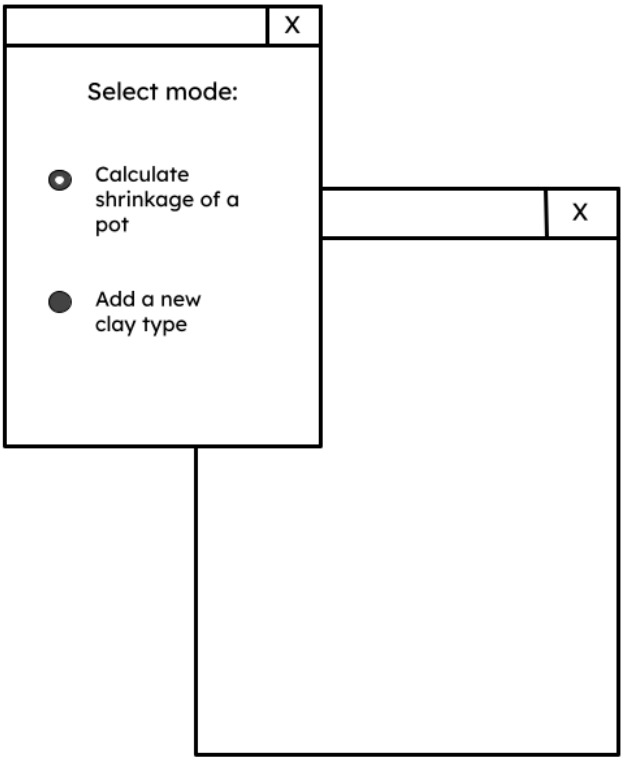
Criterion B: Design

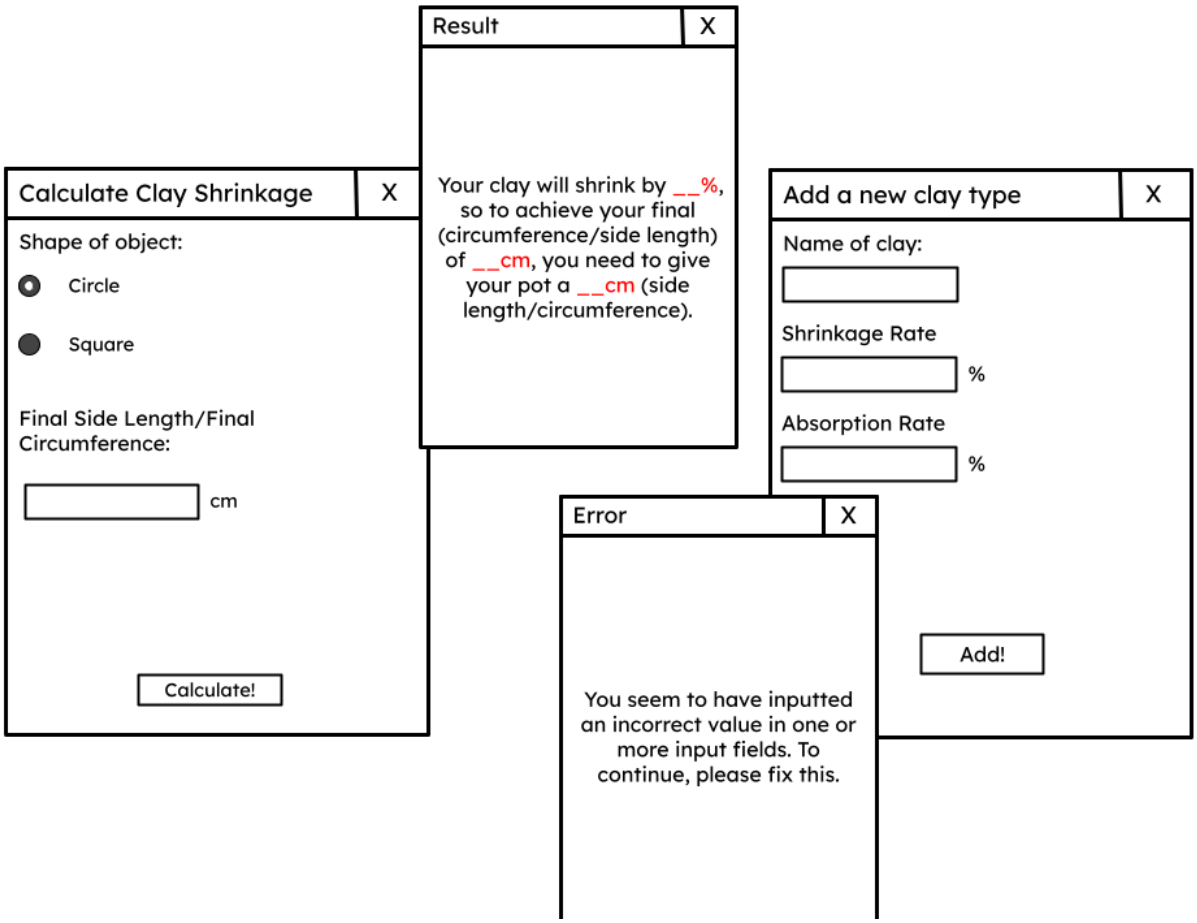
Design Overview

The program will consist of two standard parts: the frontend and the backend. The frontend will contain a TKinter-based GUI containing easy to use drop-down menus, text input fields, and lists. When the program is opened, the user will begin by choosing what they would like to do with the program.



Currently, the two planned options are adding a new clay type and calculating pot shrinkage; a visual of this menu is shown above. A dialogue box containing these two options will show up and upon confirmation from the user, the main window will be populated with the respective options of

their selected mode, shown below.



On the left is the window that appears when you choose to calculate the shrinkage of a pot, and includes appropriate fields. On the right is a similar window that shows up when you want to add a new clay. This looks similar to the other window save for the absence of predefined options and increase in text fields.On the bottom is a universal error window that appears whenever an incorrect value is entered into a text field, such as an alphabetic character. The top window is a result window, and it displays the results of the backend's calculations to the user. The values are highlighted in red in this design as a placeholder. However, they may also be highlighted in the final design to make them easier to spot.

On the backend, there will be a database containing 4 columns: the primary key, the name of the clay, the absorption rate, and the shrinkage rate. When the program is first used, the database will contain ten different types of clay. The user will of course be able to add more types of clay by choosing the addition mode when the program is launched. When the program is in calculation mode, they are able to choose between two different shapes: circle and square. The choice of shape dramatically affects how the calculations are done.

- Circle:
    - The entered dimension is circumference, due to it being the easiest to quantify. However, diameter is a more useful measurement for calculating shrinkage. The

program will first convert the circumference to diameter by dividing it by pi. It will then calculate how much the diameter shrinks by multiplying the diameter by (1.00 + (shrinkage rate / 100)). Finally, the diameter will be turned back into circumference and displayed to the user.

