



# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

**For:** GestureTunes – Hand Gesture-Based Music Controller using Spotify

**Version:** 1.0

**Developed by:** Sparsh

**Date:** 10 July 2025

---

## 1. Introduction

### 1.1 Purpose

This SRS outlines the requirements for **GestureTunes**, a Python-based application that allows users to control Spotify playback using hand gestures captured via a webcam. It is intended for academic demonstration and resume enhancement.

### 1.2 Scope

GestureTunes uses computer vision (OpenCV + MediaPipe) to detect specific hand gestures and perform music playback actions such as play, pause, next track, and previous track on the Spotify platform.

### 1.3 Definitions

- **OpenCV:** Computer vision library.
- **MediaPipe:** Real-time ML solutions for hand tracking.
- **Spotify:** Python client for Spotify Web API.
- **Gesture Recognition:** Identifying hand shapes/movements.
- **ROI:** Region of Interest in a camera frame.

### 1.4 Overview

This document describes the functional and non-functional requirements, system features, assumptions, constraints, and design considerations for GestureTunes.

---

## 2. Overall Description

### 2.1 Product Perspective

GestureTunes is a standalone desktop software system that uses an attached webcam to capture hand gestures in real-time and translates them into Spotify control commands.

### 2.2 Product Functions

- Capture live video frames from webcam

- Detect hand gestures using MediaPipe
- Map gestures to predefined Spotify actions
- Control Spotify playback using Spotify
- Display GUI overlays (FPS, gesture, track info)
- Log gesture-action pairs

### **2.3 User Classes and Characteristics**

- **End Users:** Anyone using a laptop/PC with Spotify and webcam.
- **Developers:** Data Science / CS students aiming to explore CV + APIs.

### **2.4 Operating Environment**

- OS: Windows 10/11
- Python 3.10+
- Webcam (external/internal)
- Spotify premium account (for API control)

### **2.5 Design and Implementation Constraints**

- Requires internet for Spotify API.
- Mediapipe needs good lighting for gesture tracking.
- Limited to controlling playback on active Spotify devices.

### **2.6 Assumptions and Dependencies**

- The user has an active Spotify account.
- Environment variables are stored in .env.
- Proper API credentials are generated via Spotify Developer Dashboard.

## **3. Specific Requirements**

### **3.1 Functional Requirements**

<b>Req ID</b>	<b>Description</b>
FR1	The system shall detect the OPEN_PALM gesture and trigger PLAY
FR2	The system shall detect the TWO_FINGERS gesture and trigger PAUSE
FR3	The system shall detect SWIPE_LEFT gesture and trigger NEXT TRACK
FR4	The system shall detect SWIPE_RIGHT gesture and trigger PREVIOUS TRACK
FR5	The system shall show overlay text on video frame
FR6	The system shall log all gesture-action events in <code>gesture_log.txt</code>

### **3.2 Non-Functional Requirements**

<b>NFR ID</b>	<b>Description</b>
NFR1	The system should detect gestures in real-time with minimum lag

NFR ID	Description
NFR2	Cooldown of 1.5s should be applied to swiping gestures
NFR3	The FPS must be displayed for performance reference
NFR4	The system shall be modular and support future features (like volume control)

---

## 4. External Interface Requirements

### 4.1 Hardware Interfaces

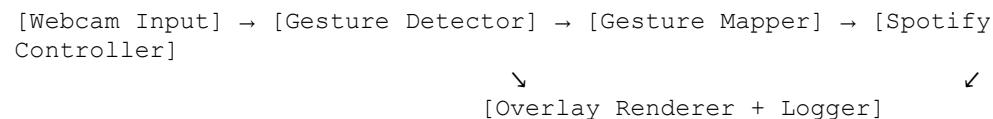
- Webcam for input

### 4.2 Software Interfaces

- OpenCV (video streaming)
- MediaPipe (hand tracking)
- Spotify (Spotify API calls)
- dotenv (credentials)

---

## 5. System Architecture



## 6. Future Enhancements

- Volume control using two-finger up/down gestures
- Album art in overlay
- Custom user-defined gesture training
- Cross-platform packaging (EXE/MacOS)