CAPSTONE PROJECT

TOPIC : Cloud based Threat Intelligence Dashboard

Software Design Document (SDD)

Submitted by:

Student Name	Enrollment Number
Chhavi Sharma	BT22GCS074
Khushi Jha	BT22GCS188
Sachin Warrior	BT22GCS165

Under the guidance of our faculty mentor : Manish Hurkat Sir

---

1. Introduction

1.1 Purpose

The purpose of this document is to define the **software architecture and design** of the **Cloud-Based Threat Intelligence Dashboard**. It serves as a guide for developers, architects, and stakeholders, ensuring a structured and efficient implementation of the system.

1.2 Scope

This document provides a **detailed system design**, covering:

- System architecture
- Component interactions
- Data flow, control flow, and deployment
- Class structure and state transitions
- User interactions and workflows

The system will aggregate threat intelligence data from external sources, process it for analysis, and present it via an interactive dashboard for cybersecurity professionals.

1.3 References

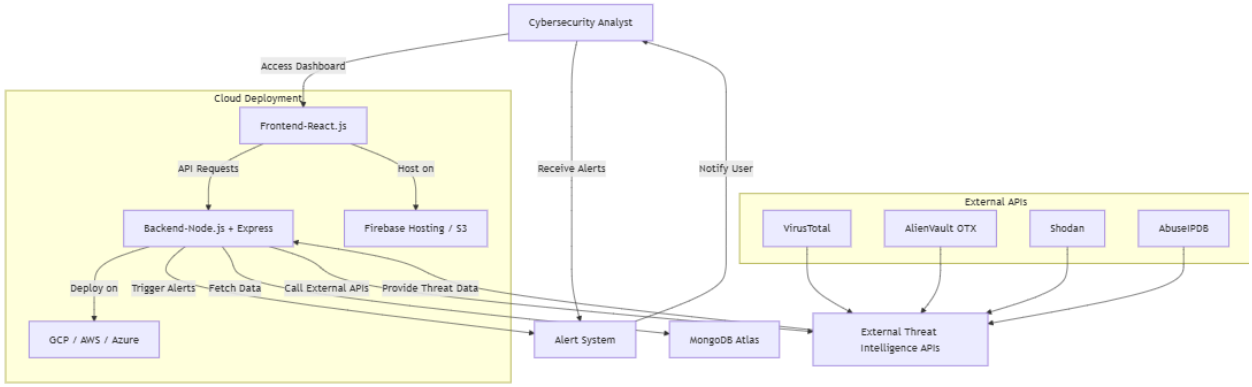
- **Software Requirement Specification (SRS)**
- **External API Documentation** (VirusTotal, AlienVault OTX, Shodan, AbuseIPDB)
- **MongoDB Atlas Documentation**

---

2. System Overview

The **Cloud-Based Threat Intelligence Dashboard** consists of three primary components:

1. **Frontend (React.js):** A user-friendly dashboard for viewing real-time cyber threat intelligence.
2. **Backend (Node.js + Express):** Manages data processing, API requests, and alert generation.
3. **Database (MongoDB Atlas):** Stores aggregated and normalized threat intelligence data.



---

3. Design Considerations

3.1 Assumptions

- The system will be **deployed on cloud infrastructure** (GCP/AWS/Azure).
- Threat intelligence data will be retrieved via **REST APIs**.
- The database will use **MongoDB Atlas** for **scalability and flexibility**.
- The alert system will be based on **heuristic rule-based detection**.

3.2 Constraints

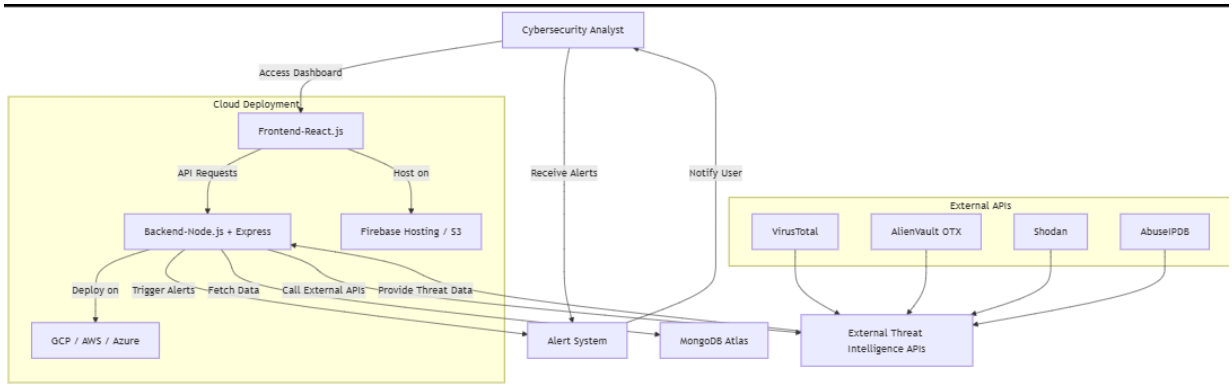
- **Real-time performance:** Threat data should be updated within **5 seconds**.
- **Security compliance:** Must follow **HTTPS**, **role-based access control (RBAC)**, and **data encryption**.
- **Cloud resource optimization:** Deployment should utilize **free-tier cloud services** where possible.

