BSCS FINAL PROJECT

Requirements Specification

TruSec – Trucking Surveillance System



Project Advisor

**Syed Nisar Ali Balti**

Presented by:

**Group ID: F23CS112**

**Student Reg# Student Name**

L1F19BSCS0139 Ahmed Naeem

L1F19BSCS0582 Najam Irfan

**Faculty of Information Technology**

**University of Central Punjab**

Software Requirements Specification

Version 1.0

TruSec – Trucking Surveillance System

Advisor: Syed Nisar Ali Balti

Group: F23CS112

|  |  |
| --- | --- |
| Member Name | Primary Responsibility |
| Ahmed Naeem | Backend Development and Embedded Systems |
| Najam Irfan | Client-Side Development and AI Model Training |

Table of Contents

Table of Contents i

Revision History ii

Abstract 1

1. Introduction 2

1.1 Purpose 2

1.2 Document Conventions 2

1.3 Intended Audience and Reading Suggestions 3

1.4 Project Scope 3

1.5 Objective(s)/Aim(s)/Target(s) 3

1.6 Challenges 3

1.7 Learning Outcomes 4

1.8 Nature of End Product 4

1.9 Completeness Criteria 4

2. Overall Description 4

2.1 Product Perspective 4

2.2 Product Features 6

2.3 User Classes and Characteristics 6

2.4 Operating Environment 7

2.5 Design and Implementation Constraints 7

2.6 User Documentations 7

2.7 Assumptions and Dependencies 7

3. Product Features / Functional Requirements 8

3.1 Remote Region Operation and Facial Expression Detection 9

3.2 Security and Data Integrity during RF Transmission 10

3.3 RF Transmission to ESP32 and Web Socket Communication 11

3.4 Analysis and Modeling of Requirements 12

ER Diagram 12

Class Diagram 13

Sequence Diagrams 14

Circuit Diagrams 16

4. External Interface Requirements 18

4.1 User Interfaces 18

4.2 Hardware Interfaces 18

4.3 Software Interfaces 19

4.4 Communication Interfaces 20

5. Other Nonfunctional Requirements 21

5.1 Performance Requirements 21

5.2 Safety Requirements 21

5.3 Security Requirements 22

5.4 Additional Software Quality Attributes 23

6. Other Requirements 24

7. Revised Project Plan 25

7.1 Gantt Chart 26

8. References 26

Appendix A: Glossary 27

Appendix B: IV & V Report 28

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Abstract

This project aims to develop a practical end-to-end solution for monitoring trucks carrying sensitive consignments over long distances in remote areas where secure surveillance systems are lacking. The solution integrates various technologies to ensure real-time visibility and enhanced security throughout the transportation process.

The system includes GPS tracking devices installed in each truck to monitor their precise location at all times. Raspberry Pi 4 device collect and transmit important vehicle data, such as speed, driver’s facial expression, and Geo Location enabling real-time monitoring of the truck's condition.

To ensure the proper handling of sensitive cargo, IoT sensors is deployed to monitor environmental factors such as temperature, humidity, and light exposure. These sensors provide continuous data transmission to a centralized control center, ensuring compliance with specific requirements.

In addition, strategically placed surveillance cameras inside the trucks capture the status of the consignments and allow live image feeds to be transmitted to the control center. This provides visual monitoring and enables immediate action in case of security breaches or irregularities.

To establish effective communication, a reliable network infrastructure is implemented, utilizing cellular networks and radio communication to transmit data from the trucks to the control center. This ensures uninterrupted monitoring even in remote areas with limited connectivity.

The control center serves as the centralized hub for data analysis, where incoming information is processed using machine learning algorithms to detect patterns, anomalies, or potential risks. Alarms and alerts are triggered when deviations from normal behaviors are identified, allowing for timely response and necessary interventions.

# Introduction

## Purpose

The main focus of TruSec will be to secure the trucks carrying sensitive consignments by allowing them to communicate effectively even over the low-bandwidth network. The current solutions in the market are costly because they are using satellite communication which is an expensive approach. Our aim is just to demonstrate the effective communication between the truck and the control center over low network bandwidth, and this will be our primary goal.

## Document Conventions

**Headings**:

Font Style: Times

Font Size: 18

Font Weight: Bold

**Sub Headings**:

Font Style: Times

Font Size: 14

Font Weight: Bold

**Links:**

Font Style: Times

Font Size: 12

Font Weight: Bold

**Others**:

Font Style: Times New Roman

Font Size: 12

Font Weight: Normal

**Table:**

Font Style: Vendera

Font Size: 12

Font Weight: Normal

## Intended Audience and Reading Suggestions

For **Developers**, start with the "Introduction" and dive into the technical details in "Overall Description," "Product Features," and "Functional Requirements." This sequence provides insights into the project's purpose, scope, and specific implementation requirements.

**Users** should begin with the "Abstract" to understand the project's goals. Explore "User Interfaces" for insights into the dashboard's features and then focus on "Security Requirements" for assurances of data protection. The "Functional Requirements" section is crucial for understanding system capabilities.

**Testers** should first look into "Functional Requirements" to understand expected system behavior. Focus on "Security Requirements" for testing data protection measures and refer to "Performance Requirements" for insights into system responsiveness.

## Project Scope

Trucks carrying sensitive consignments travel over long distances into remote areas, there is no secure surveillance system to monitor their status. This project aims to provide a practical end-to-end solution to overcome this problem. Moreover, during long driving hours, truck drivers may get drowsy or less attentive. To tackle this, we will embed a facial expressions detector that will cause an alarm whenever the driver will have a negative attitude.

We are developing a solution which uses a Raspberry Pi 4 microcontroller, paired with a Wi-Fi camera, GSM Module and Radio Waves Module. This Raspberry Pi 4 device will communicate with our Backend service hosted on AWS EC2 to post the image data and other relevant information to monitor the truck and truck driver.

The Raspberry Pi 4 device will periodically take snapshots using the Wi-Fi cameras. These Wi-Fi cameras will be connected locally with the microcontroller. Then the Raspberry Pi 4 MCU will send these snapshots over to our REST API. The embedded microcontroller will use the GSM Module (SIM Module), to transmit data primarily. As backup the device will use Radio Waves to transmit data. However, the snapshots transmitted using Radio Waves will have low image quality. These snapshots will also be used by our backend service to detect the facial expressions of the driver. These results will then be showed on the Angular dashboard.

## Objective(s)/Aim(s)/Target(s)

* PCB Circuit for Transmitter and Receiver with RF & GSM Module
* Admin Dashboard for monitoring of the truck
* Backend Service to send and receive image data to dashboard.
* AI Model for image classification to detect the driver’s behavior

## Challenges

* Understanding and implementing the RF communication protocol
* Fault tolerant communication from remote areas
* High accuracy AI model to detect driver’s behavior.
* Understanding and implementation of PCB circuit design

## Learning Outcomes

This project will help us understand the fundamentals of RF and GSM communication. Also, the hands-on implementation of AI model to detect the user expressions will allow us to become familiar with machine learning. This project also has a web development part which allow the students to comprehend basic fundamentals of web programming like WebSockets, Rest API etc. Simultaneously, expertise in PCB circuit design ensures efficient integration of RF modules and the AI model for a cohesive and reliable system.

## Nature of End Product

The end product will be a pair of MCUs one will be deployed on the truck which will work as a transmitter to send data, and the other one will be deployed in the control center which will work as a receiver to receive data from truck and upload it to the server. Along with the hardware we will have the admin dashboard where user can monitor the real-time updates from the truck.

## Completeness Criteria

Our primary goal will be to demonstrate the communication over the radio frequency from the receiver placed in MCU on the truck to the transmitter placed in the MCU at the monitoring station. Along with the high accuracy of our trained AI Model so the results can be validated.

|  |  |  |
| --- | --- | --- |
| **S#** | **Criteria** | **Weightage %** |
| 1 | Web Dashboard | 10% |
| 2 | Backend API | 15% |
| 3 | AI Model | 15% |
| 4 | Receiver/Transmitter MCU with IoT Sensors | 60% |

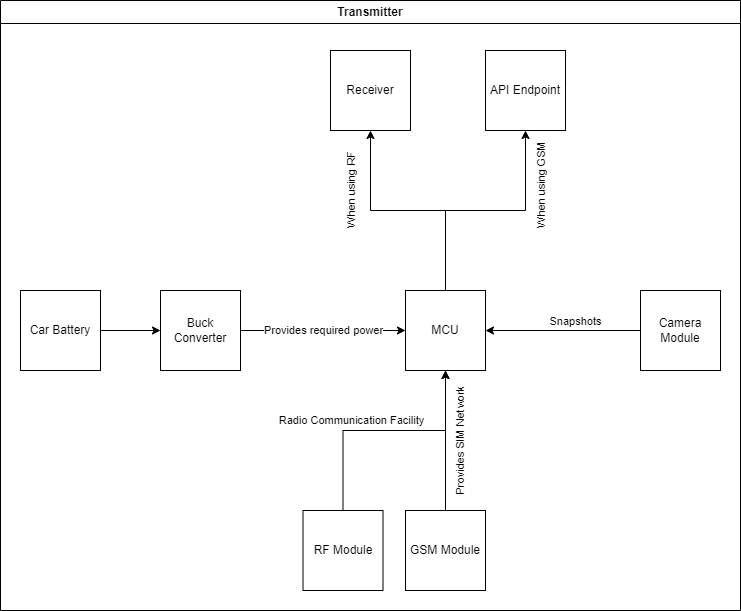
# Overall Description

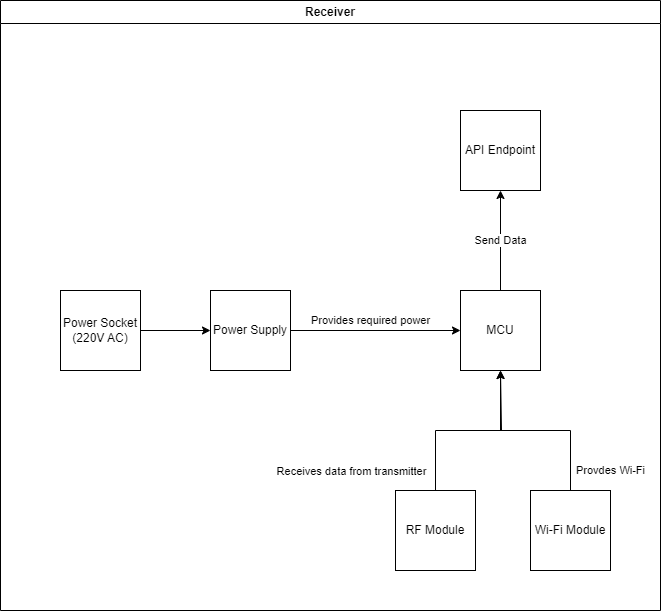
## Product Perspective

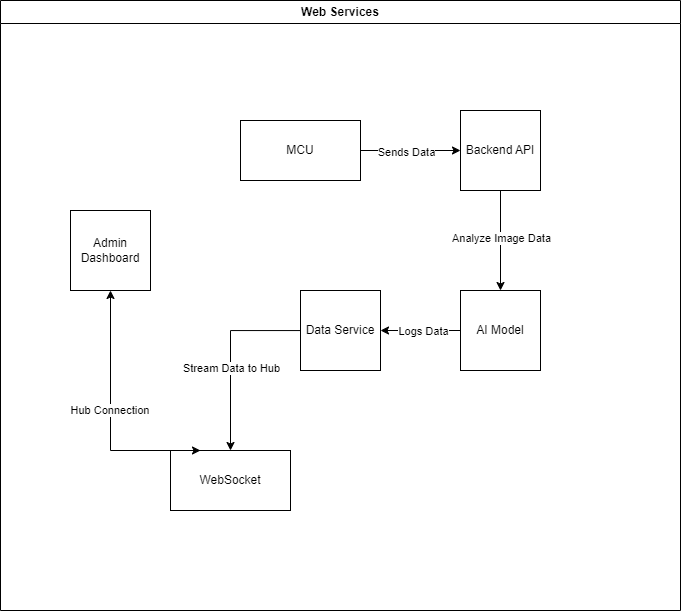
The project aims to provide a practical end-to-end solution for monitoring trucks carrying sensitive consignments in remote areas. By integrating GPS tracking, IoT sensors, surveillance cameras, communication networks, and data analytics, this solution offers real-time visibility, enhanced security, and efficient monitoring capabilities.

The implementation of such a comprehensive system enables continuous monitoring of the truck's location, vehicle condition, and cargo environment. The centralized control center, equipped with a monitoring dashboard and data analytics tools, empowers operators to detect anomalies, security threats, or deviations from normal behavior.

By leveraging technologies like networks, cloud platforms and IoT, the solution addresses the unique challenges presented by remote areas. It overcomes the limitations of existing surveillance systems and provides an efficient, reliable, and secure framework for monitoring trucks and their sensitive consignments. Following are the block diagrams for more clear understanding of the whole system:

****

****

****

## Product Features

TruSec will consist of three major parts, a transmitter, a receiver, and an admin dashboard. The transmitter and receiver will be standalone hardware devices. The transmitter’s main function will be to send the information about the truck that it is attached to. This information will include GPS location, driver behavior etc. The receiver will receive the data from the transmitter and send it to the admin dashboard using backend API. The admin dashboard will display the data it received from the backend API and in case of any alarming situation, the admins will be notified.

## User Classes and Characteristics

**Monitoring Staff:** These are the users who will be monitoring the truck from the monitoring station. In case of any problem on the truck they will notify the particular authorities.

**Administration Staff:** These are the users who will be supervising the whole system and take actions based on the results received from the truck. They will have technical expertise and will always be available.

**Truck Drivers:** Truck drivers are one of the main stakeholders in this application. We will take their facial expressions and send it back to the monitoring staff. Also, these users will get some troubleshot training about the MCU placed in the truck so we can have a seamless process.

## Operating Environment

The embedded device (Raspberry Pi 4) will operate on Linux operating system. The admin dashboard and the backend API will be hosted using Docker image on a AWS EC2 instance. As AWS EC2 have its own hardware allocation mechanism which will keep the software system to peacefully coexist. Users will be able to get the results and monitor the truck from any browser on the internet.

## Design and Implementation Constraints

**Performance and scalability**: The system should be able to handle a high volume of traffic and perform well even under heavy load.

**Security**: The system should implement appropriate security measures to protect against potential attacks or data breaches, such as HTTPS encryption, password hashing, and cross-site scripting (XSS) protection.

