INTELLIGENT VIDEO SURVIALLENCE SYSTEM

SYSTEM REQUIRMENT SPECIFICATION

EDITION ONE

Software Engineering

DEPARTEMENT OF COMPUTER SCIENCE
(RACHNA COLLEGE OF ENGINEERING AND TECHNOLOGY,
GUJRANWALLAH

Prepared for
CS 365-L —Software Engineering
Instructor: Dr. Natasha Nigar, Ph.D.
Fall 2023

DATED:Sunday, November 12, 2023

Submitted By:

01
Daniyal Tariq
21-CS-406

02
Suleman Ayoob
21-CS-416

03
Saqib Iqbal
21-CS-424

NOTE: Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984).

Revision History

Date	Description	Author	Comments
2023-11-12	Version 1	Daniyal Tariq	Requirements can be change in 2nd version of SRS

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Daniyal Tariq Suleman Ayoob Saqib Iqbal	Frontend Developer Architecture Developer Backend Developer	2023-11-12
	Dr. Natasha Nigar	Instructor, CS 365-L	2023-11-12

Table of Contents

1. Introduction

This Software Requirements Specification (SRS) document provides a comprehensive overview of the requirements, design, and functionality of the "Intelligent Video Surveillance System." The purpose of this document is to serve as a complete reference for software engineers, developers, and other stakeholders involved in the design, development, and implementation of the software product.

1.1 Purpose

The primary purpose of this SRS document is to define and describe the requirements for the "Intelligent Video Surveillance System." It aims to establish a clear and detailed understanding of the software's capabilities, features, and interactions, enabling software engineers to develop a system that meets these specifications.

1.2 Scope

1.2.1 Software Product

The software product to be produced is the "Intelligent Video Surveillance System." This system is designed to enhance security and safety measures through advanced video analysis and intelligent incident detection. It proactively identifies various incidents, such as human falls, vehicle crashes, and social distancing violations, in real-time.

1.2.2 Software Product Description

The "Intelligent Video Surveillance System" will:

- Provide real-time incident detection and alerting.
- Utilize YOLO-based object detection algorithms to classify and identify incidents.
- Support various video input sources, including webcams, local storage, online videos, and live Android camera feeds.
- Feature a user-friendly web-based application with a Python backend and React frontend.
- Incorporate an email notification system for immediate alerts to authorities in case of incidents.
- Address the evolving needs for security and safety in diverse settings, including businesses, institutions, and communities.
- The software product will not: Address non-security-related applications, as its primary focus is on intelligent video surveillance for security and safety enhancement.

1.2.3 Application of the Software

The "Intelligent Video Surveillance System" will be applied to the following areas:

1.2.3.1 Consistency with Higher-Level Specifications

- *Enhance security and safety measures.*
- *Improve incident response times.*
- *Provide real-time incident detection and alerting.*
- *Revolutionize the way organizations monitor and secure their premises.*
- *Position the product as an innovator in the field of intelligent video surveillance.*

- *Ensure flexibility in video input sources and support multiple devices.*
- *Meet market demands for proactive security measures and advanced video analytics.*
- *Comply with evolving regulatory requirements for safety and security.*
- *Enable remote surveillance capabilities for monitoring and responding to incidents.*

1.2.3.2 Consistency with Higher-Level Specifications

This scope is consistent with similar statements in higher-level specifications, such as the System Requirement Specification (SRS), if they exist. The objectives and goals outlined in this document align with the broader objectives of providing an advanced video surveillance solution and addressing the identified business problem.

1.3 Definitions, Acronyms, and Abbreviations

- **IVSS**: Intelligent Video Surveillance System
- **SRS**: Software Requirement Specification
- **YOLO**: You Only Look Once (object detection technology)
- **UI**: User Interface
- **API**: Application Programming Interface
- **HTML**: Hypertext Markup Language
- **CSS**: Cascading Style Sheets
- **IoT**: Internet of Things
- **GUI**: Graphical User Interface
- **URL**: Uniform Resource Locator
- **SMTP**: Simple Mail Transfer Protocol
- **SQL**: Structured Query Language
- **RDBMS**: Relational Database Management System
- **SOP**: Standard Operating Procedure
- **CPU**: Central Processing Unit
- **GPU**: Graphics Processing Unit

