

A complex network graph with numerous light blue nodes and connecting lines, set against a dark blue background.

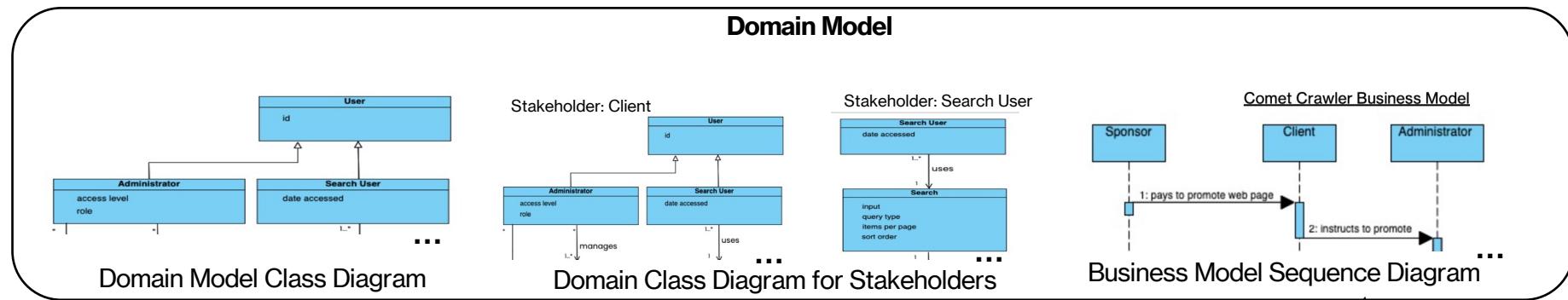
Interim Project: Phase 2

Team 1:
Areebah Fatima
Tyler Hargreaves
Darrien Kramer
Ilhaam Syed
Nathan Heindl
Matthew Bedford

System Overview

- This 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 provided to the end user will be a sorted and filtered list of web page URLs.
 - The system allows sponsors to pay to promote their websites to the top of the search result list.
 - 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.
-

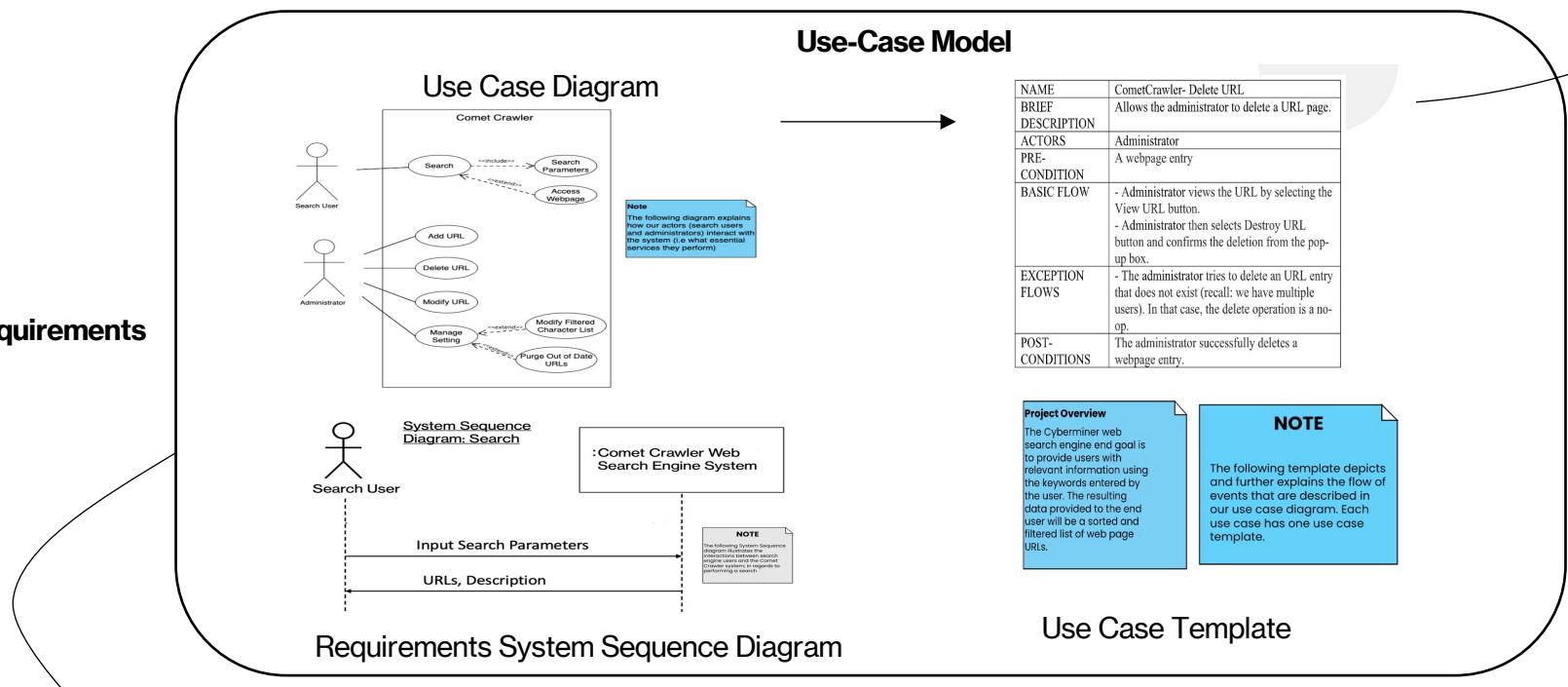
Business Modeling



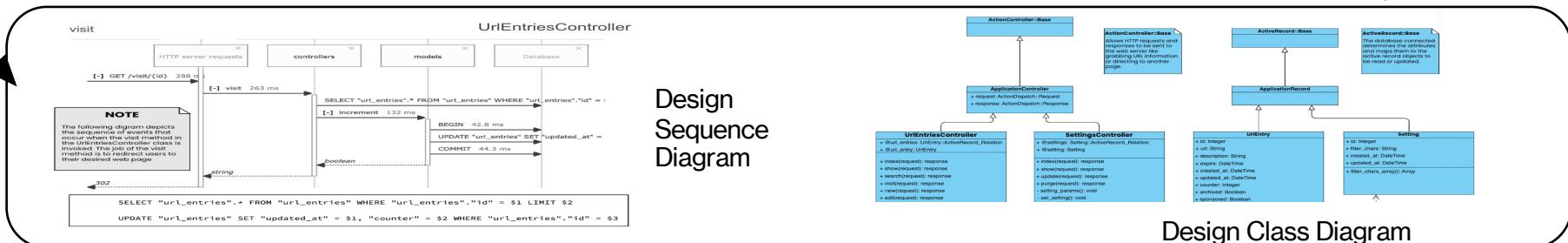
NOTE

The following diagram (part 1 & 2) illustrates our presentation and project roadmap. It highlights the diagrams and project specifications made at each step

Requirements



Design



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”.

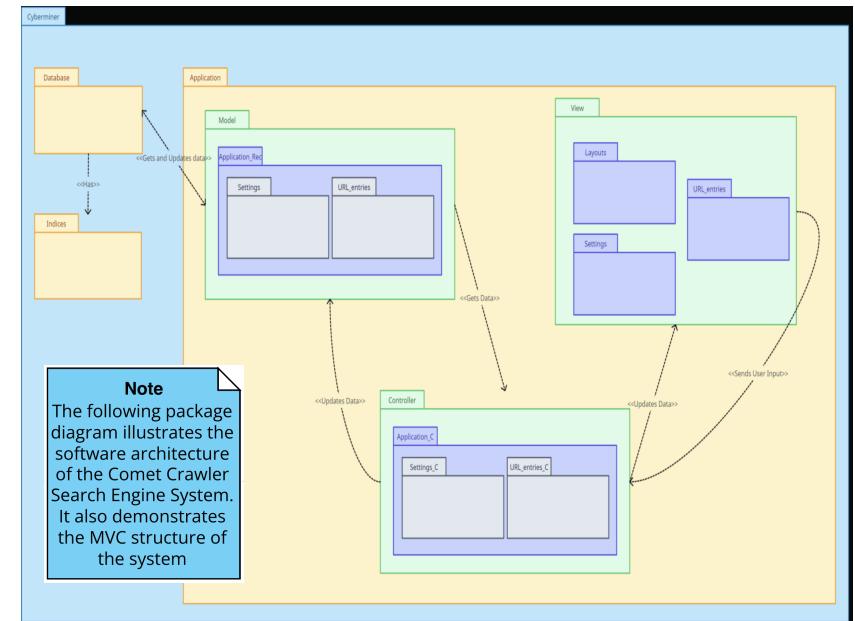
- Example: Rails makes the assumption that you will want CRUD functionality in your HTTP controller, so it includes default functions for these actions that can be extended with specific business logic.

Implementation

The Rails philosophy includes two major guiding principles:

1. Don't Repeat Yourself: Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.
2. 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 require that you specify minutiae through endless configuration files.

Language and server-side web application framework



Package Diagram

Testing

```
class SettingsControllerTest < ActionDispatch::IntegrationTest
  setup do
    @setting = settings(:one)
  end

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

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

  test "should create setting" do
    assert_difference("Setting.count") do
      post settings_url, params: { setting: { filter_chars: @setting.filter_chars } }
    end
  end

  assert_redirected_to setting_url(setting.last)
end
```

```
class UrlEntriesControllerTest < ActionDispatch::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
```

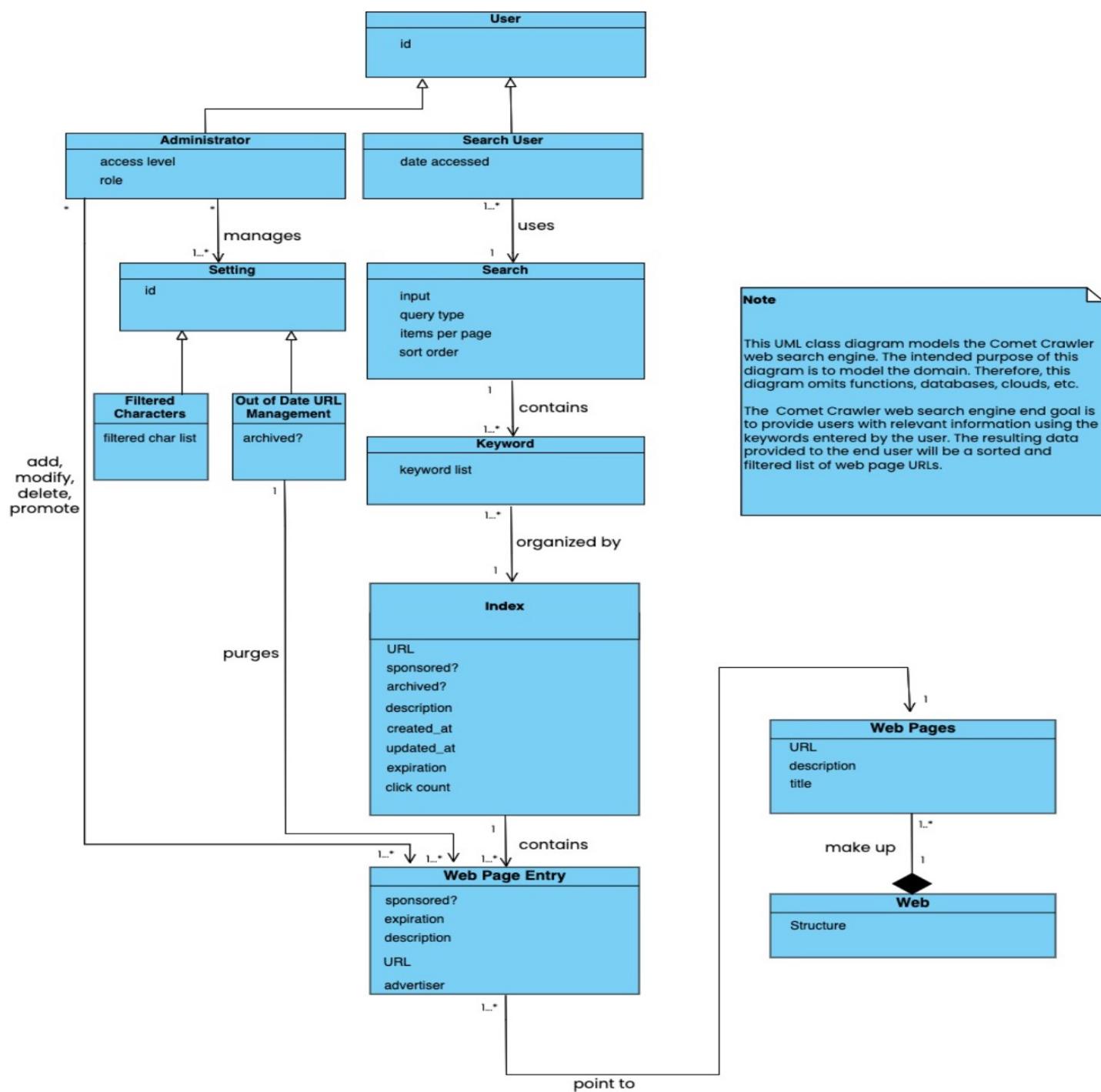
More Test Cases can be found on our GitHub [repository](#)

Test Case

URL	Description	Clicks	Expire	Sponsored?
https://team1preliminaryprojectplan.tiny.site/	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/~chung/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/~chung/OOD/syllabus.htm	TEST TEST TEST	1	2023-09-08 01:15:00 UTC	No

Indices Listing

Domain Model Class Diagram

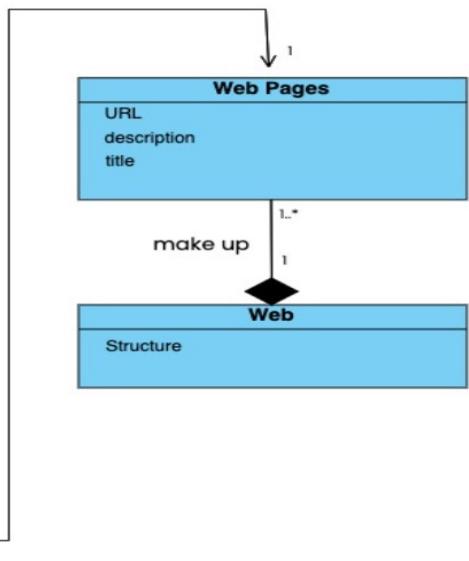
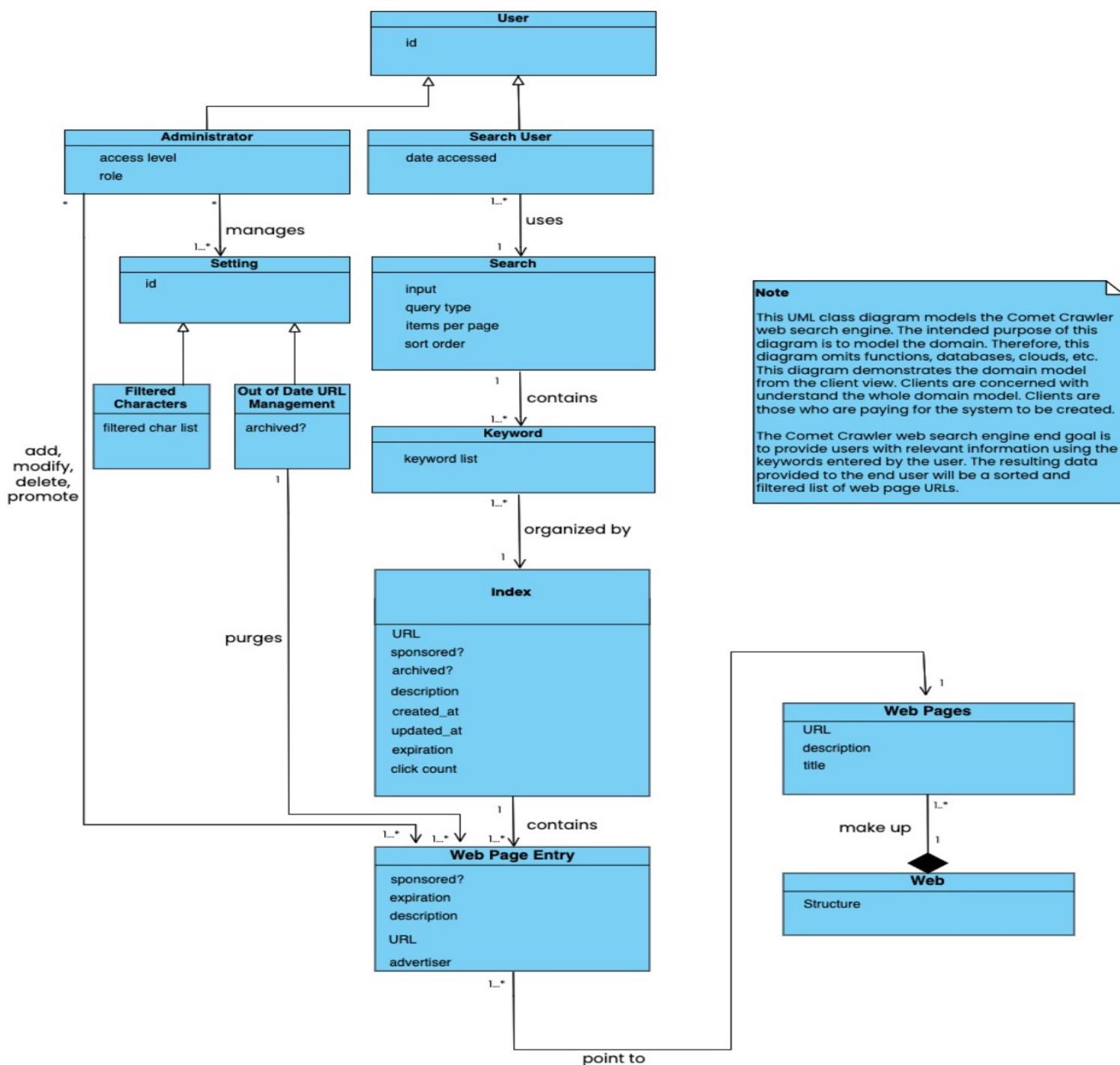


This diagram is a part of **Business Modeling**

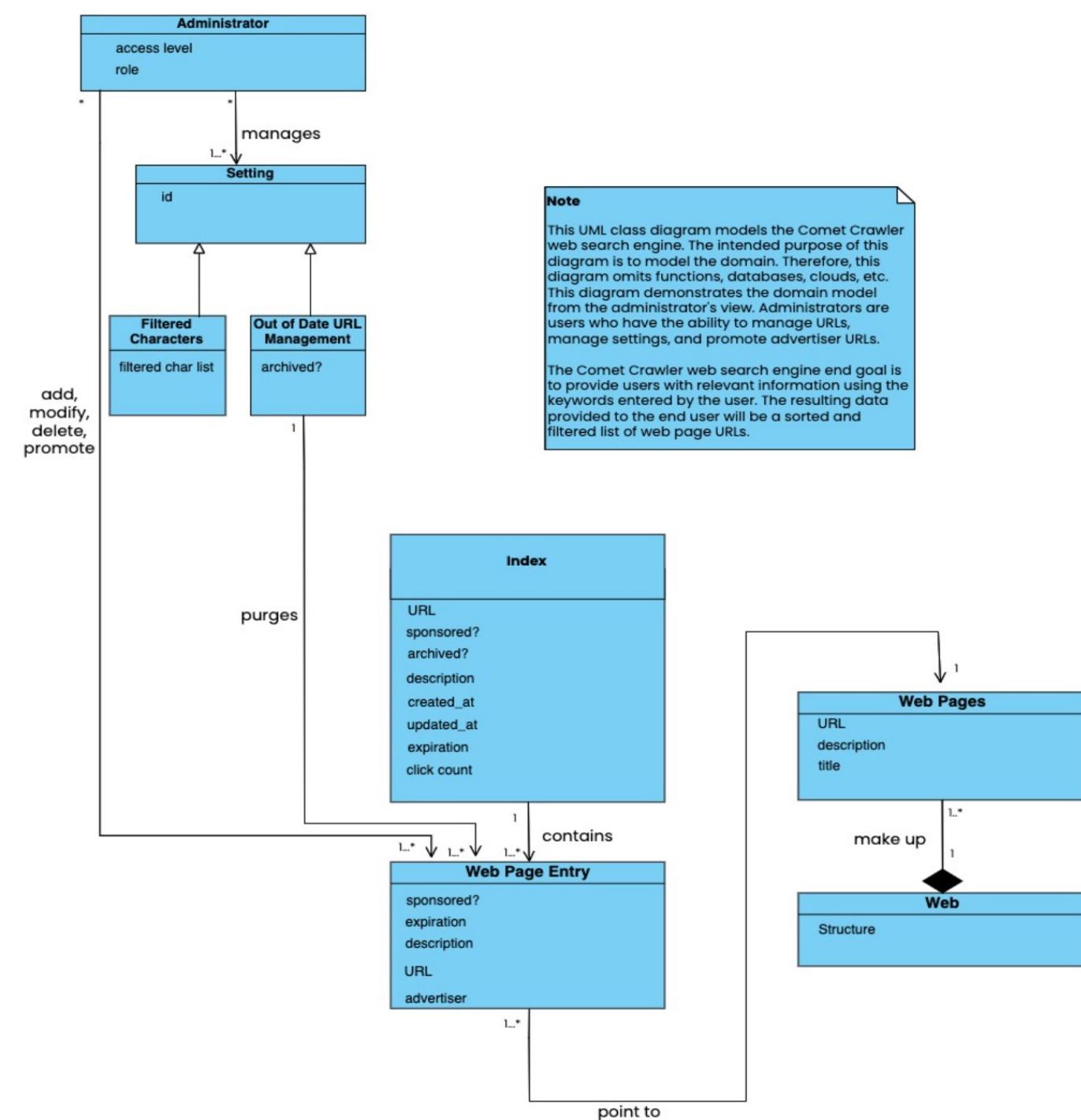
Comet Crawler Stakeholders

- **Administrator:** Are privilege 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 makes them the recipients of project deliverables and the resulting end product.
 - **Sponsor:** Are the entities that pay money to promote their websites to the top of the search result list. This stakeholder is the central piece of the business model.
 - **Search Users:** 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.
-

