ECE Inventory System: Evolution 1

Tyler Bletsch, Duke University

January 13, 2017

1 Introduction and Use Cases

The ECE department would like an inventory system to track items ranging from $0.05 transistors to $5k oscilloscopes. This system will serve the following use cases:

- ECE lab managers will record the quantity, location, model numbers, vendor info, and documentation for varied lab assets.
- Duke students and faculty will be able to browse this available inventory information.
- Duke students and faculty will be able to request items from the inventory; managers will be able to approve or deny these requests, and the system will log approved disbursements as they reduce the inventory.

2 Defintions

- **Item**: A type of entity tracked in the inventory system, such as "breadboard". Sometimes, to make clear that this term refers to a category of thing rather than an instance, the phrase *distinct item* may be used.
- **Instance**: An specific instance of an item, such as one specific breadboard. To clarify, if the system has five breadboards, that's one item with five instances.
- **Tag**: A small piece of text used to flexibly describe an item, e.g. "resistor", "microcontroller", "oscilloscope", etc. An item can have zero or more tags.
- **Transaction**: Any operation which alters the state of the inventory, including acquisitions, disbursements, and administrative corrections.
- **Disbursement**: The act of outright giving instances of an item to a user, thus removing them from the system.
- **Available pool**: The quantity of an item on hand.
- **Outstanding request**: A user request that has not yet been acted upon by an administrator.
- **Approved request**: A user request that has been approved.
- **Denied request**: A user request that has been denied.

3 Requirements

1. Server

   1.1. Your software must have a server that supports an arbitrary number of users.

   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.

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

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

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

2. Basic inventory tracking functionality

   2.1. The admin user shall be able to create new distinct items. New item records include a unique name (string, required), starting quantity (integer, required), a model number (string), description (multi-line string), location (string), and zero or more *tags*.

   2.2. The admin user shall be able to modify the name, model number, description, location, and tags of items.

   2.3. The admin user shall be able to create and delete tags.

   2.4. The admin user shall be able to log the acquisition of additional instances of existing items.

   2.5. The admin user shall be able to log the loss/destruction of instances of existing items.

   2.6. The admin user shall be able to modify items in any way (including directly editing quantities), as well as delete items. These operations are to correct errors and override system rules; confirmation should be requested before such operations are carried out.

   2.7. A user shall be able to view the list of items.

      2.7.1. Users shall be able to filter the view with a set of required and/or excluded tags. The system will make apparent the selection of tags available.

      2.7.2. Users shall be able to search the view by name.

      2.7.3. The response time of the interface shall not depend on the quantity of matching items.

   2.8. Users shall be able to access a detailed view of a given item.

      2.8.1. All attributes of the item shall be shown.

      2.8.2. Users shall be able to see the quantity of instances available as well as any outstanding requests for the item made by the user.

      2.8.3. The admin user shall additionally be able to see *all* of the above, not just those relevant to the active user.

   2.9. Users shall be able to request a given quantity of a resource; each request must include text response with the reason for the request. Requests do NOT remove resources from the available pool; requests can therefore be "oversubscribed" (i.e. the total quantity requested exceeds available inventory).

2.10. A user shall be able to view a listing of requests.

2.10.1. Non-admin users should only see their own requests, while the admin user should see requests of all users. The requestor's reason input should be displayed.

2.10.2. Requests should be distinguished visually depending on if they're outstanding, approved, or denied; outstanding requests should be especially easy to find.

2.10.3. The admin user should be able to either approve an outstanding request for disbursement (thus removing the item instances from inventory) or deny it. A comment may be provided at this time.

2.10.4. Users shall be able to cancel outstanding requests, thus removing them outright.

2.10.5. Users shall be able to see if requests were approved or denied, and the associated comment (if any).

3. Documentation

3.1. **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.

3.2. **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.