Self Hosted Configurable/Modular Home Surveillance System

The Watchmen - Brenden Guillen, Jeremy Bell, Jonathan Thornton

1. Introduction

Opening Paragraph

- *What is the project?*
- *What was the motivation for the project?*
- *What is novel about it?*
- *Anything else to orient the reader?*

This project is to create a self hosted home surveillance system for family/small business users. There are many different types of this software on the market (both paid and unpaid), we would like to create a highly configurable and versatile application that uses security cameras and other optional security measures (ie, motion detection, flashing lights, sirens, noise detection). This software may include support for a small form factor (such as a Raspberry Pi) to full server support to handle larger workloads. The software may also include using other brand name cameras (such as Ring or Wyze cameras) as well as cheap web cameras in the same system. The system should also support different ways to record video and other sensor data, and different ways to notify the user to events. The various configurations will make the project appealing to many types of users.

Challenges

- *Briefly describe the main challenges for the team.*
- *How did you address them?*
- *Was the technology known or new to the team?*

Some of the challenges so far have been creating a proper GUI that meshes with the rest of the system, creating and linking the AI model properly, creating a proper video module, and dealing with many hardware/software issues in implementing the modules.

How we addressed these problems was mainly with more research even after the initial research on the individual sections. We also moved testing to all in one pc's that each of us now have (that have the same specs and OS installed), so the team was able to test on the same hardware and software, when before we ran into issues with the virtual machines we were using in detecting/using cameras.

The backend of accessing and using a video camera is new to us along with building a GUI, but AI models and databases (both building and using them) are known to the team.

Highlights
▪ *Highlight what was accomplished.*

We were able to get a working version of the AI module and the video module (both of which are not finished though), and a now working version of the GUI.
Changes
▪ *Summarize any major changes to any aspect of the project during the semester.*
▪ *Include the date, motivation, description, and implications of each change.*

The only change from the Project Proposal was made on 10/29/24, said change was to focus on only Linux OS's (mainly Fedora that we are testing on) for the time being as each OS platform (Windows, Mac, and Linux) has a different way to interact with webcams, which would triple (if not more) the size of the video module. This will limit the project to only working on a Linux machine, but this should allow us to make a better project in terms of bug fixing and progression.

Tool Links
• *Include direct links for your repo, project management tool, document/file share, and*
communication tool

Discord Link for main meeting/communication stuff:
https://discord.gg/FMYezBwRqm
Link to research documents/file share:
https://drive.google.com/drive/folders/1raV2WIC7KvB_1mbxC_mjki7QS2URfQ1I?usp=drive_link
Link to github repo:
https://github.com/bmgb5h/Capstone-1-Security-System-Project/tree/main
Link to trello board:
https://trello.com/b/xHrGiYEB/capstone-project

Presentation link in Panopto:
https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=a6b1716d-582d-4042-b27e-b23c005b998e

2. Customer Need

- *Who is the primary customer, outside the team?*
- *Describe the real customer, if there is one.*
- *Who are the secondary stakeholders?*
- *What are the stakeholder wants? Why?*
- *What is their desired overall experience?*

The primary customers would be those that do not wish to spend large amounts of money on a full security system, when they can use any type of camera they have at hand (or can find for cheap), and prevent name brand surveillance systems from giving out your information (like Ring has done in the past). One person that wishes to do this, is one of the team's brothers, (Jonathan's brother, referred to as JB). As the project has not reached much of a "proper" form, he has not been included in testing so far, but will next semester. A secondary stakeholder may be a small business that may want to use this system in place of expensive/subscription based applications. The stakeholders would want a streamlined but also configurable application to host their surveillance network. This should have a basic set of default options for computer/security illiterate users while being highly configurable for power users that may want many different options (different recording formats, notification settings, recording rollover options, motion/normal recording save path options, ect).
The desired experience should essentially be an easy setup for the user, followed by the easy use of the software. The customer should easily be able to set up the environment as they wish and feel as though they are in control. They should be able to make changes (adding cameras, changing configurations, etc.) just as easily. Along with the easy setup and changing of configurations, each user should have a user-friendly experience when using the surveillance system. The experience should allow the user to be notified of any security breaches, and easily give access to recordings captured by the system.

User Requirements

- *Include SMART user stories, written using the "As a ... I want ... so that ..." template.*

▪ *Include acceptance tests for the user stories, using the "Given ... when .... then ..." template.*

User stories:

1. As a user, I want to answer my door through my device so that I don't have to physically be home to answer the door.
2. As a user, I want to be sent a notification on my device whenever there is activity detected so that I am always aware of what's happening at my home.
3. As a user, I want important recordings to be automatically stored so that I can view them later whenever I please.
4. As a user, I want to be able to manually delete recordings that I deem unimportant so that I do not run out of storage.
5. As a user, I want unimportant recordings to be automatically deleted whenever I am nearing my storage capacity so that I don't have to manually sort through every recording.
6. As a user, I want audio to be recorded so that I can hear what's going on at home.
7. As a user, I want my recordings to be categorized into different events so that I can look for a specific event that happened.
8. As a user, I want to be able to upgrade my storage so that I can store more clips
9. As a user, I want to be able to access and view clips in the app so that I can view recorded clips from anywhere.
10. As a user, I want the option to store my clips locally so I don't have to pay to use storage services
11. As a user, I want the system to work with different kinds of recordings, (thermal, night vision, etc.) so that I can view recordings in different formats.

12. As a user, I want the system to be modular so that I can make the system as large or small as my needs are.
13. As a user, I want timestamps to be added to recordings so that I can keep track of certain events and the timeline when looking at recordings.
14. As a user, I want to be able to set up the recording loopback rules so that storage space is saved while keeping a record of a set amount of time in the recordings.
15. As a user, I want the system to have options in terms of how to access it (remotely via LAN/WAN, directly, via app, ect) so that I can look at my recordings without compromising my security.
16. As a user, I want the system to set up unique login identification for remote viewing so that the system stays secure.

Acceptance tests (in line with same numbered user story):
NOTE: when talking about "access to the system", it refers to having the correct credentials to login to the system and view/modify things in the system on the current device (phone, computer ect), and the system is currently able to be connected to from said device (ie, if internet is down, a phone would not be able to connect to it).

1. Given I have access to the system, when there is an alert at my door, then I can view and answer the door without being present.
2. Given I have access to the system, when there is a presence that is detected by the system, then there is a notification sent to my device(s).
3. Given I have access to the system, when the system detects an event, then the system will store these recordings for me to look at later.
4. Given I have access to the system, when I review and do not want to keep a recording and tell the system to remove it, then the system will "delete" that recording (whether that be a true deletion, or marked for deletion)

5. Given the system is running, when the system reaches its storage capacity or when it reaches its specified loopback time, then the system will start recording over the oldest recordings to save space
6. Given there is at least one microphone connected to the system, when recording the audio will also be recorded and stored within the system, then I can view the recording with audio.
7. Given the system is running, when the system detects an “event” (motion is detected, audio is detected, ect), then the system will mark this recording, then the user can quickly sift through recordings for important information.
8. Given the system’s hardware supports more storage, when there is more storage added and the system is directed to use said new storage, then the system will incorporate that storage into the pool of storage for the system to record to.
9. Given I have access to the system, when I wish to view stored recordings, then the system will send those requested recordings to my device.
10. Given the system is running and has storage attached to it, when the system is recording, then the recordings will be stored on the storage attached to it.
11. Given the system is running, when connected to different types of sensors (thermal, normal camera, infrared, laser, motion, magnetic, ect), then the system should be able to use those sensors to store and record the information being received from those sensors.
12. Given the system is being installed, when the user wants to remove or add functionality to the basic system, then the functionality is removed/added to the system when installing.
13. Given the system is running, when recording/processing video, then timestamps will be overlaid onto the video that is recorded for the user.
14. Given the system is running, when setting up certain options, then the system will abide by those options set by the user (this one will apply to any option setting user needs).

15. See AT #14
16. Given the system is running, when logging into the system, the system should create a unique identifier for the user logging into so the system stays secure.