Stakeholder Domain Model Class Diagram: Client



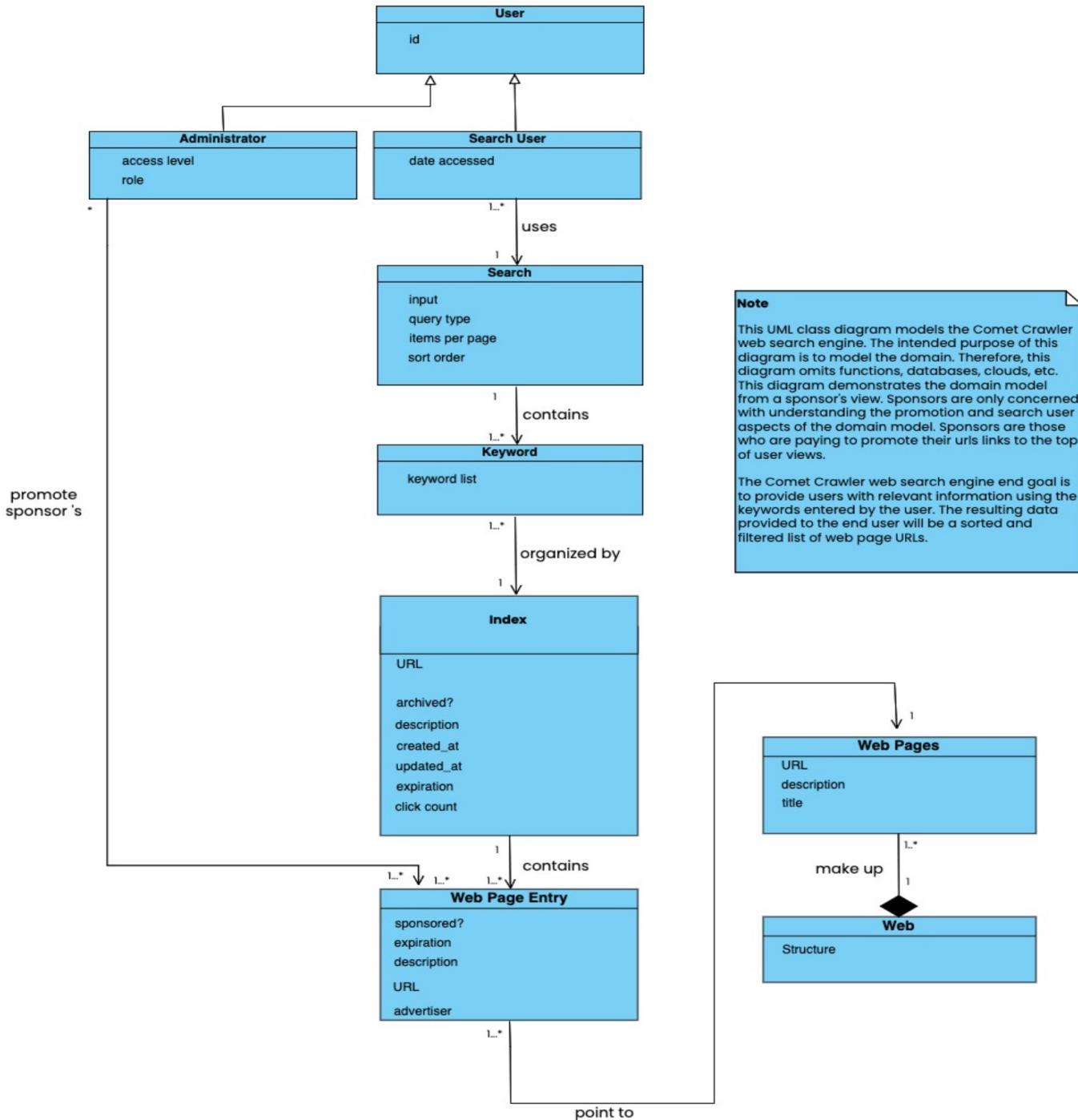
This diagram is a part of **Business Modeling**



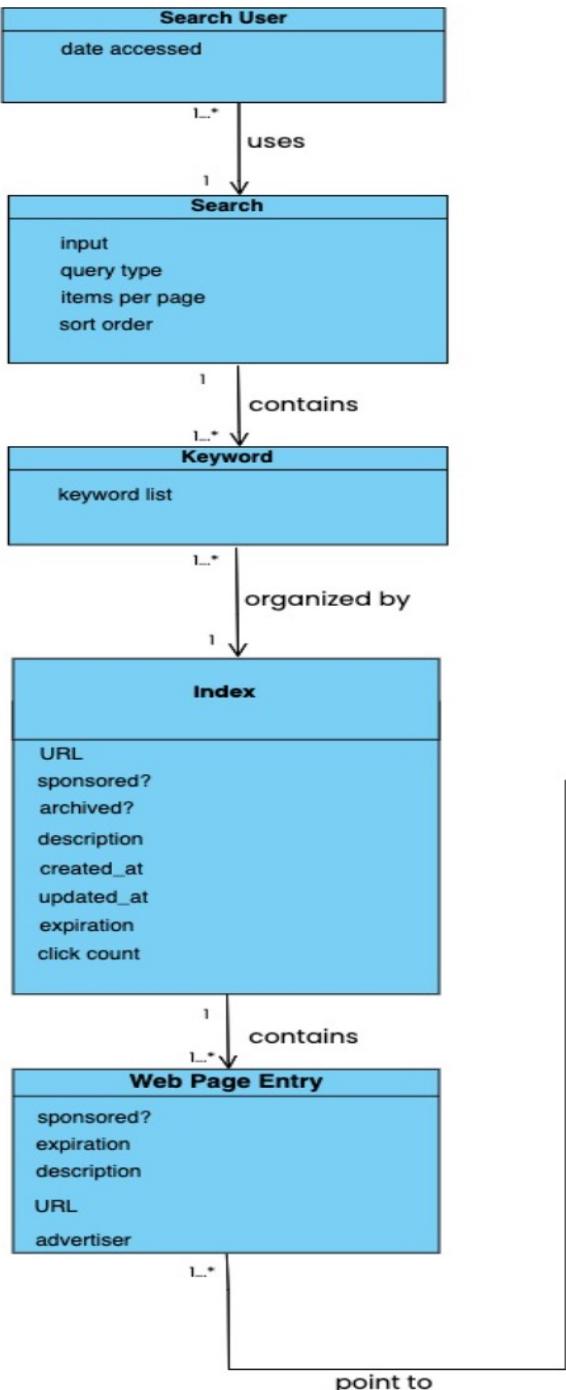
Stakeholder Class Diagram : Administrator

This diagram is a part of **Business Modeling**

Stakeholder Class Diagram : Sponsor



This diagram is a part of **Business Modeling**

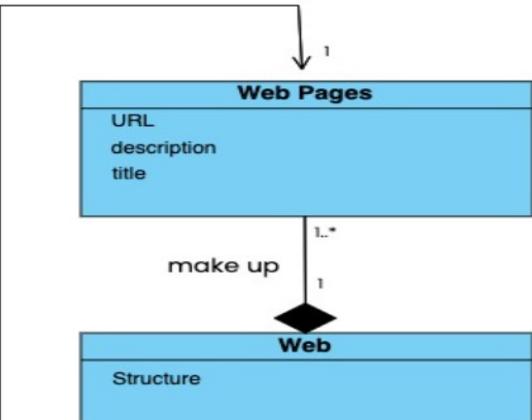


Note

This UML class diagram models the Comet Crawler web search engine. The intended purpose of this diagram is to model the domain. Therefore, this diagram omits functions, databases, clouds, etc. This diagram demonstrates the domain model from the search user's view. Search users are people who have access to the web engine url and enter search queries.

The Comet Crawler web search engine end goal is to provide users with relevant information using the keywords entered by the user. The resulting data provided to the end user will be a sorted and filtered list of web page URLs.

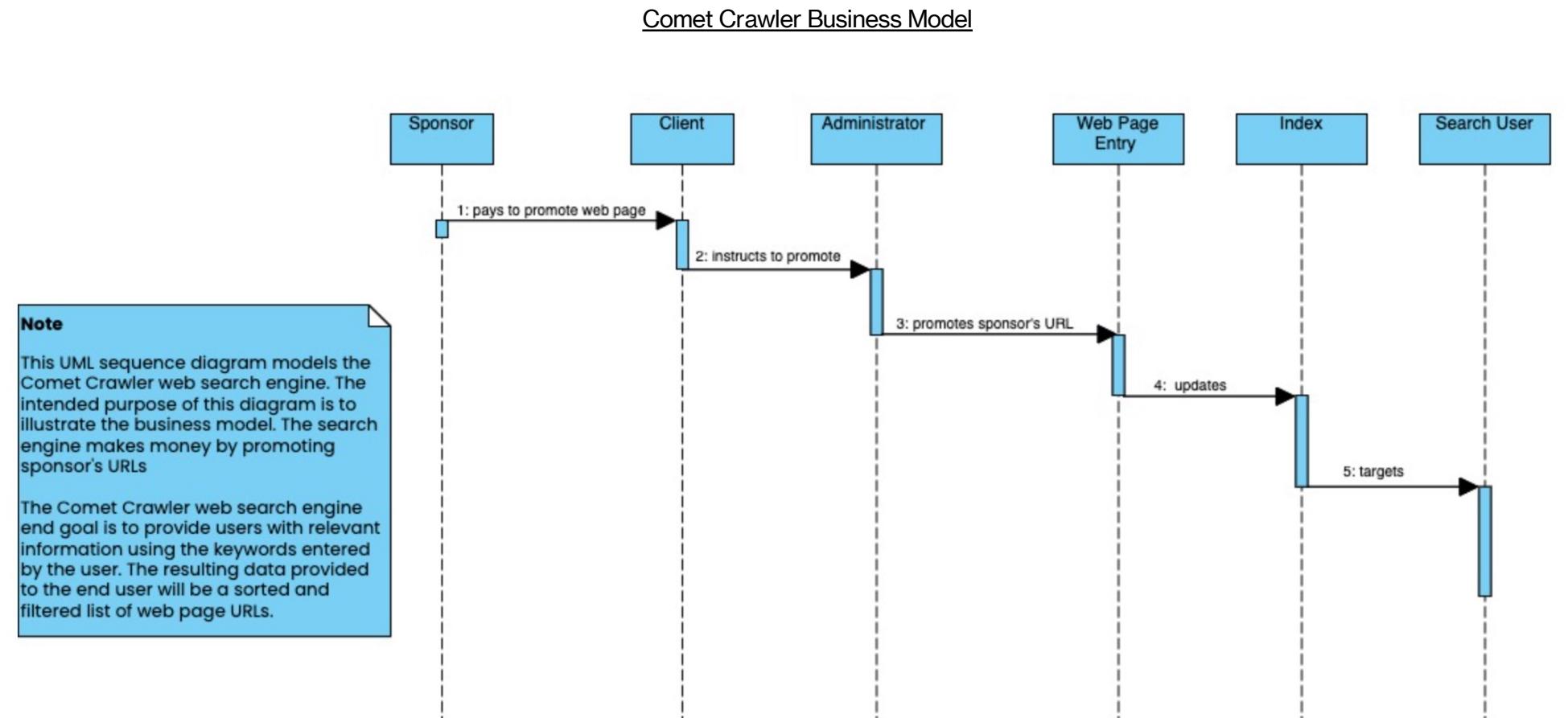
Stakeholder Class Diagram : Search Users



This diagram is a part of **Business Modeling**

Business Model Sequence Diagram

This diagram is a part of **Business Modeling**



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 URL and descriptions:** The system shall provide 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.

