



Analysis Report Assignment

Group 13

Göktuğ Yılmaz

Fatih Sultan Mehmet Çelik

Mehmet Anıl Akgül

Bensu Şeker

Introduction	2
Proposed system	3
Overview	3
3.2 Functional Requirements.....	4
3.3 Nonfunctional Requirements.....	6
3.4 Pseudo requirements	7
3.5 System models.....	7
3.5.1 Scenarios.....	7
3.5.2 Use case model	9
3.5.3 Object and class model.....	10
3.5.4 Dynamic models.....	11
3.5.5 User interface - navigational paths and screen mock-ups	15
4. Glossary	18
5. References	19

Introduction

With artificial intelligence, it is feasible to make sense of this vast amount of data. By making sense of the data and processing it with artificial intelligence, it is feasible to utilize it in practically any industry. With data analyzed by artificial intelligence, it is feasible to achieve superiority in the military, make significant advancements in energy efficiency, and even have intelligence capacities that are historically unparalleled. Alternately, smart urbanization applications may be used to improve human lives and achieve significant advantages in a variety of fields, including corporate intelligence, cyber security, and finance. Effective usage of artificial intelligence and data collaboration is important. The link between artificial intelligence algorithms and data sets is significant and essential. Without a vast number of carefully collected data sets, artificial intelligence is unable to draw conclusions that are both correct and meaningful in the moment. To protecting corporate privacy, most institutions do not share their data when it comes to the acquisition of the data required for artificial intelligence. Nevertheless, it is now possible to collect data from international platforms.

Our website project that serves as a repository for datasets for data science and machine learning. We then utilize that same computer to function as the website's server and as a database, performing relevant dataset searches on sites that share artificial intelligence data. We will display datasets on this website that people typically seek for on other websites (e.g., Kaggle, Google dataset search). Using keyword extraction, this is what we intend to do. The most used and significant words and expressions are mechanically extracted from a page using a text analysis approach called keyword extraction. It aids in identifying the primary ideas discussed in texts and identifying their key ideas. "ML dataset repositories," or collections of datasets for machine learning or data science applications, are maintained by the "Data Cosmos" initiative. Since many of the data sets are "open source," anyone anywhere in the world can access them. A computer will be used as both a database and a server for the website as we move forward. Ten websites with datasets were found. For instance, we plan to write Java code that can retrieve these datasets from Google or Kaggle automatically.

2. Current system (if any)

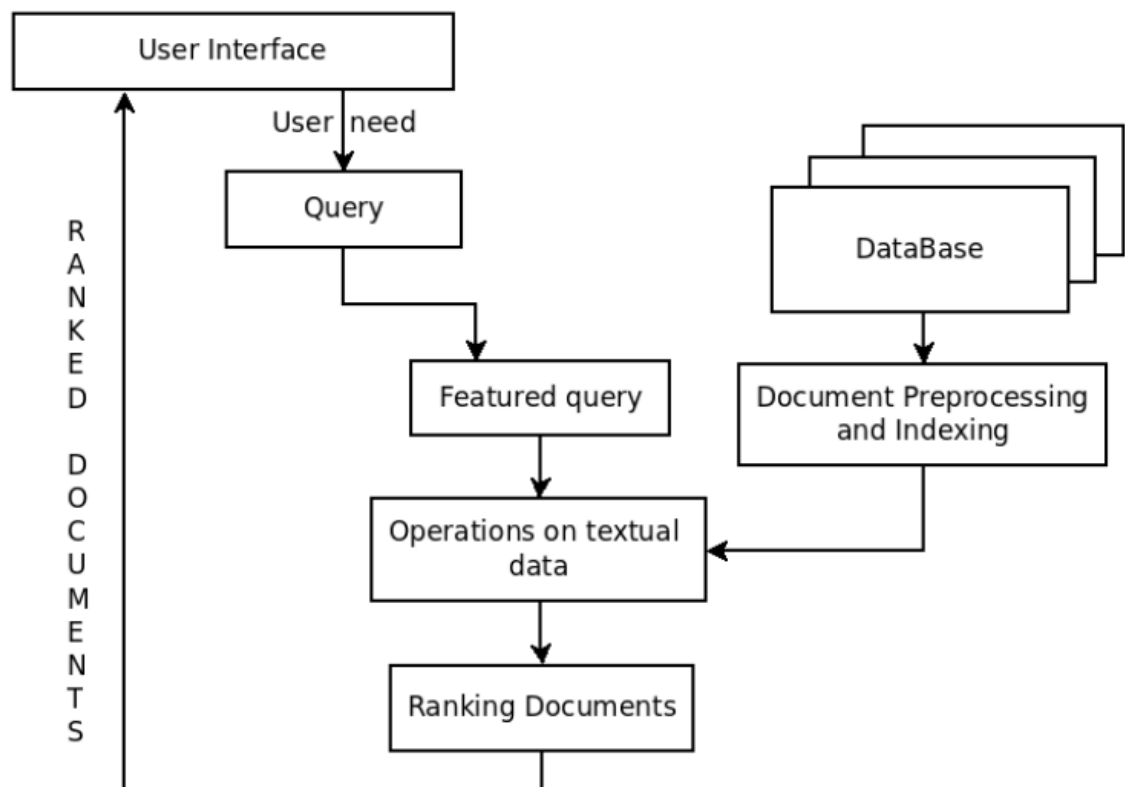
Proposed system

Overview

The client submits an HTTP request to get access to a resource. The resource's encoded data is returned by the server in the form of an HTTP response. Both REST message types are self-descriptive, which means they include instructions on how to be read, understood, and handled.

important components;

- Client or piece of software that launches communication on a user's PC or mobile device;
- Server that makes its data or features accessible through an API.
- Resource, which is any type of material the server may give the client; and a server (datasets).
- Providing the material with use information retrieval.



3.2 Functional Requirements

The project manages "ML dataset repositories," or collections of datasets for data science or machine learning applications. The fact that many of the data sets are "open source" makes them available to everyone worldwide. We will go on by using a computer as both a database and a server for the website. 10 dataset-containing websites:

- I. Kaggle
- II. UCI
- III. FiveThirtyEight
- IV. Quandl
- V. Data.World
- VI. Data.gov
- VII. Reddit
- VIII. Socrata
- IX. Google Dataset Search
- X. Lionbridge AI

- In a client-server system, a resource is a crucial component that must be referred to. HTML pages, images, text files, videos, and other types of content are all treated as resources by the architecture. The REST server offers resource access, while a client is used to access and modify resources. Its resources are all uniquely identified by URIs, or uniform resource identifiers.
- A Uniform Resource Identifier (URI) can be used to identify the resource (URI). The most widely used protocol for interacting with external systems in web-based applications is HTTP. A URI may be used to locate a particular resource.

GET	To get a collection or a single resource
POST	To create a new resource
PUT	To update an existing resource
DELETE	To delete a collection or a single resource

- GET retrieves a representation of the resource at the specified URI. The body of the response message contains the details of the requested resource.

- POST creates a new resource at the specified URI. The body of the request message provides the details of the new resource. Note that POST can also be used to trigger operations that don't actually create resources.
 - PUT either creates or replaces the resource at the specified URI. The body of the request message specifies the resource to be created or updated.
 - PATCH performs a partial update of a resource. The request body specifies the set of changes to apply to the resource.
 - DELETE removes the resource at the specified URI.,
-
- To analyze the contents of the sources (documents);
 - To represent the contents of the analyzed sources for matching with the users' queries;
 - To match the search statement with the stored database;
 - To retrieve the information that is relevant; and
 - To make necessary adjustments in the system based on feedback from the users.
-
- Making an extremely abstract representation of the object-oriented program. Without understanding how the data is organized, mapping defines the link between an item and the data.
 - A connection between the application and the SQL code required to manage data activities may then be made using the model. We can save a ton of time by not having to completely rewrite the "plumbing" sort of code.

Due to the volume of database interfaces needed, the design will need to take into consideration the fact that many of the functionalities involve storing, editing, or reading evaluation forms. Concurrent readings from and writes to the database must be supported by the system. Additionally, the system may have the need to interface with several external APIs. The system will probably need to interface with tools for generating reports and forms from outside sources, even though the specific services are unknown at this time. Architecturally speaking, these features are important because the system should be built in a modular way that makes it simple to swap in and out external services to handle future updates.

Testing

Unit testing: It enables us to determine if each piece of code is functioning correctly or not.

Integration testing: It concentrates on the project's development and design. We must check to verify if the integrated equipment are operating faultlessly.

System testing: Using this technique, the entire project is assembled and tested at once. This testing approach examines the usability, security, and portability, among other things.

3.3 Nonfunctional Requirements

Searching every dataset published by over 10 different sources large and small in the last 4 years. Returns datasets for data scientist, programmes etc. It is perfect for use with anywhere someone want to use live up-to-date datasets. Repository website containing datasets for machine learning and data science. The necessary datasets will be executed in accordance with the search terms entered on the websites where artificial intelligence data is shared. The most commonly used and significant words and expressions are mechanically extracted from a page using a text analysis approach called keyword extraction. It aids in identifying the primary ideas discussed in texts and identifying their key ideas. Many of the data sets are "open source," making them accessible to everyone in the globe. Since many of the data sets are "open source," anyone anywhere in the world can access them.

- **Correctness:** The software's capacity to adhere to the specification.
- **Performance:** How easily the program completes the tasks it is intended to complete.
- Typically, reaction time or throughput are monitored.
- **Reliability:** The software's capacity to carry out its necessary tasks under specified circumstances for a predetermined amount of time.
- **Robustness:** The software's capacity for handling execution-related faults.
- **Scalability:** The software's capacity to gracefully manage increasing workloads.
- **Security:** How well the program defends against threats.
- **Usability:** The simplicity with which a particular user may utilize the program to accomplish a certain goal.

3.4 Pseudo requirements

Uniform interface

In the system, a resource should only have one logical URI that may be used to access related or extra data. It is usually preferable to use a website as a synonym for a resource. Additionally, the representations of resources throughout the system should adhere to predetermined standards like naming conventions, link formats, and data formats (XML or/and JSON).

Client–server

A server is someone who retains the resources but is not concerned with the user interface or user status, whereas a client is someone who requests resources but is not concerned with data storage, which is kept internal to each server. They are able to change on their own. Client and server don't need to have any knowledge of business logic or front-end user interface.

Cacheable

A client is someone who seeks resources but is unconcerned with data storage, which is maintained internal to each server, whereas a server is someone who keeps the resources but is unconcerned with the user interface or user status. They have the capacity to alter on their own. Business logic and the front-end user interface are not required to be understood by the client or server.

Layered System

There may be several intermediary servers between the client and the final server, but each tier only has knowledge of its own local layer.