3. Project Goals

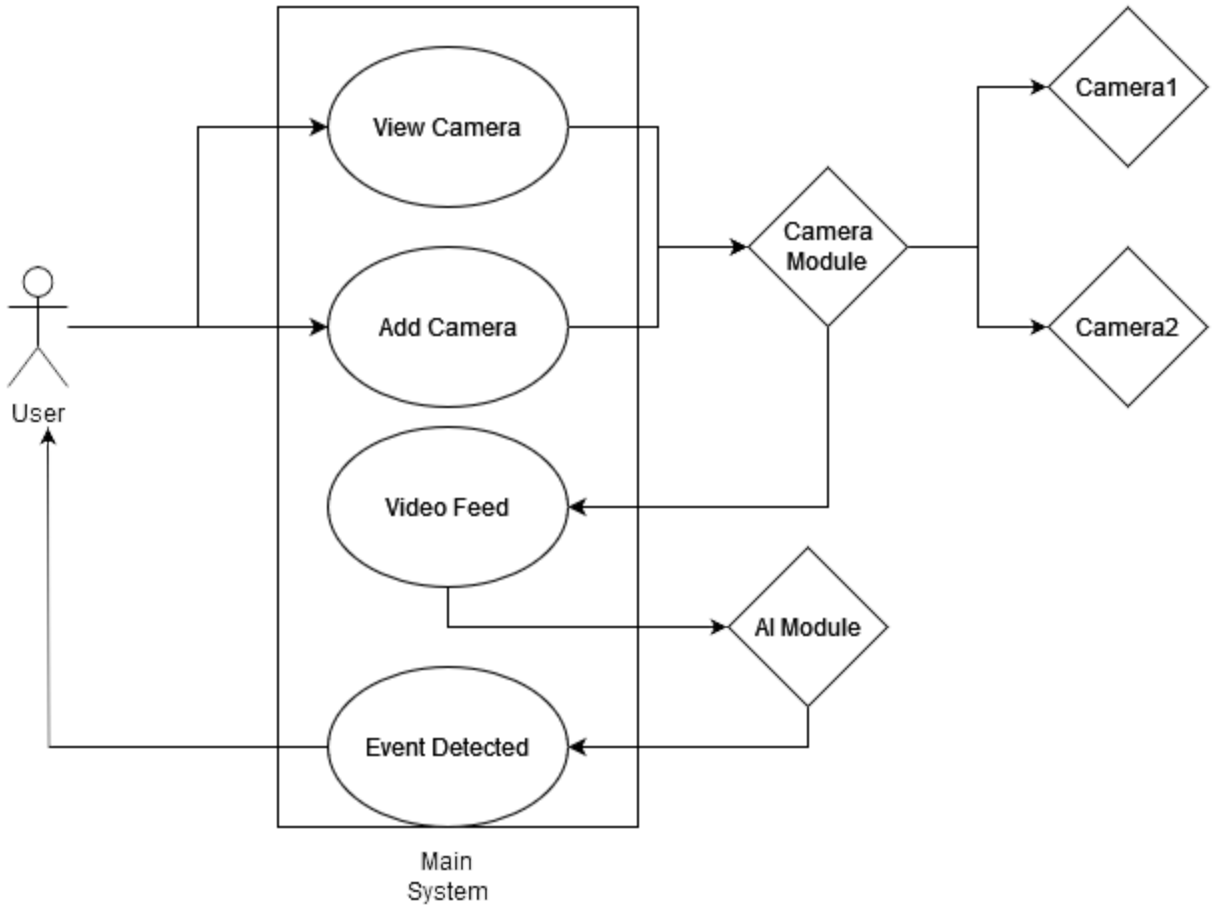
▪ *What customer problem did you choose to address?*
▪ *In implementation-free terms, what user benefits does your system provide?*
▪ *How does the benefit support the customer's desired overall experience?*
▪ *Who did you validate the idea with?*

We are addressing the problem that proper home/small business security systems are expensive and may not be secure/private as many security systems will store data on their own servers, and in some cases, will allow authorities to view said data without consent of the customer. In order to do this, our system will allow in the very least personal devices (ie, webcams, microphones, ect) and possibly name brand devices to be used in the safety, security, and privacy of their own home as a security system. This will benefit the customer by giving them the option to have more control over recorded information that they may deem sensitive. Put simply, it gives the customer the option to remove the middle man. We asked JB, and he agreed with the privacy and expense problems and that free, self hosted, and secure software to use webcams or other camera/surveillance devices would help fix both of these problems.

Uses Cases

▪ *Include a use case for each main user goal for a primary or secondary customer.*
▪ *Show the title, user goal, and full basic flow for each use case. Choose meaningful titles.*
▪ *For alternative flows that have been implemented, give only the title, a one-line description, and how the alternative flow connects with its basic flow.*

Use cases:
Diagram for use cases:
Note on the diagram: Camera1/Camera2 are not the "max" amount of cameras that can be added to the system.



Title: Basic flow: View Camera
User Goal: View a camera's feed
Basic Flow:

1. The user logs into the system
2. The user clicks on camera(s) tab in UI
3. The user selects camera they wish to view
4. The system will find and display that specific camera's live feed
5. The user exits the system after some time.

Title: Basic flow: Add Camera
User Goal: User Adds a camera
Basic Flow:

1. The user logs into the system
2. The user clicks on camera(s) tab in UI
3. The user clicks "add camera"
4. The system will try to detect any cameras connected and display them
5. The user clicks on a camera that has been detected

6. The user changes settings to their liking (name of camera, resolution, frame rate, ect)
7. The user submits the changes
8. The system saves the settings and needed data to a config file
9. The user leaves the “add camera” tab
10. The user exits the system

Title: Alternative flow: Camera not detected
Description: if the system does not detect the wanted camera
After step 4 in Add Camera:

1. The user will select “camera not found” tab in “add camera”
2. The user will input the camera’s USB port, wifi configuration, or some other path to said camera needed
3. The system will connect to the given camera

Then proceeds at step 6

Title: Basic flow: Event detected
User Goal: Be notified of an event and look at it
Basic flow:

1. The system detects a movement in a video feed via event detection module
2. The system sends some form of an alert to the user (via phone notification, sounds or other ways)
3. The user logs into the system
4. The user clicks on the “events” tab
5. The system collects all stored events and displays them for the user to see
6. The user clicks on an event they would like to view
7. The system opens that saved video of the event and plays it for the user
8. The user closes the “events” tab
9. The user exits the system

Measures of Success

- *How do you know whether the customer got their desired benefits?*
- *What are your customer-centric measures of success?*

JB will help in testing the project, as he has been looking into a security system that is self hosted and secure. One of the ways will be through testing with JB and updating things based on his account of the testing. Our main customer-centric measures of success include overall customer satisfaction and customer effort (how easily a

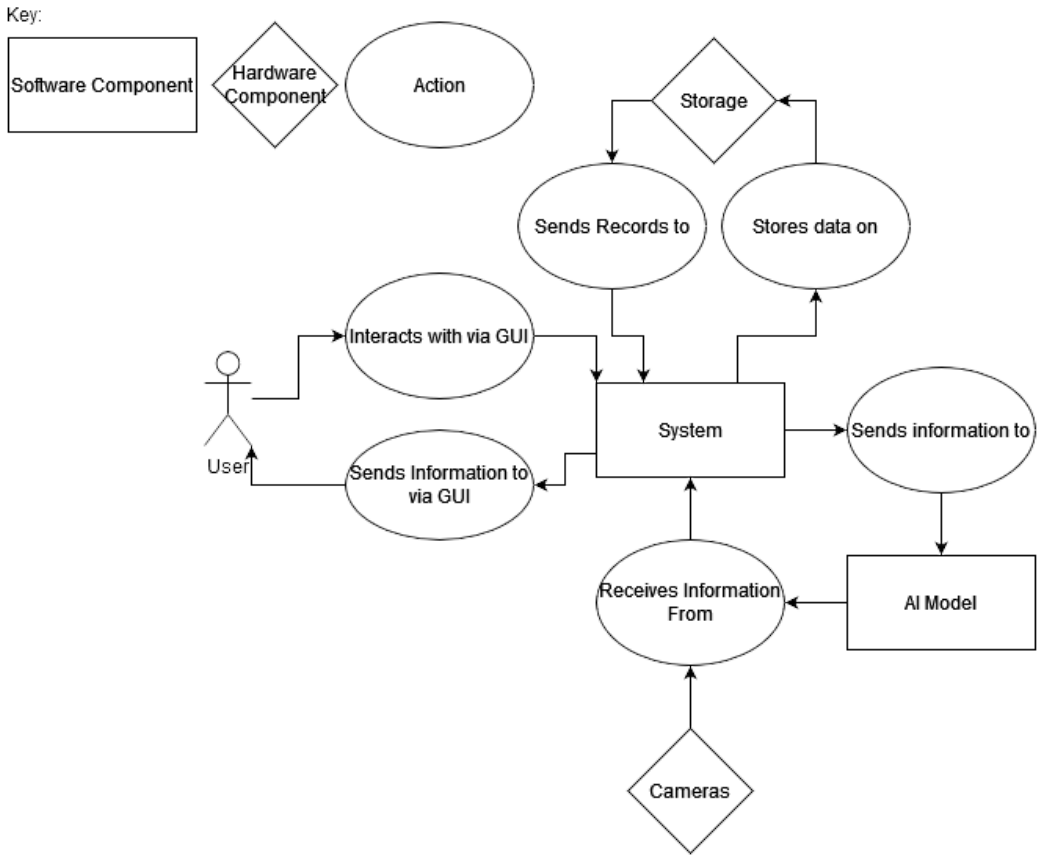
customer can interact with our system). We intend to collect these measurements by doing product tests ourselves as well as with our single outside "customer." Analyzing these results will give us an idea of if the customer actually got their desired benefits. Sadly, we were not able to test anything with our customer JB, as the project has not reached a "user friendly" form just yet.

