Allsky Camera Network for Detecting Bolides

Tyler Turner
Vincent Quintero
Jean-Pierre Derbes
Charles Derbes
Dr. Csaba Palotai

Task Matrix (Milestone 1)

Task	Completed	Tyler	Vincent	Jean-Pierre	Charles
Diagnose current issues	75%	Fix stop camera error	Frontend, Improve image composites	Fix state	Fix unknown camera control errors
Create system architecture diagram	100%	All contribute	All contribute	All contribute	All contribute
Storyboard the frontend	100%	Brainstorm	Create	Brainstorm	Brainstorm
API design	100%	All contribute	All contribute	All contribute	All contribute
Design CLI	50%	Create	Brainstorm	Brainstorm	Brainstorm

Task Discussion

Diagnosing Current Issues - *75%*

The list of issues grows by the day, this is more of a continual task that involves patching and fixing bugs as they appear. Every time we fix something a new bug appears.

Fixed Issues:

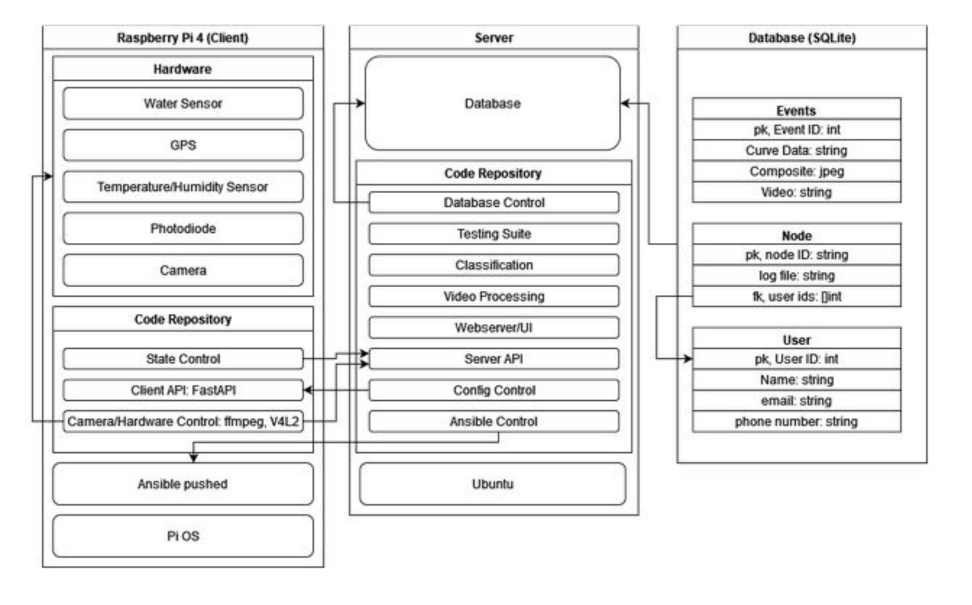
- Unknown camera control bug (incompatible config settings)
- Stop camera bug (camera controls were ignored)
- State control bug (invalid state lock on bad input)

Known Issues:

- “Morning chores” script fails
- Family of poll timeout related issues

Task Discussion

System Architecture Diagram - *100%*



Events
pk, Event ID: int
Curve Data: string
Composite: jpeg
Video: string

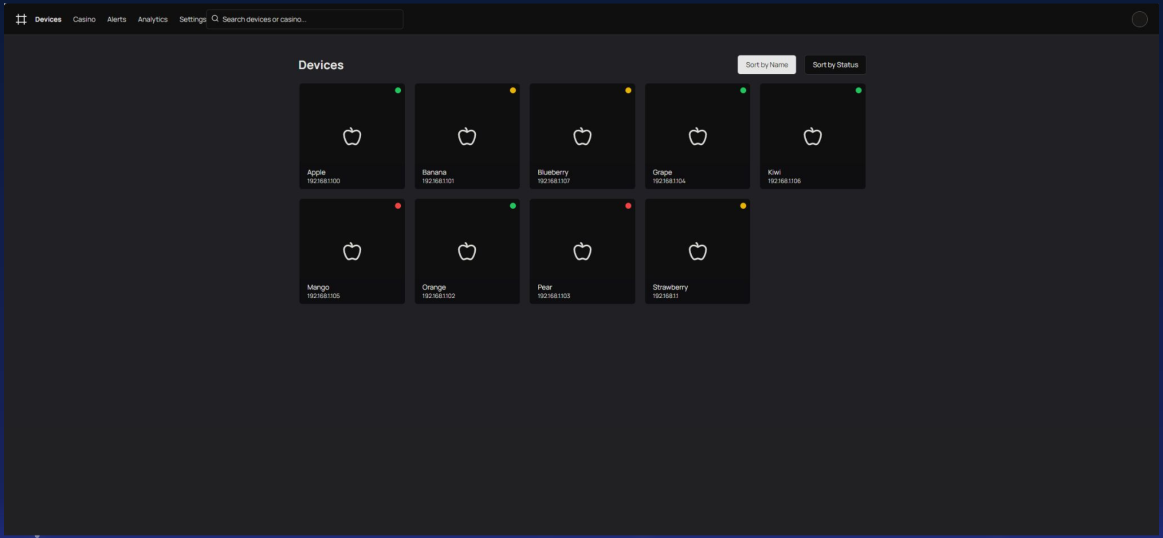
Node
pk, node ID: string
log file: string
fk, user ids: []int

