Design Document

1. Introduction

1.1 Purpose

The purpose of this software is to have a few features which include: an option to convert pen input to keyboard text in a fast and efficient manner, clean up pen input white noise, an option to "clean up" the pen input. This software is intended for those who often use a stylus for pen input on tablets, those with hand tremors, and those with difficult to read handwriting.

1.2 Scope

The goals of this software is provide people with hand tremors, and difficult to read handwriting an option that is easy for them to use. Those with hand tremors can communicate with others more efficiently using this software. The goal is that this software will improve the readability of pen input.

1.3 Reference Material

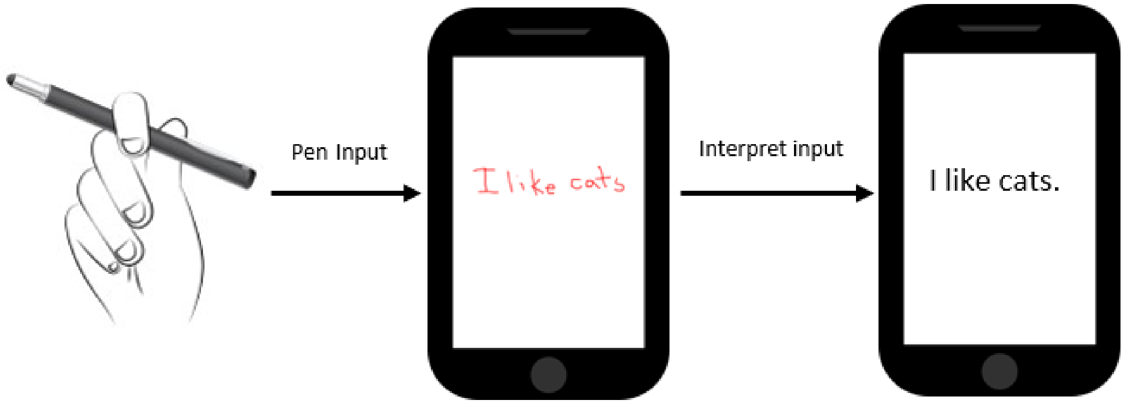
1.4 Definitions and Acronyms

2. System Architecture

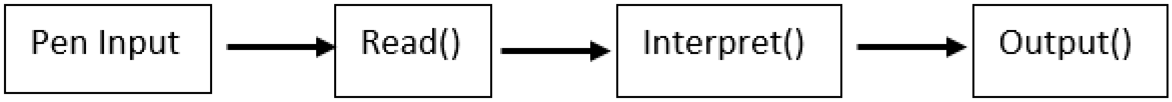
2.1 Architectural Design

Semester 1 - Pen input to text:

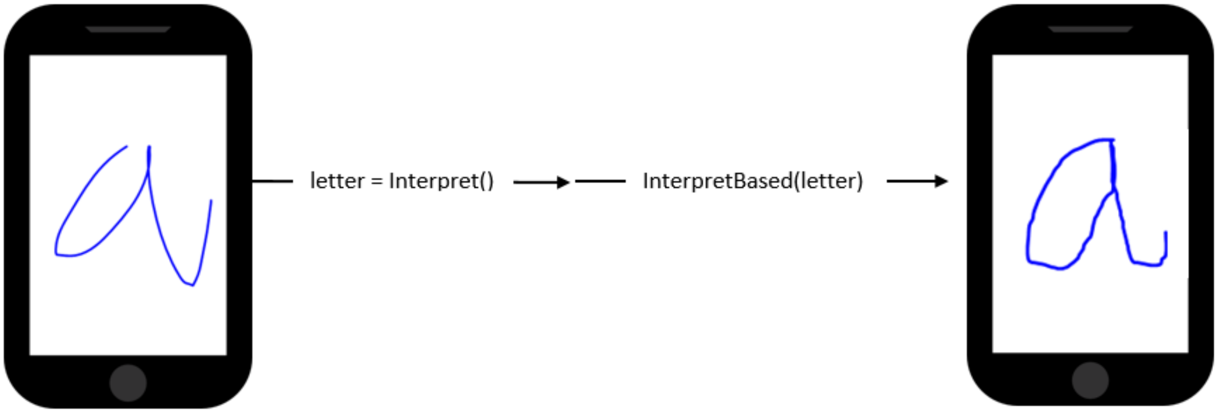
The different modules would be pen input to the tablet, interpretation of that pen input, and then output of that interpretation.



We will have a section of code to read in the pen input. That would then be read into another section of code that will have math that will interpret the input. The result would then be output to a text box on the tablet screen.



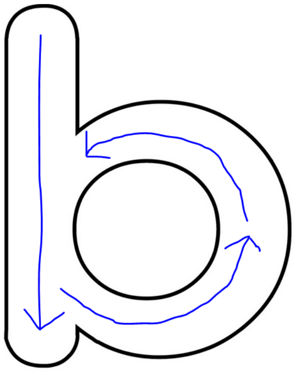
Semester 2 - Add features:
The pen input will be interpreted using the Interpret() method to determine the letter, and based on that we will use a new InterpretBased() method that will look more closely at the dots in relation to the predicted letter. A 'cleaner letter' would then be outputted:



3. Data Design

3.1 Data Description

We will have one database that contains the different stroke directions taken when a letter is written that we will refer to when interpreting the pen input, for example a standard 'b' changes stroke direction 3 times:



Many different people write letters differently so we will set up a webpage that will allow users to input certain letters and read them into a database that the software will refer to when predicting the pen input.

The pen input will be stored as dots on a graph, these dots will then be processed into a method that will keep track of when the stroke changes (the dots slope changes).

4. Component Design

To break down the pen input, the software breaks it down into the different strokes/directions.

5. Human Interface Design