**Radio Frequencies**: For the demonstration of the system, we will be using the public bands for radio communications.

## User Documentations

We will deliver the User Manual along with each of the MCU which will contain the information about the installing the device into the truck and at the monitoring station. User Manual will also consist of instructions and pictorial demonstration so that users can perform each step accordingly. Along with User Manual we will have a troubleshoot guide which includes all the information related to the troubleshooting of the MCUs.

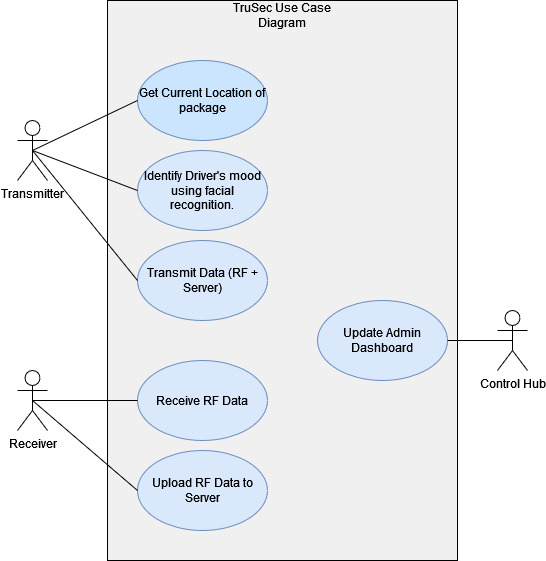
## Assumptions and Dependencies

**Radio Frequency Compatibility**: It is assumed that users will have access to the public bands.

**Web browser compatibility**: It is assumed that users will be using a compatible web browser to access the website.

**Device compatibility**: It is assumed that users will be using a compatible device to access the website.

# 3. Product Features / Functional Requirements



## 3.1 Remote Region Operation and Facial Expression Detection

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | **Remote Region Operation and Facial Expression Detection** | |
| **Purpose** | | Validate the surveillance system's operation in remote regions and the accuracy of facial expression detection in challenging environments. | |
| **Priority** | | High | |
| **Pre-conditions** | | The surveillance system, including the facial expression detection model, is deployed on trucks in remote regions. | |
| **Post-conditions** | | The surveillance system successfully operates in remote regions with limited infrastructure. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Deploy trucks with the surveillance system in a remote region. | | The system initializes and establishes RF communication despite limited infrastructure. |
| **2** | Capture driver inside the trucks. | | The facial expression detection model accurately processes the image feeds, providing real-time insights into the emotional states of individuals. |
| **3** | Introduce environmental challenges such as dust or vibrations during truck movement. | | The surveillance system maintains stable operation, and facial expression detection remains accurate despite environmental challenges. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Simulate extreme weather conditions, such as heavy rain or snow. | | The surveillance system continues to operate, demonstrating resilience to extreme weather conditions. |
| **2** | Test the system in areas with limited RF signal strength. | | The system adapts to low RF signal strength, ensuring reliable data transmission and facial expression detection. |

Table : Remote Region Operation and Facial Expression Detection

## 3.2 Security and Data Integrity during RF Transmission

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | **Security and Data Integrity during RF Transmission** | |
| **Purpose** | | Validate the security and data integrity of the surveillance system during RF transmission. | |
| **Priority** | | High | |
| **Pre-conditions** | | The trucks are equipped with the surveillance system and RF communication modules. | |
| **Post-conditions** | | The transmitted data is secure and free from unauthorized access. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Initiate facial expression detection and transmit data over RF bands. | | The system encrypts the transmitted data, ensuring the security of sensitive information. |
| **2** | Monitor the RF transmission for potential security breaches. | | The surveillance system detects and prevents any unauthorized attempts to access or manipulate the transmitted data. |
| **3** | Ensure that the transmitted facial expression data matches the original detections. | | Data integrity is maintained during RF transmission, and the facial expression data received is accurate and unaltered. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Simulate a hacking attempt on the RF transmission. | | The system identifies the hacking attempt and implements security measures to safeguard the data. |
| **2** | Introduce interference during RF transmission. | | The system employs error-checking mechanisms to detect and correct any data corruption caused by interference. |

Table 2: Security and Data Integrity during RF Transmission

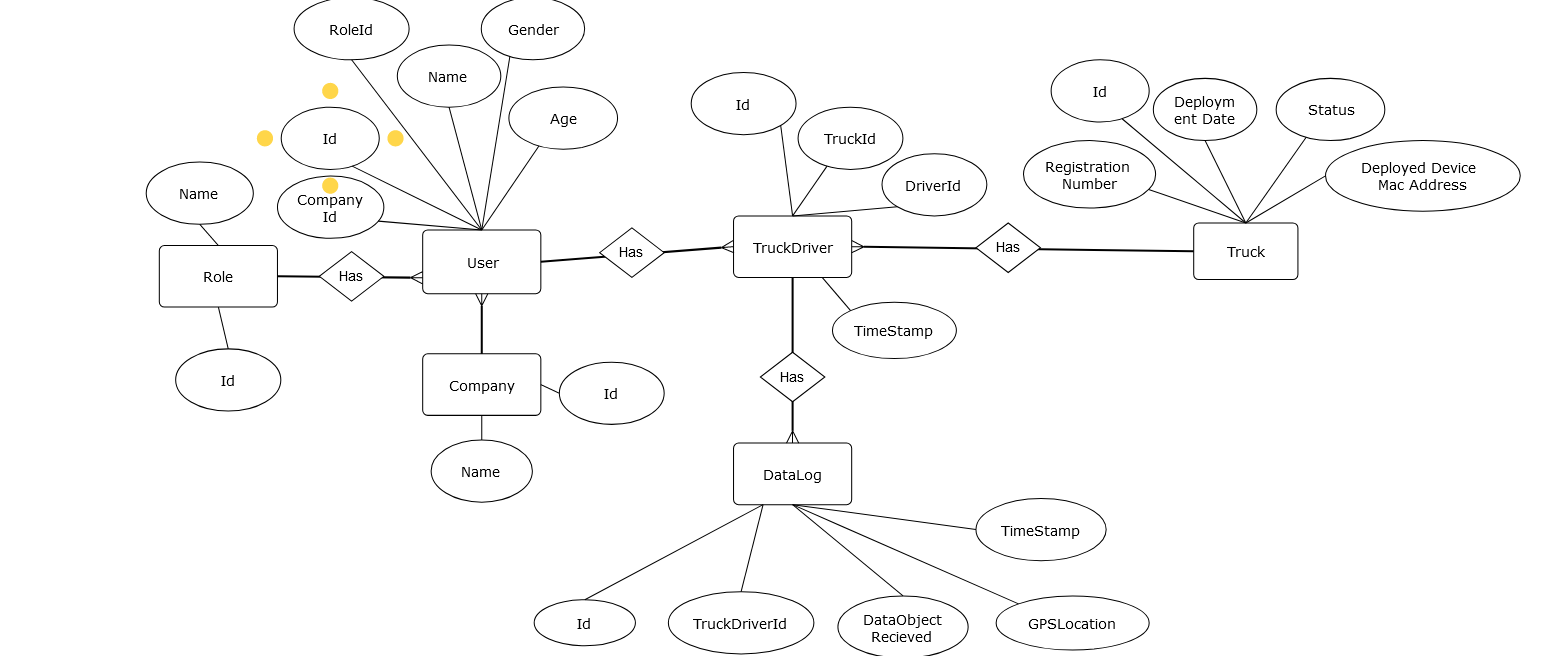
## 3.3 RF Transmission to ESP32 and Web Socket Communication