4. System Description
System Overview
▪ *Introduce the system and the main challenges.*
▪ *Summarize the rationale for the design.*
▪ *Include a Context Diagram showing how the system interacts with external services, databases, and so on. Clearly mark the boundaries of the system.*
▪ *Briefly explain the system's external interactions.*
Guides to the Main Architectural Views
▪ *For each view, provide a Primary Diagram.*
▪ *For the architectural elements in a Primary Diagram, provide an Element Catalog with the name and purpose of each element.*
▪ *For each element, identify the single owner in the team, even if multiple team members contributed to the element.*

The system currently consists of an API that acts as the middleman between the user and the things the system interacts with, like the database and cameras. The camera module is mostly complete, with some issues regarding how to address each camera (mentioned below). This module will first read a configuration file (and if one is not present, generate one along with a readme file for the user to currently look at for information about the settings and let them know to change the settings in it), and use the settings from that file (like resolution of the video, frame rate of the video, ect) to open up the camera and start creating temp video files for the main system to read and use. Currently, the GUI is linked with the AI module, but the video module is NOT linked to the GUI/AI module.
The AI module currently uses its own camera opening code (so no interaction with the camera module for now), and will look for "packages" and will record a picture of the box detected.
The GUI currently only has an interface for starting detection, viewing live feed, deleting tables from the database, adding cameras to a database, and viewing the current cameras in the database (NOTE: this currently does NOT actually add the camera(s) to the system/config, only to the database and does not interact with the video module, but does with the AI module)

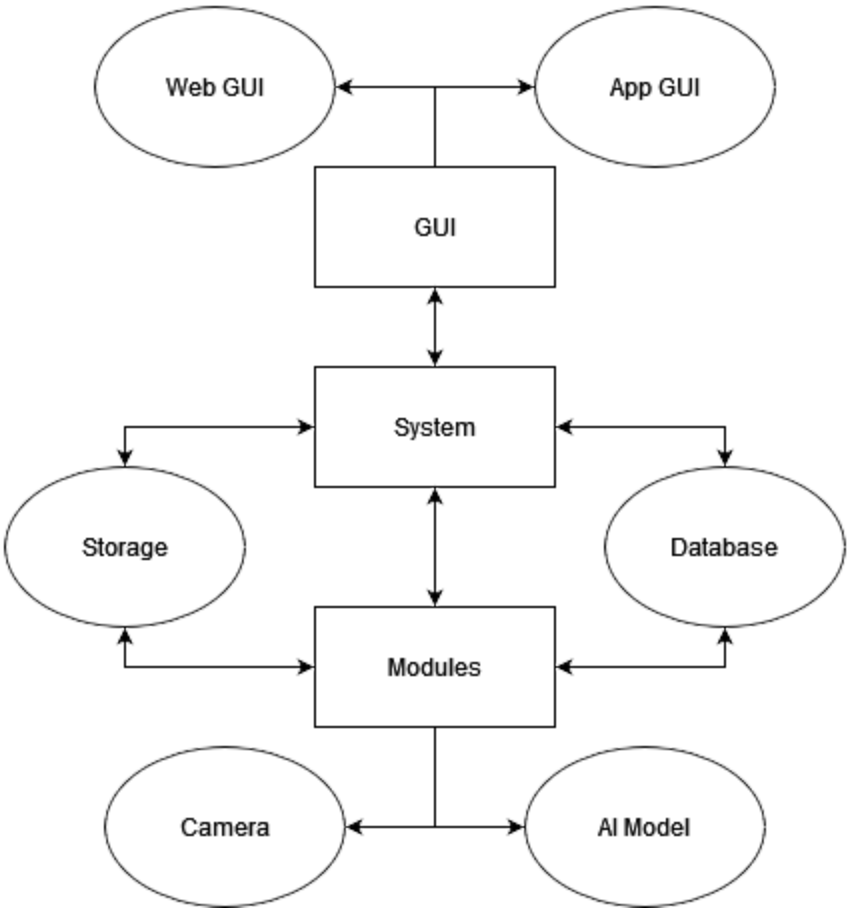
Rational: Since many of the modules can be added/removed from the system, the system has been/is being designed to work with different modules/lack thereof. While some modules will be interconnected (ie, the AI model will require the video module to function), the non-interconnected modules will not interact with each other, nor be affected by the presence/lack of presence of other modules.

Context diagram:



External Interactions: The system will interact with the AI model (which can be changed by the user, or removed), a database that will be created and managed by the system (so semi external), cameras that are attached to the system, storage attached to the system, and user inputs.

Main architecture views:
Logical View:

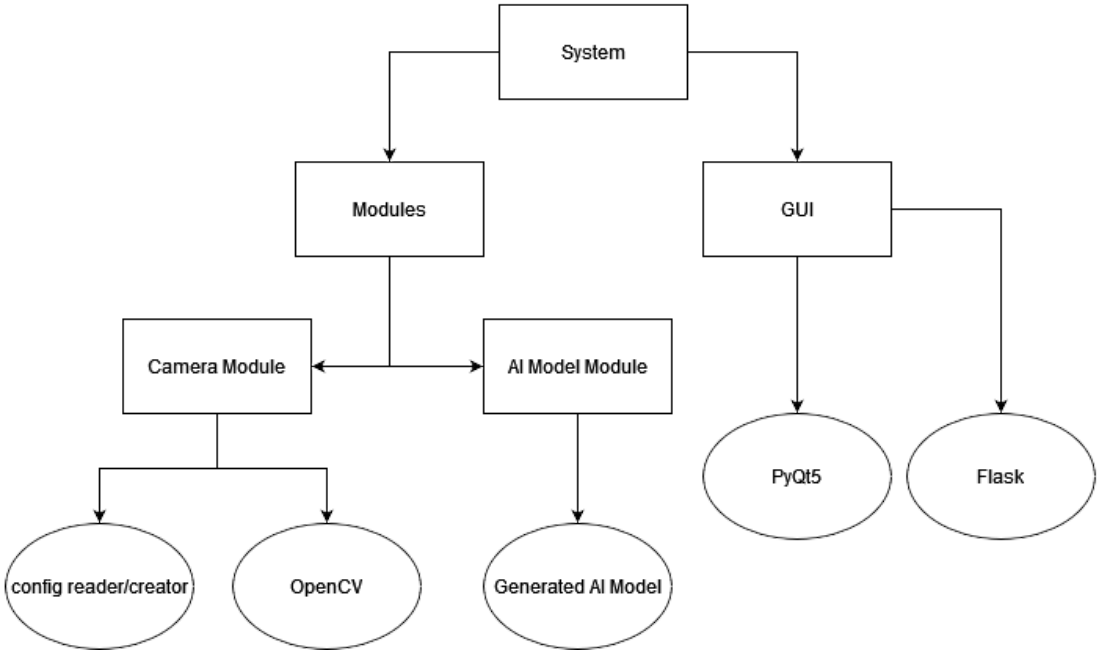


Element Catalog:
GUI: Owner: Jeremy
Purpose: Display information about the system to the user, and allow the user to interact with the system, so far everything under GUI has been created by Jeremy

System: Owner: Brenden
Purpose: Take in information from the modules and relay said info to the appropriate area (GUI, storage, database, other modules), and will control all other modules (and GUI), so far everything under system, and the AI module has been created by Brenden

Modules: Owner: Jonathan
Purpose: Use external devices/programs (cameras, AI models, ect) and convert the information given from said devices/programs to a format the system can use (mainly for different formats from the same type of source, ie, different video cameras may record in different formats, the module will handle converting all of these to the same type), so far everything under the camera module has been created by Jonathan

Development View:

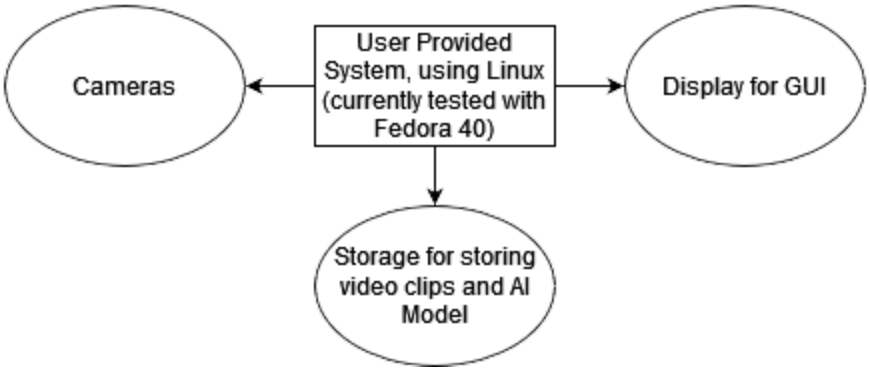


No changes to the above element discussion

Dynamic View:

No changes to the above element discussion

Deployment View:



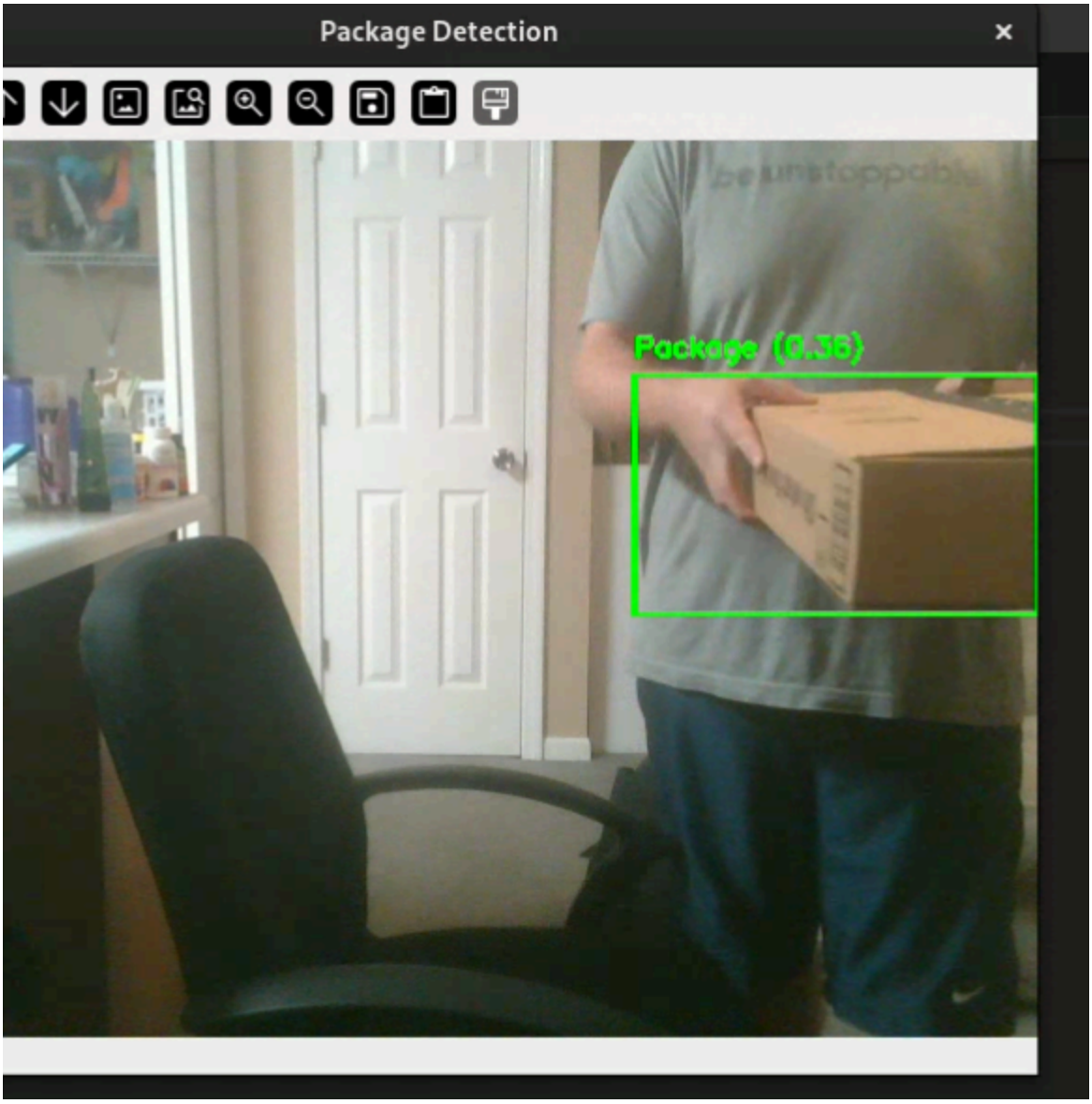
No changes to above element discussion. Seems very simple though.

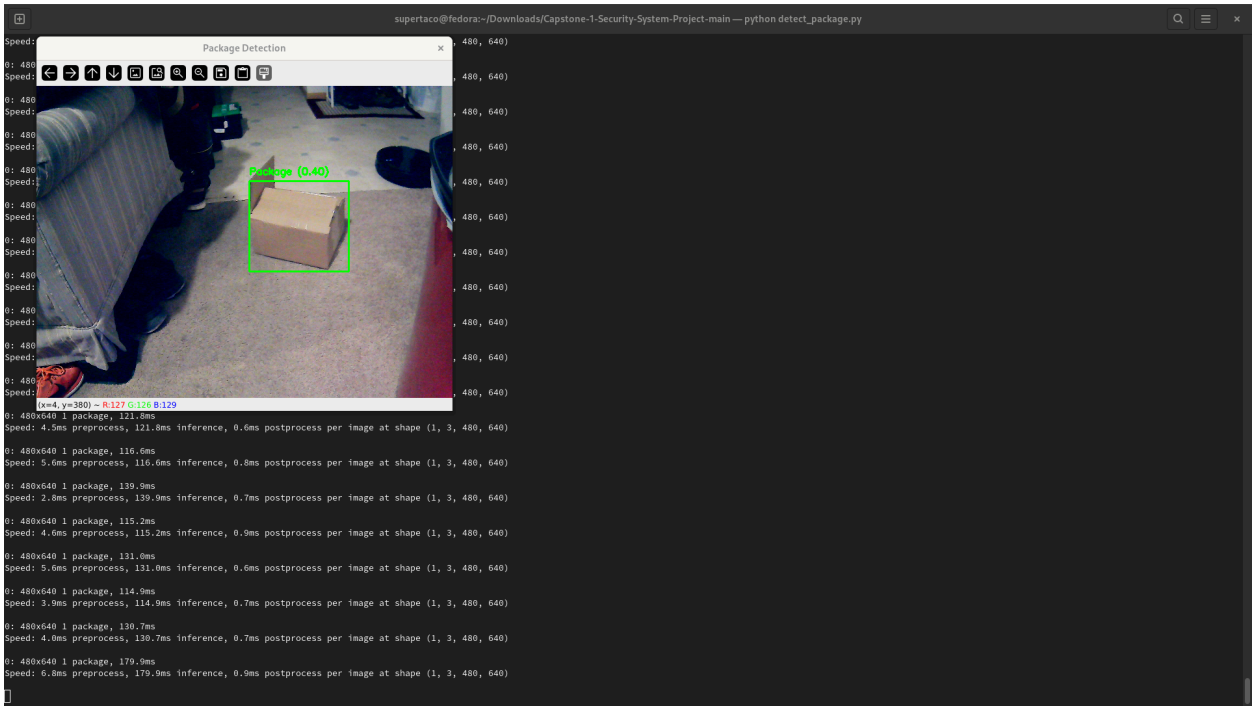
5. Final Status

- *What works? Include screenshots.*
- *What tests have you run?*
- *Estimate the adequacy of your tests*
- *How many lines of code has the team written altogether?*

