# USAL

# Data Leakage Detection System

By: Ali Haydar-1123060

Mohammad Ashek-1123197

Hani Hammoud-1123108

To The Instructor:Dr.Ibrahim Dhainy

**SDLC MODEL:**

For the Data Leakage Detection System project, the Waterfall Model was selected as the SDLC approach. This model is appropriate because:  
  
1. Sequential Development Process:

- The project follows a structured step-by-step approach:  
 - Requirement Analysis – Identifying the need for data leakage detection.  
 - System Design – Planning data allocation strategies and implementation techniques.  
 - Implementation – Developing the system and applying security techniques.  
 - Testing – Ensuring the software functions correctly and securely.  
 - Deployment & Maintenance – Final implementation and updates.  
  
2. Comprehensive Documentation: The project includes detailed documentation, making it well-suited for the Waterfall Model, which requires clear planning before execution.  
  
3. No Iterative Development: Unlike Agile, the development was completed in a single phase without frequent revisions.

### Feasibility Study:

Technical Feasibility:  
- Use of Machine Learning Algorithms to detect anomalies in data access.  
- Compatibility with existing security infrastructures.  
- Secure database storage and encryption for sensitive information.  
  
Economic Feasibility:  
- Reduces financial losses caused by data breaches.  
- Cost-effective deployment using cloud-based solutions.  
- Open-source tools can be leveraged to minimize expenses.  
  
Legal Feasibility:  
- Ensures compliance with GDPR, HIPAA, and other data protection laws.  
- Encrypts data to protect against unauthorized access.  
- Implements user authentication and authorization mechanisms.

### **Market Feasibility:**

The Data Leakage Detection System addresses a growing need in the cybersecurity market, especially for organizations that handle sensitive data. As data privacy regulations tighten globally, the demand for tools that prevent unauthorized data access and leaks is increasing. This system offers a competitive solution by combining data tracking, agent analysis, and fake record detection. Therefore, there is a clear market opportunity for such a solution, especially in sectors like finance, healthcare, and government.

### **Operational Feasibility:**

The system is operationally feasible because it fits well within the standard practices of IT departments in organizations. It can be managed by system administrators and integrated into existing workflows with minimal training due to its user-friendly interface. Its modular design allows for easy updates and maintenance, supporting long-term use. Users and administrators can operate the system effectively with basic technical knowledge.

### **Scheduling Feasibility:**

The project timeline is manageable using a Waterfall approach, allowing each development phase to be completed in order. Since the project has clear functional and non-functional requirements, tasks such as design, coding, testing, and deployment can be scheduled realistically. The system can be developed within a standard academic or professional project timeline ( 8–12 weeks). Any delays can be mitigated through careful planning and progress tracking.

## Table of Contents

1- Introduction

1.1 Product Scope

1.2 Product Value

1.3 Intended Audience

1.4 Intended Use

1.5 General Description

2- Functional Requirements

3- External Interface Requirements

3.1 User Interface Requirements

3.2 Hardware Interface Requirements

3.3 Software Interface Requirements

3.4 Communication Interface Requirements

4- Non-Functional Requirements

4.1 Security

4.2 Capacity

4.3 Compatibility

4.4 Reliability

4.5 Scalability

4.6 Maintainability

4.7 Usability

4.8 Other Non-Functional Requirements

## System Model

The Data Leakage Detection System was developed using the Waterfall SDLC model, which follows a structured and sequential approach to ensure systematic development and security.

## Purpose of the Document

This document describes the software project Data Leakage Detection System, which is designed to identify and prevent unauthorized access to sensitive information. The system enhances data security, compliance, and real-time threat detection, reducing risks associated with data breaches.

## Introduction

The **Data Leakage Detection System** is a security-focused software application developed to address the risk of sensitive information being leaked by third-party agents. As organizations increasingly share confidential data with external parties, the risk of unauthorized disclosure grows. This system helps detect such leaks and identify the responsible agents using smart data distribution strategies. It serves as a proactive solution to protect data integrity and support compliance with data protection laws.