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | **RF Transmission to ESP32 and Web Socket Communication** | |
| **Purpose** | | Validate the transmission of facial expression data from Raspberry Pi to the ESP32 via RF bands and subsequent communication with a web dashboard through web sockets. | |
| **Priority** | | High | |
| **Pre-conditions** | | The facial expression detection model is integrated with the Raspberry Pi 4. RF module is installed on the MCU. A web dashboard is set up to receive real-time data through web sockets. | |
| **Post-conditions** | | Facial expression data is successfully transmitted from the Raspberry Pi to the ESP32 over RF bands. The web dashboard displays real-time facial expression information. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Initiate facial expression detection on the Raspberry Pi. | | The detected facial expression data is transmitted to the ESP32 over RF bands. |
| **2** | ESP32 receives the transmitted data. | | ESP32 processes the received data and establishes a connection with the web dashboard using web sockets. |
| **3** | Display the real-time facial expression information on the web dashboard. | | The web dashboard updates in real-time, showing the detected facial expressions from the Raspberry Pi. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Introduce RF interference during transmission. | | The system handles RF interference gracefully, and the ESP32 successfully recovers and establishes communication with the web dashboard. |
| **2** | Simulate high traffic on the web dashboard. | | The web dashboard efficiently handles high volumes of real-time facial expression data without performance degradation. |

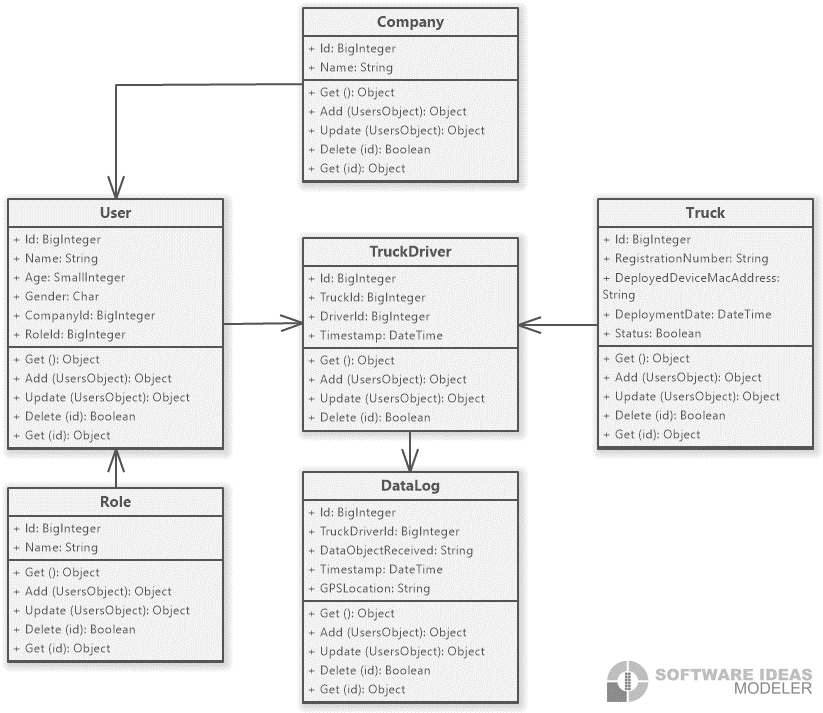
Table 3: RF Transmission to ESP32 and Web Socket Communication

