SOFTWARE REQUIREMENTS SPECIFICATION

Crisis Management System

For College Campus Emergency Response

Version 4.0 - IEEE 830 Compliant

December 5, 2025

Prepared By:

Arjun Thilak
Anish Sriram B
Gunvant Rao

Revision History

Version	Date	Description	Author
1.0	Oct 15, 2025	Initial draft	Team
2.0	Nov 10, 2025	Added UML diagrams	Team
3.1	Dec 2, 2025	Concise edition	Team
4.0	Dec 5, 2025	IEEE 830 compliance	Team

Contents

Chapter 1

Introduction

1.1 Purpose

This Software Requirements Specification (SRS) describes the Crisis Management System for handling emergency situations in college campuses. The system provides automated crisis detection, intelligent analysis, and coordinated alert distribution.

Intended Audience:

- Software Engineering students and faculty
- Project stakeholders and administrators
- Future developers and system maintainers
- Campus security and emergency response personnel

1.2 Scope

Product Name: Crisis Management System (CMS)

What the system will do:

- Accept crisis complaints from students, faculty, and staff
- Automatically detect and filter spam using machine learning
- Classify crises by type and severity
- Send targeted alerts via multiple channels (SMS, calls, push notifications)
- Integrate with IoT sensors for automated detection
- Provide analytics dashboard for monitoring and reporting

What the system will NOT do:

- Replace 911 or external emergency services
- Provide medical diagnosis or treatment advice
- Control emergency equipment (sprinklers, locks, etc.)
- Store medical records or HIPAA-protected data

Benefits: Reduce emergency response time, improve coordination between stakeholders, and enable data-driven safety improvements.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Definitions

Crisis An emergency situation requiring immediate response

Complaint User-submitted report of a crisis incident

Alert Automated notification sent to relevant parties

Spam False, duplicate, or malicious complaints

1.3.2 Acronyms and Abbreviations

Term	Definition
2FA	Two-Factor Authentication
API	Application Programming Interface
GPS	Global Positioning System
IoT	Internet of Things
JSON	JavaScript Object Notation
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
OWASP	Open Web Application Security Project
SMS	Short Message Service
SRS	Software Requirements Specification
UML	Unified Modeling Language
WCAG	Web Content Accessibility Guidelines

1.4 References

1. IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*
2. ISO/IEC 25010:2011, *Systems and software Quality Requirements and Evaluation*
3. WCAG 2.1, *Web Content Accessibility Guidelines*, W3C
4. OWASP Top 10, *Web Application Security Risks*
5. ISO 22320:2018, *Security and resilience - Emergency management*

1.5 Overview

This document is organized into five chapters:

- **Chapter 1:** Introduction, scope, and definitions
- **Chapter 2:** Overall system description and context
- **Chapter 3:** Detailed functional and non-functional requirements
- **Chapter 4:** UML diagrams showing system structure and behavior

- **Chapter 5:** Summary and conclusions

Requirements use "shall" for mandatory and "should" for recommended features. Each requirement includes a unique identifier (e.g., FR-UM-01) and priority level [ESSENTIAL], [CONDITIONAL], or [OPTIONAL].

Chapter 2

Overall Description

2.1 Product Perspective

The Crisis Management System is a new, self-contained system designed to integrate with existing campus infrastructure. It operates independently but interfaces with:

- **User Devices:** Web browsers and mobile apps (iOS/Android)
- **IoT Sensors:** Fire detectors, security cameras, environmental monitors
- **External Services:** SMS/call providers, push notification services
- **Database Systems:** For storing complaints, user data, and analytics

The system consists of two main subsystems:

1. **Interface Application:** Handles user interactions and alert distribution
2. **Analysis Application:** Performs ML-based spam detection and classification

2.2 Product Functions

The system provides seven major function areas:

1. **User Management:** Registration, authentication, and role-based access control for students, faculty, staff, operators, and administrators
2. **Crisis Reporting:** Accept complaints with text descriptions, location coordinates, photos, and automatic metadata capture
3. **Spam Detection:** Analyze complaints using ML models to identify false reports and duplicates
4. **Crisis Classification:** Automatically categorize crises (Fire, Medical, Security, Infrastructure, Natural Disaster, Other) and assign priority levels (1-5)
5. **Alert Distribution:** Send notifications via SMS, phone calls, and push notifications based on crisis severity and user location
6. **IoT Integration:** Receive real-time data from sensors and auto-generate complaints when thresholds are exceeded
7. **Analytics & Reporting:** Provide real-time dashboard, historical trends, and performance metrics

2.3 User Characteristics

2.3.1 Students

- Age: 18-25 years
- Technical skill: Basic to intermediate smartphone usage
- Primary role: Report crises, receive safety alerts
- Expected frequency of use: Occasional (during emergencies)

2.3.2 Faculty and Staff

- Age: 25-70 years
- Technical skill: Basic to intermediate computer/smartphone usage
- Primary role: Report crises, receive alerts, view incident history
- Expected frequency of use: Occasional to moderate

2.3.3 Emergency Operators

- Background: Campus security or safety personnel
- Technical skill: Intermediate (trained on system)
- Primary role: Monitor incoming complaints, coordinate response, update status
- Expected frequency of use: Daily, during work shifts

2.3.4 System Administrators

- Background: IT professionals
- Technical skill: Advanced
- Primary role: System configuration, user management, data analysis
- Expected frequency of use: Regular maintenance and monitoring

2.4 Constraints

2.4.1 Regulatory Constraints

- Must comply with FERPA (Family Educational Rights and Privacy Act)
- Must meet ADA (Americans with Disabilities Act) accessibility requirements
- Should follow GDPR principles for data privacy (if applicable)

