Design Specification

for

MotionTracker

Version 1.1

Prepared By:

Humaid Mustajab, Samuel Tsui, Sean Rhee, Logan Kramsky, Johnson Wu, Evan Pugh

Advisor:

Prof. Jeff Salvage

Stakeholder:

Prof. Jeff Salvage

Table of Contents

1. Document History

Name	Date	Reason	Version
Humaid M, Samuel T, Sean R, Logan Kramsky, Johnson Wu, Evan Pugh	May 9, 2023	Initial Draft	1.0
Humaid M, Samuel T, Sean R, Logan Kramsky, Johnson Wu, Evan Pugh	May 9, 2023	Final Draft	1.1

2. Introduction

2.1 Purpose

This document specifies the software architecture and design specifications for MotionTracker. The design decisions outlined in this document were made in compliance with the software constraints and functionality requirements outlined in the MotionTracker Software Requirements Specification document.

2.2 Scope

This document describes the software requirements for the MotionTracker project. The initial release is tailored to be a proof of concept or a minimum viable product against the requirements. This document is intended for end-users and developers interested in the application’s functionality and future evolution.

2.4 References

This document references the requirements in the Software Requirements Specification for MotionTracker document. Additionally, information about validation tests for these requirements are found in the Acceptance Test Plan for MotionTracker.

This document uses the Senior Design Project ‘WAVED’ as a reference and guide for general formatting.

