

Phase I Project Deliverable

Areebah Fatima
AXF190025@utdallas.edu

Tyler Hargreaves
TTH150630@utdallas.edu

Darrien Kramer
DLK210000@utdallas.edu

Ilhaam Syed
IXS180013@utdallas.edu

Nathan Heindl
NJH180002@utdallas.edu

Matthew Bedford
MDB190007@utdallas.edu

Abstract— The paper describes the implementation of a simple web search engine. The project uses an Object-Oriented Analysis and Design. A prototype is built using an Object-Oriented Program. The scope of the paper is Phase I of the project.

Keywords—component, index

I. INTRODUCTION

The project's end goal is to create a system that will provide users with relevant information using the keywords entered by the user. The resulting data will be a sorted and filtered list of web page URLs to the end user. The major components of the search engine will include a search interface, indexing system, result filtering, query processing, etc. The search interface will allow the end user to interact with the system, enter search queries, and retrieve information. The indexing system will organize, store, and rank information to allow for a reasonably fast search. Result filtering will ensure that our system provides users with relevant, up-to-date data in sorted order. Finally, the query processing component of the project will be responsible for interpreting user inputs, identifying keywords in said input, performing index lookup, and retrieving results.

This document is the final project report. This document aims to explain our project's Unified Process and highlight all significant project changes. It includes Use Case Diagrams, Sequence Diagrams, and Class Diagrams for our Comet Crawler Web Search Engine.

A. Scope

- This project strictly focuses on the Comet Crawler Search engine and its additional functions.
- The system comprises a search engine including a search interface, indexing system, result filtering, query processing, etc.
- The system does not have privileges for specific functionalities. Therefore, search users can perform searches, add URL entries, modify the index, delete URL entries, and manage system settings.
- The system allows multiple users to search at the same time.

B. Assumptions

The following assumptions constraints the Comet Crawler Search Engine:

- Because we are implementing a long-term-supported version of Ruby on Rails, we will be operating under the presumption that HTTP dispatching, database access, and other core functionality is handled appropriately. This design philosophy allows us to consider engineering

the search algorithm and other important application-specific features.

- The application is dependent on Rails and its dependencies. A complete list of requirements can be found at: <https://github.com/tyharg/CS-4376/blob/main/Gemfile>.

C. Project Unified Process (UP)/ Roadmap

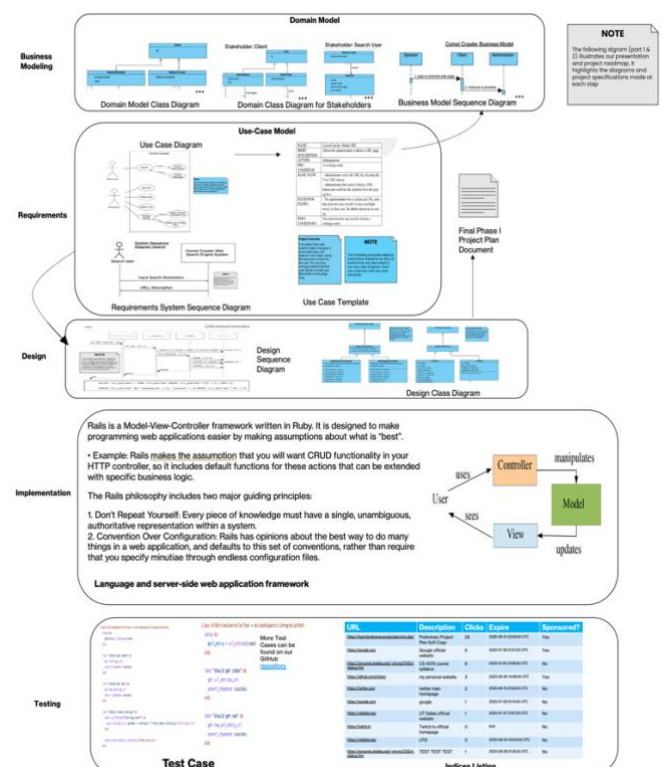


Fig. 1. The Comet Crawler System's Unified Process.

II. MAIN BODY

A. *Domain Modeling & Requirements Specifications*

A.1 Functional Requirements

1F. **Case-sensitive Search:** The system shall store and retrieve the input exactly as provided, preserving case sensitivity.

2F. **Hyperlink enforcement:** When a user clicks on a URL obtained from a query result, the system shall redirect the user to the corresponding website.

3F. **Specifying OR/AND/NOT Search:** The system shall allow users to specify the search mode (OR, AND, or NOT) for keyword-based searches.

4F. **Multiple search engines:** The system shall support the concurrent execution of multiple search engines.

5F. **Purging of out-of-date URLs and descriptions:** The system shall provide the functionality to delete outdated URLs and their corresponding descriptions from the database.

6F. **Query result listing options:** The system shall offer the ability to list query results in ascending alphabetical order, most frequently accessed order, or based on payment preferences.

7F. **Customizable result display and navigation:** The system shall allow users to set the number of results displayed per page and provide navigation functionality between pages.

8F. **Autofill with error correction:** The system shall automatically provide autofill suggestions while correcting typographical errors.

9F. **Symbol filtering based on user configuration:** The system shall filter out symbols that are not considered meaningful, according to user-configured settings.

10F. **Sponsored Web Entries with Priority Display:** The system shall allow web entries to be designated as sponsored if an advertiser has made a payment for promotional purposes. Sponsored URLs shall be given priority in the display of query results, ensuring they are shown prominently before non-sponsored entries.

11F. **Search Query Result List:** The system shall provide users with the ability to input a search query, and upon submission, generate and display a list of URL entries as the output, based on the search query.