2.4.2 Technical Constraints

- Must work on mobile devices with limited processing power
- Must function with intermittent network connectivity
- Must interface with legacy database systems

2.4.3 Operational Constraints

- System must operate 24/7 with minimal downtime
- Must support peak loads during simultaneous emergencies
- Must maintain audit trails for all critical operations

2.5 Assumptions and Dependencies

2.5.1 Assumptions

1. Campus has reliable cellular and Wi-Fi network coverage
2. At least 80% of students/faculty own smartphones
3. Emergency operators are available 24/7
4. Users have valid email addresses and phone numbers
5. Campus has existing IoT sensor infrastructure (or will deploy)

2.5.2 Dependencies

1. Third-party SMS/call service providers must be operational
2. Push notification services must be available
3. ML models require training data from historical incidents
4. IoT sensors must follow MQTT communication protocol
5. Database system must support real-time queries

2.6 Apportioning of Requirements

Requirements are divided into three implementation phases:

2.6.1 Phase 1 - Core System (MVP)

- User registration and authentication
- Basic complaint submission (text + location)
- Manual classification by operators
- SMS alerts to operators
- Simple dashboard
- *Target: End of Q2 2026*

2.6.2 Phase 2 - Intelligent Features

- ML-based spam detection
- Automatic crisis classification
- Multi-channel alerts (SMS + calls + push)
- Photo upload capability
- Advanced analytics
- *Target: End of Q4 2026*

2.6.3 Phase 3 - IoT Integration

- IoT sensor integration
- Automated complaint generation
- Predictive analytics
- *Target: Q2 2027*

Chapter 3

Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- UI-01: System shall provide web interface accessible via standard browsers
- UI-02: System shall provide mobile app for iOS and Android devices
- UI-03: Complaint submission form shall include fields for description, location, and photo upload
- UI-04: Dashboard shall display active crises, response status, and key metrics
- UI-05: Interface shall meet WCAG 2.1 Level AA accessibility standards

3.1.2 Hardware Interfaces

- HI-01: System shall receive GPS coordinates from user devices
- HI-02: System shall accept photo/video uploads from device cameras
- HI-03: System shall interface with IoT sensors via MQTT protocol
- HI-04: IoT sensors shall transmit data in JSON format

3.1.3 Software Interfaces

- SI-01: System shall interface with SMS/call service provider APIs
- SI-02: System shall interface with push notification services
- SI-03: System shall store data in relational or NoSQL database
- SI-04: System shall provide REST APIs for external integration

3.1.4 Communications Interfaces

- CI-01: All data transmission shall use encrypted connections
- CI-02: System shall support HTTP/HTTPS protocols
- CI-03: IoT communication shall use MQTT protocol
- CI-04: Data shall be exchanged in JSON format

3.2 Functional Requirements

3.2.1 User Management (FR-UM)

1. **FR-UM-01:** System shall allow user registration with email, password, name, phone, and role. [ESSENTIAL]
2. **FR-UM-02:** System shall send email verification link upon registration. [ESSENTIAL]
3. **FR-UM-03:** System shall authenticate users with email and password. [ESSENTIAL]
4. **FR-UM-04:** System shall support five user roles: Student, Faculty, Staff, Operator, Administrator. [ESSENTIAL]
5. **FR-UM-05:** System shall enforce password requirements: minimum 8 characters, mixed case, numbers, special characters. [ESSENTIAL]
6. **FR-UM-06:** System shall support two-factor authentication for operators and administrators. [CONDITIONAL]
7. **FR-UM-07:** System shall allow users to update profile information. [ESSENTIAL]
8. **FR-UM-08:** System shall allow users to request account deletion. [CONDITIONAL]

3.2.2 Complaint Management (FR-CM)

1. **FR-CM-01:** System shall accept complaints with description (10-500 characters), GPS location, and optional photo. [ESSENTIAL]
2. **FR-CM-02:** System shall validate complaint data before submission. [ESSENTIAL]
3. **FR-CM-03:** System shall assign unique identifier to each complaint. [ESSENTIAL]
4. **FR-CM-04:** System shall track complaint status: Submitted, Analyzing, Classified, Alerting, In Progress, Resolved, Rejected. [ESSENTIAL]
5. **FR-CM-05:** System shall allow operators to add notes to complaints. [CONDITIONAL]
6. **FR-CM-06:** System shall allow operators to manually update complaint status. [ESSENTIAL]
7. **FR-CM-07:** System shall prevent duplicate submissions within 2 minutes from same user. [CONDITIONAL]

3.2.3 Spam Detection (FR-SD)

1. **FR-SD-01:** System shall analyze complaints using machine learning models to detect spam. [ESSENTIAL]
2. **FR-SD-02:** System shall detect duplicate complaints within 15-minute window with 90%+ similarity. [CONDITIONAL]

3. **FR-SD-03:** System shall provide confidence score (0-1) for spam detection. [ESSENTIAL]
4. **FR-SD-04:** System shall achieve minimum 95% spam detection accuracy. [ESSENTIAL]
5. **FR-SD-05:** System shall allow administrators to review and override spam classifications. [CONDITIONAL]

3.2.4 Crisis Classification (FR-CC)

1. **FR-CC-01:** System shall classify complaints into 6 categories: Fire, Medical, Security Threat, Infrastructure, Natural Disaster, Other. [ESSENTIAL]
2. **FR-CC-02:** System shall assign priority level (1-5 scale, 5 being highest) based on severity. [ESSENTIAL]
3. **FR-CC-03:** System shall achieve minimum 92% classification accuracy. [ESSENTIAL]
4. **FR-CC-04:** System shall allow manual reclassification by operators. [CONDITIONAL]
5. **FR-CC-05:** System shall provide confidence scores for classifications. [CONDITIONAL]

