MICHIGAN STATE
UNIVERSITY

Project Plan
ADAS Tagging Tool

The Capstone Experience

Team Bosch

Adam Gongol
Alex Norris
Sam Richardson
Hannah Striebel

Department of Computer Science and Engineering
Michigan State University

Fall 2020

From Students…
…to Professionals

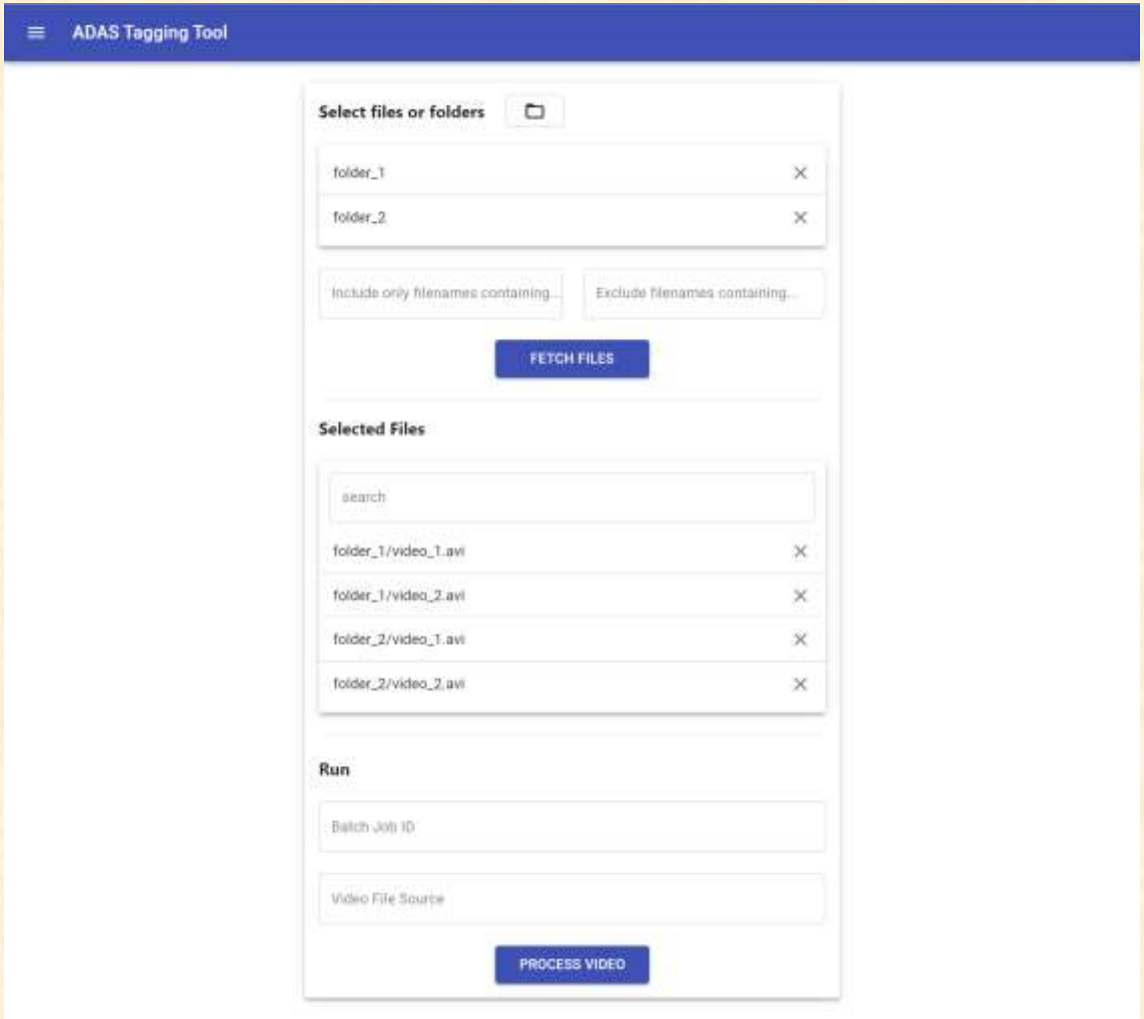
Functional Specifications

- The Problem:
  - Employees manually searching for video among thousands of hours of footage for videos containing specific features/environments.
- The Solution:
  - Utilize computer vision to automatically tag & categorize footage based on features/environments
  - Footage categorized once, then tags saved for later searching.