### ****1.1 Product Scope:****

The **Data Leakage Detection System** is designed to monitor and control the distribution of sensitive data shared with third parties or agents. Its main goal is to detect unauthorized sharing of data and identify the source of the leak. The system includes features like file tracking, fake data injection, agent monitoring, and secure access control. It is suitable for organizations that handle confidential information and need an internal safeguard against data breaches.

### 1.2 ****Product Value:****

This system provides significant value by protecting sensitive data from unauthorized distribution, thereby reducing the risk of financial loss or reputational damage. It helps organizations comply with legal and regulatory standards such as GDPR and HIPAA. By detecting leaks early and identifying responsible agents, it enhances accountability and data security. It also minimizes the need for external forensic investigations.

### 1.3 ****Intended Audience:****

The intended users of this system include **IT administrators, security analysts**, and **data compliance officers** within organizations. It is also relevant for **government agencies, healthcare providers, financial institutions,** and **any enterprise managing confidential records.** The system supports teams responsible for ensuring data integrity and protecting against insider threats. Additionally, software developers may use it for integrating data-leak detection features into larger systems.

### 1.4 ****Intended Use:****

The system is intended to securely distribute data to trusted agents while monitoring for leaks. It allows uploading files to agents, securing them with secret keys, and tracking any attempts to duplicate or misuse the data. If a leak occurs, the system helps trace it back to the responsible party using allocation models and fake object tracking. It is also used for alerting administrators about suspicious activity in real-time.

### 1.5 ****General Description:****

The Data Leakage Detection System is a web-based application that combines secure data sharing with intelligent monitoring to prevent unauthorized access. It supports uploading, locking, and tracking files sent to multiple agents. Using techniques like fake data insertion and secret key validation, it can identify agents responsible for leaks without modifying original data. The system is built for scalability, reliability, and ease of use within enterprise environments.

## 2. Functional Requirements

## The **Data Leakage Detection System** includes key functionalities to monitor, detect, and prevent unauthorized data access. It continuously tracks user activities, file transfers, and access logs to identify suspicious behavior. The system employs **anomaly detection algorithms** to recognize unusual data movements and unauthorized sharing. It enforces **Role-Based Access Control (RBAC)** and **Multi-Factor Authentication (MFA)** to restrict access to sensitive information. Additionally, it generates **real-time alerts** for potential data leaks and maintains detailed **audit logs** for forensic analysis. The system also supports **data encryption**, automated incident reporting, and seamless integration with **SIEM (Security Information and Event Management) tools**, ensuring compliance with security regulations like **GDPR and HIPAA** while providing a comprehensive data protection framework.

## External Interface Requirements

### **3.1 User Interface Requirements**

The **Data Leakage Detection System** features a user-friendly interface designed for security analysts and administrators. It includes an **interactive dashboard** displaying system alerts, real-time monitoring data, and reports. The interface provides **graphical representations** of detected threats, customizable settings for access control, and an intuitive layout for easy navigation. Users can view logs, configure security policies, and manage incident responses efficiently.

### **3.2 Hardware Interface Requirements**

The system is designed to run on **standard enterprise hardware**, including **servers, workstations, and cloud environments**. It requires sufficient **processing power, storage, and memory** to handle large-scale data monitoring and analysis. The system must be compatible with **firewalls, intrusion detection systems (IDS), and other network security devices** to ensure seamless operation within an organization's IT infrastructure.

### **3.3 Software Interface Requirements**

The system integrates with existing **security software**, including **SIEM (Security Information and Event Management) tools, database management systems, and encryption frameworks**. It supports various **operating systems (Windows, Linux, and cloud-based platforms)** and works alongside **identity management solutions** for authentication and access control. The system also provides an **API** for integration with third-party cybersecurity tools.

### **3.4 Communication Interface Requirements**

Secure communication is essential for the system’s operation. It supports **encrypted data transmission** using **TLS (Transport Layer Security) and HTTPS** protocols to protect sensitive information. The system also enables **real-time notifications** via email or SMS to alert administrators of potential data breaches. Additionally, it facilitates **secure API communication** with external security tools for enhanced monitoring and response.

