Interim Project I: Phase 1

Team 1: Areebah, Tyler, Darrien, Ilhaam, Nathan, Matthew

Home | New url entry | Settings | Readme | Preliminary Project Plan

Search Results

Demo

Search for:

Query Type: OR

Sort Order: Most Accessed

Items Per Page: 50

Search

Clear

SPONSORED
Url: https://team1preliminaryprojectplan.tiiny.site/

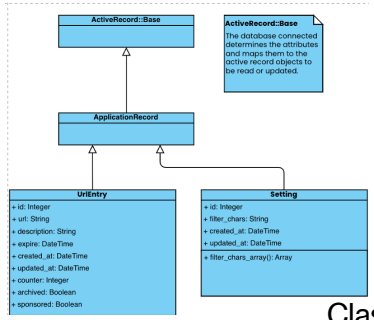
Description: Preliminary Project Plan Soft Copy

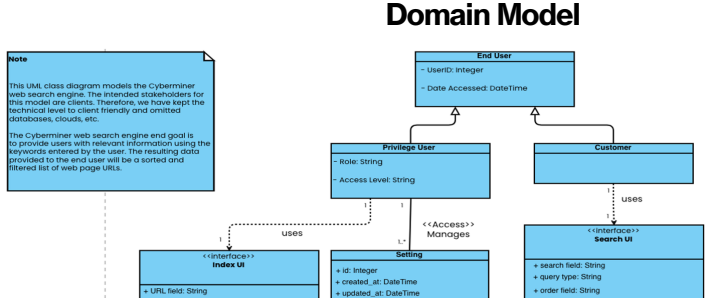
Clicks: 28

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

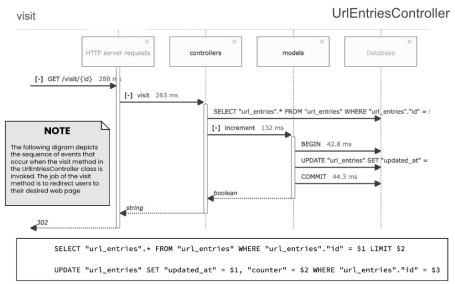
Business Modeling

Domain Model





Class Diagram for Stakeholders



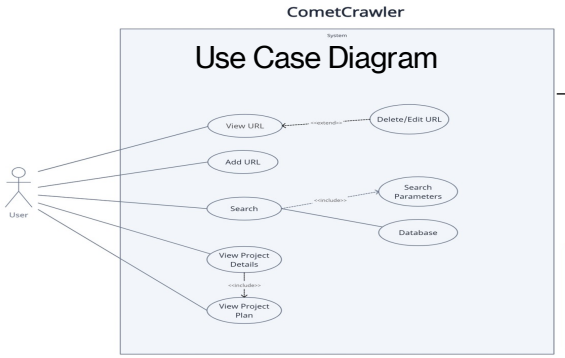
Sequence Diagram

NOTE

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

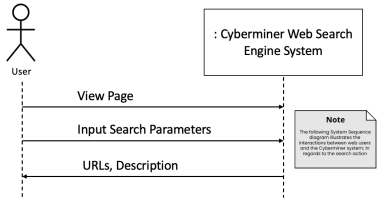
Requirements

Use-Case Model



We have one actor-User. The user can refer to the team members, the professor, or any person that has access to the project and the URL

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.



