Attempt upload during a network failure	System prompts a retry for the upload			

8.3 TC08: Test <Document & Change Management with KPI Dashboard

Module> Subsystem 2: <Monitor KPIs (UC08)>

- (a) 8.3.1 TC008_01: Test <Monitor KPIs (SD08 - NF)>
- (b) 8.3.2 TC008_02: Test <Monitor KPIs (SD08 - AF)>
- (c) 8.3.3 TC008_03: Test <Monitor KPIs (SD08 - EF)>

8.3.1 TC008_01: Test <Monitor KPIs (SD08 - NF)>

Test Case ID	TC003_01	Test Case Description	Test the Normal Flow of KPI Monitoring		
Created By	Hazim	Reviewed By	Rasyid	Version	1
QA Tester's Log		Dashboard rendering test. Real-time data from Audit and CAPA modules matched the chart output			
Tester's Name		Date Tested			Test Case (Pass/Fail/Not)
S#	Prerequisites:		S#	Test Data	
1	User is logged in as Manager or Auditor; the system has existing data logs for CAPA, Training, and Audits from the last 30 days.		1	Date Range: 01/01/2026 to 15/01/2026	
			2	Metric Type: "CAPA Closure Rate"	
Test Scenario Verify that a manager can view the KPI dashboard.					
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Manager opens the KPI dashboard	Real-time charts/reports are displayed			
2	Review metrics against targets)	Metrics are calculated accurately			

8.3.2 TC008_02: Test <Monitor KPIs (SD08 - AF)>

Test Case ID	TC003_02	Test Case Description	Verify behavior when data is missing/incomplete		
Created By	Hazim	Reviewed By	Rasyid	Version	1.1
QA Tester's Log		Null data handling test. Dashboard successfully displayed 'Empty Set' graphic for departments with no logs.			
Tester's Name		Date Tested			Test Case (Pass/Fail/Not)
S#	Prerequisites:		S#	Test Data	
1	User is logged in; some departments (e.g., "Janitorial") have not yet uploaded any monthly quality logs.		1	Department: "Janitorial"	
Test Scenario Verify behavior when data is missing/incomplete					
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Access dashboard with incomplete training logs	System raises flags for missing data			

8.3.3 TC008_03: Test <Monitor KPIs (SD08 - EF)>

Test Case ID	TC003_03	Test Case Description	Verify CAPA trigger when KPI is below target		
Created By	Hazim	Reviewed By	Rasyid	Version	1.2
QA Tester's Log		Database timeout simulation. Verified that the system shows a user-friendly error instead of a crash.			
Tester's Name		Date Tested			Test Case (Pass/Fail/Not)
S#	Prerequisites:		S#	Test Data	
1	User is logged in; the connection to the SQL database is currently interrupted or experiencing high latency.		1	Year: 2099	
			2	Search Query: "I@#\$%^&*()" (Invalid characters to test boundary)	
Test Scenario Verify CAPA trigger when KPI is below target					
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	KPI (e.g., CAPA closure) falls below target	System triggers UC05: Initiate CAPA			

8.4 TC09: Test <Document & Change Management with KPI Dashboard Module> Subsystem 2: <Manage Change (UC09)>

- (a) 8.4.1 TC008_01: Test <Manage change (SD09 - NF)>
- (b) 8.4.2 TC008_02: Test <Manage change (SD09 - AF)>
- (c) 8.4.3 TC008_03: Test <Manage change (SD09 - EF)>

8.4.1 TC008_01: Test <Manage change (SD09 - NF)>

Test Case ID	TC003_01	Test Case Description	Test the Normal Flow of Change Management	
Created By	Hazim	Reviewed By	Rasyid	Version
QA Tester's Log	Full lifecycle test of Change Request. Automated notifications were sent to the review committee.			
Tester's Name			Date Tested	Test Case (Pass/Fail/Not)
S#	Prerequisites:		S#	Test Data
1	User logged in as Admin		1	Change_Type: "Machine Update"
2	Change Impact Assessment completed		2	Priority: "High"
Test Scenario	Verify CR is successfully added to the "Change Log" with a "Pending" status.			
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Admin selects "Manage Change"	Change details and impact are prompted		
2	Manager reviews and approves the proposal	Request is routed to UC02 for final approval		
3	System updates change logs	Change log is updated and traceable		

8.4.2 TC008_02: Test <Manage change (SD09 - AF)>

Test Case ID	TC003_02	Test Case Description	Verify behavior when a change is rejected	
Created By	Hazim	Reviewed By	Rasyid	Version
QA Tester's Log	User cancellation test. Verified that the 'Withdraw' action removes the request from the pending queue.			
Tester's Name			Date Tested	Test Case (Pass/Fail/Not)
S#	Prerequisites:		S#	Test Data
1	CR status is "Pending Review"		1	Action: Withdraw Request
2	Original requester is logged in.			
Test Scenario	Verify the status changes to "Cancelled" and disappears from the manager's queue.			
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Manager rejects the change request	Request is archived with comments		

8.4.3 TC008_03: Test <Manage change (SD09 - EF)>

Test Case ID	TC003_03	Test Case Description	Verify behavior when approval is delayed	
Created By	Hazim	Reviewed By	Rasyid	Version
QA Tester's Log	Data integrity test. Form submission was blocked until all required impact assessment fields were filled.			
Tester's Name	Date Tested		Test Case (Pass/Fail/Not)	
S#	Prerequisites:	S#	Test Data	
1	Admin is on the "New Change Request" screen	1	Impact_Description: [Leave Blank]	
		2	Action: Click Submit	
Test Scenario	Verify that the system highlights the field in red and prevents submission.			
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Approval remains pending past deadline	Escalation is triggered by the system		

