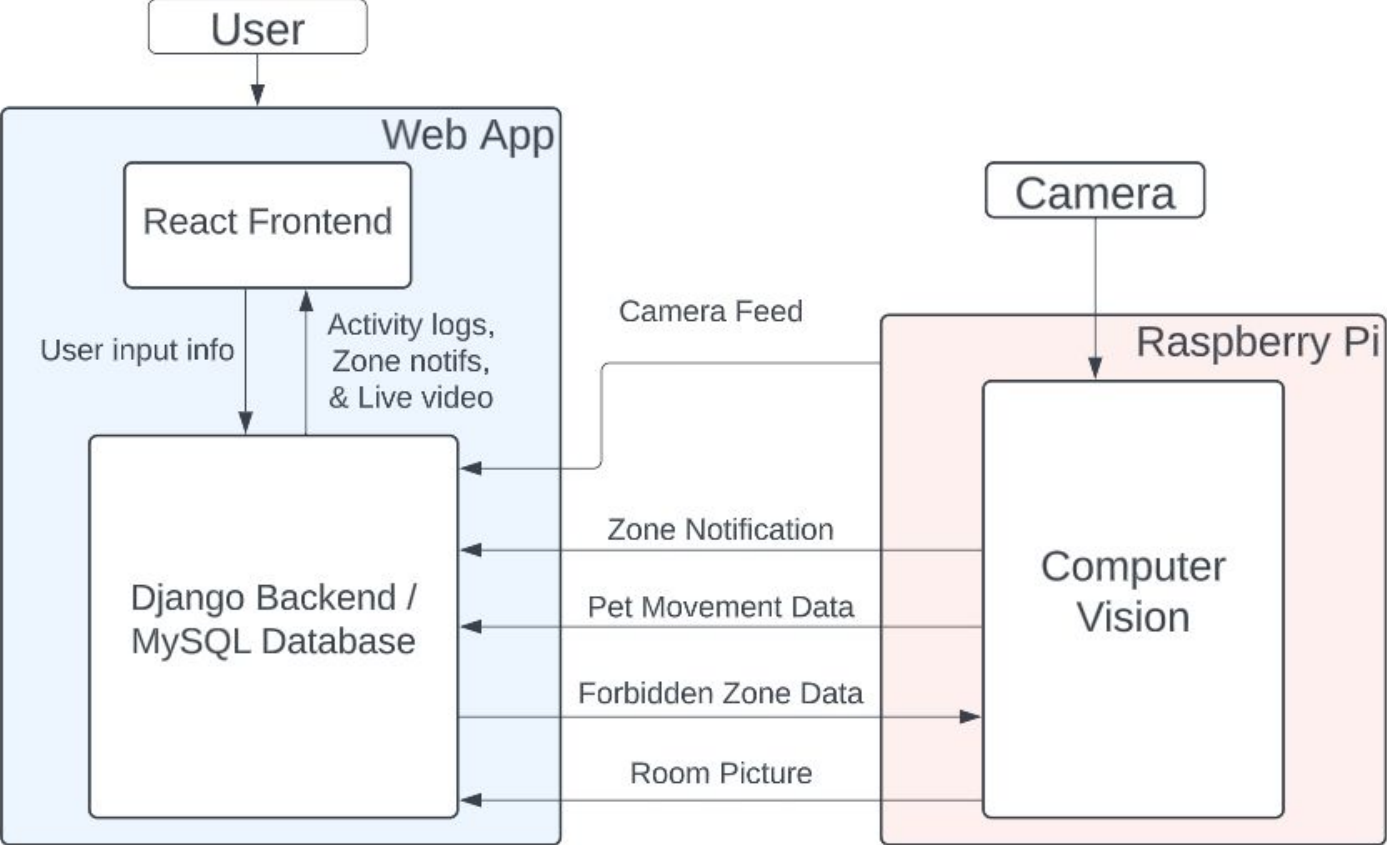


# Use Case Requirements

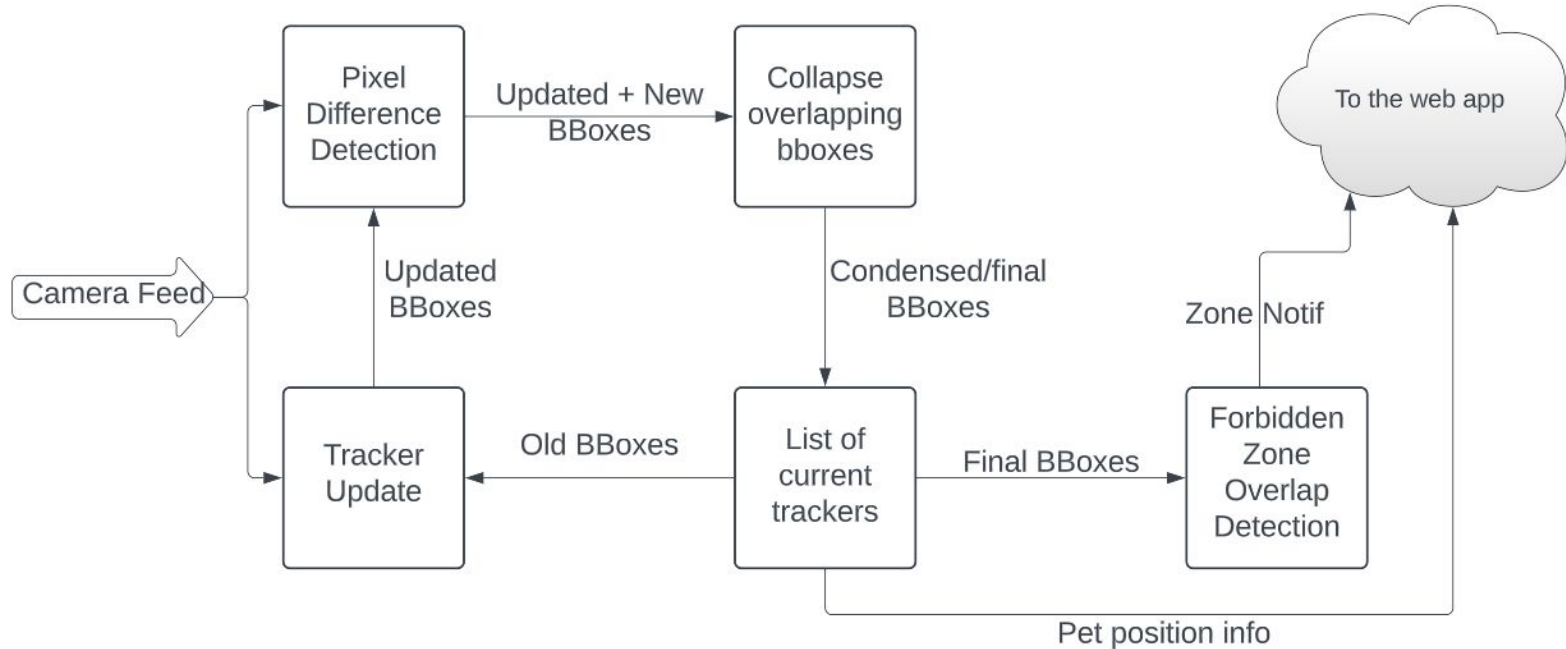


The overall goal	Our use case requirement(s)	The design requirement(s)
Detect and report when a pet goes somewhere it shouldn't	False positives <b>&lt;10%</b> of the time	Tracking accuracy w/in <b>1ft</b>
	zone report speed <b>&lt;10 seconds</b>	Zone detection w/in <b>1 sec</b>
Provide a log of pet activity	Logs are <b>&gt;90% accurate</b>	Tracking accuracy w/in <b>1ft</b>
		New animal detection w/in <b>5 sec</b>
Maintain system accessibility	system setup in <b>&lt;5 min</b>	-
	system cost <b>&lt;\$100</b>	Use RPi and simple camera
	<b>&gt;95%</b> of users can accomplish tasks easily	-

