Dummy Project Design Document

1. Summary of Application

The application I will be creating is fundamentally a webpage containing a list of common acronyms used by LASP along with their associated description and affiliation. Aside from obtaining experience with coding, web-development, GUIs, and databases, the purpose of this application is to provide an easy way for people to look up commonly used acronyms by employees at LASP. Ideally, this application will be useful for incoming employees that are not yet familiar with the extensive amount of acronyms used at LASP. The app will allow the user to search through the table of acronyms for a specific acronym to find its meaning and its description, as well as add, edit, and delete acronyms from the database. From the webpage, the user can type in a search bar to search the database for a specific acronym or sort the acronym list alphabetically by either the acronym or the affiliation. Through the add acronym and edit acronym GUI’s, the user can add new acronyms to the database, or edit/delete existing acronyms from the database, which will update the display on the web page.

1. Meeting Requirements
   1. **Web**: The main display page of the application will be a web page. The page will have the list of acronyms displayed in a table format, with other information and descriptions included on the page. The webpage will have a component that allows the user to sort the list alphabetically by either the acronym or its affiliation. It will also allow the user to search through the list for a specific acronym.
   2. **GUI**: The main GUI part of my project will be the “Add Acronym” and “Edit/Delete Acronym” GUIs that can be ran from the terminal. The Add Acronym GUI will have the user fill out basic information about the acronym before adding it to the database. The Edit/Delete Acronym GUI will have the user search through existing acronyms in the database and either edit it’s corresponding information or delete its row of data entirely from the database. There will also be error and warning pop up GUIs that ensure the entries/updates are filled out correctly as well as ask the user if they are sure they’d like to make these changes to the database.
   3. **Non-GUI**: The non-GUI component will be the functionality of adding an acronym to the table and having all of the user inputs correspond to an entry in the database. For each entry change to the Add Acronym GUI, the inputs are checked and the tooltips are updated based on what still needs to be filled out. The Add Acronym button is only enabled when the form is filled out correctly. When the button is clicked, I will have the program search existing acronyms to check if the acronym already exists. If it exists, a warning will be displayed to the user and they can either enter it into the database anyway or discard the entry. When editing or deleting an acronym from the database, the user is required to search for the specific acronym in a search bar. If there are multiple acronyms with this same name, the results will appear as tabs and the user will be able to select the appropriate tab containing the acronym they wish to edit/delete.
   4. **Database**: The acronym list will be stored in a database along with what they stand for, are affiliated with (i.e. a specific mission, or LASP-general) and an optional description of what they are used for. The “Add Acronym” GUI will handle inputs into the database and the “Edit/Delete Acronym” GUI will handle edits to the database and deleting entries from it. Both GUI’s incorporate searching through the database and displaying data to the user.
2. High Level component interaction
   1. Feature set (core functionality):
      1. The application will have a webpage where the table of acronyms is displayed.
      2. The user can run the “Add Acronym” and “Edit/Delete Acronym” GUIs from the terminal to handle inputs and edits to the database.
      3. The “Add Acronym” GUI allows the user to input an acronym along with its details to the database.
      4. Before allowing the user to add the acronym, the “acronym,” “stands for,” and “affiliation” entries will be checked for completeness and display error message boxes if the user needs to make changes to their entry.
      5. If the acronym that the user is trying to add already exists, a warning will be displayed to the user that asks them if they would like to add the acronym anyway or discard the entry.
      6. The “Edit/Delete” GUI will allow the user to search for an acronym in the database and either edit the details or delete its “row” entirely from the database.
      7. There will be a “Sort by” option on the webpage where the user is able to sort the list of acronyms alphabetically by either the acronym itself or by its affiliation.
      8. There will be a searching component on the web page that allows the user to find and display the attributes of a specific acronym.
   2. Stretch Goals (bonus functionality):
      1. For the Edit Acronym GUI, if there are duplicate results for an acronym search, the results will display in tabs where the user can select the acronym they wish to edit.
      2. For searching through the list on the website, instead of entering an acronym and clicking a “Search” button to see if it exists, the website will automatically filter the table as the user is typing in the acronym to display all possible matches.
      3. To consolidate space on the webpage, instead of displaying the entire table on one webpage, there may be about 20 rows initially displayed along with tabs at the bottom of the table that take the user to the next page.
3. Basic design of key components

Feature set - Webpage:



￼Feature set - “Add Acronym” GUI:



\*Note: Custom Affiliation is only writable if the user chooses the “Custom” option from the Affiliation drop down menu.

Feature set - Duplicate Handling:



Feature set - Edit/Delete Acronym GUI:

