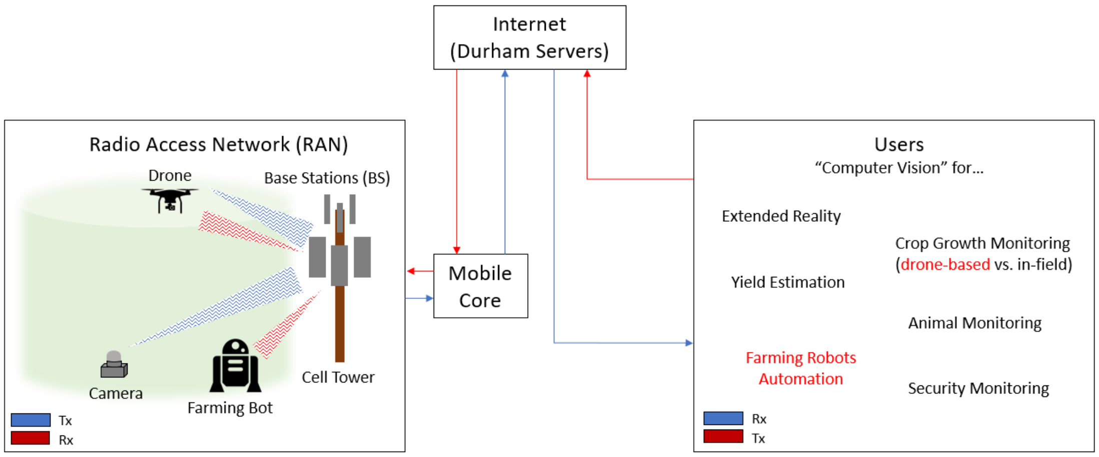
4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

4.1 Unit Testing – Sam/CK

What units are being tested? How? Tools?



Since we are using the ARA network, the infrastructure like the base station on the cell towers will likely already be configured to a certain modulation scheme to optimize performance of specific applications. If not, this is one area of the project we will need to test. Along with the transmission of the video feed from user equipment (UE) in the field, these units will be tested for the characteristics defined in our requirements by Dr Hongwei (latency, throughput, and delay jitter to name a few). We are currently in discussion with graduate students on how this will specifically be tested but we would imagine for the latency, testing will be like the echo and response used to ping.

If we decide to add automation, say we did not want to use edge computing and decided to make decisions "far away" from the device in automation. We will need to ensure low latency of transmitted and received commands that are not calculated and made locally. If we were using XR, we will need to ensure both successfully transmitted and received data to successfully display the video feed at a meaningful quality for the user.

4.2 Interface Testing - CK

Although we have not yet developed specifics for the front-end of our 5G application, we will at the very least have an app for users to be able to see the video feed virtually anywhere. This video feed will be used for "computer vision". Computer vision is the process of analyzing digital images/video for quantified information or decision making. We have not yet defined what specifically the video feed will be analyzed for yet. However, we have considered applications such as crop/livestock monitoring data and automation.

Interface testing will consist of accuracy of information we determine we want to provide to the user. If we test for livestock count, we want to make sure our interface recognizes the appropriate number of livestock on the interface via ML image recognition. Additionally, if our interface was intended to make programming automation for farming robots easier for users, we want to test the application to intended users for feedback. If our intent was to make in-field robots capable of being controlled remotely, we want to test latency on when the action was called and when it was performed. In this case, we may have a team member on-call in the field whereas the other team member would be remote calling the commands.

4.3 Integration Testing - Vibhu

What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?

Our project has 4 main aspects that must be interfaced together. These include the User and UI, The ARA wireless network, the system where data analysis is done, and the cameras and sensors that provide input. I'm not sure if it's self sufficient but the SRS ran code base contains some functionalities to test the integration of systems. With integration testing we want to make sure all parts of the application from computer vision to the end users UI must all properly respond to each other. One thing to consider is how the other parts of the system will react if a component fails to respond.

4.4 System Testing - Andrew

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

The end goal of our group is to have each component of design work cohesively together to form one overall design. In the end, that will mean connecting each component of our design together to verify that they function together. However, before that final test there will be a variety of tests in between. We will want to verify that each component in the design is working as desired individually, then we will look through the chain of the design to analyze what connections between components could be tested to verify they work together. If our design is of the form Camera → UE → Network → Computer → Processing → Output, then we will want to conduct testing between the different subsections. A logical breakdown be testing Camera → UE → Network → Computer with one test, then test Computer → Processing → Output with another. Unfortunately, some items are hard to test without connecting many pieces. The only way to verify if the UE is interacting correctly with the network is to analyze what comes through to the computer. It would be hard to isolate the testing further. Some testing will obviously be planned, while other testing will be done to troubleshoot as we face issues. During this testing we want to be checking for video quality, latency, and correct analysis of the video feed.

4.5 Regression Testing - Jake

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure do not break? Is it driven by requirements? Tools?

Since we are working with an ISU lab and ARA, we want to respect their resources and other current projects. To make sure we don't cause any damage to either group's lab equipment or infrastructure, we will make sure to have the proper training and approval needed to operate it.

For our own project, we want to have checks and documentation in place, so we are keeping track of changes within the project's hardware and software along with safe testing. To make sure the software doesn't lose functionality with new additions, we will use GitHub's tracked changes to monitor the edits that are made to the code. The software team will also add comments and documentation to the code so that new users can follow the code and know how it works.

To maintain our project's hardware, we will make sure that each member of the hardware team is made aware of the limitations of each component and create a list of possible issues or risks that each component could encounter. This will allow us to create a protocol when working with the hardware so that its safety and maintenance is always prioritized. Since we will be working with outdoor cameras, we will also need to research the camera's durability to the weather in case we need to construct additional protection for them. Overall, we just need to be mindful and careful with our equipment to minimize accidental damage.

4.6 Acceptance Testing - Caleb

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

Functionality requirements are quite easy to demonstrate being meant. The video connection and the data analysis can be shown quite easily. For example, showing the video feed and live data on a screen would more or less suffice by showing the functionality requirements. To really go into detail, we will be characterizing the performance of the systems to show the throughput, delay, delay jitter, and overall performance of the system. This will be how we show a high-level view of the functionality requirements.

The non-functional requirements will be more difficult to show. This will be the user interface and the front end of our system. How easy is it to navigate? How easy is it to understand? Would commercial farmers be capable of using this system to its fullest? These are some of the questions we will look to address once the functionality of our project is fulfilled. To test this, we will have many different people use our project and give feedback on the experience and how it could be improved.

4.7 Security Testing (if applicable)

N/A

4.8 Results

Not useful in our case, as we have not tested yet