Use Case 9: Refine existing query and update results

Description: The user can optimize the query to make it faster or change what the query searches to change the results from the query

Primary Actor

A researcher or programmer interested in a GitHub repository

Stakeholders

The searcher - Needs faster results or more specific results.

The people the searcher is searching on behalf of (ex: their team)

Preconditions

A query must already exist in the database.

Success Guarantee

The speed of the user receives the results is faster. The user receives new results.

Main Success Scenario

1. The user selects filters to show results by (ex: show only commits from the past week or by X contributor)

2. After clicking submit, the system finds all results for the query matching both the keyword and filters.

3. The system displays these results to the user

4. The user refines the searched query.

Extensions

2a. the repository doesn't exist:

1. The user inputs the URL of the desired Github repository

2. The user then inputs a query.

3. After clicking "submit", the system attempts to search the repository, but fails due to the repository not being found.

4. The system generates an error and displays it to the user

3a. the query was not found: An error message should display that the filters don't match any current query.

Special Requirements:

-Results must be returned quickly and there should be a variety of relevant filters

-Filters should not make the screen looked cluttered and difficult for mobile browser users

-A message should display showing the success of the user's actions.

Technology and Data Variation List

1a. Filtering may be done in a few ways, but chiefly either by filtering as it searches or first finding all the textual matches, then filtering

1b. Filtering may be done in JavaScript, PHP, or other applicable languages

2a. Searching may use JavaScript, PHP, or other applicable languages.

2b. Searching may be done for a query string or keywords

Frequency of Occurrence

Not often, just whenever a user wants to refine a query.

Use Case 10: Sort Results

Description: The user is able to filter the results by specific key words.

Primary Actor

A researcher or programmer interested in a GitHub repository

Stakeholders

The searcher – Wants specifics from search results

The people the searcher is searching on behalf of (ex: their team)

Preconditions

The output of the search must have results that match the filters.

Success Guarantee

The user is able to sort the results and is able to find what he or she needs faster due to sorting.

Main Success Scenario

1. The user inputs the URL of the desired GitHub repository
2. The user types in the query
3. After clicking submit, the system finds all results for the query matching the key word
4. The system displays these results to the user
5. The user selects filters to sort the results.

Extensions

3a. the repository doesn't exist:

1. The user inputs the URL of the desired Github repository
2. The user then inputs a query.
3. After clicking "submit", the system attempts to search the repository, but fails due to the repository not being found.
4. The system generates an error and displays it to the user

4b. No results were found:

1 . The user inputs the url of the desired Github repository

2. The user then inputs a query

3. After clicking submit, the system attempts to search the repository

but does not find any results

4. The system generates an error and displays it to the user

4a. the system suggest alternate search filters based on what it did find

Special Requirements:

-Results must be returned quickly and there should be a variety of relevant filters

-Filters should not make the screen looked cluttered and difficult for mobile browser users

-A message should display showing the success of the user's actions.

Technology and Data Variation List

3a. Searching may use JavaScript, PHP, or other applicable languages.

3b. Searching may be done for a query string or keywords

5a. Filtering must be done after the search is over.

5b. Filtering may be done in JavaScript, PHP, or other applicable languages

Frequency of Occurrence

Whenever the user wants to make the search more efficient by sorting the results

Use Case 11: Display/hide search history log file.

Description: The user is able to hide the history log file while typing in a search.

Primary Actor

A researcher or programmer interested in a GitHub repository

Stakeholders

The searcher – Does/doesn't want the past results to show while searching.

Preconditions

The log file is not empty.

Success Guarantee

The log history is displayed how the user desires.

Main Success Scenario

1. The user begins a new search.

2. The history log will start to show while the user is inputting the search.

3. An option to hide the file will appear above the first result in the log file.

4. The log file will disappear when the option is clicked or reappear if it is already hidden.

5. The user completes the search by using one of the past reaches or by searching something new.

Extensions

2a. the log file doesn't exist:

1. No history will show and the search will continue
2. The option to hide the log file will still be present.

Special Requirements:

Technology and Data Variation List

1a. Searching may use JavaScript, PHP, or other applicable languages.

1b. Searching may be done for a query string or keywords

2a. The log file will be help in a table of past results.

Frequency of Occurrence

Very frequently. Every time a search is done the function should apply.

Use Case 12: Clear the log file

Description: The user is able to remove all past searches from the log file.

Primary Actor

A researcher or programmer interested in a GitHub repository

Stakeholders

The searcher – Wants to remove all previous searches from the history log file.

Preconditions

The log file is not empty.

Success Guarantee

The log history will be empty and the data will not be retrievable after clearing the file.

Main Success Scenario

1. The user begins a new search.

2. The history log will start to show while the user is inputting the search.

3. An option to clear the log file will appear below all results from the file.

4. The option to clear the file is selected and the file is cleared.

Extensions

2a. the log file is empty:

1. No history will show and the search will continue
2. The option to hide the log file will still be present.

3a. A message to confirm the users request will appear before clearing the file. This way, the user is less likely to accidentally clear the history.

Special Requirements:

Technology and Data Variation List

1a. Searching may use JavaScript, PHP, or other applicable languages.

1b. Searching may be done for a query string or keywords

2a. The log file will be help in a table of past results.

Frequency of Occurrence

Whenever the user wants to clear the history in the log file.