## 4. Non-Functional Requirements

### **4.1 Security**

Security is a core aspect of the system, ensuring that **sensitive data is protected from unauthorized access and breaches**. The system implements **Multi-Factor Authentication (MFA)**, **Role-Based Access Control (RBAC)**, and **end-to-end encryption (AES, TLS)** for data security. Additionally, it logs all activities to support **forensic investigations** and maintains **intrusion detection mechanisms**.

### **4.2 Capacity**

The system is designed to handle **large volumes of data and concurrent users** efficiently. It supports **high-speed data processing** and ensures **minimal performance degradation** under heavy load. The storage system is optimized to log extensive security events while maintaining fast retrieval times.

### **4.3 Compatibility**

The system is compatible with **multiple platforms**, including **Windows, Linux, and cloud-based environments**. It integrates with **existing security infrastructure**, such as **firewalls, SIEM (Security Information and Event Management) tools, and identity management systems**, ensuring seamless deployment across different IT environments.

### **4.4 Reliability**

The system must provide **continuous protection** with minimal downtime. It guarantees **99.9% uptime** through **automatic error handling, redundancy mechanisms, and failover support**. In case of system failures, **backup and recovery processes** ensure uninterrupted monitoring.

### **4.5 Scalability**

The system is **scalable** to accommodate growing security needs. It supports **enterprise-level deployments**, allowing organizations to **expand monitoring capabilities** without compromising performance. Cloud compatibility ensures **dynamic resource allocation** based on demand.

### **4.6 Maintainability**

Regular **updates, patches, and system optimizations** are essential for maintaining high security and performance. The system is built with a **modular architecture**, making it easy to add new features and integrate with evolving cybersecurity standards. **Automated diagnostics and monitoring tools** help detect issues early.

### **4.7 Usability**

Designed with a **user-friendly interface**, the system ensures **easy navigation, graphical dashboards, and real-time alert notifications**. Security teams can efficiently analyze reports, configure policies, and respond to threats with minimal training.

### **4.8 Other Non-Functional Requirements**

Additional requirements include **compliance with global security standards (GDPR, HIPAA, ISO 27001)**, **performance benchmarking for detection accuracy**, and **low resource consumption** to ensure optimal efficiency without overloading enterprise networks.

## Definitions and Acronyms

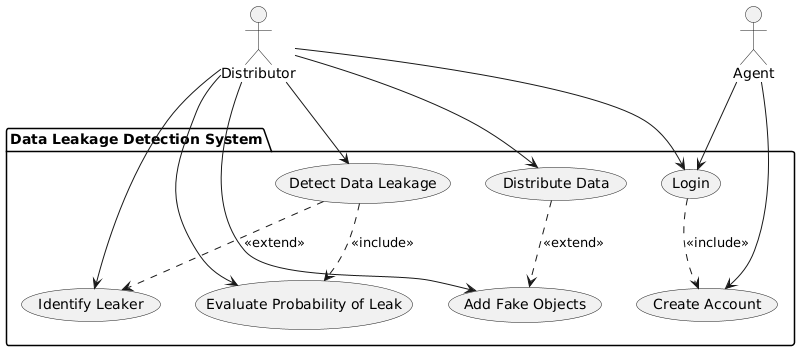
- DLP (Data Loss Prevention): Security strategy to prevent unauthorized data disclosure.  
- SIEM (Security Information and Event Management): A system that collects and analyzes security data.  
- GDPR (General Data Protection Regulation):EU law governing data privacy.  
- HIPAA (Health Insurance Portability and Accountability Act): US law ensuring patient data security.  
- RBAC (Role-Based Access Control): Security model restricting access based on roles.  
- TLS (Transport Layer Security): Protocol for secure data transmission.  
- MFA (Multi-Factor Authentication): Security mechanism requiring multiple verification factors.

