

Software Requirements Specification (SRS) for Rodent Video Tracking Software

Version 1.0

Prepared By: Yifei Chen, Bazil Akram, Andrew Abdelshahed

Prepared For: Dr. Yang Yang, Dr. Sayed Mohsin Reza, Dr. Truong Xuan Tran, Dr. Jim
Robertson

Date: 12/14/2024

1. Introduction.....	4
1.1 Purpose.....	4
1.2 Scope.....	4
1.3 Definitions, Acronyms, and Abbreviations.....	4
1.4 Overview.....	4
2. Overall Description.....	5
2.1 Product Perspective.....	5
2.2 Product Features.....	5
2.2.1 Real-Time Tracking.....	5
2.2.2 Multi-Rodent Support.....	5
2.2.3 Data Analysis.....	5
2.2.4 Behavior Identification.....	5
2.2.5 Data Export.....	5
2.2.6 Visualization.....	6
2.2.7 User Management.....	6
2.3 User Classes and Characteristics.....	6
• Admin.....	6
• Researcher (Primary User).....	6
• Lab Technicians.....	6
2.4 Operating Environment.....	7
2.4.1 Hardware Requirements:.....	7
2.4.2 Software Requirements:.....	7
2.4.3 Network Requirements:.....	7
2.5 Assumptions and Dependencies.....	7
3. Functional Requirements.....	7
3.1 Video Input.....	7
3.1.1 Supported Video Formats.....	7
3.1.2 Real-Time Video Stream Integration.....	7
3.1.3 Video Pre-Processing.....	8
3.2 Rodent Detection and Tracking.....	8
3.2.1 Single and Multiple Rodent Tracking.....	8
3.2.2 Tracking Algorithms.....	8
3.2.3 Pose Estimation.....	8
3.2.4 Trajectory Calculation.....	8
3.3 Statistical Summary.....	8
3.4 Data Output.....	8
3.5 User Interface.....	8
3.5.1 Graphic Interface.....	8
3.5.2 Visualization Overlays.....	9

4. Non-Functional Requirements.....	9
4.1 Performance Requirements.....	9
4.2 Reliability Requirements.....	9
4.3 Security Requirements.....	9
4.4 Usability Requirements.....	9
4.5 Scalability Requirements.....	9
4.6 Portability Requirements.....	9
4.7 Maintainability Requirements.....	10
5. System Models.....	10
5.1 User Interface Mockup.....	10
5.1.1 Video Overlay.....	10
5.1.2 Login Screen.....	11
5.1.3 Main Menu.....	11
5.1.4 New Process Window.....	12
5.1.5 Previous Videos Window (w/ search filter).....	13
5.1.6 Livestream Overlay.....	14
5.2 Database Schema.....	14
6. External Interface Requirements.....	14
6.1 Hardware Interfaces.....	14
6.1.1 Cameras and Video Capture Devices.....	14
6.1.2 GPU Acceleration.....	15
6.1.3 Storage Device.....	15
6.2 Software Interfaces.....	15
6.2.1 Third-Party Libraries and Frameworks.....	15
6.2.2 Operation System.....	15

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to define the requirements and constraints for the rodent video tracking software. This document establishes a comprehensive foundation for the system's design, development, testing, and validation. It communicates to all stakeholders—including researchers, lab technicians, and administrative personnel—the intended functionality, performance criteria, user interactions, and external interfaces of the software.

1.2 Scope

The rodent video tracking software is intended to automate the detection, tracking, and analysis of rodent behavior in both live and recorded video experiments.

- Integration with video inputs (local files, live streams).
- Pose estimation
- Data export in common formats
- Visualization tools
- Real-time and offline tracking of single or multiple rodents.

1.3 Definitions, Acronyms, and Abbreviations

- FPS (Frames Per Second): A measure of how many frames are processed or displayed per second.
- Pose Estimation: A method of identifying the position and orientation of an animal's body parts in the video frames.
- OpenCV: An open-source computer vision library used for image processing and object detection.

