Business Logic:

1. The project is used for check input word spelling and return multiple suggestions for the user.
2. We type in the word we want to check spelling and send the word to web services by http get method as this is just a simple method which pass 1 parameter to the web service.
3. We use angularjs to configure the webserivce for which web method to call and how to send response to the user.
4. On the server side, the algorithm is not created by us but we follow the Damerau-Levenshtein distance algorithm which is found online. The algorithm is for building a double dimensional matrix for comparing two words for computing the edit distance between the two words.

e.g. a b c d e f

0 1 2 3 4 5 6

a 1 0 1 2 3 4 5

& 2 1 1 2 3 4 5

c 3 2 2 1 2 3 4

e 4 3 3 2 2 2 3

d 5 4 4 3 2 3 3

and 3 is the edit distance.

1. On the big picture, for every words, we need go through wordlist for candidates for comparison. Since we set the max edit distance to be 1, we will first compute the edit distance for each word in the wordlist to the input words. If the length difference is equals or more than 2, we just ignore the word as the edit distance is at least 2.

Then for each candidate, we run Damerau-Levenshtein algorithm for calculating the edit distance. If the comparison matrix shows the edit distance equals or is more than 2, we will ignore this candidate and go to next one which all the candidates have been checked and the ones satisfied the requirements are stored in the suggestion list with edit distance, term, count values stored waiting for being sent back to client.

1. When server first received an input from the user, the server will create the wordlist in terms of the text file provided for creation. Then the server will run the algorithm through the wordlist and get suggestions for the user and put the result in the response configured with AngularJs.
2. Then client is able to retrieve the response from the sever with suggestions showed in the client html page.
3. Every time when we receive a request from the client, we will check the size of the wordlist, if it is empty, we will get the text file and build the wordlist, if not we will use the wordlist directly. The advantage is obviously that we need not create or modify the wordlist every time which is time saving. However, we are not able the update the wordlist.
4. Work flow: Client input -> configure with AngularJs -> Send Http request -> Server create wordlist -> run algorithm for suggestions -> fill suggestions in response -> Client get response and show data.