3. System Overview

3.1 Context Diagram

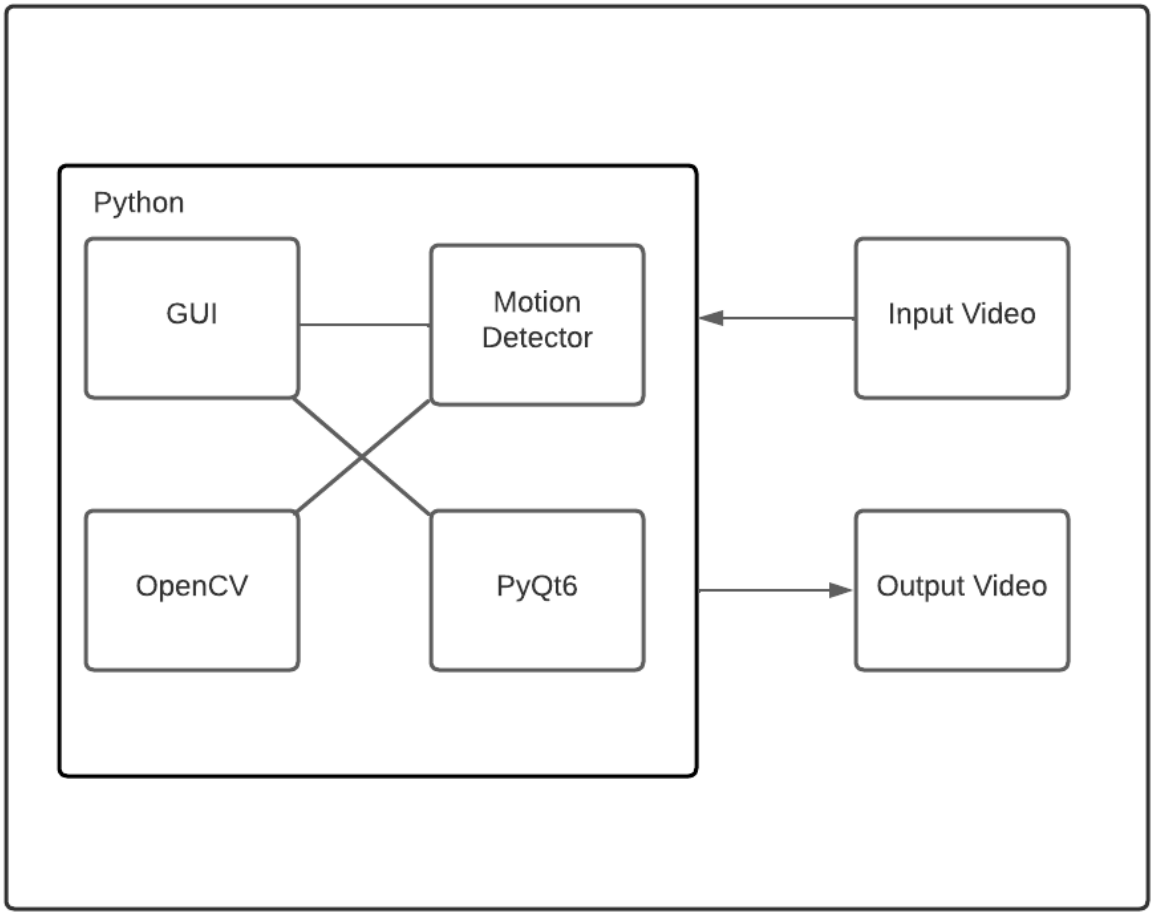

Figure 1. MotionTracker Context Diagram

Figure 1 shows the three main components of the MotionTracker application. The first component is MotionTracker's GUI. This is how users interact with the system. Through the GUI, users upload video clips and input different parameters for the output clip. The GUI is supported by the utilization of PyQt6, which is a Python library for creating GUI applications. The third component is OpenCV. OpenCV is a computer vision program which is responsible for scanning through a user's inputted video clip to detect times where motion is occurring.

3.2 Technologies Used

MotionTracker's GUI relies on PyQt6 for functionality. The application also leverages the third party computer vision library OpenCV. OpenCV contains algorithms which conduct motion detection; MotionTracker relies on these algorithms provided by OpenCV.