1.4 Overview

This SRS begins by introducing the software's purpose, scope, key definitions, and references (Section 1). It then provides a high-level description of the software's overall functionality, user classes, operating environment, and assumptions (Section 2). Next, it details the functional requirements (Section 3) and non-functional requirements (Section 4) that the software must meet. System models (Section 5) and external interface requirements (Section 6) follow, providing diagrams and specifications for hardware, software, and communication interfaces.

2. Overall Description

2.1 Product Perspective

The rodent video tracking software is designed to integrate seamlessly with laboratory environments, enhancing the study of rodent behavior by providing an automated, accurate, and efficient tracking solution. It operates as a standalone web application but can integrate with existing video capture systems and lab management tools. The software employs advanced computer vision techniques to track rodents' movements and analyze their behavior. This tool is intended to replace manual tracking methods, increasing the reliability and reproducibility of behavioral experiments.

2.2 Product Features

2.2.1 Real-Time Tracking

- Detect and track rodent movements in live or pre-recorded videos.
- Each video environment will typically have one rat; however, some videos may contain up to four rats in separate mazes.

2.2.2 Multi-Rodent Support

- Track multiple rodents when they are in separate environments (e.g., four mazes in one video).
- Ensure non-interaction between rodents in separate environments during tracking.

2.2.3 Data Analysis

- Provide metrics such as time reached and spent in landmark zones, behavioral event counts and durations for actions like "stay still" and "nose poking."
- Support comparison of manual scoring (e.g., Limelight's "Scored Behaviors") with system-generated results for validation.

2.2.4 Behavior Identification

- Automatically identify and log specific behaviors such as staying still or nose poking (head movement to the left or right when the body is stationary).
- Expand the behavior list dynamically based on research needs.

2.2.5 Data Export

- Export analyzed data into various formats, such as CSV or JSON, for further study.

2.2.6 Visualization

- Display rodent trajectories and/or heatmaps for visual analysis.

2.2.7 User Management

- Support for multiple users with role-based permissions.

2.3 User Classes and Characteristics

- **Admin**
 - Responsibilities
 - Oversee and manage user accounts.
 - Configure system settings (e.g., video input settings, tracking parameters).
 - Access all data, reports, and configurations.
 - Manage software updates and maintenance.
 - Permissions
 - Full access to all features.
 - Manage other user roles and permissions.
- **Researcher (Primary User)**
 - Responsibilities:
 - Load and process video data (import video files or set up live streams).
 - Define experimental parameters (e.g., tracking zone, rodent behavior metrics).
 - Review and analyze tracking results.
 - Visualize movement paths, heatmaps, and behavioral data.
 - Permissions:
 - Access video processing tools.
 - Generate and export tracking results.
 - Customize tracking configurations (if needed).
- **Lab Technicians**
 - Responsibilities
 - Perform detailed analysis on the tracking data (e.g., calculating speed, time in zones, path length).
 - Generate statistical reports and visualizations.
 - Provide insights and recommendations based on data.
 - Permissions
 - Access to generated tracking data.
 - View and manipulate reports.
 - Export data for further analysis.

2.4 Operating Environment

2.4.1 Hardware Requirements:

- High-resolution video cameras for recording.
- Computers with GPUs for real-time processing (e.g., NVIDIA GTX 1660 or higher).
- Minimum RAM: 16 GB.

2.4.2 Software Requirements:

- Operating Systems: Windows 10 or higher, or Linux (Ubuntu).
- Required Libraries: OpenCV (may or may not change)
- Database: Oracle
- Language: Python
- Framework: Django

2.4.3 Network Requirements:

- Local or cloud storage is used to save video and analysis files.

2.5 Assumptions and Dependencies

- The software assumes that the videos provided are of sufficient quality (e.g., well-lit, stable camera placement) to ensure reliable tracking.
 - The system depends on compatible hardware, including cameras and GPUs, to deliver optimal performance.
-

3. Functional Requirements

3.1 Video Input

3.1.1 Supported Video Formats

The video shall support common video formats (e.g., MP4, AVI, MOV).

3.1.2 Real-Time Video Stream Integration

The software shall be able to process a continuous video stream from a live camera feed in real-time.

3.1.3 Video Pre-Processing

The software shall provide optional video pre-processing capabilities such as background subtraction.

3.2 Rodent Detection and Tracking

3.2.1 Single and Multiple Rodent Tracking

The system shall detect and track the positions of one or more rodents.

3.2.2 Tracking Algorithms

The system shall implement a configurable tracking algorithm using OpenCV or another model for improved tracking accuracy.

3.2.3 Pose Estimation

The system supports pose estimation (key point detection) when integrated with OpenCV or similar libraries.

3.2.4 Trajectory Calculation

The system shall record and store the rodents' centroid positions frame-by-frame, and calculate time spent by each rodent in landmark zones (e.g., corners, center) of the environment, enabling trajectory calculation over time.

3.3 Statistical Summary

The system shall generate summary statistics (such as total time, entry zone, etc) on demand.

3.4 Data Output

The system shall store processed results in a local folder and maintain an Oracle database for saving the associated directory of the results of each video.

3.5 User Interface

3.5.1 Graphic Interface

The software shall provide a graphical user interface (GUI) that allows users to load videos or connect to live streams, start/stop tracking, configure tracking and analysis parameters, visualize trajectories and metrics.

3.5.2 Visualization Overlays

The GUI shall overlay rodent positions, trajectories, and optional pose markers onto the video frames for immediate visual feedback.

4. Non-Functional Requirements

4.1 Performance Requirements

The system shall process video streams at a minimum of 30 FPS for single-rodent tracking in real-time.

4.2 Reliability Requirements

The system shall gracefully recover from transient errors, such as a momentary loss of camera input by attempting to reconnect or skip frames while ensuring that its database and output files are regularly backed up or exportable to prevent data loss.

4.3 Security Requirements

The system shall implement password-protected user accounts and encrypt sensitive information (e.g., login credentials) both at rest (in the database) and in transit.

4.4 Usability Requirements

The GUI shall be intuitive enough that a new user can load a video, run a basic tracking analysis, and export results after a short training session by providing clear documentation, online help resources, and tooltips within the software interface.

4.5 Scalability Requirements

The architecture shall support integration with more advanced tracking or behavior analysis modules (e.g., plug-in frameworks), and shall be able to handle larger datasets (longer videos, higher frame rates) by leveraging GPU acceleration or distributed processing options in future upgrades.

4.6 Portability Requirements

The software shall be compatible with Windows 10 and above, and Ubuntu Linux distributions commonly used in research labs.

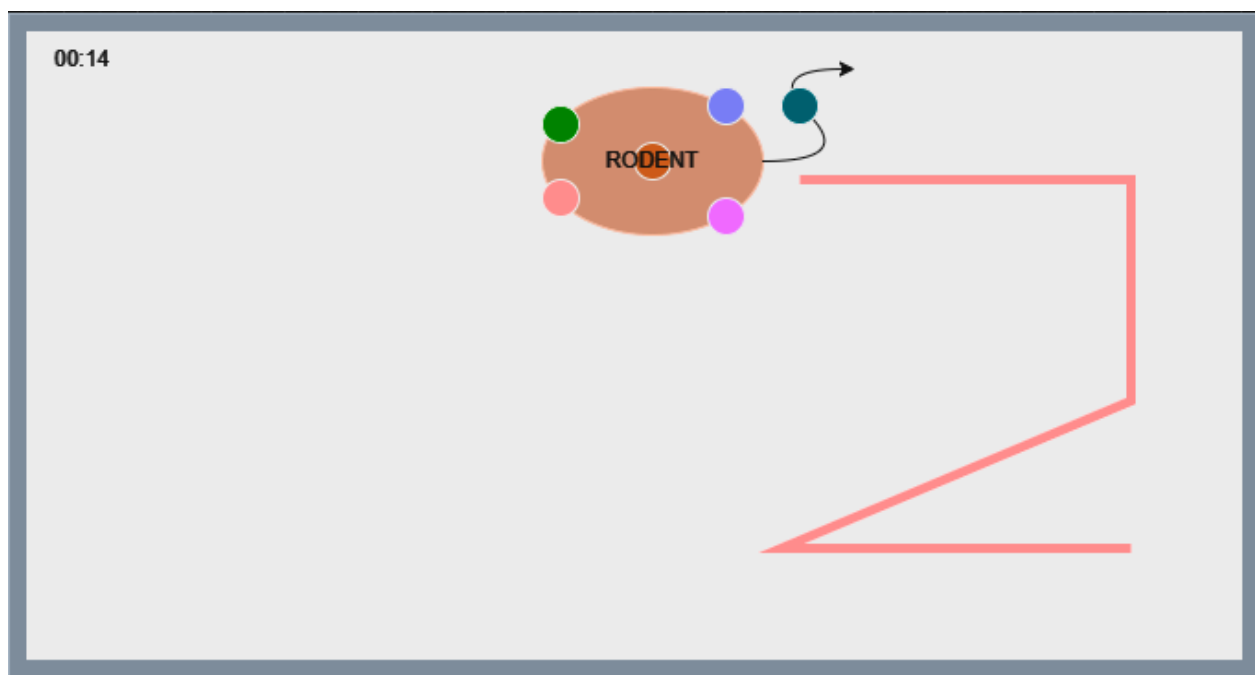
4.7 Maintainability Requirements

The codebase shall be documented following standard conventions and best practices, making it easier to update tracking algorithms.

