Initial Report

# Written Introduction

# For our team project, we have been given the task to create a language translator that translates any given word or phrase from English into any language(s) of our choice and vice versa. The translations made must be true to as many official translations as possible, so we are to be aware that certain phrases in English will include certain words that will be translated differently in other languages as opposed to the translation of a single word so we are to ensure that the program recognises these phrases and avoids any mistranslations.

The data structure we will be using for this task will be a binary tree. The languages we have chosen to translate from/to English are French and Spanish. The wordlists we have considered and may use for this program are listed below:

<https://www.101languages.net/spanish/spanish-word-list/>

<https://www.101languages.net/french/french-word-list/>

# Requirements

## Functional Requirements

RF1 - The program shall have a dictionary text file that has translations from Spanish to English and English to Spanish.

RF2 - The program shall be able to save to the dictionary file.

RF3 - The program shall be able to load from the dictionary file.

RF4 - The program shall be able to translate individual words from English to Spanish.

RF5 - The program shall be able to translate individual words from Spanish to English.

RF6 - The program shall be able to translate phrases from English to Spanish.

RF7 - The program shall be able to translate phrases from Spanish to English

RF8 - The program shall provide the option to load then translate a text file and output the translation time in words per second.

RF9 - The program shall have an option to allow the user to add new words to the dictionary when a translation can not be found.

RF10 - The program shall provide an option to remove a word or phrase from the dictionary.

RF11 - The program shall provide an option to display the dictionary files.

RF12 - The program should provide the option to support the translation from English to French

RF13 - The program should provide the option to support the translation from French to English

RF14 - The program may be able to improve the translation quality by handling grammar.

RF15 - The program may have a Graphical User Interface (GUI).

## Non-Functional Requirements

RNF1 - The system shall already have Java Runtime Environment (JRE) installed.

RNF2 - The program shall be presented in English.

RNF3 - The program shall run within the windows operating system.

RNF4 - The program shall refrain from using copyrighted material not belonging to the creator(s).

RNF5 - The final program shall be provided as a standalone java program, that is capable of running without the use of an IDE.

# Initial Design

The program will contain a binary tree which will have 26 nodes (one for each letter of the alphabet), there will be a file for each letter of the alphabet.

Each line in the file will contain the word to be translated and the translated word.

## Pseudocode and Flowcharts

### Translating A Word

Prompt the user to enter the word they want translated

Get the first letter of the word

Search the binary tree for that letter

Search the file of that letter for the line that contains the word that you are trying to translate

If the word is found

Get the translated version of the word

Display the translated version to the user

Return translated word

Otherwise

Display that there is no translation for the entered word

If the user wants to add their own translation

Run ‘Add a translation’ method

Return translated word

Else

Return word

### Translating a Phrase

Prompt the user to enter the phrase they want translated

Split the phrase into individual words

Repeat the ‘Translating a word’ method for each word in the phrase, keeping track of each translation

If any words are not translated and the user wants to add their own translation

Run ‘Add a translation’ method for each translation that is being added

### Add a Translation

Ask the user to enter the translation to be added

Save the entered word-translation to the correct file

Save the reverse of the translation (translation-entered word) to the correct file

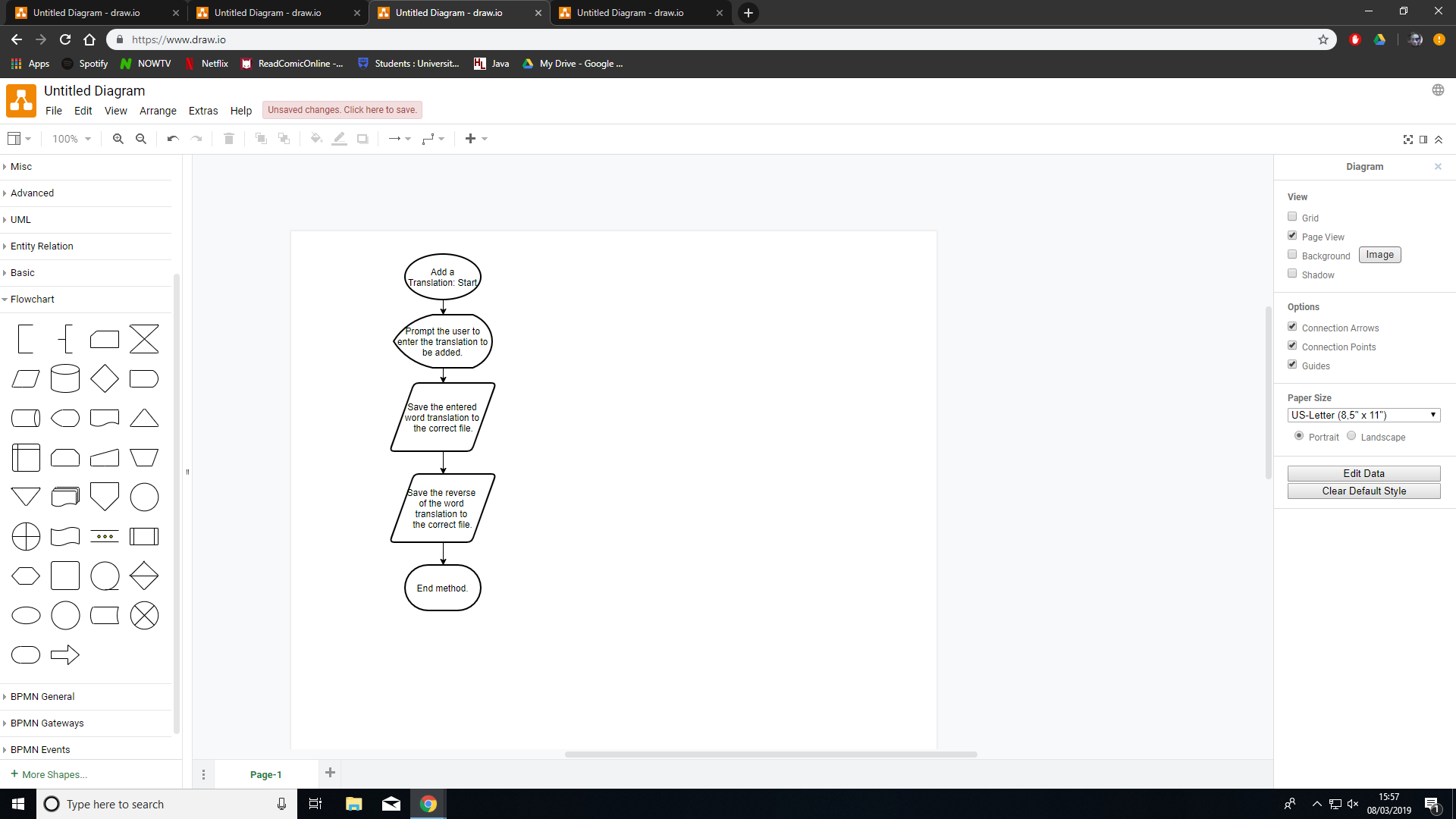
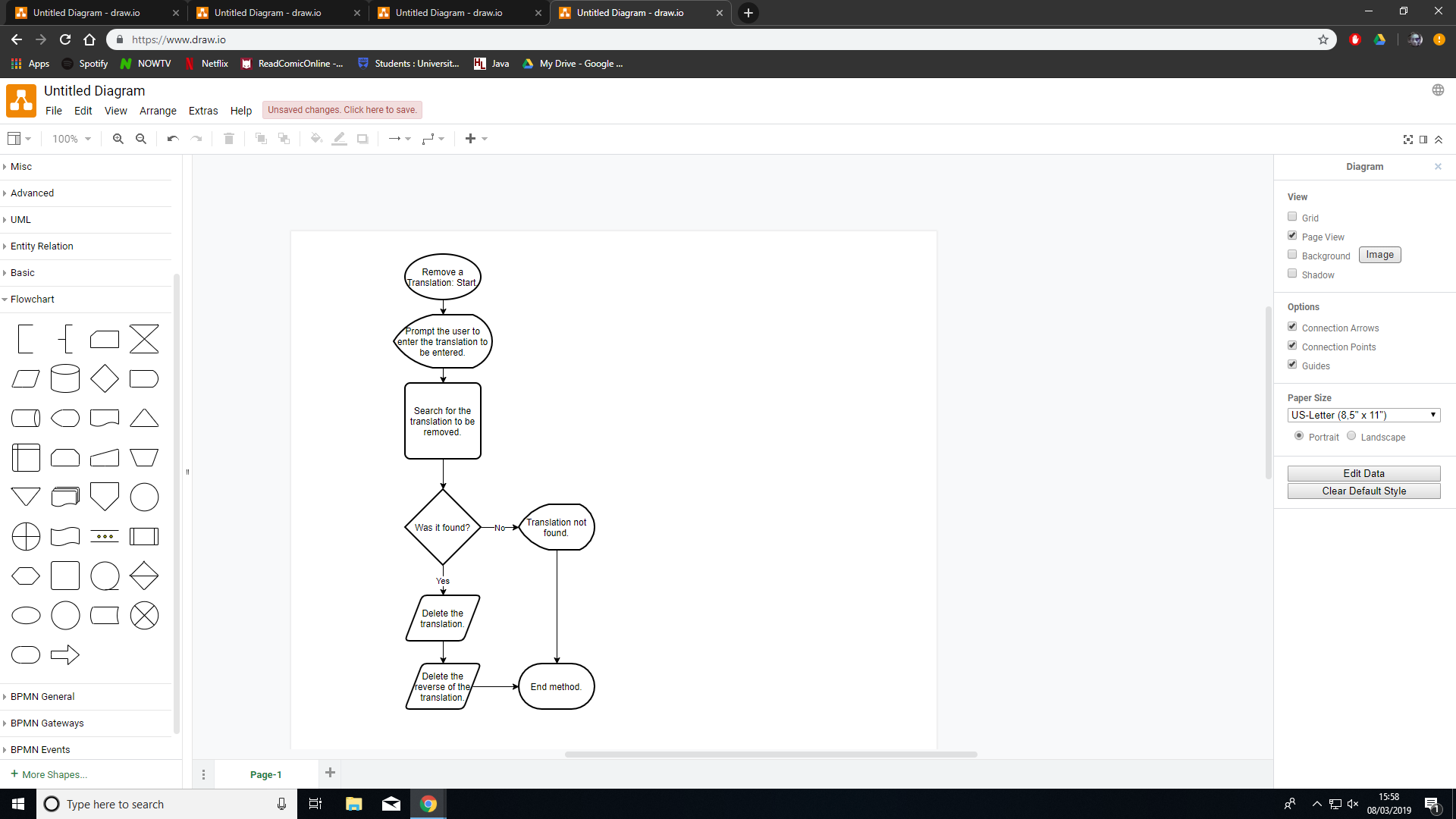
### Remove a Translation