12F. **Parsing and Filtering of User Input Queries:** The system shall parse user input queries to extract relevant keywords while ignoring specified filter characters. The extracted keywords shall be utilized to generate the resulting output list of URL entries.

13F. **URL Entry Management:** The system shall provide specific users with the ability to add, modify, delete, and promote URL entries. These privileged users shall have the necessary access and permissions to perform these actions on the URL entries within the system. This privileged user functionality must be implemented during the second iteration of the project.

A.2 Non-Functional Requirements

1N. **Performance:** The Comet Crawler system shall provide fast and efficient search results, ensuring minimal response times even when handling many concurrent user requests.

2N. **Compatibility and Portability:** The Comet Crawler system shall be compatible with commonly used web browsers, ensuring proper functionality and consistent user experience across different browser versions. It should adhere to web standards and compatibility guidelines.

3N. **Usability:** The Comet Crawler system shall have a user-friendly interface, providing a seamless and intuitive search experience for users. It should be accessible across different devices and platforms.

4N. **Error Handling:** The Comet Crawler system shall implement comprehensive error-handling mechanisms to handle exceptions and provide informative error messages to users.

5N. **Reliability:** The Comet Crawler shall be highly reliable, providing consistent and uninterrupted search functionality to users. It should minimize system failures and errors, ensuring reliable access to search capabilities and delivering accurate results.

A.3 Comet Crawler Stakeholders

- **Administrator:** These are privileged users with the ability to add, delete, and modify URL entries. In addition to this, they also manage the system filtering and purging settings.
- **Client:** Are the ultimate owner of the project. This ownership allows them to specify project requirements, making them the recipients of project deliverables and the resulting product.
- **Sponsors:** These are the individuals that pay money to promote their websites to the top of the search result list. Stakeholders are the central piece of the business model as they bring monetary incentives for clients.
- **Search Users:** These are the individuals who interact with the Comet Crawler System. They will provide the search parameters and view the associated results. Sponsors will target these users.

A.4 Domain Model Diagrams

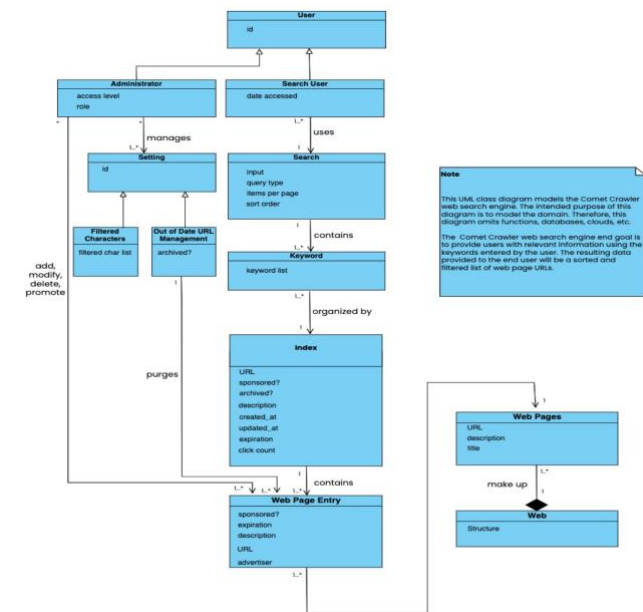


Fig. 2. Class Diagram for the Domain Model.

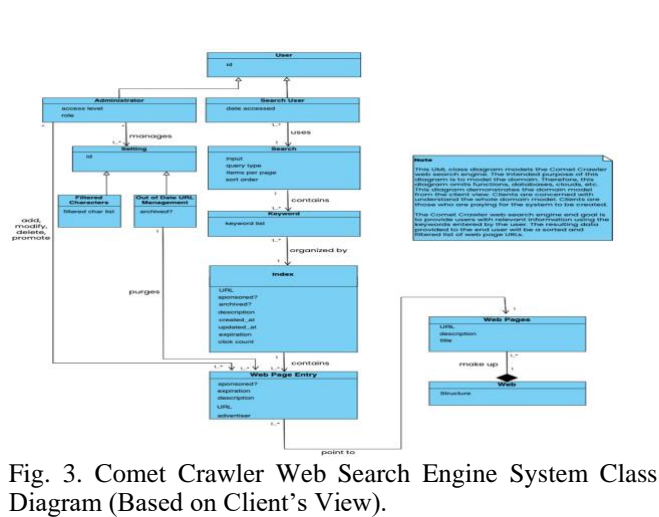


Fig. 3. Comet Crawler Web Search Engine System Class Diagram (Based on Client's View).

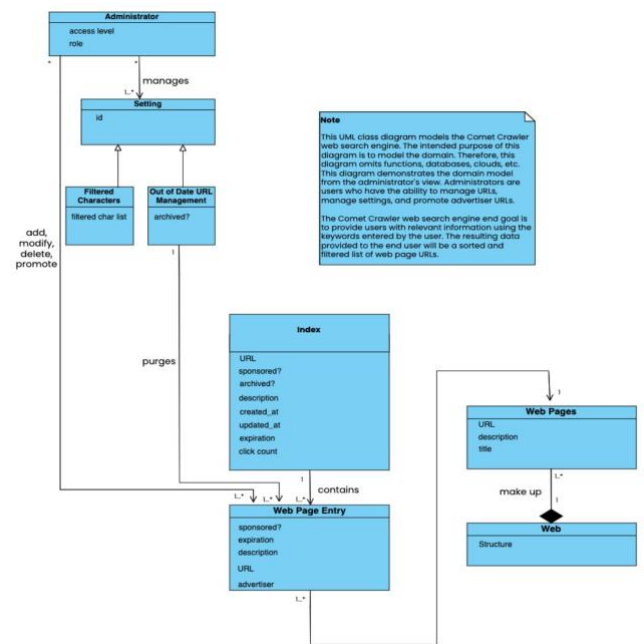


Fig. 4. Comet Crawler Web Search Engine System Class Diagram (Based on Administrator's View).