System Sequence Diagram

NAME	CometCrawler- search engine
BRIEF DESCRIPTION	User searches the database for relevant URLs to various websites.
ACTORS	User. The user can be a team member, professor, or anyone with access to the project and CometCrawler's URL.
PRE-CONDITION	Query type, Sort Order, Items per page. Default is provided for all of them.
BASIC FLOW	Search: - User enters keywords in search bar. - Keywords are interpreted and matched with database. - The results are posted according to the preconditions provided. Add URL: - User must select New URL Entry from Home page. - User provides the URL to be added, a description of the URL and an expiration date. - The URL entry is created when user clicks create URL Entry. View URL: - To view URL description, user selects view URL in the URL box. Destroy URL: - User views the URL by selecting the View URL button. - The user then selects Destroy URL button and confirms the deletion from the pop-up box. Edit URL: - User views the URL by selecting the View URL button. - The user then selects the Edit URL button. - The user then updates the information and confirms the edit. Visiting the webpage: - The user simply clicks on the URL to visit the page. Viewing Project description/Plan: - The user selects Preliminary Project plan button on top of the Homepage in the navbar.
EXCEPTION FLOWS	Search: - The user enters keywords that are not in the database. - The search will lead to 0 results.
POST-CONDITIONS	User can do a successful search. User can view/delete/update URL entries successfully

Use Case Template

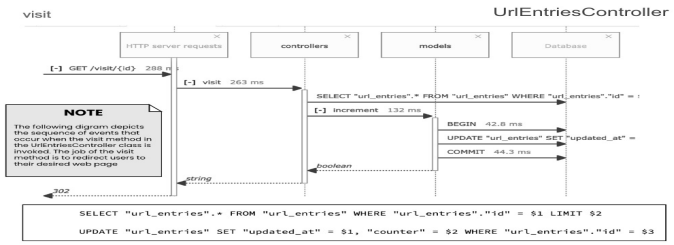
NOTE

The following template depicts and further explains the flow of events that are described in our use case diagram

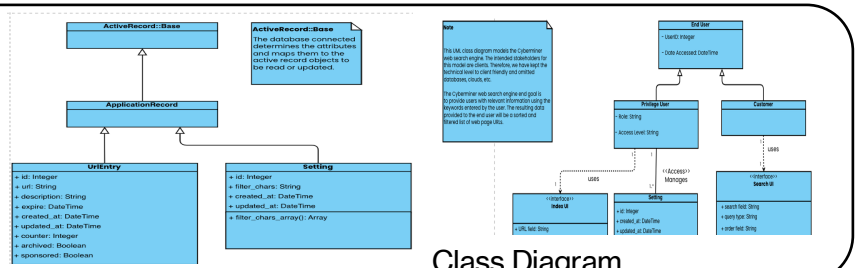
Project Overview
The Cyberminer web search engine end goal is to provide users with relevant information using the keywords entered by the user. The resulting data provided to the end user will be a sorted and filtered list of web page URLs.

Preliminary Project Document

Design



Sequence Diagram



Class Diagram

Implementation

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.

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

Testing

```ruby
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

    assert_redirected_to setting_url(Setting.last)
  end
```