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.

URL Syntax BNF

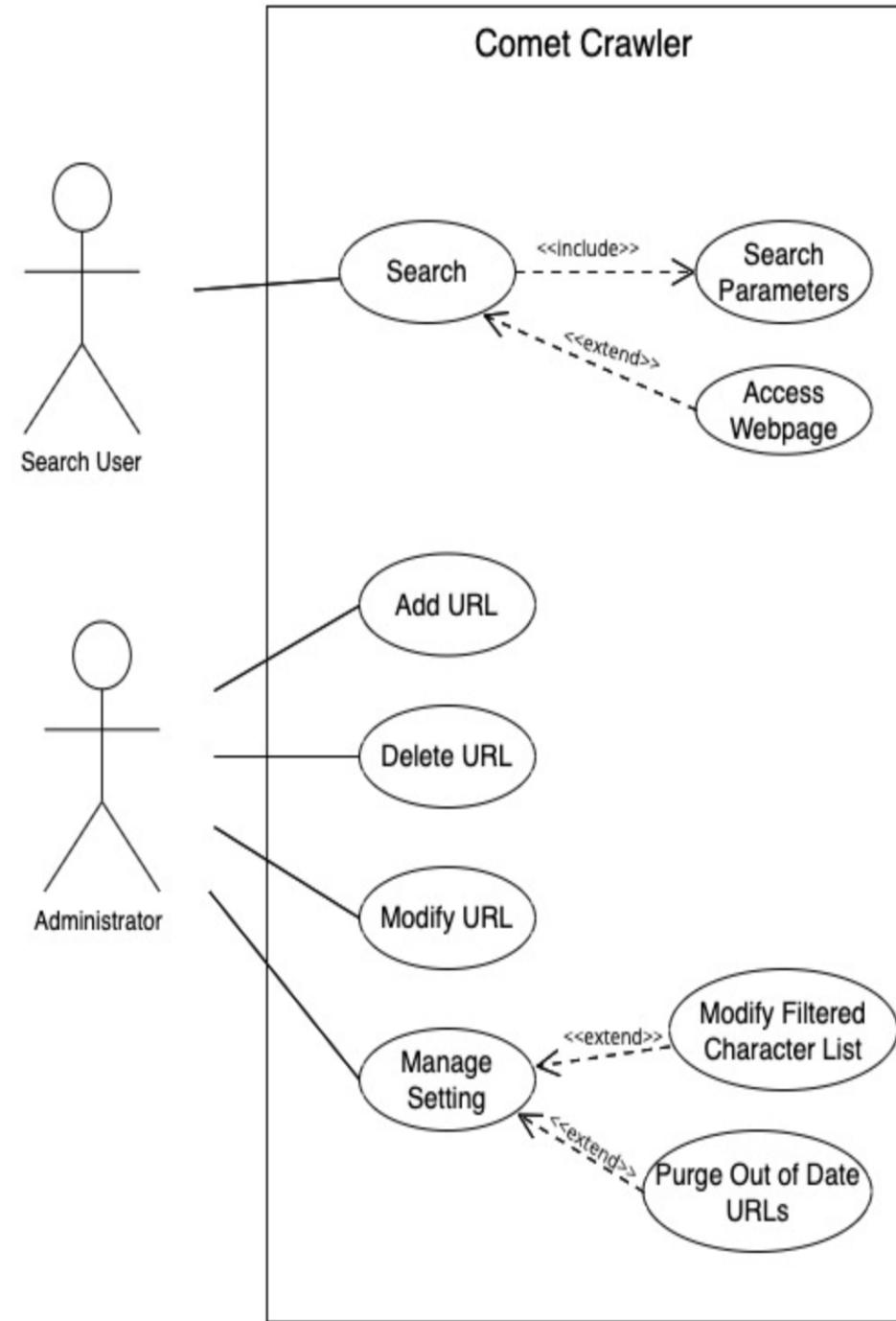
URL ::= {[`http://`|`https://`] } {www.} Identifier ':' [`edu` | `com` | `org` | `net` | `gov` | `mil`]

Identifier ::= [letter | digit]⁺

letter = ['a' | 'b' | 'c' | 'd' | ... | 'y' | 'z' | 'A' | 'B' | 'C' | 'D' | ... | 'Y' | 'Z']

digit := ['1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' | '0']

Use Case Diagram



Note

The following diagram explains how our actors (search users and administrators) interact with the system (i.e what essential services they perform)

Project Overview

The Cyberminer web search engine end goal is to provide users with relevant information using the keywords entered by the user. The resulting data provided to the end user will be a sorted and filtered list of web page URLs.

This diagram is a part of **Requirements**

Use Case Template

These diagrams are a part of **Requirements**

NAME	CometCrawler- search
BRIEF DESCRIPTION	Allows search users to search URL pages based on a single or a set of keywords. It also extends the search by allowing search users to view the return pages.
ACTORS	search user
PRE-CONDITION	Query type, Sort Order, Items per page. Default is provided for all of them.
BASIC FLOW	<p>Search:</p> <ul style="list-style-type: none"> - User enters keywords in the search bar. - Keywords are interpreted and matched with entries. - The results are posted according to the preconditions provided. - The user accesses the webpage(s)
EXCEPTION FLOWS	<p>Search:</p> <ul style="list-style-type: none"> - The user enters keywords that are not in the database. - The search will lead to 0 results.
POST-CONDITIONS	Users can do a successful search. Users can view/delete/update URL entries successfully

NAME	CometCrawler- Add URL
BRIEF DESCRIPTION	Allows the administrator to add a URL page.
ACTORS	Administrator
PRE-CONDITION	URL entry form
BASIC FLOW	<ul style="list-style-type: none"> - Administrator must select a new webpage entry from the home page. - Administrator provides the URL to be added, a description of the URL, and an expiration date. - The new webpage entry is created when the Administrator clicks Create URL Entry.
EXCEPTION FLOWS	<ul style="list-style-type: none"> - The administrator tries to add an URL entry that is already in the database. In that case, the create falls back to modify operation.
POST-CONDITIONS	The administrator successfully adds a new webpage entry or modifies an existing webpage entry.

NAME	CometCrawler- Delete URL
BRIEF DESCRIPTION	Allows the administrator to delete a URL page.
ACTORS	Administrator
PRE-CONDITION	A webpage entry
BASIC FLOW	<ul style="list-style-type: none"> - Administrator views the URL by selecting the View URL button. - Administrator then selects Destroy URL button and confirms the deletion from the pop-up box.
EXCEPTION FLOWS	<ul style="list-style-type: none"> - The administrator tries to delete an URL entry that does not exist (recall: we have multiple users). In that case, the delete operation is a no-op.
POST-CONDITIONS	The administrator successfully deletes a webpage entry.

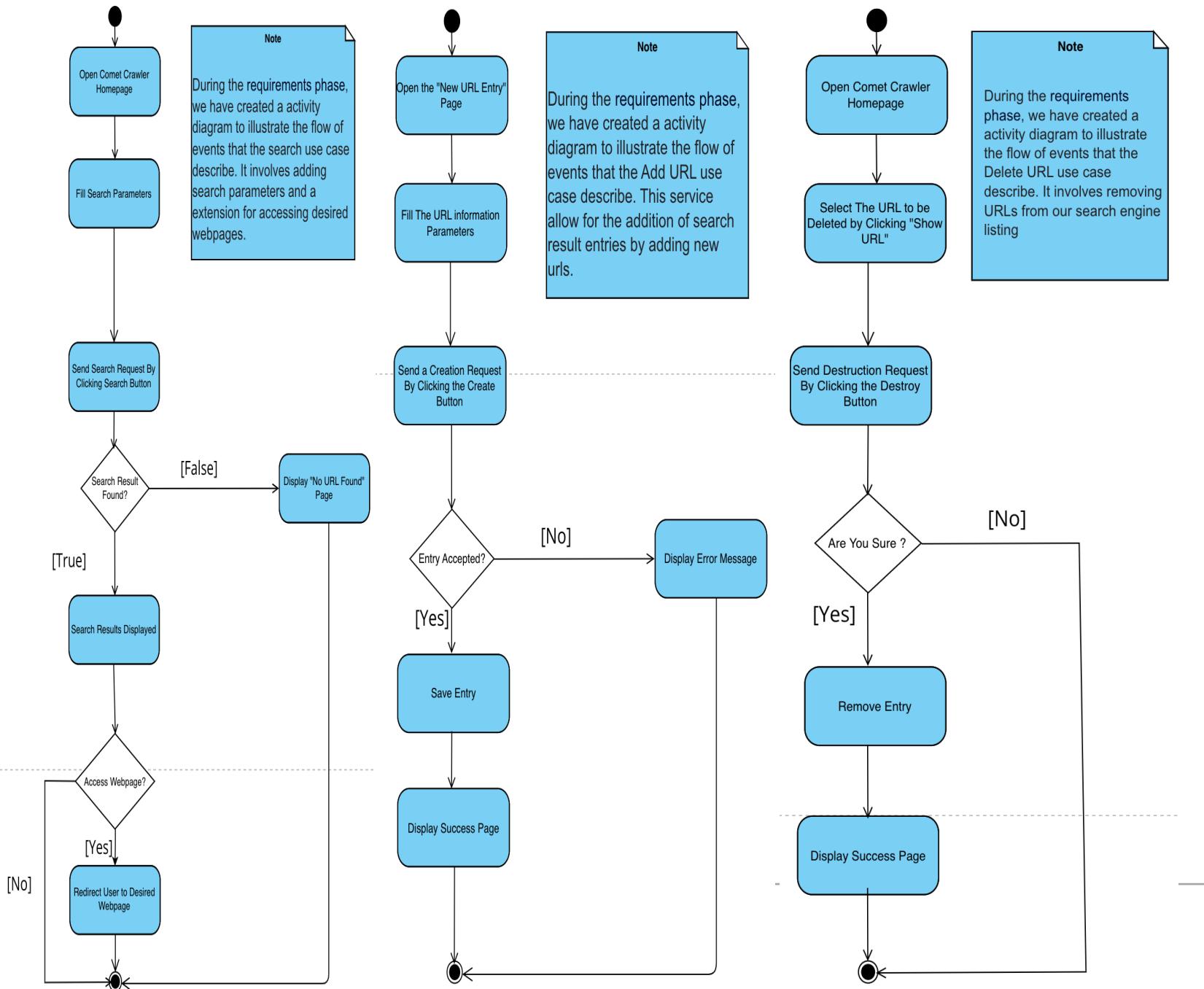
NAME	CometCrawler- Modify URL
BRIEF DESCRIPTION	Allows the administrator to modify a URL page.
ACTORS	Administrator
PRE-CONDITION	A webpage entry
BASIC FLOW	<ul style="list-style-type: none"> - Administrator views the URL by selecting the View URL button. - The Administrator then selects the Edit URL button. - The Administrator then updates the information and confirms the edit.
EXCEPTION FLOWS	<ul style="list-style-type: none"> - The administrator tries to modify an URL entry that is no longer in the database (recall: we have multiple users). In that case, the modify is a no-op operation.
POST-CONDITIONS	The administrator successfully modifies a webpage entry.