---

4. Architectural Design

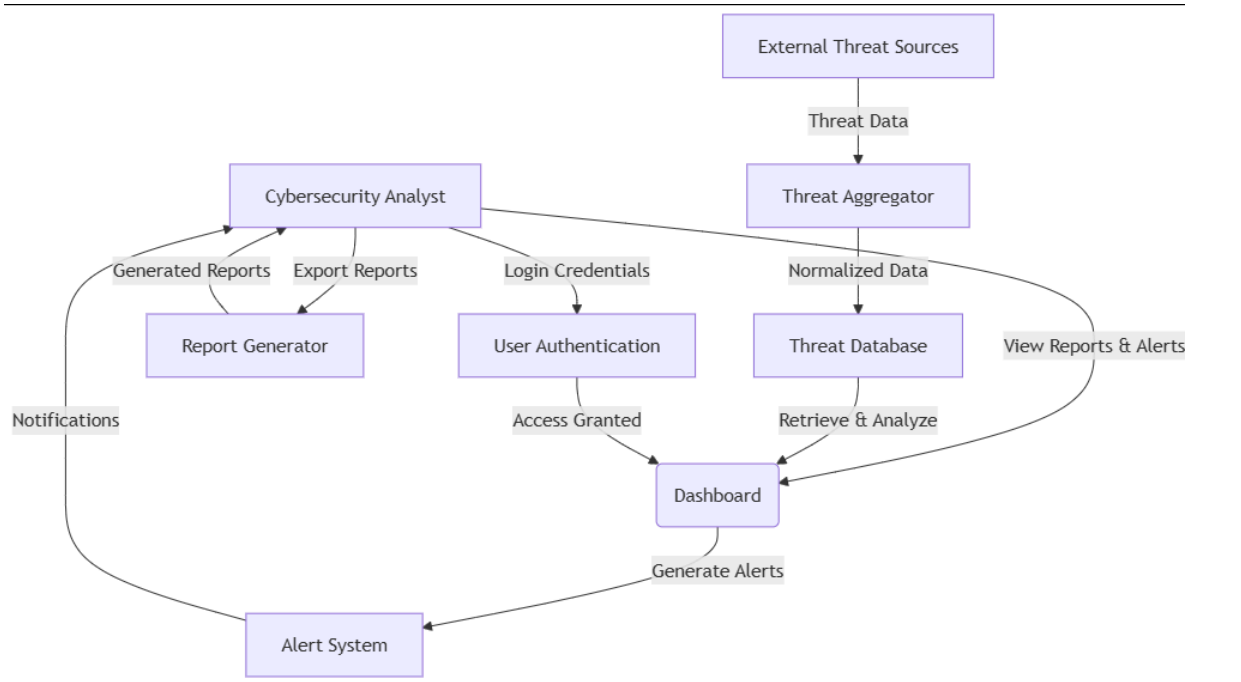
4.1 System Architecture

The system is designed as a **cloud-based, microservices-driven** application with separate layers for **frontend, backend, and data storage**.



4.2 Data Flow

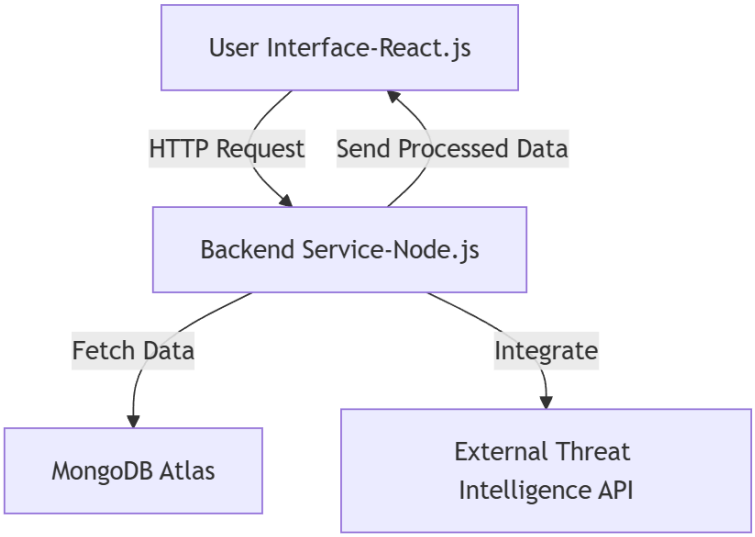
The system follows a structured **data ingestion, processing, and visualization pipeline**.



5. Detailed Design

5.1 Component Design

Each system component and its interactions are defined below:



5.1.1 Frontend (React.js)

- Displays **real-time threat analytics** and alert notifications.
- Fetches data via **REST API calls** to the backend.
- Enables **report generation and export functionality**.

5.1.2 Backend (Node.js + Express)

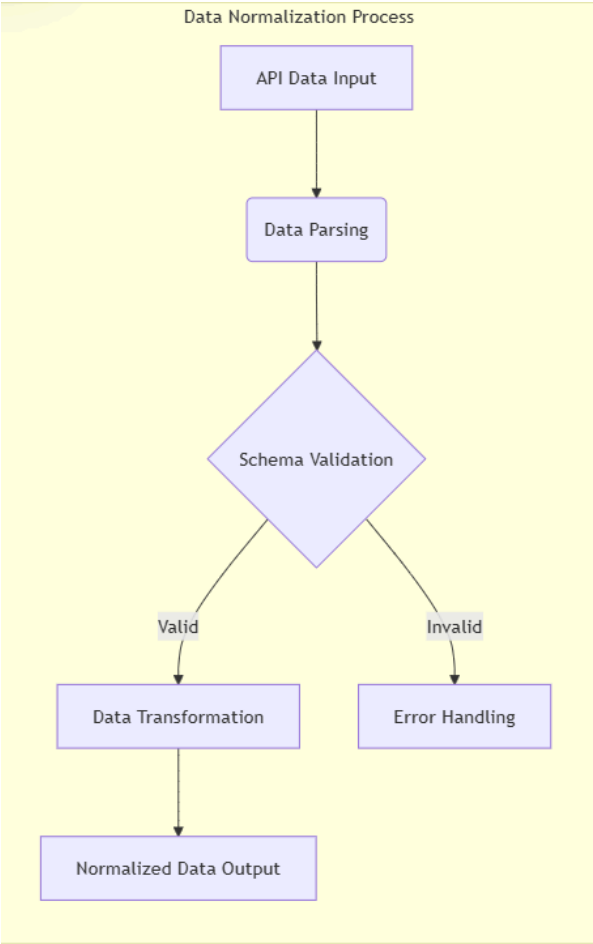
- Handles API requests and processes threat data.
- Retrieves and stores threat intelligence in **MongoDB Atlas**.
- Implements **role-based access control (RBAC)**.

5.1.3 Database (MongoDB Atlas)

- Stores aggregated and normalized threat data.
- Maintains user authentication and access control.

