|  |
| --- |
| HDSSD |
| Requirements Specification (RS) |
| Ranking News Articles on Prestige |

|  |
| --- |
| Paul Horan  11/23/2017 |

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 14/10/2005 | 1 | Create | AB | X | X |
| 21/10/2005 | 2 | Update | CD |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Eamon Nolan | Lecturer |  |
| Michael Bradford |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[1 Introduction 4](#_Toc316977392)

[1.1 Purpose 4](#_Toc316977393)

[1.2 Project Scope 4](#_Toc316977394)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc316977395)

[2 User Requirements Definition 4](#_Toc316977396)

[3 Requirements Specification 4](#_Toc316977397)

[3.1 Functional requirements 4](#_Toc316977398)

[3.1.1 Use Case Diagram 5](#_Toc316977399)

[3.1.2 Requirement 1 <name of requirement in a few words> 5](#_Toc316977400)

[3.1.3 Requirement 2 <name of requirement in a few words> 6](#_Toc316977401)

[3.2 Non-Functional Requirements 7](#_Toc316977402)

[3.2.1 Performance/Response time requirement 8](#_Toc316977403)

[3.2.2 Availability requirement 8](#_Toc316977404)

[3.2.3 Recover requirement 8](#_Toc316977405)

[3.2.4 Robustness requirement 8](#_Toc316977406)

[3.2.5 Security requirement 8](#_Toc316977407)

[3.2.6 Reliability requirement 8](#_Toc316977408)

[3.2.7 Maintainability requirement 8](#_Toc316977409)

[3.2.8 Portability requirement 8](#_Toc316977410)

[3.2.9 Extendibility requirement 8](#_Toc316977411)

[3.2.10 Reusability requirement 8](#_Toc316977412)

[3.2.11 Resource utilization requirement 8](#_Toc316977413)

[4 GUI 8](#_Toc316977414)

[5 System Architecture 8](#_Toc316977415)

[6 System evolution 8](#_Toc316977416)

# Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of a Search Engine for News. The Search Engine will bring up reliable news sites/articles. As we are trying to exclude fake news and the sites that provide these fake articles. The search engine should be optimized to exclude all fake news outlets and their articles.

The intended customers are people who keep up to date on current events/ worldwide news. Eliminating fake news from their feeds. The Search Engine will be created using different languages such as Java, Javascript, C++, Python and MySql, Service API’s will also be used.

I have always been interested in the likes of Google and Yahoo, how they work and the algorithms used to create software that is simple to use. Using the likes of Google and Yahoo is very easy, anyone could use it. But it is the background of this that I am interested in the most.

## Project Scope

The scope of the project is to develop a Search Engine to provide the users with News. The system shall have a ……………

This section also details any constraints that were placed upon the requirements elicitation process, such as schedules, costs, or the software engineering environment used to develop requirements.

# User Requirements Definition

The Search Engine should be as easy to use as Google. Using a search function for News from around the world. Real news from around the world will be provided as End Users do not want to read Fake News articles. The simple interface and use is extremely important.

The Search Engine will be optimized to show news outlets websites from around the world. Articles will be organized by category. The Search function will pick up what the user is looking for and will output news related to their search.

# Requirements Specification

Experienced controllers shall be able to use all the system functions after a total of one hour training session. After this training, the average number of errors made by experienced users shall not exceed two per day.

## Functional requirements

1. User Login

User Sign up/ Sign In is needed to use the system. Will allow users to keep track of their search history.

