Things that still need to be done/ things our group could not complete in time:

**Camera Trap System:**

* Combine router and master so that the master device can serve as the router for the network so we can have one less device in the network. This device needs to be a Raspberry Pi Zero W.
* Provide a clean, organized way to assign device numbers to each Pi, so far, we need to hardcode it into the code (assign it manually in the code, which is complicated and can be messy and disorganized)
* Provide a way to turn on the whole system without the use of the mobile app, perhaps through a Python or shell script
* Make the system headless, so that we do not need to log into the computer to start the code
* Make the camera trap system codes run as services, so they can be started and stopped cleanly.
* We did not test how long the system runs for, how much power consumption each device takes
* It might be possible to reduce latency between photos/devices by optimizing the code (multithreading, spawning another process), but we could not complete this
* Enclosures still need to be printed out, the one we provided initially is too small. We have updated the 3D model file by increasing the size of the enclosure so it can fit the small battery and the Raspberry Pi Zero W but it still needs to be set up
* Provide a way to change the master device with and without the mobile app. Make sure all devices have the master code and software.
* Provide a way to know that the code is running or not (perhaps with an LED?)
* Provide a proper way to label the devices so they do not mixed up/ cause confusion (put a sticker with a number of the device on the enclosure)
* We did not do testing with proper orientation of the devices/ proper settings (how to set them up so they always take a photo when something is sensed)
* Provide hostnames to all devices in the network through the DNS server
* Reinstall the RTC on the master Pi, and work on the time synchronization script
* Provide a way to make backups of the photos on the devices automatically (through a cron job?)
* We did not do testing of the latency between the photos, but it was mostly consistent (less than 100 ms), try to reduce this if possible
* Provide a way so that the system stops at night, but turns on in daytime – a bit challenging, but you would need electronics/circuits with some software knowledge but this would help a lot in terms of power consumption and system runtime.

**Mobile App/ Mobile Server:**

* Make the App more user friendly/ clean it up and make it look nice
* Provide a way to a clean way to delete photos from all Pis. (right now when you start the camera trap code, it will automatically delete all photos currently stored in the Pi)
* Provide a way to gracefully start the system
* Provide a way to gracefully stop the system
* Provide a way to view photos on the Pis before downloading them
* Have the mobile app source code start/run an FTP server by itself without the need/use of an external app