1.4 References

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv preprint *arXiv:1506.02640*.
2. Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. arXiv preprint *arXiv:1612.08242*.
3. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint *arXiv:1804.02767*.
4. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint *arXiv:2004.10934*.
5. Zhang, D., Qiao, Y., Chen, D., & Liu, D. (2005). Visual Surveillance for Moving Object Detection--An Overview. Visual Surveillance and Sensor Networks, 2, 14. *https://www.mdpi.com/1424-8220/16/10/1758*.
6. Chen, T. H., Chen, H. W., & Chao, W. C. (2012). Crowd Disturbance Detection in Video Surveillance. In 2012 Fourth International Conference on Multimedia Information

Networking and Security (MINES) (pp. 125-129). IEEE. https://ieeexplore.ieee.org/ielaam/5962385/8392548/7949100-aam.pdf.

7. Chakraborty, A., & Das, D. (2020). A machine learning approach to detect social distancing violation in COVID-19 videos. Journal of Ambient Intelligence and Humanized Computing, 1-9. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/.
8. Wang, X., Cong, Y., Lin, S., & Lu, H. (2021). Human Detection in Surveillance Videos Using YOLOv3 and Deep Sort. Electronics, 10(2), 175. https://asp-eurasipjournals.springeropen.com/articles/10.1186/1687-6180-2013-176.

1.5 Overview

This section provides an overview of the Software Requirement Specification (SRS) document, describing its contents and organization:

1.5.1 Description of the SRS Content

The SRS document outlines the detailed requirements and specifications for the "Intelligent Video Surveillance System (IVSS)." It provides comprehensive information to guide software engineers in designing and implementing the software product. The document covers a wide range of topics, including the project's scope, specific requirements, analysis models, and more, to ensure clarity and completeness in the software development process.

1.5.2 Organization of the SRS

The SRS document is structured into several sections to facilitate navigation and understanding. The document begins with an introduction, followed by sections outlining the scope, definitions, references, and an overview. The core of the document is dedicated to defining the general and specific requirements of the software product, encompassing external interfaces, functional and non-functional requirements, design constraints, and more. Additionally, analysis models are presented to enhance comprehension. The document concludes with a section on the change management process and a series of appendices containing supplementary information.

2. General Description

This section provides a general description of the factors influencing the product and its requirements. It offers an understanding of the product's context, functions, user characteristics, constraints, assumptions, and dependencies.

2.1 Product Perspective

The "Intelligent Video Surveillance System (IVSS)" is situated within the context of related projects and systems. It operates as a stand-alone software solution designed to enhance video surveillance capabilities. The system utilizes mobile phones as its primary cameras, which can be remotely accessed for real-time video analysis. While the IVSS can function independently, it can also integrate with existing video surveillance setups, making it versatile for various deployment scenarios. The system interfaces with the end-users through a web-based application.

2.2 Product Functions

The primary functions of the IVSS include real-time video analysis, object detection, incident reporting, and user interaction through the web application. These functions enable the system to detect critical incidents, such as human falls, vehicle crashes, and social distancing violations, while providing a user-friendly interface for monitoring and response.

2.3 User Characteristics

The IVSS is designed to cater to a diverse user base, including security personnel, system administrators, and end-users seeking surveillance information. User characteristics influencing the requirements encompass various levels of technical expertise, from individuals with basic computer skills to professionals with in-depth knowledge of video surveillance systems.

2.4 General Constraints

Several general constraints shape the development of the IVSS:

- **Real-time Processing:** The system must process video feeds from mobile phones in real-time, imposing constraints on algorithm efficiency and hardware capabilities.
- **Data Privacy:** Compliance with data protection laws and user privacy concerns are integral to the IVSS's design.
- **Scalability:** The system should accommodate future integrations and expanding mobile phone camera feeds without substantial architecture changes.
- **Resource Availability:** Availability of mobile phones with functioning cameras may impact system performance.

2.5 Assumptions and Dependencies

The requirements outlined in this SRS are influenced by several assumptions and dependencies:

- **Operating System:** It is assumed that a compatible operating system will be available on the designated hardware (mobile phones).
- **Data Sources:** The system relies on the availability and accessibility of mobile phones with functioning cameras.
- **Internet Connection:** Users are assumed to have internet connectivity for accessing the web-based application.
- **Third-party Services:** Dependencies on third-party services, such as email providers for notifications, may impact system functionality.

These assumptions and dependencies serve as essential considerations for the IVSS's development and adaptation in response to unforeseen changes or limitations.

3. Specific Requirements

This section outlines the detailed requirements that guide the design, implementation, and testing of the "Intelligent Video Surveillance System (IVSS)." Each requirement in this section adheres to the following attributes:

- Correct
- Traceable (both forward and backward to prior/future artifacts)
- Unambiguous
- Verifiable (i.e., testable)
- Prioritized (with respect to importance and/or stability)
- Complete
- Consistent
- Uniquely identifiable (usually via numbering like 3.4.5.6)

The requirements specified here avoid over-constraining the software project and provide a foundation for the system's development and functionality.

3.1 External Interface Requirements

This section describes the external interfaces of the Intelligent Video Surveillance System (IVSS), which encompass various aspects like user interfaces, hardware interfaces, software interfaces, and communication interfaces.

3.1.1 User Interfaces

The IVSS offers user interfaces that facilitate user interaction with the system. These interfaces are designed to be intuitive and user-friendly.

3.1.1.1 User Login Interface

A web-based login interface will be provided for users to access the system securely. Users can enter their credentials to log in.

3.1.1.2 User Dashboard

Upon successful login, users will be directed to a user dashboard interface. The dashboard will present real-time video feeds and incident alerts, allowing users to monitor the surveillance system's status.

3.1.1.3 Camera Configuration Interface

Users will have access to a camera management interface that allows them to configure, add, or remove mobile phone cameras. This interface will provide settings for camera selection and video stream setup.

3.1.2 Hardware Interfaces

The IVSS interacts with specific hardware components to collect video data and support its functionality.

3.1.2.1 Mobile Phones as Cameras

Mobile phones, acting as cameras, will serve as a hardware interface to the IVSS. These devices will capture video footage and transmit it to the system for analysis.

3.1.2.2 Email Service

The system will utilize email services as a hardware interface to send real-time email notifications to authorities and users in case of incidents. These notifications will rely on SMTP servers for effective communication.

3.1.3 Software Interfaces

The IVSS integrates various software interfaces to perform advanced video analysis and data processing.

3.1.3.1 Object Detection Algorithms

The system will integrate YOLO-based object detection algorithms for real-time incident detection. These algorithms will provide the software interface to process video feeds and identify incidents.

3.1.3.2 Backend Services

The web application will communicate with backend services through software interfaces. These services include data processing, incident storage, and notification management.

3.1.4 Communications Interfaces

The IVSS relies on communication interfaces to transfer data and notifications effectively.

3.1.4.1 Data Transmission

Video data from mobile phones will be transmitted to the system for analysis through communication interfaces. These interfaces will ensure the seamless transfer of video streams.

3.1.4.2 Email Notifications

The IVSS will use communication interfaces to send incident notifications via email. It will interact with email servers to deliver real-time alerts.

These external interfaces are vital components of the IVSS, enabling efficient interaction between users, hardware, software, and communication channels. The system's functionality heavily relies on the seamless operation of these interfaces.

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 Incident Detection and Analysis

3.2.1.1 Object Detection Algorithm Integration

- Introduction: The system shall integrate YOLO-based object detection algorithms for real-time incident detection and analysis.
- Inputs: Video streams captured by cameras.

- Processing: Apply YOLO algorithms to identify and classify incidents, including human falls, vehicle crashes, and social distancing violations.
- Outputs: Generate immediate alerts and initiate incident categorization.
- Error Handling: Implement mechanisms to handle algorithm errors, ensuring reliable incident detection.

3.2.1.2 Real-time Incident Notification

- Introduction: The system shall provide real-time notifications for detected incidents.
- Inputs: Incident alerts triggered by object detection algorithms.
- Processing: Initiate immediate notifications to designated authorities via the email notification system.
- Outputs: Authorities receive timely alerts with incident details.
- Error Handling: Implement a reliable notification mechanism and handle delivery failures.

3.2.2 User Interface and Interaction

3.2.2.1 Web Application Integration

- Introduction: The system shall integrate a user-friendly web application for monitoring and managing surveillance activities.
- Inputs: User interactions with the web application.
- Processing: Retrieve and display real-time incident alerts, video feeds, and system status.
- Outputs: Present a responsive and intuitive web interface for users.
- Error Handling: Display error messages for failed data retrieval or application interactions.

3.2.1.1 Dashboard Features

- Introduction: The system shall include a comprehensive dashboard with key features for incident monitoring.
- Inputs: User interactions with the dashboard interface.
- Processing: Display incident details, categorized alerts, and relevant video feeds.
- Outputs: Present a user-friendly dashboard for efficient incident management.
- Error Handling: Provide feedback for unsuccessful dashboard interactions.

3.2.3 System Configuration

3.2.3.1 Camera Integration and Configuration

- Introduction: The system shall support the integration and configuration of various camera sources, including mobile phones.
- Inputs: User-defined camera settings.
- Processing: Establish communication with cameras for video data transmission.
- Outputs: Confirm successful camera integration and configuration.
- Error Handling: Provide feedback for unsuccessful camera integration attempts.

3.2.1.1 Camera Selection and Setup

- Introduction: The system shall allow users to select and set up cameras for surveillance.
- Inputs: User selections for camera configuration.
- Processing: Configure selected cameras for video streaming.
- Outputs: Confirm successful camera setup.
- Error Handling: Provide feedback for unsuccessful camera setup attempts.
-

These functional requirements are derived from the project's goal to create an Intelligent Video surveillance System, emphasizing incident detection, real-time notifications, user interface functionality, and system configuration flexibility. Each requirement is aligned with the system's objectives and contributes to the overall effectiveness of the surveillance solution

3.3 Use Cases

3.3.1 Use Case 1: Real-time Incident Detection and Notification

3.3.1.1 Description

- **Overview:** The system detects incidents in real-time using YOLO-based object detection and notifies authorities promptly.
- **Actors:** Surveillance System, Authorities
- **Preconditions:** The system is operational, and cameras are actively streaming.

3.3.1.2 Main Flow

1 **Actor Action:** Surveillance System processes video streams using YOLO algorithms.
2 **System Action:** Identify and classify incidents, such as human falls, vehicle crashes, or social distancing violations.
3 **System Action:** Trigger immediate incident alerts.
4 **Actor Action:** Authorities receive real-time email notifications.
5 **System Action:** Log incident details for future reference.

3.3.1.3 Alternate Flow - Incident Classification

Condition: If the incident type is unclear.
1 **Actor Action:** Authorities review incident details.
2 **System Action:** Provide additional context or images for better classification.
3 **Actor Action:** Authorities confirm or update incident classification.

3.3.1.4 Exception Flow - Algorithm Error

Condition: YOLO algorithms encounter errors.
1 **System Action:** Log the error.
2 **System Action:** Implement error recovery mechanisms.
3 **Actor Action:** Authorities receive a notification of algorithm error.

3.3.2 Use Case 2: User Interaction and Monitoring

3.3.2.1 Description

- **Overview:** Users interact with the web application to monitor and manage surveillance activities.
- **Actors:** Surveillance System Users
- **Preconditions:** The web application is accessible and the system is operational.

3.3.2.2 Main Flow

1. **Actor Action:** User logs into the web application.
2. **System Action:** Authenticate user credentials.
3. **Actor Action:** Access the dashboard displaying real-time incident alerts.
4. **Actor Action:** View categorized incident details and video feeds.
5. **Actor Action:** Acknowledge or dismiss incidents as needed.
6. **System Action:** Update incident status and notifications.

3.3.2.3 Alternate Flow - Dashboard Customization

Condition: User wants to customize the dashboard view.

1. **Actor Action:** User selects preferences for incident display.
2. **System Action:** Apply selected preferences to the dashboard.
3. **Actor Action:** View the customized dashboard.

3.3.2.4 Exception Flow - User Authentication Failure

Condition: Incorrect user credentials.

1. System Action: Display authentication failure message.
2. Actor Action: Retry login with correct credentials.

These use cases outline key interactions within the Intelligent Video Surveillance System, covering real-time incident detection and user interactions through the web application. Each use case defines the main flow, alternate flows, and exception flows to ensure comprehensive coverage of system functionality and user interactions.

3.4 Classes / Objects

3.4.1 Object: SurveillanceSystem

- **Description:** Represents the main intelligent video surveillance system.
- **Attributes:**
  - **-detectionAlgorithm: String**

    ➪ ***Description:*** The type of detection algorithm used (e.g., YOLO).

  - **-incidentAlerts: List<String>**

    ➪ ***Description:*** A list of incident alerts generated by the system.

  - **-liveVideoFeeds: List<VideoFeed>**

➥ ***Description:*** A list of live video feeds available in the system.

- **-videoFromSource: VideoSource**

➥ ***Description:*** The source of video input (e.g., webcam, YouTube).

- **-selectedDetectionType: String**

➥ ***Description:*** The currently selected type of detection (e.g., distance monitoring, fall detection, crash detection).

- **-isRecording: boolean**

➥ ***Description***: Flag indicating whether real-time recording is active.

- **-users: List<User>**

➥ ***Description:*** List of registered users.

- **Functions:**

- **+ startMonitoring(): void**

➥ ***Description:*** Initiates the monitoring process to detect and classify incidents.

- **+ notifyAuthorities(incident: String): void**

➥ ***Description:*** Notifies relevant authorities about detected incidents.

- **+ setVideoSource(source: VideoSource): void**

➥ ***Description:*** Sets the source of video input.

- **+ selectDetectionType(detectionType: String): void**

➥ ***Description:*** Allows the user to select the type of detection (e.g., distance monitoring, fall detection, crash detection).

- **+ startRecording(): void**

➥ ***Description:*** Activates real-time recording.

- **+ stopRecording(): void**

➥ ***Description:*** Deactivates real-time recording.

- **+ registerUser(user: User): void**

➥ ***Description:*** Adds a new user to the system.

- **+ loginUser(username: String, password: String): boolean**

➥ ***Description:*** Authenticates a user login.

3.4.2 Object: VideoFeed

- **Description:** Represents a live video feed.
- **Attributes:**

- **feedID: String**

➾ Description: Unique identifier for the video feed.

- **videoSource: VideoSource**

➾ Description: Source of the video feed (e.g., webcam, online video).

✚ **Functions:**

(No specific functions at this level. Handled by SurveillanceSystem)

3.4.3 Object: VideoSource

✚ **Description:** Represents the source of video input.

✚ **Attributes:**

- **sourceType: String**

➾ ***Description:*** Type of video source (e.g., webcam, YouTube).

✚ **Functions:**

(No specific functions at this level. Handled by SurveillanceSystem)

3.4.4 Object: User

✚ ***Description:*** Represents a user of the intelligent video surveillance system.

✚ **Attributes:**

- **username: String**

➾ ***Description:*** The username of the user.

- **password: String**

➾ ***Description:*** The password of the user.

✚ Functions:
(No specific functions at this level. Handled by SurveillanceSystem)

3.5 Non-Functional Requirements

3.5.1 Performance

3.5.1.1 Performance Requirement 1

Description: The system shall achieve real-time processing for video analysis.
Measurable: 95% of video frames must be analyzed within 500 milliseconds.

3.5.1.2 Performance Requirement 2

Description: The system shall support simultaneous monitoring of at least five live video feeds.
Measurable: The system should handle a minimum of five live video feeds concurrently without performance degradation.

3.5.2 Reliability

3.5.2.1 Reliability Requirement 1

Description: The system shall have a Mean Time Between Failures (MTBF) of at least 30 days.
Measurable: The system is expected to operate continuously without failure for a minimum of 30 days.

3.5.2.2 Reliability Requirement 2

Description: The probability of false alarms shall not exceed 1%.
Measurable: The system's false alarm rate should be maintained below 1% under normal operating conditions.

3.5.3 Availability

3.5.3.1 Availability Requirement

Description: The system shall have an availability of at least 99%.
Measurable: The system should be available and operational 99% of the time during standard operating hours.

3.5.4 Security

3.5.4.1 Security Requirement 1

Description: User authentication shall be performed before accessing the system.
Measurable: All users must provide valid credentials (username and password) for system access.

3.5.4.2 Security Requirement 2

Description: Video data transmission shall be encrypted to ensure data integrity and confidentiality.
Measurable: All video data transmitted over networks must be encrypted using industry-standard encryption protocols.

3.5.5 Maintainability

3.5.5.1 Maintainability Requirement 1

Description: The system shall provide automated software updates.
Measurable: Software updates should be automatically delivered to the system, ensuring it stays current with the latest features and security patches.

3.5.5.2 Maintainability Requirement 2

Description: The system shall have a modular architecture to facilitate easy maintenance.

Measurable: The system's architecture should allow for the independent update or replacement of components without affecting the entire system.

3.5.6 Portability

3.5.6.1 Portability Requirement

Description: The system shall be compatible with multiple operating systems, including Windows, Linux, and macOS.
Measurable: The system should be able to run on various operating systems to accommodate user preferences and organizational requirements.

3.5.7 Usability

3.5.7.1 Usability Requirement

Description: The user interface shall be intuitive and user-friendly.
Measurable: Users should rate the system's usability at a minimum of 80% in usability testing.

3.5.8 Interoperability

3.5.8.1 Interoperability Requirement

Description: The system shall be compatible with standard video file formats.
Measurable: The system should successfully process video files in commonly used formats, such as MP4 and AVI.

3.6 Logical Database Requirements

3.6.1 Database Utilization

Description: The system will utilize a database to store relevant information, including incident records, user data, and system configurations.
Rationale: Employing a database allows for efficient data management, retrieval, and ensures the persistence of critical information.

3.6.2 Data Formats

Description: Data within the database should adhere to standardized formats, facilitating consistency and ease of retrieval.
Rationale: Standardized formats promote interoperability and simplify data processing and analysis.

3.6.3 Storage Capabilities

Description: The database must have sufficient storage capacity to accommodate a growing volume of video footage, incident records, and user data.
Rationale: Adequate storage ensures that the system can operate effectively over extended periods without the risk of data loss.

3.6.4 Data Retention

Description: Define policies for data retention, specifying the duration for which different types of data (e.g., video recordings, incident logs) will be preserved.

Rationale: Establishing data retention policies aligns with privacy regulations and optimizes storage resources.

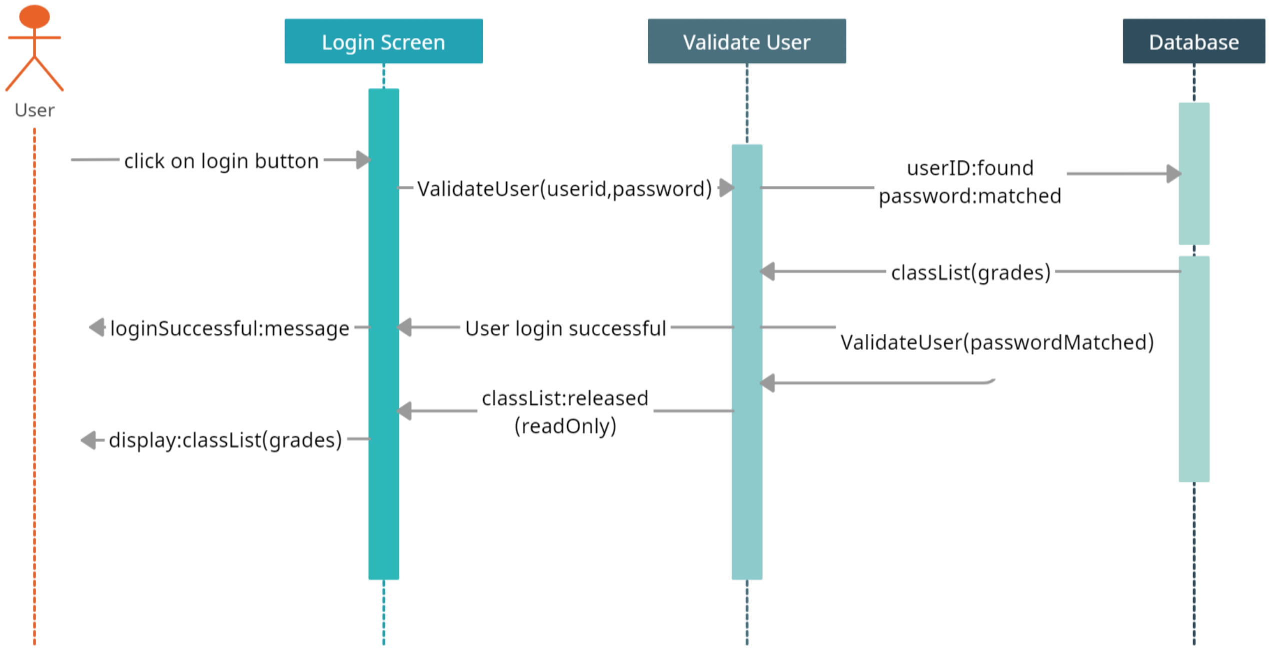
3.6.5 Data Integrity

Description: Implement mechanisms to ensure data integrity, including error checking, validation rules, and encryption for sensitive information.
Rationale: Maintaining data integrity is crucial for reliable and accurate system operation, preventing errors or corruption in stored information.