8.5 TC003: Audit Management Module

Subsystem 1: Schedule Audit (UC03)

This test contains the following test cases:

1. TC003_01: Test <Schedule Audit> (SD003 - NF)
2. TC003_02: Test <Schedule Audit> (SD003 - AF1)
3. TC003_03: Test <Schedule Audit> (SD003 - AF2)
4. TC003_04: Test <Schedule Audit> (SD003 - AF3)

8.5.1 TC003_01: Test <Schedule Audit> (SD003 - NF)

Test Case ID	TC003_01	Test Case Description	Scheduling an Audit	
Created By	Rasyid	Reviewed By	Hazim	Version
QA Tester's Log	Verify notification timestamp, check audit entry in system logs.			
Tester's Name	Afiq Shahir	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)
S#	Prerequisites:	S#	Test Data	
1	Manager or Auditor logged in	1	Audit date = 20/01/2026	
2	Audit scheduling rights enabled	2	Scope = "Testing Dept"	
		3	Auditor = "Ali"	
Test Scenario	Verify that a manager/auditor can schedule an audit with valid inputs.			
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Login as Manager or Auditor	Access granted		
2	Navigate to "Schedule Audit" page	Schedule audit page opens		
3	Enter audit date, scope and department	Data can be inserted based on requirement		
4	Click "Submit"	Audit record saved, confirm message display and notification sent to assign auditor		

8.5.2 TC003_02: Test <Schedule Audit> (SD003 - AF1)

Test Case ID	TC003_02	Test Case Description	Invalid Audit Date		
Created By	Rasyid	Reviewed By	Afiq Shahir	Version	1
QA Tester's Log	Confirm error message is user-friendly and logged in system traceability.				
Tester's Name	Hazim	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Manager or Auditor logged in		1	Audit date = 01/01/2026	
2	Audit scheduling rights enabled				
Test Scenario	Verify that the system rejects scheduling audits with past dates				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Enter past date	Error message display, audit not scheduled			

8.5.3 TC003_03: Test <Schedule Audit> (SD003 - AF2)

Test Case ID	TC003_3	Test Case Description	Audit Reminder Notifications		
Created By	Rasyid	Reviewed By	Afiq Irfan	Version	1
QA Tester's Log	Confirm notification delivery method (email/system alert)				
Tester's Name	Afiq Shahir	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Audit scheduled for next day		1		
Test Scenario	Verify that the system sends reminder notifications before scheduled audits				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Schedule audit	Audit record saved			
2	Wait for reminder notification	Reminder sent to assigned auditor 24 hours before audit			

8.5.4 TC003_04: Test <Schedule Audit> (SD003 - AF3)

Test Case ID	TC003_4	Test Case Description	Access Control Validation		
Created By	Rasyid	Reviewed By	Rafiq	Version	1
QA Tester's Log	Verify RBAC enforcement				
Tester's Name	Hazim	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Engineer logged in (no audit scheduling rights)		1		
Test Scenario	Verify that the system sends reminder notifications before scheduled audits				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Attempt to schedule audit	Access denied, message "You do not have permission to schedule audits"			

8.6 TC004: Audit Management Module

Subsystem 2: Record Audit Findings (UC04)

This test contains the following test cases:

1. TC004_01: Test <Record Audit Findings> (SD004 - NF)
2. TC004_02: Test <Record Audit Findings> (SD004 - AF1)
3. TC004_03: Test <Record Audit Findings> (SD004 - AF2)

8.6.1 TC004_01: Test <Record Audit Findings> (SD004 - NF)

Test Case ID	TC004_01	Test Case Description	Record Audit Findings		
Created By	Rasyid	Reviewed By	Rafiq	Version	1
QA Tester's Log	Verify linkage to CAPA module, check audit trail immutability				
Tester's Name	Afiq Irfan	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Auditor logged in		1	Audit ID = AUD2026_01	
2	Scheduled audit exists		2	Finding = "Non-conformance: Calibration overdue"	
Test Scenario	Verify that an Auditor can record findings during a scheduled audit				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Open scheduled audit record	Audit records display			
3	Enter findings	Findings found			
4	Click "Save"	Findings stored, linked to audit record, visible in audit trail			

8.6.2 TC004_02: Test <Record Audit Findings> (SD004 - AF1)

Test Case ID	TC004_2	Test Case Description	Missing Evidence		
Created By	Rasyid	Reviewed By	Afiq Shahir	Version	1
QA Tester's Log	Confirm mandatory evidence enforcement				
Tester's Name	Afiq Irfan	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Auditor logged in		1	Audit ID = AUD2026_02	
2	Scheduled audit exists		2	Finding entered without evidence attachment	
Test Scenario	Verify that the system enforces mandatory evidence attachment for non-conformance findings				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Enter finding text only.	Error display			

8.6.3 TC004_03: Test <Record Audit Findings> (SD004 - AF2)

Test Case ID	TC004_3	Test Case Description	CAPA Trigger from Audit Findings		
Created By	Rasyid	Reviewed By	Afiq Shahir	Version	1
QA Tester's Log	Verify CAPA linkage traceability in requirements matrix				
Tester's Name	Afiq Irfan	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Auditor logged in		1	Audit ID = AUD2026_03	
2	finding marked "non-conformance"		2	Finding = "Training record missing"	
Test Scenario	Verify that non-conformance findings can initiate CAPA creation				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Record finding	Audit record found			
2	Select "Initiate CAPA"	CAPA record created, linked to audit finding, CAPA ID generated			