## Conclusion

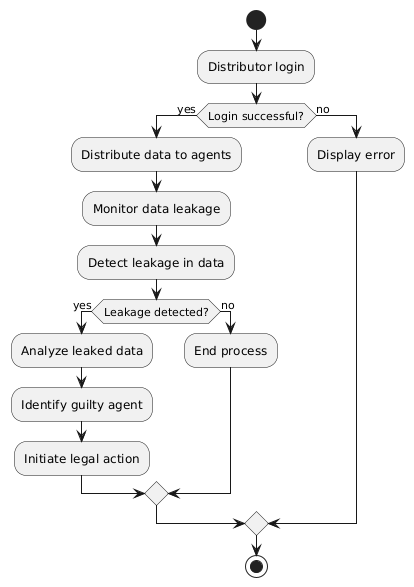
The Data Leakage Detection System is a critical cybersecurity solution designed to monitor, detect, and prevent unauthorized data access. By leveraging real-time monitoring, anomaly detection, and encryption, this system enhances data security and compliance with legal standards. With a scalable and maintainable architecture, organizations can efficiently safeguard their sensitive information against internal and external threats.

Phase 2:

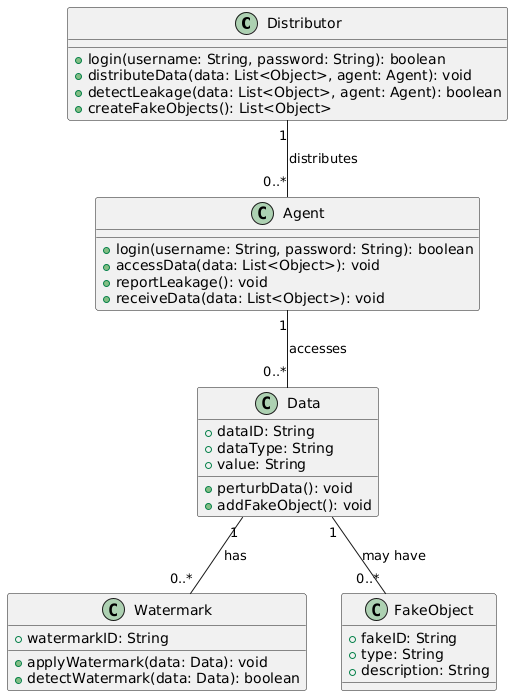
Use Case Diagram:



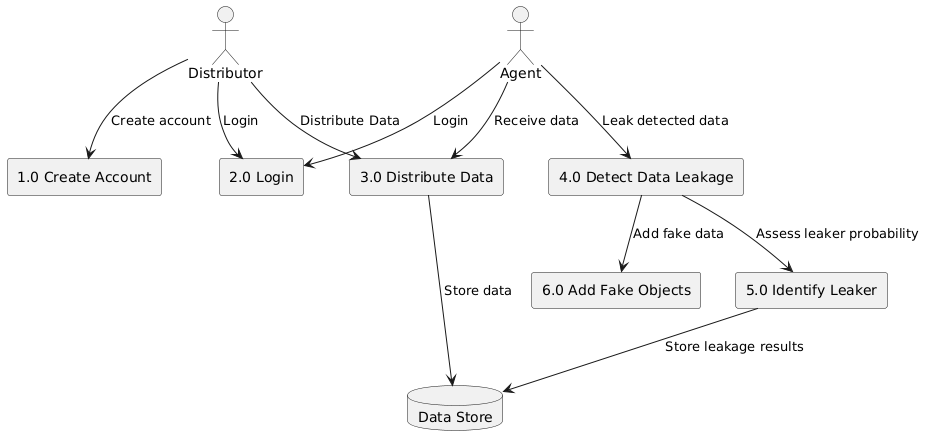
Activity Diagram:



Class Diagram:



Data Flow Diagram:



Phase 3:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task ID** | **Task Name** | **Predecessor** | **Duration** | **ES** | **EF** | **LS** | **LF** | **Critical?** | **Margin** |
| 1 | Create Account | — | 2 | 0 | 2 | 0 | 2 | Yes | 0 |
| 2 | Login | 1 | 1 | 2 | 3 | 2 | 3 | Yes | 0 |
| 3 | Upload File | 2 | 2 | 3 | 5 | 3 | 5 | Yes | 0 |
| 4 | Watermark Data | 3 | 3 | 5 | 8 | 5 | 8 | Yes | 0 |
| 5 | Distribute Data | 4 | 2 | 8 | 10 | 8 | 10 | Yes | 0 |
| 6 | Leak Detection Module | 5 | 4 | 10 | 14 | 10 | 14 | Yes | 0 |

2

2

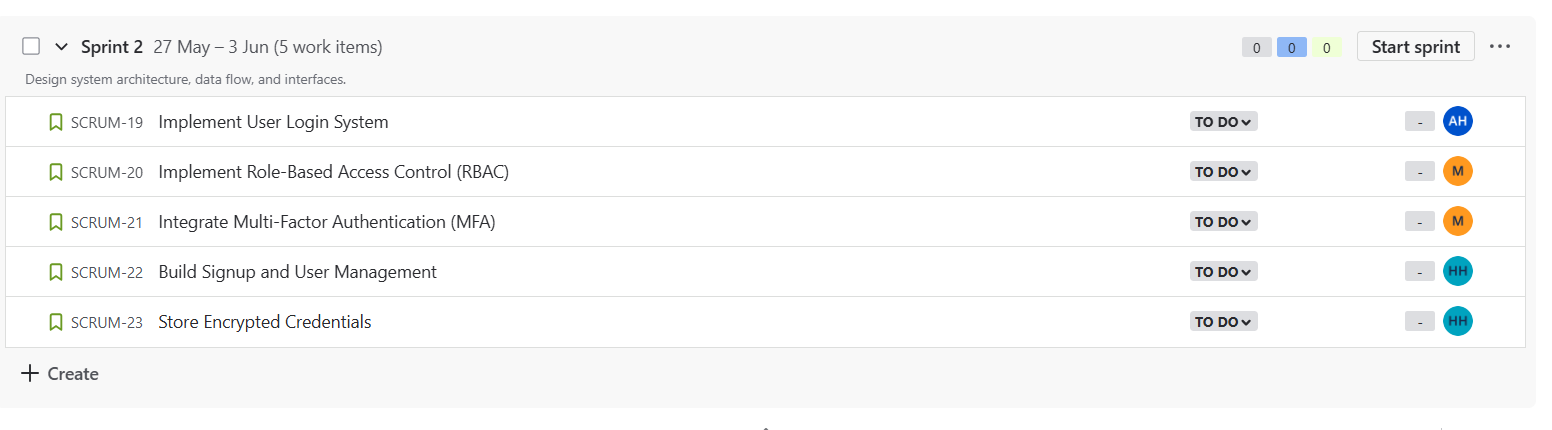
1

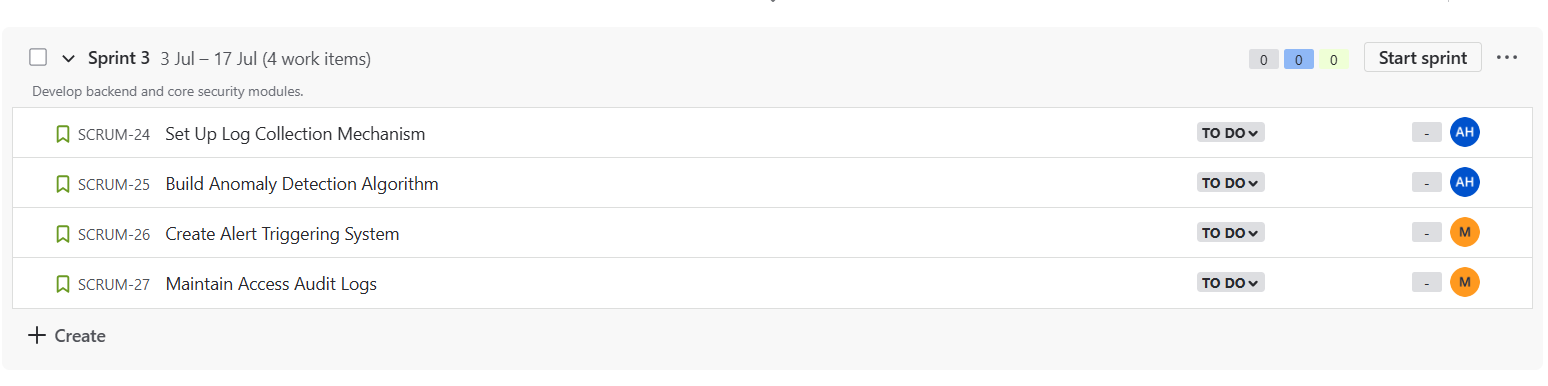