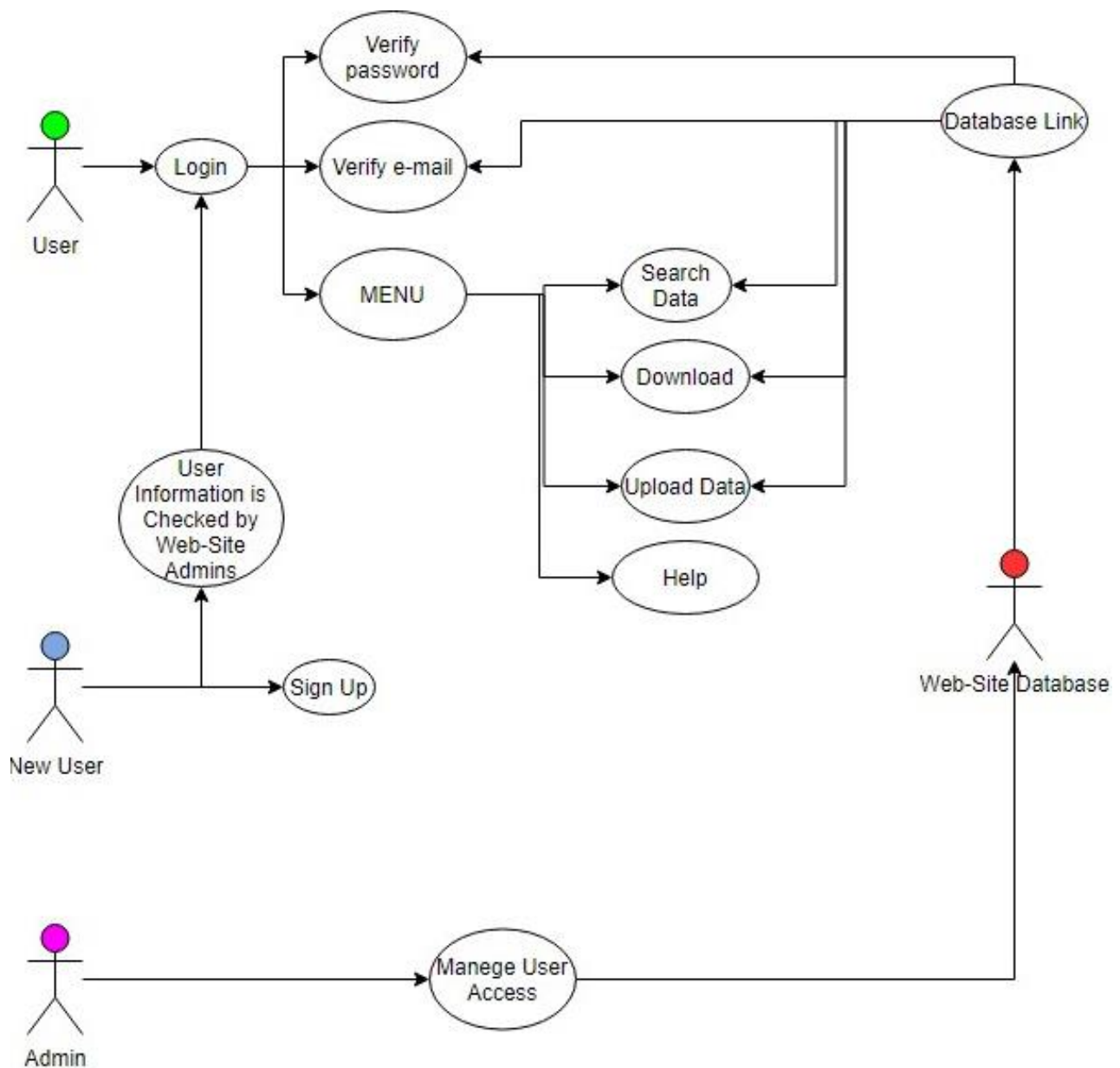
3.5 System models

3.5.1 Scenarios

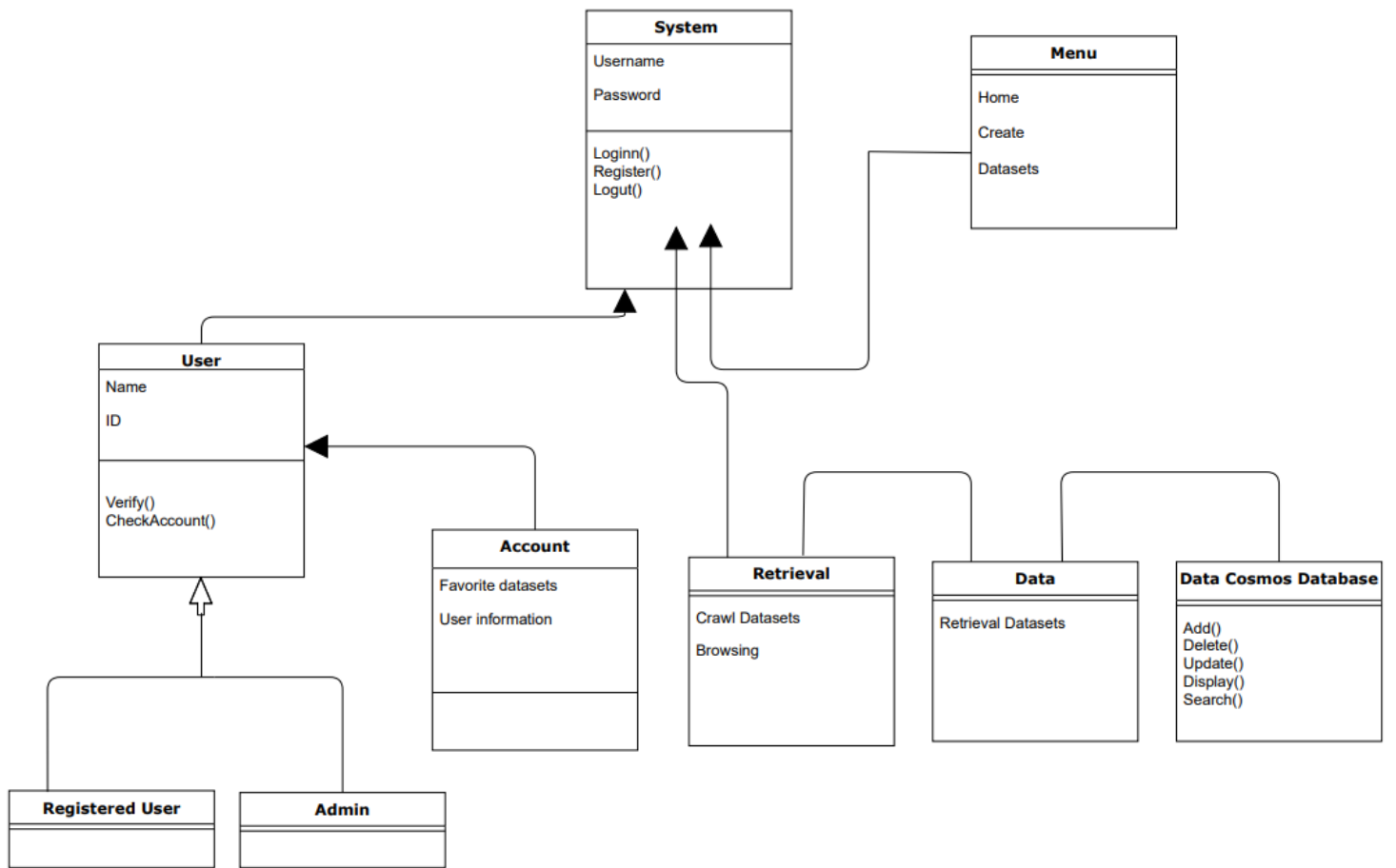
- user will be able to login with a valid username and valid password.
- user cannot login with a valid username and an invalid password.
- the 'Forgot Password' functionality.
- messages for invalid login.
- correct data is getting saved in the database upon a successful page submit.
- values for columns that are not accepting null values.
- For every database add/update operation logs should be added.

