Meeting 2 Questions and Guide

Goal:

- Clarify functional and non functional requirements
- Data flow diagram (both cases)
- Clarify what a successful end project looks like
- Clarify what applications must be created, draw some lofis
- Gain an understanding of what we are integrating into (what do they already have? Are we doing a separate app?)
- Sign NDA
- Schedule/plan to meet with customers

Questions

- What is our project name?
    - FieldSense
    - CropIQ
    - Agritron
- What does the current in-combine app look like?
- What does their database look like?
- Must the farmer select their land? Would that be relevant information? Would it not be tracked during harvest?
- Where will our project be? Our private github in a complete app that we share?
- Is tech stack a part of requirements? (angular/backend)

—----------

Meeting Notes

October 19 5-7pm: meeting with farmers 20-30 minutes (general meet and greet, ask 2-3 questions)

Background information

- Infrared spectrometer (protein, moisture content, etc.) might not be available but must be incorporated.

Data

- GTA
    - Jojo made a dataflow diagram
        - Draw.io, on sharepoint
        - Most of it is still in planning

    - Tablet app accesses JSON file from Jetson Nano which contains data from infrared and computer vision models (sepearated by different keys). Tablet app is where farmers will upload to GCP. (we do not worry about)
    - Tablet will have to store on the cloud
        - Connectivity: Jetson updates through modem or tablet and sent to cloud
            - Modem: most common case for passing data (could use instead of faking data)
        - No connectivity: tablet connects to Jetson and must be brought back to wifi or a circuit modem.
        - Offline mode - entirely offline right now. Don't have ability to send to cloud rn
            - Upload or sync (button?)
            - Simulate a modem on JETSON (real time)
            - Send to cloud from modem
- Process
    - Coordinates are stored in Jetson
        - Only storing prediction with Geo location (minute by minute)
            - Subsciption to notification for location prediction
    - Can cache data (coordinates) locally instead of grabbing thousands of data points.
        - Interface has to have caching ability

Features

- Get data
    - Must periodically get coordinates to see a path of all combines.
        - If we are getting coordinates more often then we get the sample data then we will only get the latest coordinates rather than coordinates and prediction data.
- Farmer needs to input header size of combine (also depends where the GPS is placed on the combine)
- Automatic land titles based on location
    - Need to get from user where the excact geolocation thing on combine (off to right). Do this for now
    - Ask the farmers why they don't automate location already

Requirements

- Combine can only see their data/app, farmer main app sees all

Tools

- Mapbox

    - Fee for service must ve reasonable and must not take our data. Open remote had a bad license.
- Spring boot for Java backend
    - Get data from API backend
- Analysis on the cloud can be done by scheduling lambda functions
- Mobile kotin and swift on flutter using api backend where java backend is hosted.

MVP Planning

- Start:
    - Incorporate GPS data to approximate map
    - Tool to extend/draw map
    - Autopopulate map
    - Later for storage:
        - Need mechanism to subdivide crops
- MVP2: gradient
    - Quadrant field out and take average of data supplied.
    - For each grade, have a colour representing it (Red, Yellow, Green)
    - Break final map into quadrants and apply gradient
    - Add gradients between the data once field is filled out
- MVP3: gamification
    - Why: combine drivers get very bored and mess around all day. 80% of the time they are on their phones, keep them interacting.
    - Fun competition between combines (who has brought in the best quality)
    - Badges ex: you have brought in 17 football fields of high quality grain
        - Progress bar to the next badge?
    - Quirky fun things (we get creative)
    - Could measure by square feet and compare.
- MVP 4: Path analysis
    - Why: Ensure quality grain is kept high quality so the value is higher.
    - Analysis after harvest (assume same crops for next year)
    - Print optimal path of least distance travelled.
    - Must consider efficiency (how often do they unload)
    - Could optimize grain collection to be based on combines. Ex: combine 1 targets good grain, combine 2 targets grade 2 grain, grade 3 for the last combine.
        -
    - Analysis would run on the cloud. Instance running on the cloud.
        - Python to run an instance in the cloud on the server, store in DB, and then we can just grab it.
    - Hesitancy from farmers to use:
        - Must not reduce efficiency. Turns are bad because it slows things down. Straight lines are optimal because you are keeping constant speed.
    - Potential price increase may not be worth it if efficiency decreases (most farmers only care about getting it all done quickly)

- Might be different crop on same field
- Optimize path for good grain for next year
- Least distance and least repetition
- Each combine has different desired quality so 1 has best combine
- May increase time to harvest due to turns