Sri Lanka Institute of Information Technology

Smart Breathalyzer Device

Software Requirement Specification

Professional Engineering Practice and Industrial Management - IE2090

Project ID: PEP_07

Submitted by:

Registration #	Student Name
IT22088000	W. Y. M. G. Thamasha
IT22073228	D. W. A. I. Abeynayake
IT22073082	E. M. B. V. Bandara
IT22081520	K. M. D. N. B. Ariyasena

Submitted to:

(Supervisor's signature)

……..

Name of the supervisor

Date of submission

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The Software Requirements Specification (SRS) document is a tool for laying out the plan for developing an embedded-based smart breathalyzer device, serving as a communication medium between stakeholders, defines the project scope, outlines functional requirements such as adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability and includes non-functional requirements like performance, reliability, security, usability, and scalability. The Software Requirements Specification also outlines constraints and assumptions that may impact the system's development or usage, such as hardware limitations or regulatory assumptions. It also documents interfaces between components, ensuring seamless integration and interoperability. The SRS serves as a basis for validation and verification activities, ensuring the final product meets the specified requirements and user needs.

1.2 Document Conventions

Chapter title – font size – 18, bold, font – times new roman
Chapter subtitle - font size-14, bold, font -times new roman
Description - font size – 12, font – times new roman

1.3 Intended Audience and Reading Suggestions

Intended Audience	**Reading suggestion**
Client	From introduction
Developers	From internal interface requirements

1.4 Product Scope

The Smart Breathalyzer Device project aims to enhance road safety by developing a device that analyzes a person's breath for alcohol levels, providing timely alerts to prevent accidents caused by drunk driving. The project includes the creation of a user-friendly mobile application to accompany the device, catering to both drivers and law enforcement officers. The device will detect the alcohol level and notify the user through the mobile application. Utilizing available sensors, the device ensures precise alcohol detection for personal and professional use. Better security, ease of use, mental clarity, and energy economy are among the advantages. Reliability, compatibility, security features, and an easy-to-use interface are among the goals of the software. Innovating, customer-focused solutions and utilizing technology to tackle new market trends are corporate goals that the system supports.

1.5 References

- *https://rathburnlaw.com/breathalyzer-test-limitations/*

2. Overall Description

2.1 Product Perspective

The Smart Breathalyzer Device specified in this Software Requirement Specification(SRS) is a self-contained product designed to ensure responsible alcohol consumption and enhance the road safety. This project originated from the need to reduce the road accidents that happen due to drunk driving. The device contains advance sensor technology with wireless connectivity ensuring accurate data transmission to the mobile application for real time feedback and user interaction.
Major Components of this device includes a mp3 alcohol gas sensor for the alcohol level detection, an ESP8266 module for the data processing and communication, a LED bulb and a speaker for alerts, and a digital screen for monitor the battery percentage of the device. A Blynk application is used as the user interface where the communication between the device and the user is established.

2.2 Product Functions

1) Implementing Alcohol Sensor
    - Connect the alcohol sensor to the ESP8226 module.
    - Write code on the ESP8226 to interact with the alcohol sensor.
    - Calibrate the sensor for accurate alcohol level detection.
    - Create a logic to process sensor data and give feedback.

2) ESP8226 Module Implementation
    - Combine the ESP8226 module with the hardware.
    - Develop firmware to manage the ESP8226 and communicate with other parts.
    - Enable Wi-Fi for remote monitoring and control
    - Ensure compatibility with the mobile app for seamless use.

3) LED Bulb Implementation
    - Attach the LED bulb to the microcontroller.
    - Write code to control the LED bulb based on sensor data.

- Make sure the LED bulb works seamlessly with other components for synchronized feedback.

4) Speaker Implementation
    - Attach the speaker to the microcontroller.
    - Develop code to play audio alerts and notifications.
    - Make certain that feedback is heard and clear for users.

5) LED Display Implementation
    - Connect the LED display to the microcontroller.
    - Write code to show battery percentage on the display.
    - Make sure the display is clear and easy to read for users.

6) Creating a Mobile Application
    - Make the mobile app easy to use.
    - Add features to track alcohol levels in real time.

