Software Requirements and Specifications

**Project Name: SPARQ: Spark-powered Audio Recognition and Querying**

**Aligned SDG: 11. Sustainable Cities and Communities**

**Team Members: 1. Shaik Bavikadi al Hafeez**

**2. Srikanth Gangishetty**

**3. Sai Kumar Chandu**

**4. T Vineeth Kumar**

**Mentor Name: Dr. Purushotham Muniganti**

**Contents**

[1. Introduction 3](#_Toc141781869)

[1.1 Purpose and Intended Audience 3](#_Toc141781870)

[1.2 Project Scope 3](#_Toc141781871)

[1.3 Terms, Definitions, and Acronyms 4](#_Toc141781872)

[1.4 References 4](#_Toc141781873)

[2. Overall Description 4](#_Toc141781874)

[2.1 Product Perspective 4](#_Toc141781875)

[2.2 Product Features 4](#_Toc141781876)

[2.3 User Classes and Characteristics 5](#_Toc141781877)

[2.4 Operating Environment 6](#_Toc141781878)

[2.5 Design and Implementation Constraints 6](#_Toc141781879)

[2.6 Assumptions 6](#_Toc141781880)

[3. Specific Requirements 6](#_Toc141781881)

[3.1 Functional Requirements. 6](#_Toc141781882)

[3.2 Non-Functional Requirements 7](#_Toc141781883)

[3.3 External Interface Requirements 7](#_Toc141781884)

[3.3.1 User Interfaces 7](#_Toc141781885)

[3.3.2 Hardware Interfaces 7](#_Toc141781886)

[3.3.3 Software Interfaces 7](#_Toc141781887)

**References**

# Introduction

The subsections under this section are a brief overview of the Software Requirements and Specifications (SRS) document for the application program titled “**SPARQ: Spark-powered Audio Recognition and Querying**”. The document highlights various tools, methodologies, techniques and approaches that have been used for the completion of the program.

## Purpose and Intended Audience

This document provides a software requirements specification (SRS) for above mentioned project. The purpose of the system is that it can detect the sound of sparks from raw audio data. It is made with the intention of it being:

* A provision containing a brief overview of the application to individuals.
* A means of communication between the team members and the mentor.
* A documented assistance with frontend, backend, software design and quality control tasks.
* A document highlighting the validating system testing activities.
* A document that visualizes supporting Verification and Validation activities.

## Project Scope

This application has the scope of being a platform for individuals who gets a response in case of any spark detection. The individual can be from any background, including industrial, domestic, or any other research-orientated group.

* With the usage of this application, individuals will be notified in case of any sparks which can lead to any property damage/loss of life or any machine damage.
* Currently, the built of the application is for prototype purpose only, however it has the potential and the scope to scale into an enterprise edition, wherein it shall be usable by communities and industries for more large-scale tracking of spark detection, if there are any.

## Terms, Definitions, and Acronyms

This document is prepared according to the IEEE SRS Standards and uses technical terminology, different formats of text and abbreviations to provide clarity and distinction to its readability.

* **Bold** text refers to headings
* **SRS:** Software Requirement Specification
* **SPARQ:** Spark-powered Audio Recognition and Querying
* **Project:** refers to the application program that helps in detection of spark from raw data and using machine learning to help the user to take any safety measures in case of a higher spark.

## References

* IEEE Recommended Practice for Software Requirements Specifications by IEEE Computer Society (20 October 1998)
* [IEEEXplore-SRS-template.pdf](file:///C:\Users\hi\Downloads\IEEEXplore-SRS-template.pdf)

# Overall Description

This section provides a background for the requirements in the making of the application program to make it layman friendly while also as a stencil for the development team to stick to (in the preparation of the project).

## Product Perspective

This project helps users to get notified in case of any spark in their surroundings. It can lead to the development of a spark sound event detection system.

The giveaway of this segment is that it uses Waveform analysis and for audio classification to perform its main functionality, which is to help the user save any machinery or any property from getting damaged. The target audience for our application are people in domestic, industrial sectors.

Payment

Processing

## Product Features

This section discusses some of the features that define the many functionalities of the application program. They are listed as:

* Consists of simple led showing real-time spark detection or a buzzer giving alerts.
* Visualization tools powered by Python to make the data comprehensible and flexible to train the system to max the accuracy.
* SparkFun RedBoard Artemis ATP –classify spark sound with the board.
* Arduino Due – to control the electric spark generation.
* SparkFun Micro OLED Breakout (Qwiic) – employing an OLED screen to display classification results.
* SparkFun RedBot Buzzer.

## User Classes and Characteristics

There is only one main class of users for the application program, while keeping in mind its built to prototype level only.

1. Users: Users must be active with device and will be notified at the time of spark detection.

## Operating Environment

The operating environment for application would be the SparkFun RedBoard Artemis ATP board which detects the spark and displays a message on the screen.

The application is compatible in Operating System of Windows 10 (or) higher.

## Design and Implementation Constraints

* Doesn’t require any internet access for the device.
* Interface is available only in English.
* Requires a power connection to the Artemis ATP board for working.
* Need a device which can display the desired output or a buzzer which gives alerts.

## Assumptions

* Every user shall be having the appropriate hardware and software components as per the requirements.
* All variables will be flexible with Waveform Analysis for classification used for Spark sound detection.

# Specific Requirements

This section features key requirements that the application program must fulfill.

## Functional Requirements.

The application has various performative functionalities to it, which shall include the following:

* Users can access the notifying system which uses SparkFun RedBoard Artemis ATP board to recognize and classify sound data.
* The system should be able to integrate the Artemis board with the user’s device.

## Non-Functional Requirements

* The project shall be: Reliable: manages to become convenient and helpful to detect any spark activities
* Scalable: the program carries potential to scale to enterprise level and until then it can allow usage for limited users.
* User-friendly: simple yet robust
* Accurate
* Maintenance-free: does not require any kind of maintenance, is completely stand-alone

## External Interface Requirements

While keeping in mind the possibility of the application being enterprise-appropriate, each individual, belonging to either of the user classes, shall have a distinctive access to this platform, paired with unique IDs to access. Other interfaces described below are:

### User Interfaces

\*Refer to section 2.2 Product Features\*

### Hardware Interfaces

\*Refer to section 2.2 Product Features\*

### Software Interfaces

There are various tools and techniques that shall be used for the creation of this project. The tools that will be using majorly are:

* Python 3 and a local programming environment set up on your computer
* Jupyter Notebook/PyCharm or any python console.
* Microsoft Excel: to store the user data in a structured format, and increased readability of .csv files via Python.
* Modules to be installed: pyAudio, Wave, Matplotlib, IPythonDisplay, Librosa, Scikit-learn, CNN, NumPy, Tenserflow, Keras, PySerial
* Arduino IDE
* Legacy Arduino

**REFERENCES:**

https://www.hackster.io/324122/spark-sound-detection-from-raw-audio-data-ab60b9

https://www.hackster.io/

https://ieeexplore.ieee.org/Xplore/home.jsp

https://medium.com/

https://www.hackster.io/macwirecom/harmonic-sounds-a-symphony-of-sustainable-recycling-5382e4

https://www.kaggle.com/code/devilsknight/sound-classification-using-spectrogram-images

https://towardsdatascience.com/machine-learning-for-anomaly-detection-and-condition-monitoring-d4614e7de770

https://www.hackster.io/noah16/ai-powered-trash-can-dd46a8

https://towardsdatascience.com/advanced-waste-classification-with-machine-learning-6445bff1304f

https://medium.com/mlearning-ai/audio-classification-using-wavelet-transform-and-deep-learning-f9f0978fa246