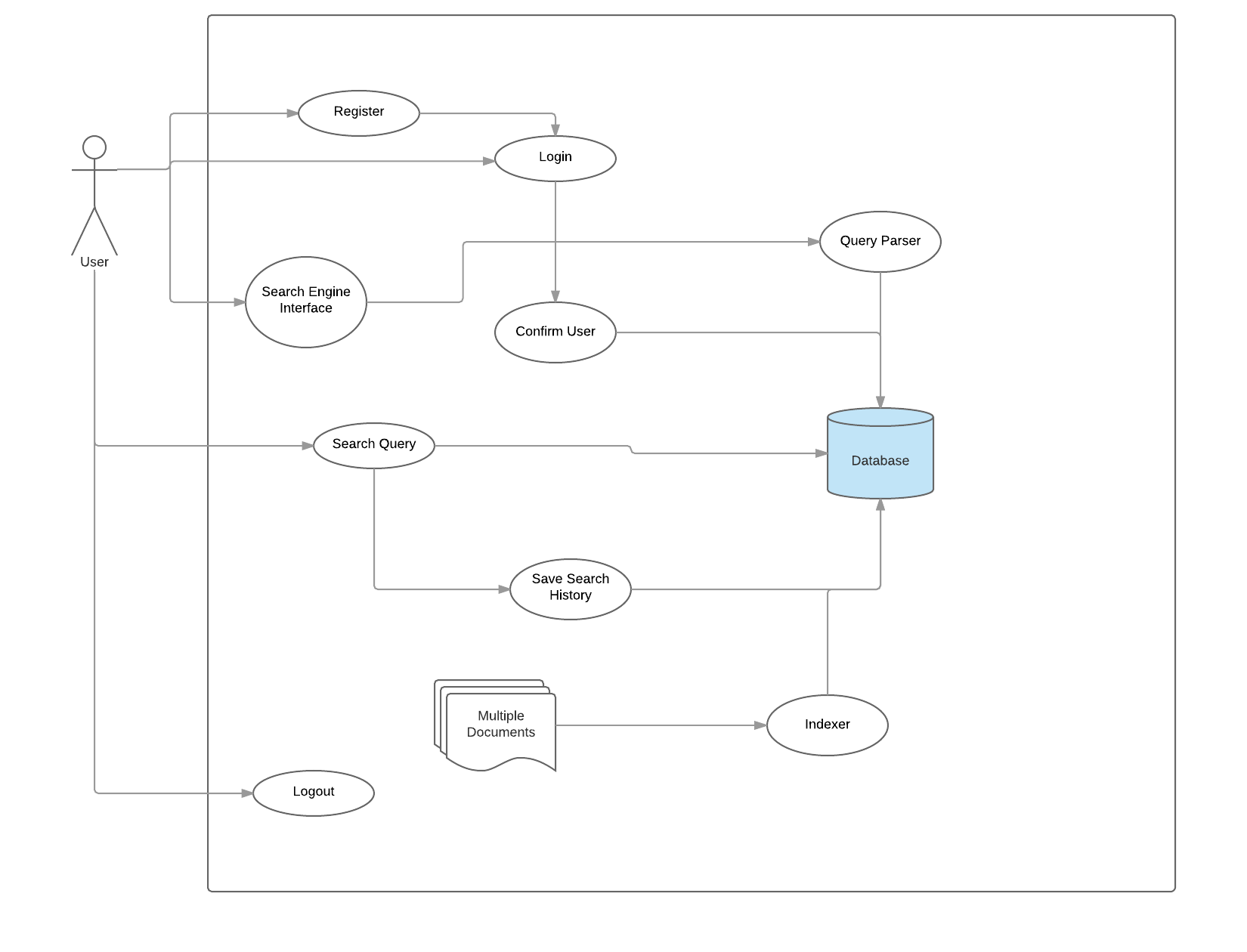
1. Search Function

The Search function should be quick and simple. With a spell checker implemented to suggest alternate spellings.

### Use Case Diagram

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

The Use Case Diagram provides an overview of all functional requirements.



### Requirement 1: User Login

The user should be able to register/Login to the application

#### Description & Priority

User will need to Register / Login to use the Search Engine. This will also aid in saving the search history of the user.