# Solution Approach (Overall)



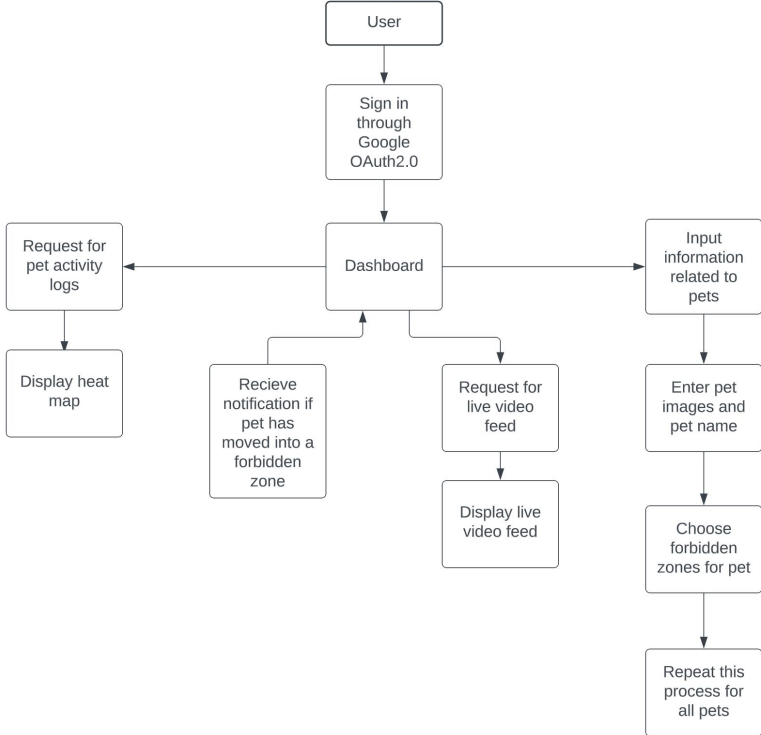
# Solution Approach: Computer Vision



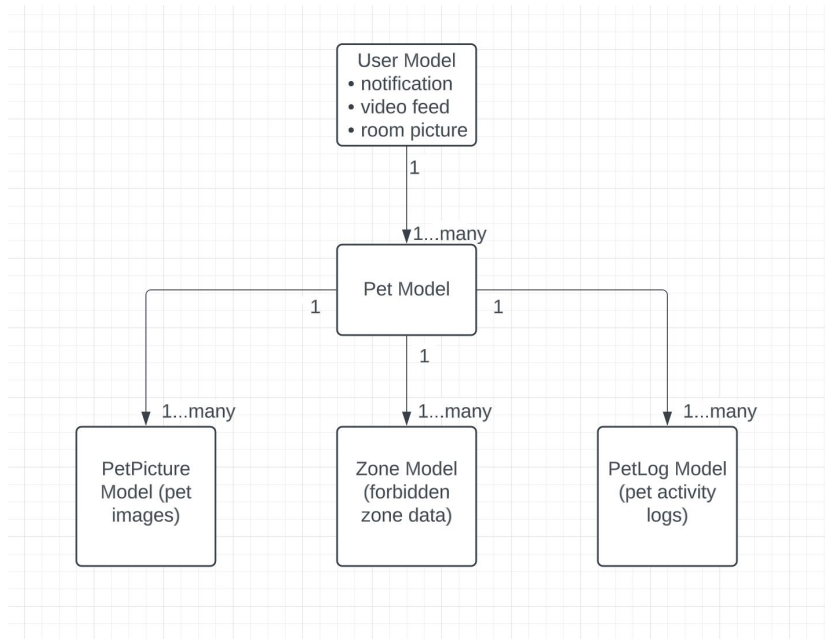
# Solution Approach (Web App)



## Frontend



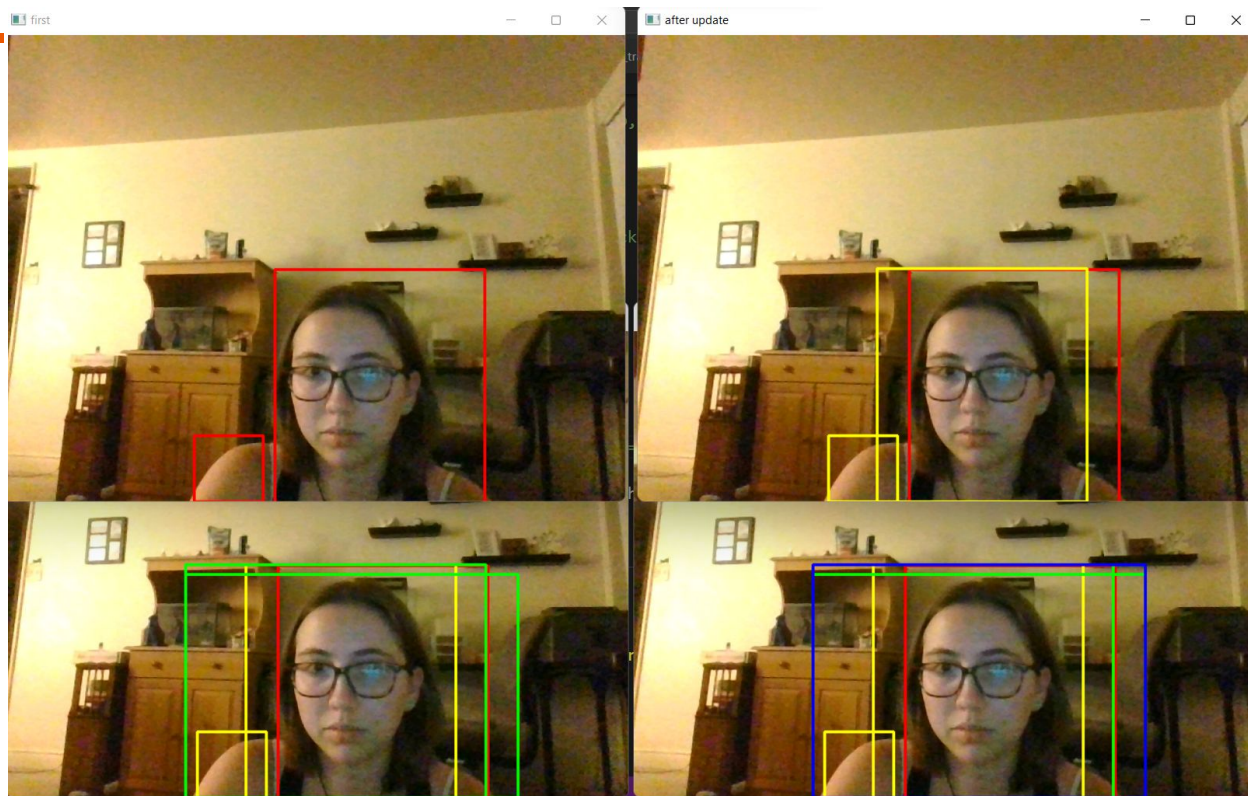
## Backend



# Complete Solution: Detection & Tracking

starting boxes  
from the last  
iteration (red)

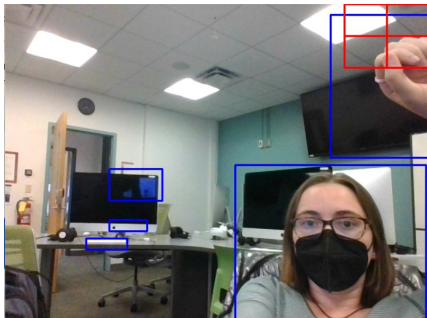
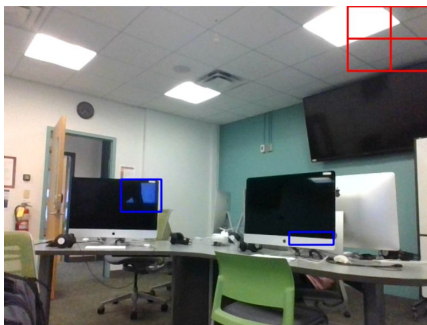
New BBoxes  
from detected  
movement(s)  
(green)



