SE 4485: Software Engineering Projects

Spring 2024

**Requirement Documentation**

|  |  |
| --- | --- |
| Group Number | 10 |
| Project Title | Internet Research Assistant |
| Sponsoring Company | The Fellows Consulting Group |
| Sponsor | Jeff |
| Students | * Bakr Alkayali * Chloe Pascual * Vi Le * Ikraam Rahman * Mohammad Chauhan |

ABSTRACT

* This Requirement Documentation describes the functional and non-functional requisites for the development of a search tool designed to enhance user experience through intelligent query processing and results display. Incorporating a comprehensive suite of use cases, including search query entry, validation, result display, and error handling, this document serves as a blueprint for the system’s architecture. Emphasizing usability and precision, the outlined requirements cater to a broad spectrum of user interactions, ensuring a seamless and intuitive search process. Through detailed textual and graphical models, alongside a clear rationale for chosen methodologies, this document aims to guide the project from conceptualization to realization, ensuring alignment with project goals and user expectations.

# TABLE OF CONTENTS

[LIST OF FIGURES 1](#_Toc1586173988)

[LIST OF TABLES 1](#_Toc593252192)

[INTRODUCTION 1](#_Toc1821037890)

[TEXTUAL DESCRIPTION: 2](#_Toc1834291941)

[RATIONALE FOR YOUR USE CASE MODEL 2](#_Toc109003628)

[USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS 3](#_Toc1361218985)

[Network of Models for Reusable Use Cases 6](#_Toc1264519305)

[NON-FUNCTIONAL REQUIREMENTS 7](#_Toc944448658)

[EVIDENCE THE DOCUMENT BEEN PLACED UNDER CONFIGURATION MANAGEMENT 8](#_Toc256849383)

[ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS 8](#_Toc1172771516)

[ADDITIONAL REFERENCES 8](#_Toc127893799)

# LIST OF FIGURES

Figure 1 ...................................................................................................................................................... 2

Figure 2 ...................................................................................................................................................... 8

# LIST OF TABLES

There are no tables needed.

# INTRODUCTION

This document outlines the requirements for developing a search tool to improve query processing and display results of said search query. It aims to detail the necessary functional and non-functional specifications to guide the tool's design and implementation ensuring an intuitive and efficient user experience.

Purpose and Scope

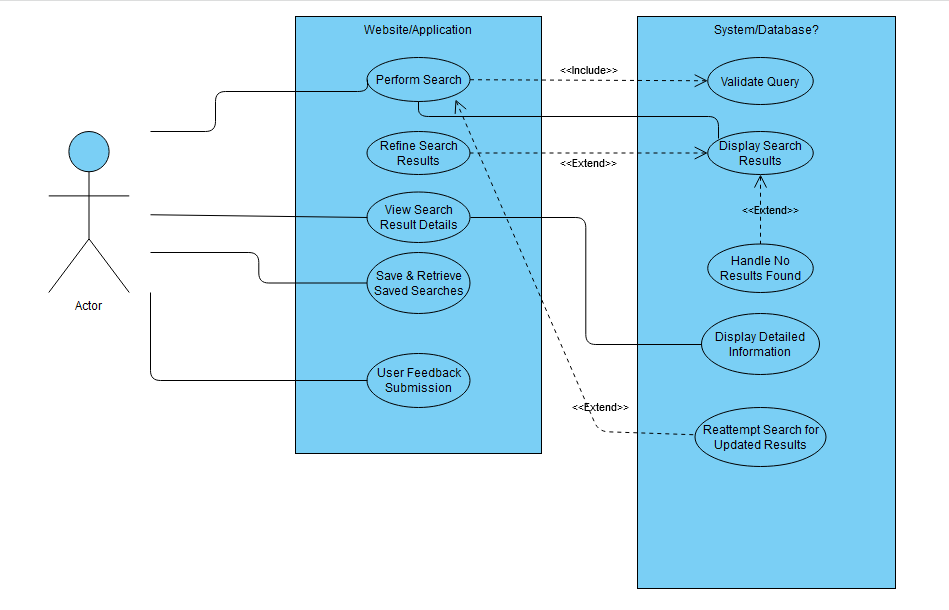
The purpose of this document is to define the search tool's development requirements, covering design, implementation, and user interaction guidelines. Moreover, it details project stakeholders and outlines the project's goals and boundaries.

Document Structure

The document is organized to provide clarity on different facts of the project including sections on the project overview, use cases, and system requirements for comprehensive understanding of project goals and progress metrics.