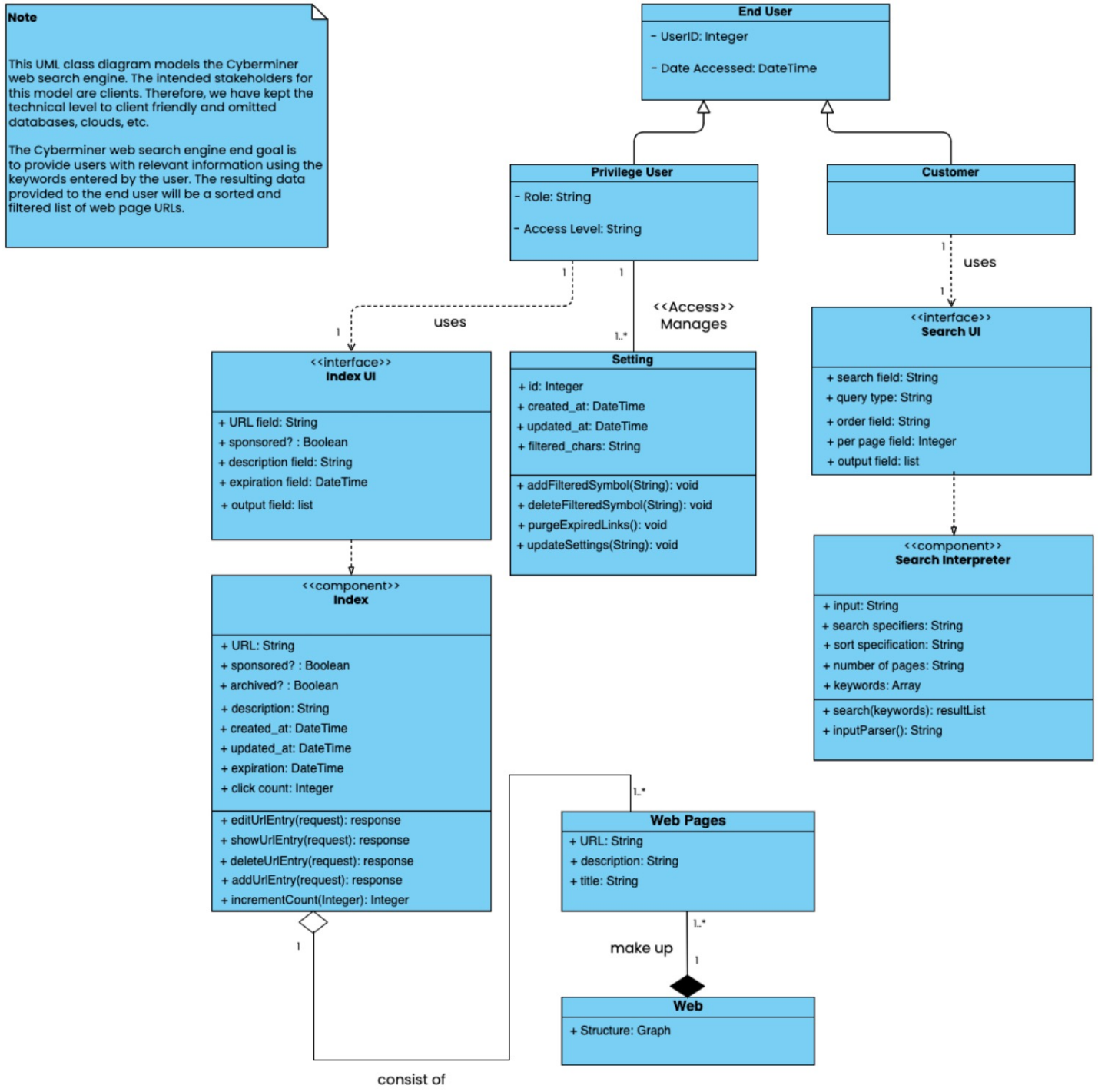
```ruby
class UrlEntriesControllerTest < ActionDispatch::IntegrationTest
  setup do
    @url_entry = url_entries(:one)
  end

  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
```

Test Case

URL	Description	Clicks	Expire	Sponsored?
https://team1preliminaryprojectplan.tiiny.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/tyharq	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

