

Harvest 2018 Project Notes

Overview

- Track yield collections
- Help farmers optimise harvest yields.
- Help prevent theft.
- Manage admin (not a priority though)

Login System

- Farmers log in with (Email or Google Sign In)
- Foreman must log in with phone numbers.
 - Are asked to select the farm they work for. If they don't work for any a message says they aren't in any farm. And must check with the farmer if his correctly added him in.

Yield Collector

- Tracks foreman location throughout a session. Can be viewed as a red line in the sessions view.
- Each time a "+" is pressed on a workers name, the date and location for that pickup is logged against that worker.

Information

- Primary purpose for information is for working with the rest of the system. We only require what is necessary for using the "Yield Collector", "Sessions" and "Stats"
- Required fields:
 - Farms:
 - name
 - Orchards:
 - name
 - assigned farm
 - Worker:
 - first name

- last name
 - Foreman:
 - first name
 - last name
 - phone number
- Workers and foremen are all in the same view. Hence they must search "foreman" and "workers" to only show foremen and workers respectively.
- Search on mobile only does a string contains over multiple fields
- On web a search query is broken into 'tokens' split by spaces. Each token is then searched separately over the multiple fields.

Sessions

- Shows each sessions raw data.
- Orchard colours are determined by hashing their assigned farm into a hue between 80 - 360 degrees. The saturation is determined by hashing the upper half of the orchards database id and the lower half of the id determines the brightness. Both are generate a value between 0.6 and 1.0.
- DSL query like language. Has same search functionality as Information search described above. But also supports 'functions'. Adding in a token like "average" will display the average bags collected per worker in the display list.
 - All Functions
 - average or avg
 - sum or total
 - stddev or stdev
 - count or countWorkers
 - countOrchards
 - range
 - best or max
 - worst or min
 - duration or length
 - mode or common
- Time filters. To limit the sessions displayed to only within a certain time frame.
 - last # day(s)
 - last # week(s)
 - last # month(s)
 - today
 - yesterday
 - this week
 - last week

- this month
- last month
- this year
- last year
- Property filters. To only show query that match against a specific property simply add that property name in the query. For example "foreman name peter" will only show foreman that have the name peter.
 - All properties:
 - assigned orchard
 - company
 - crop
 - crop
 - cultivar
 - details
 - email
 - farm
 - farm company
 - farm email
 - farm name
 - farm nearest town
 - farm phone number
 - farm province
 - foreman
 - foreman assigned orchard
 - foreman id
 - foreman name
 - foreman phone number
 - id
 - irrigation type
 - name
 - nearest town
 - orchard
 - orchard crop
 - orchard cultivar
 - orchard irrigation type
 - orchard name
 - phone number
 - province
 - type
 - worker
 - worker assigned orchard
 - worker id

- worker name
- worker phone number

Stats

- Graph properties
 - Comparison: select one of:
 - Farm
 - Worker
 - Orchard
 - Foreman
 - Selection: select entities based on what was chosen in 'Comparison'
 - Time Period: is the step which to show amount of bags collected
 - Hourly
 - Daily
 - Weekly
 - Monthly
 - Yearly
 - Time Interval: From when to whens data must be shown.
 - Today
 - Yesterday
 - This Week
 - Last Week
 - This Month
 - Last Month
 - This Year
 - Last Year
 - Between Exact Dates. User must specify start and end dates.
 - Accumulation: how bags collected can be accumulated together.
 - None: No grouping takes place.
 - By Entity: sums the amounts of each entity on a specific date together
 - By Time: sums the amounts of each date together for each entity
 - Show Expected Lines: should the expected lines be displayed
 - Show Average Lines: should the average line be displayed
 - Line Type: how should the lines be displayed
 - Linear
 - Curved
 - Stepped
- Legend colours: each entities colour is determined by hashing their database ids. Colours generated are within the range (hue: 80-360 degree, brightness: 0.6 - 1.0, saturation: 0.6 - 1.0)

- Uses sine graph $a * \sin(b * x + c) + d$ to predict yield since it can model seasons.
- Get basic sine prediction using:
- Optimise sine graph with genetic algorithm
 - Chromosome: 4 floats to represent a, b, c and d.
 - Population: 100 chromosomes
 - Initialisation: random pool of genes within between the basic sine graph prediction values.
 - Selection: maintains best 100 chromosomes
 - Fitness function: mean squared error function
 - Crossover operator: uniform crossover
 - Mutation: uniform
 - Termination condition: 25 generations (why not until a certain error? server time is expensive)