**Figure 1**

*Overall High-level use case Graphic Model.*



## TEXTUAL DESCRIPTION:

* The use case model visually represents the interactions between the users and the search system of the website/application. It consists of the following main functionalities:
  + **Perform Search**: This is the primary action where a user initiates a search by entering a query into the system.
    - Sub Use Case: Validate Query ensures that the input from the user meets the system’s requirements for a valid search term.
  + **Refine Search Results**: Post-search, users can refine the results they receive, filtering them according to various criteria for more precise information.
  + **View Search Result Details**: Users can select a search result to view more detailed information about the item.
  + **Save & Retrieve Saved Searches**: This functionality allows the user to save their searches and access them later.
  + **User Feedback Submission**: Users can submit feedback about the system, which is independent of the search process itself.
* The System’s backend functionalities include:
  + **Display Search Results**: Upon a successful search query, the system processes and displays the results to the user.
  + **Handle No Results Found**: If no results are related to the query, the system handles this event, typically by informing the user.
  + **Display Detailed Information**: Further information about a selected search result is provided to the user upon request.
  + Reattempt Search for Updated Results: Users have the option to re-execute a search to get updated results.
* Note: Each use case is linked to the next logical step in the process with solid lines indicating direct relationships, while dashed lines with extend indicate optional paths that users may take.

# RATIONALE FOR YOUR USE CASE MODEL

* The rationale behind this use case model is to provide a clear, high-level view of the system’s functionality from the user’s perspective. It is designed to:
  + **Highlight Key Interactions:** The model emphasizes the core functionalities that users will interact with, such as searching, refining results, and viewing details, ensuring that the system’s primary purpose is well understood.
  + **Allow for Scalability**: By separating website and system/database functionalities, the model allows for the system to be scaled and evolved over time, with new features potentially being added as extensions to existing use cases.
  + **Enhance Communication**: It provides a visual tool that can be used to communicate the system’s functionality to stakeholders, developers, and users in a straightforward and accessible manner.
  + **Streamline Testing and Validation**: The model serves as a guide for creating test cases, ensuring that all user pathways are covered and work as intended.

# USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS

**High-Level Use Case**

***1.*** ***Perform Search***

\*Sub Use Cases:

1.1 Enter Search Query

* **Use Case Name -** Enter Search Query
* **Participating Actors -** User, Database
* **Entry Condition(s) -** 
  + User wants to make a search
  + User clicks on search bar
* **Normal Flow of Events**
  + User enters their search query
  + System processes query keywords
  + System narrows down search results related to query
  + Search results are displayed to user
* **Exit Condition(s)**
  + User clicks off of search bar
* **Exceptions (Alternate Flow of Events)**
  + User clicks on search bar
  + User clicks off search bar
* **Special Requirements**
  + None

***2. Validate Query***

* **Use Case Name -** Validate Query
* **Participating Actors -** User, Database
* **Entry Condition(s) -** 
  + A search query has been entered and submitted by user
* **Normal Flow of Events**
  + Keywords are evaluated by system
  + Relevant pages indexed in database are brought up
* **Exit Condition(s)**
  + Query is validated and results are found
* **Exceptions (Alternate Flow of Events)**
  + Keywords are evaluated by system
  + There is a typo in keywords
  + System suggests replacement for typo in query
  + User selects updated query
  + Relevant pages indexed in database are brought up
* **Special Requirements**
  + Display loading icon or message

***3. Display Search Results***

* **Use Case Name -** Display search results
* **Participating Actors -** User, Database
* **Entry Condition(s) -** 
  + Indexed pages related to search query are found
* **Normal Flow of Events**
  + Indexed pages relevant to query are found
  + Summaries of relevant information on top pages are formed
  + Summaries of information are displayed to user
* **Exit Condition(s)**
  + Information is displayed to user
* **Exceptions (Alternate Flow of Events)**
  + No indexed pages relevant to query are found
  + Error message is displayed to user
* **Special Requirements**
  + Results are displayed in a timely manner

***4. Handle No Results Found***

* **Use Case Name -** Handle no results found
* **Participating Actors -** User, Database
* **Entry Condition(s) -** 
  + No indexed pages related to search query are found
* **Normal Flow of Events**
  + No indexed pages relevant to query are found
  + Display error message to user
  + (Optional): display possible relevant and/or popular results
* **Exit Condition(s)**
  + Error message is displayed to user
* **Exceptions (Alternate Flow of Events)**
* **Special Requirements**
  + Reusable Use Cases:
    - Validate Query
    - Display Error Message

***5****.* ***Refine Search Results-***

\*Sub Use Cases:

2.1 Apply Filter

* **Use Case Name -** Apply filter
* **Participating Actors -** User, Database
* **Entry Conditions -** 
  + User filters search to narrow results
* **Normal Flow of Events -**
  + User enters search query what they’d like to search
  + System will display the search results
  + System will allow the option for the user to filter search results
* **Exit Conditions -**
  + Exits once the system has displayed search results
* **Exceptions -**
  + User decides to search again
  + No search results are found
    - Gives error message

