Software Requirements Specifications

for

ELEKTOO

Dated:
17th September 2015

Team Members:

Vinay Kumar Singh
Ayushi Goyal
Aman Rai
Prachi Agrawal

Table of Contents:

1. Document purpose

The purpose of this document is to give a detailed description of the requirements for the “Elektoo” search engine. The document is intended to describe accurately the capabilities that the search engine should provide to its end users and also to specify all the non-functional requirements that the application should implement.

2. Document overview

The remainder of the document has three major sections.

- The first section offers the general description of the search engine along with its benefits.
- The second section lists the functional requirements that the search engine should meet. It contains description of actors and the use cases.
- The final section exposes the non-functional requirements of the application, such as user-interface etc.

3. General description of the product

3.1 The current situation

The current search engines provides its end users with numerous results, amongst which most are irrelevent. Thererfore the users ends up makin wrong choices (choose something that wasn't expected initially). Also, as the users are not very clear about their choices, they may fall in dilemma as to how to shortlist the results as they dont have a specific count of the number of results being displayed. Also the ambiguous queries like “Big screen size” for phones are not filtered appropriately.

3.2 Purpose of the product

The search-cum-choice engine ”ELEKTOO” is intended to enable its users to make an informed and quick search choice about a product ,with the purpose of buying the product online or offline by providing them with narrowed results, making it easier for them to choose within the results. The application also focuses on resolving the “ambiguous qeuries” (as mentioned above) to concrete search results.

3.3 Brief Product Description and Product context

The application deals with designing two kinds of searching query methods:

1. Choice Query (text box)
2. Choice Matrix

The first option to choose from is “Choice query”. The user would type in the query and the corresponding results will get displayed. To filter all the unrelated results shown, the application would try to resolve the ambiguity in the search query and replace it with concrete suggestions provided by Elektoo and narrow down the results to some specific numbers. This “count” of specific numbers will get displayed on the side of the search box so as to benefit the user as in how specific the query needs to be.

If the user is very specific about the product he/she wants to buy, then the second option “Choice Matrix” will be always a better choice as it provides information about the product and its specifications too in a tabular form.

Context:

The query fed in by the user will be search in the database tables corresponding to the product and the appropriate results will be displayed. The users would access the application by using their web browser, so an internet connection is necessary .

3.4 Benefits

This appliation is supposed to satisfy a user's needs regarding a more prominent and appropriate search tool that provides narrowed results and also a more “user-friendly” tool as it also narrows the ambigous query results.

4. Functional Requirements

4.1 Actors

The End-users will be customers who like or wish to shop and purchase items through online retail store.

- There is no constraint on the section of society the customer belongs.
- The user must not be blind or without limbs.
- The user must have a basic knowledge of using any technical device like computer or smart phone.

4.1.1 Users

The user may perform any of the following operations:

- Choose between 'choice search' and 'choice matrix'
- Type in the query in the provided search box prompt.
- Navigate through the search results.
- select/ deselect any tab (for coice matrix).

4.1.2 System

The system stored the database to processthe search query and also stored the queries fed in by the user.

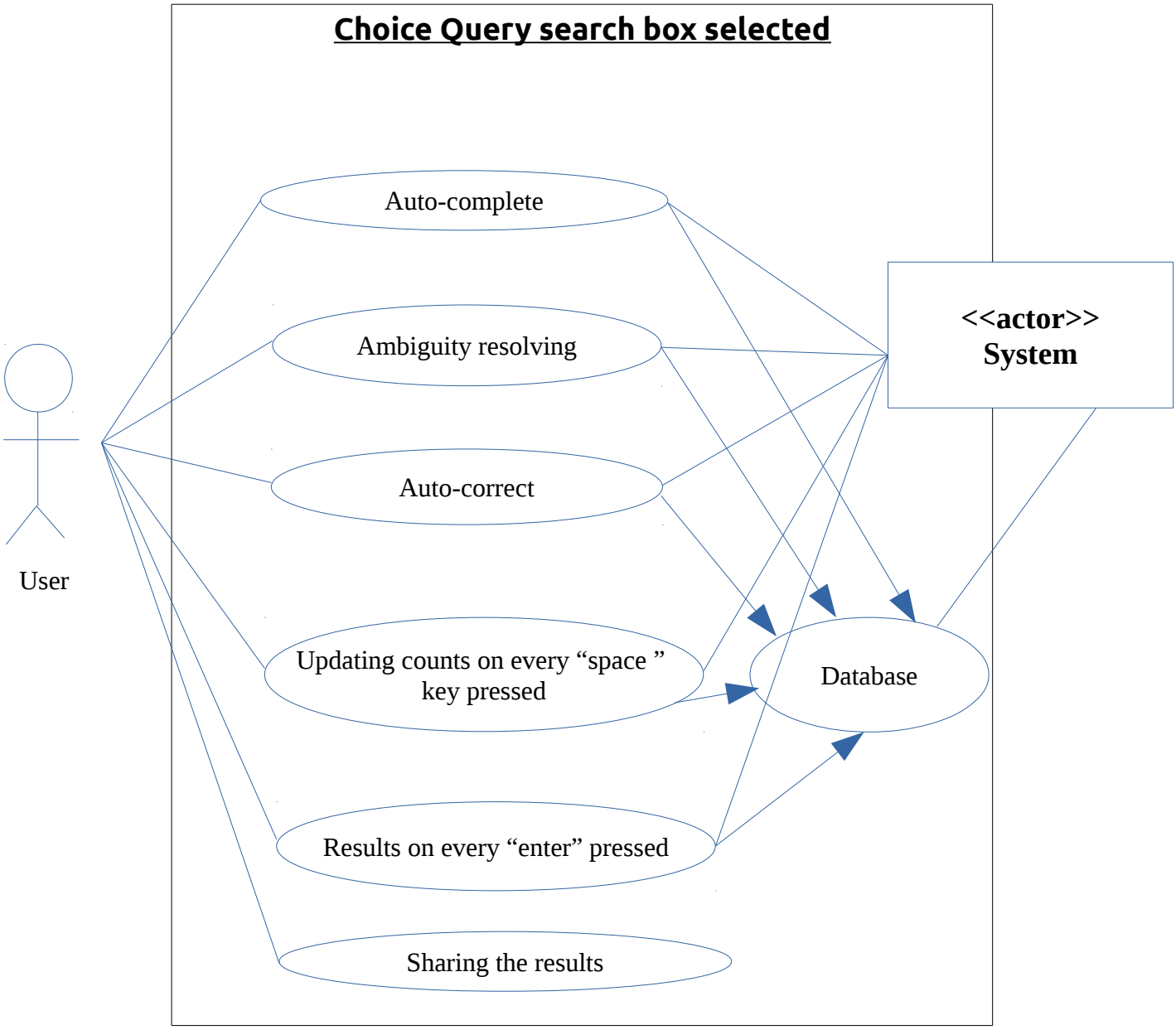
4.2 Use cases diagrams and description

4.2.1. Make a choice

Use Case Number:	UC-01
Use Case Name:	Choosing either “Search Query box” or “Choice Matrix”.
Overview:	It lets the user to choose from one of the respective methods to search and proceed in the respective direction.
Actors:	User,system
Pre-condition:	The user shoul be on the webpage of the application(Elektoo).
Flow:	Main (success) flow: 1. open the webpage of the application. 2. giving two options to search for their query. 3. Choosing one type from the search query type . • The choice query text box. • The choice matrix.
Post-Condition:	Based ont he choice made by the user the application forwards the user to either “Choice Query search” or “Choice Matrix” page to proceed further with his/her requirements.

4.2.1.1 Choice query search box

4.2.1.1.1 Use Case Diagram



4.2.1.1.2 Use case Description:

4.2.1.1.2.1 Choice query box selected

Use Case Number:	UC-02
Use Case Name:	Choice query text box is selected as the method of the searching a query.
Overview:	Allows the user to search for the product by typing a query or phrase describing the product in the search box provided.
Actors:	user
Pre-condition:	The user must have selected the “Choice Query text box” in the kind of search option in the last use case.
Flow:	Main (success) flow: 1. User goes to the query box provided. 2. a search box is provided to enter the required product description. 3. user starts typing in the search box.
Post-condition:	During the input of text or phrases invoking the unique inbuilt functions which make the ELEKTOO a choice engine.

4.2.1.1.2.2 Auto Complete

Use Case Number:	UC-03
Use Case Name:	Auto-complete on every character press.
Overview:	On every character entered by the user in the query box the the auto-complete function suggests list of words that the user can select from.
Actors:	User, System
Pre-condition:	The User maust have selected the Choice Query box.
Flow:	Main (success) Flow: 1. The user is given a prompt for feeding in the input. 2. The user inputs certain characters . 3. On every character input, the list of “suggested” words appears just below the search query box. 4. The “suggested” words are those containing the starting characters as typed in by the user. 5.The user can select any of the suggested word and auto-complete the typed-in word.
	Alternate flow: 5. a) User can ignore the suggested word and continue giving in the input.
Post-condition:	The user must end with a word either suggested by the system or entered by the user.

4.2.1.1.2.3 Resolving Ambiguity

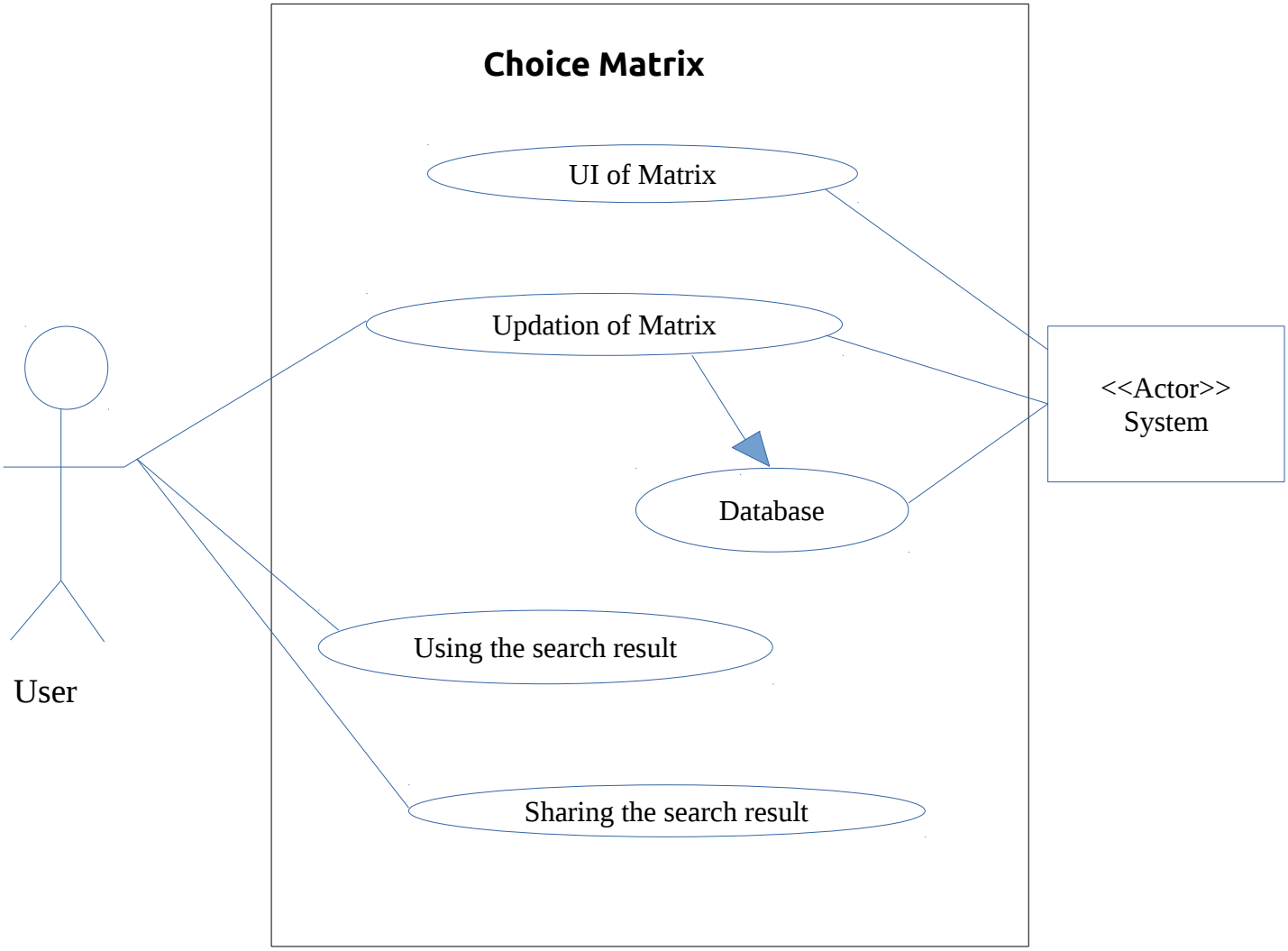
Use Case Number:	UC-04
Use Case Name:	Resolving ambiguity.
Overview:	The ambigouis query phrases are to be replaced by some concrete suggentions provided from the system.
Actors:	User, System
Pre-condition:	User maust have selected the choice query option in the search option, and must have started typing in the search query box.
Flow:	Main (success) Flow: 1. User types a query to search for in the search box. 2. if the phrase seems ambigous the suggestions will be invoked. 3. the main feature of the suggestion is that it replaces the ambigous phrase with a concrete sentence to search for. 4. user selects one of the concrete suggestions provided by the system and move on with entering the query.
	Alternate Flow: 3. if the user does not finds any suggestions useful ask user to define a specific defination of the ambigous phrase he/she used.
Post-condition:	Update the learning table in database, and display the count of results in the box provided for the same.