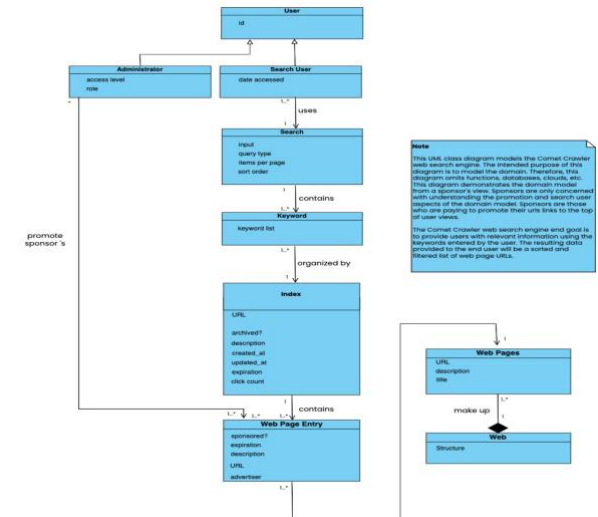


Fig. 5. Comet Crawler Web Search Engine System Class Diagram (Based on Sponsor's View).

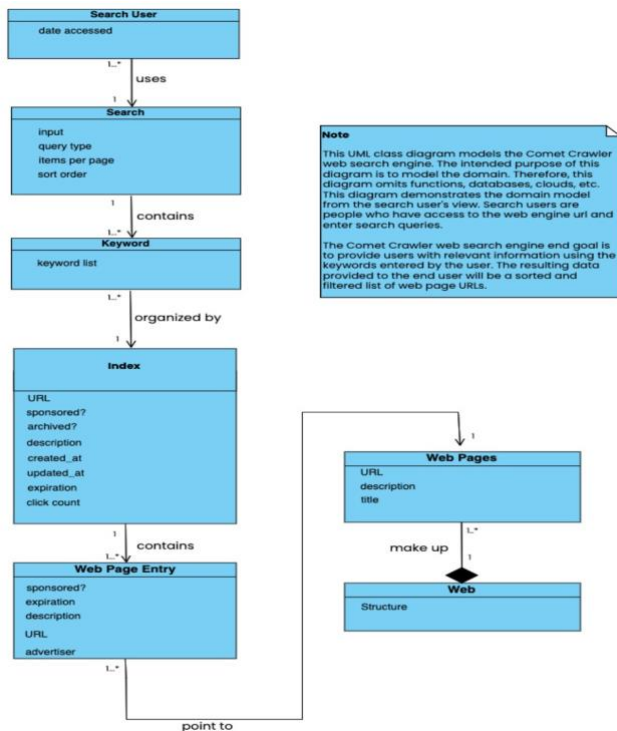


Fig. 6. Comet Crawler Web Search Engine System Class Diagram (Based on Search User's View).

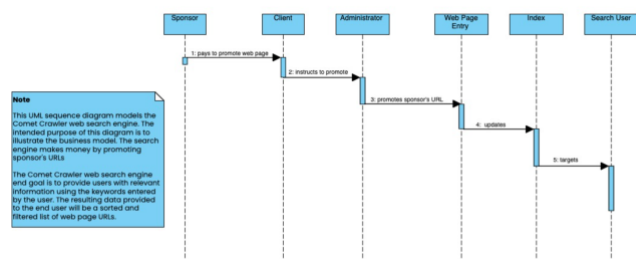


Fig. 7. Sequence Diagram to Model the Comet Crawler Web Search Engine's Business Model.

A.4 Requirements Specification Diagrams

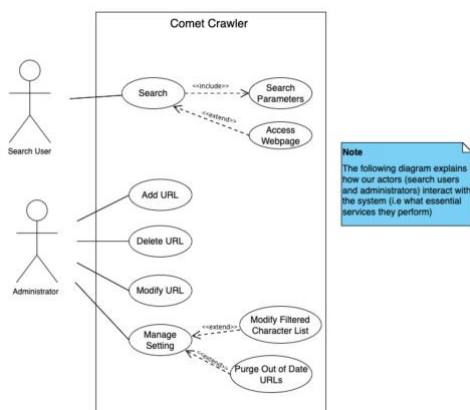


Fig. 8. Comet Crawler Web Search Engine System Use case diagram.

The Search User actors will enter search parameters to perform a search and have the option to click a URL from their search results list to redirect themselves to a different webpage. Administrator actors will add URLs, delete URLs, modify URLs, and manage system settings. It is assumed they have logged in before performing any such activities [1].

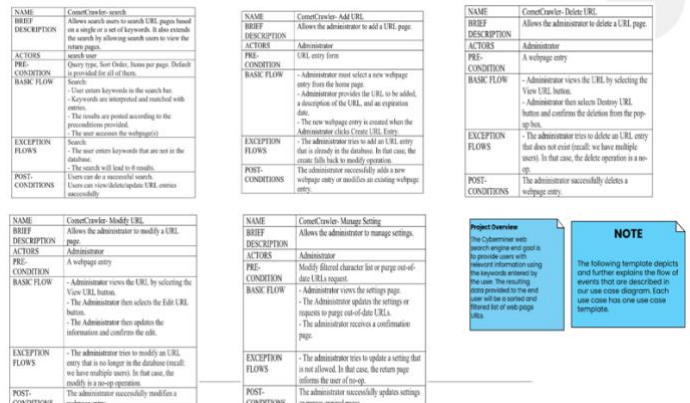


Fig. 9. Comet Crawler Web Search Engine System Use case templates.