3.2.5 Alert Distribution (FR-AD)

1. **FR-AD-01:** System shall send SMS alerts to users within configurable radius (100-500m) of crisis location. [ESSENTIAL]
2. **FR-AD-02:** System shall make automated phone calls to emergency operators for Priority 4-5 crises. [CONDITIONAL]
3. **FR-AD-03:** System shall send push notifications to mobile app users. [ESSENTIAL]
4. **FR-AD-04:** SMS alerts shall be limited to 160 characters. [ESSENTIAL]
5. **FR-AD-05:** System shall log all alert delivery attempts and confirmations. [ESSENTIAL]
6. **FR-AD-06:** System shall retry failed alert deliveries up to 3 times. [CONDITIONAL]
7. **FR-AD-07:** System shall allow users to opt-out of non-critical alerts. [OPTIONAL]

3.2.6 IoT Integration (FR-IOT)

1. **FR-IOT-01:** System shall receive real-time data from IoT sensors via MQTT protocol. [CONDITIONAL]
2. **FR-IOT-02:** System shall auto-generate complaints when sensor readings exceed thresholds (e.g., smoke >50ppm, temperature >150°F). [CONDITIONAL]
3. **FR-IOT-03:** System shall process sensor data with less than 2 seconds latency. [CONDITIONAL]

4. **FR-IOT-04:** System shall monitor sensor health status and alert on failures. [OPTIONAL]

3.2.7 Analytics and Reporting (FR-AR)

1. **FR-AR-01:** System shall provide real-time dashboard showing active crises and system status. [ESSENTIAL]
2. **FR-AR-02:** Dashboard shall display KPIs: average response time, resolution rate, alert delivery success rate. [ESSENTIAL]
3. **FR-AR-03:** System shall generate historical reports with trend analysis. [CONDITIONAL]
4. **FR-AR-04:** System shall support data export in CSV, PDF, and JSON formats. [CONDITIONAL]
5. **FR-AR-05:** System shall provide filtering and search capabilities for complaints. [ESSENTIAL]

3.3 Performance Requirements

1. **NFR-P-01:** System shall process complaint submissions within 2 seconds for 1000 concurrent users. [ESSENTIAL]
2. **NFR-P-02:** ML analysis (spam + classification) shall complete within 5 seconds. [ESSENTIAL]
3. **NFR-P-03:** Alert delivery shall be initiated within 10 seconds of crisis confirmation. [ESSENTIAL]
4. **NFR-P-04:** Dashboard shall update in real-time with less than 2-second latency. [CONDITIONAL]
5. **NFR-P-05:** System shall handle 10,000 complaints per day without performance degradation. [CONDITIONAL]

3.4 Logical Database Requirements

1. **DBR-01:** Database shall store user profiles, complaints, analysis reports, alerts, and sensor data. [ESSENTIAL]
2. **DBR-02:** Database shall maintain referential integrity between related entities (users, complaints, alerts). [ESSENTIAL]
3. **DBR-03:** Complaint data shall be retained indefinitely; personal data shall be anonymized after 7 years per FERPA. [ESSENTIAL]
4. **DBR-04:** Database queries shall return results within 100ms for 95% of requests. [CONDITIONAL]
5. **DBR-05:** Database shall support point-in-time recovery with less than 1-hour data loss. [CONDITIONAL]

6. **DBR-06:** Database shall enforce unique constraints on user email addresses and phone numbers. [ESSENTIAL]

3.5 Design Constraints

1. **DC-01:** System shall comply with IEEE 830-1998 for requirements specification. [ESSENTIAL]
2. **DC-02:** System shall follow ISO 22320:2018 for emergency management. [CONDITIONAL]
3. **DC-03:** User interfaces shall comply with WCAG 2.1 Level AA accessibility standards. [ESSENTIAL]
4. **DC-04:** Security shall align with OWASP Top 10 best practices. [ESSENTIAL]
5. **DC-05:** System shall use RESTful API design for external interfaces. [CONDITIONAL]
6. **DC-06:** System shall follow modular architecture for maintainability. [CONDITIONAL]

3.6 Software System Attributes

3.6.1 Reliability

- System shall maintain 99.9% uptime (less than 8.76 hours downtime per year)
- System shall implement automatic failover with less than 10 seconds recovery time
- System shall perform daily backups with recovery time objective (RTO) less than 1 hour

3.6.2 Availability

- System shall operate 24/7/365
- System shall support zero-downtime deployments for updates
- System shall implement load balancing for high availability

3.6.3 Security

- System shall encrypt data in transit using industry-standard protocols
- System shall encrypt sensitive data at rest
- System shall implement session timeout after 30 minutes of inactivity
- System shall enforce rate limiting to prevent abuse: 5 login attempts per 15 minutes, 10 complaints per hour per user, 100 API calls per minute
- System shall maintain audit logs of all security-relevant events for 180 days

3.6.4 Maintainability

- System shall use modular architecture for easy updates
- System shall provide comprehensive API documentation
- Code shall maintain 80%+ test coverage
- System shall support automated deployment

3.6.5 Usability

- New users shall require less than 30 minutes of training for basic operations
- Crisis complaint submission shall require maximum 3 clicks/taps
- System shall provide clear error messages and recovery instructions
- System shall support keyboard navigation for accessibility
- System shall maintain 4.5:1 minimum color contrast ratio for text

3.6.6 Portability

- System shall be deployable on multiple platforms (Windows, Linux, macOS servers)
- System shall support multiple browsers (Chrome, Firefox, Safari, Edge)
- Mobile apps shall support iOS 14+ and Android 10+
- System shall use standard protocols to ensure interoperability

Chapter 4

UML Diagrams and System Models

This chapter presents UML diagrams illustrating the structure, behavior, and deployment of the Crisis Management System. These diagrams provide multiple perspectives on the system design.