## 3.4 Analysis and Modeling of Requirements

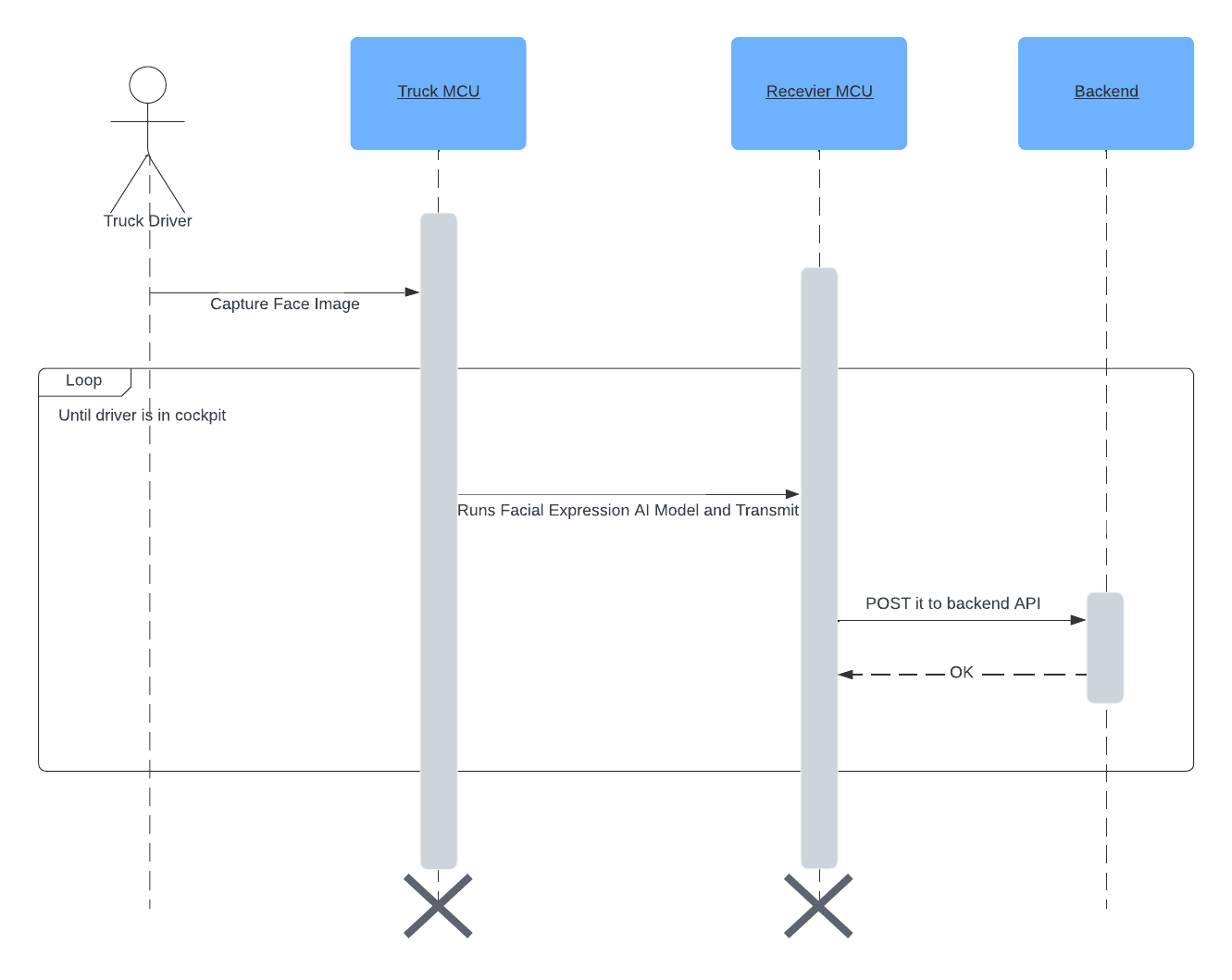
### ER Diagram



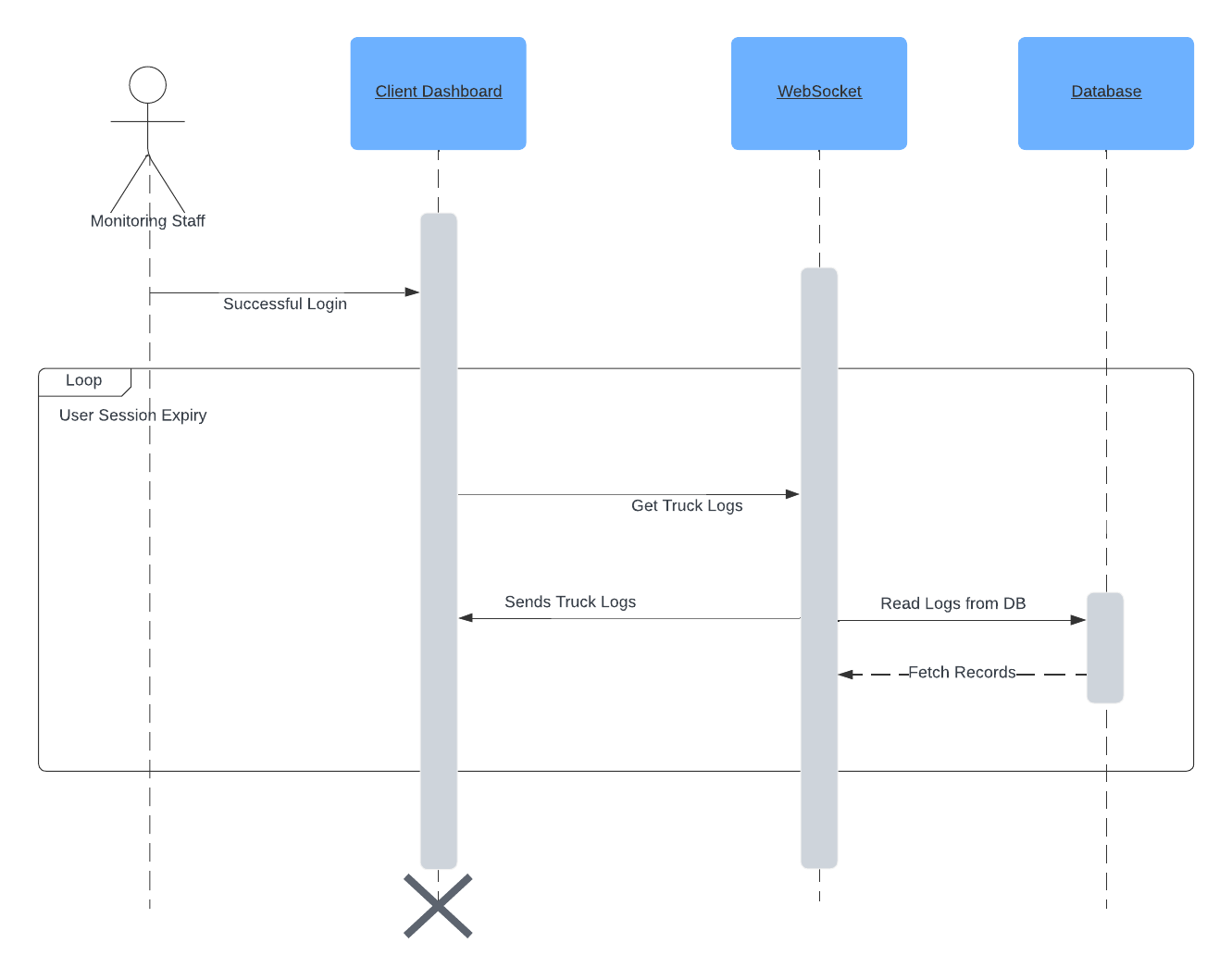
### Class Diagram



### Sequence Diagrams

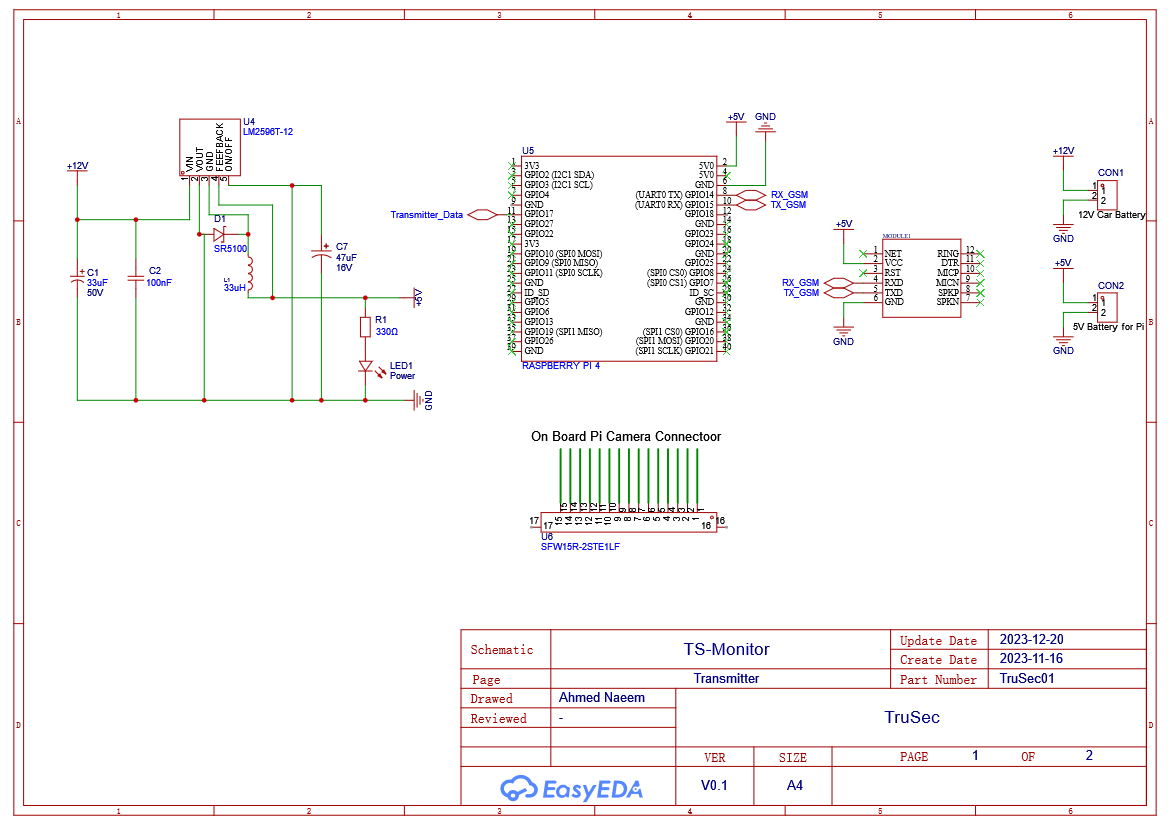
Capture Driver Behavior

Stream Truck Logs on Dashboard

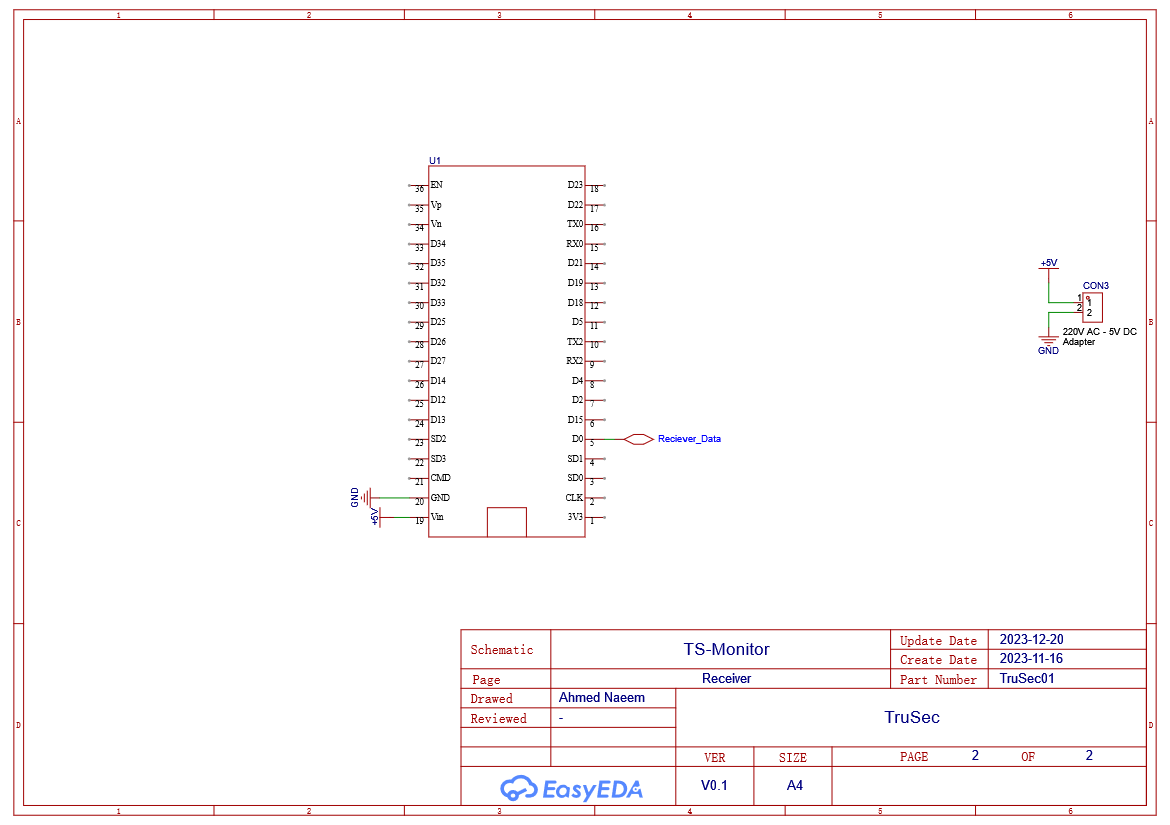


### Circuit Diagrams

Transmitter



Receiver



# External Interface Requirements

## User Interfaces

This software system will contain a user interface for the monitoring station. The UI will be very easy to use and simple design. The main screen will consist of a list of trucks where we can see all the onboarded trucks in the system. If the user clicks on the truck info the application will be navigate to the live feed screen where the real-time results from the truck will show up along with its GPS coordinates.

## Hardware Interfaces

The TruSec software system interfaces with specific hardware components to ensure seamless functionality. Each interface is characterized by both logical and physical aspects:

**Raspberry Pi 4:** The software communicates with the Raspberry Pi 4 microcontroller, interacting with its GPIO pins for data transmission and reception. The interface involves connecting to the GPIO pins of the Raspberry Pi 4, utilizing communication protocols such as SPI, I2C, or UART for data transfer.

**GSM Module and Radio Waves Module:** The software interfaces with the GSM Module and Radio Waves Module for data transmission, utilizing specific communication protocols relevant to each module. Connection to the microcontroller involves utilizing appropriate interfaces and pins for each module, ensuring reliable communication over the selected frequencies.

**IoT Sensors:** The software interacts with IoT sensors to gather environmental data, including temperature, humidity, and light exposure. Connectivity involves integrating the sensors with the microcontroller through designated input pins, utilizing communication protocols such as I2C or analog interfaces.

**Wi-Fi Cameras:** The software communicates with Wi-Fi cameras to capture live image feeds for facial expression detection. The interface includes connecting the cameras to the Raspberry Pi 4 through USB or other compatible interfaces, utilizing camera APIs for data retrieval.

**RF Transmission Module:** The software interfaces with the RF Transmission Module to transmit data over radio frequencies. Connection to the microcontroller involves utilizing specific pins for RF communication, implementing RF communication protocols for seamless data transmission.

## Software Interfaces

The TruSec software system interfaces with various software components to achieve its objectives. These interfaces encompass connections with databases, operating systems, tools, libraries, and integrated commercial components:

**Backend API (Application Programming Interface):** The software interacts with a backend API hosted on AWS EC2. The system sends truck information, including GPS location, driver behaviour, and live image feeds, to the backend API. It receives acknowledgments and responses from the API. Facilitates the exchange of real-time data between the surveillance system and the backend server for further processing and storage.

**Admin Dashboard:** The software interfaces with an admin dashboard hosted on AWS EC2. The system sends processed data and alarms to the admin dashboard. It receives user interactions and requests for specific truck information. Enables administrators to monitor and manage the surveillance system through a user-friendly web interface.

**Operating System (Linux on Raspberry Pi 4):** The software runs on the Linux operating system deployed on the Raspberry Pi 4. The system interacts with the Linux OS for resource management, process control, and data storage. Ensures seamless operation and integration with the underlying operating system environment.

**Docker Containerization:** The software uses Docker containers for hosting the admin dashboard and backend API on AWS EC2. The system interacts with Docker containers for deployment, scaling, and isolation of software components. Facilitates consistent deployment and management of software components in a containerized environment.

**Web Browsers:** The software interfaces with web browsers used by monitoring and administration staff. The system sends real-time updates and monitoring information to web browsers. It receives user interactions and requests for specific truck details. Provides a user-friendly interface for staff to monitor trucks and respond to alarms.

**SQL Database:** The software interacts with an SQL database to store and retrieve essential records related to trucks, users, and system logs. The system sends and retrieves data such as truck information, user credentials, and system logs to and from the SQL database. Ensures persistent storage of critical data for historical tracking, user management, and system maintenance.

**TensorFlow (Machine Learning Library):** The software integrates with TensorFlow for implementing the machine learning model responsible for facial expression detection. The system sends image data to TensorFlow for processing and receives the model's predictions and confidence scores. Facilitates the implementation of the facial expression detection model, enhancing the system's capability to monitor driver behaviour.

These software interfaces play a crucial role in enabling data exchange, user interactions, and system functionality. The identified connections contribute to the overall effectiveness and integration of the TruSec surveillance system.

## Communication Interfaces

The TruSec software system relies on various communication interfaces to enable seamless interactions between its components. By implementing following communication interfaces, the TruSec surveillance system achieves effective and secure data exchange between its components, supporting real-time monitoring and data integrity. These interfaces encompass communication protocols, data formatting, security measures, and synchronization mechanisms:

**RF Communication Protocol:** The software utilizes RF communication for transmitting data between the trucks and the control center. Standard RF communication protocols are employed to ensure reliable and secure data transfer. Data encryption is implemented to secure sensitive information during RF transmission. The system adheres to specified data transfer rates for efficient communication. A synchronization mechanism is in place to coordinate data transmission and reception between the trucks and the control center.

