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 iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 Objective(s)/Aim(s)/Target(s) 1

1.6 Challenges 1

1.7 Learning Outcomes 1

1.8 Nature of End Product 1

1.9 Completeness Criteria 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentations 2

2.7 Assumptions and Dependencies 3

3. Product Features / Functional Requirements 4

3.1 Name of Use-Case 1 4

3.2 Name of Use-Case 2 (and so on) 4

3.3 Requirement 3 (and so on) 4

3.4 Analysis and Modeling of Requirements 5

4. External Interface Requirements 5

4.1 User Interfaces 5

4.2 Hardware Interfaces 5

4.3 Software Interfaces 5

4.4 Communication Interfaces 5

5. Other Nonfunctional Requirements 6

5.1 Performance Requirements 6

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Additional Software Quality Attributes 6

6. Other Requirements 6

7. Revised Project Plan 6

8. References 7

Appendix A: Glossary 8

Appendix B: IV & V Report 9

# 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

Avoid using jargon or technical language that may be unfamiliar to the reader. Instead, use simple, straightforward language to clearly communicate the requirements. Organize the document using headings and subheadings to help the reader navigate and understand the content. Use bullet points or numbered lists to clearly present lists of items or requirements. For heading and paragraph font Times and Times New Roman respectively with font size. Headings are bold.

**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

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## 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% |

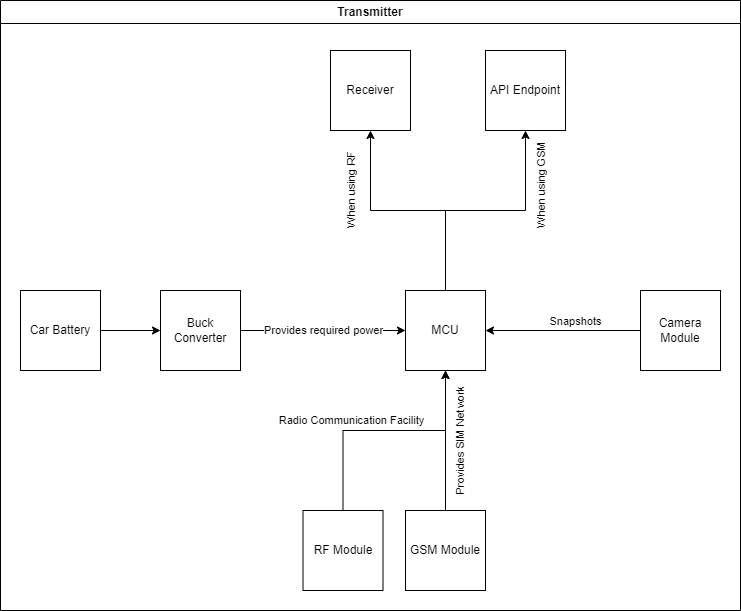
# Overall Description

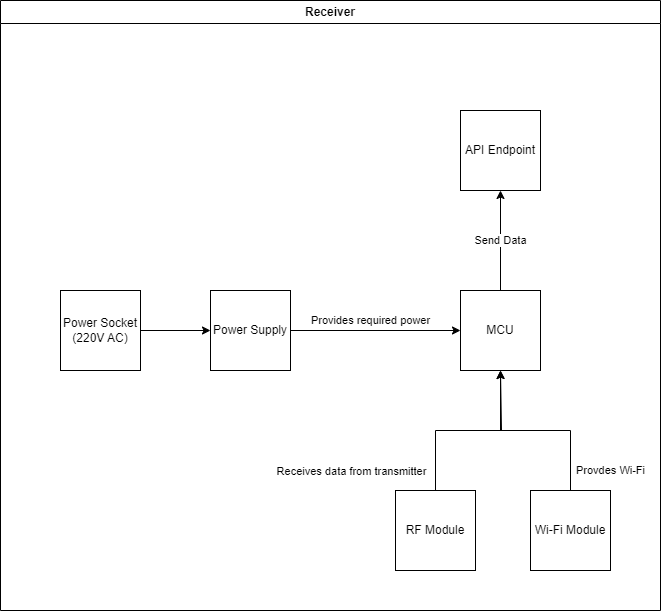
## Product Perspective

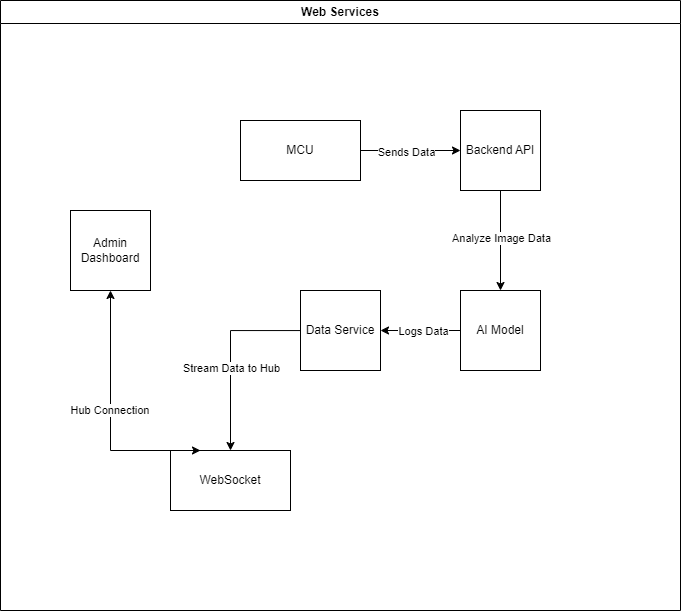
The project aims to provide a practical end-to-end solution for monitoring trucks carrying sensitive consignments in remote areas. By integrating GPS tracking, IoT sensors, surveillance cameras, communication networks, and data analytics, this solution offers real-time visibility, enhanced security, 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 1: 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

<Include the following analysis models: use-case diagram (provide before Section 3.1), entity-relationship diagram, abstract class diagram, sequence diagram (to model interactions between system and external world). Additional diagrams may be added for example state diagram, data flow diagram (to model interactions between system and external world), decision table, flow chart, circuit diagram, event table etc.>

# 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

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communication Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Additional Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. These might include database requirements, external (hardware, software, or communication) interface requirements, internationalization requirements, legal requirements, and reuse objectives for the project.>

# Revised Project Plan

<Provide current status of the project in accordance with the plan provided in project proposal. Gantt chart should be used in this regard. Use Microsoft Office to develop the Gantt chart. Also provide an updated project plan.>

# References

*<List all books, conference papers, journal articles, websites, etc. used in preparing the content of this SRS. Provide enough information so that the reader could access a copy of each reference, including title, author, volume/edition number, page number(s), and publication year. Mention complete URLs for websites.>*

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

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 3: List of non-trivial defects**