3.3 Deployment Diagram

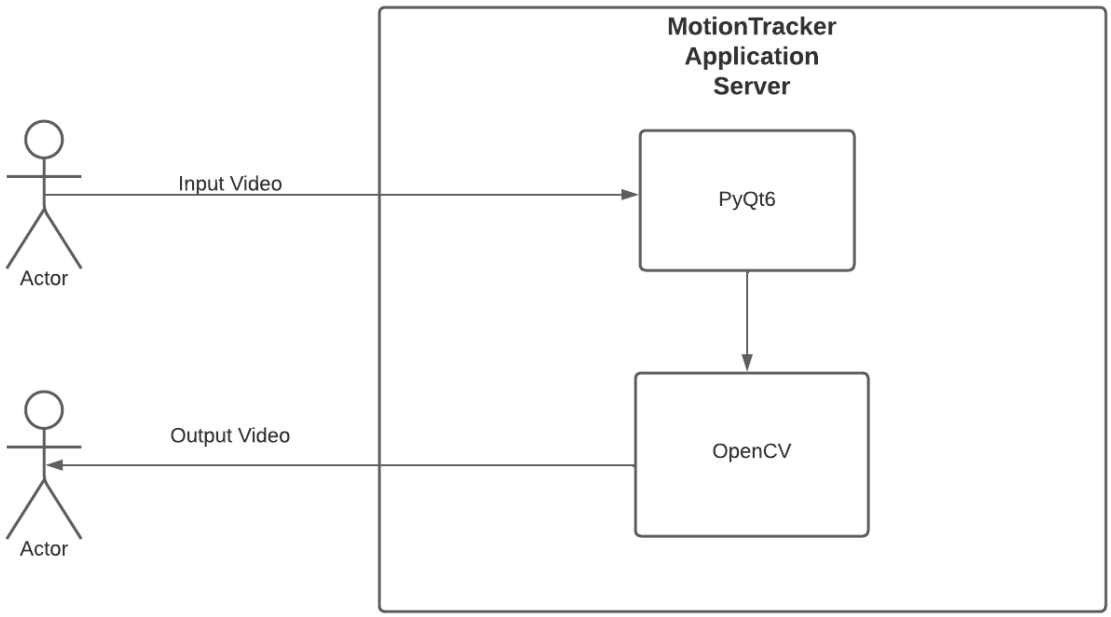

Figure 2. Deployment Diagram

3.4 Architecture Diagram

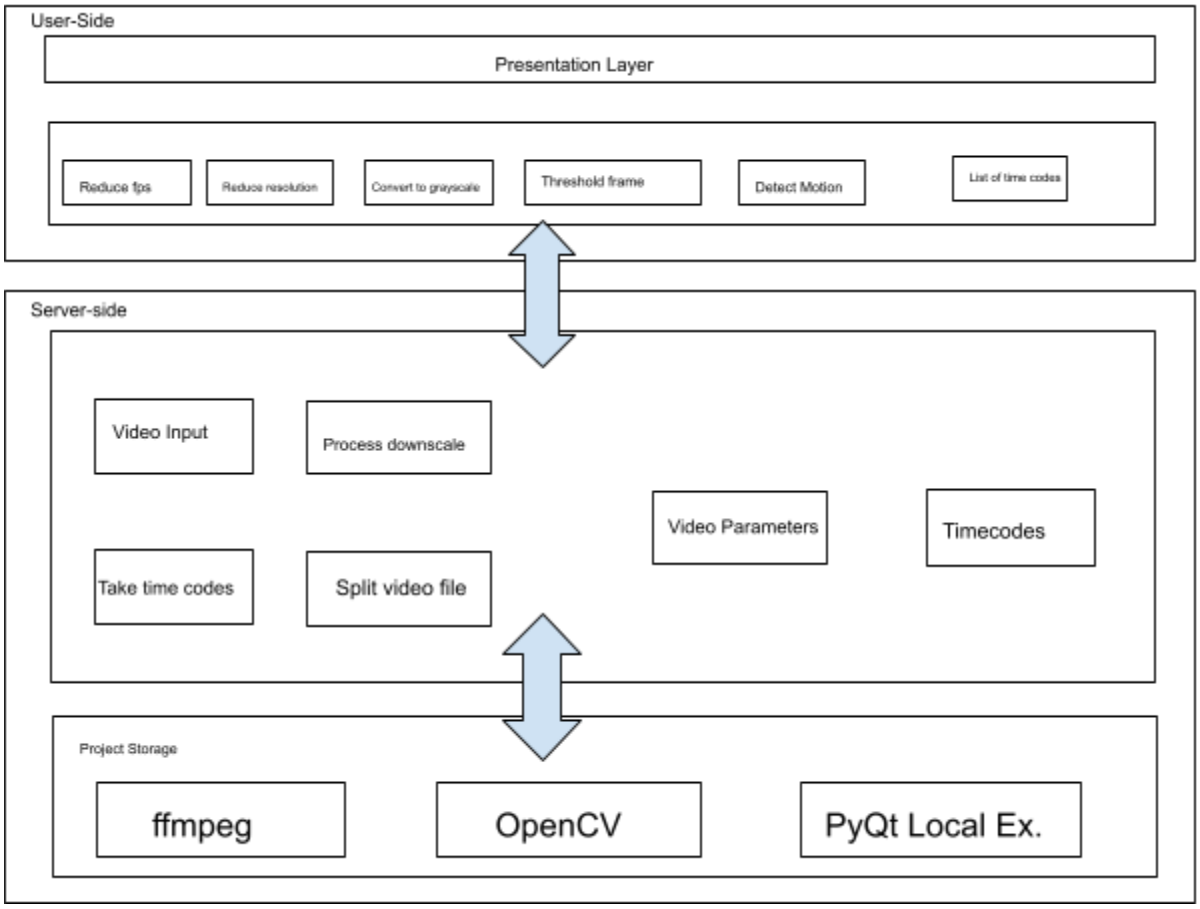

Figure 3. MotionTracker Architectural Diagram

3.5 Model-View-View Model Diagram

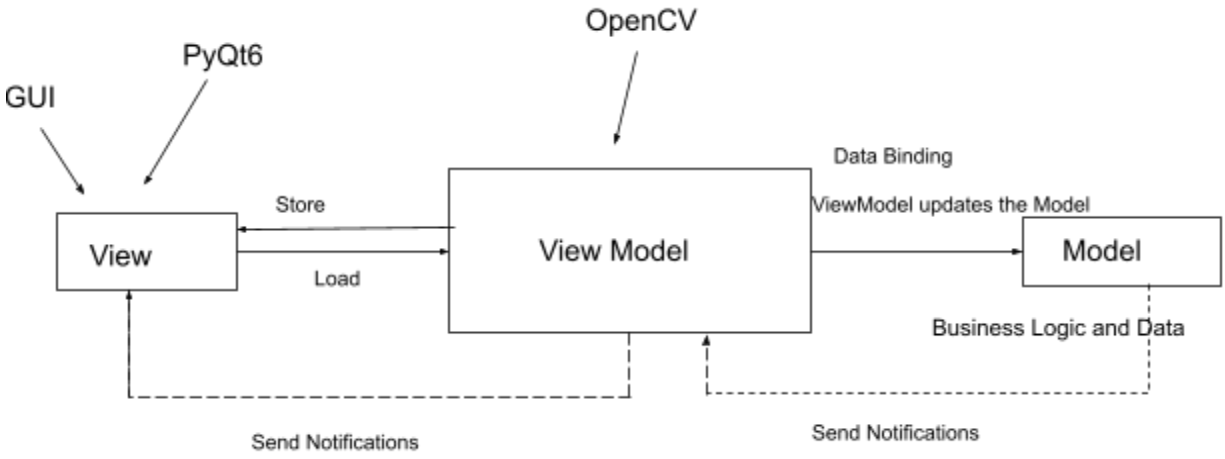

Figure 4. MVVM Diagram

3.6 Component Diagram

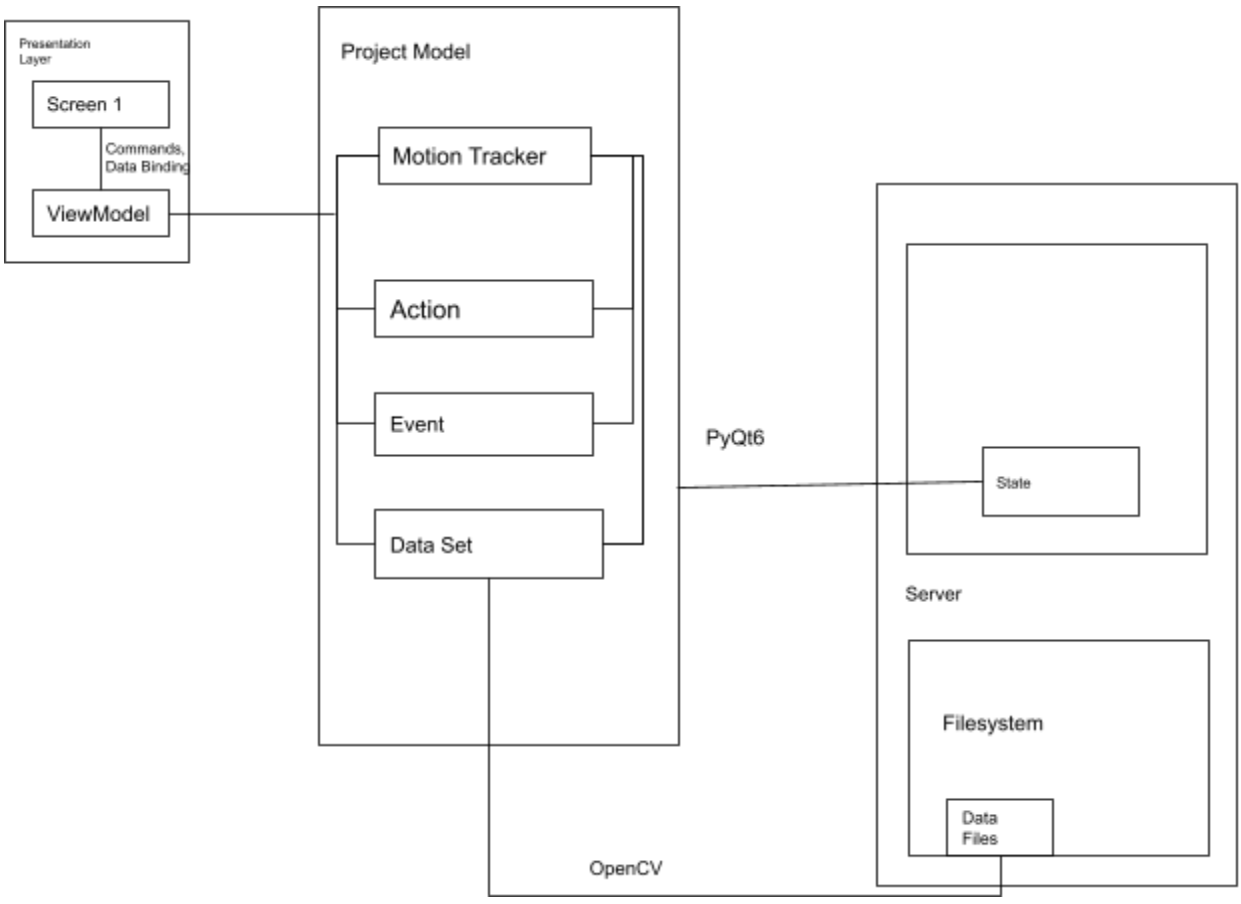

Figure 5. Component Diagram

3.7 Sequence Diagrams

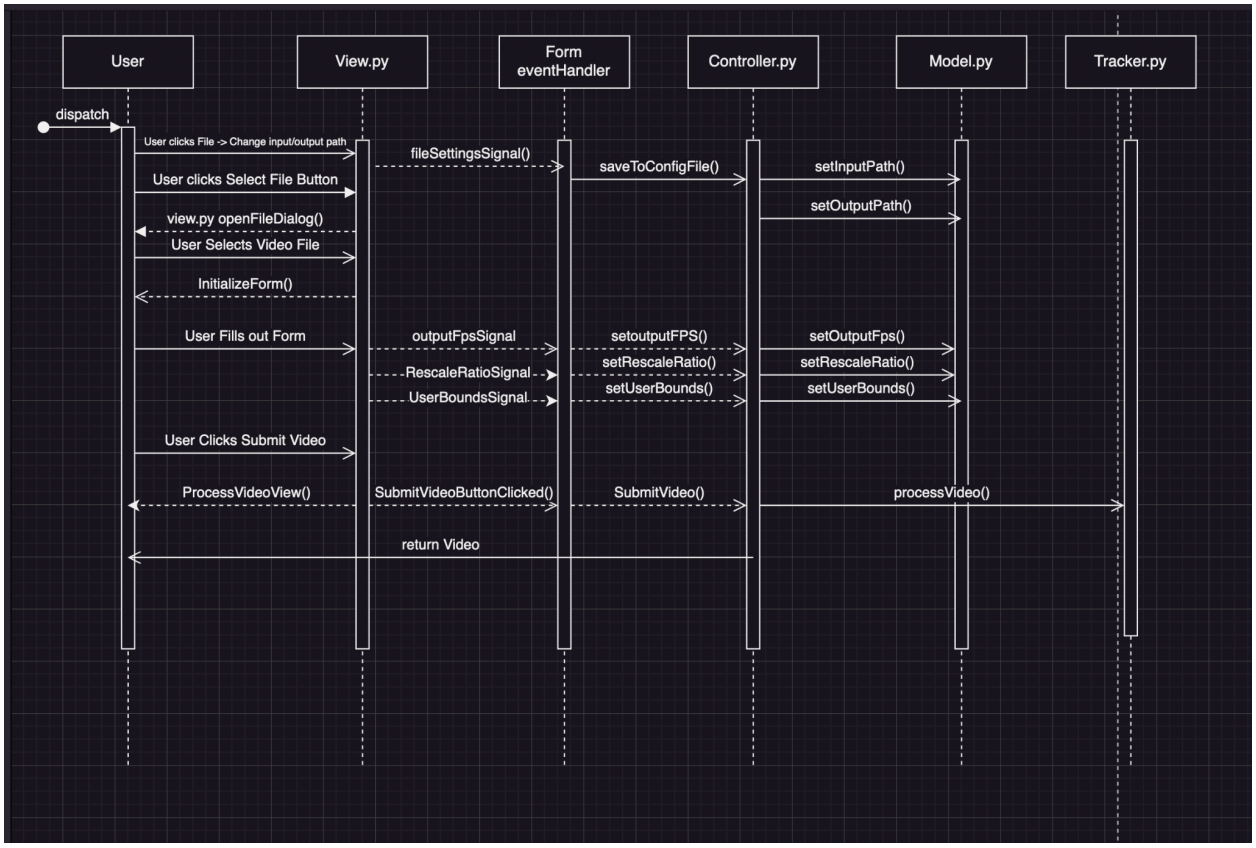

Figure 6. MotionTracker Sequence Diagram

4. Traceability Matrix

Requirement	Description	Function Name
R3.1	‘Screen 1’ Requirements	
R3.1.1	Upon launch, choose ‘Select File Button’	openFileDIalog
R3.1.2	Selects a video file	InitializeForm
R3.1.3	‘Screen 2’ Requirements	
R3.1.3	Enter input parameters for the range of video height/width	setUserBounds
R3.1.3.1	Values are within range of original clip	setUserBounds
R3.1.3.2	Prompted again if value is not accepted by criteria specified	setUserBounds
R3.1.3.3	See range affecting video frame	setUserBounds
R3.1.4	Input Integer for video frame rate	setOutoutFPS
R3.1.4.1	Default value is native frame rate	setOutoutFPS
R3.1.4.2	Decides to change default value, integer value>0	setOutoutFPS
R3.1.5	Input for Rescaling	setRescaleRatio

	Ratio	
R3.1.7	Must click ‘Submit.’	submitVideo
R3.1.8	Pick Direction of Motion to filter using UI	submitVideo
R3.1.9	Progress Bar Displayed	processVideo
R3.1.10	Notification on process completion	processVideo
R3.1.11	MotionTracker Exits	processVideo
R3.1.12	MotionTracker opens a File Explorer at output directory	processVideo