**Features/functions of electronic equipment tracking and maintenance system**

Equipment Data to Be Collected (imagine in spreadsheet form) – assume 1 row of data for each piece of equipment and each item listed below is a column describing the equipment

* Type of equipment **(equip)**
* Manufacturer **(equip)**
* Date of manufacture **(equip)**
* Serial number **(equip)**
* Date equipment became active at the facility **(not present)**
* Date equipment no longer viable **(not present)**
* Equipment location – facility name **(facility)**
* Equipment location – GPS coordinates **(facility – might have to geocode from address?)**
* Ownership type (Government, private owned, etc) **(facility)**
* Level of health facility (e.g Referral lab, province lab) **(facility)**
* Name of engineer performing most recent calibration **(not present)**
* Post of engineer performing most recent calibration **(equip)**
* Most recent calibration date **(not present)**
* Next calibration date **(not present)**
* Most recent maintenance date **(not present)**
* Name of engineer performing most recent maintenance **(not present)**
* Post of engineer performing most recent maintenance **(equip)**
* Next maintenance date **(not present)**
* Equipment retirement flag **(not present)**
* Retirement request date **(not present)**

Tool Functionality

* Map – Highlight facilities where equipment needs to be either calibrated or where maintenance needs to be performed in the next 10 days (yellow) or is past due (red)
  + May also want to consider a different color for equipment that should be removed
* Data input tables – Include inside the tool, the ability to filter to data for a single facility or just those that either require maintenance or calibration
  + Add the ability to change fields related to maintenance or calibration activities – engineer info or dates of activities
  + Ability to ‘Flag’ a piece of equipment as removed (or retirement if equipment no longer in use) – this may be preferable rather than allowing the engineer to remove the data from the tool.
* As new equipment is put in place, develop a process to add equipment to the tool.
  + Could do this using the tool – however it might be cumbersome to add each of the fields above using a web form
  + Could also develop a process to manually upload these data to the tool’s underlying data source if it would be easier to do it this way
* Added functionality to track service times and technician travel – **this may be quite difficult and out of the general scope of the other activities I have heard outlined.**

Primary Work Products (Travis)

* Tool outlined above
  + Simple interface allowing the viewer to see which facilities need attention, and update specific equipment information within the tool directly – will update underlying data source automatically
* Executive dashboard
  + Updated regularly
  + Allows the viewer to see which facilities require attention at a given time – due soon (yellow) past due (red)
  + Facility report showing which equipment needs attention
  + Report showing what equipment has been flagged
  + Summary report describing active equipment by status – due soon, past due
* Public facing Progress report
  + Map showing current state of equipment by facility
  + Statistics and charts describing equipment by facility and current state (due soon, past due, up to date) etc
* Final report and summary
  + Document describing how the tool was implemented and maintained and equipment summary

Necessary Software for the Deliverables Outlined Above

* Google Sheets – data storage (free)
  + The data above seems to be relatively simple, google sheets is a free, reliable, and secure way to store data
  + Also, data in google sheets is easily accessible using libraries in common analytical software such as R or Python
  + A google account could be setup as the host account to field questions regarding the tool and host the data for the project.
* R statistical software – development of above work products (free)
* Shinyapps.io account – deployment of tool and other work products (process scales with functionality)
  + The above work products can be easily deployed to shinyapps.io, a repository for hosting web applications developed in R Shiny
  + Shinyapps.io are hosted on secure servers by Rstudio (see <https://www.shinyapps.io/> for additional info)
  + Pricing – there are several price points for different license levels. There is a free version, but the apps would be accessible to anyone.
  + User authentication is available for the ‘Standard’ account ($1,100/year) – the administrative settings are straightforward. For each application, a user list in the form of email addresses can be specified. When users are added to the access list they will receive an email instructing them to authenticate via their google credentials, or if they do not have a google account, they may choose to setup a shinyapps.io account for the purpose of authentication.