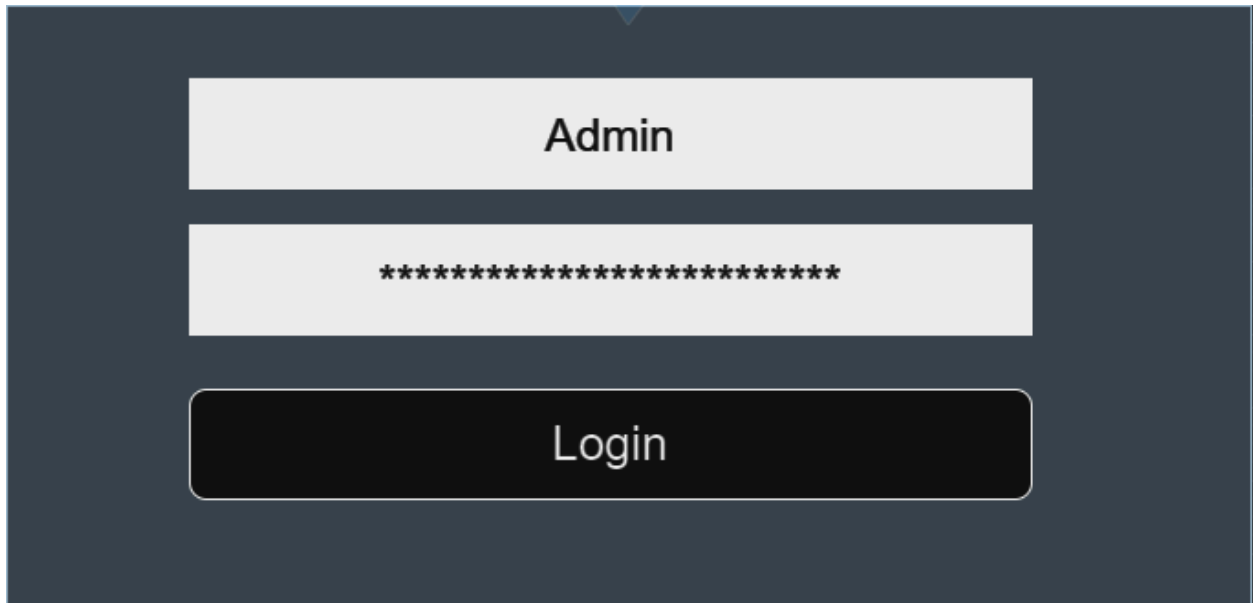
5. System Models

5.1 User Interface Mockup

5.1.1 Video Overlay



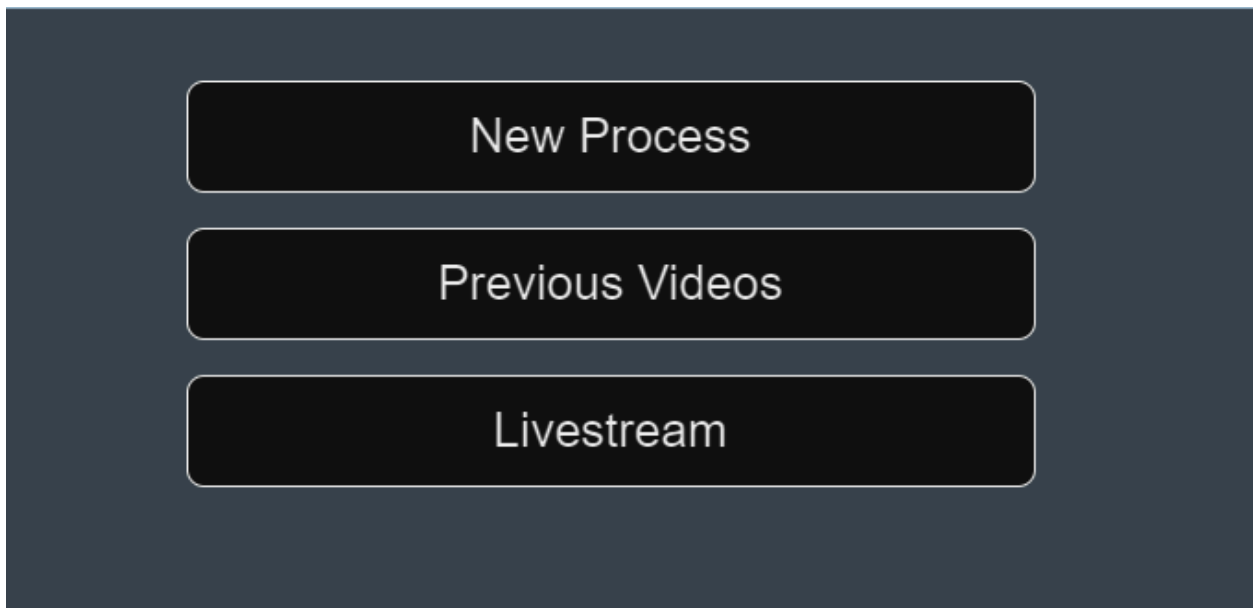
5.1.2 Login Screen

The login screen features a dark blue background. At the top center, there is a light gray rectangular box containing the text "Admin". Below this, there is another light gray rectangular box containing a series of asterisks "*****", representing a password field. At the bottom center, there is a dark blue rounded rectangular button with the text "Login" in white.