2.3 User Classes and Characteristics

Drivers:

Drivers utilize the Smart Breathalyzer Device to guarantee their safety and the safety of others while driving. They come from diverse backgrounds in terms of technical skills but are able to effortlessly navigate the device due to its user-friendly design. They depend on the device to give precise alcohol level measurements and value the instant notifications received vis the mobile app.

Police Officers:

Police Officers use the Smart Breathalyzer Device to detect and deal with cases of drunk driving. They are skilled experts who can analyze device readings and respond accordingly. Police Officers depend on the device for precise and trustworthy alcohol level readings, which help in enforcing traffic safety regulations.

2.4 Operating Environment

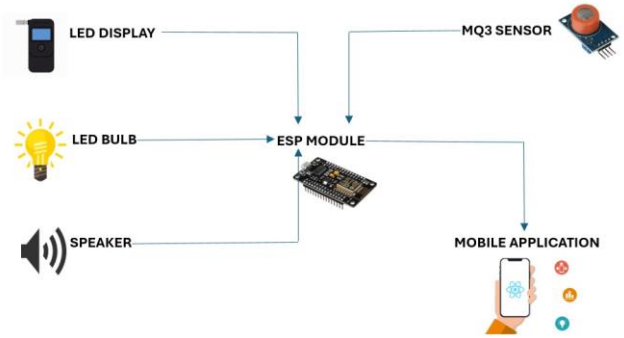
Hardware:

The Smart Breathalyzer device's hardware platform includes key components like an MP3 alcohol gas sensor and an ESP8226 module. The MP3 alcohol gas sensor accurately detects the alcohol level in user-provided breath samples. The ESP8266 module servers as the central microcontroller, overseeing sensor data purchasing, processing, and communication tasks. In addition, the device features an LED bulb for visual status indication, a speaker for audible alerts, and a digital screen for displaying battery percentage. These hardware components collaborate to allow the device to measure alcohol levels, offer user feedback, and manage power effectively. The combination of these parts creates a sturdy base for the smart breathalyzer device, making it easier to use and connect with the user and other systems.

Software:

The smart breathalyzer device's software platform includes various essential components to make it work. The ESP8266 module is programmed with Arduino-compatible firmware, using libraries like ESP8226WiFi and ESP8226HTTPClient for networking function. The firmware handles sensor connections, sending data to the Blynk mobile app, and other system operations. The Blynk app acts as the mobile interface, receiving sensor information via WI-FI from the ESP8266 module. It works on both IOS and Android, showing alcohol level readings and driving safety details in a user-friendly way. When you use Blynk, it makes it easy for your device to talk to your phone in real-time. This allows you to keep an eye on things and get alerts when needed. The software we have put together covers everything from creating the firmware for the device to integrating it with the mobile app and setting up the network protocols. It is complete and efficient solution for detecting alcohol levels and making sure everyone stays safe.

2.5 Design and Implementation Constraints



The accuracy of the test can be affected by how well it is calibrated. If the machine is not properly calibrated, it may give inaccurate results.

The test can be affected by how deep a person's breaths are. If a person takes shallow breaths, the machine may not be able to get an accurate reading.

The test can be affected by what a person has eaten or drunk before taking the test. Certain foods or drinks (such as mouthwash) can cause the machine to give a false positive reading.

The sensor value does not represent the exact gas concentration. It reflects the approximated trend of gas concentration in a permissible error range.

The detection of certain components in the air usually requires a more precise and costly instrument, which cannot be done with a single gas sensor.

2.6 Project Documentation

The smart breathalyzer device is intended to give you accuracy in measuring alcohol levels while also being convenient. As it is a variation of the method used by the law enforcement already it has a very niche use, however this model is miles ahead of the current method in terms of accuracy, convenience, operability, reliability.

2.7 Assumptions and Dependencies

The sensor value does not represent the exact gas concentration. It reflects the approximated trend of gas concentration in a permissible error range. The detection of certain components in the air usually requires a more precise and costly instrument, which cannot be done with a single gas sensor.

