**Spectrum Analyzer Tool**

**Design Document**

# 

Chris Williams, George Vendeta, Seema Kumaran, Jeffrey Amakihe, Alexandro Pinion

Date: 10/25/2023

Department: Computing and Software Engineering

Course: SWE 7903 - Capstone

Professor: Dr. Parizi

**Table of Contents**

[1. INTRODUCTION 2](#_Toc148874040)

[1.1 Purpose 2](#_Toc148874041)

[1.2 Scope 2](#_Toc148874042)

[1.3 Overview 2](#_Toc148874043)

[1.4 Reference Material 2](#_Toc148874044)

[1.5 Definitions and Acronyms 2](#_Toc148874045)

[2. SYSTEM OVERVIEW 3](#_Toc148874046)

[3. SYSTEM ARCHITECTURE 4](#_Toc148874047)

[3.1 Architectural Design 4](#_Toc148874048)

[3.2 Design Rationale 7](#_Toc148874049)

[4. DETAILED DESIGN 7](#_Toc148874050)

[5. DATA (DATABASE) DESIGN 7](#_Toc148874051)

[6. HUMAN INTERFACE DESIGN 7](#_Toc148874052)

[6.1 UI design 7](#_Toc148874053)

[6.2 UX design 7](#_Toc148874054)

[7. APPENDICES 7](#_Toc148874055)

### INTRODUCTION

## Purpose

The purpose of this System Design Document (SDD) is to provide an extensive blueprint that describes the architecture and design of a software. Its main goal is to give a precise and comprehensive explanation of the software's architecture, design, component interactions, and means of meeting the requirements. The SDD fulfills several crucial functions. It serves as a communication tool to assist developers, testers, project managers, and other team members in comprehending the architecture of the software system. Throughout the software development lifecycle, the development team uses it as a reference document. It facilitates the documentation of design choices and justifications. The intended audience for this SDD is for developers, Robins AFB stakeholders, testers and project managers.

## Scope

The goal of this Software project is to create, test, and implement a user-friendly program that the Robins AFB can use to solve video analysis and spectrum analyzer difficulties. An independent deployment environment for the client's platform, formatted CSV data representing spectrum analyzer signals, and image signal processing of video are some of the key features. The project's goal is to provide a reliable, scalable solution that addresses the client's issue within the allocated spending limit and time frame.

## Overview

This document will go over the system architecture, data flow, user interfaces, algorithms, data structures, security considerations, error handling, and other subjects are covered in detail in this document. It serves as an essential means of communication for development teams, testers, project managers, and stakeholders, guaranteeing that all parties participating in the project comprehend the functioning and design of the product. An integral part of the software development lifecycle, the SDD also acts as a point of reference for scalability and future maintenance.

## Reference Material

*This section is optional.*

List any documents, if any, which were used as sources of information for the test plan.

## Definitions and Acronyms

*This section is optional.*

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

### SYSTEM OVERVIEW

As a key aviation hub, Robins Air Force Base (AFB) uses outdated Spectrum Analyzers to measure the quality of transmission signals within their ground stations. These Analyzers gather information while in flight, producing hours' worth of crucial data that is necessary to preserve the integrity of communications networks. However, there are operational inefficiencies and a high human resource cost associated with the current manual procedure of examining video recordings of these datasets for anomalies in signal quality. In response to this challenge, KSU's Master of Software Engineering Capstone students have been requested by Robins AFB to develop an automated system for the analysis of flight data. This will guarantee prompt identification of problems with signal quality while optimizing operations and resource use. This group of students behind this project will create an open-source standalone application that will operate on the computers that are intended for use by Robins AFB personnel. The application will offer a guided process for loading relevant data, processing it, and producing output resources that will be most helpful to the stakeholders at Robins AFB.

### SYSTEM ARCHITECTURE

## Architectural Design

This section includes a high-level structure design that outlines the main elements, how they relate to one another, and how the application is organized overall. It acts as a construction blueprint for the software, assisting developers in comprehending how different system components interact to produce the intended functionality. Important non-functional needs like performance, maintainability, and scalability are covered. This section will also provide a high-level road map that directs the development team in creating a system that satisfies the project's functional and quality objectives while taking flexibility and maintenance ease into account is the aim of architectural design. For the Spectrum Analyzer Tool application to remain viable and successful over the long term, a well-designed architectural design is essential. Below are the main components of the systems architecture:

* **User Interface (UI) Layer**:
  + The desktop application's user interface is known as the UI.
  + It offers an easy-to-use interface so people can engage with the program.
  + It has options for image processing, video data display, and file input for choosing video files.
  + Additionally, the UI provides controls for exporting CSV files and starting the data conversion process.
* **Logic Layer**:
  + Video Processor:
    - This part extracts pertinent data by processing the loaded video frames.
    - It could involve filtering, feature extraction, picture analysis, and other image processing methods.
    - The data has been processed and is ready to be converted to CSV format.
  + This layer will perform the business logic for the application.
* **Data Management Layer:**
  + CSV Data Converter:
    - This module creates a CSV file from the processed video data.
    - It arranges the information in a tabular style that can be exported as CSV.
    - The CSV format may be configured by users with options to set headers and delimiters, for example.
  + This layer will be responsible for creating all of the non-volatile data storage for the application.
* **Exceptions and Logging Layer**:
  + Put in place an error-handling system to handle unforeseen problems that arise when loading, processing, or converting videos.
  + Produce logs for reporting errors and debugging.
* **Distribution Layer**:
  + Builds all packages and bundles all preset libraries, dependencies, and components that makes it simpler to deploy the application to the user.
  + Generates a standalone executable that operates independently on a computer system without the requirement for extra resources or dependencies.

The Figure 1 below will showcase an overview diagram of the architectural design for the Spectrum Analyzer Tool and how each of the components in the application provide key functionality. Each of the diagram’s components are described above.

A diagram of a software application

Description automatically generated

Figure - Architectural Diagram of Spectrum Analyzer Tool

## Design Rationale

Discuss the rationale for selecting the architecture (and its underlying techniques or technologies/services) described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn’t choose them.

### DETAILED DESIGN

In this section, provide details (with respect to 3.1) at what each component does in a more systematic way. For instances, provide UML Class diagrams with all methods, attributes and relationships for all involving objects, if you are using OO approaches, provide a summary of algorithms in procedural description language (PDL) or pseudocode.

### DATA (DATABASE) DESIGN

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

### HUMAN INTERFACE DESIGN

## UI design

## UX design

### APPENDICES

*This section is optional.*

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.