Admin

Login

5.1.3 Main Menu

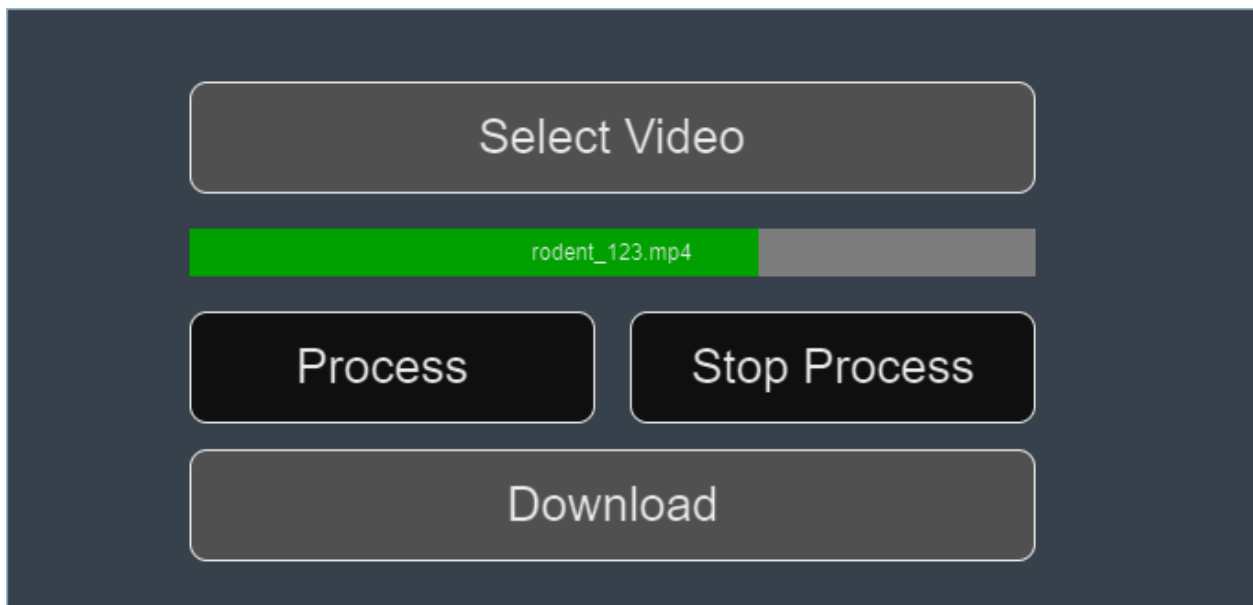
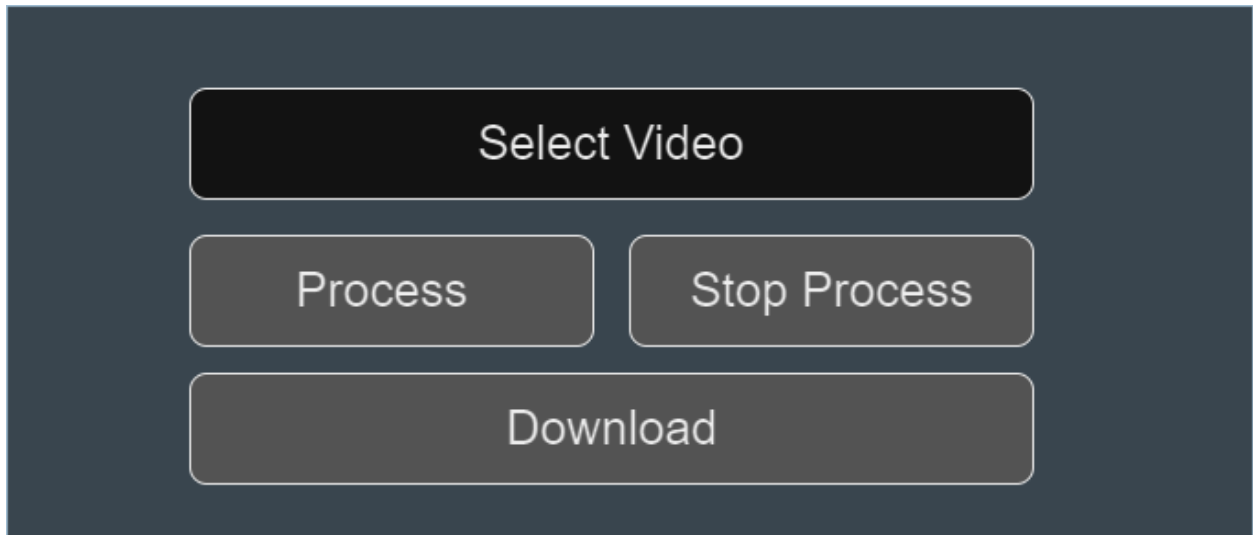
The main menu is displayed on a dark blue background. It consists of three vertically stacked, dark blue rounded rectangular buttons with white text. The top button is labeled "New Process", the middle button is labeled "Previous Videos", and the bottom button is labeled "Livestream".