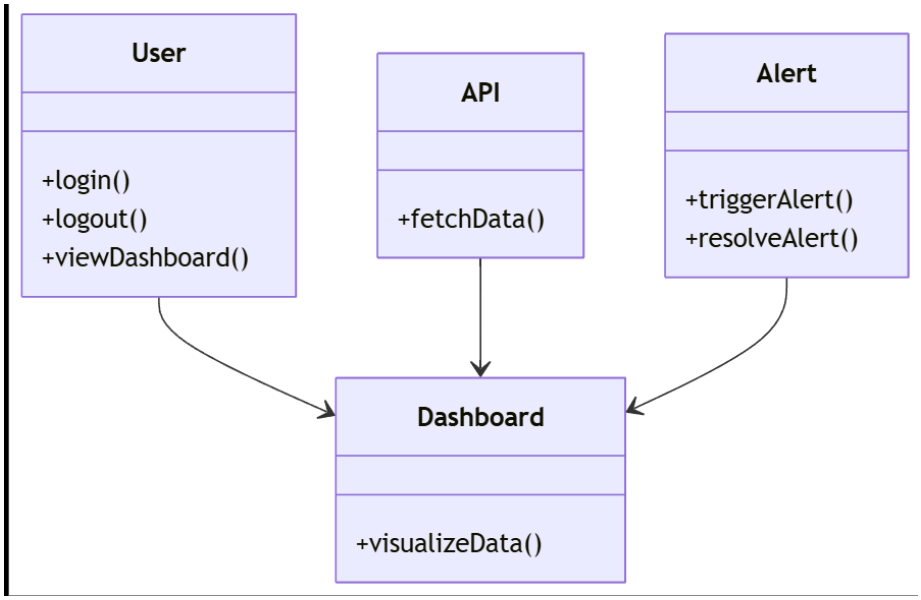
5.2 Data Normalization

The system normalizes **raw threat intelligence data** into a structured schema.



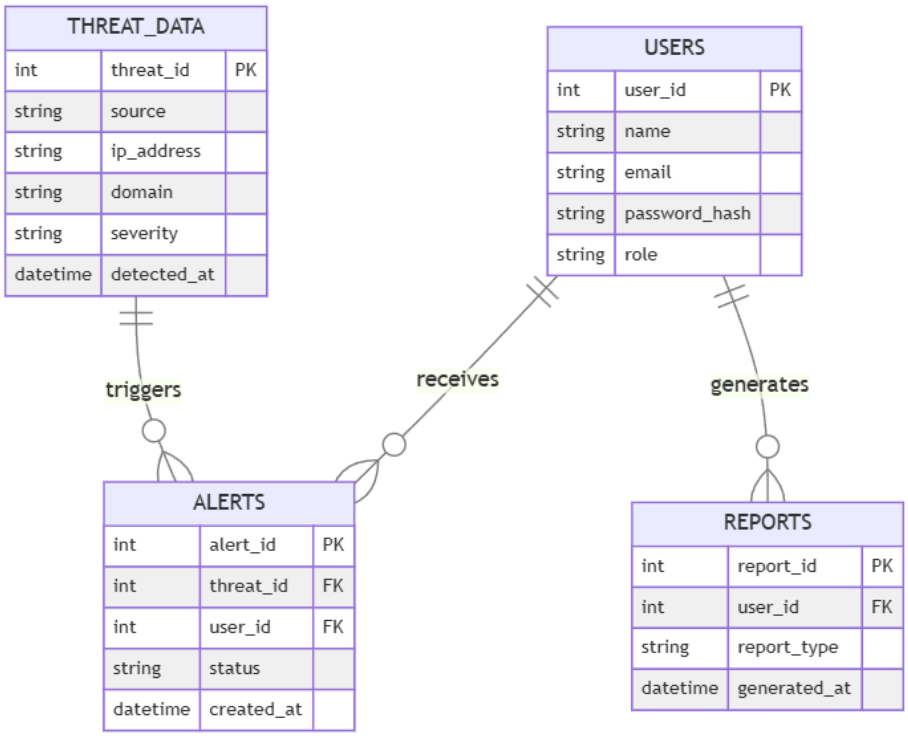
5.3 Class Design

Defines the **object-oriented structure** of the system.



5.4 ER Diagram

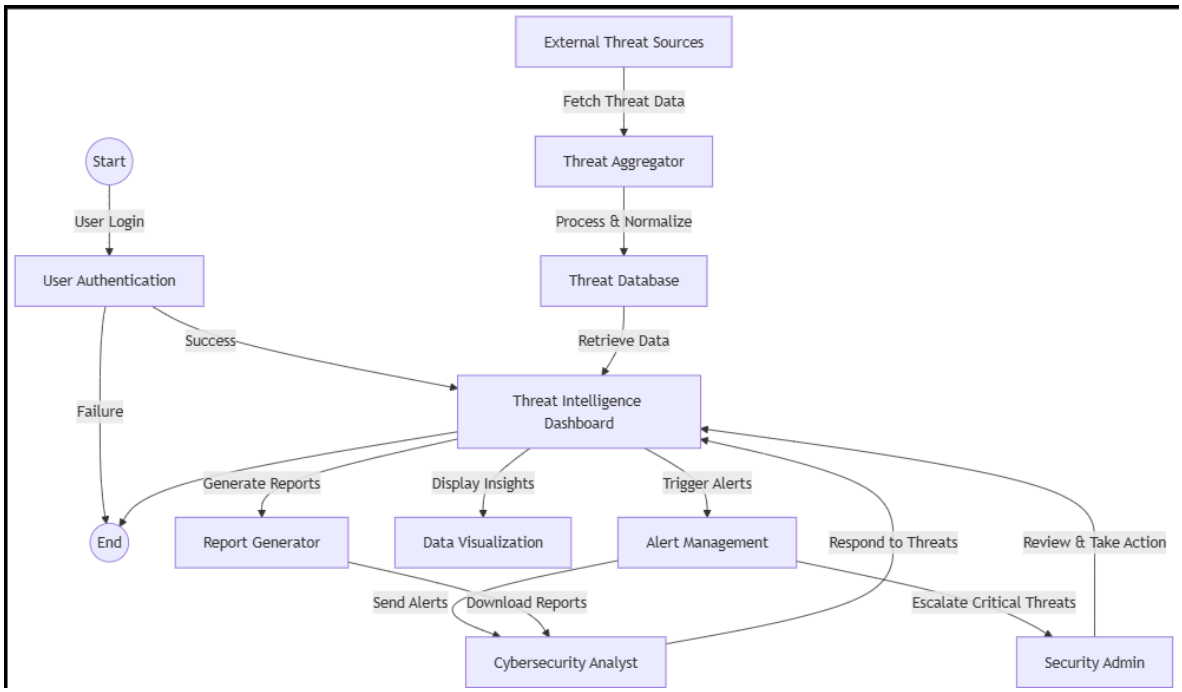
The **ER Diagram** represents the **database schema** of the **Cloud-Based Threat Intelligence Dashboard**. It defines the **entities**, their **attributes**, and the **relationships** between them.



