Criterion A: Planning

Advisor: Aiden - Friend and classmate

Description of Scenario

      My client, yy, is into pottery. She makes bowls, plates, even goblets. This is done with many different types of clay with many different mechanical properties such as elasticity and rigidity. However it can often be hard to judge the final size of a pot (Note: in this context, pot will refer to anything made out of clay and fired in a kiln) due to the way in which it shrinks during its firing. Often, different types of clays have different shrinkage and absorption rates which can affect their final appearance and functionality.

      She currently does not have any concrete method of predicting the size of pots before they are fired, and this can, and has, led to wasted effort and pots coming out too abnormal or heavy to use. This also leads to greater consumption of clay, which costs extra money.

Rationale for Proposed Product

      I have determined that I would be able to create a program that would predict the shrinkage of different types of clay that my client uses. It will have a Tkinter frontend and allow the user to input dimensions of the final product, the type of clay in use (from a drop-down menu), and the shape of the item. It will output the final side lengths (if the product is a square) or circumference (if the product is a circle). It should be noted that the shrinkage rate does <u>not</u> change with the wall thickness, according to my client.

      I have conversed with my client about the many different types of clay she uses, and she linked me to a [helpful resource](#) that contains data about most if not all of the types of clay she uses. I can input this information in a database and apply it to the dimensions that are put into the program.
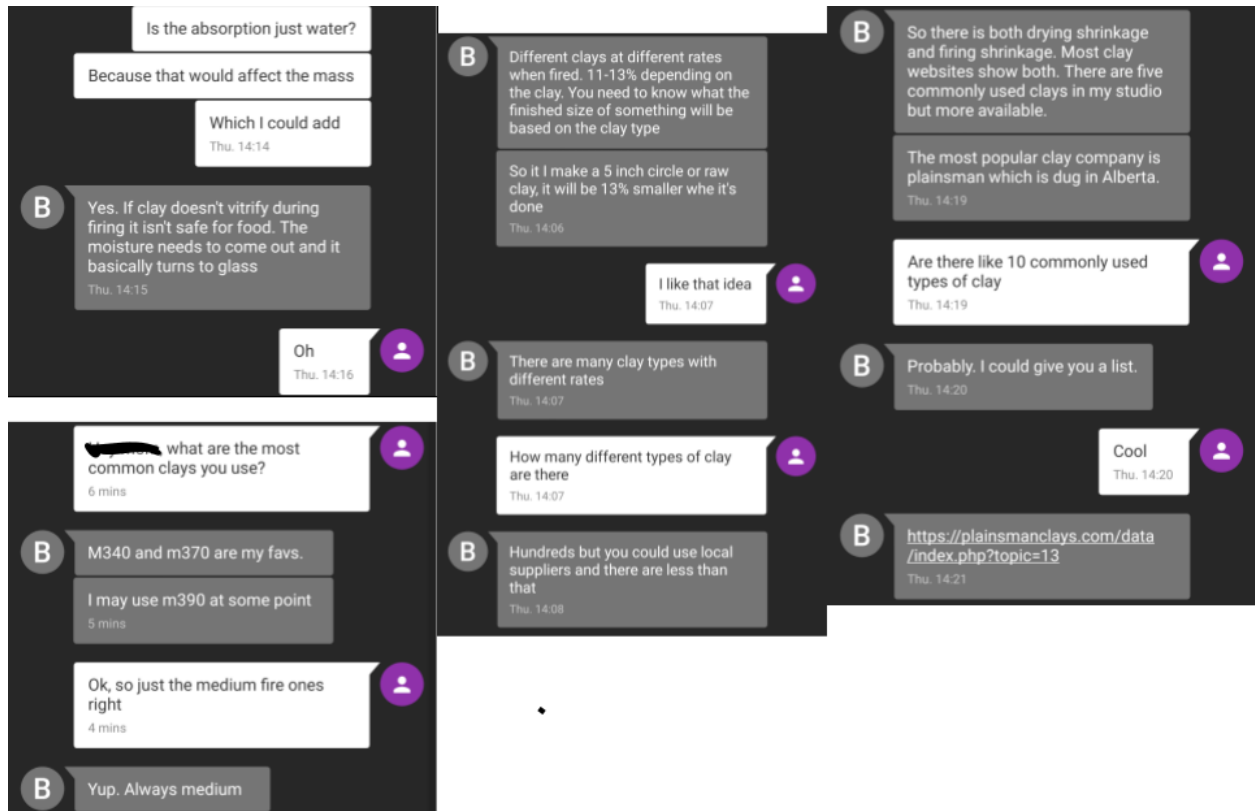
Success Criteria

- Program will contain a functional GUI, allowing the user to input values
- Program will contain a database of different clays, and allow the user to input new ones
- Program will be intuitive, and warn users when they input a bad value (such as an alphabetic character where a numeric value should be)

Word count: 370