**Web Sockets:** Web sockets are used for real-time communication between the Raspberry Pi and the ESP32, and between the ESP32 and the web dashboard. WebSocket communication protocols are implemented to establish and maintain a continuous connection. Secure WebSocket protocols are utilized to encrypt data during transmission. The system adheres to specified data transfer rates for real-time updates on the web dashboard. Web sockets provide a synchronized communication channel for efficient data exchange.

**RESTful API:** The software communicates with the backend API using RESTful principles. HTTP or HTTPS protocols are used for data exchange with the backend API. HTTPS is implemented to ensure secure communication and data integrity. The system adheres to specified data transfer rates for efficient API communication. RESTful interactions follow a stateless model, allowing for flexible and scalable communication.

**Database Communication:** The software interacts with the SQL database for storing and retrieving records. SQL communication protocols (e.g., SQL over TCP/IP) are implemented for database interactions. Access controls and authentication mechanisms are in place to secure database communication. The system adheres to specified data transfer rates for efficient database transactions. Database interactions follow transactional and ACID properties to ensure data consistency.

# Other Nonfunctional Requirements

## Performance Requirements

This system must meet specific performance criteria to ensure optimal functionality under various circumstances. These requirements are outlined to guide developers in making suitable design choices and ensuring the system's effectiveness in real-world scenarios.

The system should demonstrate high responsiveness in real-time monitoring, with a maximum latency of 1 second for updating the live feed on the admin dashboard. This ensures that monitoring staff can receive timely information about the truck's status and respond promptly to any detected anomalies.

To support the scalability of the system, it should be capable of handling up to 100 simultaneous connections from monitoring staff accessing the admin dashboard. This requirement ensures that the system remains responsive and efficient, even during peak usage periods.

Additionally, the facial expression detection model, implemented using TensorFlow, should achieve a minimum accuracy rate of 95%. This high accuracy is crucial for reliable detection of driver behavior, allowing the system to trigger alarms accurately in case of negative expressions or drowsiness.

In terms of data transfer, the RF communication between the trucks and the control center should maintain a minimum data transfer rate of 256 kbps. This ensures that real-time data, including GPS coordinates and facial expression information, is transmitted efficiently over the RF bands, even in remote regions with limited connectivity.

These performance requirements aim to guarantee a responsive, scalable, and accurate surveillance system, meeting the needs of real-time monitoring in challenging environments and ensuring the safety and security of sensitive consignments during transportation.

## Safety Requirements

The safety requirements for the TruSec surveillance system are paramount to ensure the well-being of individuals and the secure transportation of sensitive consignments. These requirements address potential risks and outline safeguards to mitigate any possible loss, damage, or harm resulting from the use of the product.

**Facial Expression Detection Accuracy:** The facial expression detection model must achieve a minimum accuracy of 95%. Ensures accurate detection of driver behavior, reducing the risk of false alarms or overlooking potential safety concerns.

**Driver Alerting Mechanism:** The system must trigger an alert in real-time when the facial expression detection model identifies signs of drowsiness or negative behavior in the truck driver. Enables immediate intervention and prevents potential accidents or harm caused by a drowsy or distracted driver.

**Secure RF Communication:** Implement encryption for RF communication to ensure the confidentiality and integrity of transmitted data. Safeguards against unauthorized access to sensitive information, preventing potential misuse or tampering during data transmission.

## Security Requirements

The security requirements for the TruSec surveillance system are essential to safeguard sensitive data, ensure user privacy, and protect against potential threats. These requirements address security and privacy issues associated with the use of the product and define measures to secure data and user identities.

**Data Encryption:** Implement end-to-end encryption for all data transmissions, including RF communication and interactions with the backend API. Protects sensitive information from unauthorized access, ensuring the confidentiality and integrity of data during transit.

**User Authentication:** Implement a robust user authentication system for access to the admin dashboard and backend API. Ensures that only authorized personnel can access and manage the surveillance system, preventing unauthorized manipulation or misuse.

**Access Control:** Define and enforce role-based access control to restrict users' access to specific functionalities based on their roles. Prevents unauthorized access to sensitive features and data, enhancing overall system security.

**Incident Logging and Monitoring:** Implement comprehensive logging of security incidents and establish real-time monitoring for anomalous activities. Enables quick detection and response to potential security threats, enhancing the system's resilience against malicious activities.

**Secure Storage:** Employ secure storage mechanisms for sensitive data, including user credentials and facial expression data. Minimizes the risk of unauthorized access to stored data, maintaining the confidentiality and integrity of critical information.

**Privacy Compliance:** Ensure compliance with privacy regulations and standards governing the collection and processing of facial expression data. Protects user privacy and ensures adherence to legal requirements, building trust with users and regulatory authorities.

These security requirements form the foundation for a robust security framework, ensuring the TruSec surveillance system's resilience against potential threats, protecting user privacy, and complying with industry best practices and regulations.

## Additional Software Quality Attributes

In addition to functional and performance requirements, the TruSec surveillance system must adhere to specific quality attributes that are crucial for both customers and developers. These attributes contribute to the overall effectiveness, reliability, and maintainability of the system.

**Reliability:** The system should achieve a minimum uptime of 99.9%. Ensures continuous and reliable operation, minimizing downtime and disruptions in surveillance.

**Maintainability:** The codebase must be well-documented, and updates should be deployable with minimal system downtime. Facilitates easy maintenance and updates, reducing the risk of errors and ensuring the system's adaptability to evolving requirements.

**Usability:** The admin dashboard should have a usability score of at least 80% based on user feedback and usability testing. Enhances user satisfaction and efficiency, promoting effective monitoring and management of the surveillance system.

**Interoperability:** The system should be compatible with major web browsers (Chrome, Firefox, Safari) for the admin dashboard. Ensures seamless integration with commonly used platforms, enhancing user accessibility and flexibility.

**Scalability:** The system should scale to support multiple trucks without a significant degradation in performance. Supports the system's adaptability to growing demands and ensures responsiveness even as the number of monitored trucks increases.

**Testability:** The system should have a test coverage of at least 90% for critical functionalities. Facilitates effective testing and quality assurance, reducing the likelihood of undetected bugs and ensuring software reliability.

**Adaptability:** The system should be adaptable to changes in environmental conditions, such as variations in RF signal strength or temperature. Enhances the system's resilience in diverse operating conditions, ensuring consistent performance in different scenarios.

**Flexibility:** The system architecture should support the integration of additional sensors or communication modules without requiring major code modifications. Enables flexibility in system expansion and customization, supporting future enhancements and upgrades.

# Other Requirements

In addition to the previously specified requirements, there are several miscellaneous requirements that play a crucial role in the successful development and deployment of the TruSec surveillance system:

**Database Backup and Recovery:** Implement a regular backup mechanism for the SQL database to ensure data integrity and facilitate efficient recovery in case of data loss or system failures.

**External Interface Compatibility:** Ensure compatibility with common external interfaces, such as USB ports, to facilitate potential future hardware additions or modifications on the Raspberry Pi 4 devices.