8.7 TC005:Test <CAPA Management Module> Subsystem 4: <InitiateCAPA (UC05)>

- (a) TC005_01: Test <InitiateCAPA (SD05 - NF)>
- (b) TC005_02: Test <InitiateCAPA (SD05 - AF)>
- (c) TC005_03: Test <InitiateCAPA (SD05 - EF)>

8.7.1 TC005_03: Test <InitiateCAPA (SD05 - NF)>

Test Case ID	TC005_01	Test Case Description	Verify CAPA creation from a Failed Audit Log		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.1
QA Tester's Log	Normal Flow				
Tester's Name	Rasyid	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:				
1	An "Audit Finding" exists in the system.				
2	User has "Lead Auditor" or "Manager" privileges.				
3					
4					
S#	Test Data				
1	Source Issue: Audit #AUD-2026-001				
2	Root Cause: "Calibration error on Machine B"				
3	Action Owner: Engineering Manager				
4	Deadline: [Today + 14 Days]				
Test Scenario	Verify that linking an Audit ID and entering Root Cause generates an Open Plan.				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the Audit Log and select Audit #AUD-2026-001.	System calls InitiateCAPAUI and auto-fills the Source Issue field.			
2	Enter the Root Cause analysis details.	Text field accepts the input.			
3	Select Action Owner and set the Deadline.	Fields are populated correctly.			
4	Click "Create Corrective Plan".	System generates a new CAPA ID.			
5		Output: Status set to "Open Corrective Action Plan".			
6		Original Audit record is linked to this CAPA.			

8.7.2 TC005_03: Test <InitiateCAPA (SD05 - EF)>

Test Case ID	TC005_02	Test Case Description	Verify CAPA creation from a KPI Performance Drop.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.2
QA Tester's Log	Alternate Flow				
Tester's Name	Rafiq	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:				
1	A KPI Dashboard shows a metric below the accepted threshold.				
2					
3					
4					
S#	Test Data				
1	Source Issue: KPI "Production Yield" (< 90%)				
2	Root Cause: "Raw material quality variance"				
3	Action Owner: Procurement Manager				
4					
Test Scenario	Verify that selecting a failed KPI triggers the correct CAPA workflow.				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	On KPI Dashboard, click the "Alert/Action" button on the failed KPI.	InitiateCAPAUI screen opens with Source Type set to "KPI".			
2	Input Root Cause and assign Action Owner.	System accepts the KPI-specific context.			
3	Submit the form.	Output: "Open Corrective Action Plan" is created.			
4		Notification sent to Procurement Manager.			
5	Verify KPI Dashboard status.	The failed KPI now shows a "CAPA In Progress" tag.			

8.7.3 TC005_03: Test <InitiateCAPA (SD05 - EF)>

Test Case ID	TC005_03	Test Case Description	Verify validation error when Root Cause is missing.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.3
QA Tester's Log	Exception Flow				
Tester's Name	Hazim	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Trigger received from any source.		1	Source Issue: Incident #INC-055	
2			2	Root Cause: [Left Empty]	
3			3	Action Owner: [Selected]	
4			4		
Test Scenario	Verify that the system prevents moving to "Open Corrective Action Plan" if mandatory analysis is missing.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Open the CAPA initiation form for Incident #INC-055.	Form loads correctly.			
2	Leave Root Cause blank. Select an Owner.	Owner is selected, but Root Cause is empty.			
3	Click "Create/Submit".	System fails to generate the plan.			
4		Error Message: "Root Cause Analysis is required."			
5	Verify Database.	No new "Open Corrective Action Plan" record is created in the database.			

8.8 TC006: Test <CAPA Management Module> Subsystem 4: <Manage Competency & Training (UC06)>

- (a) TC006_01: Test <Manage Competency & Training (SD06 - NF)>
- (b) TC006_02: Test <Manage Competency & Training (SD06 - AF)>
- (c) TC006_03: Test <Manage Competency & Training (SD06 - EF)>

8.8.1 TC006_03: Test <Manage Competency & Training (SD06 - NF)>

Test Case ID	TC006_01	Test Case Description	Verify Staff Competency is updated to "Competent" upon passing training.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.1
QA Tester's Log	Normal Flow				
Tester's Name	Rasyid	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Manager logged in.		1	Staff ID: STF-101 (Rasyid)	
2	Training Gap identified for Staff.		2	Course: ISO 9001 Basics	
3			3	Score Input: 85/100 (Pass threshold is 70)	
4			4		
Test Scenario	Verify that entering a passing score updates the log to "Competent".				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Navigate to Training UI and select Staff STF-101.	System shows "Training Gap" identified status.			
2	Schedule the training and change status to "Training Completed".	System prompts for Attendance and Evaluation results.			
3	Enter Attendance: "Present" and Score: "85". Click Update.	System records the score.			
4		Staff Competency Status updates to "Competent".			
5	Check the "Competency Log" for STF-101.	The log shows the training date and "Competent" status.			

8.8.2 TC006_03: Test <Manage Competency & Training (SD06 - AF)>

Test Case ID	TC006_02	Test Case Description	Verify system triggers "Retraining Required" when staff fails evaluation.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.2
QA Tester's Log	Alternate Flow				
Tester's Name	Rafiq	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Manager logged in.		1	Staff ID: STF-102 (Hazim)	
2	Training session concluded.		2	Course: Advanced Welding	
3			3	Score Input: 45/100 (Fail)	
4			4		
Test Scenario	Verify that a failing score prevents the "Competent" status and flags for retraining.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Select the staff member STF-102 awaiting evaluation.	Input fields for Score are enabled.			
2	Enter Score: "45". Click Update.	System displays an alert: "Score below passing threshold."			
3	Confirm submission of the failing score.	Staff Competency Status updates to "Not Competent" or "Retraining Required".			
4		The Training Gap remains open.			
5	Verify the Training Schedule.	System suggests or allows rescheduling the training.			