A sequence diagram models the sequence of interactions (as time flows downward) to help understand and communicate how various system components cooperate to bring a use case functionality into fruition.

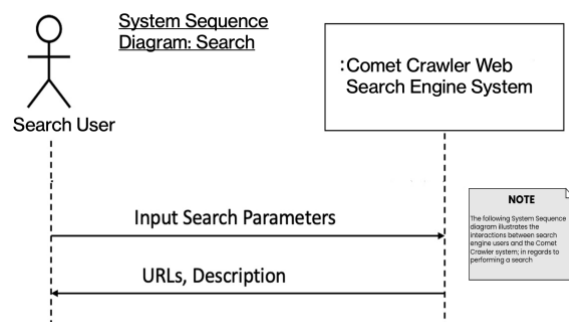


Fig. 10. Comet Crawler Web Search Engine System Sequence diagram for Search.

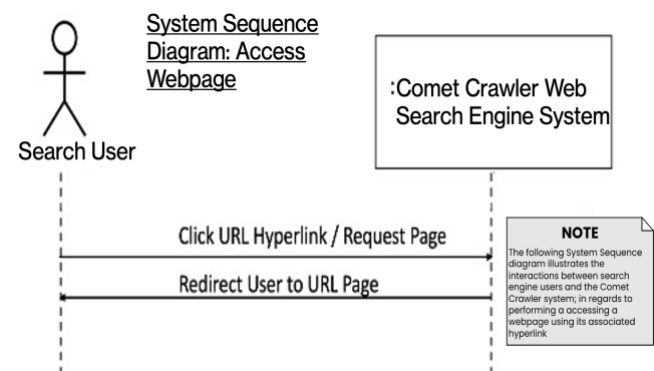


Fig. 10. Comet Crawler Web Search Engine System Sequence diagram for Accessing Web pages.

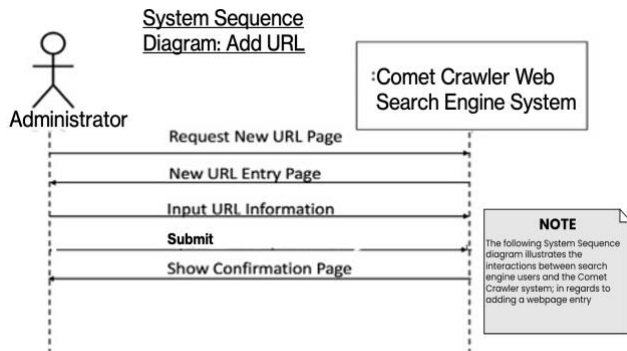


Fig. 11. Comet Crawler Web Search Engine System Sequence diagram for Adding URLs.

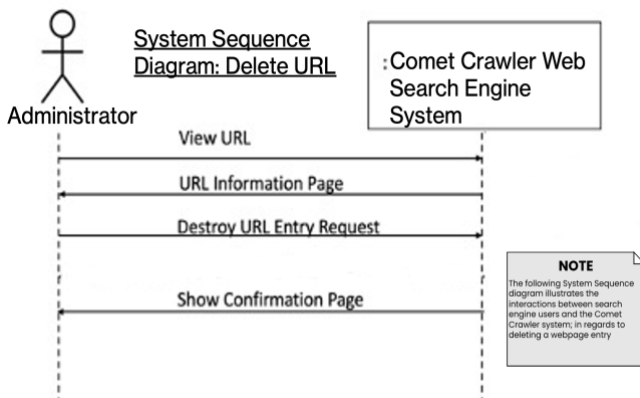


Fig. 12. Comet Crawler Web Search Engine System Sequence diagram for Deleting URLs.

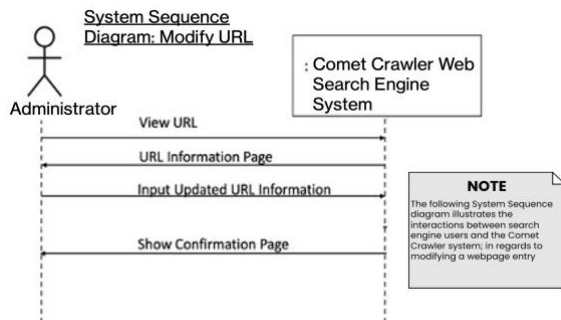


Fig. 13. Comet Crawler Web Search Engine System Sequence diagram for Modifying URLs.

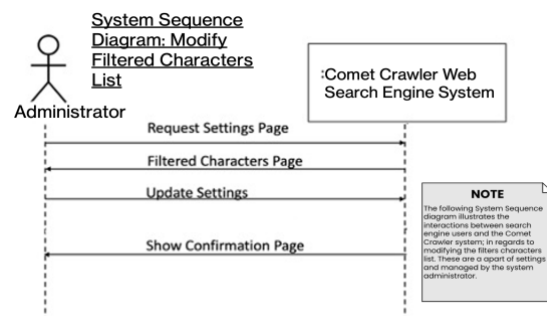


Fig. 14. Comet Crawler Web Search Engine System Sequence diagram for Modifying the System Setting's Filtered Characters List.

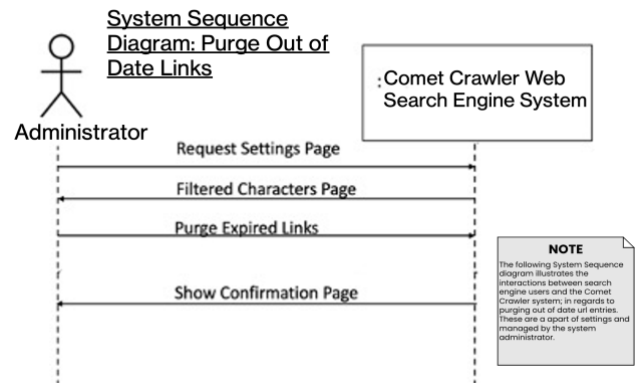


Fig. 15. Comet Crawler Web Search Engine System Sequence diagram for Purging Out of Date URLs.

