

Commands

* (4)List all bases
* (4) What chemical should be used to dissolved a given metal
* (8)What element has a given atomic number or atomic mass
* (8) List all elements with an atomic mass in a given range
* (8) List all chemicals in a given range???
* (8) What elements are in a given in compound
* (8) What compounds is a given element in
* (8) What type of object is the chemical with a given name
* (5) Add new acid
* (5) Add new base
* (5) Add new compound
* (5) Add new metal
* (5) Add new element (name, atomic number, atomic mass)
  + Option to specify that it is a metal (optional selection of an acid that dissolves it
* (7) Modify the amount of a chemical in our inventory
* (6) Modify the atomic number of an element
* (4) Modify or add which acid dissolves an existing metal
* (5, 6) Add, modify, and delete operations for the relationship between a compound and the list of elements that it is made of
* (7) Generate a report which chemicals are currently low in our inventory (asks where text find output should go)

Story

* (4) The User should be able to see a list of all acids currently in the database. At the top of the panel, there should be buttons to switch to other tables of the database. The user should be able to perform the same actions on any of these tables
  + Display list of chemicals in a panel
  + Add tabs to the top of the panel for each table in the database
    - Tabs should display the relevant chemicals
      * [Element][Metal][Compound][Base][Acid]
* (5) The user should be able to see an add button at the bottom of a table window. When pressed, a window should pop up allowing the user to fill in the appropriate fields for the acid table
  + Create add chemical button
  + Add window that displays relevant empty fields to the current table for the user to fill in
  + Add a submit button to that window. The submit button should persist it to the database
  + Report for validating the information entered
* (6) The user should be able to click on an entry in the table, the entry should highlight and the user should be able to click the modify button. A window will pop up with the information already in that row. The user can change that information and save it.
  + Create a “Modify” button and pop open a new JFrame of modify options.
  + This JFrame should have JTextFields for each value the Chemical holds, and be automatically filled with the values the selected entry has. This is different for each different type of chemical/etc. This JFrame also has buttons to quit/cancel the modification, and a button to submit.
  + Once the user presses submit, it will close the modification window and attempt to update the database. We will test to make sure it actually updates the entry, if it does, it will
  + display a success message or failure message.
  + Generate a report of success or failure
  + After closing the final window, we should update the panel the user is on to ensure it is displaying the most up-to-date version of the database.
* (7) The user should be able to click a button on the chemical screen to generate a report of all chemicals that are running low. A window should popup displaying chemical names and the quantity currently in stock.
  + Create a report button and link the appropriate command
  + Generate a report of the chemicals that are low
  + Create a pop up box to display which chemicals are running low
* (8) The user should be able to pull up a filter. The filter should have options relevant to the table the user is currently looking at. The user should be able to submit the filter and have only entries relevant to the filter be displayed in the list.
  + If Element:
    - Filter by inventory, atomic mass, name, atomic number
  + If metal:
    - Filter by inventory, atomic mass, name, atomic number, dissolveBy, moles
  + If Compound
    - Filter by name, inventory, and elements it contains.
  + If base
    - Filter by name, inventory, solute
  + If acid
    - Filter by name inventory, solute
  + If ALL:
    - Filter by name, inventory