Bishop's University
Basic Spell checker - Challenge 5
Final report -Pirashanth Satkunabalasingam- 002244891

Problem definition

The main problem of the basic spell checker is about the spell detection algorithm of the project. It is planned in the presumption that all word errors are the words that are NOT in the dictionary. These are delegated nonword spelling error. Be that as it may, there are situations where spelling mistake isn't just a "spelling mistakes", think about the following scenario:

I would like a pieece of cake as disert.

By simply looking at the words in the sentence above, you can tell the mistakes. However, errors still occur as the word "pieece" and "disert" are not suitable to the context. They are called real world spelling errors. In a spell checker that uses dictionary check, this kind of error will have to be checked in the first place.

It is clear that dictionary check is the main optimal spelling detection method. In addition, there are four types of the problem on spell suggestion as well. In the basic spell checker above, the two types of the spelling mistakes has been given. However, There are another two types of mistakes are given in major problems. Such as " contan" instead of "contain" or "seroius" instead of "serious". can all spelling suggestions be done by just analysing a single word? It is easier to think this problem with a case on search engines. In search engines, imagine a search on word with mistakes "ebautiful" and "beautiflu". Both are the same kind of mistake It is impossible for a search engine but it required different kind of algorithm to figure out. There is a problem derived: single word has insufficient power on searching. Similarly, in spell checking, a word on its own provide insufficient information for analysis. Look at the example below:

Is htere a solution to thier problem for whn they're traavelling?

Four nonword spelling errors occur in the context above. For a basic spell checker, those errors will be classified correctly. However when user wanted to get a list of suggestions, the same suggestions will be given to all four words, regardless of the algorithm used in suggestion stage. Sometimes it will even suggest a wrong word to the context. Hence, matching to dictionary is not the best approach for a spelling correction. But picking up the word which occurred most frequently will solve most of the issue that occurred during the suggestion. Because of the ability to detect to a real word error and to give a suitable suggestion by analysing the word and the frequency. Hence, it made the spell checker program in an advanced way as the problem

Summary of the solution

As a solution, Program has been developed by using Java language and Own algorithm for the Spelling check for the four main types :

- One character in the string gets deleted incorrectly
- One character in the string is incorrectly replaced by another one
- While typing hurriedly, the user ends up swapping one pair of consecutive characters
- The user ends up inserting one extra character somewhere in the string

The project doesn't contain any kind of library for the implementation. The project has been provided with a Corpus of text which program can read in as a file in my implementation. it is placed in the same

folder as my program. The program will read in this text, and build up a dictionary of words and the frequency with which those words occur. As a first step user will get a user interface with option to read file. It has two option whether he needs to display all the words and the frequencies of the file or not, BufferedReader functionality of the java has been used to read file from the given corpus file. While reading file program is clearing many unwanted symbols(/*+=_)(*&^%$#@!:;][?><) other than hyphens and/or apostrophes as mentioned in the requirement. Then every word has been added to the HashSet based with the frequency. After those process user can be able to select the input type as N*N input or single input. When user enters the input word or words it will be added in a list. The list will start to analyse the word by comparing the words in the HashSet as a first step it will analyse whether the word is available in hashmap or not. If not it will go to the first spell checking problem.

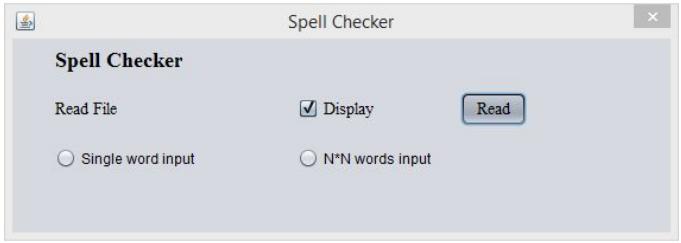
The first algorithm for the problem solved two problems at the same time “One character in the string gets deleted incorrectly & One character in the string is incorrectly replaced by another one”. While comparing input word with hashmap if the input word one letter less than hashmap word it will start function. The first step is counting match letters from front and store the total of first count. Then same process from behind. If the total of the first count and the second count one letter less than a correct word it will change the correct word to lowercase and store in a lexicographicallist. If that doesn't happen it will go to the second algorithm.

The second algorithm is same as process one but it has small changes in it. When it is matching from front it will store unmatched two letters as letter1 and letter2.(Montreeal) If both letter one and letter 2 are equal like the word given. It will store both “EE” from the previous word. Other than that if the total of the matching letters from **frontcount + backcount-1=correctwordcount** it will added in the list as lowercase also if the extra word doesn’t match also it will do the same process for more efficiency comparison. If the word doesn't match this scenario it will lead to the third algorithm for the fourth problem.

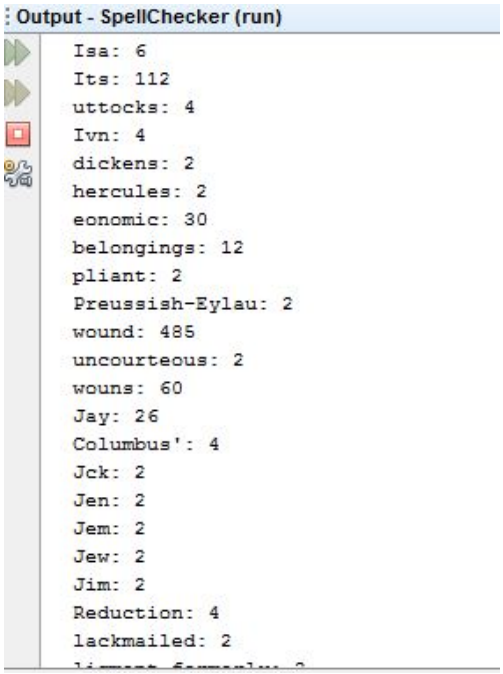
The final algorithm for the While typing hurriedly, the user ends up swapping one pair of consecutive characters. If The basic process of the algorithm will be start function when the letters count of the real word is equal letters count of the input word . The first process of the algorithm is to match the whole word one by one in for loop and count if the word has 2 unmatched letter continuously but also the first letter of the input word match with 2nd letter of the real word and second letter of the input word match with first letter of the real word it will add 2 counts to the match count. In that case if a match count of the letter is equal to real word the lowercase of the real word will be stored in list. If my program doesn't match any of the scenario the program will output the same mis-typed word it self.

That’s how program has restricted the choice of each popular case of the problems. If the word has many suggestion input for a word it will pick up the word which occurred most frequently. When it is N*N If there are multiple such words which occurred most frequently, then it will display which occurs first in lexicographical order using Collections.sort(list).

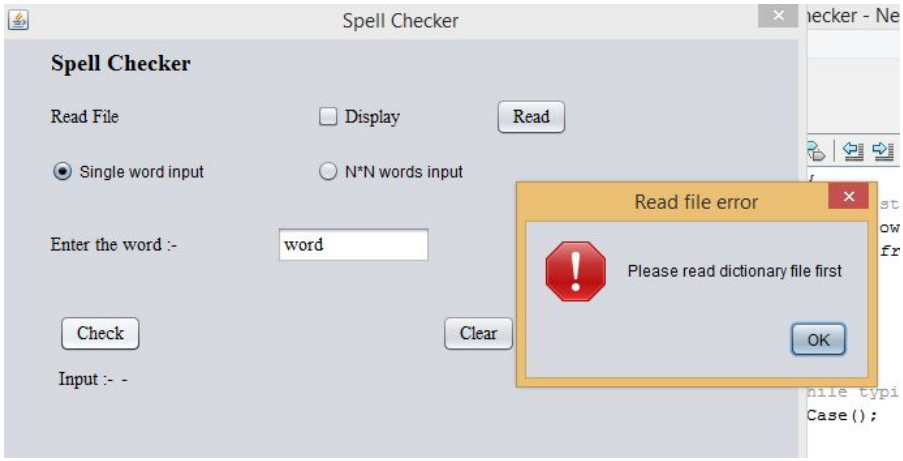
Program and Results



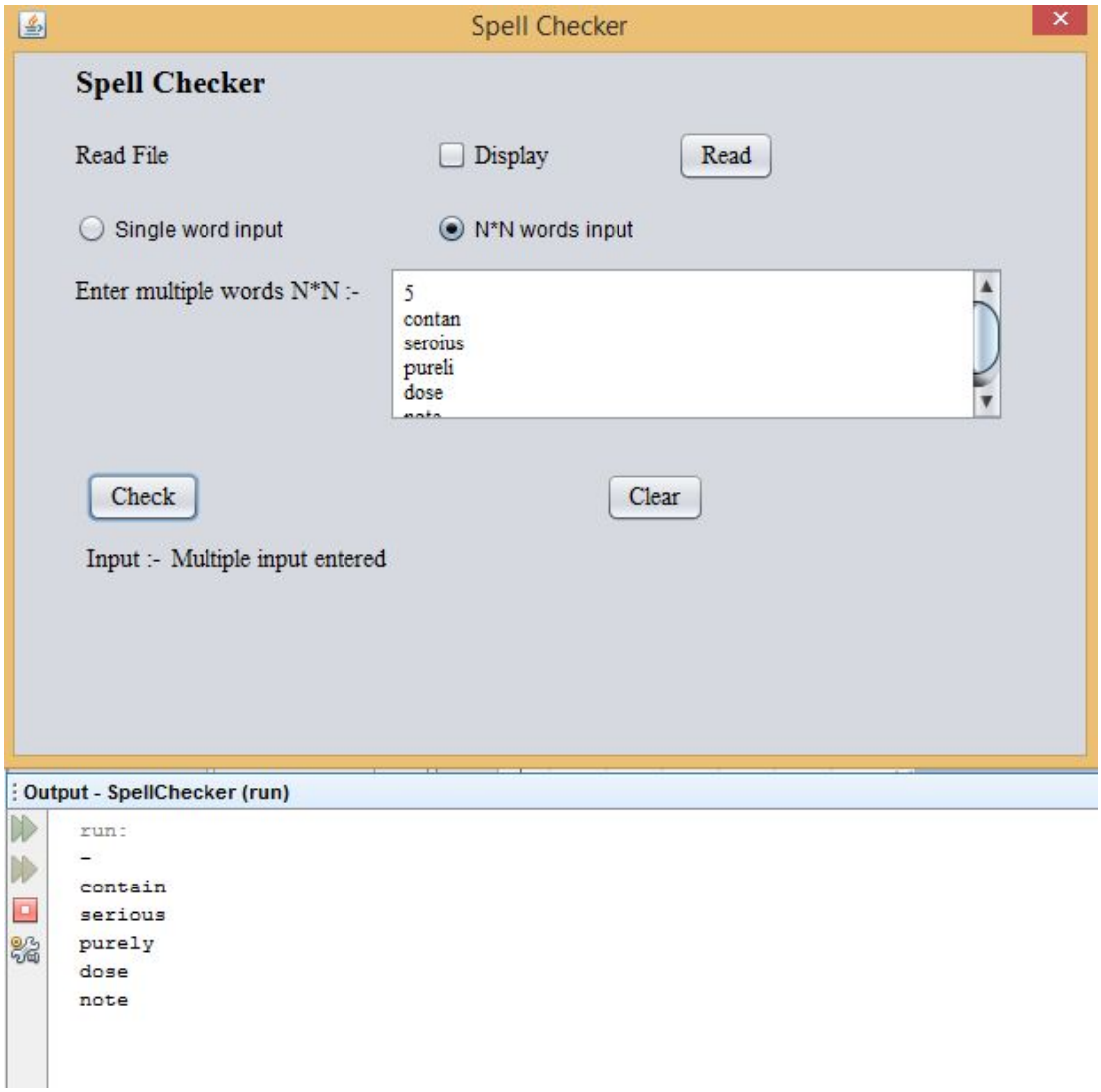
This image shows the basic idea about reading a file. User can be able to select the option to display the words and frequency. Moreover, after reading the file user can be able to select one of input section such sa single word input or he wants to enter multiple words inputs.



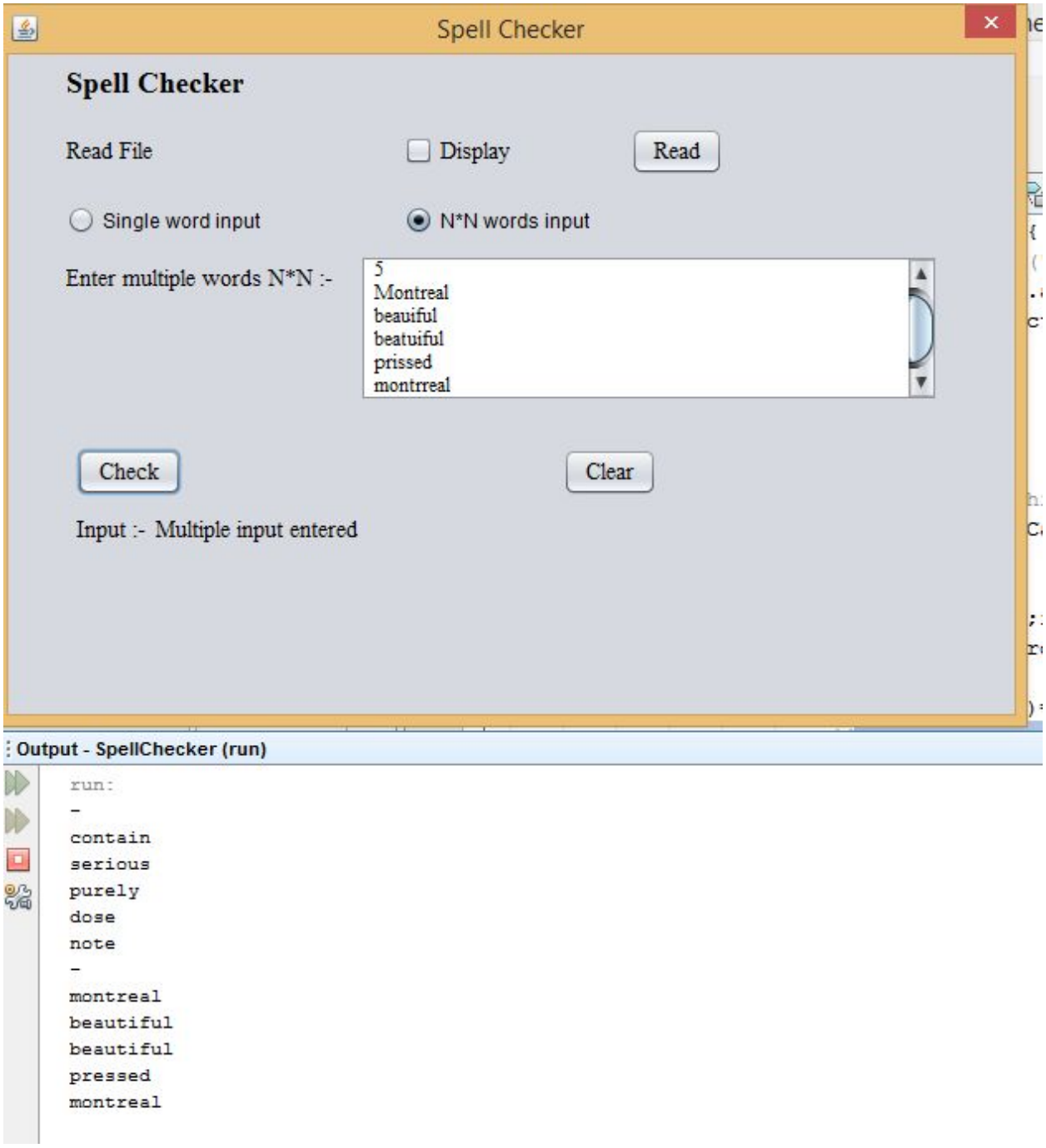
If the user tick the display function it will show in the output console. But it will consume a lot of time since it is showing the whole content. So it is preferred to untick the display function in the previous picture. Also It wont show the duplication of the same word or any unwanted symbols as you see in the image.



When the user forget to read the file and went to further function it will show the error popup. So user can read file in the same process without restarting the whole process.

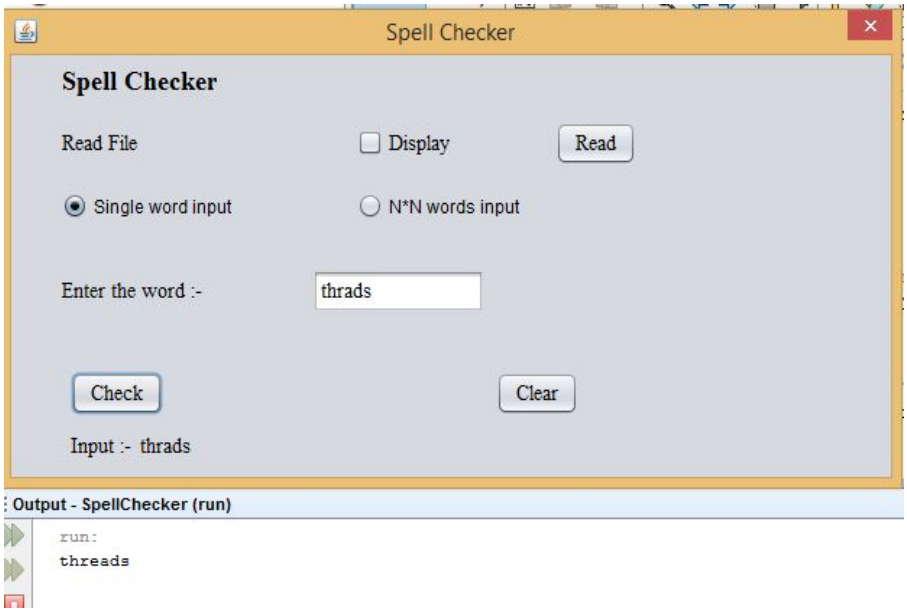


This image shows the first sample N*N input function as required in requirement. The first element if the size of the words which has been entered below. And the second input is"contan" which one character in the string gets deleted incorrectly. Third input is "seroius" While typing hurriedly, the user ends up swapping one pair of consecutive characters. And forth input is "pureli"One character in the string is incorrectly replaced by another one. Fifth and six are "dose" and "note" which input are correct values. As you can see in the output in the console successfully filtered the correct words for all input.

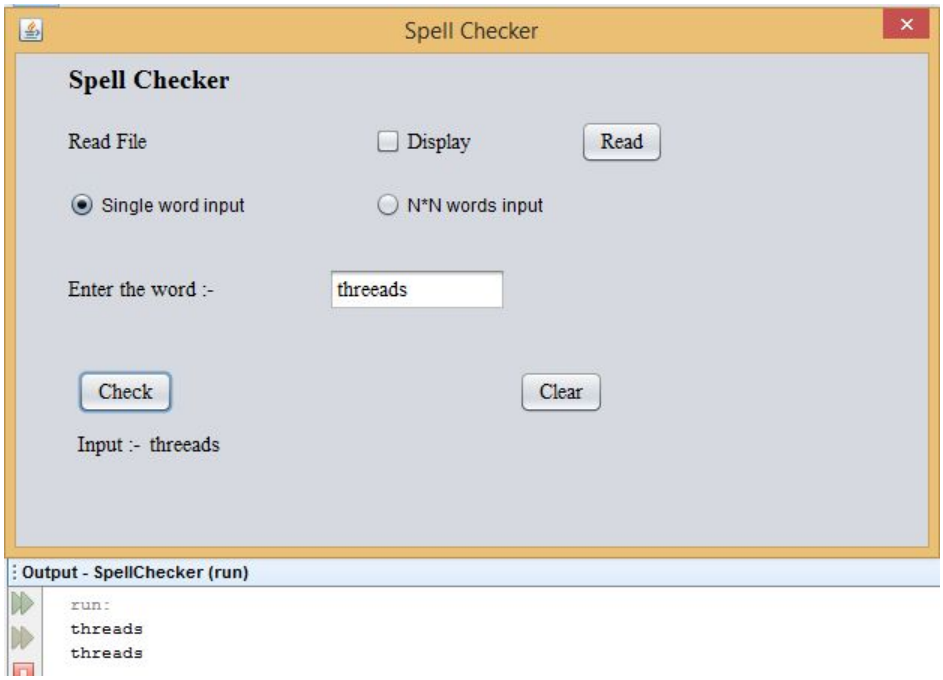


This image shows another sample of the different 5 inputs with 1 correct word and 4 special case of the problem in the same console of the previous sample. The first element if the size of the words which has been entered below. And the second input is"Montreal" which is a correct available word in the dictionary. Third word is "beauiful" which one character in the string gets deleted incorrectly. Fourth input is "beatuiful" While typing hurriedly, the user ends up swapping one pair of consecutive characters. And fifth input is "prissed "One character in the string is

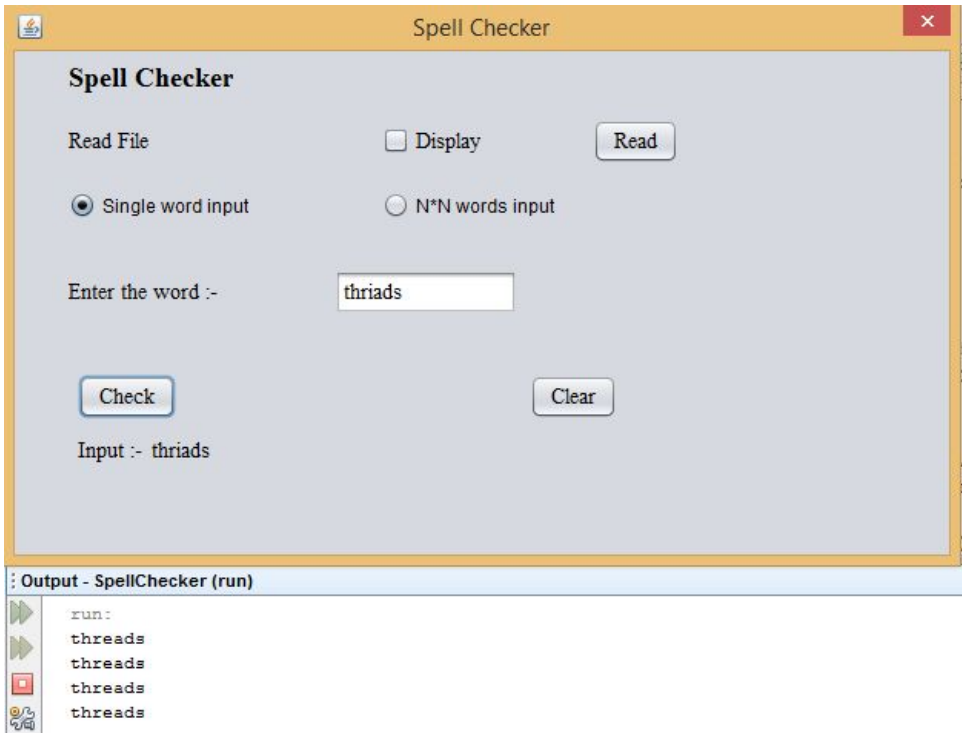
incorrectly replaced by another one. And the sixth input is “montrreal” one letter The user ends up inserting one extra character somewhere in the string.



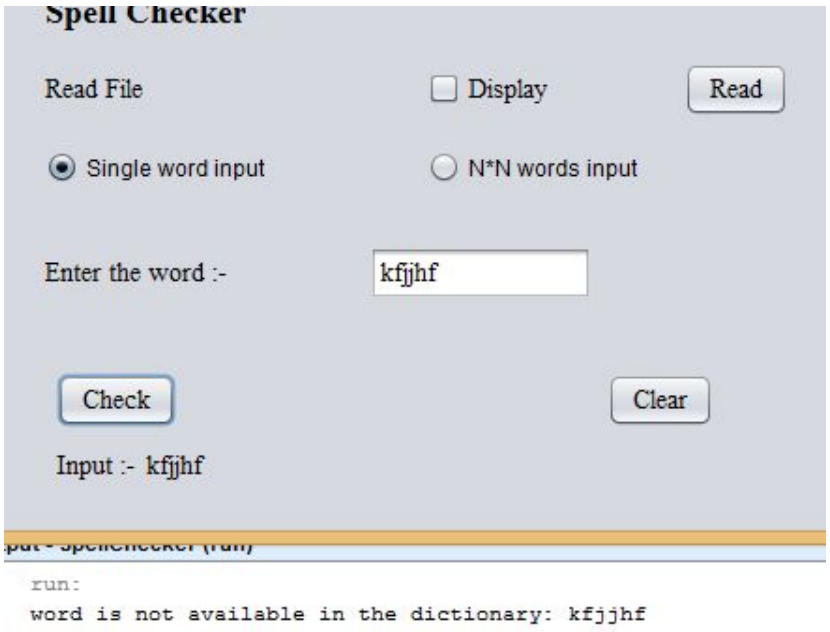
This image shows the single word input scenario. User has entered the word “thrads” which has missing one letter in the middle. But the program detect the word and displayed correctly.



This image shows the single word input scenario. User has entered the word “threeads” which has an extra letter “e” in the middle. But the program detect the word and displayed correctly.



This image shows the single word input scenario. User has entered the word “thriads” which means instead of “e” user has entered “i” in the middle. But the program detect the word and displayed correctly.



This image shows the single word input scenario. User has entered the complete wrong word which is not available in the dictionary “kfjjhf” But the program detect the wrong and displayed the word that user entered with correct message saying word not available.

Conclusion

The project has been developed and well tested according to requirement and scenario. The results of the shows most accurate output.
All the test went smoothly without any problems, But there might have possibilities for the wrong results for the different type of errors as an input .An interesting future development can be covered to solve that issue.

References

Websites

- https://www.w3schools.com/java/java_intro.asp - Basic Java
- https://www.educba.com/java-list-vs-array-list/ Difference of array vs list
- https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html- BufferReader for reading the file
- https://www.javatpoint.com/java-list -Java List

Books

- **Head First Java** - Originally published: 2003 - Authors: Kathy Sierra, Bert Bates
- **Java Concurrency in Practice** - Book by Brian Goetz