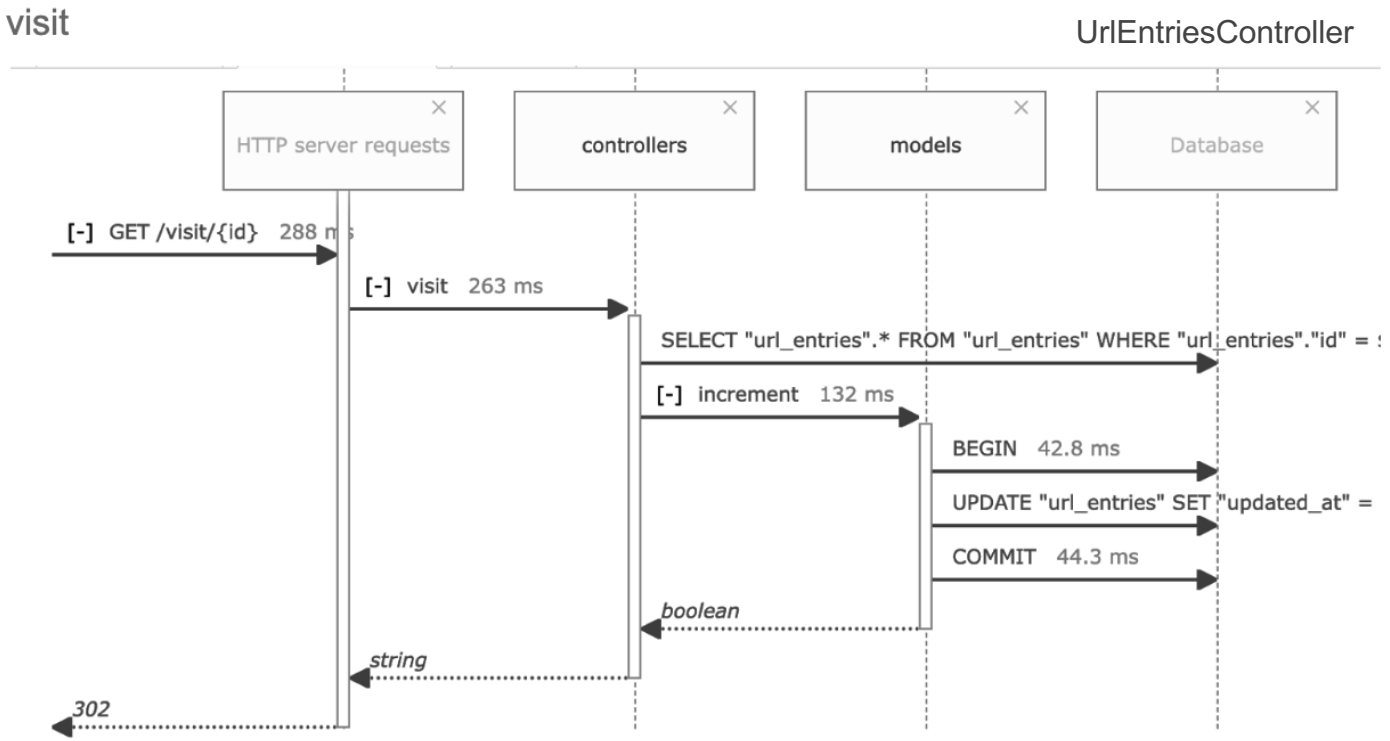
Indices Listing

Stakeholder Class Diagram : Client



Stakeholder Class Diagram : Developers

Sequence Diagram: Request and Response

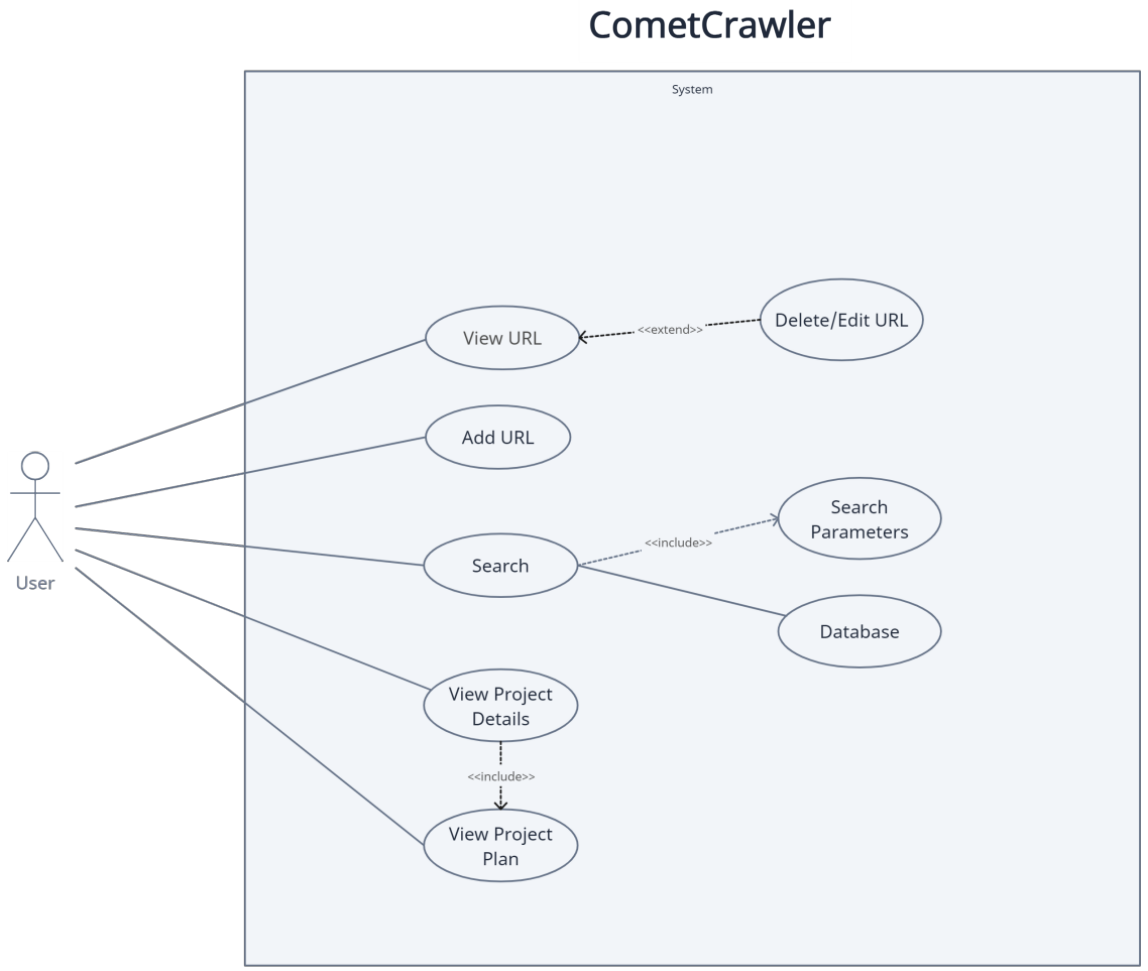



```
SELECT "url_entries".* FROM "url_entries" WHERE "url_entries"."id" = $1 LIMIT $2

UPDATE "url_entries" SET "updated_at" = $1, "counter" = $2 WHERE "url_entries"."id" = $3
```

NOTE

The following digram 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

Use Case Diagram



We have one actor-User. The user can refer to the team members, the professor, or any person that has access to the project and the URL

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.

Use Case Template

NAME	CometCrawler- search engine
BRIEF DESCRIPTION	User searches the database for relevant URLs to various websites.
ACTORS	User. The user can be a team member, professor, or anyone with access to the project and CometCrawler's URL.
PRE-CONDITION	Query type, Sort Order, Items per page. Default is provided for all of them.
BASIC FLOW	Search: - User enters keywords in search bar. - Keywords are interpreted and matched with database. - The results are posted according to the preconditions provided. Add URL: - User must select New URL Entry from Home page. - User provides the URL to be added, a description of the URL and an expiration date. - The URL entry is created when user clicks create URL Entry. View URL: - To view URL description, user selects view URL in the URL box. Destroy URL: - User views the URL by selecting the View URL button. - The user then selects Destroy URL button and confirms the deletion from the pop-up box. Edit URL: - User views the URL by selecting the View URL button. - The user then selects the Edit URL button. - The user then updates the information and confirms the edit. Visiting the webpage: - The user simply clicks on the URL to visit the page. Viewing Project description/Plan: - The user selects Preliminary Project plan button on top of the Homepage in the navbar.
EXCEPTION FLOWS	Search: - The user enters keywords that are not in the database. - The search will lead to 0 results.
POST-CONDITIONS	User can do a successful search. User can view/delete/update URL entries successfully

NOTE

The following template depicts and further explains the flow of events that are described in our use case diagram

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.