B. Design Specifications

B.1 Design Specification Diagrams

A class diagram provides a static view of the system's structure. It highlights key concepts from the problem and solution domain to better understand the structure of classes and how they interact. A class diagram shows classes, class attributes, class methods, and relationships between classes.

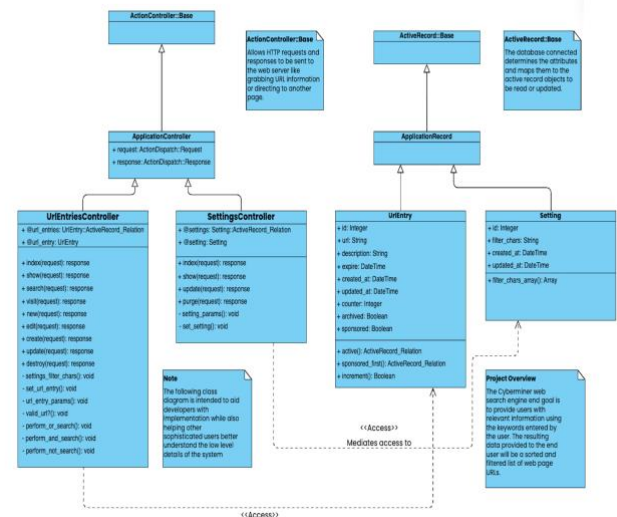


Fig. 16. Comet Crawler Web Search Engine System's Design Class Diagram.

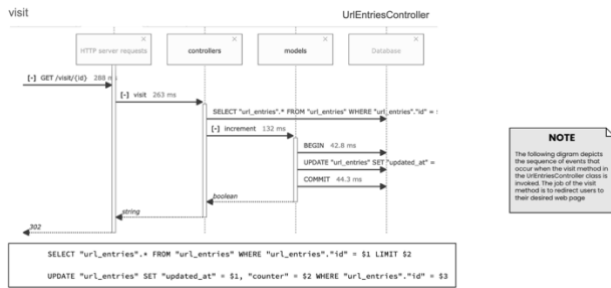


Fig. 17. Comet Crawler Web Search Engine Sequence diagram for visit function.

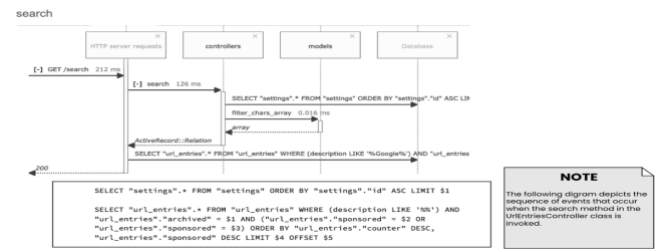


Fig. 21. Comet Crawler Web Search Engine Sequence diagram for the search function. (Part of UrlEntriesController).

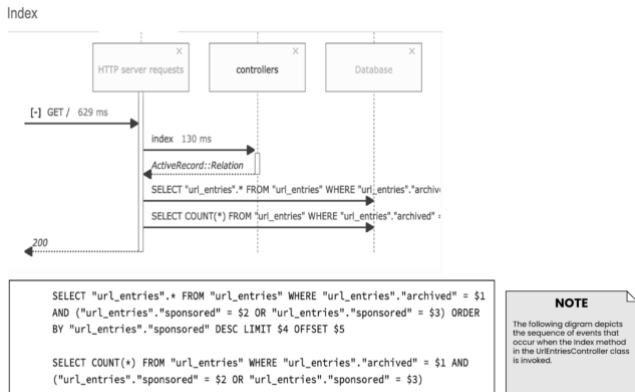


Fig. 18. Comet Crawler Web Search Engine Sequence diagram for the System's Index (Part of UrlEntriesController).

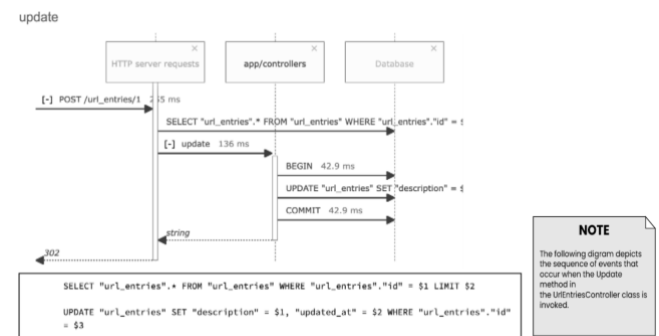


Fig. 22. Comet Crawler Web Search Engine Sequence diagram for the update function. (Part of UrlEntriesController)



Fig. 19. Comet Crawler Web Search Engine Sequence diagram for the show function. (Part of UrlEntriesController).

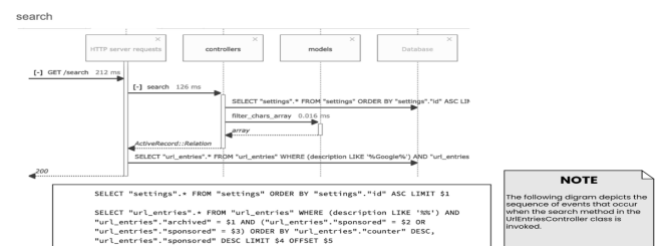


Fig. 23. Comet Crawler Web Search Engine Sequence diagram for the search function. (Part of UrlEntriesController).