NAME	CometCrawler- Manage Setting
BRIEF DESCRIPTION	Allows the administrator to manage settings.
ACTORS	Administrator
PRE-CONDITION	Modify filtered character list or purge out-of-date URLs request.
BASIC FLOW	<ul style="list-style-type: none"> - Administrator views the settings page. - The Administrator updates the settings or requests to purge out-of-date URLs. - The administrator receives a confirmation page.
EXCEPTION FLOWS	<ul style="list-style-type: none"> - The administrator tries to update a setting that is not allowed. In that case, the return page informs the user of no-op.
POST-CONDITIONS	The administrator successfully updates settings or purges expired pages.

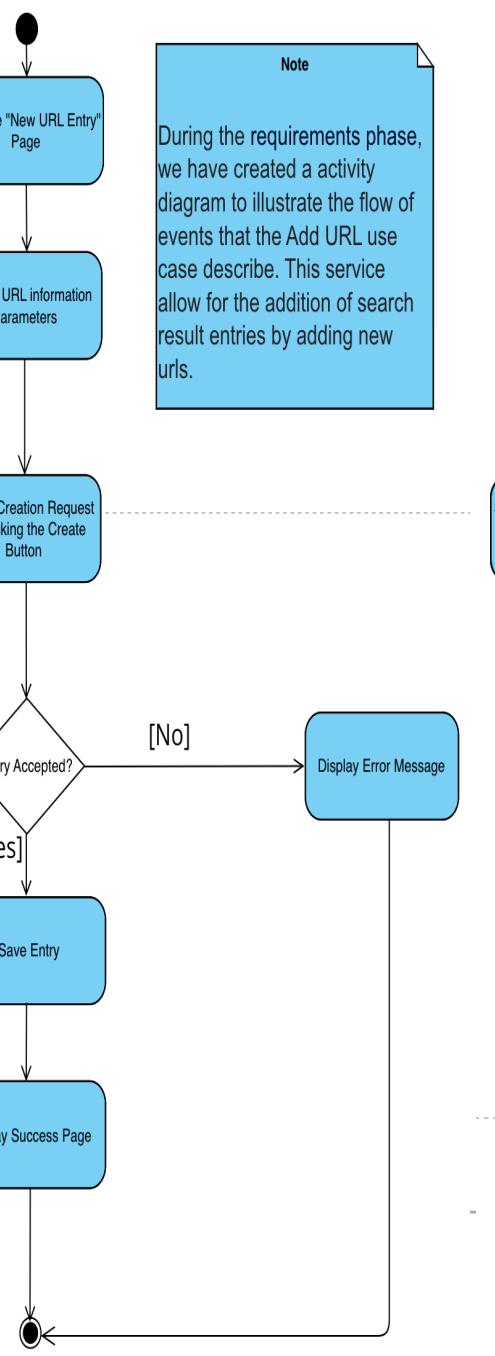
Project Overview
The Cyberminer web search engine end goal is to provide users with relevant information using the keywords entered by the user. The resulting data provided to the end user will be a sorted and filtered list of web page URLs.

NOTE
The following template depicts and further explains the flow of events that are described in our use case diagram. Each use case has one use case template.