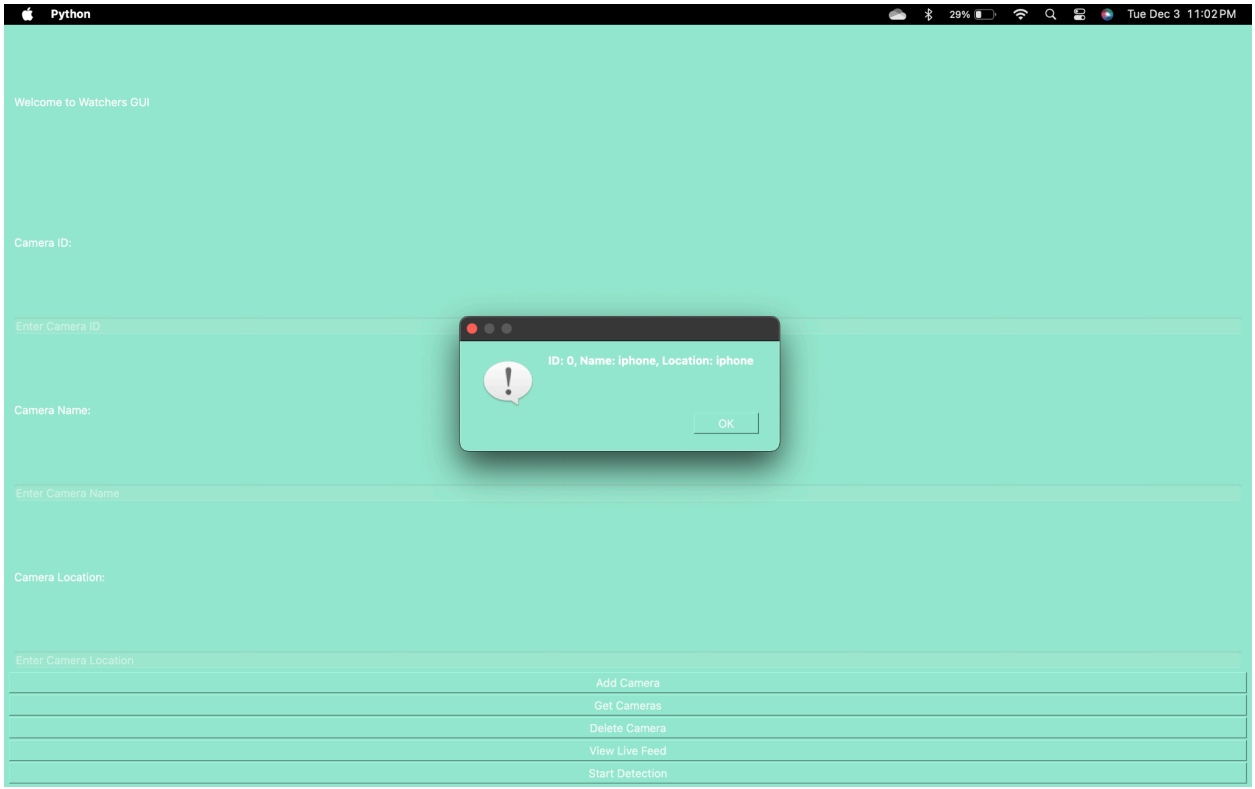
Currently, the video-to-model system is functional, but the model isn't detecting as well as it should be. This is probably due to the small dataset we are using. Unfortunately, there isn't a large publicly available dataset that is also annotated, so we may just have to deal with it for now. In the future, we may build our own dataset by scraping images of packages of the web and then manually annotating the images ourselves. Another idea could be to annotate the frames that come directly from the security cameras, that way the model could be more specialized for this task.
The frontend page for adding a camera to the system is also functional, and does correctly update the database, but does not interact with cameras or config files generated by the video module.
The video module is mostly working, still has the issue of not grabbing camera ID's that work with OpenCV, but otherwise works fine.

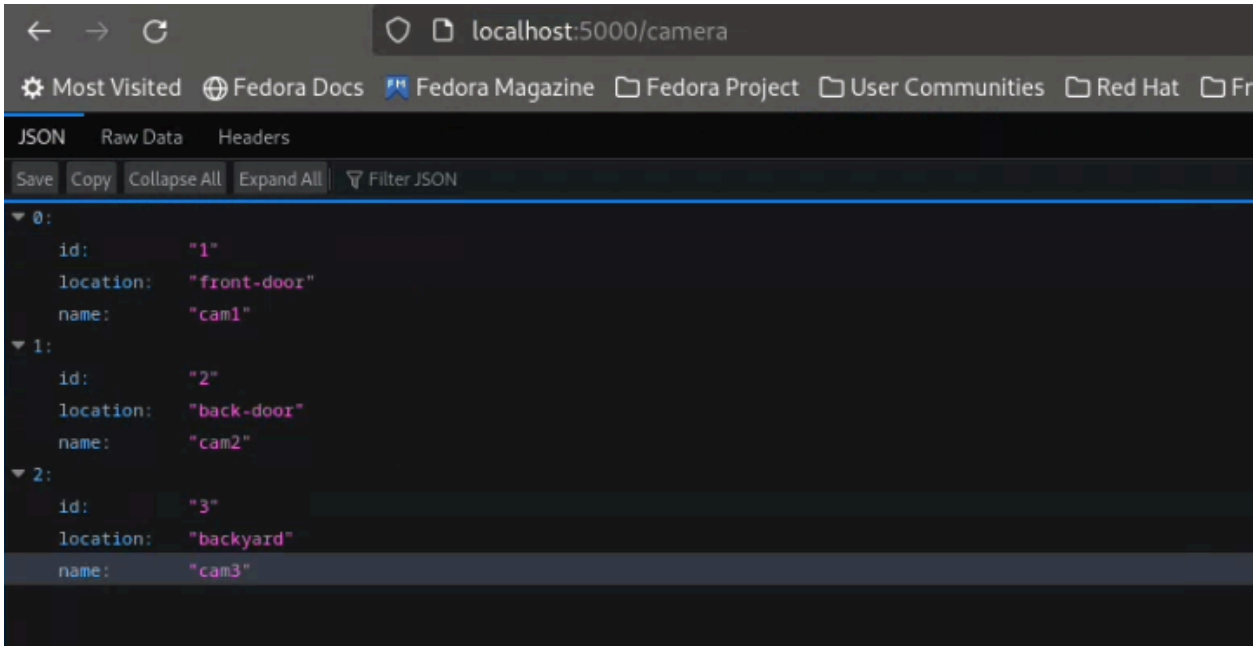
Package Detection/AI screenshots:


Package Detection
Package (0.36)



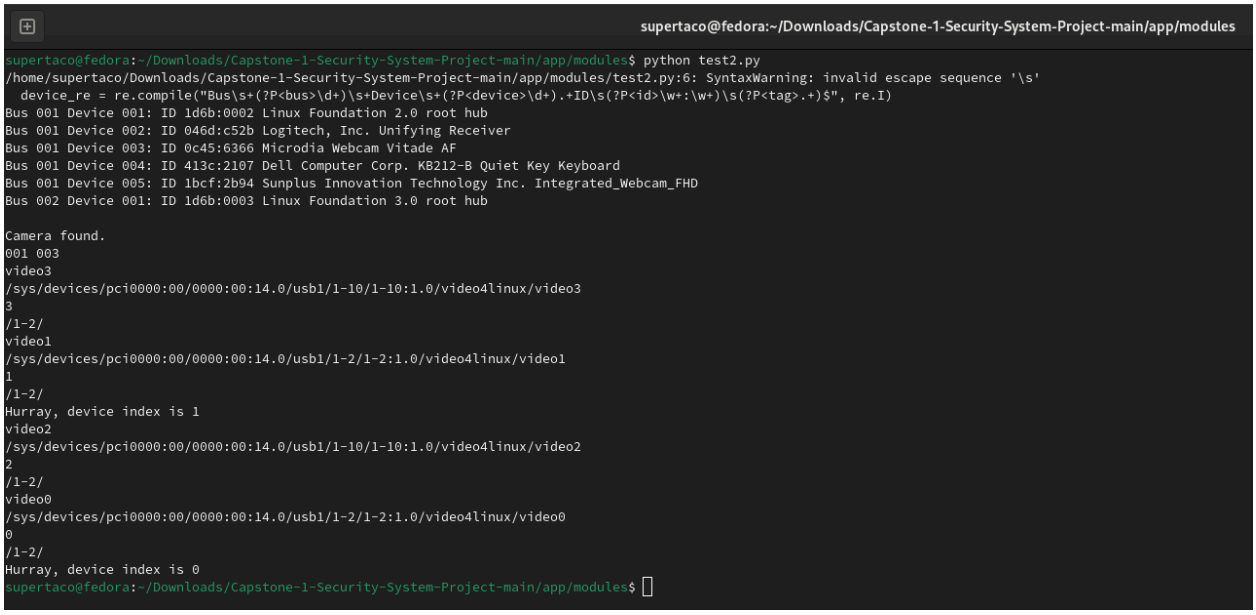
GUI screenshots:



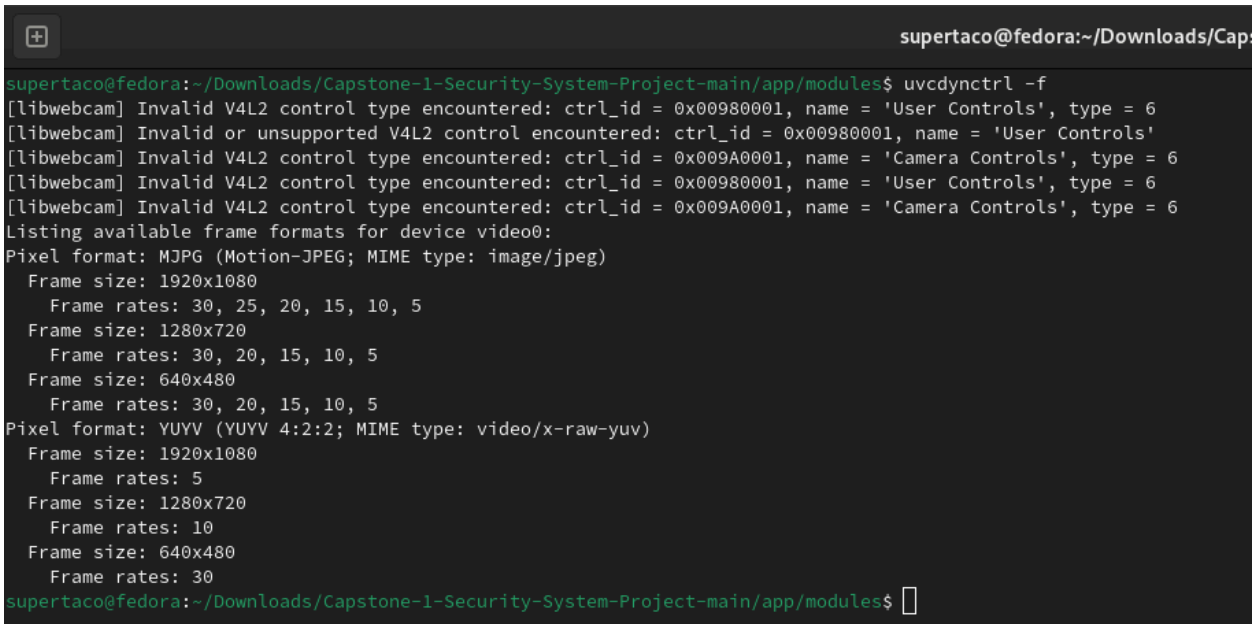


Video Module Screenshots:

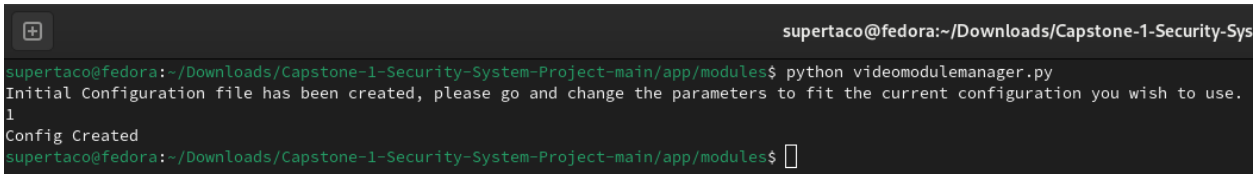
Running the USB device enumerator (not the one OpenCV uses), Note that this has the device ID for the webcam that we would like to use, and displays both cameras on the system (the external one, device 3, and the integrated one, device 5).



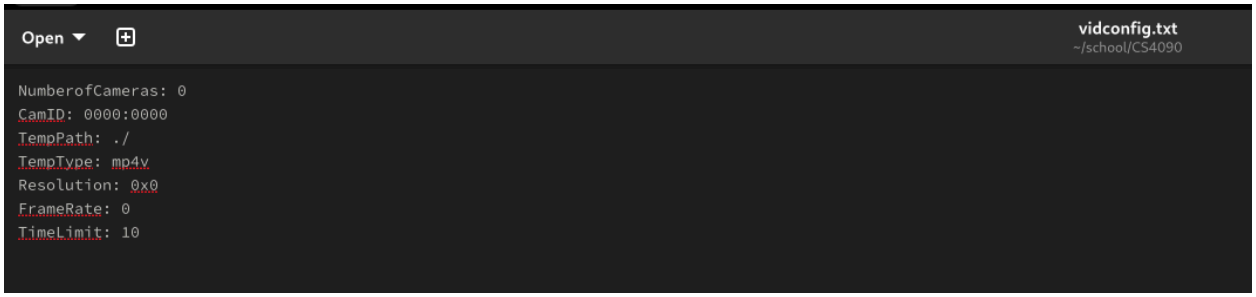
Trying to find the correct camera using the device ID and name and linking it to the correct video number that OpenCV uses, and this was successful (in this case, we were looking for the external camera, and this is occupied by both index 1 and 0)



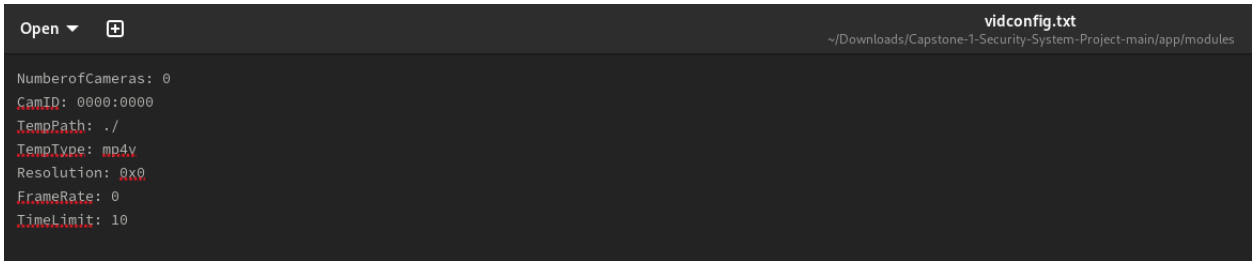
Looking at the camera details, this shows the supported file types, resolution, and frame rate of the camera



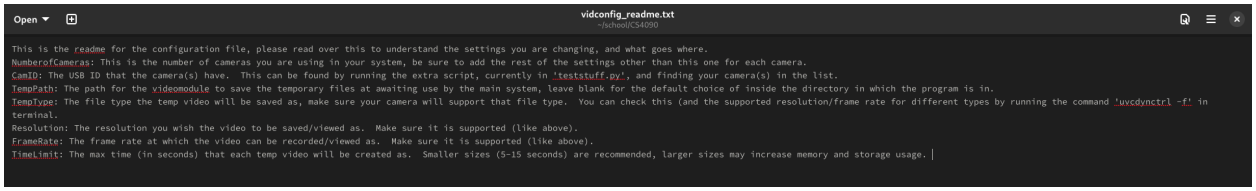
Config creation test



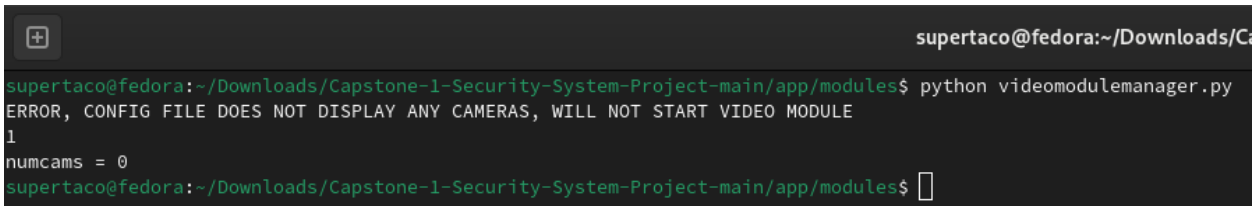
Generated Config



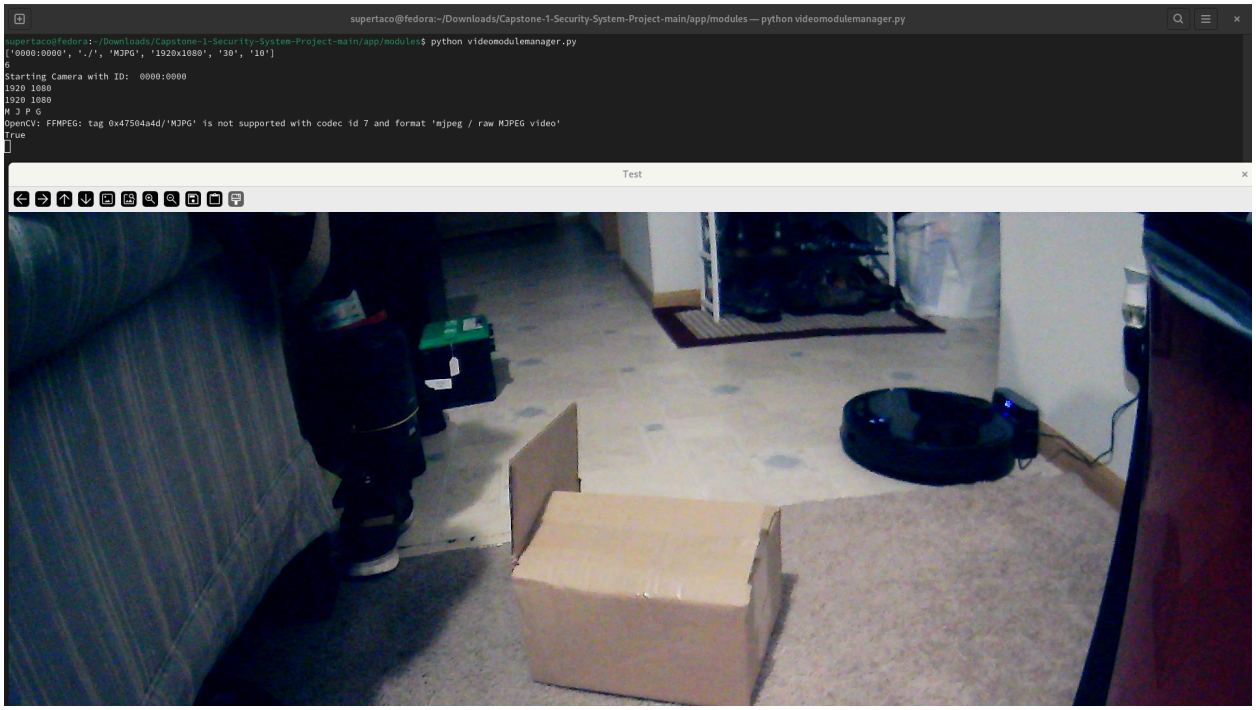
Changed Config, note: the camID is currently blank, and the timelimit is unused for the time being