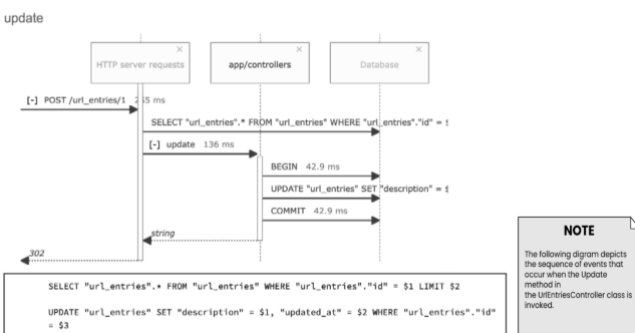


Fig. 20. Comet Crawler Web Search Engine Sequence diagram for the update function. (Part of UrlEntriesController)

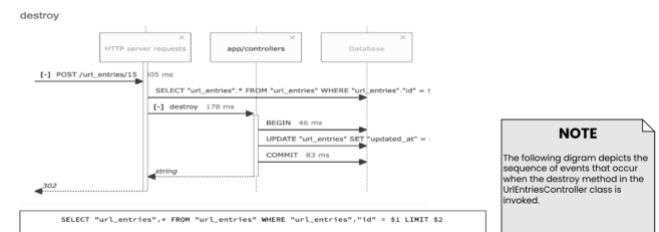


Fig. 24. Comet Crawler Web Search Engine Sequence diagram for the destroy function. (Part of UrlEntriesController)

III. PROTOTYPE IMPLEMENTATION

The Prototype can be found here:

<https://cs-4376-cyberminer.herokuapp.com/>

A. Language and Server-side Web Application Framework

Rails is a Model-View-Controller framework written in Ruby. It is designed to make programming web applications easier by making assumptions about what is “best” [2].

- Example: Rails assumes that you will want CRUD functionality in your HTTP controller, so it includes default functions for these actions that can be extended with specific business logic.

The Rails philosophy includes two major guiding principles:

- Don't Repeat Yourself: Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.
- Convention Over Configuration: Rails has opinions about the best way to do many things in a web application, and defaults to this set of conventions, rather than requiring that you specify minutiae through endless configuration files.

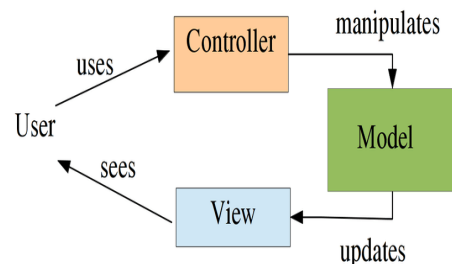


Fig. 30. Illustrates how the MVC architecture in the Rails framework operates.

Developing a prototype for a Ruby on Rails search engine application requires a systematic approach with a focus on technical implementation and attention to best practices, alongside following conventions. Initially, defining the project's scope, identifying key features, and outlining the user experience are crucial steps. Selecting suitable gems and plugins to provide authentication and optimize search queries is essential. The application's core involves implementing search queries, input/view handling, and displaying search results with pagination, filters, and sorting options. A comprehensive test suite focusing on unit tests is vital to detect and resolve potential issues early in the development process. Additionally, our team focuses on requirements analysis and user input to give us insight for further refining the application. Based on feedback and search algorithm improvements, we can provide a platform

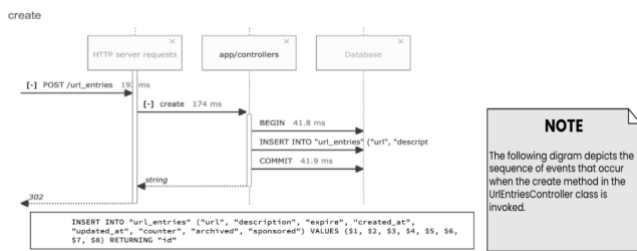


Fig. 25. Comet Crawler Web Search Engine Sequence diagram for the create function. (Part of UrlEntriesController)

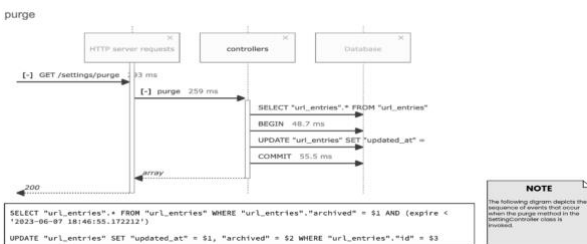


Fig. 26. Comet Crawler Web Search Engine Sequence diagram for the purging expired links function. (Part of SettingController)

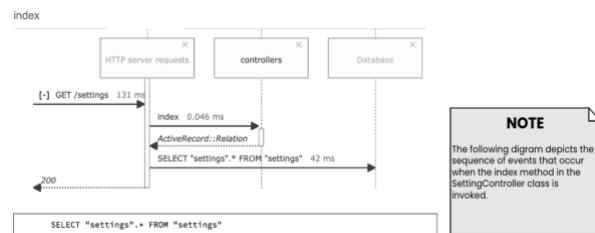


Fig. 27. Comet Crawler Web Search Engine Sequence diagram for the Setting's Index. (Part of SettingController)



Fig. 28. Comet Crawler Web Search Engine Sequence diagram for the show settings function. (Part of SettingController)

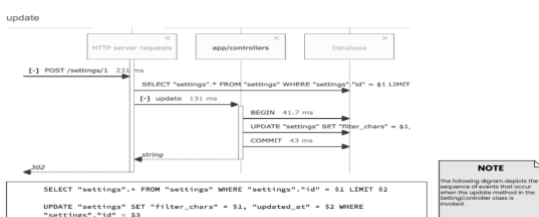


Fig. 29. Comet Crawler Web Search Engine Sequence diagram for the update settings function. (Part of SettingController)

where search functionality can be carried out, as well as handle future features to be integrated.

IV. TESTING

All our test cases can be found here:

<https://github.com/tyharg/CS-4376/tree/main/test>

```
class SettingsControllerTest < ActionController::IntegrationTest
  setup do
    @url_entry = url_entries(:one)
  end

  test "should get index" do
    get settings_url
    assert_response :success
  end

  test "should get new" do
    get new_settings_url
    assert_response :success
  end

  test "should create setting" do
    assert_get_response_success do
      post settings_url, params: { setting: { filter_shares: @url_entry.filter_shares } }
    end
    assert_redirected_to settings_url(settings.last)
  end
end

class UrlEntriesControllerTest < ActionController::IntegrationTest
  setup do
    @url_entry = url_entries(:one)
  end

  test "should get index" do
    get url_entries_url
    assert_response :success
  end

  test "should get new" do
    get new_url_entry_url
    assert_response :success
  end
end
```

Fig. 31. Depicts a couple of our project test cases.

URL	Description	Clicks	Expire	Sponsored?
https://teamtwinkl.com/education/primary/plan-software/	Preliminary Project Plan Soft Copy	28	2025-08-01 23:59:00 UTC	Yes
https://google.com	Google official website	6	2023-07-29 21:57:00 UTC	Yes
https://personal.utdallas.edu/~chuna/OOD/syllabus.htm	CS 4376 course syllabus	6	2023-10-20 21:59:00 UTC	No
https://github.com/tyharg	my personal website	3	2023-06-30 14:48:00 UTC	Yes
https://twitter.com	twitter main homepage	2	2023-06-16 21:59:00 UTC	No
https://google.com	google	1	2023-07-29 01:14:00 UTC	No
https://utdallas.edu	UT Dallas official website	1	2023-07-27 21:57:00 UTC	No
https://twitch.tv	Twitch.tv official homepage	0	N/A	No
https://utdallas.edu	UTD	0	2023-08-04 03:34:00 UTC	No
https://personal.utdallas.edu/~chuna/OOD/syllabus.htm	TEST TEST TEST	1	2023-09-08 01:15:00 UTC	No

Fig. 32. Shows a listing of our indices content. This list was last updated on 06/15/2023.

V. USER MANUAL

The Phase I User Manual can be found here: [User Manual](#)

This manual applies to the first phase version of the Comet Crawler Web Search Engine. This means there are no Login or privileged user capabilities yet. These functionalities will be made available in the phase 2 version of the system which is expected to come out on July 13, 2023

Authors:
Areebah Fatima,
Ilhaam Syed,
Mathew Bedford,
Nathan Heindl,
Tyler Hargreaves,
Darrien Kramer

Fig. 33. Illustrates the title page of our User Manual.

CONTENTS

1 ACCESSING THE SEARCH ENGINE SYSTEM

How to Access the Website Through a Hyperlink	1-1
How to Access the Website Through a Hyperlink Window	1-2

2 STEP BY STEP SYSTEM FUNCTIONS

How to Carry Out a Search	2-1
How to View Information About a URL Entry	2-2
How to Edit / Delete a URL Entry	2-3
How to Create a URL Entry	2-4
How to Modify the Filtered Character's List	2-5
How to Purge All Out-of-Date URLs	2-6

Fig. 34. Illustrates our User Manual's Table of Contents.

1

ACCESSING THE SEARCH ENGINE SYSTEM

How to Access the Website Through a Hyperlink

Step 1: Hover your mouse over the following link, which is underlined and highlighted in blue.

Search Engine Web Page URL: <https://cs-4376-cyberminer.herokuapp.com/>

Search Engine Web Page URL: <https://cs-4376-cyberminer.herokuapp.com/>

Figure 1.1 When you have successfully hovered your mouse point over the hyperlink it should look something like this.

Step 2: Once your mouse point is hovering over the link, press down on your mouse to click the link.

Step 3: Viola, you have found your way to the Comet Crawler Search Engine Web Page

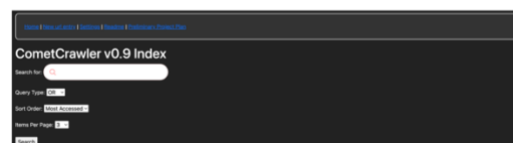


Figure 1.2 The following image illustrates the Comet Crawler Web Search Engine homepage.

Fig. 35. Shows the steps to access the Comet Crawler Webpage

How to Access the Website Through a Web Browser Window

Step 1: Open a browser window of your choice.

Figure 1.3 The following image illustrates a Safari web browser window.

Step 2: Click the textbox located at the top of your preferred browser window. It might be labeled with a command such as “Search or enter a website name”.

Step 3: Type the following into your search box: <https://cs-4376-cyberminer.herokuapp.com/>.

Step 4: Once you hit enter or click the search icon you should be redirected to the Comet Crawler web search homepage.

Fig. 36. Shows the alternative steps to access the Comet Crawler Webpage

2 STEP BY STEP SYSTEM FUNCTIONS

How to Carry Out a Search

Note: If you have not yet opened the Comet Crawler Homepage, please follow the instructions in Section 1 of this user manual.

Step 1: Hover your mouse point over the search bar. Click the search bar and type in your search query. Note: while typing you may notice some autofill options, if one of them match what you're looking for just click on it. Otherwise, finish typing in your search query and click to close the box.

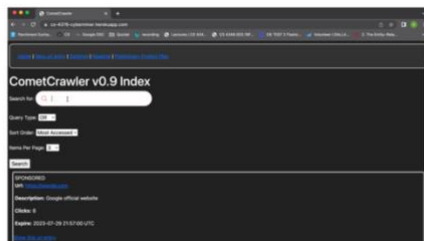


Figure 2.1 The following highlights how to hover your mouse point over the search bar.