Design Specifications

- Process video files to extract tags
- Manually Review questionable tags
- Search for videos containing specific tags
- Create new tags
- Utilize Google’s Material Design

Screen Mockup: Process Video



Select files or folders

- folder_1 ✕
- folder_2 ✕

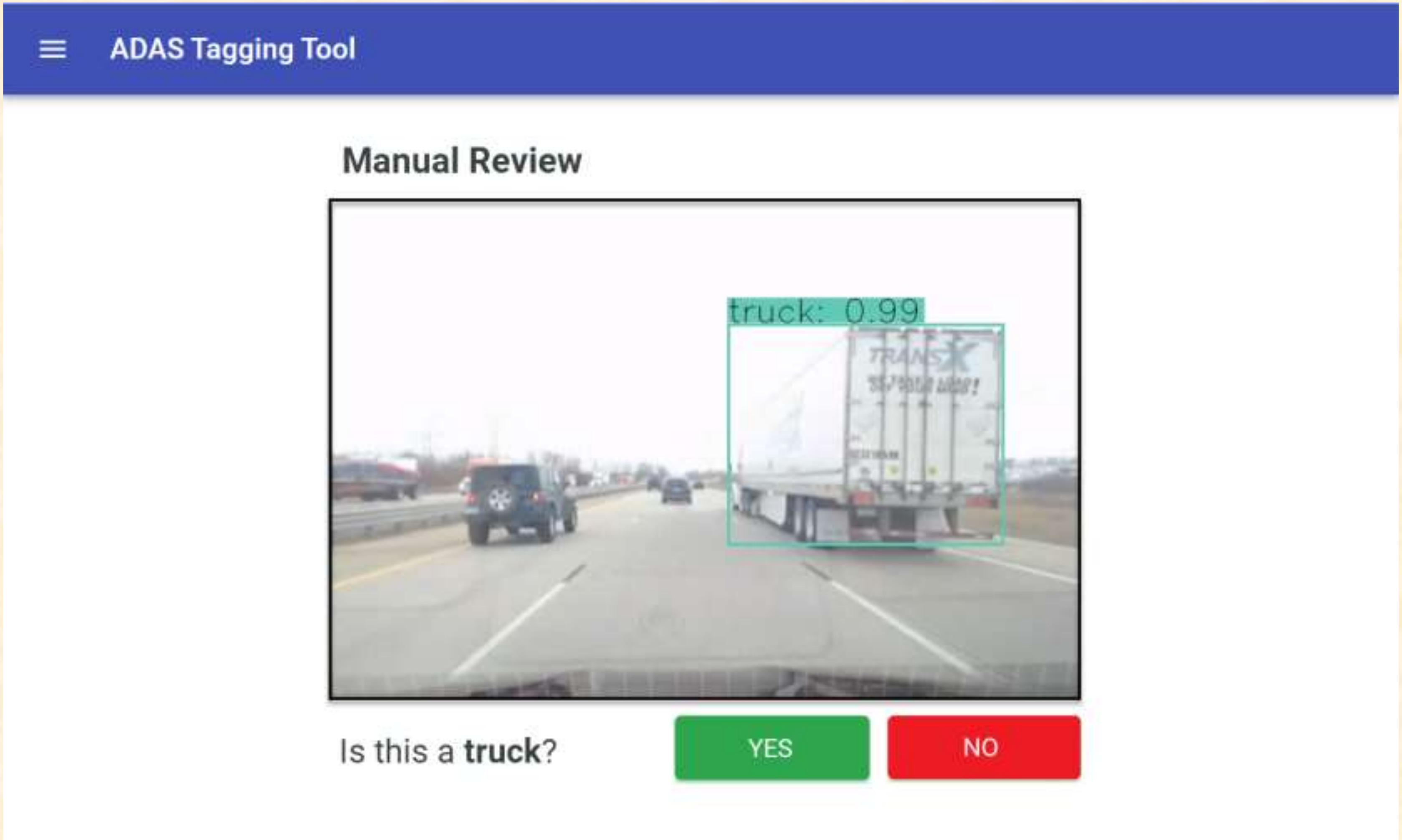
Include only filenames containing...
Exclude filenames containing...
FETCH FILES

Selected Files

- folder_1/video_1.avi ✕
- folder_1/video_2.avi ✕
- folder_2/video_1.avi ✕
- folder_2/video_2.avi ✕

Run

Screen Mockup: Manual Review

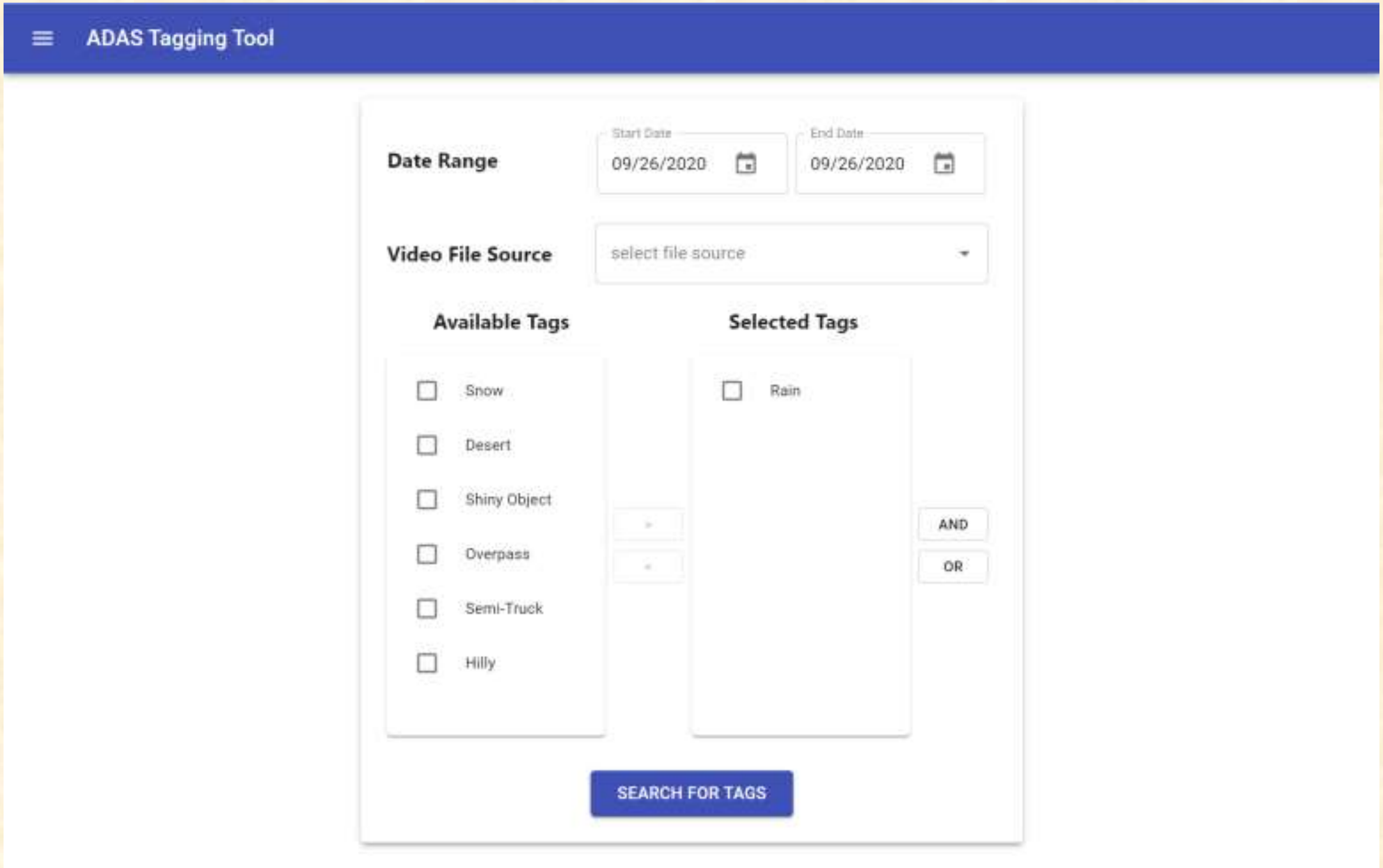


ADAS Tagging Tool

Manual Review

Is this a **truck**?