#### Use Case

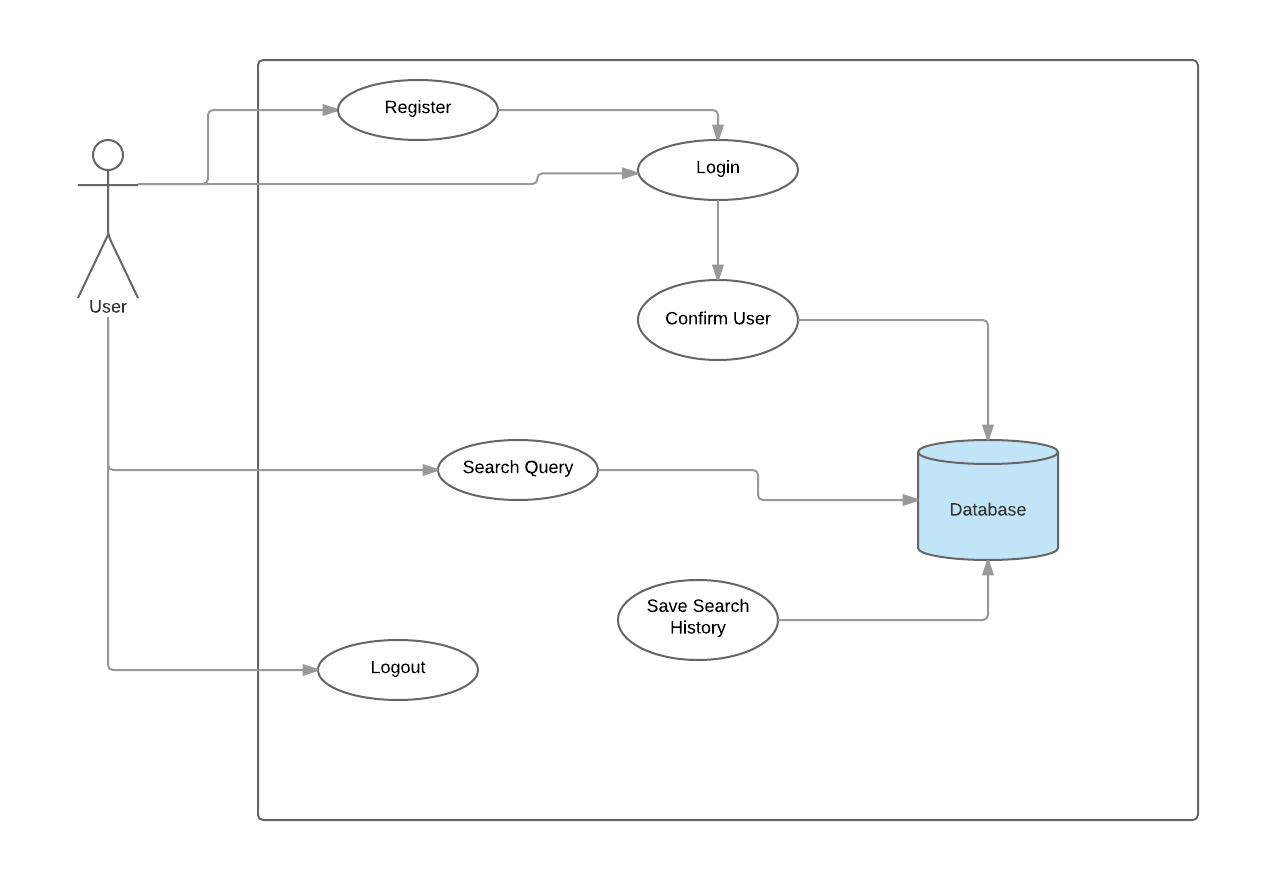
**Scope**

The scope of this use case is to show the user signing up/logging into the system.

**Description**

This use case describes the process of the user either Signing Up or Logging in. Then they can access the Search Engine.

**Use Case Diagram**

**Flow Description**

**Precondition**

The system is in initialisation mode

**Activation**

This use case starts when a User opens the application

**Main flow**

1. The User launches the application

2. The User enters their credentials

3. The system verifies the details and the user is logged in

4. The User is presented with the Search Engine/ Home Screen

**Alternate flow**

If the user is unregistered the system will perform the following.

1. The system detects an unregistered user

2. The User is asked to register

3. The User completes the registration

4. The User can now login and access the search engine and provided features

**Exceptional flow**

The user decides not to register and closes the application

1. The system goes into a resting state.

**Post condition**

The system goes into a wait state until it receives a new interaction from the user.

### Requirement 2: Search Function

#### Description & Priority

The search function should be very easy for the end users to use. Allowing for user based errors such as typos/spelling errors a spell check should be implemented to allow the user to have an easy going experience while searching.