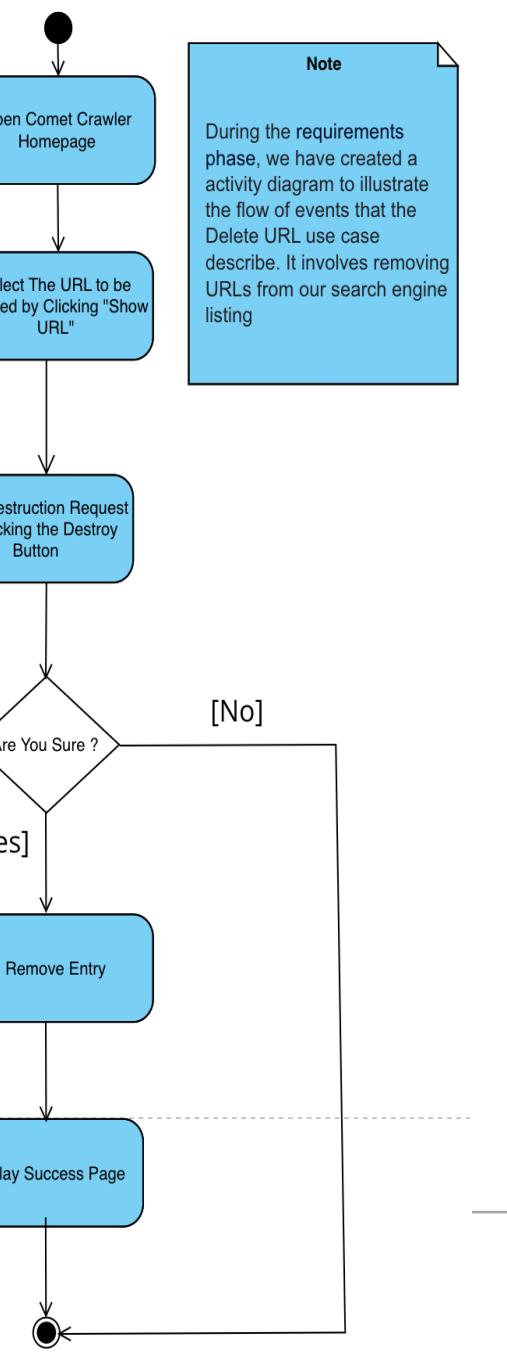
Search Activity Diagram



Add URL Activity Diagram

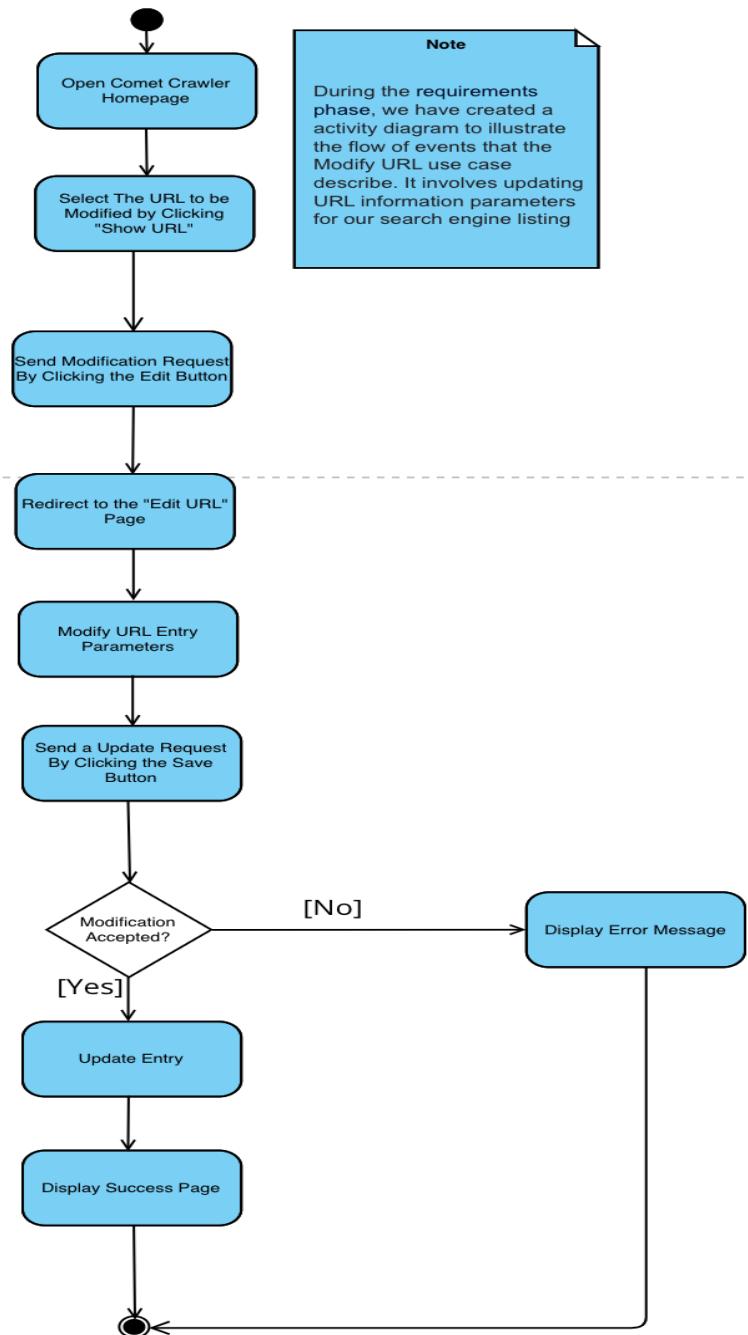


Delete URL Activity Diagram

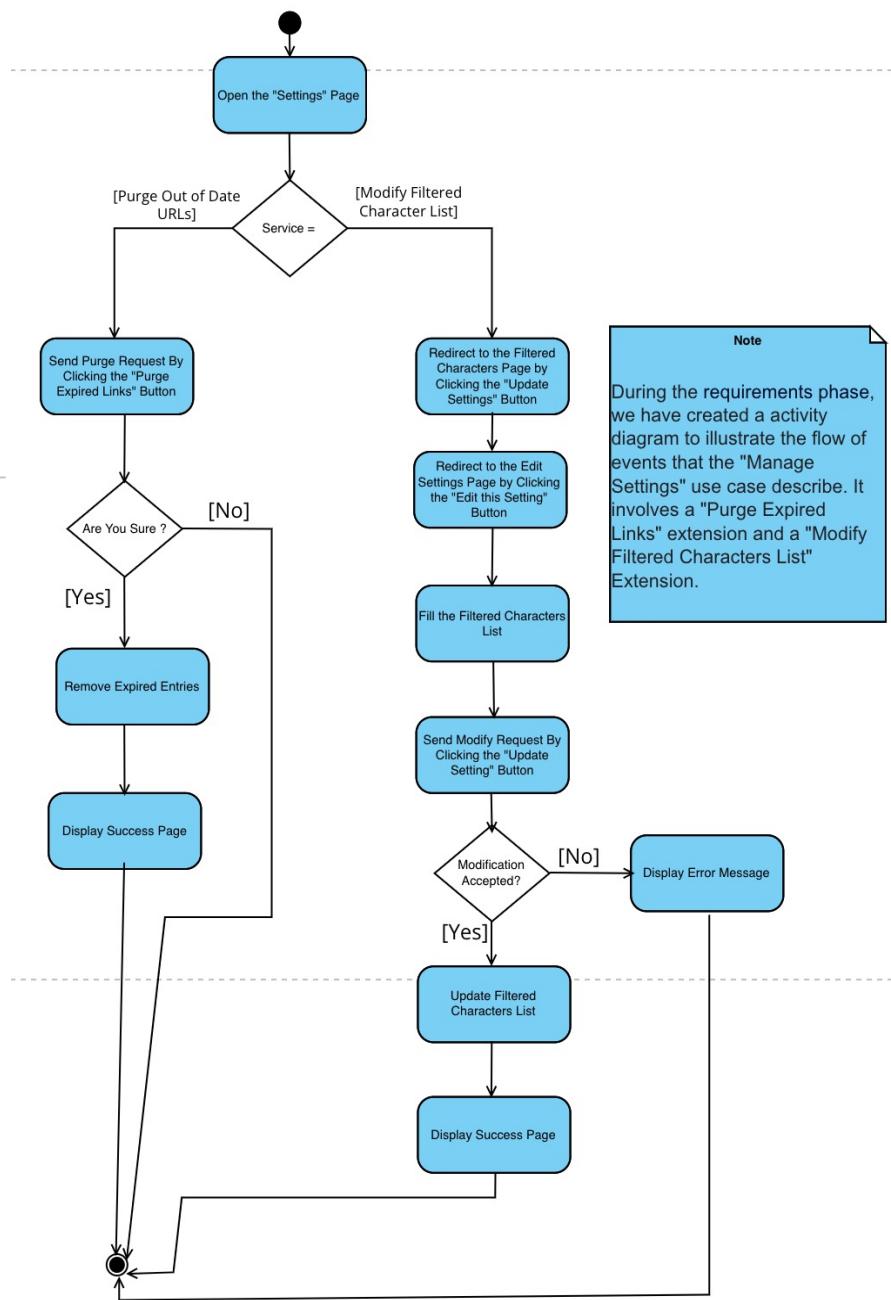


Activity Diagrams

Modify URL Activity Diagram



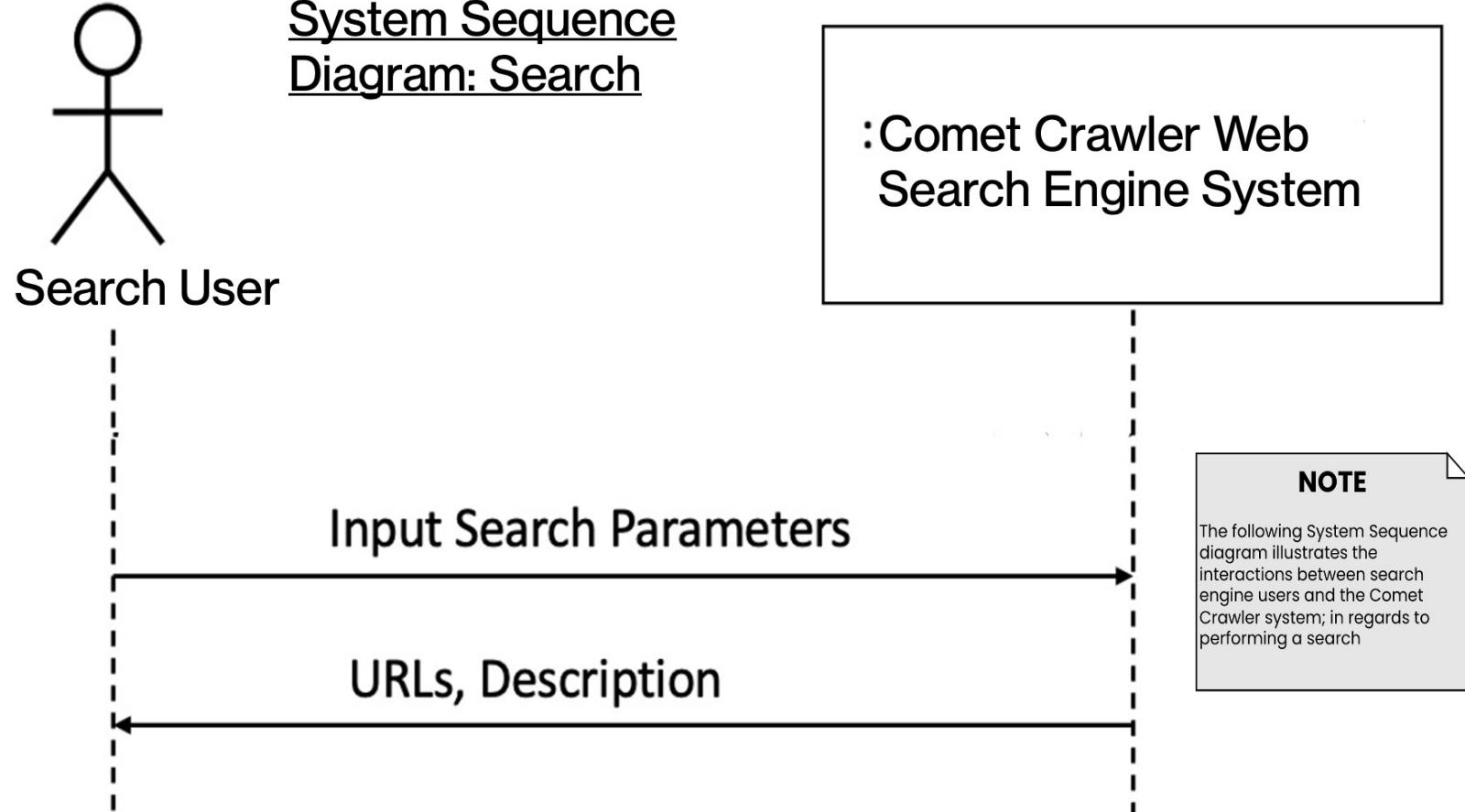
Manage Settings Activity Diagram



Activity Diagrams

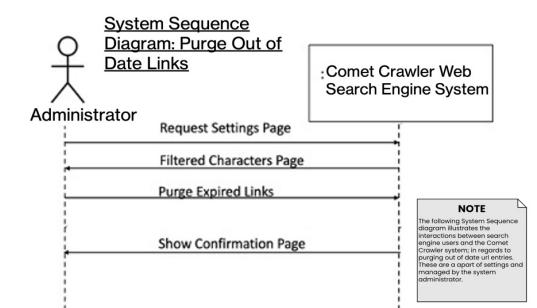
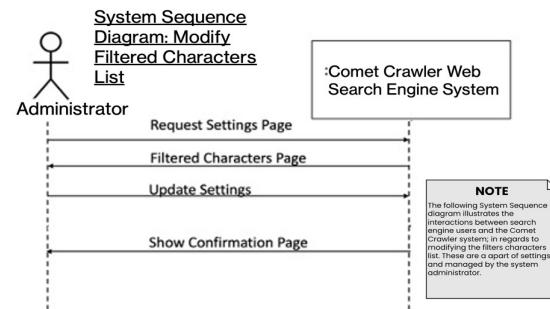
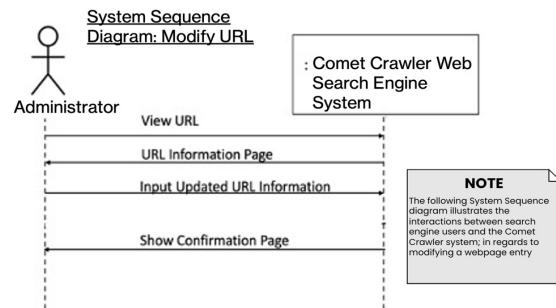
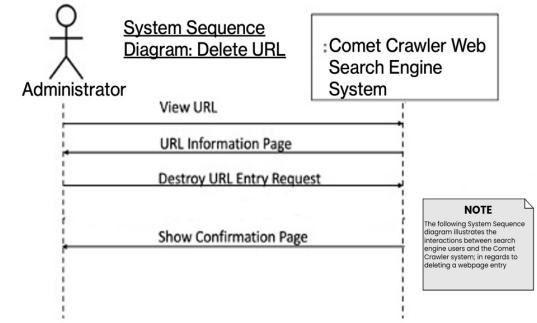
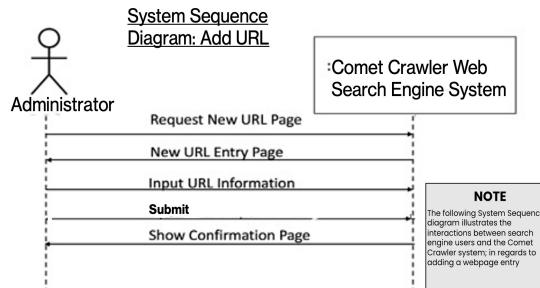
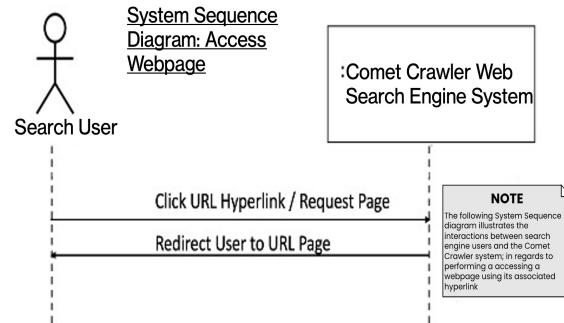
System Sequence Diagrams