3. External Interface Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

- ESP8266 Wi-Fi module
- MQ-3 alcohol gas sensor
- 3.3v linear regulator
- 9v battery
- LED Battery
- Speaker
- LED

3.3 Software Interfaces

- Arduino IDE 2.3.2
- Blynk app

3.4 Communications Interfaces

- ESP 8266 Module uses Wi-fi

4.System Features

4.1 System Feature 1

F1	Alcohol Level Detection
Input	Positive sensor reading from the mp3 alcohol gas sensor
Process	Identifying whether the alcohol level is higher or lower that the legal limit
Output	Detect the alcohol level
Definition	Legal limit – The maximum amount of alcohol allowed before driving (0.08%)

F2	Implementing Wi-Fi Communication
Input	Digital alcohol concentration data from the ESP module
Process	Format the alcohol concentration data for transmission over Wi-Fi and establish a Wi-Fi connection with the mobile application
Output	Send the formatted data to the mobile application

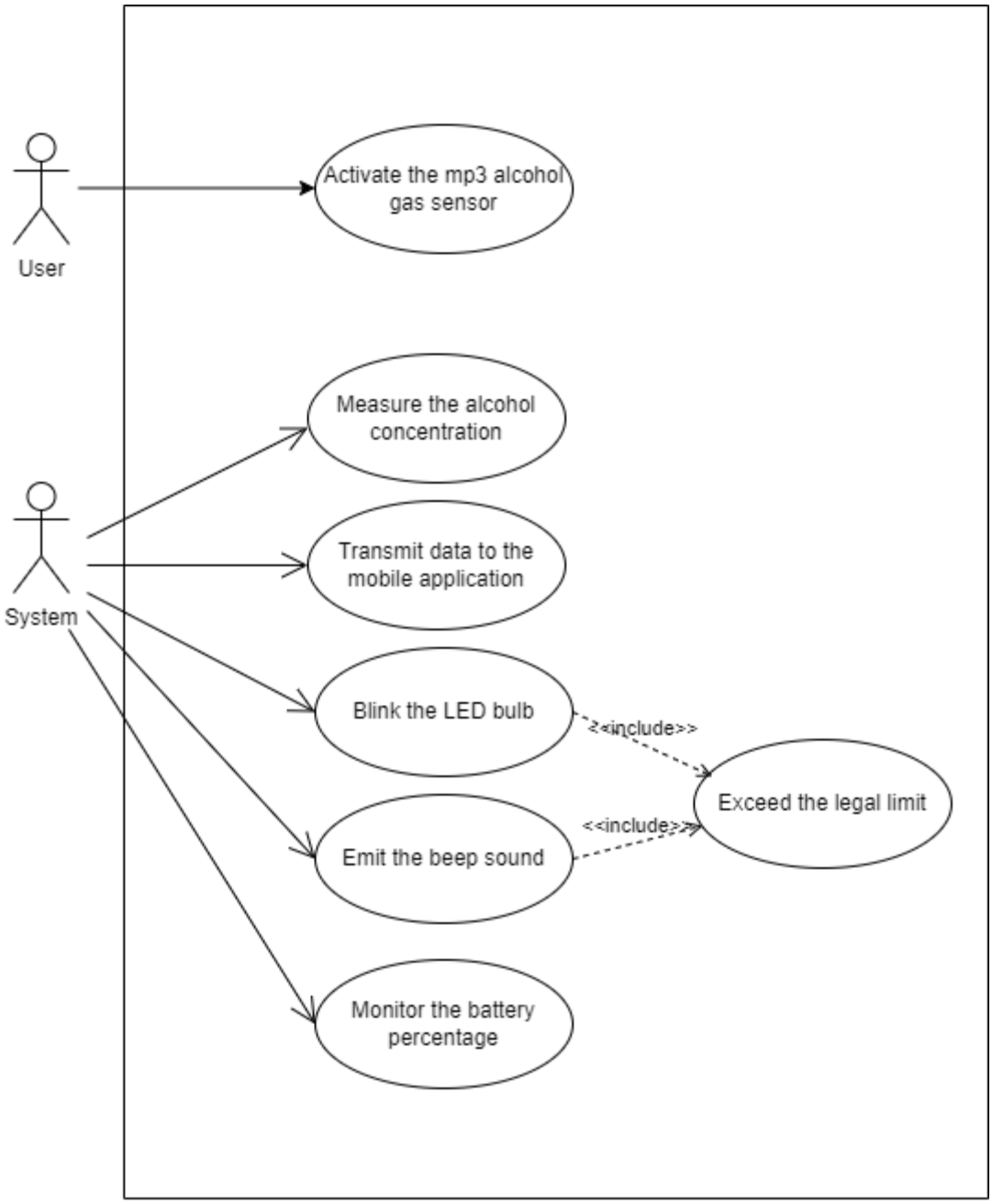
F3	Displaying the data in the mobile app interface
Input	Alcohol concentration data received via Wi-Fi communication
Process	Format the data for display on the mobile app interface
Output	Display the alcohol level with a message which indicates whether it is safe to drive or not