***6. View Search Result Details-***

\*Sub Use Cases:

3.1 Select Search Result

* **Use Case Name -** Select Search Result
* **Participating Actors -** User, Database
* **Entry Conditions -** 
  + User wants to make a search
  + User enters their search into the search bar
  + System displays information regarding the search
* **Normal Flow of Events -**
  + User wants to make a search
  + User enters their search into the search bar
  + System displays information regarding the search
  + System then gives the user relevant topics that correlate to the search
* **Exit Conditions -**
  + The system has displayed the search results
* **Exceptions -**
  + If the user decides to search another topic
  + If the user decides to click on the relevant topics that the system has provided

***7. Display Detailed Information -***

* **Use Case Name -** Display Detailed Information
* **Participating Actors -** Database
* **Entry Conditions -** 
  + User enters search into search bar
  + System displays search results to the user
  + System orders information from most relevant to least
* **Normal Flow of Events -**
  + User enters search into search bar
  + System displays search results to the user
  + System orders information from most relevant to least
  + System displays other topics that can provide more information to the user about the topic
* **Exit Conditions -**
  + The system displays the results of the search query to the user
  + No search results are found
    - Gives error message
* **Exceptions -**
  + No search results found
    - Gives error message to user
  + User continues searching
* **Special Requirements -**
  + Reusable Use Cases:
    - Log User Interaction (for improving search relevance)
    - Display Other Relevant Information
    - Display error message

***8. Save and Retrieve Saved Searches\**** Optional Implementation

● Use Case Name - Save Search Query

● Participating Actors: User, Database

● Entry Conditions:

○ Users want to save their search for future reference.

● Normal Flow of Events:

○ User saves a search query.

○ System confirms the search has been saved.

● Exit Conditions:

○ Users can access saved searches later.

● Exceptions:

○ Issues with saving due to system error or connectivity issues.

● Special Requirements:

○ Provide a user-friendly interface for managing saved searches.

***9. Reattempt Search for Updated Results\**** Optional Implementation -maybe take out at end

● Use Case Name: Reattempt Search for Updated Results

* Participating Actors: User, Database
* Entry Conditions:
  + User has previously executed a search and wants to re-execute it for a potentially better response/answer.
* Normal Flow of Events:
  + User selects a previously executed search query to reattempt it.
  + System processes the search query as a new request.
  + System displays the search results updated since the last search was made.
* Exit Conditions:
  + User views the updated search results.
* Exceptions:
  + There are no new or better results since the last search then the system informs the user that there is no update to the former answer.
* Special Requirements:

***10. User Feedback Submission***

* Use Case Name: User Feedback Submission
* Participating Actors: User, System
* Entry Condition:

The user has feedback to submit.

* Normal Flow if Events:

1. The user initiates the feedback by clicking a button.
2. System prompts the user to enter their feedback.
3. The user enters their feedback to the chat interface.
4. System confirms receipt of the feedback and thanks the user.
5. System sends feedback to database for review and action.

* Exit Conditions:
  + The user’s feedback has been successfully submitted and acknowledged by the system.
* Exceptions (Alternate Flow of Events):
  + The user cancels the feedback submission process before completing it. In this case, the system acknowledges the cancellation and returns to the Home interface.
  + The system fails to submit the feedback due to a system error. The system informs the user about the error and asks them to try again later.
* Special Requirements:
  + The submission process should be simple and straightforward to ensure users can easily submit their feedback.

# Network of Models for Reusable Use Cases

1. Validate Query

Ensures the search query meets the system's requirements for processing.

2. Display Search Results

Shows the user the results of their search in a structured format.

3. Display Error Message

Informs the user of any errors that occur during their interaction with the system.

4. Display Confirmation Message

Confirms to the user that an action has been successfully completed.

5. Validate User Authentication

Checks if the user is logged in for actions that require authentication.

6. Display Other Relevant Information

Show user relevant information about search query

7. Autocorrect wrong words for user

Show search results for words spelt slightly wrong

8. Display Past Search Results

Shows the user a drop-down menu of past search queries

# NON-FUNCTIONAL REQUIREMENTS

* Usability – User Interface:

Requirement: The user interface should be designed for maximum usability, focusing on simplicity and intuitiveness.

Design Principles: Prioritize user-centric design principles, minimizing the learning curve for users.

Target: Achieve a usability score of 90 or above in user feedback surveys, indicating a highly user-friendly interface.

Scenario: Users, especially business professionals, should find the tool's interface easy to navigate, enhancing their overall experience.

* Usability - History and Navigation:

Requirement: The tool should maintain a comprehensive search history, allowing users to easily navigate back and forth through their research stages.

Storage: Store a record of user searches, including queries, results, and timestamps.