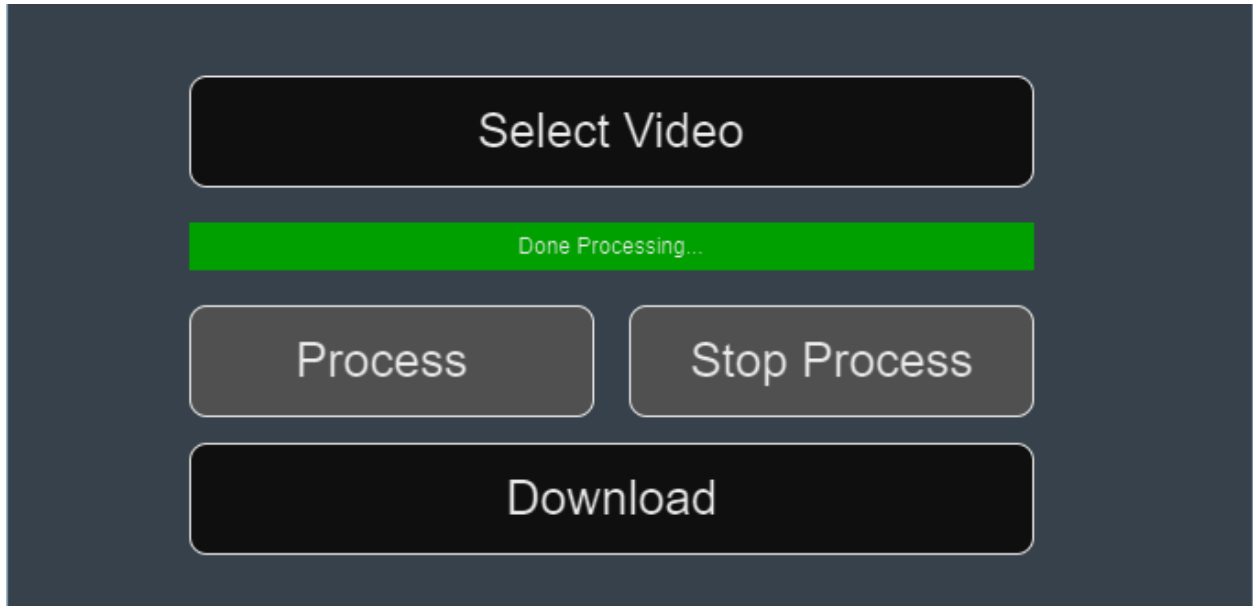
New Process

Previous Videos

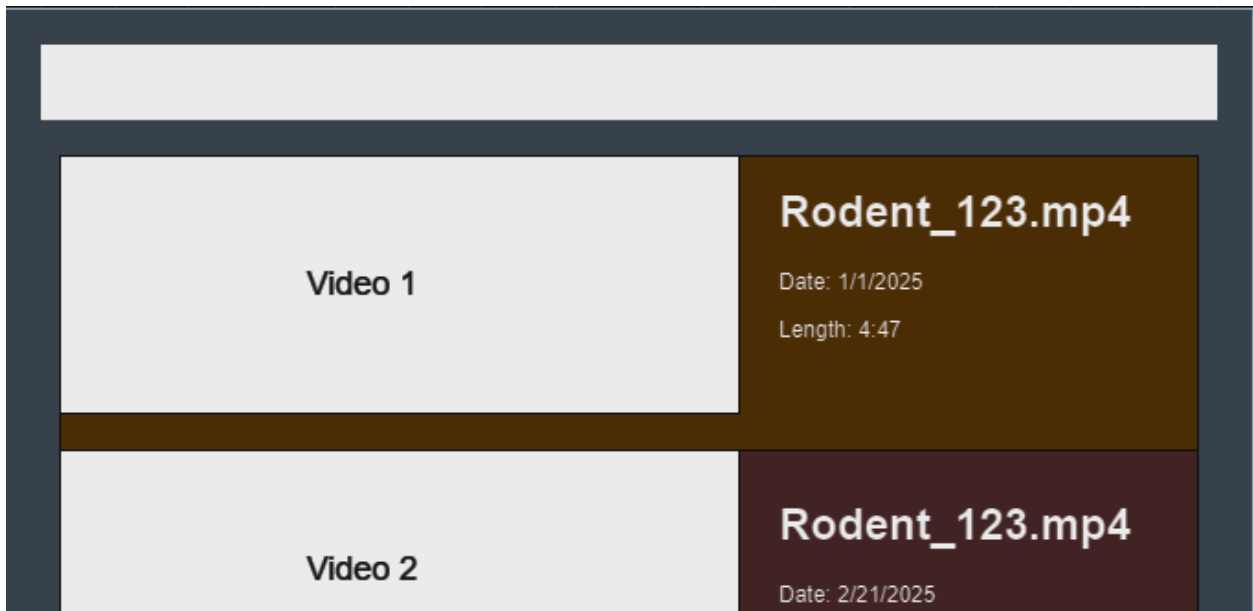
Livestream

5.1.4 New Process Window

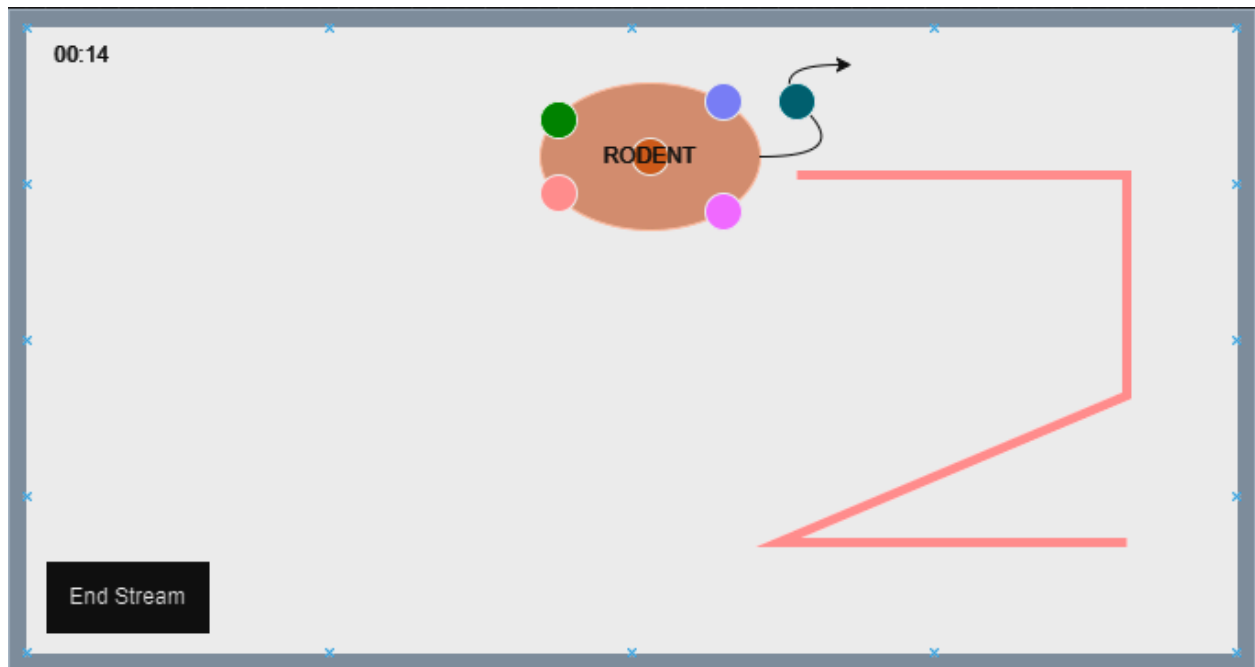




5.1.5 Previous Videos Window (w/ search filter)



5.1.6 Livestream Overlay



5.2 Database Schema

Processed Videos		
varchar	Name	PK
varchar	Directory	
datetime	Timestamp	

6. External Interface Requirements

6.1 Hardware Interfaces

6.1.1 Cameras and Video Capture Devices

The software shall support integration with common video capture devices, including USB-connected cameras and IP-based cameras (e.g., RTSP streams).

6.1.2 GPU Acceleration

The system relies on GPU acceleration to achieve the performance required for real-time tracking and, optionally, pose estimation.

6.1.3 Storage Device

The system stores video analysis results and database files locally or on network-attached storage.

6.2 Software Interfaces

6.2.1 Third-Party Libraries and Frameworks

The tracking and analysis functionalities depend on computer vision and machine learning libraries (e.g., OpenCV, DeepLabCut) for detection, tracking, and pose estimation.

6.2.2 Operation System

The software is designed to run on Windows 10 (or later) and Ubuntu Linux distributions commonly used in research labs.

Signature: _____ Date: _____