Appendix L: Translation Process

```
private String key;
public Translation(String apiKey) {
		key = apiKey;
	}

	String translate(String text, String from, String to) {
	StringBuilder result = new StringBuilder();
	try {
		String encodedText = URLEncoder.encode(text, "UTF-8");
		String urlStr =
		"https://www.googleapis.com/language/translate/v2?key=" + key +
		"&q=" + encodedText+ "&target=" + to + "&source=" + from;
// api request with the appropriate info stored in strings for ease of input
		URL url = new URL(urlStr);

		HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();
		// open connection
		InputStream stream;
		// taking in all the bits as a stream
		if (conn.getResponseCode() == 200) // success
		{
			stream = conn.getInputStream();
		} else
			stream = conn.getErrorStream();

		BufferedReader reader = new BufferedReader(new
InputStreamReader(stream));
		String line;
		while ((line = reader.readLine()) != null) {
			result.append(line);
			}
			JsonParser parser = new JsonParser();
			// create a parser
			JsonElement element = parser.parse(result.toString());
			// back to string
			if (element.isJsonObject()) {
				JsonObject obj = element.getAsJsonObject();
				if (obj.get("error") == null) {
String translatedText =
obj.get("data").getAsJsonObject().get("translations").getAsJsonArray().get(0).getA
sJsonObject().get("translatedText").getAsString();

					return translatedText;
				}
			}

			if (conn.getResponseCode() != 200) {
				System.err.println(result);
			}
		} catch (IOException | JsonSyntaxException ex) {
			System.err.println(((Throwable) ex).getMessage());
		}
		return null;
	}
```

Synopsis of what is happening:

- Make utf-8, access api with URL given appropriate data, from, to and key.
- Add all lines to the stringbuilder “Result”, create JSON parser and parse.
- Add translated text to a string by grabbing JSON array as a string.

```
	public static String readFileAsString(String filePath) throws IOException {
		StringBuffer fileData = new StringBuffer();
		BufferedReader reader = new BufferedReader(new
FileReader(filePath));
		char[] buf = new char[1024];
		int numRead = 0;
		while ((numRead = reader.read(buf)) != -1) {
			String readData = String.valueOf(buf, 0, numRead);
			fileData.append(readData);
		}
		reader.close();
		return fileData.toString();
	}
```

Synopsis of what this method does:

- Creates a stringbuffer for the data.
- Creates a reader to read the data as it argument.
- Reads in all of the file.
- Returns the file as a string that can be easily used.