Ask the user to enter the translation to be removed

Search for the translation to be removed

If found

Remove the translation to be found

Remove the reverse of the translation that was found

# Task Allocation

We will all work on the adding translations to the translation files (min 5000 translations) as this will be sharable as there will be 26 files and a lot of translations to add

**Beth** - Group management, GUI design and development, adding improved grammar handling, binary tree from first principles classes/methods, user manual

**Holly** - Translating a word method, searching for a translation, calculating the translation time (words per second)

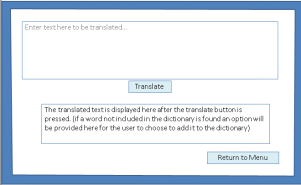
**Jack** - add a translation method, remove a translation method, translate from a file method

**Michael** - displaying the whole dictionary method (in order), translating a phrase method (relies on translating a word method), adding another language

As some tasks rely on other tasks being done, the task allocation may change but this is our initial plan for this project. We have distributed the coding tasks as evenly as possible as some methods will be larger or more complex than others.

User Interface Initial Design





Group Project Management

**Wednesday 6th March - 1pm-2pm at Library**

* All four group members present.
* Discussed further meet up plans for completing initial design document due Friday and detailed our ideas on how we would each prefer to go about it.
* Came up with an effective method of completing the document by assigning each of us tasks to complete by friday, at which point we will all look over the document together and amend as necessary.

**Thursday 7th March - 3pm-5pm at QMB**

* All four group members present.
* Worked on the initial report and design, finishing the requirements and creating a rough draft of our collective thoughts on the program design. Checked up on each others progress so far and gave each other pointers where necessary.

**Friday 8th March - 2pm-4pm at QMB**

* All four group members present.
* Completed initial report document and checked we were all satisfied with each element of it.
* Allocated programming tasks for us to begin working on individually.