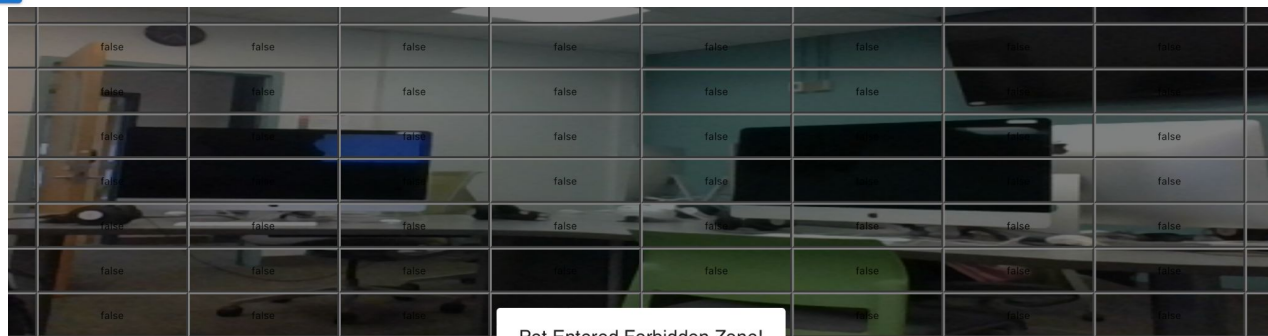
OpenCV  
tracker  
update  
(yellow)