User
pk, User ID: int
Name: string
email: string
phone number: string

Task Discussion

Storyboard the Frontend - *100%*

Successfully created mockup UI's



Devices

Apple
192.168.1.100

Banana
192.168.1.101

Blueberry
192.168.1.107

Grape
192.168.1.104

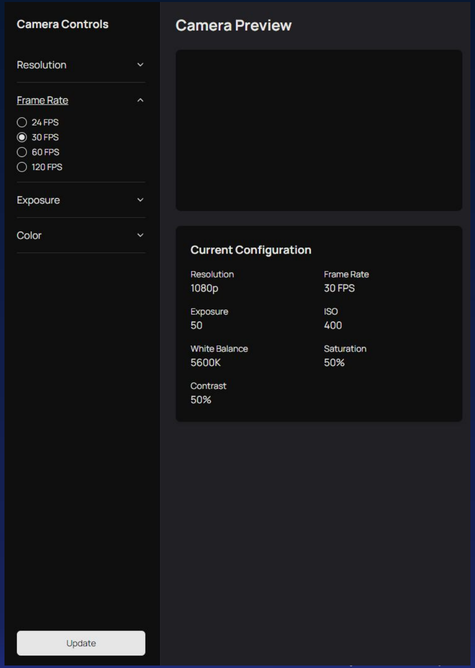
Kiwi
192.168.1.106

Mango
192.168.1.105

Orange
192.168.1.102

Pear
192.168.1.103

Strawberry
192.168.1.1



Camera Controls

Frame Rate

○ 24 FPS
● 30 FPS
○ 60 FPS
○ 120 FPS

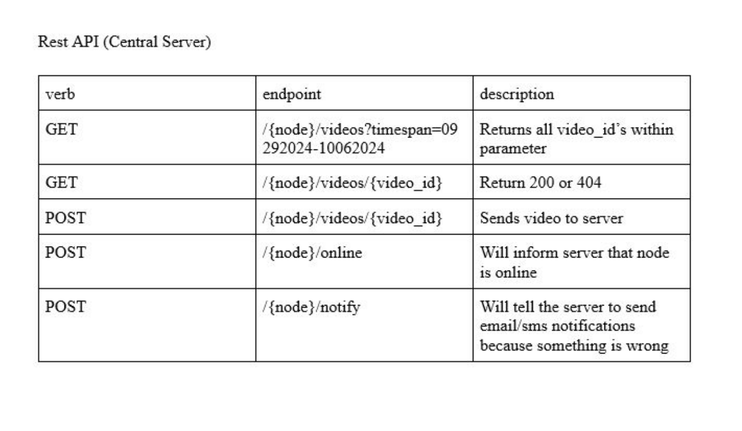
Camera Preview

Current Configuration

Resolution 1080p	Frame Rate 30 FPS
Exposure 50	ISO 400
White Balance 5600K	Saturation 50%
Contrast 50%	