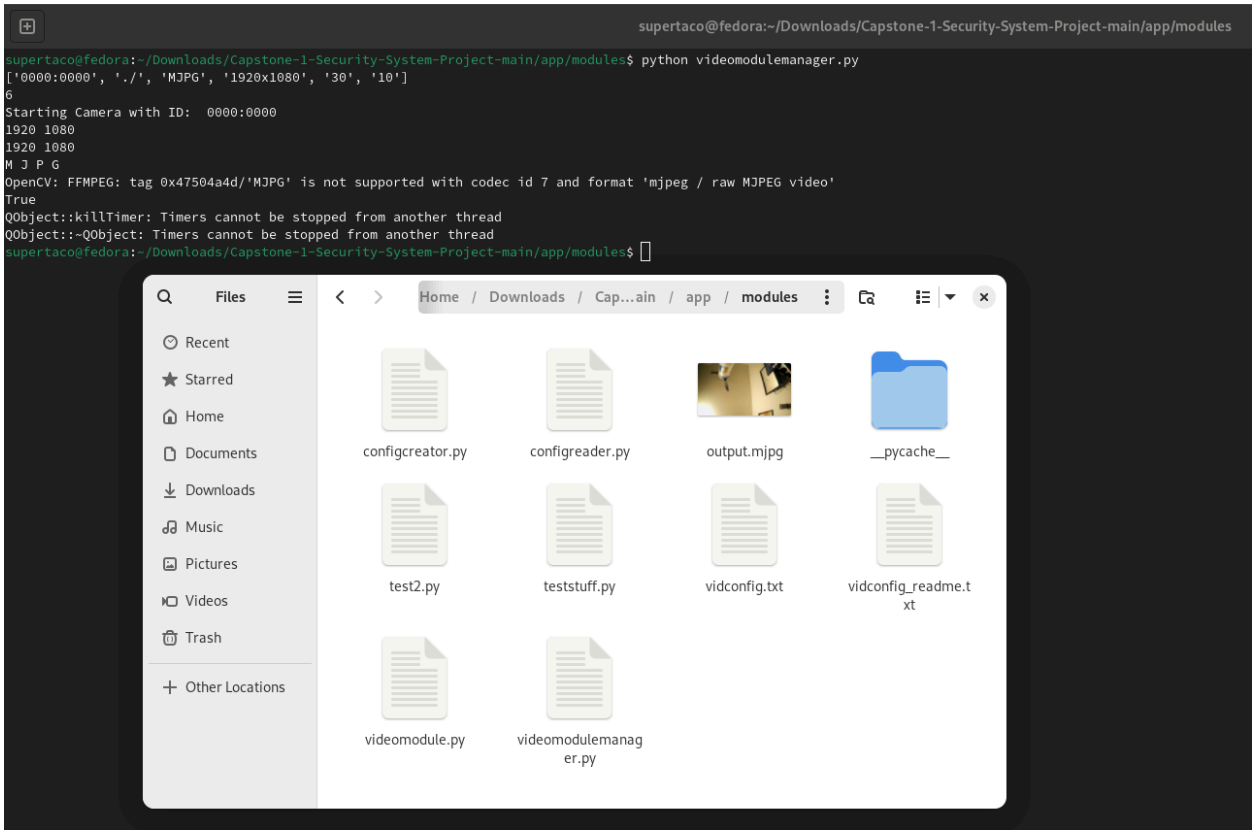
Generated Config readme



Running the Module *without* changing configuration file



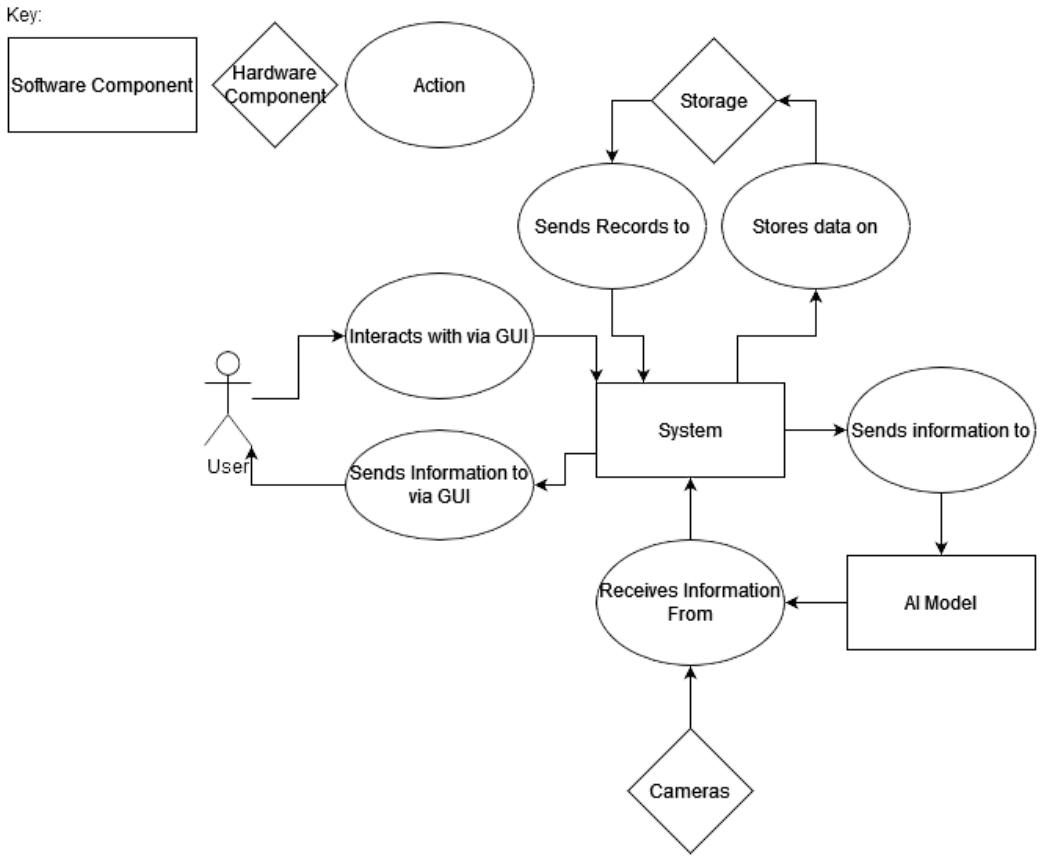
Running once again *after* changing config file, part of video shown (screen size is 1920x1080, video is running at 1920x1080, so only part can be shown with terminal output), Most of the displayed text is currently generated as internal debug other than the “warning” and “WARN” lines (those are generated by OpenCV or other modules it uses), the internal camera was also tested, and working. Note: during testing, we noticed that the internal and external video camera index switched after restarting the system, giving further proof that relying on the OpenCV device enumeration is unstable and needs to be fixed.



After exiting the program (I believe the the “timers” notifications are about how threading is implemented and possibly exited incorrectly), note the generated output.mjpg file, this file does show the camera feed, but not at the correct framerate (possibly due to the type of format), and is much slower.