F4	LED Alert Activation
Input	Sensor reading from the mp3 alcohol gas sensor
Process	Compare the alcohol level with the legal limit
Output	Active the LED to turn on and blink if the alcohol level exceeds the legal limit

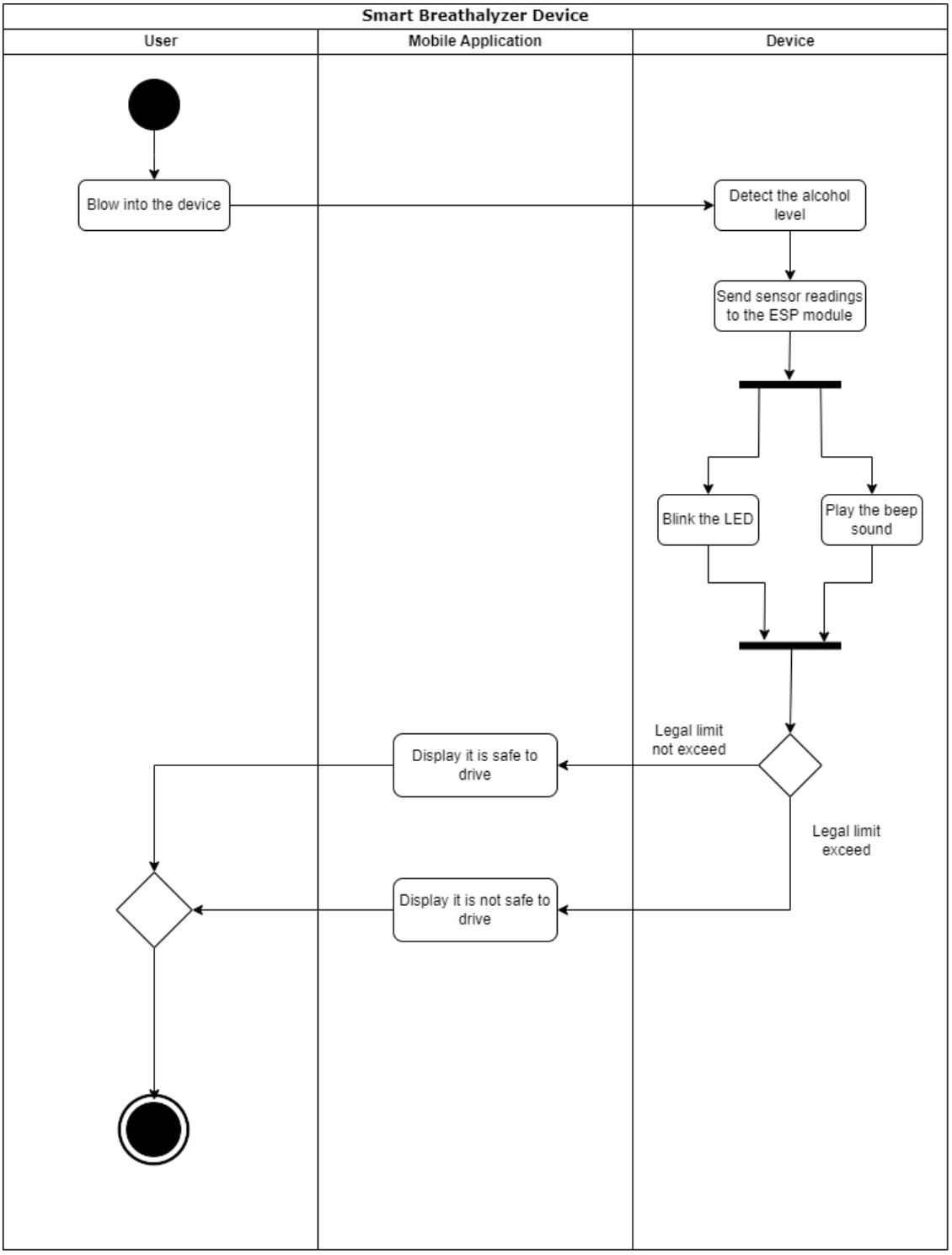
F5	Speaker Alert Activation
Input	Sensor reading from the mp3 alcohol gas sensor
Process	Compare the alcohol level with the legal limit
Output	Emit a beep sound from the speaker if the alcohol level exceeds the legal limit

