

Computing Project

Toki Pona spell checker

PROJECT PLAN

Contents

1. Outline of the project	3
2. Initial designs, ideas and prototypes	4
3. Additional resources required	5
4. List of tasks	5
5. Weekly Plan	6

1. Outline of the project

- Title: “Spell Checker and Predictive Text for Toki Pona Language with Name and Place Recognition”
 - Introduction –The purpose of this project is to develop a spell checking and predictive text system for the Toki Pona language. The Toki Pona language has a vocabulary of only about 130 words, and the system checks words and sentences for the validity of names of people and places. The system will also be designed to provide predictive text suggestions and check the grammar of input sentences.
 - Background of the project-Toki Pona is an artificial language created in 2001 by Canadian linguist and translator Sonja Lang. The language is designed to be simple and expressive, with a vocabulary of just 120-130 words. But Toki Pona still has a small and loyal following around the world.
Since Toki Pona is a relatively new language, there are very few language processing tools available. The Toki Pona spell checker and predictive text system will help make the language easier to use.
 - Problem Definition- The problem I need to investigate is that the limited vocabulary of the Toki Pona language poses a challenge to making a spell checker, as the system needs to be able to recognize and suggest the correct spelling of words, while also considering the context of the sentence.
I hope the project has a function that can accurately check the spelling of words in Toki Pona sentences and suggest appropriate corrections. I hope to develop a system that can check the grammar of the input sentences.
 - The project's audience includes anyone who can speak, read, or write the Toki Pona language, the native speakers and people learning the language as a second language.
 - Motivation- The reason I chose Toki Pona Spell Checker for my computer project was because of its uniqueness, Toki Pona has a small and dedicated user community. Creating a spell checker for Toki Pona would be a great project, so I'm very interested in it.
In terms of industry, a spell checker and predictive text system for Toki Pona can be useful for various applications, such as communication tools, translation software, and language learning applications. It can also be useful for businesses that operate in regions where Toki Pona is used, allowing them to communicate more effectively with their customers and stakeholders.
- Objectives/Aim
1. Develop a spell-checking system that can accurately check the spelling of words in Toki Pona sentences and suggest appropriate corrections.
 2. Develop a grammar checking system that can check the grammar of input sentences.
 3. Complete the java code within 3 weeks.
- Method
 - 1.Valid word list: To check whether a given word is spelled correctly, the system needs to have a valid word list in the Toki Pona language.

2.List of valid names: Since the spell checker needs to recognize names of people and places, a separate list of valid names is required.

3.Function to check if a word is spelled correctly: This function takes a word as input and returns true if the word appears in the list of valid words or names.

4.A function to suggest words based on a prefix: This function takes a prefix as input and returns a list of words starting with that prefix.

5.Function to check sentence grammar: This function takes a sentence as input and checks whether the sentence follows Toki Pona's basic grammar rules.

- Time Plan

Toki Pona text data collection: Collect all words of Toki Pona text data to spell checker. It is expected to be completed in 2-3 days.

Basic Spell Check: Includes checking if a given word is spelled correctly and suggests alternative words based on prefixes. It is expected to be completed in 2-3 days.

Proper Noun Recognition: A spell checker that recognizes names of people and places. It is expected to be completed in 2-3 days.

Grammar Check: Check whether a sentence follows Toki Pona's basic grammar rules. Expect to take 4-6 days.

Testing and Validation: Write test code and test the spell checker on various Toki Pona texts and verify its accuracy and performance. Expect to take 2-3 days.

2. Initial designs, ideas and prototypes

Design and Ideas of my project:

Check if a word is spelled correctly

Recognize people's names and place names

Suggest words based on a given prefix

Check if a given sentence is grammatically correct.

Pseudocode:

Class TokiPonaSpellChecker

List validWords

List validNames

```
function isSpelledCorrectly(word)
```

```
    if word is in validWords or validNames
```

```
        return true
```

```
    else
```

```
        return false
```

```
function suggestWords(prefix)
```

```
    for each word in validWords and validName
```

```
        if word starts with prefix
```

```
            add word to suggestions
```

```
    for each name in validNames
```

```
        if name starts with prefix
```

```
            add name to suggestions
```

```
    return suggestions
```

```
function isGrammarCorrect(sentence)
```

```
    split sentence into words
```

```
    check grammar rules based on the number of words and their forms
```

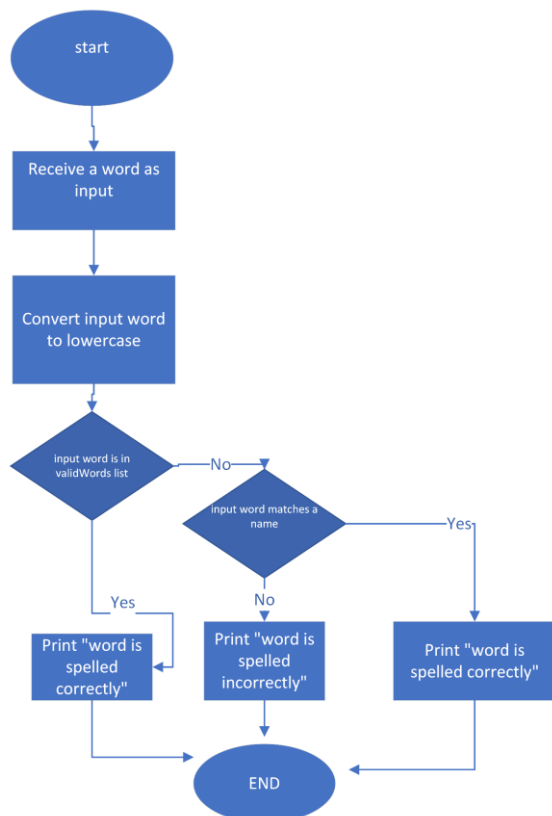
```
    return true if grammar rules are met, else return false
```

Flow charts:

TokiPonaSpellChecker class:



Spell Checker method Flowchart:



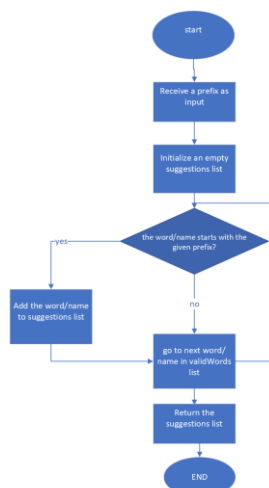
addWord method:



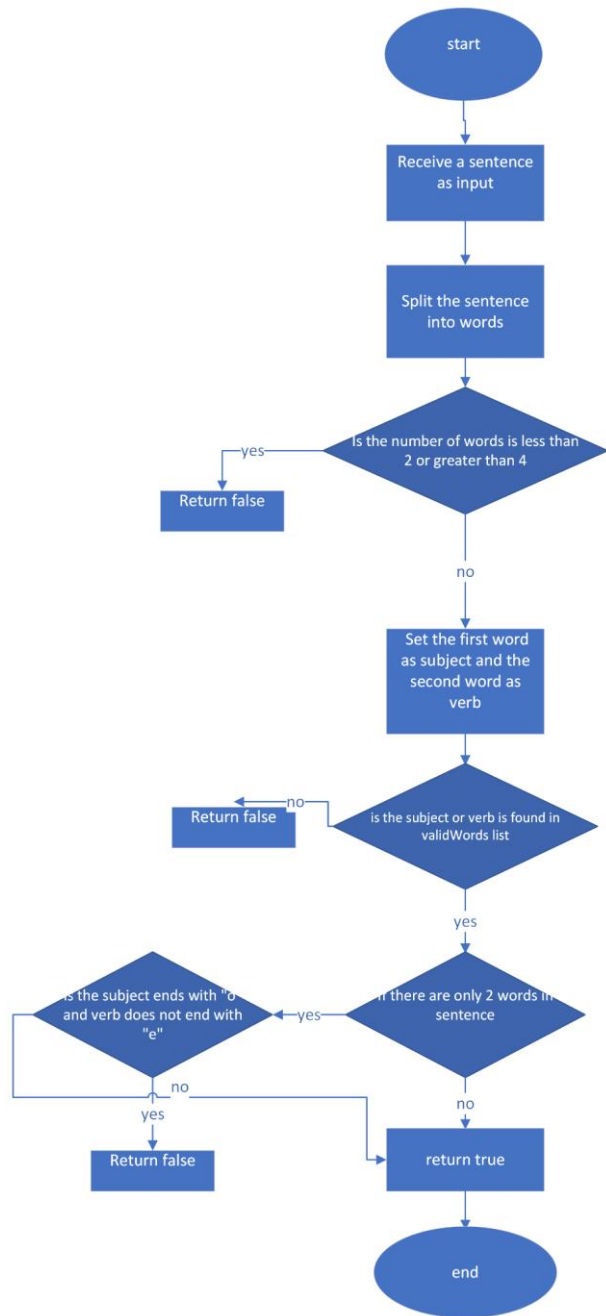
addName method:



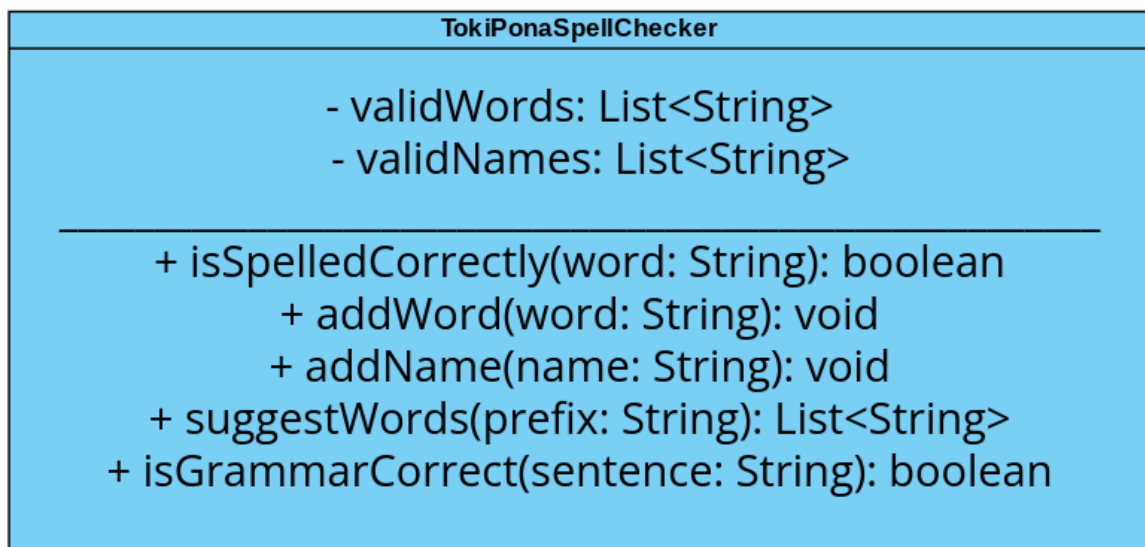
Suggestions Flowchart:



Grammar Checker Flowchart:



Class diagram:



Choices and Reasons:

Choice of language: I have decided to use Java as the programming language because I have spent the last 9 weeks learning Java, and I feel confident in my ability to use it.

Data structure for valid words and names: I have used two separate lists, one for storing valid words and one for storing valid names. It can simplify my checking process and allows for easy modification and adding new words and names.

Grammar checking code: To check if a given sentence is grammatically correct, I use a rule-based approach that takes into account the number of words in the sentence and their forms. It can help me to add more grammar rules easily.

Suggestion code: In my suggestion code is a prefix-based search that will find words and names starting with the given prefix in the valid words and names lists. I think it will work well for a language with a small number of words like Toki Pona.

3. Additional resources required

- Research:
Research the Toki Pona language and its vocabulary to ensure that the spell checker accurately checks the spelling and grammar of sentences.
Research software testing methods to make sure spell checker is thoroughly tested and bug-free.
- Skills/Technologies:
Java Programming skill to develop the spell checker.
BlueJ for Java development.
Software testing skills to ensure that the spell checker is thoroughly tested and free of bugs.
- People:
Check with the teacher to make sure the spell checker is checking the spelling and grammar of the sentences accurately.
- Materials:
A computer with Java development environment and bluej installed.
Toki Pona language resources and materials such as all 130 words and grammar of Toki Pona.
- Acquiring Resources:
Research on the Toki Pona language through online resources e.g. Toki Pona official site (<https://tokipona.org/>)
- Dev Diary:
All algorithms will be developed and tested throughout the project.
Test code will be written for each algorithm to ensure its correctness.
All my code optimizations will be done as needed for efficiency.
- Prose Description:
The challenging algorithm is the grammar checker, which will need to accurately identify the subject, verb, and object of a sentence and determine whether the sentence is grammatically correct according to Toki Pona rules. it is not easy to do it, I think I hope I will not get a bug here.
- Testing Plan:
I will write a Test Code to ensure that the spell checker is functioning correctly and free of bugs.
- Evidence and Actions:
All my testing will be recorded in the report that we need to in week 5, including the results of each test and any bugs or issues found.

4. List of tasks

Research and gather resources on the Toki Pona.(all 130 word and Grammar)

download/update BlueJ.

Develop a basic spell checker algorithm that can check for spelling errors in a sentence.

Develop an algorithm for suggesting correct spellings for misspelt words.

Develop an algorithm for detecting and suggesting corrections for grammar errors.

Write tests code for each algorithm and ensure their correctness.

ask the teacher/classmate to test it and get feedback.

Make improvements and fixes based on feedback and testing results.

make the report document for the project in week 5.

Evaluation against Time Plan:

Each of my tasks have a deadline and are prioritised according to its importance and impact on the overall project. If a task is taking longer than expected, I will adjust accordingly.

Third Party Feedback:

I will ask the teacher/classmate to test it and get feedback when I complete my code.

Development Review:

I will check the progress of the project at the end of each week and identify anything that needs improvement.

Additions/Changes with More Time/Information:

If I had more time or information, I might add features like multilingualism, and further optimise the algorithm of the grammar checker to make it more efficient. Additionally, I will spend more time testing to ensure that the spell checker is thoroughly tested and bug-free.

Grade: 90% (30 pts possible)

An excellent plan. Flowcharts and pseudo code is sufficiently detailed.

Class diagram likely to be rather incomplete and omits getters and setters.

Reasoning and arguments are sound. Well done.