Fig. 37. Aims to teach System users how to search the Comet Crawler Webpage.

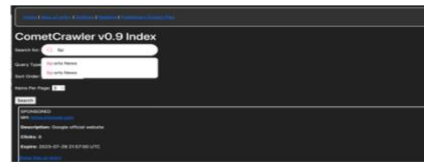


Figure 2.2 The following image illustrates the autofill feature.

Step 2: Click Each drop-down box and specify your search parameters.

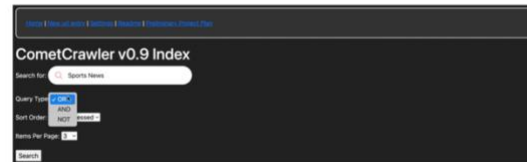


Figure 2.3 The following image illustrates the drop-down boxes for the search parameters.

Step 3: Click the grey box labeled search once you have specified all your search parameters. This should redirect you to the search result page, where you will filled all the relevant search URLs.

Fig. 38. Continues to teach System users how to search the Comet Crawler Webpage.

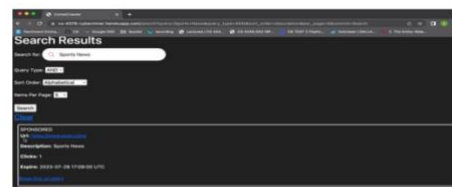


Figure 2.4 The following image illustrates the search result page.

How to View Information About a URL Entry

Step 1: Once you have located your desired URL entry, click on the blue underlined link named “Show this URL entry”. This should redirect you to the URL information Page.



Figure 2.5 The following image illustrates the URL information page.

Step 2: To return back to the website's homepage, click the blue underlined link labeled "Back to url entries".

How to Edit / Delete a URL Entry

Fig. 39. Instructs users on how to view a URL's information.

Note: If you have not yet opened the URL's entries information page, please follow the instructions in Section 2 of this user manual to do so.

Steps to delete a URL entry:

1. Open the URL entry information page.
2. Click the grey button labeled "Destroy this url entry".
3. If a message pops up at the top of your browser window, click ok to continue the deletion process. When done correctly you should see a confirmation message.



Figure 2.6 The following image illustrates the confirmation page.

Steps to modify a URL entry:

1. Open the URL entry information page.
2. Click on the blue-underlined link labeled "Edit this url entry".
3. You should see the following page:

Fig. 40. Depicts the steps a user would have to take to delete a URL entry.



Figure 2.7 The following image illustrates the modifying URL entry parameters.

Please modify all the necessary parameters and click the grey button labeled "Update URL entry" when you are done.

4. When done correctly you should see a confirmation message.

How to Create a URL Entry

Steps to Create a URL entry:

1. Click the blue-underlined link labeled "New url entry" located in the Comet Crawler Homepage Header. This should redirect you to a page that looks like Figure 2.8.



Figure 2.8 The following image illustrates the New URL entry Page.

Fig. 41. Depicts the steps a user would have to take to create a URL entry.

2. Fill out all the parameters listed on the web page. For unsponsored URLs keep the "Paid Sponsor" box unchecked.
3. When done correctly you should see a confirmation message telling you your URL entry has been created.

How to Modify the Filtered Character's List

Steps to Modify the Filtered Character's List:

1. Click the blue-underlined link labeled "Settings" located in the Comet Crawler Homepage Header. This should redirect you to a page that looks like Figure 2.9.

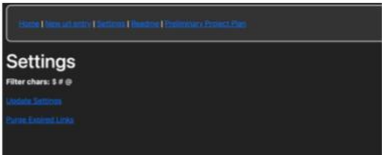


Figure 2.9 The following image illustrates the Settings Page.

2. Click the blue-underlined link labeled "Settings" to modify the filtered characters list.
3. You should be led to a page like the one shown in Figure 2.10. Click the blue-underlined link labeled "update settings" and type in the new filtered characters list.

Fig. 42. Demonstrates how to manage The Comet Crawler Search Engine's System Setting.

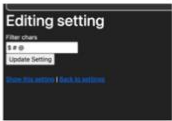


Figure 2.9 The following image illustrates the edit settings Page.

4. When done correctly you should see a confirmation message telling you that the list has been modified.

How to Purge All Out-of-Date URLs

Steps to Purge all out-of-date URLs:

1. Click the blue-underlined link labeled "Settings" located in the Comet Crawler Homepage Header. This should redirect you to a page that looks like Figure 2.9.

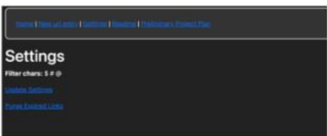


Figure 2.9 The following image illustrates the Settings Page.

2. Click the blue-underlined link labeled "Purge Expired Links" to delete out-of-date URL entries.
3. If a message pops up at the top of your browser window, click ok to continue the deletion process. When done correctly you should see a confirmation

Fig. 43. Shows how to purge expired links in order to keep search results as relevant as possible.

A. Definitions

Component Is a basic building block of a system that provides some form of functionality. Identifying key components and component relationships is a fundamental part of the design stage.

Index Is the data structure that stores and organizes the data about URL entries and web pages.

B. Abbreviations

CI/CD Continuous Integration / Continuous Delivery

MVC Model-View-Controller

UML Unified Modeling Language

REFERENCES

[1] G. BOOCH, *Unified Modeling Language User Guide*. ADDISON-WESLEY, 2017.

[2] S. Holzner, *Beginning Ruby on Rails*. Indianapolis: Wrox, 2007.