A major challenge for the camera module is how the current library that we are using tries to grab camera data. We are using OpenCV in Python, and the way this library addresses cameras is by enumerating them based on when they are connected to the system (and nothing else). Restarting the system can change this, and removing one camera can change what settings are going to be applied to what camera (which can cause the module to fail to capture data from the cameras if those settings are not supported by the “new” camera). We were trying to use the device ID of the camera to link each camera to their individual settings, but it is proving difficult to convert these IDs to something OpenCV will use or vice versa. The current testing of the module is just using the enumerated version of the camera that OpenCV uses, which allows us to test the module and *most* of the config file settings.

Aside from the subpar model, we need to ensure that the user isn’t spammed with delivery notifications. Our solution for this will likely be a timeout that stops the model from triggering a notification. We also need to figure out how to integrate the video configuration stuff with the main system.



(the block diagram and the context diagram are the same diagram, both fit well)

Most of the testing is just to make sure the parts work in some way or another. Currently the “working” part is a pass, but working properly in all sections is a fail in our opinion as the system is not in a “user friendly” form or working order for a user to test it. If we had a full system that is able to be tested properly, we would have automated tests to make sure videos are generated at the correct specifications (resolution, framerate, filetype, length, ect), GUI elements are clickable in the correct spaces and cannot be “broken”, and inputs to the GUI and config files will not break the database/backend functions, making sure the AI model will continue to find boxes, and make sure notification systems work properly. We would also send the application to JB for him to do some informal usability testing and general user testing.

Collectively, we’ve written about 700 lines of code.

6. Project Management

▪ *Describe your actual development process.*

▪ *What were the major events during the project? Include dates.*
▪ *Anything else?*

Our development process was pretty unstructured, but we tried to adhere partially to a SCRUM process. We started with the main features we wanted our product to have, then went from there. There weren't really any major events, a small (but important event) was when we decided to use the AIO computers (mentioned above). The dates that were most major were probably the days the sprint reports were due as that's about when we would make the most progress.

Team Coordination
▪ *When and how often did the team meet?*
▪ *What did you accomplish during the meetings?*
▪ *How else did you communicate?*

We would typically meet about 1-2 times a week after class. We used these meetings to discuss the direction we wanted to move forward in, progress of the different components of the project, and issues we encountered. Alternative communication was done through Discord.

7. Team
Backgrounds
▪ *What were the backgrounds of the team members?*
▪ *Did anyone have prior internships or work experience related to software?*
▪ *Had anyone on the team built something like this before?*
▪ *Were the tools known or new to the team?*

Brenden Guillen

Background: CS major with zero experience doing professional/commercial software development. I enjoy puzzles and logic, which is why I decided to go down the route of CS. I also heavily enjoy math as a whole and the math behind algorithms, especially AI/machine learning ones. Outside of school, I work as a part time package handler.

Software Development Experience: As I mentioned, I do not have any professional experience doing software development. The only software I've developed has been for course work or for personal projects.

Similar Works: I have worked on a couple AI based detection models for some of the courses I have taken. My experience with that might be useful for categorizing the content of a recorded clip (what objects were recorded in the clip.) I also have some experience working with database system software from the two database classes I have taken, which will definitely help when figuring out how to store recordings and other important data. Other than that, I don't have much other experience related to this project.

Known/New Tools: I haven't worked with cameras or any camera software directly, so that will be very new to me. I'm very familiar with Python and a couple of its machine learning related libraries. I have some familiarity with HTML, CSS, and JavaScript, so those skills will definitely be refined in this project.

Role: I will be doing a little bit of everything. Researching, developing, compiling a dataset (if necessary), testing, etc.

Jonathan Thornton

Background: has car mechanic background (no official job experience, just personal repair/helping family), tinkers with many things (both hardware and software), currently going for a CS degree with an emphasis on AI.

Prior work: did quality for a thermoset/thermoplastic machine shop (mainly helped build more attention to detail, and also created some work macros to help speed up some job processes), and other than class experience (database project, and AI model for math in machine learning project), none.

Similar works: other than the possibility of including an AI model for detecting objects/people in the surveillance system, no.

Tools known/new: I have tinkered with a few different security cameras and webcams along with some streaming software (OBS mainly), but have not dealt with low level coding for this type of project using video and such. I do have experience with python and AI models, so that may help if we get to that point in development
Planned roles: researching topics, helping create the AI module if we do that, setting up the layout of the system and the modules themselves as well as working on them, sending tests for my brother to try out and receive his feedback.

Jeremy Bell

Background: CS major with no professional software development experience, however, I do have experience in a professional role with databases. I have been involved in robotics since the 7th grade which led me to choose computer science as my field of study. Aside from attending school full-time, I am currently working as a database administrator for Missouri Department of Transportation and have the opportunity to join the company full-time upon graduation.
Prior Experience: My prior experience includes robotics, and databases. I worked on the robotics team for my high school as the team leader, and I have been working as a database administrator for the past couple of months. In addition, I have done computer science projects for previous classes.
Similar Works: This will be my first experience with any sort of surveillance system.
Known/New Tools: I have not done any work with surveillance systems, but have experience with multiple programming languages. I intend to put these skills to use in the design of this project.
Role: I plan to work on multiple aspects of the project. I will help with research, and programming where needed mostly.

Most of the project was new to the team, none of us have build a GUI before or used OpenCV (or other packages/libraries) for video cameras. The database and AI model parts were small, but the team knew those parts from previous school experience.

Roles
- *What were the roles of the team members during this project?*
- *What did each team member contribute?*
- *Use the Element Catalogs to identify individual contributions.*
- *Give rough estimates of percentage contributions by each team member.*

Brenden: Dealt with the backend aspects of the app such as app routing, model training and integration, and integrating push notifications (33%)

Jonathan: Dealt with the video module, config/readme creation for said module, and help with research. (33%)

8. Constraints and Risks
- *Were there any social, ethical, policy, or legal constraints?*
- *Did you have access to the data, services, and resources you needed?*
- *Was there anything else you needed?*

We did have some ethical and social constraints that we had to deal with. Recording is something that some people may not enjoy, especially if those people are the ones being recorded. Our product is designed to be used on the user's own private property, which is perfectly legal. There is the potential for someone to use the product for malicious purposes, and that is something that we did not consider much when developing. If the product were to become a big deal, we would have dedicated more time to prevent our product from being used for those malicious purposes.

We had access to most of the services and resources we needed, but we didn't have much access to the data we needed. We would have benefited from having more data to train our package detection model with, but it just wasn't something that there was an abundance of. One resource we also did not have much of was time. All of our schedules were packed with both work and school work, which left little time for us to collaborate in person or progress on the project as far as we hoped we would.

We needed more cameras to test with, which did have access to. We also ended up doing all of our testing on all-in-one PCs for consistent testing, and we also had access to those. The main thing we needed was more time to work on the project.

9. Reflection

- *What were the lessons learned from doing this project?*
- *What went well?*
- *Which of your practices would you consider to be best practices?*
- *What didnt go well?*
- *What isn't working and how did you work around it?*
- *For the features that were not implemented, what were the issues?*

The most important lesson we learned is that coordinating projects within a team is very hard. We can see why frequent meetings and project management tools are important when working with a team.

Despite facing many obstacles during the project, our team chemistry was pretty good. We were able to discuss issues we were having, and then bounce ideas off each other for how to solve them. Even though our project isn't as polished as we were hoping, the experience of working on something big was valuable.

We can't really say any of our practices are really best practice. One thing we did do frequently was testing, which we would say is best practice. We also all tried to write and maintain clean code, but we didn't set up many project standards for clean code, so everyone's idea of "clean" was a little different.

One of the biggest disappointments we had was the performance of our package detection model. Unfortunately, there were not many publicly available datasets, so we just settled on one with about 30 images. This was definitely not enough data to train with, but we wanted to continue making progress on the project rather than improving the model. We could have collected our own data via web-scraping or taking our own pictures, but this would have required a lot of time for annotating every image. We decided to stick with our current model and just work on more features instead. Another disappointment is that we did not finish the project in a way that can be "used" by another user (like sending it to JB for testing).

Some of the features we didn't get to implement were general event detection and automatic database population with camera data. We found that it was difficult to locate all available cameras, so the user has to submit an html form containing the id

of the camera so that the system can interface with it and allow the user to view it. We wanted the process to be more of a plug and forget type of interaction, but our current setup does require the user to be somewhat tech savvy.

Recommendations

- *What would you do differently?*
- *What advice do you have for other teams?*

If we were to go back and do this project differently, we would dedicate a lot more time to planning to ensure that everyone knew exactly what was to be expected for the project. We would have also had more frequent meetings to keep everyone on the same page. Finally, we would have started with smaller goals and worked up from them so that we aren't too focused on considering future tasks.

Our advice to other teams consists of 3 main points: plan heavily, communicate frequently, and don't bite off more than you can chew. Planning will make everyone more prepared, communicating will keep everyone on the same page, and not taking on too much will make goals more achievable.