IPO Chart

Component: Utility Services

Module Name: Error

Author: Mia Park
Reviewers: Team 15, Bob Dahlberg

Inputs	Process	Outputs
Error_Code Language	1. Input from the Application Service Broker is checked for incorrect input. 2. SB will be called with language.txt file and error_code.	Error_Message

Inputs:
- Error_Code
- Language

Process Description:
1. Input from the Application Service Broker is checked for incorrect input.
2. SB will be called with language.txt file and error_code.

Outputs:
- Error_Message

APIs/Objects:
- Service Broker
- Text Broker

Pseudocode:
```
/*****************************************************************************
 *      Error Class
 *****************************************************************************
 * Function:
 *      Based on incorrect input, outputs error message.
 *-----------------------------------------------------------------------
 *
 *      @author  Mia Park
 *      @version 04/14/2022   CMCS 355
 *****************************************************************************/
```

IPO Chart

```
{ //Start Main
  /*------------------------------------------------------------------------
   *    file - File that contains Option.txt
   *    a - Buffered Reader that reads from file
   *    line - String that stores lines read from a
   *    words - String[] that holds parsed line
   *     pb - Process builder that calls SB
   *    reader - var that reads from pb
   *    l - String that stores each line read from pb
   *-----------------------------------------------------------------------*/
  File file = File (Options.txt);              // File that contains Option.txt
  BufferedReader a = BufferedReader(FileReader(file));   // Buffered Reader that reads
                                               // from file
  String line = a.readLine()                    // String that stores lines read from a
  String[] words = "";                          // String[] that holds parsed line
  Process pb;                                   // Process builder that calls SB
  var reader;                                   // var that reads from pb
  String l = ""                                 // String that stores each line read from pb
  /*------------------------------------------------------------------------
   * @code
   * Format line by using ReplaceAll("\\s", "")
   * Set words = line.split(",", 2);
   * Set pb = ProcessBuilder("java", "src/modules/SB.java", "TB" + "," + words[1] + "," +
                                               args[0]).start()
   * TRY (set reader = BufferedReader(InputStreamReader(pb.getInputStream)))
   *      WHILE ((l = reader.readLine) != NULL)
   *      Print l
   *    END WHILE
   * END TRY
   *-----------------------------------------------------------------------*//End Main }
```