4.1 Data Flow Diagram

Shows high-level data movement between users, system components, and external services.

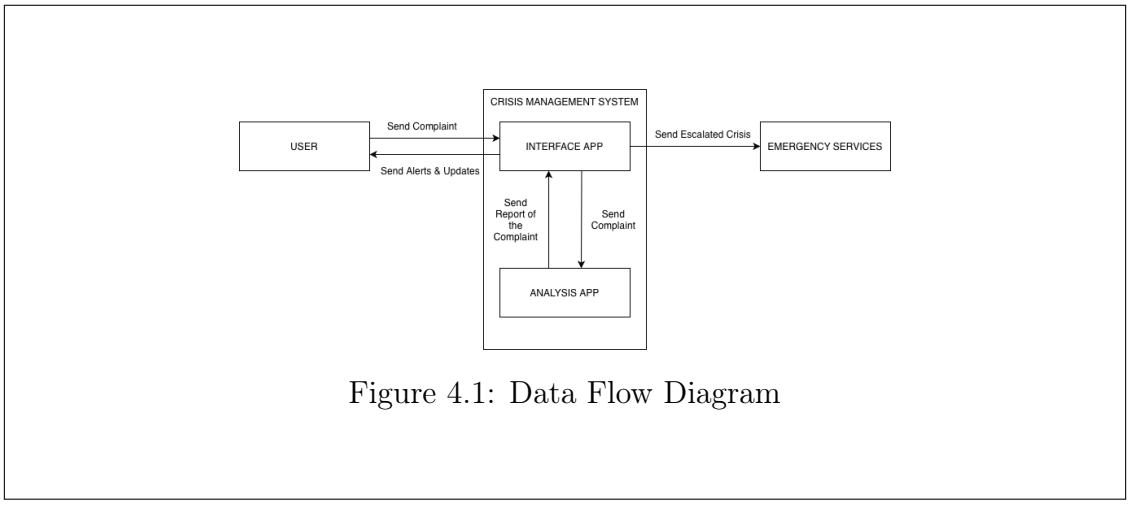

Figure 4.1: Data Flow Diagram

4.2 Package Diagram

Shows the modular structure with Interface and Analysis application packages.

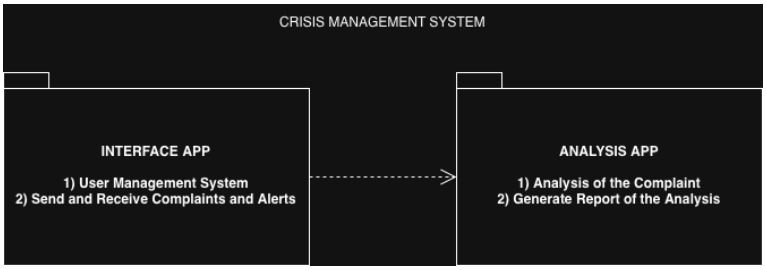

Figure 4.2: Package Diagram

4.3 Use Case Diagrams

Illustrate actor interactions with system functions.