F6	Displaying the Battery Percentage
Input	Battery voltage level measured by the ESP module
Process	Convert the battery voltage level to a percentage representing the remaining battery capacity
Output	Display the calculated battery percentage on the digital screen of the device

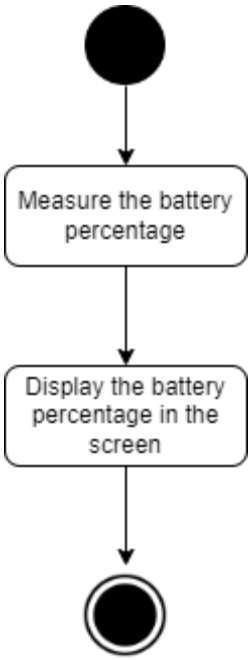
4.2 System Feature 2 : Use Case Diagram



4.3 System Feature 3 : Activity Diagram



Activity Diagram for the LED screen



5.Other Nonfunctional Requirements

5.1 Performance Requirements

- Alcohol sensor
- Wi-fi module
- Mobile Application

5.2 Safety Requirements

- The smart breathalyzer device should be designed to minimize physical harm to users and property occupants, ensuring secure mounting of hardware components like sensors and nozzle. Safety measures, such as isolation of the user's breath from other external influences should prevent accidental readings and false positives.
- The device is designed to withstand environmental factors such as temperature variations (-10°C to 40°C), humidity levels (10% to 90% RH), and mechanical shocks (up to 1 meter drop) without compromising its functionality or accuracy.

5.3 Security Requirements

The intelligent breathalyzer device must possess robust security features, ensuring impregnability against external interference or tampering attempts. Its design should incorporate advanced encryption protocols and authentication mechanisms, guaranteeing the integrity of data collection and analysis processes. This ensures utmost reliability in determining accurate blood alcohol content levels, fostering trust in its results among users and authorities alike.

5.4 Software Quality Attributes

The smart breathalyzer device can achieve a reliability metric of at least 99.9%, ensuring that it accurately detects blood alcohol content levels with minimal false positives or false negatives during regular operation.

The device facilitates easy maintenance and servicing, with a mean time to repair (MTTR) of no more than 30 minutes for common issues, allowing for swift resolution and minimal downtime.

The device's user interface adheres to established usability guidelines, achieving a System Usability Scale (SUS) score of at least 75 out of 100, ensuring intuitive operation and minimizing user errors during testing procedures.

The device is lightweight and compact, weighing no more than 300 grams and easily fitting into standard carrying cases or vehicle compartments, facilitating convenient transport and deployment in various settings.

The device is compatible with commonly used operating systems (e.g., iOS, Android) and communication protocols (e.g., Bluetooth, Wi-Fi), enabling seamless integration with existing infrastructure and enabling data sharing across platforms.

The device supports firmware updates and software upgrades to accommodate evolving regulatory requirements and technological advancements, ensuring long-term viability and compatibility with future standards.

The device maintains a high availability rate of at least 99%, minimizing downtime due to maintenance, servicing, or unforeseen technical issues, thereby ensuring continuous accessibility for users.

The device's measurement accuracy meets or exceeds industry standards, with a maximum allowable margin of error of ±5% compared to laboratory-grade equipment, ensuring confidence in its results among users and regulatory authorities.

Appendix A: Glossary

- Blynk app - A complete software suite called Blynk makes it possible to prototype, implement, and remotely manage linked electronic devices of any size.