Screen Mockup: Search For Tags



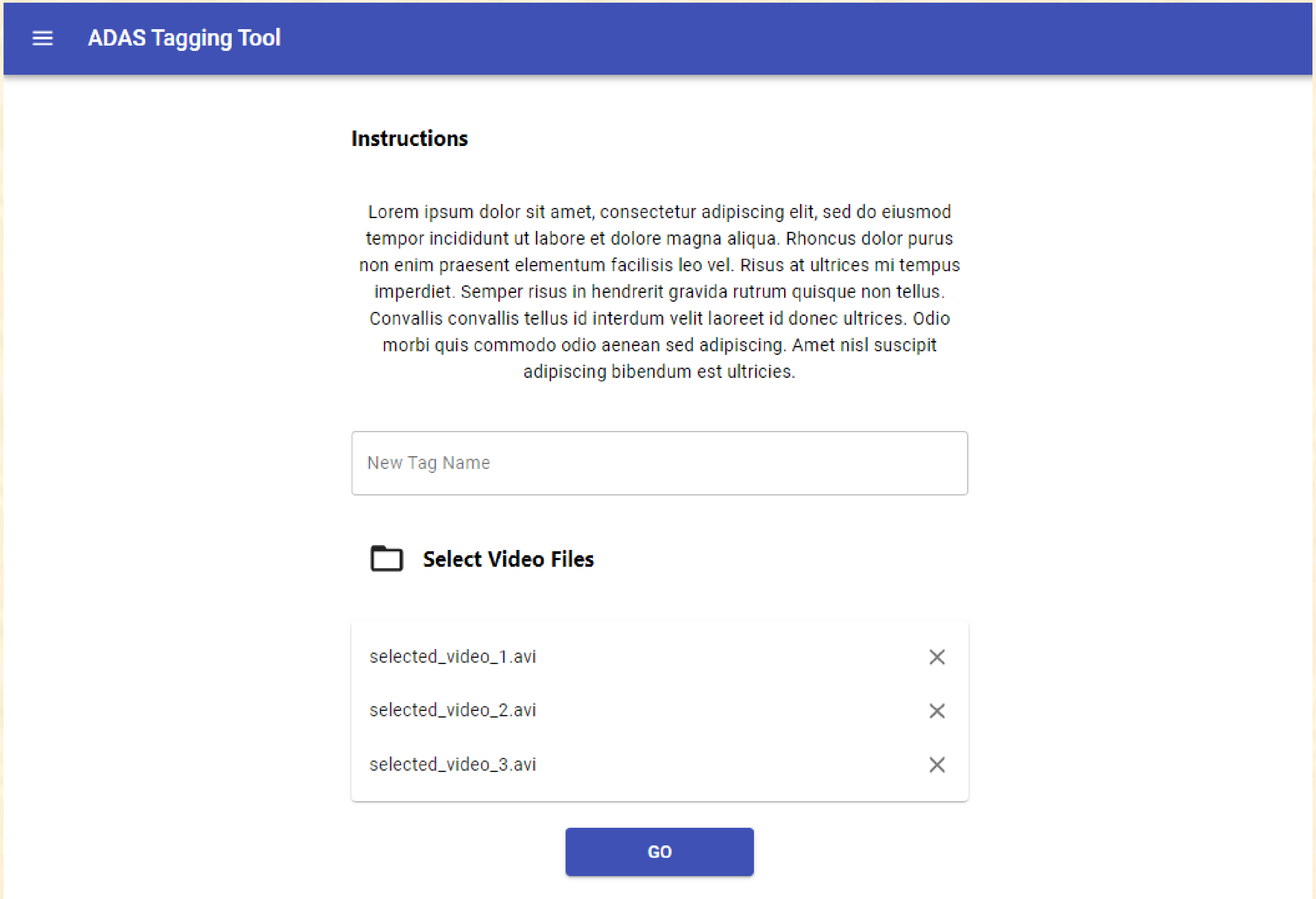
Date Range

Start Date: 09/26/2020

End Date: 09/26/2020

Video File Source: select file source

Screen Mockup: Create New Tag



Instructions

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Rhoncus dolor purus non enim praesent elementum facilisis leo vel. Risus at ultrices mi tempus imperdiet. Semper risus in hendrerit gravida rutrum quisque non tellus. Convallis convallis tellus id interdum velit laoreet id donec ultrices. Odio morbi quis commodo odio aenean sed adipiscing. Amet nisl suscipit adipiscing bibendum est ultricies.