2

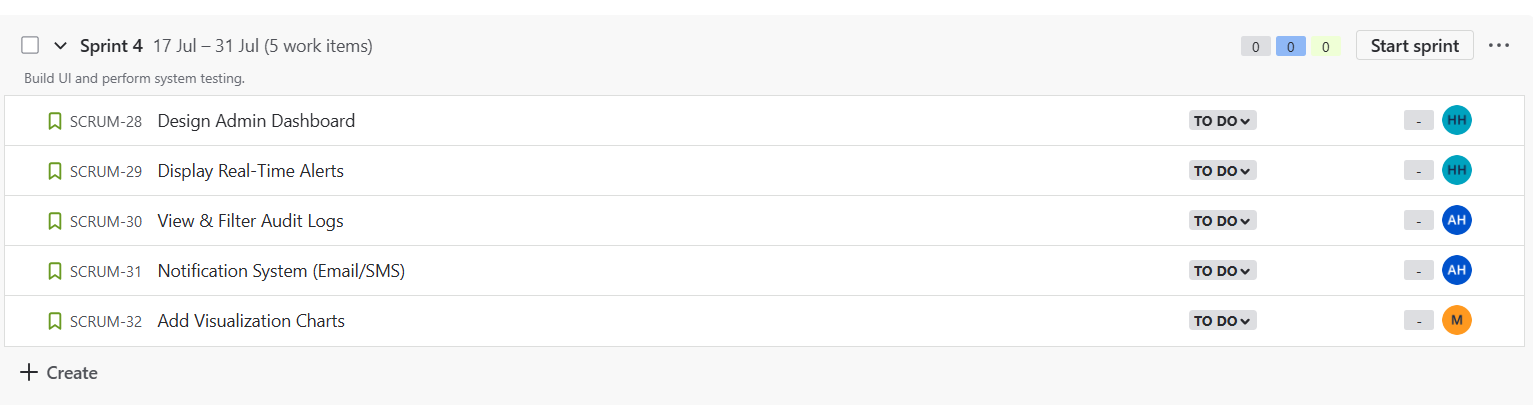
3

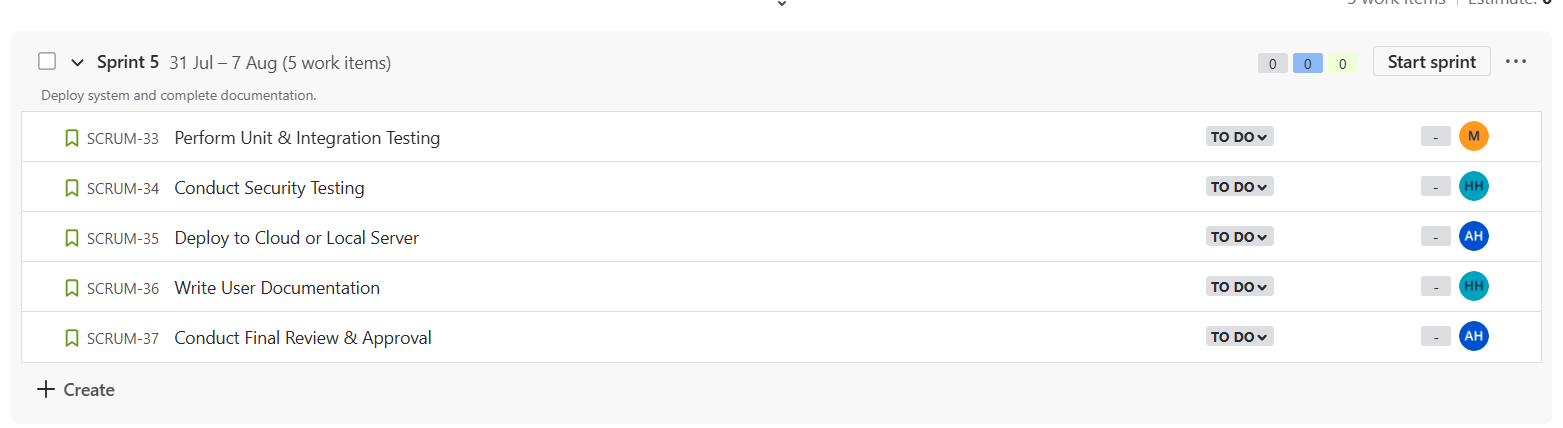
Phase 4:

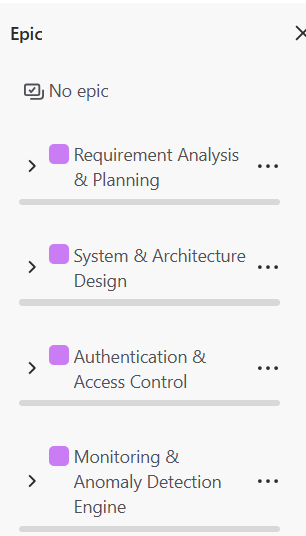


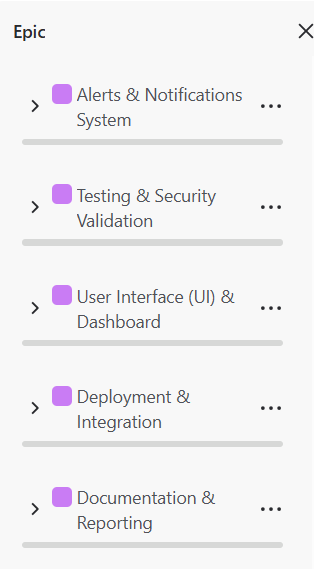


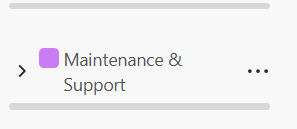


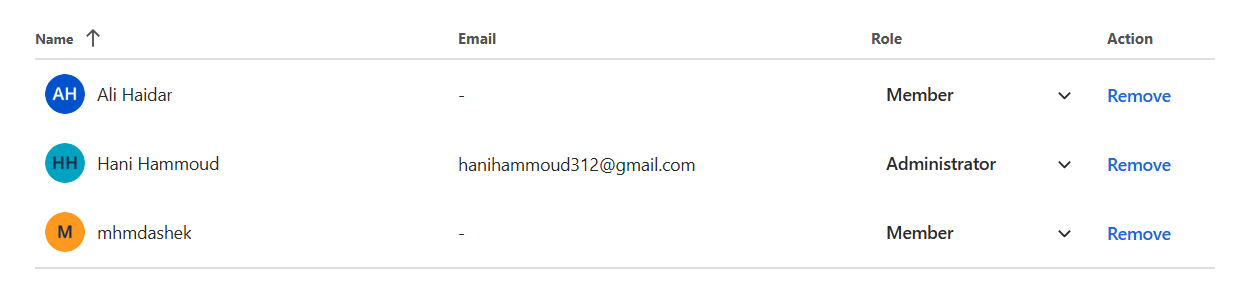












Phase 5:

