Introduction

TUchem is an automated chemical inventory database system design for use in academic environments. TUchem can be used to track inventory for locations, provide up-to-date inventory information on-demand, and automatically generate environmental and other reports from the inventory. TUchem can also provide other information in a few clicks, such as chemical hazards, MSDS's, and inventory listings for emergencies such as fires. TUchem has many unique advantages as a system, such as inventory import, report generation, location management, and a user-friendly look-and-feel.

The motivation to build TUchem derived from the inefficiencies of the current system, where all inventory information was logged and processed by hand, requiring significant time and resources. By centralizing and automating much of this work, TUchem can provide cost savings and prevent many human errors.

Glossary of Terms

EPCRA report
Emergency Planning and Community Right-to-Know Act
EPA website: http://yosemite.epa.gov/oswer/ceppoweb.nsf/content/tier2.htm

CAA
Clean Air Act

CAS
Chemical Abstracts Service
CAS registry number
Unique numerical identifiers for chemical compounds, polymers, biological sequences, mixtures and alloys
http://en.wikipedia.org/wiki/CAS_registry_number

CERCLA
Comprehensive Environmental Response, Compensation, and Liability Act

DHS
Department of Homeland Security

EHS
Extremely Hazardous Substances

MSDS
Material Safety Data Sheet
Includes chemical hazard data, clean-up, handling procedures, and manufacturer information

NFPA
National Fire Protection Association

NFPA 704
The "fire diamond"
http://en.wikipedia.org/wiki/NFPA_704

RCRA
Resource Conservation and Recovery Act

RQ
Reportable Quantity

TPQ
Threshold Planning Quantity

TQ
Threshold Quantity

Control Point
An area in which chemicals are stored. Typically defined by a room in a building and campus.

Material Database
A complete database of various chemicals allowed to be in a particular inventory.

Model-View-Control Framework (MVC Framework)
A structural design of a web application that enables a programmer to separate business logic from the user interface displayed to the user.

Controller -> Action
A controller is an object-oriented class that is instantiated when a web browser wants to access a page. Each controller may have multiple functions, each of which are responsible for displaying entirely separate pages. These pages can be accessed by the following format: http://devel.tuchem.knuunk.net/client/controller/action

Development Tools

Apache

For a web server, we used Apache. Apache is a widely-used, standard, open-source server software package. It was the choice of previous groups, and we continued to use it for its
simplicity of configuration, its performance, and its widespread use (and thus widespread support).

**MySQL**

MySQL was selected as the database server. MySQL is also a widely-used, standard, open-source server software package. The MySQL website provides significant support, and with MySQL’s acquisition by Sun Microsystems, MySQL will continue to offer support and upgrades into the foreseeable future. Another nice benefit of MySQL is its integration with PHP, our server scripting language, offering easy and fast deployment of database interactivity.

**PHP**

We chose PHP as our server scripting language for its support, its flexibility, and for its power as a language. PHP, like Apache and MySQL, is also open-source software. Using PHP, we created a model-view-controller framework for the client, with XSL as our view generator.

When we first started the project, we had considered using a different scripting language, namely JSP, however we decided against this because of the complexity of adding JSP functionality to the server, and because of the greater flexibility of PHP.

We had also considered using two PHP frameworks to simplify our efforts, Drupal and CakePHP. We decided against both of these because of the additional educational requirements using either of these would impose, both on ourselves and on future programmers..

**XML**

One of the goals of this semester’s project was to make TUchem easily extensible. XML markup of database information fulfilled this role perfectly. XML allows for the creating of many different interfaces to the same data. This extensibility also facilitates creation of reports without needing to handle large query data directly.

**XSL**

XSL was chosen to transform XML into HTML for viewing in the browser. This was the easiest, most logical choice for transforming XML since this language was specifically designed to do exactly that.

**CSS**

CSS is the de facto standard for applying styles to web pages. In this case, it also allows TUchem to be rebranded quickly and without extensive changes to the XSL itself.

**JavaScript and AJAX**

JavaScript is used significantly in TUchem for features such as log-in obfuscation, dynamic
locations, and adding multiple CAS's to a chemical. AJAX is used lightly, mainly for cosmetic
uses, such as viewing chemical information on-demand for inventory items. These two are
virtually the only standard method to achieve such effects while minimizing bandwidth
requirements.

**Technical Matters**

**MVC Framework**

Pages can be accessed by web browsers in the form of a class/function combination that exists
within the "controllers" folder. For example, http://devel.tuchem.knuunk.net/client/main/index will
instantiate the "main" class and call its "index" function to display a page. Program logic is
contained within each function, at the end of which a "template" class takes the information
generated by the controller and displays it to the user.

Apache reads ".htacess" files to determine who has access to each directory in question. We
placed these files in each folder to disable external computers from directly accessing their
contents. Instead, all web page requests are routed through a single PHP file entitled
"index.php", which is located in the "client" folder in web root directory ("/home/tuchem/www")
on the server. This file then instantiates the proper controller/action combination.

The files associated with the MVC Framework are the following:

- www/client/includes/startup.php ? Defines, Constants
- www/client/classes/template.php ? Displays the generated content to the user
- www/client/classes/controller_base.php ? An OOP super class for each controller
- www/client/classes/router.php ? Determines which controller/action to load from URL

**Separate Database Connection**

We created a separate database file, which handles all the database requests.

This design defines a clear separation between the database ("server") and the client ("logic &
user interface"). In this way, we have provided potential for the database and user interface to
be hosted on different servers. The database class has the following functions:

- authenticateUser(username, encrypted_password, nonce)
- getUserInfo(username)
- getAllUserInfo()
- addNewUser(username, password, permissionGroup, firstname, lastname, email)
- removeUser(username)
- getAllPermissions()
- getAllLocations()
- getLocation(controlPoint_key)
- add_log(user, action, message, log)
- getAllLogEntries()
• getAllChemicals()
• searchChemicals(searchString)
• isChemicalInInventory(key)
• addChemical(casArray, name, alt_name, vendor, catalog_num, msdsFileName, health_hazard, flammability, instability, special_hazards)
• updateChemical(chemical_key, casArray, name, alt_name, vendor, catalog_num, msdsFileName, health_hazard, flammability, instability, special_hazards, bReplaceMSDS, uploadedFiles)
• removeChemical(chemical_key)
• getAllInventory( page = false, $page_size = 20 )
• getEPCRAInventory(location_key = 0, location_type = 'none')
• getChemicalsOfInterestInventory(location_key = 0, location_type = 'none')
• getInventory(inventory_key)
• addInventory(chemical_key, cpoint, container_type, container_quantity, container_size, container_units)
• updateInventory(key, cpoint, container_type, container_quantity, container_size, container_units)
• ImportAddInventoryItem(entry, controlpoint)
• searchInventory(search, location_key = 0, location_type = 'none')

Each "get" function returns an XML document containing the requested information. "getAllChemicals()", for example, will return a document that looks like the one below:

```xml
<chemical_list>
  <chemical>
    <name>Water</name>
    <name_alternate>H20</name_alternate>
    <vendor>CIS Chemical Labs</vendor>
    <vendor_catalognumber>1024</vendor_catalognumber>
    <hazards>
      <health>0</health>
      <flammability>0</flammability>
      <instability>0</instability>
      <special>
        <w /></special>
    </hazards>
  </chemical>
  ...
</chemical_list>
```

In addition, this design allows us to safely prevent SQL Injection attacks by escaping the parameters in each function call of the database class, rather than handling it in each controller.

**Dynamic Scripts And AJAX**

JavaScript is used to dynamically create and display information to the user without reloading a
They have been tested for cross-browser support (Mozilla and IE7), although more extensive testing should be done.

- "Dynamic_cas.js": This script allows the web user to dynamically add and remove multiple input text fields in which they can enter CAS numbers that are associated to a particular chemical.
- "Dynamic_locations.js": On page load, this script fills four SELECT boxes with Campuses, Buildings, Rooms, and Control Points from the database. A change in Campus will have a cascade effect on its children, updating Buildings, Rooms, and Control Points with the proper information. Changing the Building has a similar effect on Rooms and Control Points. Finally, Rooms work in the same manner, only updating Control Points.
- "Dynamic_information.js": This script uses AJAX to dynamically load chemical information without having to reload or open a new page. This is accomplished by creating an absolute positioned "div" element, which contains the chemical listing, underneath the position at which the user clicked the mouse. This is extremely convenient while browsing through the inventory and the chemical's description is not readily available.

Database Result Objects

We created several classes that encapsulate the XML formatting of our database results.

Each class has setter methods that pertain to the specific class. The following is a list of each class and their respective methods and formats:

**Chemical**

- setKey(key) - The key that uniquely identifies the chemical in the database
- setName(name)
- setAlternateName(name)
- addCas(cas)
- setVendor(vendor)
- setCatalogNumber(catalog_number)
- setMSDS(msdsName)
- setHealthHazard(health)
- setFlammability(flammability)
- setInstability(instability)
- setSpecialHazards( hazards_special)

Sample XML Format

```
<chemical>
  <name>Water</name>
  <name_alternate>H2O</name_alternate>
  <vendor>CIS Chemical Labs</vendor>
  <vendor_catalognumber>1024</vendor_catalognumber>
  <hazards>
```


ChemicalList

- `addChemical(chemical)` - The Chemical object to be added to the list

Sample XML Format

```xml
<chemical_list>
  <chemical>
    <name>Water</name>
    <name_alternate>H2O</name_alternate>
    <vendor>CIS Chemical Labs</vendor>
    <vendor_catalognumber>1024</vendor_catalognumber>
    <hazards>
      <health>0</health>
      <flammability>0</flammability>
      <instability>0</instability>
      <special>
        <w /> <ox /> <rad />
      </special>
    </hazards>
  </chemical>
  ...  
  ...  
</chemical_list>
```

InventoryItem

- `setKey(key)` - The key that uniquely identifies the inventory in the database
- `setPageCount(page)`
- `setContainerCount(container_count)`
- `setContainerSize(container_size)`
- `setContainerUnits(container_units)`
- `setContainerType(container_type)`
- `setChemical(chemical)` - The Chemical object that contains the chemical information
- `setLocation(location)` - The Location object that contains the location information

Sample XML Format

```xml
<inventory_item>
```
<inventory>
  <inventory_item>
    <container_count>3</container_count>
    <container_size>500</container_size>
    <container_units>mL</container_units>
    <container_type>bottle</container_type>
    <chemical>
      <name>Water</name>
      <name_alternate>H2O</name_alternate>
      <vendor>CIS Chemical Labs</vendor>
      <vendor_catalognumber>1024</vendor_catalognumber>
      <hazards>
        <health>0</health>
        <flammability>0</flammability>
        <instability>0</instability>
        <special>
          <w /> <ox /> <rad />
        </special>
      </hazards>
    </chemical>
    <location>
      <campus key="1">Main</campus>
      <building key="10">Wachman</building>
      <room key="74">109</room>
      <controlPoint key="21">First Computer Row</controlPoint>
    </location>
  </inventory_item>
</inventory>

Inventory

- addInventoryItem(inventory_item) - The InventoryItem object that contains the inventory information

Sample XML Format
Using the Application

Common Elements

There are several common elements across the TUchem application. This section will provide an overview of these common features.

Icons

There are four icons used consistently across the web site. The magnifying glass 🕵️ represents "view" for whatever item it is associated with. The pencil 🖊 represents "edit" for whatever item it is associated with. The red X ❌ represents "remove" for whatever item it is associated with. The green plus sign + represents "add to ..." for whatever item it is associated with to another, related list.

Location Selector

The location selector, a set of four select boxes used to select locations for inventorying and filtering, is used in several places, including importing inventory, adding individual inventory items, filtering the inventory listing, and generating reports. It functions dynamically, updating lower location levels based on upper location selections, i.e. it only shows the buildings on the selected campus.

Hazards

For consistency and visibility, hazard information is highlighted with proper colors to match the NFPA coloring.

Loggin in

The login screen is fairly straightforward. A user enters their username and password in the respective fields to access the system. If the login fails, the information box showing the welcome message will change to display an error message. If the login is successful, the page redirects to the main page.
The login page makes use of a SHA1 JavaScript and a server-generated nonce to offer secure login. Because of this, JavaScript must be enabled to log into the application.

The Main Page

The login page provides a simple launchpad for the rest of the site. It displays the basic layout of the rest of the website, with a header on the top and the menu on the left. The menu will differ depending on user permissions. In the content area, a simple welcome message is displayed. The rest of the site can be accessed from the menu on the left.

Viewing Locations
Clicking "View Locations" in the menu will take the user to a page to view all locations. This page offers a simple display with no interactivity. It is accessible to all users for the purpose of easy reference.

**Viewing Users**
Clicking "View Users" in the menu will take the user to a page to view all users. This page offers a simple display with no interactivity. It is accessible by all users for quick reference and access to administrator contact information, if necessary.

**Viewing and Managing Your User Profile**
Clicking the "Manage My Account" link under the Management group in the menu will show the user a page to view and edit their own profile. In this page, a user can see their username and user group. Additionally, they can update their actual name information and contact email. Also, they can reset their password, which will be relayed to them at the contact email.

Viewing Chemicals

Simple View
The simple view of the master chemical database looks much like the above screenshot. The master chemical database contains all the chemicals that may be added to inventory, and all information related to each chemical, including hazard information and an MSDS, if these are available. The chemicals are listed in an informative table format, sorted by chemical name. First in a chemical entry are the icons to perform actions on each chemical. These will vary depending on user permissions. Users can view information on any chemical. Users with chemical management permissions may additionally edit and remove chemicals by clicking on the respective icons. Users with inventory management permissions can add chemicals to inventory, a process which will be explained later in the documentation. After the icons, the table displays the chemical name, CAS('s), vendor, hazard information if available, and a link to the MSDS file if available. The information in the chemical table is paged for performance. Links to other pages can be found after the chemical listing table. At the top of the screen is a search box, which allows the user to search based on CAS, chemical name, vendor, and vendor catalog number. Searches on CAS’s must be performed using only valid CAS numbers, with or
without separating dashes.

**Search View**

The search view of the master chemical database appears similar to the simple view of the master chemical database. There are a few primary differences, however. First, the listings are changed to reflect the search results, rather than listing all chemicals. Also, the search results page is not paginated, since most searches will not return extensive results. All other appearances and functions are preserved.

**Viewing Inventory**

**Simple View**
The simple view of the inventory database looks similar to the simple view of the master chemical database. It lists some of the information regarding the chemical, such as name and CASes, and inventory information, including location information and the total quantity of the chemical in inventory in the control point. There is also a link to the MSDS for quick access. The simple inventory view is paginated for convenience. At the top of the page is the search box, which allows both searching by CAS/keyword and filtering by location. To filter by location, the user needs to choose the location using the drop-down boxes (campus, building, room, or control point), and select the radio button next to the location filter level (campus, building, room, or control point).

### Search/Filtered View

<table>
<thead>
<tr>
<th>CAS</th>
<th>Name</th>
<th>Quantity</th>
<th>Location</th>
<th>Control Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-19-7</td>
<td>Acetic Acid</td>
<td>12.5 L</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>64-19-7</td>
<td>Acetic Acid</td>
<td>2 L</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>67-64-1</td>
<td>Acetone</td>
<td>24 L</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>9048-48-8</td>
<td>Alcohol</td>
<td>600 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>7727-54-0</td>
<td>Ammonium Persulfate</td>
<td>600 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>7790-20-2</td>
<td>Ammonium Sulfate</td>
<td>3000 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>10043-35-3</td>
<td>Aromatic Acid</td>
<td>30 kg</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>6104-52-2</td>
<td>Brilliant Blue R</td>
<td>600 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>124-65-2</td>
<td>Cadmium Acid</td>
<td>150 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>10025-04-8</td>
<td>Cadmium Chloride</td>
<td>5000 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>56-75-7</td>
<td>Chlorinated</td>
<td>1800 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>87-66-3</td>
<td>Chloroform</td>
<td>24 L</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>10510-54-9</td>
<td>Coastal Violet Acetate</td>
<td>50 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>1609-47-8</td>
<td>Diethyl Dicyanamide</td>
<td>330 mL</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>67-66-5</td>
<td>Dimethyl sulfide</td>
<td>530 mL</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>24390-14-5</td>
<td>Diphenylamine</td>
<td>56 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>67-42-5</td>
<td>EDTA</td>
<td>500 g</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>141-42-5</td>
<td>Ethanolamine</td>
<td>500 mL</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>141-78-6</td>
<td>Ethyl Acetate</td>
<td>2560 mL</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
<tr>
<td>64-17-5</td>
<td>Ethyl Alcohol</td>
<td>20 L</td>
<td>Center City</td>
<td>TUOC 504 Storage Cabinet</td>
</tr>
</tbody>
</table>
The search view of the inventory appears nearly identical to the simple view except that the results have been filtered by the search and/or filter settings and the results are not paginated.

**Managing Chemicals**

**Adding**

To add a chemical to the master chemical database, the user should click the "Add Chemical" link in the menu. This will bring up the add chemical screen. Here, the user can enter CAS, chemical name, and other pertinent information regarding chemicals. Multiple CASes can be added by clicking "Add another CAS" to reveal another CAS entry field. Up to 10 CASes can be added to a chemical. Required fields are marked with an asterisk. CASes must be valid to be entered into the database; invalid CASes will not be allowed in the database. Once all of the information has been entered, the user then hits "Submit" to enter the chemical to the database. If there are any problems with the data entered, the user will return to the add screen with messages indicating the problem fields. Otherwise, the user will return to the add screen with a message indicating that the chemical was successfully added.

**Editing**
To access the edit chemical screen, click the pencil icon in the view of the master chemical database. Searching can be helpful for finding chemicals quickly. The edit chemical screen looks identical to the add chemical screen, except that the fields are filled in with the relevant data.

Removing

To remove a chemical from the master chemical database, simply click the red X next to the chemical in the master chemical database. A confirm box will appear to confirm deletion. A chemical can only be deleted if there are no items of that chemical in inventory.

Managing Inventory

Adding

Adding inventory through the add inventory interface requires first selecting a chemical to add. A user must click the green plus sign next to the chemical in the master chemical database to select the chemical to add to inventory. The link will take the user to the add inventory page, with information about the selected chemical listed on top. Beneath this information is the form for inventory data entry. The inventory's location can be selected using the dynamic locations selector, and quantity and container information can then be entered. All fields are required.

Editing
The edit inventory form looks identical to the add inventory form, except that data will be automatically filled in. An inventory item can be selected for editing from the view inventory page.

**Removing**

Removing inventory is a similar process to removing anything else from the database. Simply navigate to the view inventory page and click the red X to remove the item. A confirm dialog, like that shown above, will appear to confirm deletion.

**Importing Inventory**

This page allows the user to add inventory to the inventory database. Additionally, it will automatically add to the master chemical database if the chemicals being imported are not in the database. In order to import inventory the user has to select the file to import, then select a
control point. Inventory items will be entered into the database if they have valid CAS number and units. If a chemical does not have a correct CAS number or correct units, it will be displayed so the user can correct the information.

The imported file must be a CSV-formatted file following the proper header conventions. To get a CSV file from an Excel file, Save as->File Type->CSV (Comma-separated), then save the file.

**Generating Reports**

To generate a report, the user has to have permission to view reports. To start generating a report, click the "Generate Report" link in the menu. Select location filtering, if any desired, and the report to be viewed. Click generate report to open the report in a new window.

**Emergency Report**

The emergency report is specifically for fire emergencies, though it may have other applications. The report displays inventory information in a simple alternating table (without the additional layout), sorted by flammability hazard, instability hazard, health hazard, and finally special hazards. This report provides this information for quick reference in an emergency to emergency personnel.

**EPCRA Report**
The EPCRA report is a report that must be filed with the government annually. It is based on a list of chemicals deemed hazardous that must be reported if in inventory. The report actually appears like the form would if it were being filled out on paper, though the generated version already has all relevant data entered. It simply would need printed, completed with additional information, and sent.

The EPCRA report is a legacy piece of the project, and does not display properly. Because of constraints, this matter still needs to be handled.

**Chemicals of Interest Report**

The chemicals of interest report lists the items in inventory that are on the Department of Homeland Security's Chemicals of Interest list. They are displayed in a standard alternating list without additional layout.

The list currently used for the chemicals of interest report is outdated and needs renewed, and grouping features and additional criteria need to be added as well.

**Managing Users**

**Viewing Users**

The user accounts can easily be managed through "Manage Users Page," The page lists all users accounts and provides information about each account. From there, the icons next to
each user can be used to manage those users. A user cannot delete themselves. To add a new user, simply click on either the top or bottom "Add new user" link to be taken to the add user page.

Adding Users

This page allows for easy creation of a new user account. In order to access the page, the user needs to have user management permission. To create a new user account, the administrator needs to enter a username, first name, last name, and email for the new user. When a new user account is created, an email will be sent to the new user with login information.

Editing Users

When the user goes to this page, the account information for the selected user will be displayed. The user has the option to reset the password for the selected user or to change his/her name and email address.

Removing Users

Removing a user is a similar process to removing anything else in the application. Simply click the red X to delete the user, and confirm.

Managing Permission Groups
Viewing Permission Groups

To view the permission groups, simply click on the "Manage Permission Groups" link in the menu. The list of permission groups will be displayed, along with the means to manage them.

- User Management: allows the user access all web pages related to user management. If the user has this permission he/she will have access to "Manage Users," "Add New User," "Edit User Info," and "Manage Locations" pages.
- Permissions Management: allows access to "Permission Group" page.
- View Chemicals: allows access to "View Chemicals" page.
- Add Chemicals: allows the user to add chemicals to the chemical database.
- Edit/Remove Chemicals: allows the user to add/remove chemicals to the chemical database.
- View Inventory: allows the user to view the "View Inventory" page.
- Add Inventory: allows the user to enter inventory in inventory database.
- Edit/Remove Inventory: allows the user to edit/remove inventory chemicals in the the inventory database.
- View Reports: allows the user to view reports like EPCRA.

Adding Permission Groups

This page has an easy to use interface by selecting/unselecting checkboxes and clicking submit the user can change the permissions for just one group or all of the existing groups.

Editing Permission Groups

A new permission group can be created by just entering a group name in the empty text box, selecting the permissions and clicking "Submit" to create the new group.

Removing Permission Groups

Like other parts of the page, removal is accomplished by clicking the red X and confirming.

Managing Locations

Viewing Locations
The user can easily view and manage locations by using this collection of pages, on for each location level (campus, building, room, control point). The opening page shows all created control points and where they are located. The links above the listing can be used to navigate between the different location levels.

### Adding Locations

A new location can be created by selecting higher location levels, then entering a control point name in the empty text box.

### Editing Locations

This page allows the user to edit location names directly in the view by simply changing the names and hitting "Submit."

### Removing Locations

Locations can be deleted by pressing the red X icon. Deletions of higher location levels cascade delete lower levels, i.e. deleting a building deletes all of the rooms in that building and all of the control points in those rooms. A location cannot be deleted until all inventory items have been removed from it.

### Project Future
What Did Not Get Done

The biggest piece of the project that did not get done is logging. However, significant progress was made towards completing this goal, and very little would need to be done to actually complete logging. Some logging and log viewing is currently installed.

Another feature not accomplished was the association between users/groups and locations. This feature would cause users to be limited only to certain locations for security purposes.

What Should Be Done

Security was a major feature that was outside of the scope of the project. While some small security measures were implemented, the major security factor that could be added would be end-to-end encryption of data through the use of an SSL certificate (https:// addressing).

What Could Be Done

Because the application was redesigned with extensibility in mind, many additional features can be added and created. In addition to adding more reports, additional client interfaces can be created for such applications as mobile phone browsing. Specialized applications can also be written to handle more complex tasks such as potentially a barcoding scheme.