8.8.3 TC006_03: Test <Manage Competency & Training (SD06 - EF)>

Test Case ID	TC006_03	Test Case Description	Verify system handles "Absent" status correctly without updating competency.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.3
QA Tester's Log	Exception Flow				
Tester's Name	Hazim	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Training session was scheduled for today.		1	Staff ID: STF-103 (Amyrul)	
2			2	Attendance Input: "Absent"	
3			3		
4			4		
Test Scenario	Verify that marking a staff member as "Absent" keeps the training gap pending.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Locate the scheduled training session for STF-103.	Session is visible in the "Scheduled" list.			
2	In the "Record Attendance" field, select "Absent".	The "Evaluation Score" field becomes disabled (grayed out).			
3	Click "Update Record".	System records the absence.			
4		Competency Status remains "Pending Training" (Does NOT change to Competent).			
5	Check the system alerts/dashboard.	Notification sent to Manager: "Staff missed scheduled training."			

8.9 TC010: Test <Admin & System Support Module> Subsystem 3: <Provide System Support (UC010)>

This test contains the following test cases:

- (a) TC010_01: Test <Provide System Support (SD010 - NF)>
- (b) TC010_02: Test <Provide System Support (SD010 - AF)>
- (c) TC010_03: Test <Provide System Support (SD010 - EF)>

8.9.1 TC010_01: Test <Provide System Support (SD010 - NF)>

Test Case ID	TC010_01	Test Case Description	Test <Provide System Support (SD010 - NF)				
Created By	Rafiq	Reviewed By	Rasyid	Version	1		
QA Tester's Log							
Tester's Name			Date Tested	Test Case (Pass/Fail/Not)			
S#	Prerequisites:		S#	Test Data			
1	Admin logged in; support portal active		1	Support ticket resolved or escalated			
Test Scenario	Verify that the Admin can successfully access and view the system audit trail.						
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended			
1	Navigate to Support Dashboard.	System displays list of "Open" tickets.					
2	Click on Ticket ID #TKT101.	System loads ticket details (Sender, Issue, Date).					
3	Enter text in "Resolution" field: "Issue resolved by resetting database permissions."	Text area accepts input without validation error.					
4	Click "Update Status to Resolved" button.	1. System updates Ticket Status to "Resolved". 2. Success message displayed: "Resolution Saved".					
5	Check "Resolved" tab in Dashboard.	Ticket #TKT101 appears in the Resolved list.					

8.9.2 TC010_02: Test <Provide System Support (SD010 - AF)>

Test Case ID	TC010_02	Test Case Description	Test <Provide System Support (SD010 - AF)	
Created By	Rafiq	Reviewed By	Rasyid	Version
QA Tester's Log				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data
1	Admin user is logged in.		1	Logs for specific users exist in the database.
Test Scenario Verify that the Admin can filter the audit trail to view activities for a specific user only.				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to Audit Logs page.	System displays full log list.		
2	Locate the "Filter by User ID" search box.	Input field is editable.		
3	Enter User ID: "AUDITOR_02".	Text field accepts alphanumeric input.		
4	Click "Search" / "Filter" button.	1. Page refreshes. 2. List only shows rows where User ID is AUDITOR_02. 3. Total record count updates to match the filtered results.		

8.9.3 TC0010_03: Test <Provide System Support (SD010 - EF)>

Test Case ID	TC010_03	Test Case Description	Test <Provide System Support (SD010 - EF)	
Created By	Rafiq	Reviewed By	Rasyid	Version
QA Tester's Log				
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data
1	Admin user is logged in.		1	URL: http://doremi-soft.com/admin/logs
2	The System Log database is temporarily disconnected or unreachable (simulated).		2	Action: Click "Retrieve" during a network timeout.
Test Scenario Verify that the system displays a user-friendly error message instead of crashing when the log database is unreachable.				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to Audit Logs page.	System attempts to load the interface.		
2	(Simulate DB Failure) Disconnect database or simulate timeout.	Network/Database connection is lost.		
3	Click "Retrieve Logs" button.	System attempts to fetch data but fails after a set timeout period.		
4	Observe the system response.	1. System does not crash (no White Screen of Death). 2. Error message displayed: "Unable to retrieve logs. Please try again later." 3. A "Retry" button is visible.		

8.10 TC011: Test <Admin & System Support Module> Subsystem 3: <Maintain System Logs (UC011)>

This test contains the following test cases:

- (a) TC011_01: Test <Maintain System Logs (SD011 - NF)>
- (b) TC011_02: Test <Maintain System Logs (SD011 - AF)>
- (c) TC011_03: Test <Maintain System Logs (SD011 - EF)>

8.10.1 TC011_01: Test <Maintain System Logs (SD011 - NF)>

Test Case ID	TC011_01	Test Case Description	Test <Maintain System Logs (SD011 - NF)>		
Created By	Rafiq	Reviewed By	Afiq Irfan	Version	1
QA Tester's Log					
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Admin logged in; logging service active		1	Logs available for audit and traceability	
Test Scenario	Verify that an Admin can view, filter, and export system logs for accountability.				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	System automatically records activities (logins, approvals, changes)	System displays a list of automatically recorded activities.			
2	Admin selects "view logs"	System displays logs matching the selected filter criteria.			
3	Admin filters and exports logs	System generates a downloadable file of the filtered logs for audit.			

8.10.2 TC011_02: Test <Maintain System Logs (SD011 - AF)>

Test Case ID	TC011_02	Test Case Description	Test <Maintain System Logs (SD011 - AF)>		
Created By	Rafiq	Reviewed By	Hazim	Version	1.2
QA Tester's Log					
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Admin logged in; logging service active		1	Logs available for audit and traceability	
Test Scenario	Verify that the system automatically detects corrupted log data and initiates a successful recovery process to maintain traceability.				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Navigate to the "System Logs" section.	System attempts to load activity logs.			
2	System encounters a "Log Corruption" state (simulated).	Corruption" state (simulated). System triggers the internal recovery protocol automatically.			
3	Monitor the recovery progress indicator/notification.	A message appears stating "Corruption detected. System is recovering logs..."			
4	Wait for the recovery process to complete.	1. Recovery finishes successfully. 2. Logs are restored to a readable state.			
5	Verify log availability.	Logs are available for audit and traceability.			

8.10.3 TC011_03: Test <Maintain System Logs (SD011 - EF)>

Test Case ID	TC011_03	Test Case Description	Test <Maintain System Logs (SD011 - EF)>		
Created By	Rafiq	Reviewed By	Afiq Syahir	Version	1.3
QA Tester's Log					
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Admin logged in; logging service active		1	Logs available for audit and traceability	
Test Scenario	Verify that the system triggers an immediate alert to the Admin if the logging service becomes unavailable.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Simulate the "Logging service down" condition.	System detects that the service is no longer recording activities.			
2	Navigate to the Admin Dashboard or Notification Center.	An alert is triggered and clearly displayed to the Admin.			
3	Attempt to perform a system activity (e.g., login/change).	The system fails to record the activity but keeps the "Service Down" alert active.			
4	Review the alert details.	Alert specifically identifies that the logging service is down.			

8.11 TC012: Test <Risk & Incident Management>

Subsystem: <Manage Risks (UC012)>

- a) **TC012_01:** Test <Manage Risk (SD012 - NF)>
- b) **TC012_02:** Test <Manage Risk (SD012-AF1)>
- c) **TC012_03:** Test <Manage Risk (SD012-AF2)>
- d) **TC012_04:** Test <Manage Risk (SD012-EF1)>
- e) **TC012_05:** Test <Manage Risk (SD012-EF2)>

8.9.1 TC012_01: Test <Manage Risk (SD012 - NF)>

Test Case ID	TC012_01	Test Case Description	Test risk identification, scoring, and mitigation submisioin		
Created By	Afiq	Reviewed By	Hazim	Version	1
QA Tester's Log					
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Manager has logged into the Risk Management UI		1	Risk Description = "Supplier delay"	
2	Risk categories are pre-defined in the system		2	Risk Category = "Compliance"	
3	Risk scoring logic (Likelihood x impact) is implemented		3	Likelihood = 4	
4	Mitigation plan field is available and editable		4	Impact = 5	
			5	Mitigation = "Switch supplier"	
Test Scenario					
Verify that when a manager (Engineer persona) enters valid risk details (description, category, likelihood, impact) and provides a mitigation plan, the system correctly calculates the risk score (Likelihood x Impact), saves the risk record into the Risk Register, and displays a confirmation message "Risk Added Successfully".					
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Manager enters risk description and category	Input accepted			
2	Manager selects likelihood and impact	Values accepted			
3	System calculates risk score	Score = 20 displayed			
4	Manager enters mitigation plan	Input accepted			
5	Manager submits risk entry	Submit button clickable			
6	System saves risk record	Message: "Risk Added Successfully"			

8.9.2 TC012_02: Test <Manage Risk (SD012-AF1)>

Test Case ID	TC012_02	Test Case Description	test system behaviour when manager or engineer enters a risk with very low likelihood and impact values		
Created By	Afiq	Reviewed By	Hazim	Version	1
QA Tester's Log					
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Manager or Engineer has logged into the Risk Management UI		1	Risk Description = "Minor UI glitch"	
2	Risk categories are pre-defined in the system		2	Risk Category = "Technical"	
3	Risk scoring logic (Likelihood x impact) is implemented		3	Likelihood = 1	
4	Mitigation plan field is available and editable		4	Impact = 1	
			5	Mitigation Plan = "Monitor and fix in next release"	
			6	Risk Score = 1 (auto-calculated by system)	
Test Scenario					
Verify that when a manager enters a risk with very low likelihood and impact values, the system calculates a low priority score, saves the risk record, and confirms that the risk is logged successfully with "Low Priority" status.					
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Manager enters risk description and category	Input accepted			
2	Manager selects likelihood = 1 and impact = 1	Values accepted			
3	System calculates risk score	Score = 1 displayed			
4	Manager enters mitigation plan	Input accepted			
5	Manager submits risk entry	Submit button clickable			
6	System saves risk record	Message: "Risk Added Successfully" with tag "Low Priority"			

8.9.3 TC012_03: Test <Manage Risk (SD012-AF2)>

Test Case ID	TC012_03	Test Case Description	Verifies the system behaviour when a manager attempts to log a risk that already exist in repository		
Created By	Afiq	Reviewed By	Rasyid	Version	1
QA Tester's Log					
Tester's Name			Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Manager has logged into the Risk Management UI		1	Risk Description = "Supplier delay"	
2	Risk categories are pre-defined in the system		2	Risk Category = "Compliance"	
3	Risk repository already contains an existing risk with the same description and category		3	Likelihood = 4	
4	Risk scoring logic (Likelihood × Impact) is active		4	Impact = 5	
5	Submit button is enabled for valid entries		5	Mitigation Plan = "Switch supplier to local vendor"	
			6	Risk Score = 20 (auto-calculated)	
Test Scenario	Verify that when a manager submits a risk entry that duplicates an existing record, the system prevents saving and alerts the user that the risk is already logged				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Manager enters risk description and category	Input accepted			
2	Manager selects likelihood and impact	Values accepted			
3	System calculates risk score	Score = 20 displayed			
4	Manager enters mitigation plan	Input accepted			
5	Manager submits risk entry	Submit button clickable			
6	System checks repository for existing risk	Duplicate detected			

8.9.4 TC012_04: Test <Manage Risk (SD012-EF1)>

Test Case ID	TC012_04	Test Case Description	verifies the system's behavior when a manager or engineer attempts to log a risk entry without filling in the mitigation plan field		
Created By	Afiq	Reviewed By	Rasyid	Version	1
QA Tester's Log					
Tester's Name			Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Manager has logged into the Risk Management UI		1	Risk Description = "Backup failure"	
2	Risk categories are pre-defined in the system		2	Risk Category = "Operational"	
3	Risk repository already contains an existing risk with the same description and category		3	Likelihood = 3	
4	Risk scoring logic (Likelihood × Impact) is active		4	Impact = 4	
5	Submit button is enabled for valid entries		5	Mitigation Plan = [empty]	
			6	Risk Score = 12 (auto-calculated)	
Test Scenario	Verify that when a manager submits a risk entry without a mitigation plan, the system prevents saving and displays a validation error message				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Manager enters risk description and category	Input accepted			
2	Manager selects likelihood = 3 and impact = 4	Values accepted			
3	System calculates risk score	Score = 12 displayed			
4	Manager leaves mitigation plan blank	Field validation triggered			
5	Manager clicks submit	Submit action blocked			
6	System displays error message	"Mitigation plan required"			

8.9.5 TC012_05: Test <Manage Risk (SD012-EF2)>

Test Case ID	TC012_05	Test Case Description	This test verifies the system's behavior when a manager or engineer attempts to log a risk using invalid scoring values for likelihood or impact (e.g., negative numbers or values greater than the maximum allowed)		
Created By	Afiq	Reviewed By	afiq irfan	Version	1
<u>QA Tester's Log</u>					
Tester's Name		Date Tested	Test Case (Pass/Fail/Not)		
S#	Prerequisites:		S#	Test Data	
1	Engineer/Manager is logged into the Risk Management UI		1	Risk Description = "Data backup delay"	
2	Risk categories are already defined in the system		2	Risk Category = "Operational"	
3	Risk scoring logic requires Likelihood and Impact values between 1-5		3	Likelihood = -1 (invalid)	
4	Mitigation plan field is available		4	Impact = 6 (invalid)	
5	Submit button is enabled for valid entries		5	Mitigation Plan = "Introduce automated backup schedule"	
			6	Risk Score = [not calculated due to invalid input]	
Test Scenario		Verify that when a manager submits a risk entry with invalid scoring values, the system prevents saving and displays a validation error message			
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	Manager enters risk description and category	Input accepted			
2	Manager enters invalid likelihood (-1) and impact (6)	System detects invalid values			
3	System attempts to calculate risk score	Calculation blocked			
4	Manager enters mitigation plan	Input accepted			
5	Manager clicks submit	Submit action blocked			
6	System displays error message	"Invalid scoring values, must be between 1-5"			

8.12 TC013: Test <Risk & Incident Management>

Subsystem: <Manage Incident & Feedback (UC013)>

This test contains the following test cases:

- a) **TC013_01:** Test <Manage Incident & Feedback (SD013 - NF)>
- b) **TC013_02:** Test <Manage Incident & Feedback (SD013-AF1)>
- c) **TC013_03:** Test <Manage Incident & Feedback (SD013-AF2)>
- d) **TC006_04:** Test <Manage Incident & Feedback (SD013-EF1)>
- e) **TC006_05:** Test <Manage Incident & Feedback (SD013-EF2)>

8.12.1 TC013_01: Test <Manage Incident & Feedback (SD013 - NF)>

Test Case ID	TC013_01	Test Case Description	This test verifies the system's behavior when an auditor, manager, or engineer logs a valid incident with supporting evidence and severity classification		
Created By	Afiq	Reviewed By	raffiq	Version	1
QA Tester's Log					
Tester's Name			Test Case (Pass/Fail/Not)		
S#	Prerequisites:			S#	Test Data
1	User (Engineer/Manager/Auditor) is logged into the Incident Management UI			1	Incident Description = "Customer complaint on product defect"
2	Incident categories and severity levels are pre-defined in the system			2	Incident Category = "Customer Feedback"
3	CAPA module is active and integrated			3	Evidence = "Image of defective product"
4	Evidence upload functionality is available			4	Severity = "High"
5	Submit button is enabled for valid entries			5	IncidentID = "INC123" (system-generated)
				6	Feedback Comments = "Product quality issue reported by customer"
Test Scenario	Verify that when a user logs an incident with valid description, evidence, and severity classification, the system records the incident, assigns a status, and escalates to CAPA if severity is high				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	User enters incident description and category	Input accepted			
2	User uploads evidence file	Evidence accepted			
3	User selects severity = "High"	Value accepted			
4	System generates IncidentID	IncidentID = "INC123" displayed			
5	User submits incident entry	Submit button clickable			
6	System saves incident record	Message: "Incident Logged Successfully"			
7	System checks severity level	Severity = High triggers CAPA			
8	System escalates incident	CAPA initiated, status = "Escalated to CAPA"			