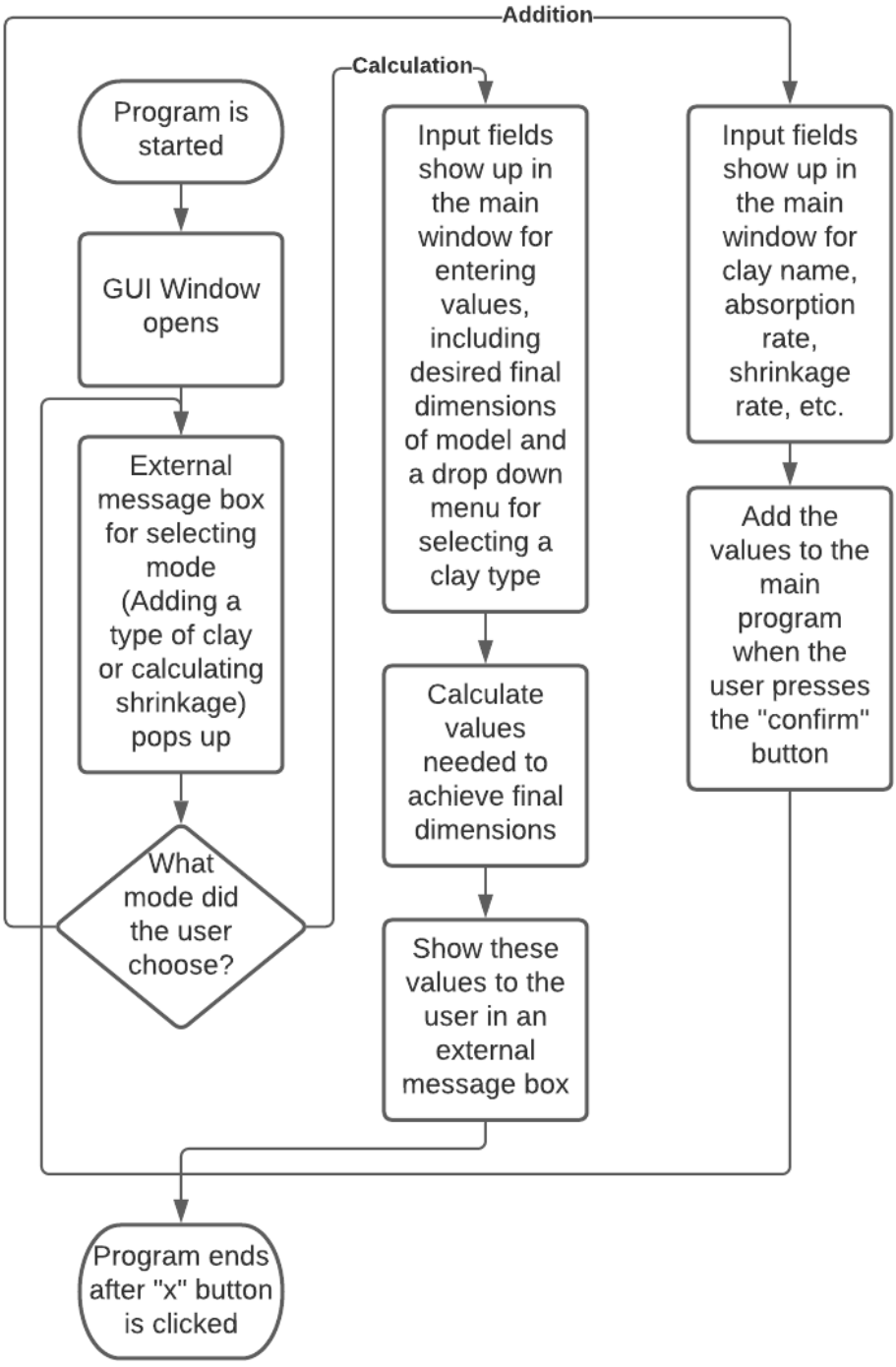
- Square
    - In this case, the entered dimension is side length. This is similarly converted to a diagonal measurement using the pythagorean theorem. This is again multiplied the same as above, converted back to its original form, and displayed to the user.

To simplify the development process, the frontend and the backend will both be their own files. This way, the calculations can be done in one file and returned, keeping a clear divide between the frontend and backend and allowing the code to be more organized.

Test Plan:

- Program will contain a functional GUI, allowing the user to input values.
    - Check that each menu works, and each value gets recorded and used appropriately.
        - Example: Load each menu one after another, input values into every box, make sure the data is retrieved, stored, and/or used properly (such as a proper shrinkage calculation).
- Program will contain a database of different clays, and allow the user to input new ones
    - Make sure adding and deleting clays works
        - Example: Adding a clay, and loading the .db file with SQLiteBrowser to make sure the data was added appropriately.
- Program will be intuitive, and warn users when they input a bad value (such as an alphabetic character where a numeric value should be)
    - Make sure each box validates user input, and doesn't try calculations while there are blank or invalid fields which could potentially crash the program
        - Example: User tries to calculate without selecting a clay, and the program outputs an error message rather than crashing or freezing

Flow Chart:



Word count: 664