Code Search Engine - Use Cases

Robert Wenger, Arman Bahraini, Marcus Stewart

1. **Search a github repository**: User searches for something in a user-specified repository either by textual query or keyword search.

    Primary Actor

    A researcher or programmer interested in a GitHub repository

    Stakeholders

    The searcher - Needs accurate, fast results

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

    Preconditions

    The user has the url of a github repository of interest.

    Success Guarantee

    Results are returned to the user. All appropriate instances matching the search query are returned.

    Main Success Scenario

    1. The user inputs the url of the desired github repository.
    2. Then the user inputs a query.
    3. After clicking submit, the system searches the given repository for the given text and compiles statistics, graphs, and other results relevant to the query.
    4. The system then displays these results to the user.

    Extensions

    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

    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.

    Special Requirements

    -Results must be returned quickly.

    Technology and Data Variation List

    -Searching may use JavaScript, PHP, or other applicable languages.

    -Searching may be done for a query string or keywords

    Frequency of Occurrence

    Depends on implementation (does our server do the searching work? Or is the user’s browser able to do everything?)

    Ideally: as often as the user likes.

2. Export results to a file: The user exports results from a previously completed search.

Primary Actor

The user who has completed a successful query

Stakeholders

The primary actor - wants to share results of their search

The recipient of the file - needs to review the data shared

Preconditions

The primary actor has completed a successful search (results were returned in their browser)

Success Guarantee

A file with results of the query has been downloaded to the user's computer.

Main Success Scenario

1. The user clicks the export button.
2. The server compiles a file with the results.
3. The server initiates a download.
4. The user uses the save dialogue box to specify a save location and filename.
5. The file is downloaded to the user's computer.

Extensions

The user cancels the download:

1. The user clicks the export button.
2. The server compiles a file with results.
3. The server initiates a download.
4. The user cancels the download via the save dialogue box.

Special Requirements

-The download should have a small filesize.

Technology and Data Variation List

-The document may be in text format, JSON, CSV, or another useful format.

-The document may be saved to the server or generated on the fly.

Frequency of Occurrence

Depends on implementation (does our server save the file?)

Ideally: as often as the user searches.