#### Use Case

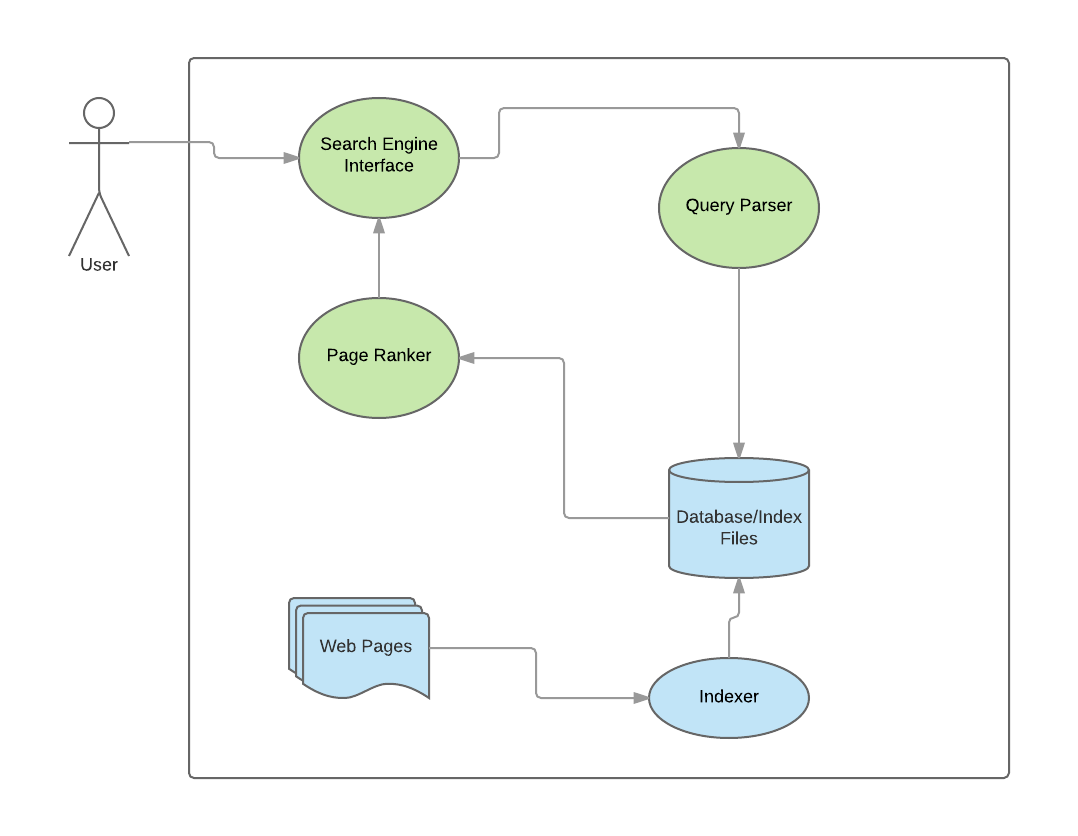
**Scope**

The scope of this use case is to show the steps of the search when submitted.

**Description**

This can only be completed if the user has previously created an account successfully, once they have logged in they then will be presented with the Home screen and will be able to also log out therefore ending the session.

**Use Case Diagram**



**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when a user has logged into the application

**Main flow**

1. User Launches Engine
2. The User registers or enters their credentials
3. User Searches a Query
4. Raw Query is sent to the query parser
5. Clean and processed query tokens are searched in the Index
   1. Read the webpages and make index
   2. Write index files to disk
6. Ranked list of items are retrieved
7. User can now browse through retrieved websites or terminate session.

**Termination**

The system restarts and shows the home page.

**Post condition**

The system goes into a wait state

**List further functional requirements here, using the same structure as for Requirements 1 & 2. Most systems would have at least five main functional requirements.**

## Non-Functional Requirements

1. Page Rank System

This will be implemented to rank each page by the prestige of the Website and Author.

1. Storage

The Database must be able to store all of the indexes, User Accounts and User Search History.

1. Optimization

The search engine will need to be optimized correctly, to eradicate the fake news websites and articles.

### Performance/Response time requirement

Performance should run smoothly, response time of under 5 seconds.

### Data Requirements

The project will be dealing with a lot of data of various types. I was planning on using MySQL to create this database. But I am also looking into Google’s Firebase which is a free online database, this is a JSON structured database The User credentials may be stored in one database and the indexes in another. If this turns out to not be a viable option it will be created using MySQL.