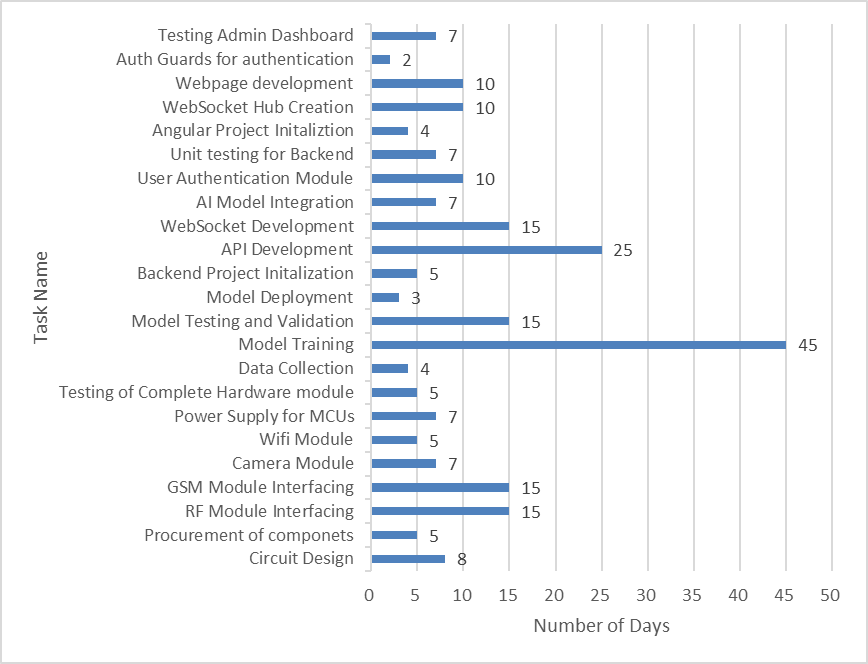
**Documentation:** Provide comprehensive documentation, including user manuals, installation guides, and API documentation, to support users, administrators, and developers in understanding and utilizing the system effectively.

**Reuse Objectives:** Design components with a modular and reusable architecture, facilitating the potential reuse of code segments for future projects or system expansions.

# Revised Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Start Date** | **Days Required** | **Due Date** | **Status** |
| Circuit Design | 17-10-2023 | 8 | 24-10-2023 | In Progress |
| Procurement of components | 25-10-2023 | 5 | 29-10-2023 | Completed |
| RF Module Interfacing | 30-10-2023 | 15 | 13-11-2023 | Completed |
| GSM Module Interfacing | 14-11-2023 | 15 | 28-11-2023 | Completed |
| Camera Module | 29-11-2023 | 7 | 05-12-2023 | In Progress |
| WIFI Module | 06-12-2023 | 5 | 10-12-2023 | Completed |
| Power Supply for MCUs | 11-12-2023 | 7 | 17-12-2023 | In Progress |
| Testing of Complete Hardware module | 18-12-2023 | 5 | 22-12-2023 | On Hold |
| Data Collection | 23-12-2023 | 4 | 26-12-2023 | In Progress |
| Model Training | 27-12-2023 | 45 | 09-02-2024 | In Progress |
| Model Testing and Validation | 10-02-2024 | 10 | 19-02-2024 | On Hold |
| Model Deployment | 20-02-2024 | 3 | 22-02-2024 | On Hold |
| Backend Project Initialization | 23-02-2024 | 5 | 27-02-2024 | In Progress |
| API Development | 28-02-2024 | 25 | 23-03-2024 | In Progress |
| WebSocket Development | 24-03-2024 | 12 | 04-04-2024 | On Hold |
| AI Model Integration | 05-04-2024 | 5 | 09-04-2024 | On Hold |
| User Authentication Module | 10-04-2024 | 5 | 14-04-2024 | On Hold |
| Unit testing for Backend | 15-04-2024 | 7 | 21-04-2024 | On Hold |
| Angular Project Initialization | 22-04-2024 | 4 | 25-04-2024 | In Progress |
| WebSocket Hub Creation | 26-04-2024 | 10 | 05-05-2024 | On Hold |
| Webpage development | 06-05-2024 | 10 | 15-05-2024 | On Hold |
| Auth Guards for authentication | 16-05-2024 | 2 | 17-05-2024 | On Hold |
| Testing Admin Dashboard | 18-05-2024 | 7 | 24-05-2024 | On Hold |

## Gantt Chart



# References

[1] “FER-2013 Facial Expressions Dataset.” <https://www.kaggle.com/datasets/msambare/fer2013>

[2] “TensorFlow lite support for microcontrollers.” <https://blog.tensorflow.org/2020/08/announcing-tensorflow-lite-micro-esp32.html>

[3] “ESP32 Wireless Communication Protocols.” <https://linuxhint.com/esp32-wireless-communication-protocols/>

Appendix A: Glossary

API: Application Programming Interface

SQL: Structured Query Language

RF: Radio Frequency

MCU: Micro-controller Unit

GSM: Global System for Mobile communication

USB: Universal Serial Bus

IoT: Internet of Things

RESTful: Representational State Transfer

TCP/IP: Transmission Control Protocol/Internet Protocol

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

PCB: Printed Circuit Board

AI: Artificial Intelligence

ACID: Atomicity, Consistency, Isolation, and Durability

GPS: Global Positioning System

AWS: Amazon Web Services

EC2: Amazon Elastic Compute Cloud

GPIO: General Purpose Input/Output

SPI: Serial Peripheral Interface

I2C: Inter-Integrated Circuit

UART: Universal Asynchronous Receiver/Transmitter

Appendix B: IV & V Report

**(Independent verification & validation)**

**IV & V Resource**

Name Signature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Defect Description** | **Origin Stage** | **Status** | **Fix Time** | |
| **Hours** | **Minutes** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| … |  |  |  |  |  |

**Table 2: List of non-trivial defects**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Semester wise SDP Meeting Report** | | | | | | | | | |
|  | Project Title: TruSec – Trucking Surveillance System | | | | | | Semester (e.g. S16): | | | F23\_\_\_\_\_\_\_\_\_\_ |
|  | **Group ID** | **Student Roll Number** | | | **Student Name and Signatures** | | | | | **Advisor** |
|  | F23CS112 | LF19BSCS0139 | | | Ahmed Naeem | | | | | Syed Nisar Ali Balti |
|  | LF19BSCS0582 | | | Najam Irfan | | | | |
|  |  |  | | | | |  |  |  | |  |
| Sr. | Date | Status (P/A/L) | | | Agenda Items | | | | | Notes |
|  |  | Ahmed Naeem | Najam Irfan |  | |  | | | |  |
| 1 | 18-10-2023 | P | P |  | | Project Initialization Discussion | | | | Identify potential challenges for RF Communication |
| 2 | 23-10-2023 | P | P |  | | Discussion of User Flow | | | | Create ERD and Class Diagrams |
| 3 | 03-11-2023 | P | P |  | | PCB Design | | | | Make PCB Model |
| 4 | 15-11-2023 | P | P |  | | Demonstrate the PCB | | | | Advised to change the Transmitter MCU to Raspberry Pi 4 |
| 5 | 23-11-2023 | P | P |  | | Review of PCB and Domain Models | | | |  |
| 6 | 14-12-2023 | P | P |  | | Started working on SRS | | | |  |
| 7 | 18-12-2023 | P | P |  | | Review of SRS Document | | | | Start working on RF communication |
|  |  |  |  |  | |  | | | |  |
|  |  |  |  |  | |  | | | |  |
|  | Date: 21-12-2023 |  | Advisor's Signatures: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | |  |