Collapse  
overlapping  
boxes: final  
result(s) in blue

# Complete Solution (Forbidden Zone Demo)

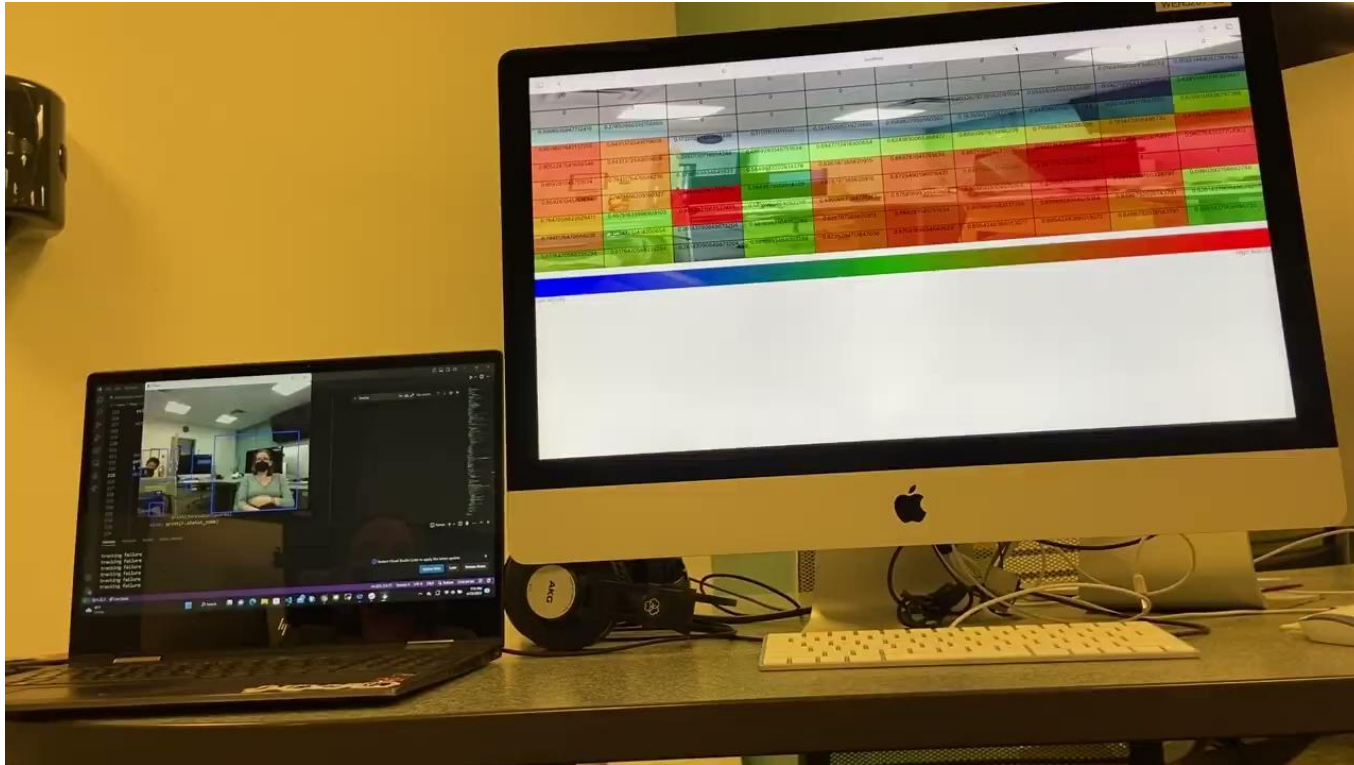


ADD





# Complete Solution (Heat Map Demo)





## Complete Solution (Other Features)

- Deployment
  - Frontend deployed through AWS S3 + Cloudfront
  - Backend deployed through AWS EC2 and Apache servers
- Live Video
  - User can request for live video feed of the room
  - CV -> Web App - images encoded into byte arrays
- Login using Google OAuth2.0
  - Used for authentication and security purposes



# Testing - Notification Speed



Test	Method	Goal	Result
Pet enter forbidden zone -> CV detection	Slow motion video (pet enters zone irl vs CV video feed)	< 1 second	0.625 seconds
Pet enter forbidden zone -> user notification	Slow motion video (CV video feed to notification)	< 10 second	1.125 seconds

- CV trade off is communication with the Web App - more frequent requests = more accurate data on the web app, but lowers the frame rate
- Web App limiting factor is polling rate (frequency of GET requests) from frontend to backend to get notification data (second result ~ polling speed)
  - Tested with GET requests / 1 second
  - Trade off is higher polling rate means faster notification to user but more server resources spent on requests

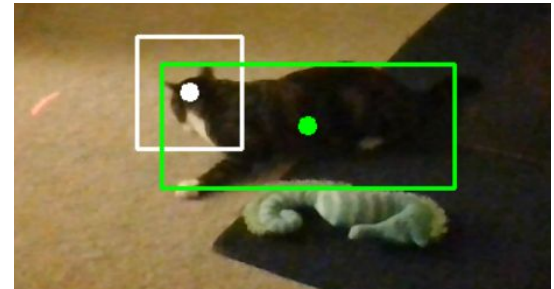
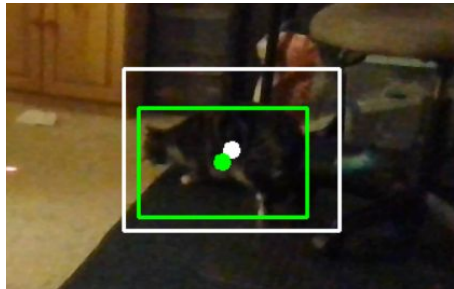
# Testing - Tracking



Test	Method	Goal	Result
Accuracy of tracker location(s)	Compare CV generated bounding box to human-chosen bounding box; Display the difference bbox centers, and estimate the distance	Within 1 foot	Generally around 3-6 inches
Detection speed of new animals	Slow motion video - pet enters frame irl -> bounding box appears	>5 seconds	~0.75 seconds on average

White: CV code

Green: human chosen





# Testing - User Accessibility



Design Requirements  
Use Case Requirements

Test	Method	Goal	Result
System Cost	Adding user costs	< \$100	Raspberry Pi + Raspberry Pi Camera = \$80
System Setup	User Testing (10 participants)	< 5 min for each participant	TBD
Accomplishment of Website Tasks	User Testing (10 participants)	9 out of 10 users can do 4 core tasks successfully with minimal guidance	TBD

# Project Management