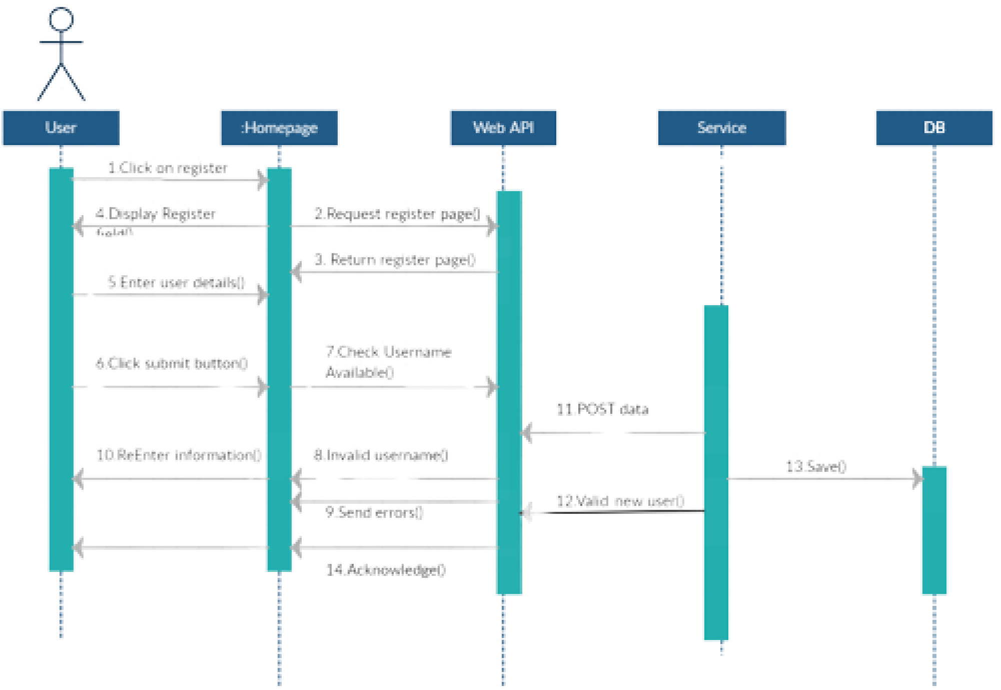
4 Analysis Models

4.1 Sequence Diagrams

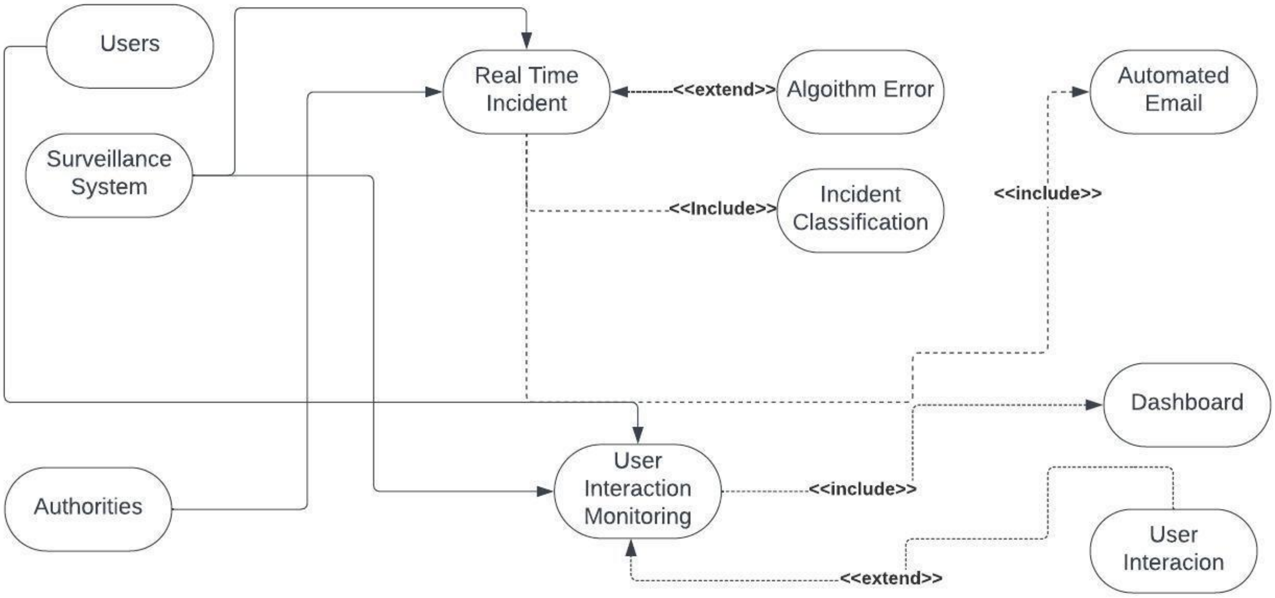
4.1.1 Sequence Diagram of User Login



4.1.2 Sequence Diagram of User Registration

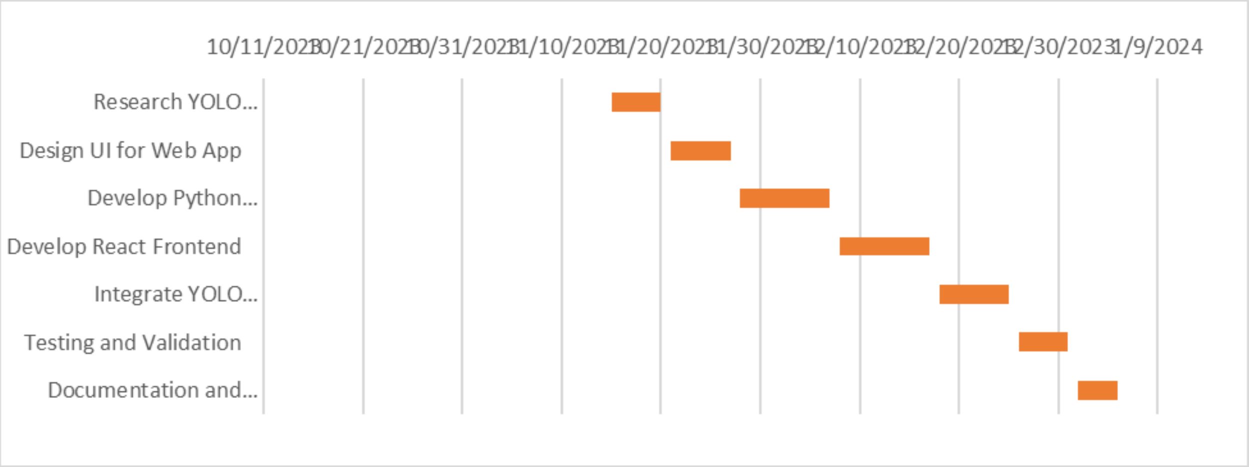


4.2 Use Case Diagram



4.3 Time Frame Gantt Chart

Start date	End date	Description	Duration
11/15/2023	11/20/2023	Research YOLO Algorithms	5
11/21/2023	11/27/2023	Design UI for Web App	6
11/28/2023	12/7/2023	Develop Python Backend	9
12/8/2023	12/17/2023	Develop React Frontend	9
12/18/2023	12/25/2023	Integrate YOLO Algorithms	7
12/26/2023	12/31/2023	Testing and Validation	5
1/1/2024	1/5/2024	Documentation and Manuals	4


10/11/2023
10/21/2023
10/31/2023
11/10/2023
11/20/2023
11/30/2023
12/10/2023
12/20/2023
12/30/2023
1/9/2024
Research YOLO...
Design UI for Web App
Develop Python...
Develop React Frontend
Integrate YOLO...
Testing and Validation
Documentation and...