Interim Report

Carim Akeju, Erik Jeny, Callum Nicholson

# Introduction

# Outline of Requirements

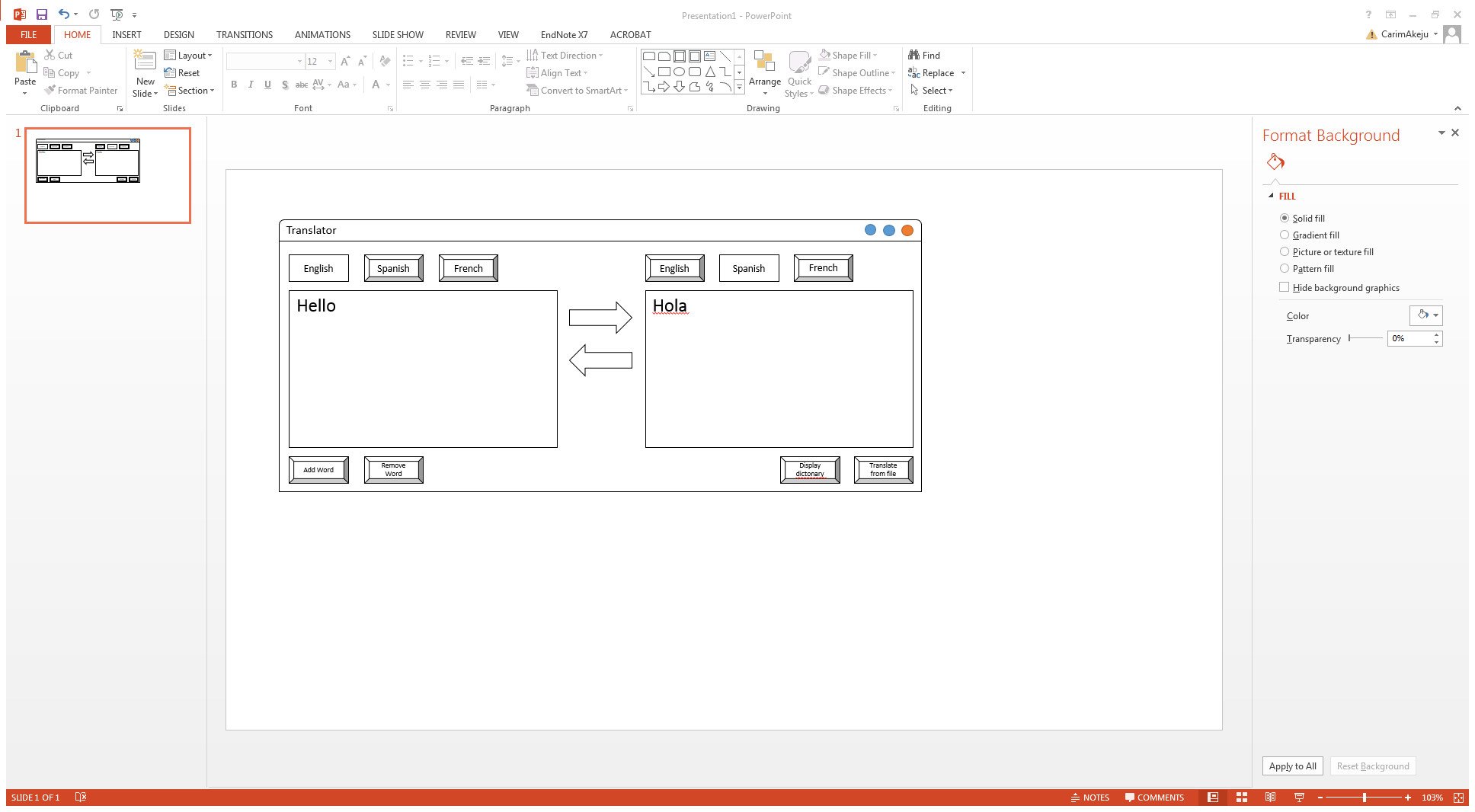
As stated in the brief, we are required to do the following:

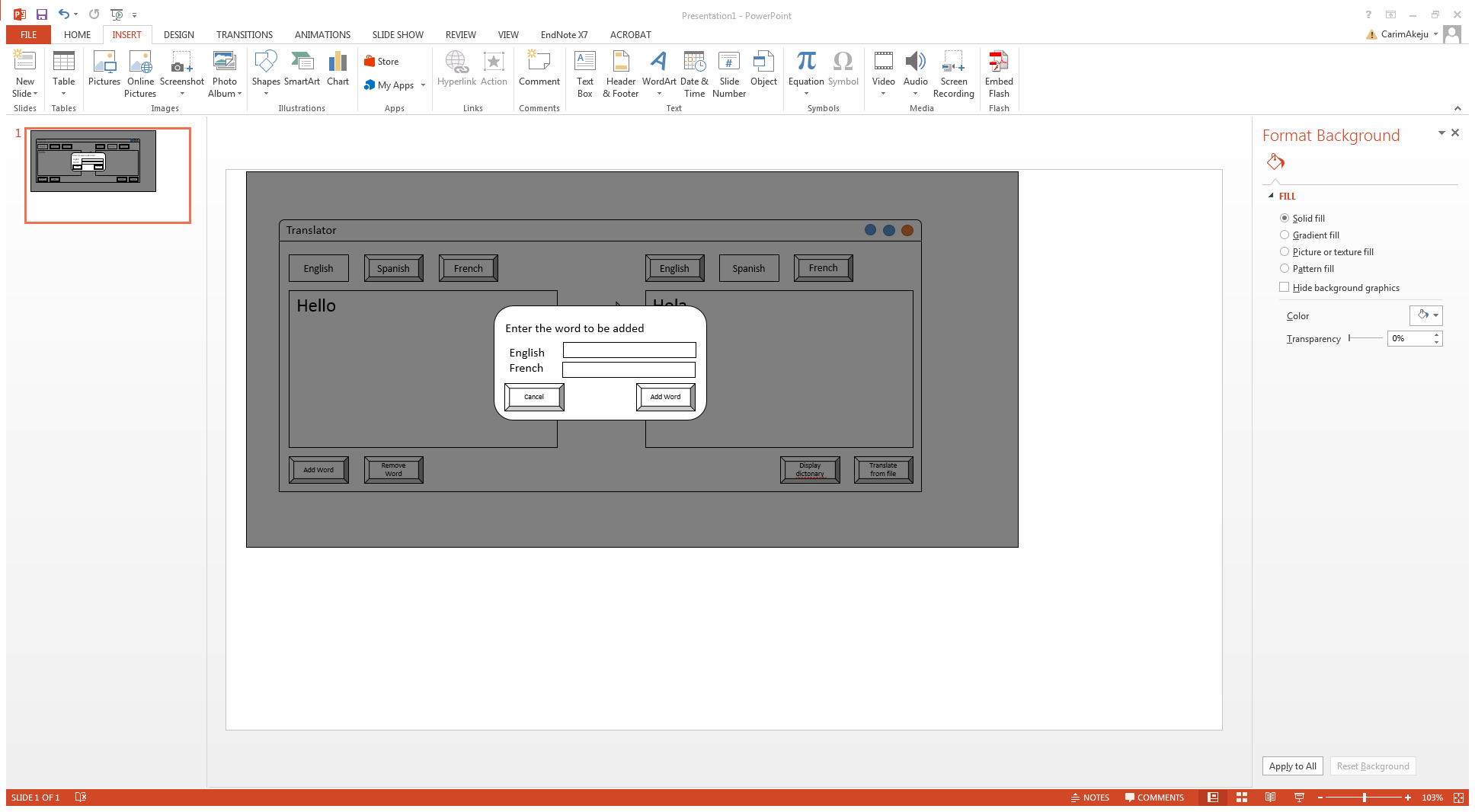
* Create a dictionary with an appropriate data structure. We are going to use a Trie Tree (also known as a digital tree or prefix tree), where nodes are not associated with a key, but nodes are defined by a single character. Words are built by traversing along individual letters and the translation is obtained at the final node required.
* Translate individual words.
* Translate phrases. We have opted to prioritise phrases during translation.
* Load a file and apply a translation.
* Allow the user to add words to the dictionary.
* Allow the user to remove words from the dictionary.
* Allow the user to display the dictionary.
* Produce a standalone executable.

We can also:

* Produce a Graphical User Interface
* Include methods of translating grammar correctly.
* Include more languages.
* Optimise our methods of searching and comparing data (for example, we aim to optimise the Trie Tree by not storing all 26 letters as a node, and only creating them when they are needed).

# User Interface Mockups





# Pseudo-code

* Adding a word
* Finding a word
* Removing a word
* Displaying the dictionary

# Task Allocation

We are opting to use GitHub (<https://github.com/CarimA/AC12001-Team-Project-Translator>) for version control and team management. Erik will be handling the GUI, Carim will be handling the Trie structure and word management, and Callum will be handling phrase management and file management.