Software Requirement Specification (SRS)

Project Title: GestureTunes - Gesture-Based Spotify Controller Using Webcam
Version: 1.0
Prepared by: Sparsh
Date: July 9, 2025

---

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the development of “GestureTunes”, a gesture-based control system that uses a laptop webcam to detect hand gestures and control Spotify playback functions in real-time. This project serves both as a resume-building project and a practical demo of real-time computer vision applications.

1.2 Scope

GestureTunes will provide real-time hand gesture recognition through a webcam and map those gestures to Spotify music control actions using the Spotify Web API. The system is intended to run on Windows OS and aims to offer background functionality with a simple overlay GUI. It is designed with modularity in mind to allow for future extension into facial gesture controls and broader media player support.

1.3 Intended Audience

- Project Mentors & Reviewers
- Resume Evaluators
- Developers interested in gesture recognition and automation
- Students and educators in computer vision and HCI domains

1.4 Technologies Used

- **Programming Language:** Python
- **Libraries/Frameworks:** OpenCV, MediaPipe, Spotipy (Spotify API), Tkinter (GUI)
- **Platform:** Windows 10/11
- **Others:** Git, GitHub for version control

---

2. Overall Description

2.1 Product Perspective

GestureTunes is a standalone desktop application that interacts with Spotify using Spotify Web API. It is based on modular architecture to enable easy integration of additional functionalities like facial gesture detection and support for other media platforms.

2.2 Product Features

- Webcam-based real-time hand gesture recognition
- Gesture mapping to Spotify music controls
- Spotify Web API integration
- Simple GUI overlay displaying current gesture and action
- Modular codebase for future extensibility
- Runs in the background without user interference

2.3 User Characteristics

- Users should be familiar with using Spotify desktop application
- No advanced technical knowledge is required for usage
- Developers contributing to the project should be comfortable with Python

2.4 Assumptions and Dependencies

- Stable internet connection is required for Spotify Web API access
- Webcam must be functional and accessible by the system
- User must authorize Spotify access via developer credentials

---

3. Specific Requirements

3.1 Functional Requirements

FR1: The system shall access the webcam and stream real-time video frames.
FR2: The system shall detect predefined hand gestures using MediaPipe.
FR3: The system shall map recognized gestures to Spotify music control actions.
FR4: The system shall display the current gesture and corresponding action on a GUI overlay.
FR5: The system shall remain active in the background.
FR6: The user shall be able to terminate the application using a system tray or GUI control.

3.2 Non-Functional Requirements

NFR1: The application shall have minimal latency in gesture-to-action response (<1s).
NFR2: The system shall run efficiently on systems with 4GB+ RAM.
NFR3: The UI overlay shall not obstruct other applications significantly.
NFR4: The system shall be modular for future gesture/action additions.

3.3 System Features

Gesture	Action
Open Palm	Play / Pause
Swipe Left	Previous Track
Swipe Right	Next Track
Two Fingers Static	Stop Playback
Two Fingers Wave Upward	Increase Volume
Two Fingers Wave Downward	Decrease Volume

4. External Interface Requirements

4.1 Hardware Interfaces

- Laptop webcam for real-time video input
- Optional: Microphone for future voice/gesture hybrid interaction

4.2 Software Interfaces

- **Spotify Web API** for authentication and media control
- **MediaPipe** for hand gesture recognition
- **OpenCV** for video stream processing
- **Tkinter** for GUI overlay

4.3 Communication Interfaces

- HTTPS requests to Spotify Web API for playback control
- Local system key events (optional) for additional control

---

5. Use Case Overview

Use Case: Play/Pause Music Using Open Palm Gesture

- Actor: User
- Trigger: User shows open palm in front of webcam
- Description: System recognizes open palm and toggles Play/Pause in Spotify
- Outcome: Music playback is toggled and GUI displays “Play/Pause Activated”

---

6. System Flowchart (Text Representation)

```
[Start Application] → [Initialize Webcam] → [Detect Gesture via MediaPipe]
→
→ [Match Gesture] → [Trigger Spotify Action] → [Display GUI Feedback] →
[Loop]
```

---

7. Future Enhancements

- Facial gesture recognition (e.g., nod to play, shake to pause)
- Integration with other platforms like VLC, YouTube
- System tray control to toggle detection ON/OFF
- User-defined custom gestures
- Voice-based gesture confirmation (“Play command recognized”)

---

8. Glossary

- **MediaPipe:** A machine learning framework by Google for real-time perception pipelines.

- **Spotify Web API:** A REST API to control Spotify playback and fetch user data.
- **Gesture Recognition:** The process of identifying and interpreting body movements via computer vision.
- **Spotipy:** A Python library for Spotify Web API integration.
- **OpenCV:** Open-source computer vision library used for image/video processing.

---

9. Approval

Name	Role	Signature	Date
Sparsh	Developer	[Signed]	July 9, 2025

End of Document