8.12.2 TC013_02: Test <Manage Incident & Feedback (SD013-AF1)>

Test Case ID	TC013_02	Test Case Description	This test verifies the system's behavior when a user logs a minor incident with valid description and severity classification set to "Low."		
Created By	Afiq	Reviewed By	Rasyid	Version	1
QA Tester's Log					
Tester's Name			Test Case (Pass/Fail/Not)		
S#	Prerequisites:			S#	Test Data
1	User (Engineer/Manager/Auditor) is logged into the Incident Management UI			1	Incident Description = "Packaging issue reported by staff"
2	Incident categories and severity levels are pre-defined in the system			2	Incident Category = "Operational"
3	Evidence upload functionality is available			3	Evidence = "Photo of packaging defect"
4	Submit button is enabled for valid entries			4	Severity = "Low"
				5	IncidentID = "INC124" (system-generated)
				6	Feedback Comments = "Minor defect, no customer impact"
Test Scenario	Verify that when a user logs a minor incident with severity set to "Low," the system records the incident, assigns a status of "Logged Only," and confirms the entry without escalating to CAPA				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	User enters incident description and category	Input accepted			
2	User uploads evidence file	Evidence accepted			
3	User selects severity = "Low"	Value accepted			
4	System generates IncidentID	IncidentID = "INC124" displayed			
5	User submits incident entry	Submit button clickable			
6	System saves incident record	Message: "Incident Logged Successfully"			
7	System checks severity level	Severity = Low			
8	System assigns status	Status = "Logged Only," no CAPA triggered			

8.12.3 TC013_03: Test <Manage Incident & Feedback (SD013-AF2)>

Test Case ID	TC013_03	Test Case Description	This test verifies the system's behavior when a user submits feedback-only data (comments, ratings) without incident details or evidence		
Created By	Afiq	Reviewed By	afiq irfan	Version	1
QA Tester's Log					
Tester's Name			Date Tested	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	User (Engineer/Manager/Auditor) is logged into the Incident & Feedback UI		1	Feedback Comments = "Service delay experienced during delivery"	
2	Feedback submission module is active		2	Rating = 2 (scale 1-5)	
3	Incident fields (description, evidence) are optional in this flow		3	Incident Description = [empty]	
4	Submit button is enabled for valid feedback entries		4	Evidence = [none]	
			5	Severity = [not applicable]	
			6	FeedbackID = "FB001" (system-generated)	
Test Scenario	Verify that when a user submits feedback-only data without incident details, the system stores the feedback, generates a FeedbackID, and confirms the submission without creating an incident record				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	User enters feedback comments	Input accepted			
2	User selects rating = 2	Value accepted			
3	User leaves incident description blank	No incident record created			
4	User does not upload evidence	Evidence optional, no error			
5	User clicks submit	Submit button clickable			
6	System generates FeedbackID	FeedbackID = "FB001" displayed			
7	System saves feedback record	Message: "Feedback recorded successfully"			
8	System confirms no incident record created	Feedback-only entry stored			

8.12.4 TC006_04: Test <Manage Incident & Feedback (SD013-EF1)>

Test Case ID	TC013_04	Test Case Description	This test verifies the system's behavior when a user attempts to log an incident without providing required evidence		
Created By	Afiq	Reviewed By	raffiq	Version	1
QA Tester's Log					
Tester's Name			Date Tested	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	User (Engineer/Manager/Auditor) is logged into the Incident Management UI		1	Incident Description = "Defect report from production line"	
2	Incident categories and severity levels are pre-defined in the system		2	Incident Category = "Operational"	
3	Evidence attachment is mandatory for incident logging		3	Evidence = [empty]	
4	Submit button is enabled for valid entries		4	Severity = "Medium"	
			5	IncidentID = [not generated due to error]	
			6	Feedback Comments = "Reported by QA staff"	
Test Scenario	Verify that when a user submits an incident without attaching evidence, the system prevents saving and displays a validation error message				
S#	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended	
1	User enters incident description and category	Input accepted			
2	User selects severity = "Medium"	Value accepted			
3	User leaves evidence field blank	Validation triggered			
4	User clicks submit	Submit action blocked			
5	System checks for evidence	Evidence missing			
6	System displays error message	"Evidence required for incident logging"			
7	Incident record not saved	No IncidentID generated			

8.12.5 TC006_05: Test <Manage Incident & Feedback (SD013-EF2)>

Test Case ID	TC013_05	Test Case Description	This test verifies the system's behavior when a user logs a high severity incident that requires escalation to CAPA, but the CAPA module is unavailable		
Created By	Afiq	Reviewed By	afiq irfan	Version	1
QA Tester's Log					
Tester's Name			Test Case (Pass/Fail/Not)		
S#	Prerequisites:	S#	Test Data		
1	User (Engineer/Manager/Auditor) is logged into the Incident Management UI	1	Incident Description = "Critical system failure in production"		
2	Incident categories and severity levels are pre-defined in the system	2	Incident Category = "Operational"		
3	CAPA module is integrated but currently unavailable	3	Evidence = "System log file"		
4	Submit button is enabled for valid entries	4	Severity = "High"		
		5	IncidentID = "INC125" (system-generated)		
		6	Feedback Comments = "Reported by IT operations team"		
Test Scenario	Verify that when a user submits a high severity incident requiring CAPA escalation, but the CAPA module is unavailable, the system records the incident, prevents escalation, and displays an error message while keeping the incident status as "Pending CAPA."				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	User enters incident description and category	Input accepted			
2	User uploads evidence file	Evidence accepted			
3	User selects severity = "High"	Value accepted			
4	System generates IncidentID	IncidentID = "INC125" displayed			
5	User submits incident entry	Submit button clickable			
6	System saves incident record	Message: "Incident Logged Successfully"			
7	System attempts CAPA escalation	CAPA module unavailable			
8	System displays error message	"CAPA service unavailable"			
9	Incident status remains	Status = "Pending CAPA"			