New Tag Name

selected_video_1.avi ✕

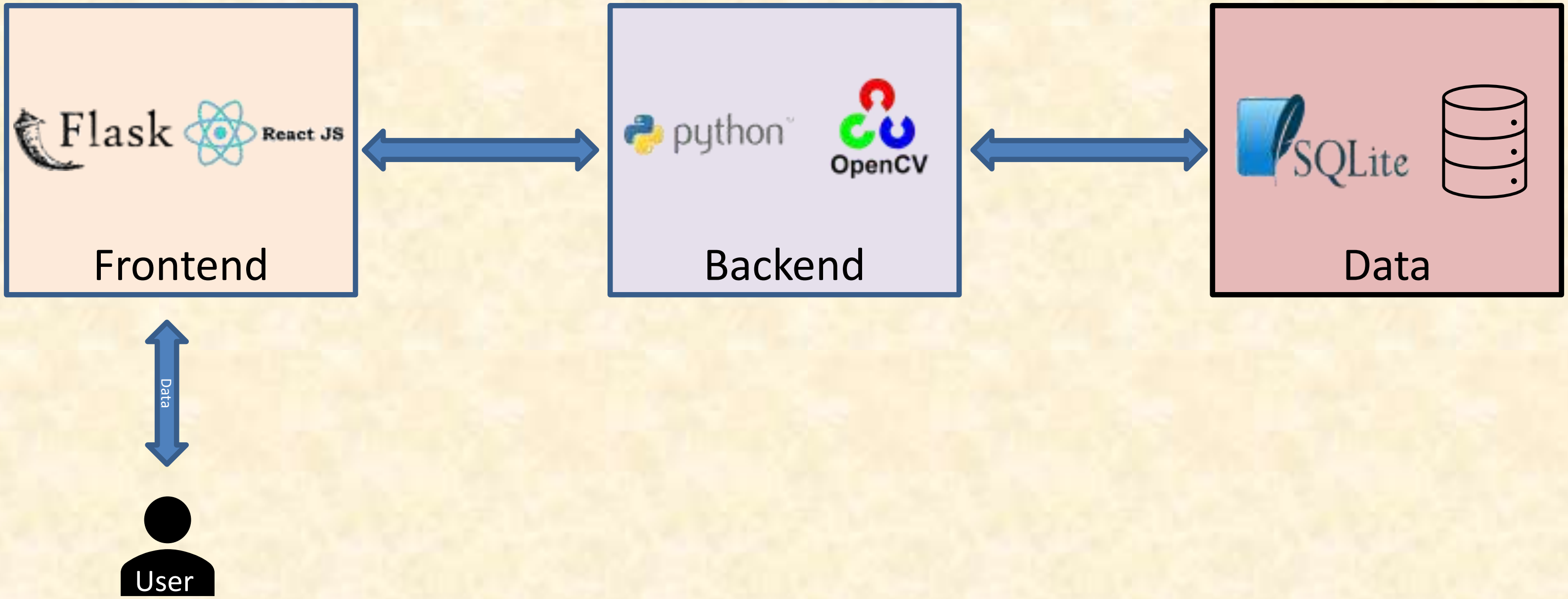
selected_video_2.avi ✕

selected_video_3.avi ✕

Technical Specifications

- Python
  - Handles backend and video processing
- OpenCV w/ COCO dataset
  - Computer vision library to analyze video
- YOLOv3
  - Algorithm for object detection
- React.js
  - Frontend JavaScript framework w/ HTML/CSS
- Flask
  - Link between front and backend
- SQLite
  - DBMS

System Architecture



System Components

- Hardware Platforms
  - Local PC
- Software Platforms / Technologies
  - Visual Studio Code IDE
  - PyCharm IDE
  - SQLite Studio

Risks

- Meeting Client’s accuracy requirements
  - Minimize the number of results to be manually reviewed
  - Implement easy manual tagging system
- Identifying objects not in COCO dataset
  - Clients want to identify bridges/tunnels
  - Training on simpler objects (hotdog) before more difficult objects
- No experience with web app frameworks
  - Unsure if we could connect the ML libraries to a web app
  - Built web app with React/Flask to show you can use python for backend

Questions?

? ? ? ?

? ?

? ? ?