This diagram is a part of Requirements



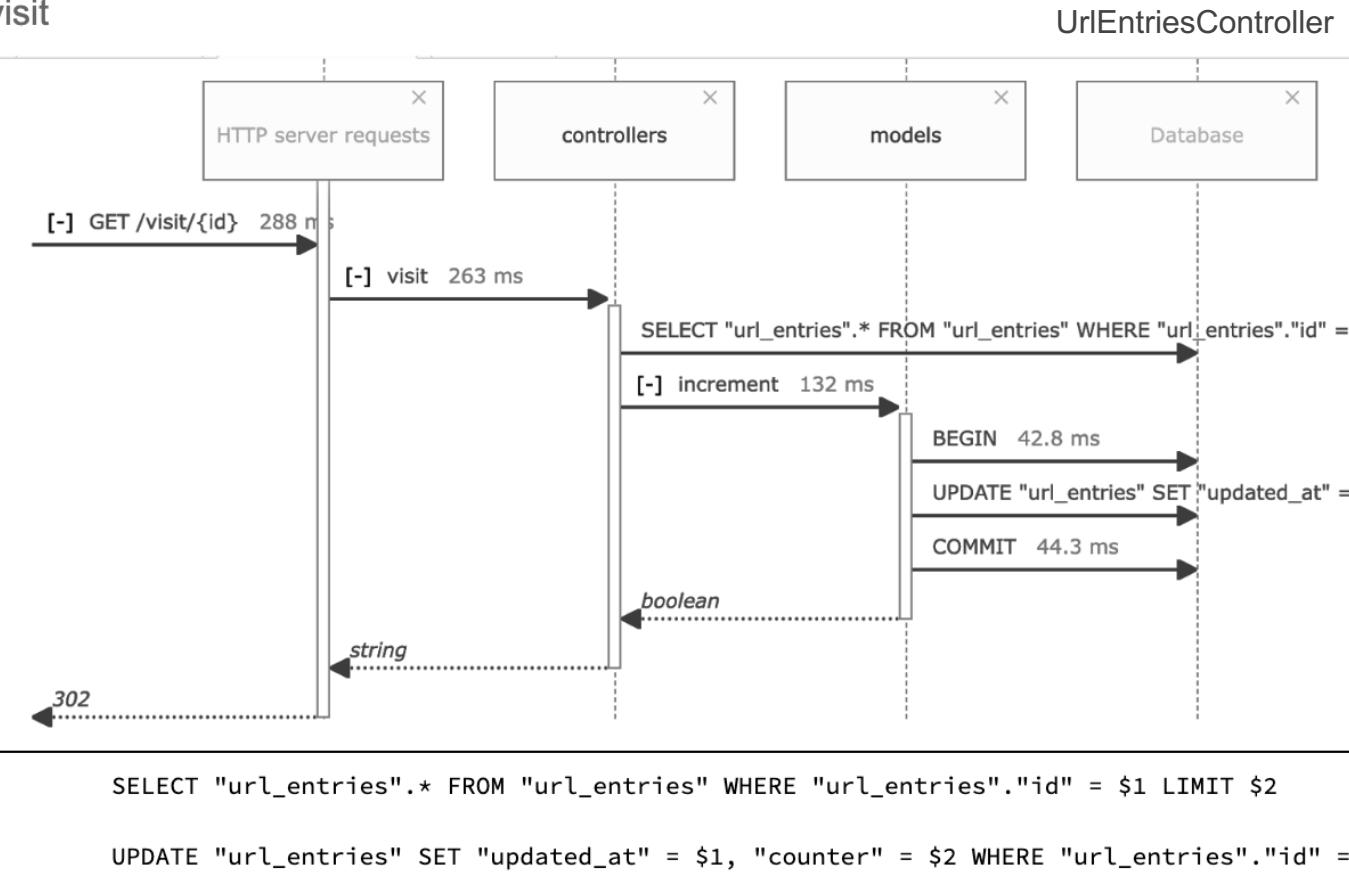
System Sequence Diagrams

These diagrams are a part of **Requirements**



Design Sequence Diagram: Request and Response

visit



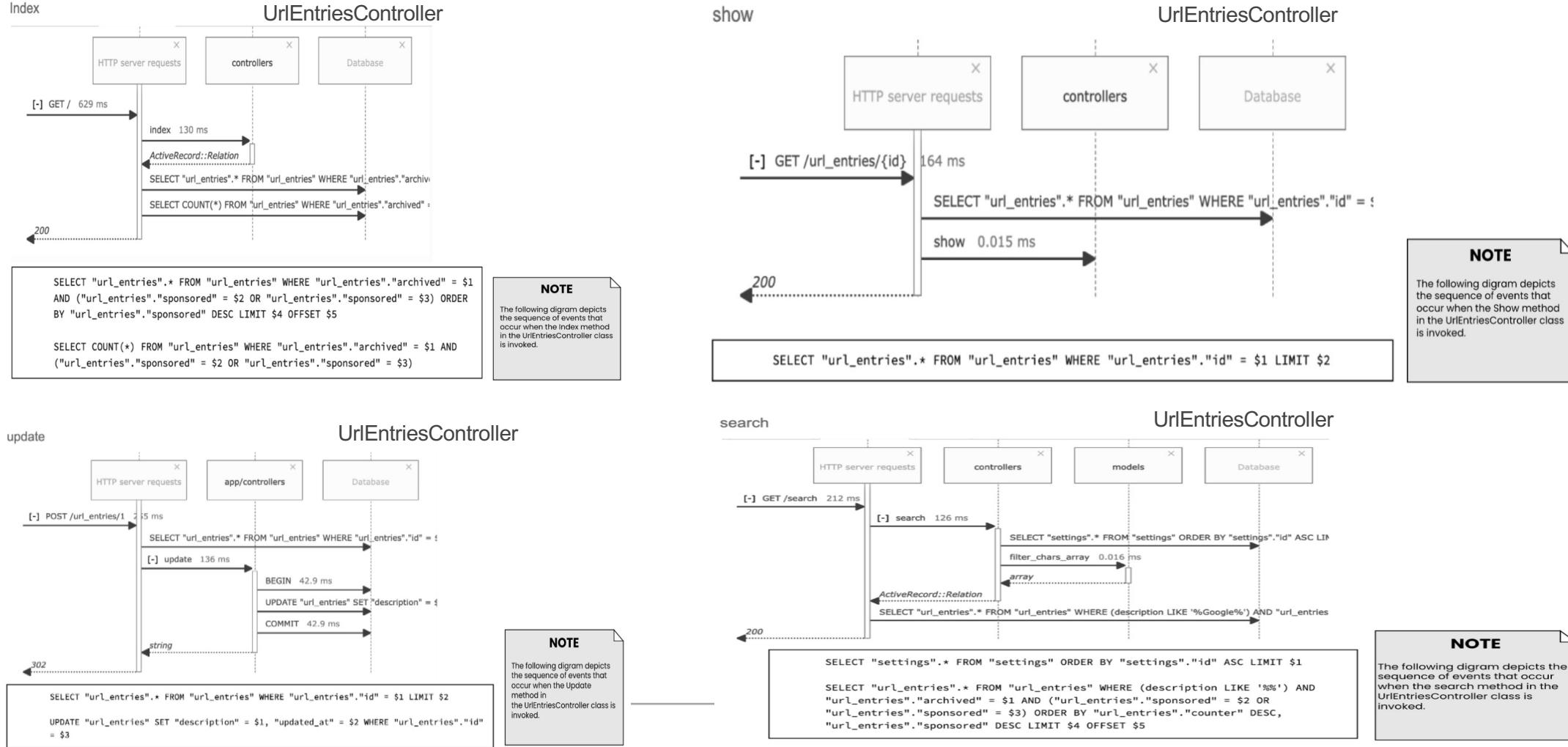
NOTE

The following diagram depicts the sequence of events that occur when the `visit` method in the `UrlEntriesController` class is invoked. The job of the `visit` method is to redirect users to their desired web page.

This diagram is a part of **Design**

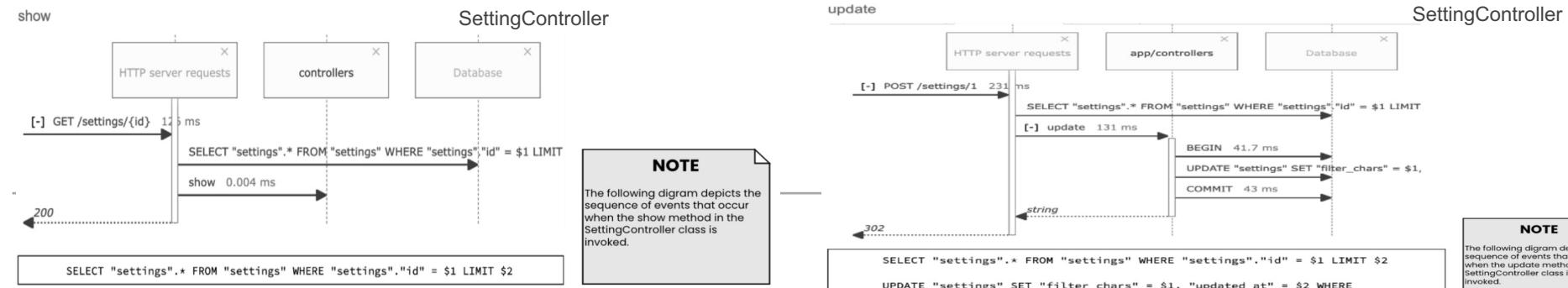
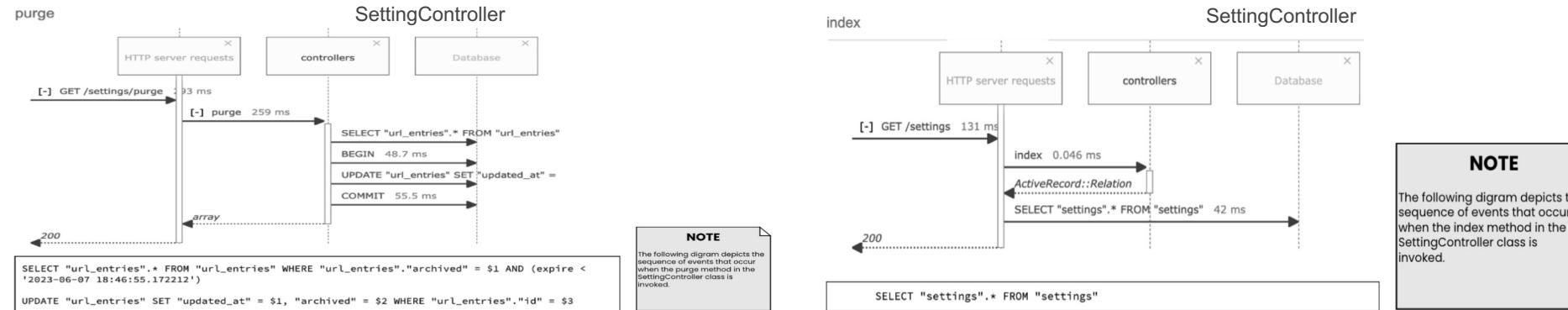
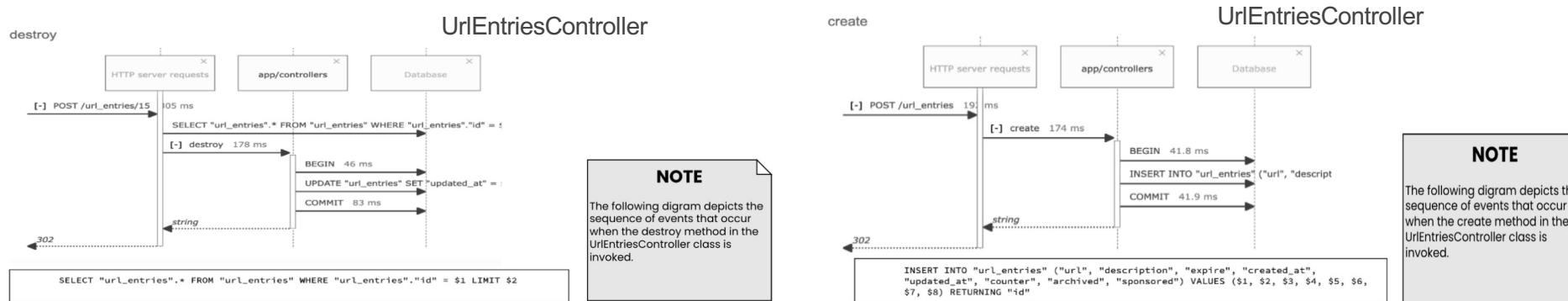
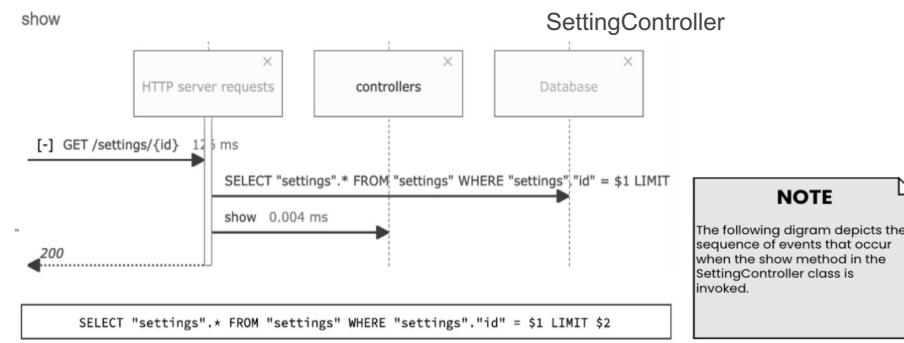
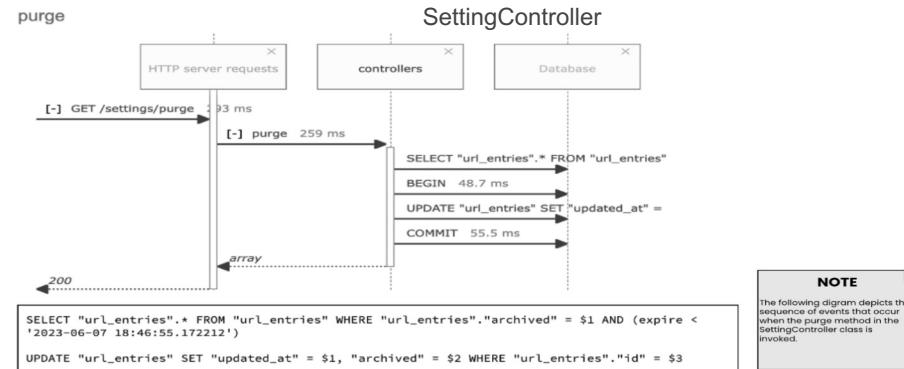
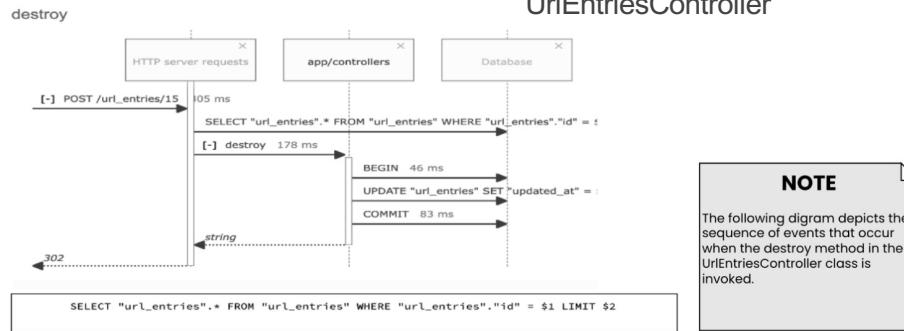
Design Sequence Diagrams