8.13 TC007: Test <CAPA Management Module> Subsystem 4:

<Manage Equipment & Inventory (UC07)>

- (a) TC007_01: Test <Manage Equipment & Inventory (SD07 - NF)>
- (b) TC007_02: Test <Manage Equipment & Inventory (SD07 - AF)>
- (c) TC007_03: Test <Manage Equipment & Inventory (SD07 - EF)>

8.13.1 TC007_01: Test <Manage Equipment & Inventory (SD07 - NF)>

Test Case ID	TC007_01	Test Case Description	Verify equipment status updates to "Ready" after successful calibration.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.1
QA Tester's Log	Normal Flow				
Tester's Name	Rasyid	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:				
1	Equipment status is currently "Due for Maintenance" or "Inactive".				
2	Engineer is logged in.				
3					
4					
Test Scenario	Verify that logging positive calibration data updates the "Equipment Readiness Status".				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Select Asset EQ-CNC-04 from the maintenance list.	System loads the input form for Calibration Data.			
2	Enter the maintenance results and mark Calibration as "Pass".	System accepts the input.			
3	Set the new "Next Maintenance Date" and click "Update".	Output: System updates status to "Ready".			
4		Maintenance Log is updated.			
5	Verify Equipment List.	Asset EQ-CNC-04 appears with a green/Ready status indicator.			

8.13.2 TC007_02: Test <Manage Equipment & Inventory (SD07 - AF)>

Test Case ID	TC07_02	Test Case Description	Verify equipment remains "Out of Service" if calibration fails.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.2
QA Tester's Log	Alternate Flow				
Tester's Name	Rafiq	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:				
1	Equipment is undergoing maintenance.				
2					
3					
4					
Test Scenario	Verify that logging negative results prevents the status from becoming "Ready".				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	Select Asset EQ-PRESS-09.	Input form loads.			
2	Enter Calibration Data indicating a failure/issue.	System flags the failure.			
3	Click "Submit Report".	Output: Status updates to "Out of Service" or "Repair Required".			
4		A work order for repair is generated (optional).			
5	Check Equipment Readiness Status.	Status is "NOT Ready".			

8.13.3 TC007_02: Test <Manage Equipment & Inventory (SD07 - EF)>

Test Case ID	TC007_03	Test Case Description	Verify system automatically flags equipment when maintenance is due.		
Created By	Afiq Irfan	Reviewed By	Afiq Shahir	Version	1.3
QA Tester's Log	Exception Flow				
Tester's Name	Hazim	Date Tested	16-Jan-2026	Test Case (Pass/Fail/Not)	
S#	Prerequisites:		S#	Test Data	
1	Automation script/Cron job runs daily.		1	Asset ID: EQ-DRILL-01	
2	System date matches a scheduled maintenance date.		2	Scheduled Date: [Today's Date]	
3			3		
4			4		
Test Scenario	Verify that the system changes status from "Ready" to "Due" without manual intervention.				
S#	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended
1	System Date rolls over to the scheduled maintenance date.	System background check runs			
2	Login as Engineer and view the Dashboard/Notification center.	Asset EQ-DRILL-01 is listed under "Upcoming Maintenance".			
3		Alert/Notification is generated.			
4	Check Asset Status.	Status may flag as "Due for Calibration".			

(a) Appendix A: Traceability Matrix

Test Case ID	Use Case ID/ Sequence Diagram ID	Package ID
TC003 for <Audit Management Module> Subsystem 1 <ul style="list-style-type: none">• TC003_01• TC003_02• TC003_03• TC003_04	UC003 <ul style="list-style-type: none">• SD003	P001
TC004 for <Audit Management Module> Subsystem 1 <ul style="list-style-type: none">• TC004_01• TC004_02• TC004_03	UC004 <ul style="list-style-type: none">• SD004	P001
TC001 for <Document & Change Management with KPI Dashboard Module> Subsystem 2 <ul style="list-style-type: none">• TC001_01• TC001_02• TC001_03	UC001 <ul style="list-style-type: none">• SD001	P002
TC002 for <Document & Change Management with KPI Dashboard Module> Subsystem 2 <ul style="list-style-type: none">• TC002_01• TC002_02• TC002_03	UC002 <ul style="list-style-type: none">• SD002	P002
TC008 for <Document & Change Management with KPI Dashboard Module> Subsystem 2 <ul style="list-style-type: none">• TC008_01• TC008_02• TC008_03	UC008 <ul style="list-style-type: none">• SD008	P002
TC009 for <Document & Change Management with KPI Dashboard Module> Subsystem 2 <ul style="list-style-type: none">• TC009_01• TC009_02• TC009_03	UC009 <ul style="list-style-type: none">• SD009	P002
TC010 for <Admin & System support Module> Subsystem 3 <ul style="list-style-type: none">• TC010_01• TC010_02• TC010_03	UC010 <ul style="list-style-type: none">• SD010	P003
TC011 for <Maintain System Logs> Subsystem 3 <ul style="list-style-type: none">• TC011_01• TC011_02• TC011_03	UC011 <ul style="list-style-type: none">• SD011	P003

TC005 for <CAPA Management Module> Subsystem 4 • TC005_01 • TC005_02 • TC005_03	UC005 • SD005	P004
TC006 for <CAPA Management Module> Subsystem 4 • TC006_01 • TC006_02 • TC006_03	UC006 • SD006	P004
TC007 for <CAPA Management Module> Subsystem 4 • TC007_01 • TC007_02 • TC007_03	UC007 • SD007	P004
TC0012 for <Risk & Incident Management Module> Subsystem 5 • TC012_01 • TC012_02 • TC012_03	UC012 • SD012	P005
TC0013 for <Risk & Incident Management Module> Subsystem 5 • TC013_01 • TC013_02 • TC013_03	UC013 • SD013	P005