Entity Class:

1. User: subclass Admin
2. Crop
3. Livestock
4. Drone

Boundary Class: (Anything that is inputted onto a screen or outputted/printed to a screen)

1. User Interface class (buttons, screen, etc.)
2. Video file list class (screen showcasing list of video recording logs)
3. Livestock counter class (screen showcasing list like a csv file displaying list of animal types and number in next column representing how many of that animal is on farm)
4. Drone data collection class (screen displaying inventory of equipment, like how many shovels, tractors, etc. screen displaying water levels, battery life of drone, etc.

Control Class:

1. Crop Management Class (deals with irrigate crops, map erosions, pollinate crops, plant seeds)
2. System Management Class (login the user/admin)
3. Drone Maintenance Class (check water system, equipment maintenance, check drone status)
4. Security management system (Admin sets a route for a drone to patrol/ delete footage/change route, record)

Team Responsibilities (dividing up Class diagrams):

**Arvine:**

Entity Class: user, Admin

Boundary Class: user interface

Control Class: system Management class

**Kile:**

Entity Class: crop

Boundary Class: Video file list class

Control Class: Crop Management Class

**Nathan:**

Entity Class: Livestock

Boundary Class: livestock counter class

Control Class: Crop Security management system

**Ayusha:**

Entity Class: Drone

Boundary Class: Drone data collection class

Control Class: Drone Maintenance Class

My thought process for the classes:

A user will interact with the user interface class and then login to system through the control class (System Management Class). Then they can interact with the 3 control classes to do what they want and depending on what control class they use which will connect them to a boundary class.

For example, User is logged in and they want to run diagnostics for their drone like check water system, and equipment inventory. Then this will take them to control class (Drone Maintenance Class) to do the actual computation of checking water levels, battery, and taking inventory of farm equipment then it will take user to boundary class (Drone data collection class) where it will display the data it collected onto the screen for user to see.

This is my thought process, if it is not right feel free to correct my errors but please share with entire group so we can all do the assignment properly.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tasks that need to be completed by Monday:

* Each member should finish the scenario for their use cases that they worked on from the last assignment.
* Finish the 3 class diagrams that are assigned from above.
* Do some research on what a state chart is, I am still a bit perplexed on what a state chart should be.

Monday’s Meeting Objectives:

* Connect all our class diagrams together
* Create a state chart