### Security requirement

User Login to view their recent search history. Having a log in can make users feel more secure with what they are using.

### Reliability requirement

Search Engine should be able to show the user the correct search results. The aim is by ranking the News Sites, Articles and Authors by prestige we can take the fake news providers out of the users search. Either by ranking them very low on the list or the main goal, take them out completely.

### User Requirements

Simplicity: Any possible user should be able to pick up the application and use it without difficulty.

Devices: The availability for the application to be created for multiple types of devices, Desktops, Laptops, Mobile/Tablet. The use of bootstrap will be used for this.

### Reusability requirement

Multiple reuse, option to search again after inputting a search

### Resource utilization requirement

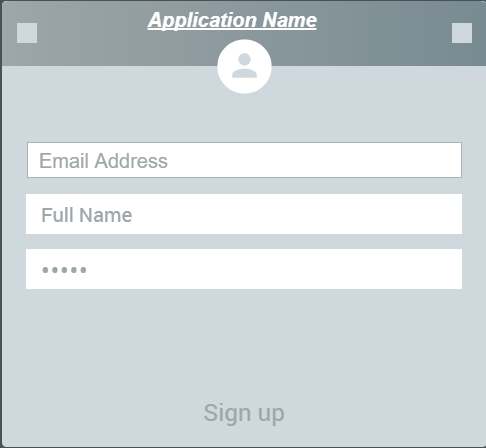
Currently researching.

# Interface requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include APIs, web services, shared memory, data streams, and so forth. Most systems would have a GUI. Add more subsections for other interfaces as reuired.

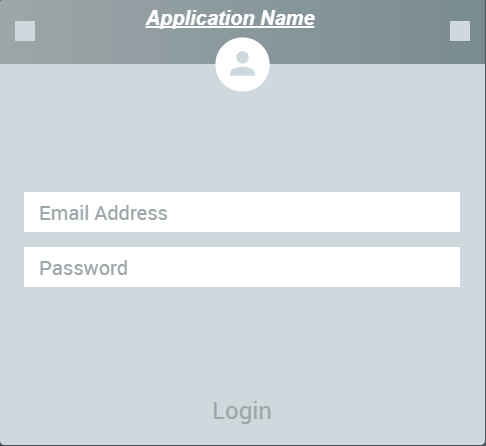
## GUI

User Registration.



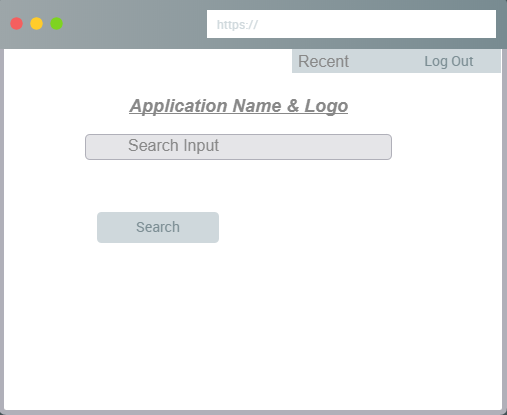
The registration page is straight forward. Asking for the users Email Address, Name and a password. After setting this up the user will be able to log in and access the home page.

User Login



The Log in page much like the registration page will be the first page the user sees. There will be an option to registerif not already signed up.

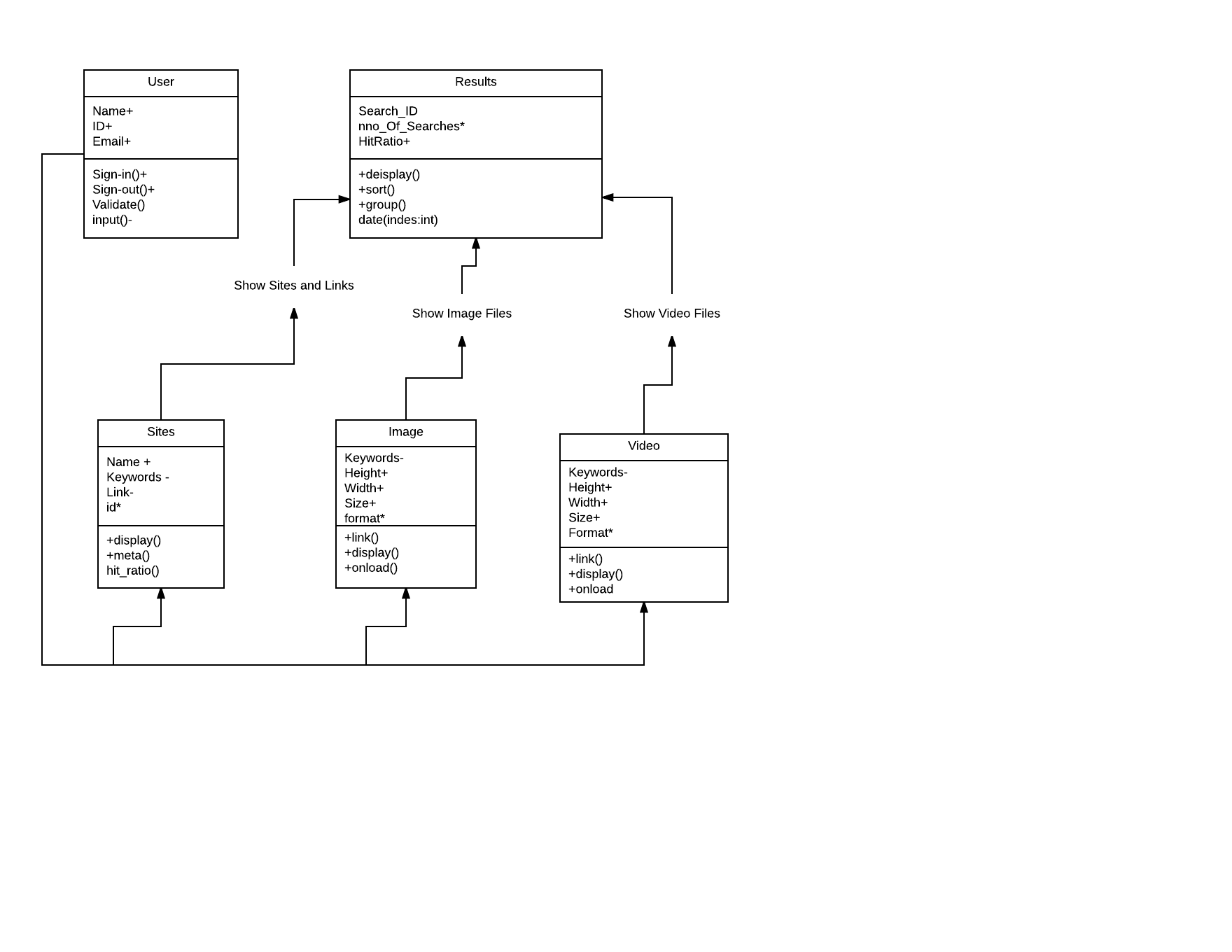
Home Screen



Once registered and logged in the user will come to the home screen. Much like google it will be plain and simple to use. There is no need for anything eye catching here. Once the input is searched the results will appear on the page in a linear fashion. With the option to search again in the header of the page.

## Application Programming Interfaces (API)

# System Architecture



For the search engine we are looking for it to carry out normal search engine functions. The diagram above outlines the user from logging in. Once they run a search this is what should happen. From their search we can see possible results.

This diagram will be modified at a further stage in the project. Expanding this by adding other functions that will help us reach the goal of eradicating fake news sites.

# System Evolution

If the system turns out to be successful, by taking out the fake news publishers altogether then I believe this could be brought into social media. With many fake accounts across all social media platforms these companies could take into account that my search engine has disregarded these fake publishers and possibly tweak the idea to locate fake accounts. As the amount of Fake accounts across all social media platforms is in the millions. Maybe the search algorithm could be changed to locate the accounts by using different factors.

Possibly searching aspects of the email address linked to the account to see if it is an active email account. Checking the accounts time online/logged in. Following certain traits like this could lead into locating these fake accounts. It could possibly be taken too this level, but my application will need to successfully eradicate the fake news publishers before anything like this could happen.