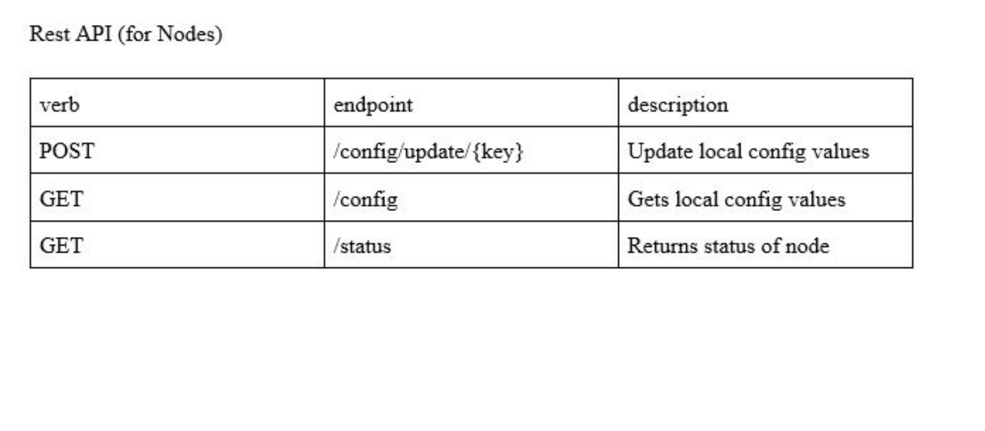
Task Discussion

API Design - *100%*



Rest API (Central Server)

verb	endpoint	description
GET	/{node}/videos?timespan=09292024-10062024	Returns all video_id's within parameter
GET	/{node}/videos/{video_id}	Return 200 or 404
POST	/{node}/videos/{video_id}	Sends video to server
POST	/{node}/online	Will inform server that node is online
POST	/{node}/notify	Will tell the server to send email/sms notifications because something is wrong



Rest API (for Nodes)

verb	endpoint	description
POST	/config/update/{key}	Update local config values
GET	/config	Gets local config values
GET	/status	Returns status of node

Task Discussion

Test Plan - *100%*

Design CLI - *50%*

The CLI aims to provide simplified control over the data processing pipeline, enabling researchers to perform more bespoke analysis on events as needed.

Implemented:

- Image composite generation

Upcoming:

- Starmaps

Contribution of Each Member

Tyler Turner

- Fixed bugs in current software
- Designed API
- Did most of the communication between the team and advisor/researchers

Vincent Quintero

- Input on API design, architecture design
- Assist on fixing C++ Sentinel communication bugs
- Create better bolide composite tooling

Contribution of Each Member

Jean-Pierre Derbes

- Fixed bugs in sentinel camera state
- Fixed bugs in communication between sentinel camera C++ and Python
- Provided input on API design for server and node database

Charles Derbes

- Fixed unknown camera control bug
- Provided input on API design
- Worked on ground up architecture redesign

Task Matrix (Milestone 2)

Task	Tyler	Vincent	Jean-Pierre	Charles
Continuously fix endless stream of issues	All contribute	All contribute	All contribute	All contribute
Add logs for easier diagnosis of issues	0%	90%	0%	10%
Replace current C++ camera code	All contribute	All contribute	All contribute	All contribute
Implement Server API	70%	5%	25%	0%
Implement Client API	0%	0%	30%	70%

Task Matrix (Milestone 2)

Task	Tyler	Vincent	Jean-Pierre	Charles
Begin writing CLI	33%	33%	33%	1%
IoT style setup	20%	0%	30%	50%

Task Discussion

Continuously fix endless stream of issues

- *Constant problem, certain boxes have certain issues and as these pop up we have to fix them.*

Add logs for easier diagnosis of issues

- *Project manager wants proper logs to be added so we know which buttons were clicked, in which order, and what the corresponding effect on the system was.*

Replace current C++ camera code

- *Currently, Cherrypy calls C++ functions that do the video processing. We would like to change this so that the the video processing is “done” through OpenCV on the server.*

Implement Server API

- *The server API handles video (GET video and POST video), node health, frontend, and user notifications.*

Task Discussion

Implement Client API

- *Implementing the client (hardware node) API using FastApi. The server will be able to get and update the client's local config values, and retrieve the client's status.*

Begin writing CLI

- *The CLI exposes internal functionality to researchers giving them more flexibility when it comes to their analysis.*

IoT Style Setup

- *Nodes will connect to the local network using an IoT style setup rather than having to tediously plug in a keyboard and monitor in order to connect.*

Faculty Advisor feedback

- Task 1: Dr. Palotai recognizes the improvements that we have made even though it is not finished.
- Task 2: Dr. Palotai approves of the system diagram.
- Task 3: Dr. Palotai believes that the proposed user interface will benefit the researchers in managing the nodes.
- Task 4: Dr. Palotai approves of the API design.
- Task 5: Dr. Palotai likes that the data processing will be easier with a command line interface.

Thanks!