

## 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 with simple menus and text fields. When the program is opened, the user will begin by choosing what they would like to do with the program.

The diagram shows a rectangular window with a title bar at the top. The title bar has a small 'X' button on the right. The main area of the window contains the text 'Select mode:' followed by three radio button options. The first option, 'Calculate shrinkage of a pot', has a selected radio button (a circle with a dot in the center). The second option, 'Add a new clay type', and the third option, 'Delete a clay type', have unselected radio buttons (empty circles).

Currently, the three planned options are displayed above. The main window will appear differently depending on what the user selects.

ResultX

Your clay will shrink by \_\_%, so to achieve your final (circumference/side length) of \_\_cm, you need to give your pot a \_\_cm (side length/circumference).

Calculate Clay

Shape of object:

☒ Circle

☐ Square

Final Side Length/Final Circumference:

 cm

Calculate!

Add a new clay typeX

Name of clay:

Shrinkage Rate

 %

Absorption Rate

 %

Add!

Delete a clay typeX

Name of clay:

 ↓

Delete

The left window is the pot size calculation menu, with an example pop-up window explaining the result with context. The center window is where new clay is added, and the right window contains a drop-down menu containing the clays in the program, that allows the user to delete from a pre-selected menu.

In the backend, there will be a database containing four important, self-explanatory values, shown below:

| clays   |
|---|
| <<pkey>> id: INTEGER<br>clay_name: STRING<br>shrink_rate: FLOAT<br>absorption_rate: FLOAT |

The program contains a user-modifiable database of common clays used by the client. When calculating, the client is able to choose between two different shapes: circle and square. Though the shape does not affect the math, I still thought it was

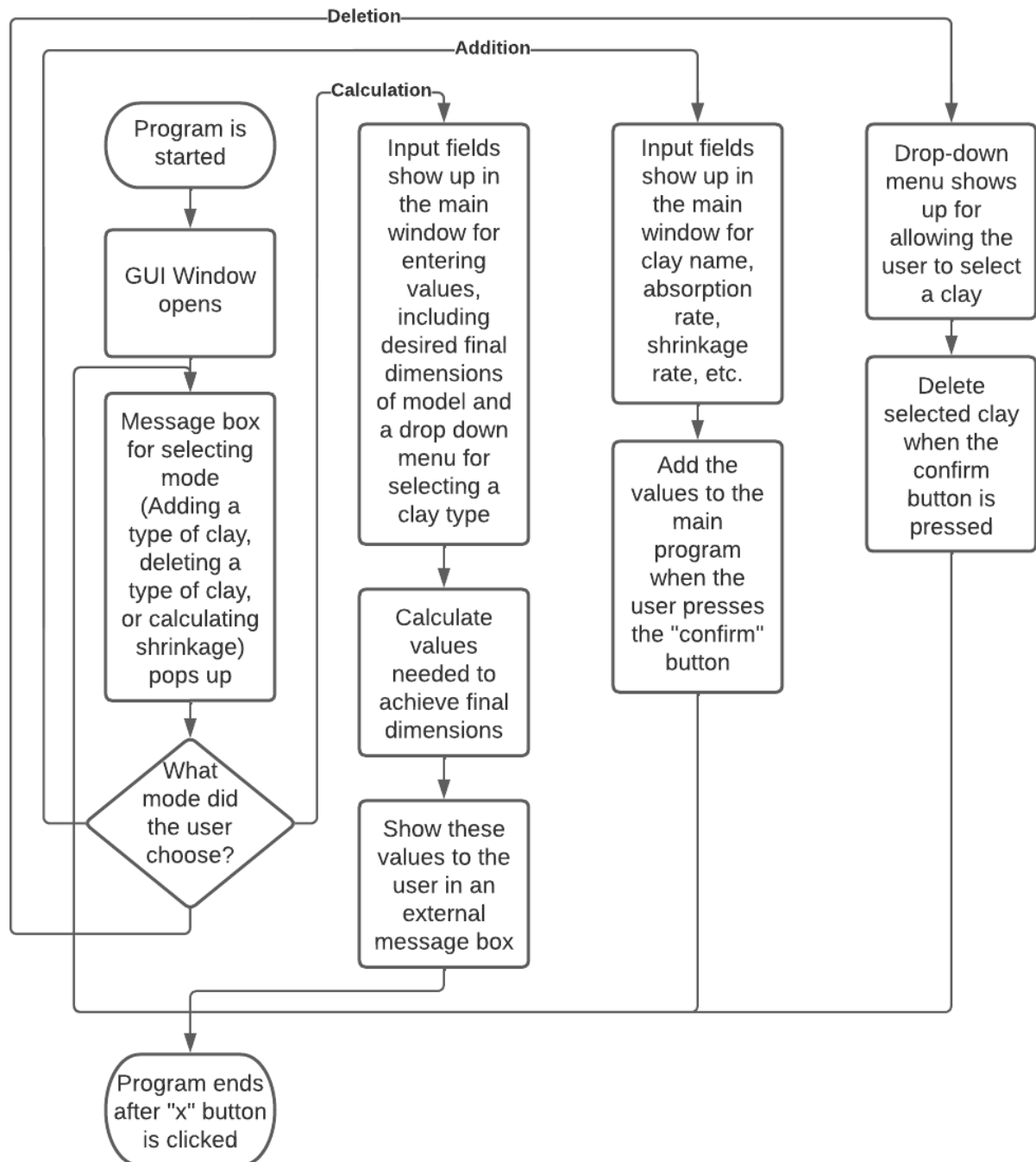
important to include to improve clarity for the end user. Whatever dimension is given is simply multiplied by the reciprocal of the shrinking percentage divided by 100.

To simplify the development process, the backend and all the different windows will be their own files. This helps to keep the code clean.

**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: Ensure inputted values are stored properly and calculated accurately.
- 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 database with an external program to view and verify.
- Program will be intuitive, and not crash when entering an incorrect value (such as an alphabetic character where a numeric value should be)
  - Make sure that no input field crashes the program
    - Example: A field is left empty or with an alphabetic character, and check if the program crashes.

## Flow Chart:



Word count: 354