4.2.1.1.2.4 Auto Correct

Use case number:	UC-05
Use case name:	Auto-correct
Overview:	Correct any misspelled word automatically or give suggestions to the users to use the correct word.
Actors:	User, System
Pre Condition:	User must input a few characters/words in the search query box and press space or enter .
Flow:	Main (success) Flow: 1. The user is given a prompt for feeding in the input. 2. The user inputs certain characters/words . 3. Each time “space” or “enter” is pressed ,and the system inspects some misspelled words, suggestions will be given to the user to correct the incorrect word by displaying a list of suggested words. 4. The “suggested” words are those containing the starting characters as typed in by the user or the grammatically correct word for the word fed as input by the user. 5.The user can select any of the suggested word or else the system will automatically correct the input word by the first suggested word in the list of suggested words.
	Alternate flow: 5. a) User can ignore the suggested word and continue giving in the input.
Post Condition:	The corrected word gets displayed on to the screen.

4.2.1.2 Choice Matrix:

4.2.1.2.1 Use Case Diagram



4.2.1.2.2 Use case Description

4.2.1.2.2.1 UI of choice Matrix

Use Case Number:	06
Use Case Name:	UI of choice matrix
Overview:	Display as shown to the user
Actors:	User
Pre-condition:	User should have selected choice matrix as the option for querying. Also the user must select “product” and “brand” name in prior.
Flow:	Main (success) Flow: 1. Open the webpage. 2. Select choice matrix as the querying interface. 3. Display a 9*6 matrix. 4. Display the options in each row specific to a specification of the product. (eg. Screen Size, Price, OS,etc) 5. User can choose from these options
Post-condition:	The choice matrix is displayed.

4.2.1.2.2.2 Matrix update

Use Case Number:	UC-07
Use Case Name:	Real time updation/disable/enable of the displayed options in the choice matrix
Overview:	It updates/disable/enable the options in the matrix based on the other options selected by the user
Actors:	User , system
Pre condition:	User should have selected choice matrix as the option for querying its choice
Flow:	Main (success) Flow: 1. Open the webpage. 2. Select choice matrix as the querying interface. 3. Select the already enabled choices. 4. System automatically disables/enables the choices based on what the user has already selected. 5. The number of results found are updated in real time 6. This is used to fetch data from database. 7. Clicks the Search button.
	Alternate Flow 1: 3. User deselects the options. Alternate Flow 2: 7. Clicks the Reset button. 8. It will reset the entire choice matrix afresh.
Post Condition:	The resulting data from the database is displayed

4.2.1.3 Usecases common for both choices

*The following use cases consists of description for both choice search query and choice matrix.

4.2.1.3.1 Update result count

Use case number:	UC-08
Use case name:	Updating the number of counts on every space key pressed(search query)/option selected/deselected(choice matrix)
Overview:	While entering the query in the query box each time the space key is pressed the total count of product gets updated.(choice search query) On any changes in the selection in choice query matrix, count of the products gets updated. (choice matrix)
Actors:	User, System
Pre Condition:	User types in the the query(say a word) and then presses the space key. User makes a selection by clicking on a particular feature in choice matrix.
Flow:	Main (success) Flow: 1. The user feeds in a space after every word. 2. The count of the number of products related to the typed in query gets updated and displayed on the right side of the search query box. 3. Viewing the number of counts of the products related to the query, the user can decide whether to see the results by presing enter of continue with typing the query.
	Alternate flow1: 3. a. If the user not presses the enter key , only the count of the number of products gets updated in real time and no result gets displayed. 4. a. If the count of the product is zero the the user is asked to select from the suggestions provided. b. i) The user can select any one of the suggested results or ii) Continue search by giving another input query into the search query box. Alternate flow2: **1.** If the option of querying is choice matrix. **2.** The user selects/deselects the option. **3.** On each selection/deselection the count is updated in real time.
Post Condition:	The count of expected results is updated. And is shown to the user on in the box besides the search box.

4.2.1.3.2 Navigate through results

Use Case Number:	UC-09
Use Case Name:	Using the Search results
Overview:	What user does with the search results displayed
Actors:	User
Pre condition:	The User should have already made search using choice matrix/choice query. The User should be on the search results page.
Flow:	Main (success) Flow: 1. The User has selected the options in choice matrix/queried in choice query. 2. Clicks the Select button. 3. The results get displayed. 4. User checks out the result. 5. User clicks on a particular result to view its full details. 6. Displays the entire data description of the particular product.
	Alternate Flows: 5. User clicks the back button. If the user uses choice matrix to query 6. Goes back to the choice matrix page. 7. Restores the last state of the Choice Matrix after “search button” is clicked. If the user uses uses choice query. 6. Goes back to choice query box.
Post Condition:	User can see the entire details of a particular product.

4.2.1.3.3 Result display

Use Case Number:	UC-10
Use Case Name:	Displaying total results on the press of “Enter” key/Search button.
Overview:	On press of the answer key all the counter results related to the query searched should be displayed on the screen.(Choice query) On clicking the search button, the total number of results found is displayed on the screen.
Actors:	User, System
Pre-condition:	The user must have given a query to search for in the query box.(Choice query) The user must have selected the desired option from the matrix(Choice matrix).
Flow:	Main(success) Flow: 1. Once the user is done with typing the query for the product, and satisfied by the count of results for the query searched. 2. The user presses enter. 3. On press of enter all the counted results will be displayed on the screen. (with pagination*). 4. In the results the user is provided with the expected products. 5. Once the user chooses any of the displayed products, all its descriptions are visible to him/her. *will be described in the pagination use-case.
	Alternate Flow: 1. a. can also continue with typing the query to be more specific and drop down the count to as minimal as possible. Alternate flow2: 1. User can also make choice using the choice matrix.
Post-condition:	All the counted expected results are displayed on the screen(with pagination).

4.2.1.3.4 Result sharing

Use case Number:	UC-11
Use case Name:	Sharing the searched results.
Overview:	Once the user has searched for a query he is provided with an option to share the result via social sites
Actors:	User
Pre-condition:	User must obtain a search result
Flow:	Main (success) Flow: 1. the user must have got a search result of the query he/she had entered. 2. Provided with a share option 3. once the user clicks the share option the URL of the page is made ready to be shared. 4. can share the results among friends or family via social sites like facebook,twitter,gmail,etc.
Post-condition:	All the people to whom the user had shared the results get's a URL of the result page.

4.2.1.3.5 Learning System

Use Case Number:	UC-12
Use Case Name:	Learning System(Updating the database and DB tables)
Overview:	Update the DB with all the ambiguity resolved, all the suggestion selected, all the final query searched.
Actors:	System
Pre-condition:	Either an ambiguity has been resolved, or a suggestion is selected for auto-complete , or a search has been done(i.e. pressed “enter”).
Flow:	Main (success) Flow: 1. one of the above 3 pre-conditions is made. 2. DB tables is updated and ranking of the search query or the suggestion is improved. 3. And the DB table is updated according to the way such that next time the user does not faces that much of problem with choosing the suggestions.
	Alternate Flow1: 1. if no suggestions are availabale for some query then store that entry as the user provides in the DB. 2. If new entry is inserted in the table keep it at the lower rank in the suggestion table. Alternate Flow2: 1. If user makes search using choice matrix. 2. Most frequently selected options are stored in the database.
Post-condition:	The Database is updated as per the selected suggestions from the user or from the search done by the user.

4.2.1.3.6 Pagination

Use Case Number:	UC-13
Use Case Name:	Pagination
Overview:	The results are displayed in form of pages to accomodate large number of results.
Actors:	User
Pre condition:	User should have made search
Flow:	Main (success) Flow: 1. User makes search. 2. Search results are displayed. 3. The results are displayed in form of pages with a fixed number of results being displayed on a particular page.
Post Condition:	Results are displayed in form of pages

5. Non-functional requirements

5.1 User-Interface Requirements

The user interface of the application must be user-friendly, intuitive and easy to use, implementing the ergonomics standards.

6. Feature requirements (as per the use cases):

No.	Use case Number	Description	Release
1.	UC-01	It lets the user to choose from one of the respective methods to search and proceed in the respective direction.	R1
2.	UC-02	Choice query text box is selected as the method of the searching a query.	R1
3.	UC-04	Resolving ambiguity.	R1
4.	UC-05	Auto-correct	R1
5.	UC-07	Displaying total results on the press of “Enter” key.	R1
6.	UC-08	UI of choice matrix	R1
7.	UC-10	Using the Search results	R1
8.	UC-11	Pagination	R1
9.	UC-12	Sharing the searched results.	R1
10.	UC-13	Learning System(Updating the database and DB tables)	R2
11.	UC-03	Auto-complete on every character press.	R2
12.	UC-06	Updating the number of counts on every space key pressed.	R2
13.	UC-09	Real time updation/disable/enable of the displayed options in the choice matrix	R2