User Accessibility: Provide an intuitive navigation feature for users to review and revisit their search history.

Privacy: Implement mechanisms to ensure the privacy and security of stored search history data.

Target: Users should be able to seamlessly navigate through their research history, improving the overall user experience and facilitating continuity in their investigative processes.

* Performance - Response Time:

Requirement: The tool should provide quick responses to user queries, minimizing waiting times.

Measurement Metric: Response time is the duration between user input and the presentation of relevant information.

Target: Maintain an average response time about 2500-3000 milliseconds, ensuring prompt retrieval of research results.

* Performance - Scalability:

Requirement: Ensure the tool is scalable to accommodate a growing user base and increasing research demands.

Measurement Metric: System performance under increasing user loads.

Target: The tool should maintain consistent response times even when user loads increase by 20% above the expected peak load.

* Portability:

Requirement: The tool should operate seamlessly within standard web browsers to ensure accessibility across various platforms.

Target Platform: Web browsers such as Chrome, Firefox, Safari, and Edge.

Scenario: Users should be able to access the Internet Research Assistant tool without the need for platform-specific installations, promoting flexibility in usage.

* Reliability:

Requirement: The system must consistently behave in a user-acceptable manner within the intended environment.

Definition: The tool should reliably generate accurate and relevant information in response to user queries.

Scenario: Users depend on the system to provide reliable research results, ensuring the tool's trustworthiness in a business context.

* Efficiency:

Requirement: The tool should efficiently utilize computational resources, including CPU cycles, memory, and disk space.

Measurement Metric: Monitor and optimize resource usage to maintain efficiency.

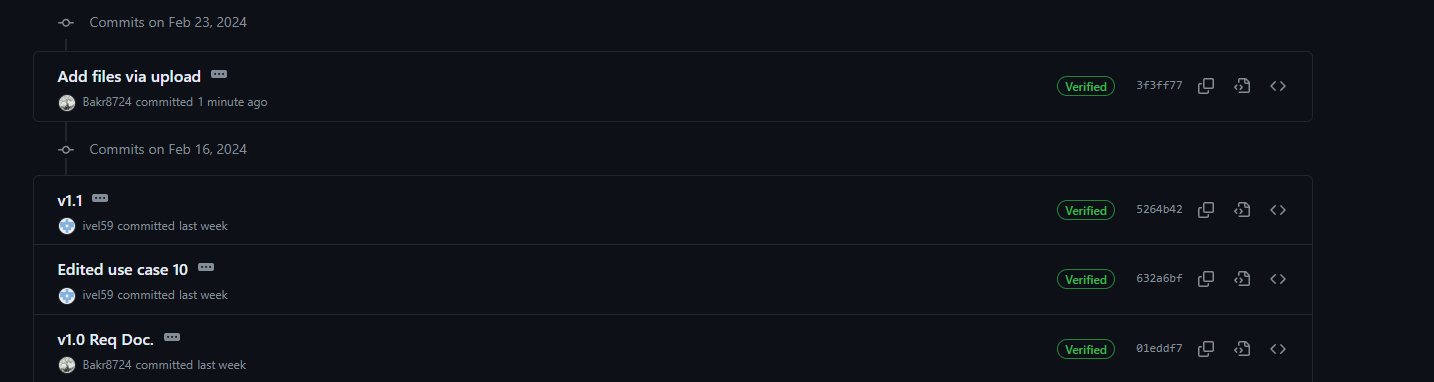
Target: Keep CPU and memory utilization below 70% on average to ensure the tool runs smoothly without excessive resource consumption.

* Accessibility:

Users should be able to access the system using a standard web browser.

No additional hardware or software installation is required.

# EVIDENCE THE DOCUMENT BEEN PLACED UNDER CONFIGURATION MANAGEMENT

1. Name of the CM tool: GitHub
2. Version number of before: [5264b42](https://github.com/Bakr8724/CapstoneSearchTool/commit/5264b422ec8b1fb02027f4ab2e860e9de74ea26f)
3. Version number after: [3f3ff77](https://github.com/Bakr8724/CapstoneSearchTool/commit/3f3ff772a48b60a8478df3c465e6cb6d241db68f)
4. Difference between the two: Added Graphic Model, Refined Use-Cases, Added Standards & References, added Abstract and Introduction, andd Graphic Model Rationale, added textural description.
5. Review of each change:
6. Other info: 

# ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS

* IEEE Std 830-1998: Software Requirements Specification.
* IEEE Std 29148: Requirements Engineering
* ISO/IEC/IEEE Std 29148-2018: Systems and Software Engineering
  + Life Cycle Processes
  + Requirement Engineering

# ADDITIONAL REFERENCES

* Lamsweerde, A.V, 2009, Requirements Engineering: From System Goals to UML Models to Software Specification