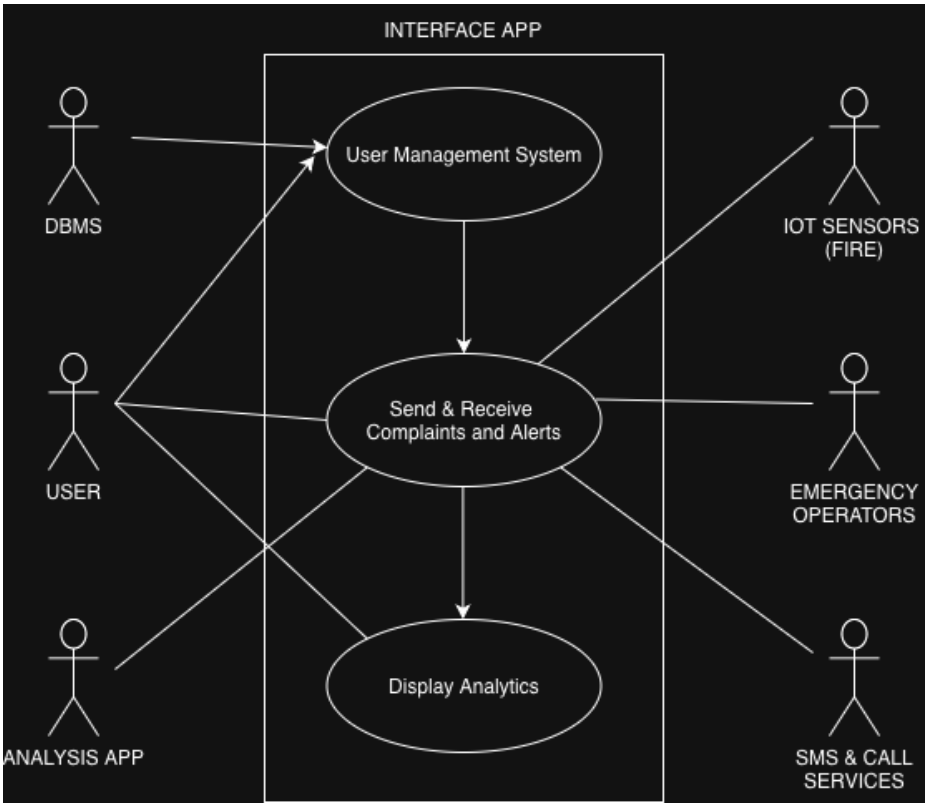

Figure 4.3: Use Case Diagram - Interface Application

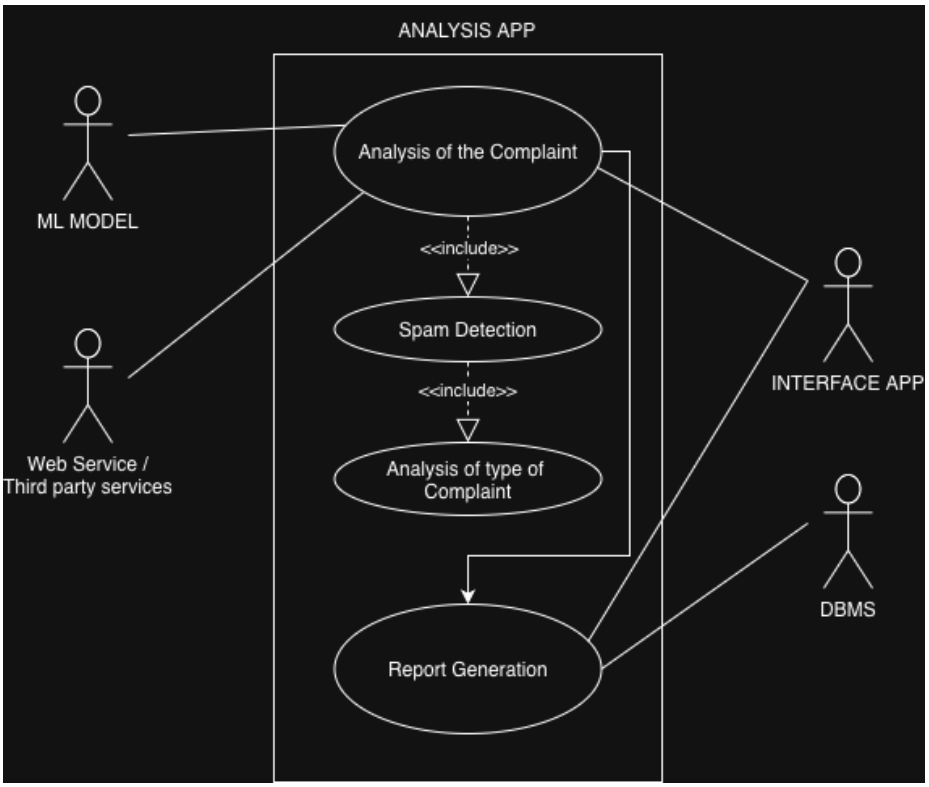

Figure 4.4: Use Case Diagram - Analysis Application

4.4 Class Diagram

Shows the main entities, their attributes, and relationships.

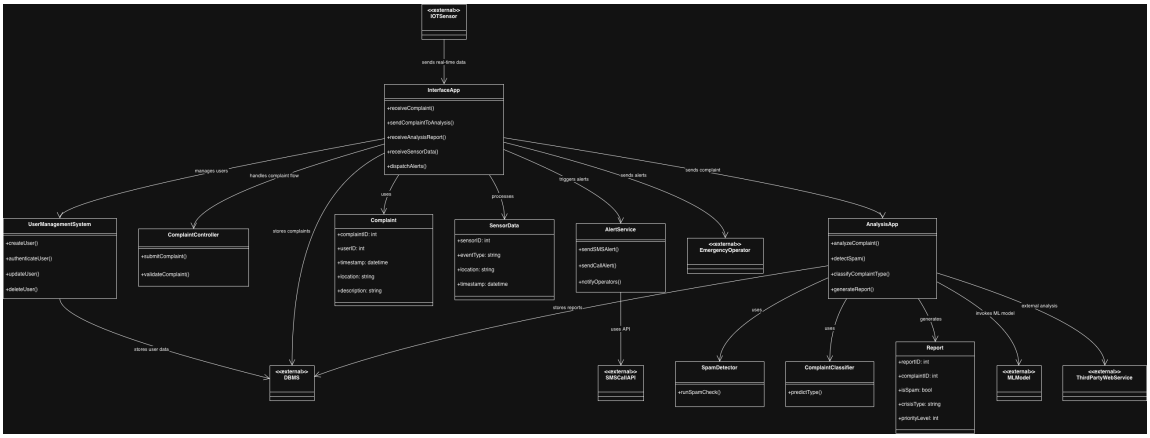

Figure 4.5: Class Diagram

4.5 Sequence Diagram

Illustrates the time-ordered interactions during complaint processing.

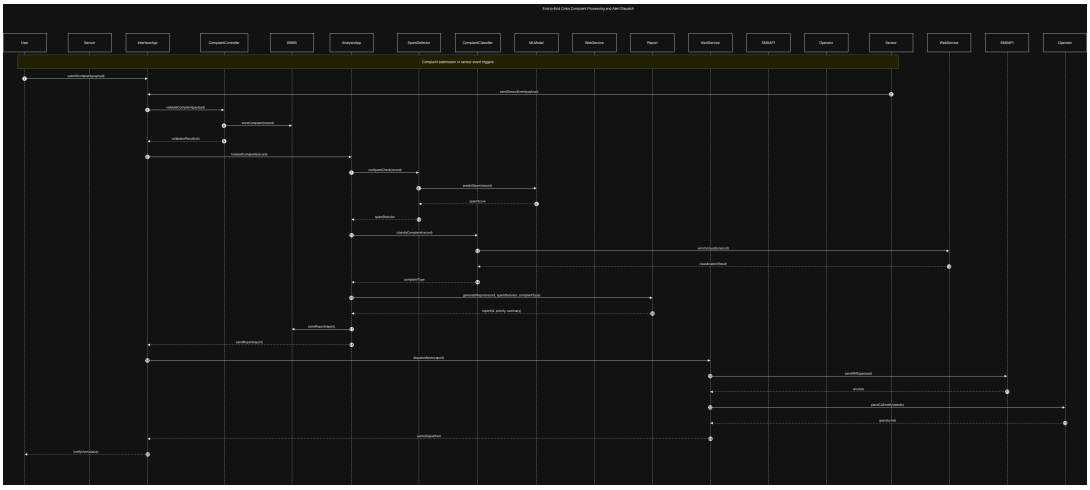

Figure 4.6: Sequence Diagram

4.6 Activity Diagram

Shows the workflow with decision points and parallel activities.

