COS 301 SOFTWARE ENGINEERING

SYSTEM PROPOSAL

Lucian Sargeant (lucian@labs.epiuse.com)
Jodan Alberts (jodan@labs.epiuse.com)

1. Introduction

Home security is an increasingly important concern in our country. For this project you will be creating a security system that will provide alerts based on intelligent pattern recognition.

2. Background & Motivation

This project will involve Computer Vision, Facial Recognition and Classification so if you're interested in using Artificial Intelligence (AI) to solve real world problems, this could be the project for you.

3. System (Functional) Requirements

3.1. System Core Requirements

You will be required to develop a Web App that provides the following functionality:

- Web dashboard for the security system
- Real-time monitoring of camera feed
- Facial detection and classification (strangers vs. known people)
- Unusual behaviour detection (eg. someone walking outside at 3 AM)
- Notification system for above detections

3.2. System Optional Requirements

- Monitor multiple camera feeds
- Unusual sound detection
- Number plate detection for known and unknown cars

4. System Constraints

The code should be open source, the system should support common video and audio codecs such as .mp4 and .mp3 and be able to run on a Linux server (Ubuntu/OpenSUSE). The app should be usable on the latest Chrome and Firefox browsers and be responsive/mobile friendly.

Lucian Sargeant (lucian@labs.epiuse.com)
Jodan Alberts (jodan@labs.epiuse.com)

5. System Design

You will need to create a Web App.

The App will require a database in order to keep track of event history and other user related data.

The App will also need to interface with the provided surveillance camera(s) and process the video feed(s) for visual and sound based detection.

6. Technology Specifications

The following are recommended technologies but if you wish to use alternatives these can be discussed:

- Python with a machine learning framework like Tensorflow or PyTorch, for training of machine learning models
- Java/Kotlin backend to provide REST based API for frontend, run machine learning models and handle event driven actions
- RabbitMQ for events
- Postgres DB
- Vue or Angular frontend