6. Process Flow

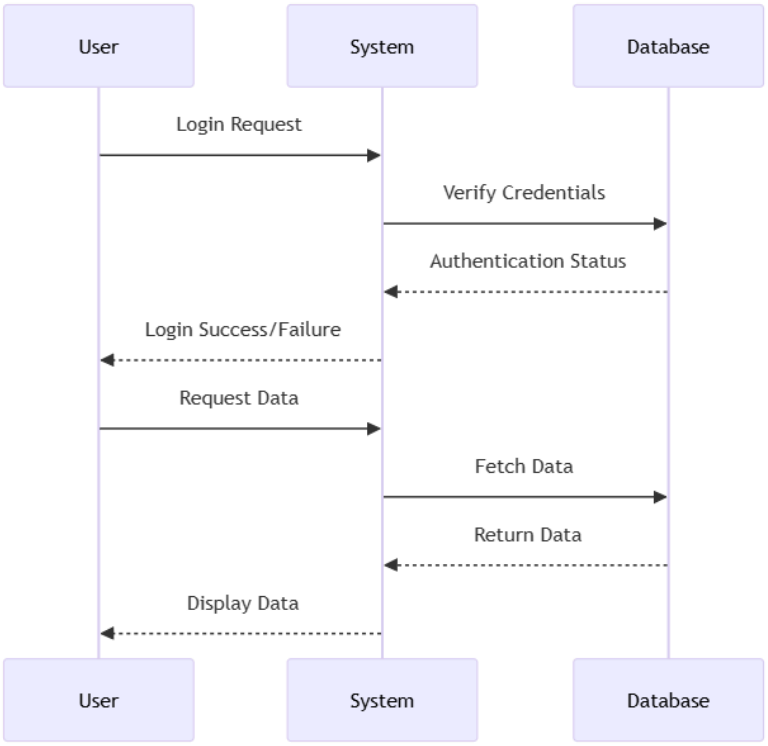
6.1 Control Flow

Illustrates the **logical flow of control** between system components.



6.2 Sequence Flow

Depicts the **order of interactions** between users, the system, and external APIs.

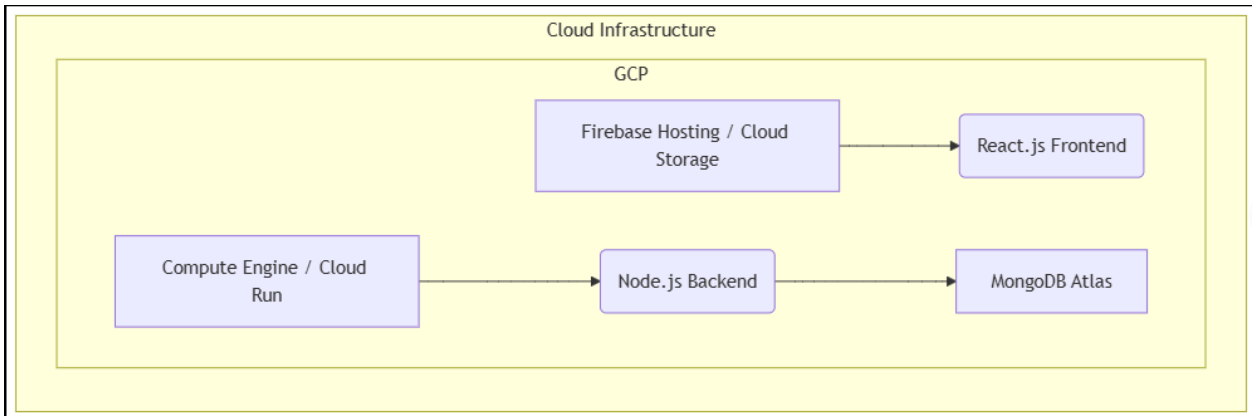


---

7. Deployment Design

7.1 Deployment Model

The system will be deployed in a **cloud-native environment**.

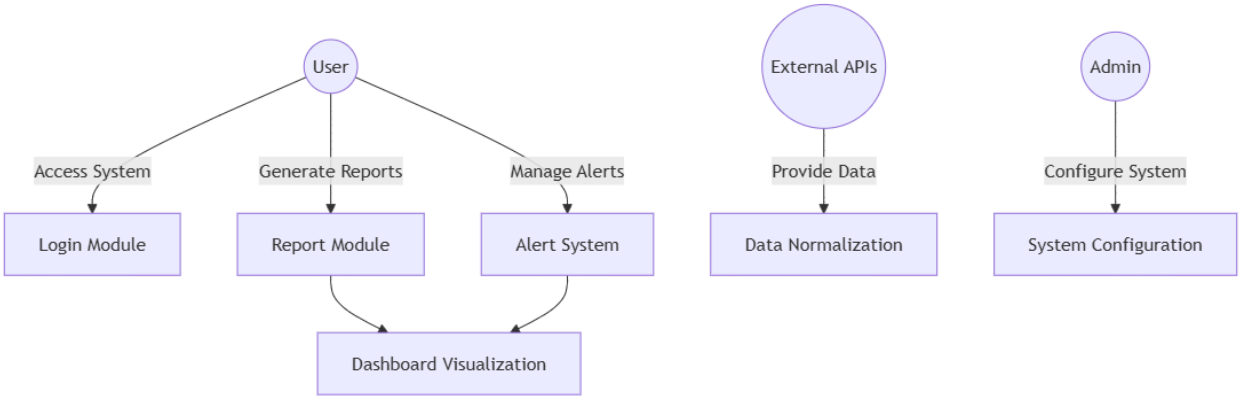


---

8. Behavioral Models

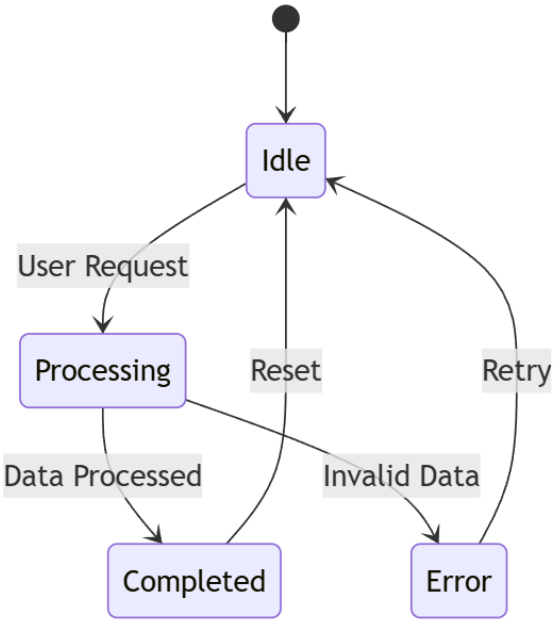
8.1 Use Case Model

Defines **how users interact** with the system.



8.2 State Model

Describes how the **system transitions between different states**.



---

9. Security and Performance Considerations

9.1 Security

- **All data transmission** must be encrypted using **HTTPS**.

- **Users must be authenticated** via a **secure login system**.
- **Threat intelligence data** should be sanitized before storage.

9.2 Performance

- The system must process **threat intelligence data within 5 seconds**.
- **Database indexing and caching** should be used for optimization.
- The frontend must be **responsive and mobile-friendly**.

---

10. Conclusion

This **Software Design Document (SDD)** defines the **architecture, data flow, components, and security measures** of the **Cloud-Based Threat Intelligence Dashboard**. It ensures a well-structured development approach that aligns with our **SRS requirements**.