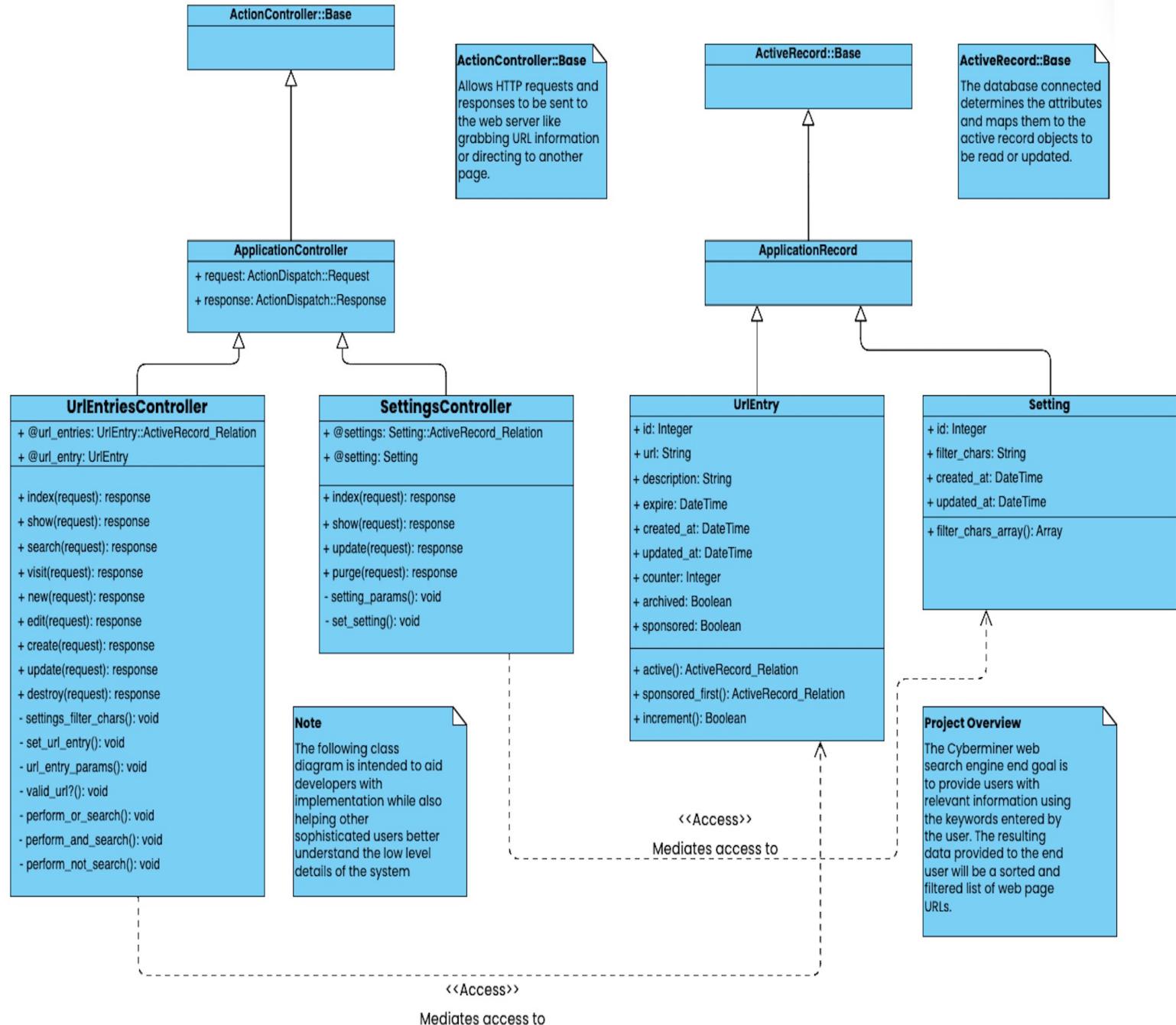
These diagrams are a part of **Design**



Design Sequence Diagrams

These diagrams are a part of **Design**

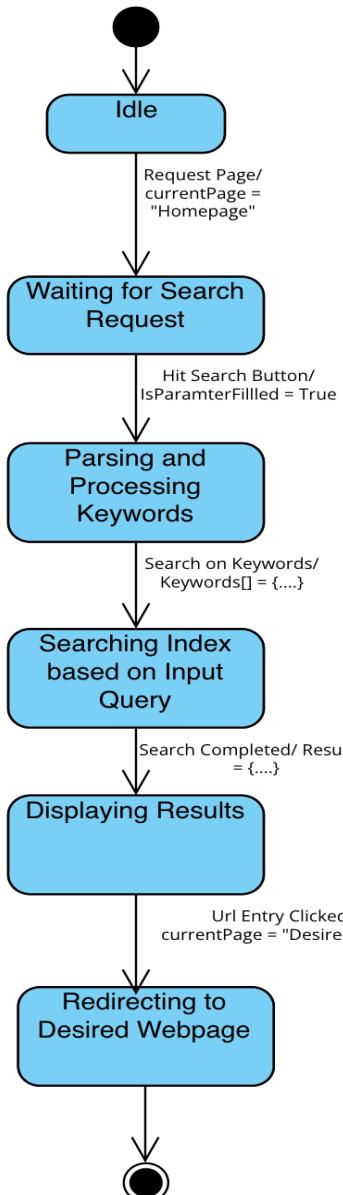




Design Class Diagram

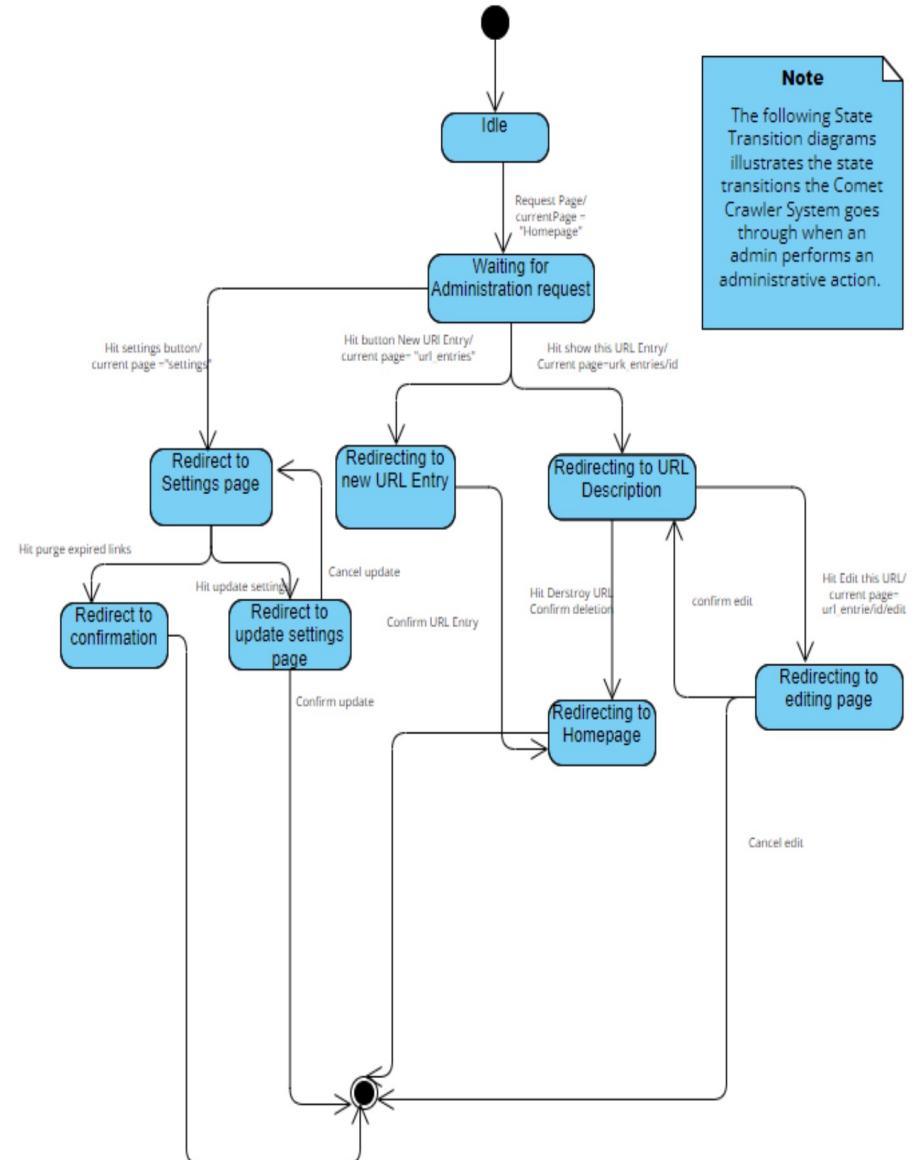
This diagram is a part of **Design**

State Transition Diagram for User Services



Note
The following State Transition diagrams illustrates the state transitions the Comet Crawler System goes through when a user performs a search.

State Transition Diagram for Administrator Services



Note
The following State Transition diagrams illustrates the state transitions the Comet Crawler System goes through when an admin performs an administrative action.

State Transition Diagrams

Search Results

Demo

Search for:

Query Type:

Sort Order:

Items Per Page:

[Clear](#)

SPONSORED

Url: <https://team1preliminaryprojectplan.tiiny.site/>

Description: Preliminary Project Plan Soft Copy

Clicks: 28

Expire: 2025-08-01 23:59:00 UTC