1 Introduction and Use Cases

Your client, Hypothetical Power Testing (HPT), is a midsize electrical engineering company specializing in testing of industrial power backup systems, especially for telecommunications customers’ critical infrastructure. Using a fleet of test and measurement equipment ranging from mundane to custom-made, they validate every aspect of these power systems. They also conduct in-house validations of power equipment to ensure correct operation under both normal and exceptional conditions.

They currently use a mishmash of spreadsheets to track their inventory of test and measurement instruments, but this is cumbersome and error-prone, so they would like a unified system to replace this. Further, it is critical that they track the calibration of this equipment to ensure it is within manufacturer specifications and that they be able to provide documentation of this validation on demand. To this end, HPT has contracted with you to develop a new equipment inventory and calibration system.

This system will serve the following use cases:

- The system will track data about various models of test hardware, as well as instances of this hardware owned by the company (known as instruments).

- Users will be able to search for and view details of models and instruments.

- Users will be able to note calibrations events in the history of most instruments; this history will be viewable. Some instruments do not need calibration (e.g., laptops); the system will simply function as an inventory tracker for these.

- The system will be able to generate a formal calibration certificate for a given calibratable instrument.
2 Definitions

1. **Model**: A kind of measurement equipment, defined by fields:
   - Vendor: a short one-line string (e.g. “Fluke”), required.
   - Model number: a short one-line string (e.g. “87V”), required.
   - Short description: a short one-line string (e.g. “Multimeter with temperature probes”), required.
   - Comment: a multi-line string (e.g. “Retired model, no warranty, replace with Fluke 101A when EOL”), optional.
   - Calibration frequency: an integer number of days indicating how often the device should be calibrated. Optional, and the absence of this figure indicates an instrument that is non-calibratable (i.e., the system will do inventory tracking only; no calibration functions).

2. **Instrument**: A specific piece of owned measurement equipment, defined by fields:
   - Model: a reference to a model, as described above. Required.
   - Serial number: a short string, e.g. ”7474-505B”. Required.
   - Comment: a multi-line string (e.g. “Reserved for Palaemon project”). Optional.
   - Calibration history: Zero or more calibration events, each one being a tuple of the following properties. For calibratable instruments only.
     - Date: The date at which this event occurred. Required.
     - User: A reference to the user logging the event. Required.
     - Comment: A multi-line string (e.g. “Calibrated with the Fluke 5700 in the lab.”). Optional.

3. **Calibratable instrument**: An instrument requiring periodic calibration, such as a multimeter. In other words, an instrument with a defined calibration frequency.

4. **Non-calibratable instrument**: An instrument not requiring periodic calibration, such as a laptop or toolbox.

5. **Calibration expiration date**: A date computed for an instrument as the most recent calibration date plus the number of days specified by its model’s calibration frequency. Undefined if an instrument has no listed calibration or is non-calibratable.

6. **In calibration**: An instrument is said to be in calibration if it has a calibration expiration date that is on or after today. Naturally, the opposite is out of calibration.
7. **Report**: A general term used in database-driven systems; refers to a well-formatted result of a query, usually summarizing a large amount of data at once.

8. **Calibration certificate**: A specific printable report for a calibratable instrument that attests as to its calibration. Usually includes technical details.

9. **Unique**: Requirements may describe a given field (or combination of fields) as unique. This means that there may be at most one record with that value (or combination of values). Attempts to violate uniqueness should generate an error.

## 3 Requirements

A note on requirements: No set of requirements is perfect, and that is certainly true here. I’m sure that contradictions, under-specified behavior, and unintended consequences will be revealed. Your overriding goal should be to produce a quality system; if you believe that goal would be better served if a requirement were altered or interpreted a certain way, ask about it, and get the conclusion in writing. The result may be a variance in a requirement for a specific team, or even modification of this requirements document for all teams. In short, if unsure, ask.

Some requirements have attached an informal tip, motivation, or example; these do not alter the requirements themselves, but are meant to answer likely questions about a requirement.

### 1. Server

1.1. Your software must have a server that supports an arbitrary number of users. A web-based solution is preferred; thick client or mobile options are available with instructor pre-approval only.

1.2. During the install/setup process, a special user named “admin” is configured.

1.3. Users must have their accounts created by the admin user before being able to use the system. Users are defined by username (e.g. “tbletsch”), display name (e.g. “Tyler Bletsch”), email address (e.g. “Tyler.Bletsch@duke.edu”), and password.

1.4. A user accessing the system prior to logging in should be able to access nothing but a login prompt. Login is via username and password.

1.5. Any stored passwords must be kept in a secure manner (i.e., salted + hashed at minimum)

1.6. All communication between the clients and server must be encrypted.

   *Tip: For web-based solutions, this means using HTTPS.*

1.7. The server must maintain state in a persistent fashion.

   *Tip: For web-based solutions, this just means using a database or similar.*

1.8. For all views which show a potentially unbounded number of records, the response time of the interface shall not depend on the quantity of records unless a full listing is explicitly requested by the user.

   *Tip: This implies some form of pagination so that only a finite number of records are retrieved at a time. Pagination can be explicit (page 1 of N) or implicit (infinite scrolling with auto-loading). The latter part of the requirement (“unless a full listing is requested”) implies a “show all” button or similar. Other UI solutions are likely also possible.*
1.9. A user input is said to be assisted if it is a user-selected reference to an existing record (model, instrument, etc.) where the UI provides a listing, inline search, autocomplete, and/or other means to allow easy and efficient selection. Unless otherwise specified in this document, all selections of an existing record should be assisted.