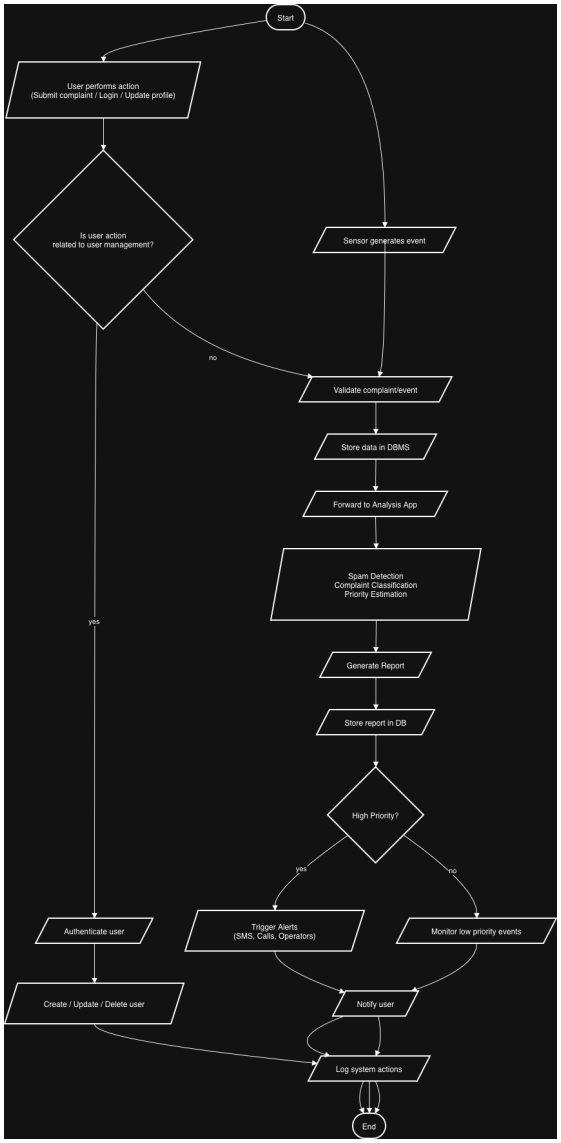

Figure 4.7: Activity Diagram

4.7 State Diagram

Shows complaint lifecycle states and transitions.

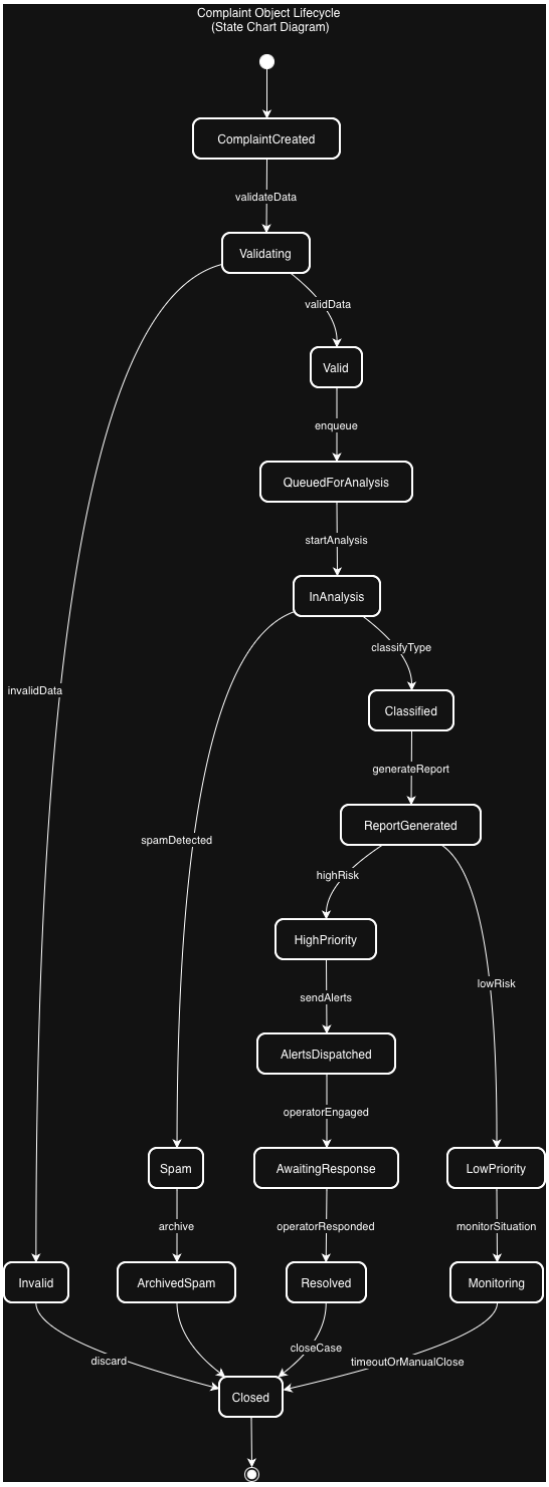

Figure 4.8: State Diagram

4.8 Component Diagram

Shows runtime components and their dependencies.