2. Instrument data management

2.1. Model management: The administrator shall be able to add, modify, and remove models within the system; users shall be able to review them.

2.1.1. Models are to be created and modified with the fields described in definition 1, subject to these additional constraints:

2.1.1.1. Vendor input should be assisted based on past-seen vendors, such as with an auto-complete system.

Motivation: This is to avoid drift in how a given vendor is input, e.g. “Rohde & Schwarz”, “Rodhe & Schwartz”, “rodhe swartz”, etc.

2.1.1.2. The combination of vendor and model number must be unique.

2.1.2. The administrator seeking to remove a model shall only be able to do so if no instruments of the model exist. Even in this case, a confirmation dialog shall be shown.

2.1.3. Users may view a table of stored models showing all the short-form fields (vendor, model number, short description, calibration frequency). Non-calibratable models should be distinguished, e.g. by color or via a symbol in the calibration frequency field.

2.1.3.1. The view should be sortable by vendor, model number, short description, and calibration frequency.

2.1.3.2. It should be possible to filter this view by keyword search on the fields of vendor, model number, and short description.

2.1.3.3. Users should be able to navigate from this to a detail view (see req 2.1.4).

2.1.4. Users may view a detail view of a model showing all fields. This view should also list the instances that exist of this model (shown by serial number) and allow navigation to an instance’s detail view (req 2.2.4).

2.2. Instrument management: The administrator shall be able to add, modify, and remove instruments within the system; users shall be able to review them.

2.2.1. Instruments are to be created and modified with the fields described in definition 2, subject to these additional constraints:

2.2.1.1. Model selection should be very assisted, with both autocomplete and picklist.

2.2.1.2. The combination of model and serial number must be unique.

2.2.2. The administrator seeking to remove an instrument will be prompted with a confirmation dialog first.

2.2.3. Users may view a table of stored instruments showing the model-derived fields vendor, model number, and short description as well as the instrument’s serial number, most recent calibration date, and calibration expiration date.

2.2.3.1. The view should be sortable by all shown fields.
2.2.3.2. It should be possible to filter this view by keyword search on the fields of vendor, model number, short description, serial number.

2.2.3.3. Some form of visual indication should be used to distinguish instruments that are non-calibratable, in calibration, out of calibration, or have calibration expiring in the next 30 days.

2.2.3.4. Users should be able to navigate from this to a detail view of an instrument (see req 2.2.4).

2.2.3.5. From this view, it should be possible for the user to generate a formal calibration certificate (req 4.1) for a particular instrument.

2.2.4. Users may view a detail view of an instrument showing all fields.

2.2.4.1. This view should show the full calibration history of the instrument in chronological order (newest first). This history should show all recorded data associated with each calibration event.

2.2.4.2. This view should allow navigation to its model’s detail view (req 2.1.4).

2.2.4.3. If the instrument is calibratable, it should be possible from here for a user to record a new calibration event (req 2.3) or to generate a formal calibration certificate (req 4.1) for this instrument.

2.3. Instrument calibration: The user may record a new event to be added to the calibration history for any calibratable instrument (definition 2 describes the fields of a calibration event).

2.3.1. The user will input a calibration comment.

2.3.2. The event date will default to today’s date, but may be back-dated to an earlier date, but not forward-dated to a future date.

2.3.3. The user field of the calibration event will be recorded as the active user.

3. Bulk import/export facility

3.1. The administrator shall be able to import new models and/or instruments into the system by means of a format compatible with modern spreadsheet software (CSV, XLSX, or similar). **HPT is accepting proposals on the format.**

3.2. The import interface shall include documentation as to the import format.

3.3. For importing new instruments, the format should be able to specify a date and description for one calibration event; the event will be ascribed to the user doing the import. **Motivation:** *When HPT implements this system in production, they’ll adapt their current records to the import format and record just the most recent calibration, probably with a description like “Initial import”.

3.4. The import action shall only occur if the entire input is free of name conflicts or otherwise problematic issues; if such issues arise, the precise nature of the error should be presented to the user in enough detail that it can be corrected.

3.5. After a successful import, a count and list of records that were added should be provided.

3.6. The system shall be able to export any of the above data in a format compatible with import. The specific records exported should be filterable in the same ways as the table views for models (req 2.1.3) and instruments (req 2.2.3).
3.7. On export, the calibration fields in the format should reflect the most recent calibration event.

4. Reporting

4.1. **Calibration certificate**: The user should be able to generate a calibration certificate (definition 8). It must be cleanly printable so it can be presented to third parties. It must include all of the following.

4.1.1. Basic instrument info: vendor, model number, model short description, serial number.

4.1.2. The most recent date of calibration and the calibration expiration date.

4.1.3. Metadata surrounding this calibration event (user, comment).

5. Documentation

5.1. **Developer guide**: A document shall be provided which orients a new developer to how your system is constructed at a high level, what technologies are in use, how to configure a development/build environment, and how the database schema (or equivalent) is laid out.

5.2. **Deployment guide**: A document shall be provided which describes how to install your software entirely from scratch. It should start by describing the platform prerequisites (e.g. Linux distro, required packages, etc.), then mechanically describe every step to deploying your system to production readiness.