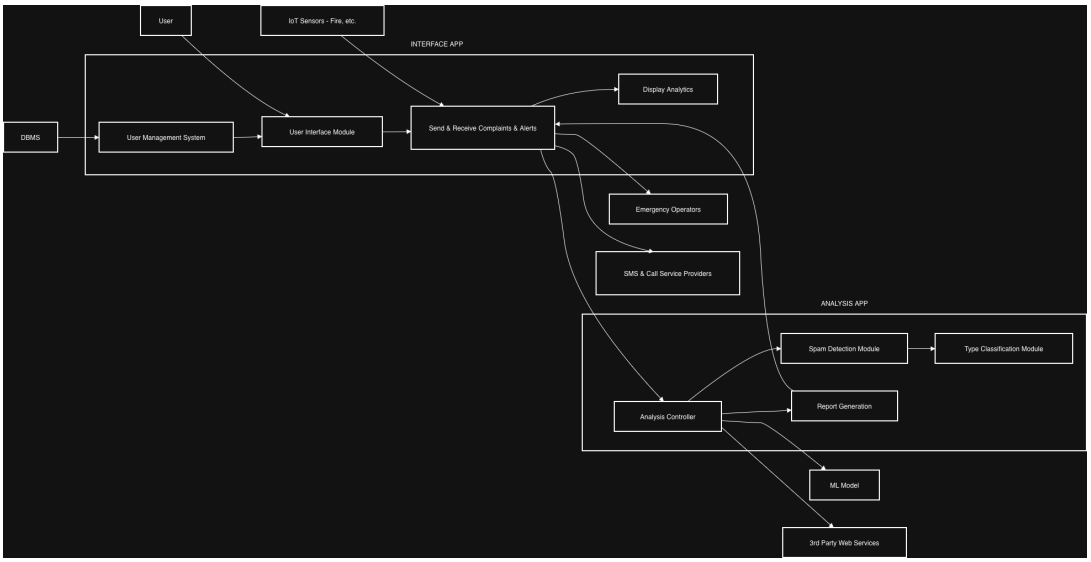

Figure 4.9: Component Diagram

4.9 Deployment Diagram

Shows physical system deployment across hardware nodes.

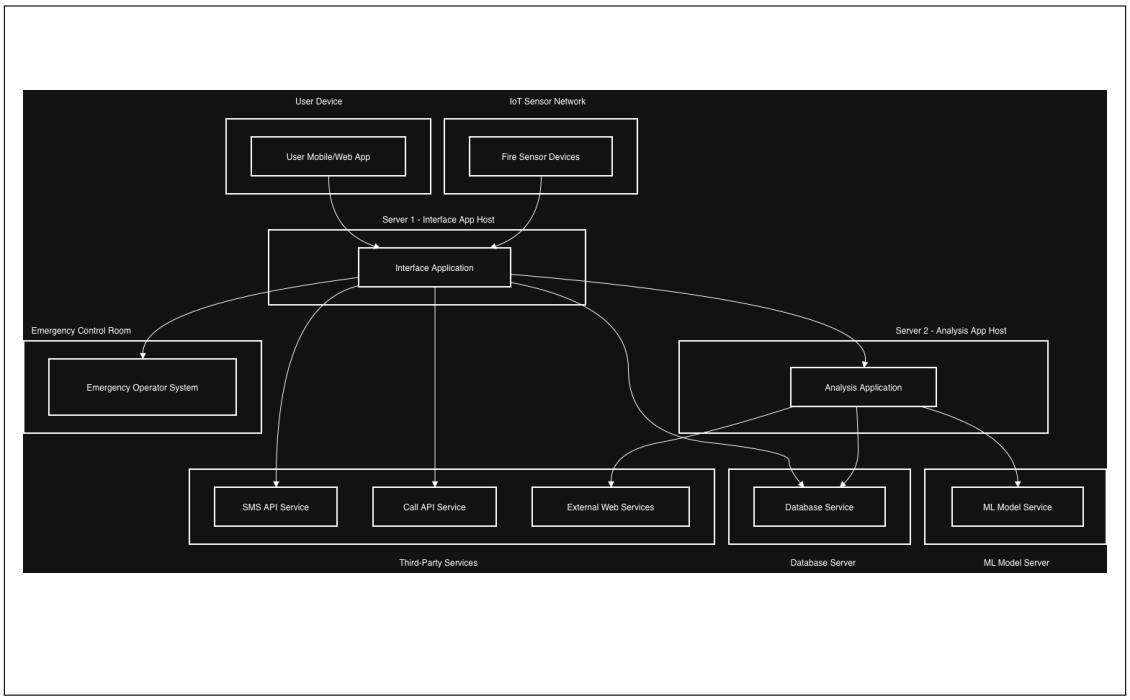

Figure 4.10: Deployment Diagram

4.10 Entity-Relationship Diagram

Shows database structure with entities and relationships.

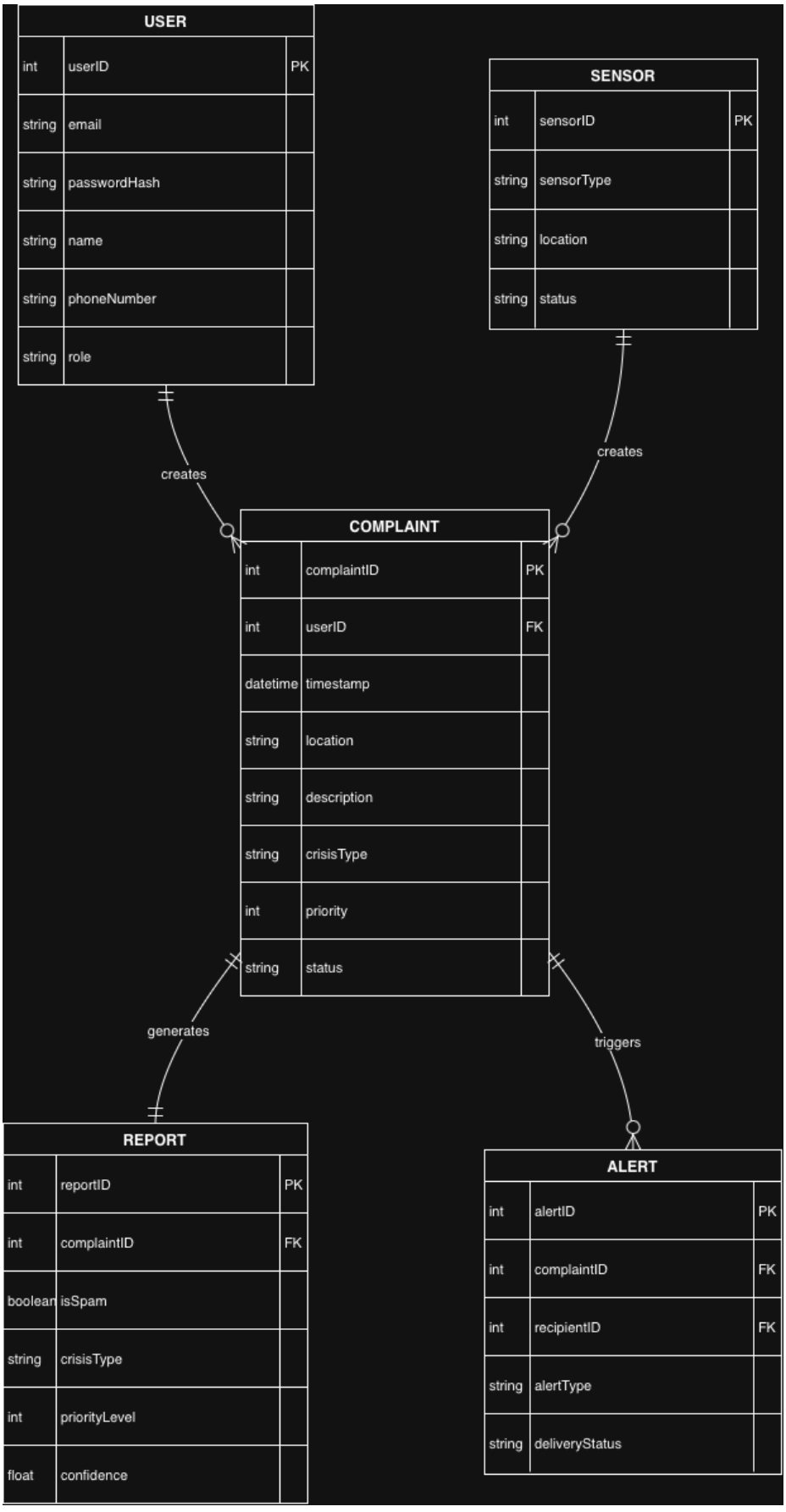
USER
int userID PK
string email
string passwordHash
string name
string phoneNumber
string role
SENSOR
int sensorID PK
string sensorType
string location
string status
creates
creates
COMPLAINT
int complaintID PK
int userID FK
datetime timestamp
string location
string description
string crisisType
int priority
string status
generates
triggers
REPORT
int reportID PK
int complaintID FK
boolean isSpam
string crisisType
int priorityLevel
float confidence
ALERT
int alertID PK
int complaintID FK
int recipientID FK
string alertType
string deliveryStatus

Figure 4.11: Entity-Relationship Diagram

Chapter 5

Conclusion

5.1 Document Summary

This Software Requirements Specification provides a comprehensive definition of the Crisis Management System for college campuses. The document follows IEEE 830-1998 standards and includes detailed requirements, UML diagrams, and system specifications.

5.2 Key Features

The Crisis Management System provides:

- **Automated Detection:** ML-based spam filtering and crisis classification
- **Multi-Channel Alerts:** SMS, voice calls, and push notifications
- **IoT Integration:** Automated crisis detection via sensors
- **Real-Time Analytics:** Dashboard with KPIs and historical trends
- **Role-Based Access:** Support for students, faculty, operators, and administrators

5.3 Requirements Coverage

This SRS defines:

- 40+ functional requirements organized by feature area
- 15+ non-functional requirements covering performance, security, and usability
- 6 database requirements for data management
- 6 design constraints for standards compliance
- External interface specifications for UI, hardware, software, and communications

5.4 UML Diagrams

The document includes 11 UML diagrams providing multiple views:

- Structural: Package, Class, Component, Deployment, ER diagrams

- Behavioral: Use Case, Sequence, Activity, State diagrams
- Data Flow: DFD showing system-level information flow

5.5 Standards Compliance

The system adheres to:

- IEEE 830-1998 for software requirements specifications
- ISO 22320:2018 for emergency management
- WCAG 2.1 Level AA for accessibility
- OWASP Top 10 for security

5.6 Implementation Approach

The system will be implemented in three phases:

1. **Phase 1:** Core functionality (user management, basic complaint handling, SMS alerts)
2. **Phase 2:** Intelligent features (ML models, multi-channel alerts, advanced analytics)
3. **Phase 3:** IoT integration and predictive capabilities

5.7 Expected Impact

Successful implementation will:

- Reduce emergency response time from 15+ minutes to under 5 minutes
- Improve coordination between campus stakeholders
- Enable data-driven safety improvements
- Enhance campus safety culture through increased awareness

5.8 Final Remarks

This SRS serves as the foundation for system design, development, and testing. It provides clear, verifiable requirements that can be validated against stakeholder needs and tested during system implementation.

End of Software Requirements Specification