- User Sign in
- User Sign out
- Click profile button
- Home page button
- Dataset button
- Search datasets bar
- Can search dataset in search bar.
- API requests to the Web Server.
- Send the request using either XML or JSON
- Add only required request headers
- Add only required fields in the request with some correct values
- Send POST request
- Verify the response code
- Verify the body of the response
- Verify service-specific headers
- POST object in one test, GET and verify in another one

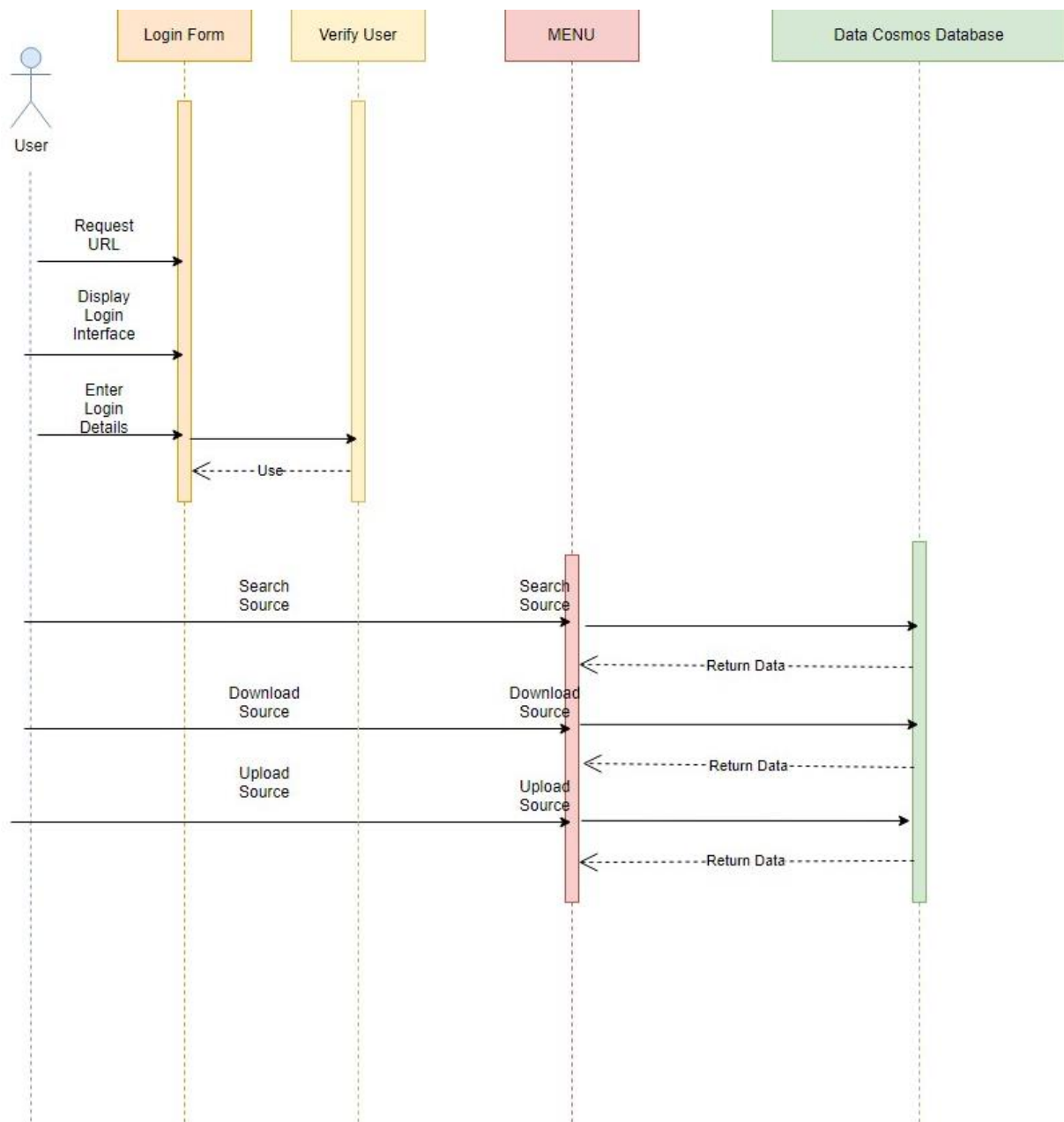
3.5.2 Use case model

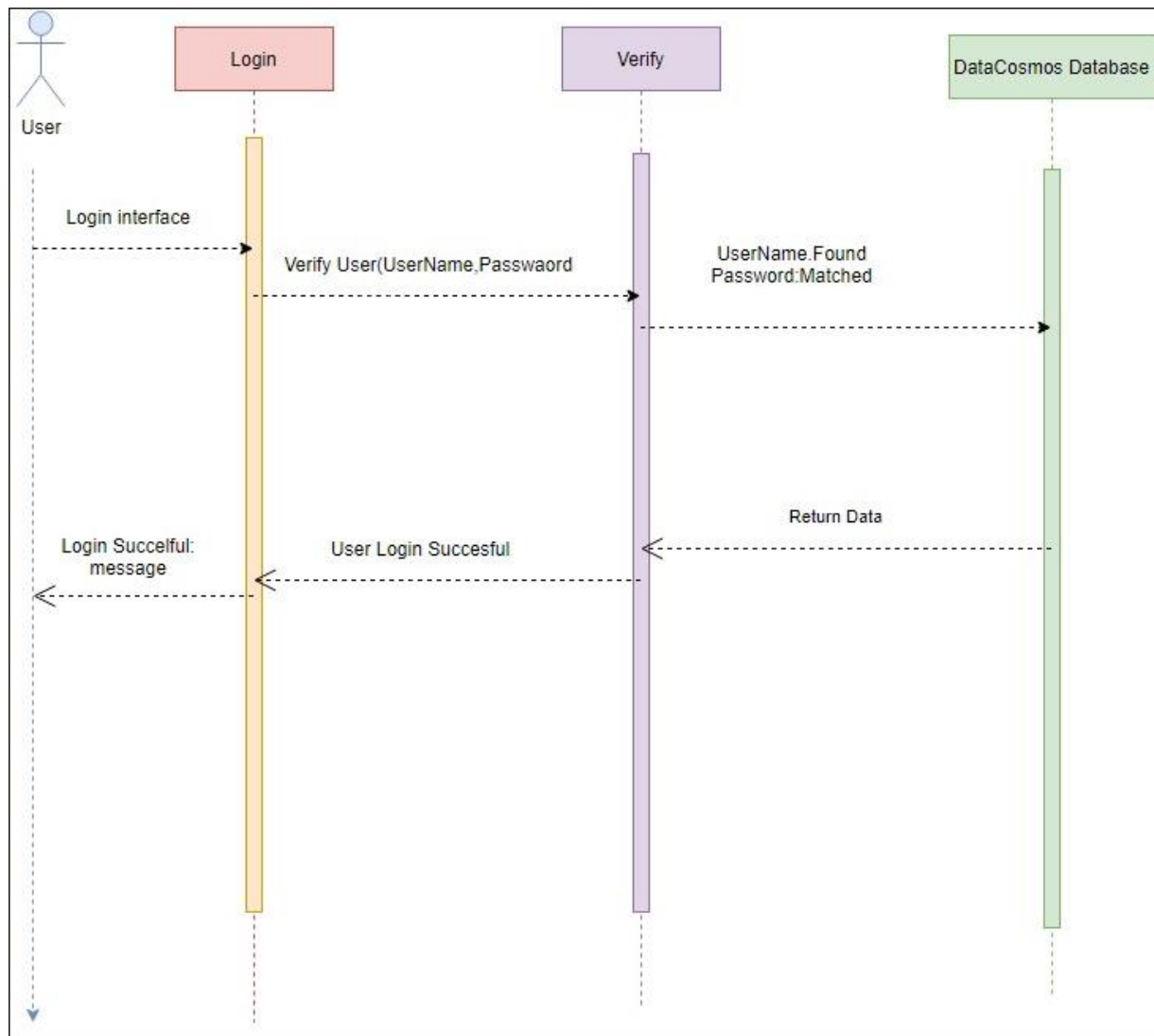


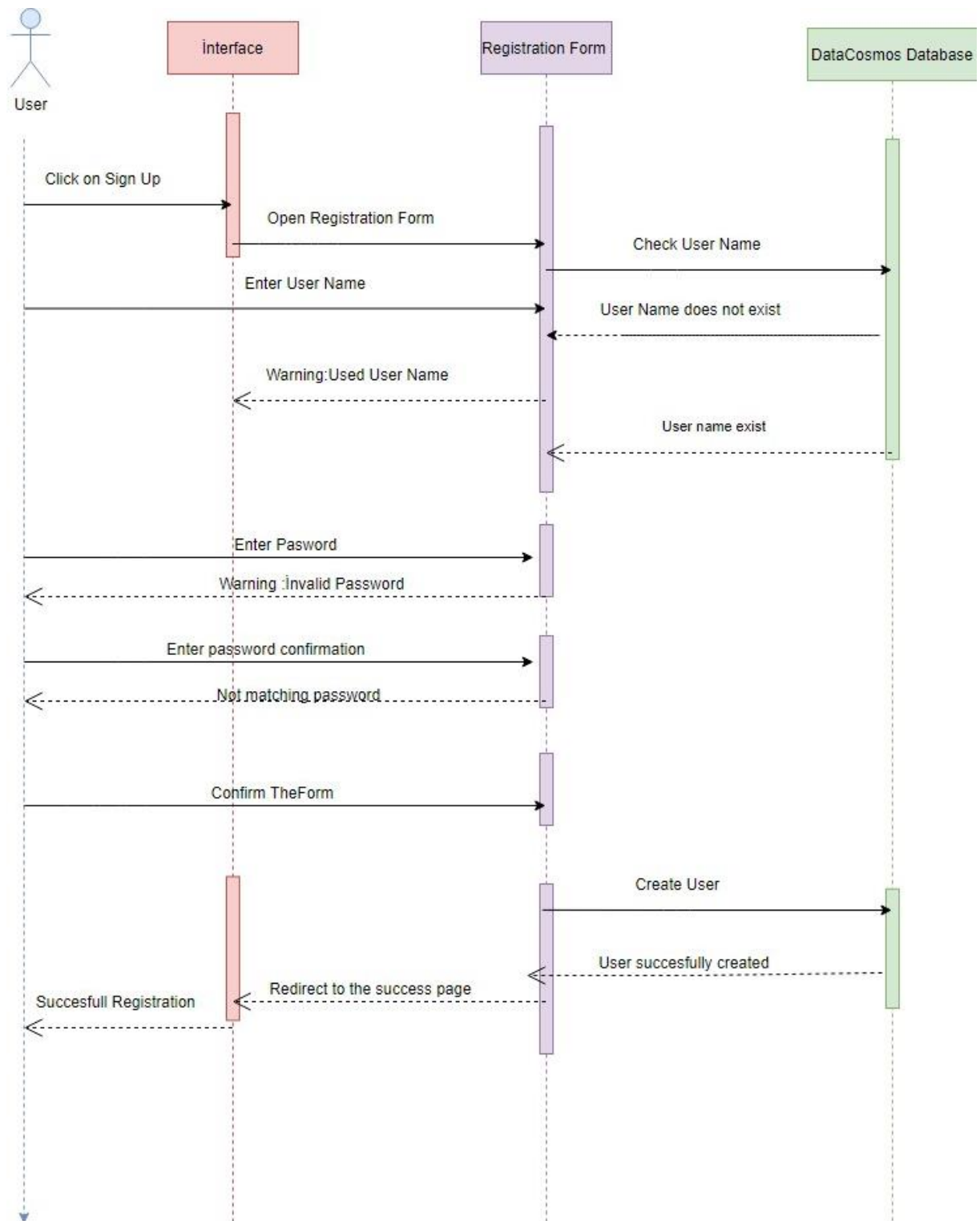
3.5.3 Object and class model

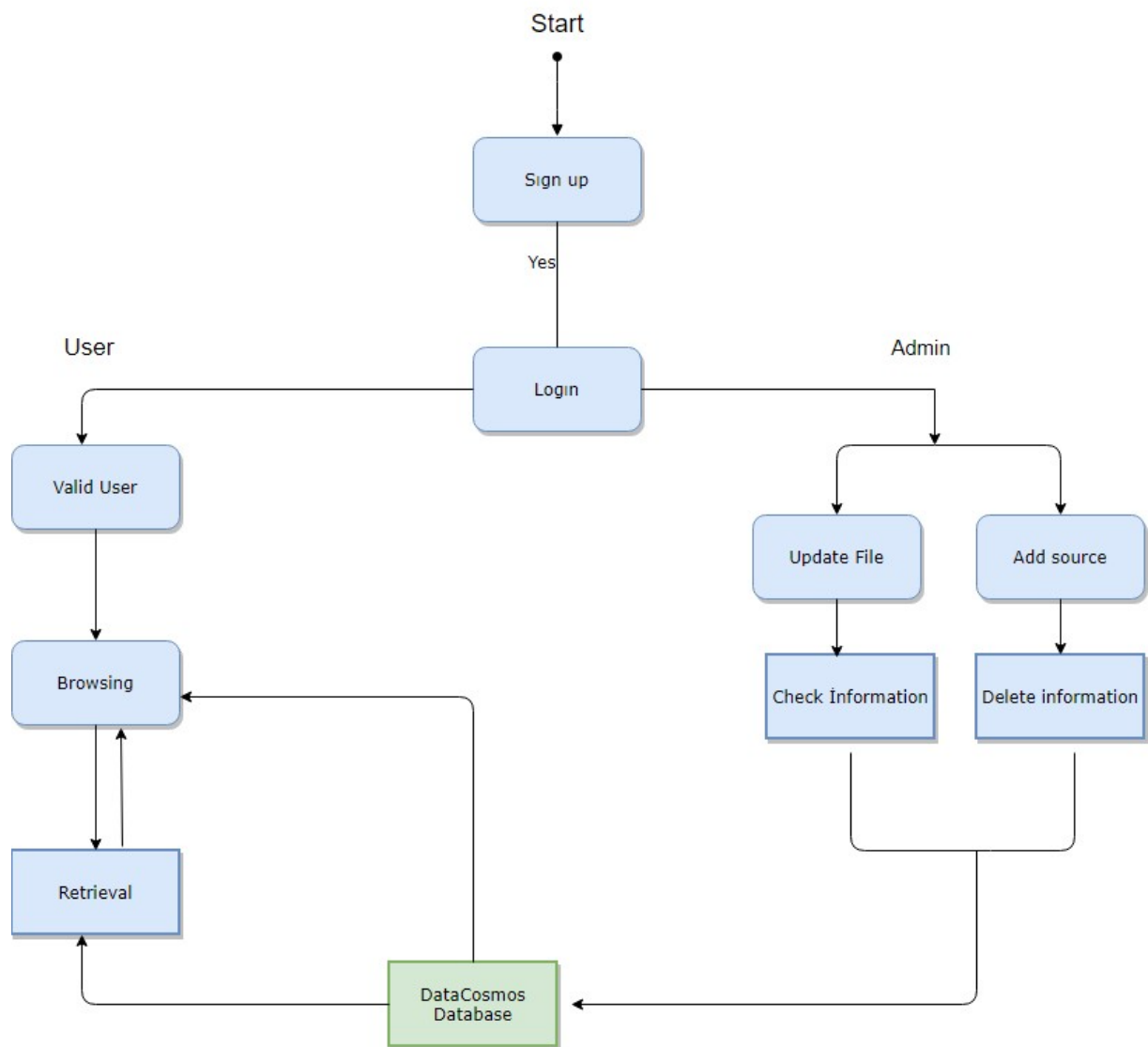


3.5.4 Dynamic models



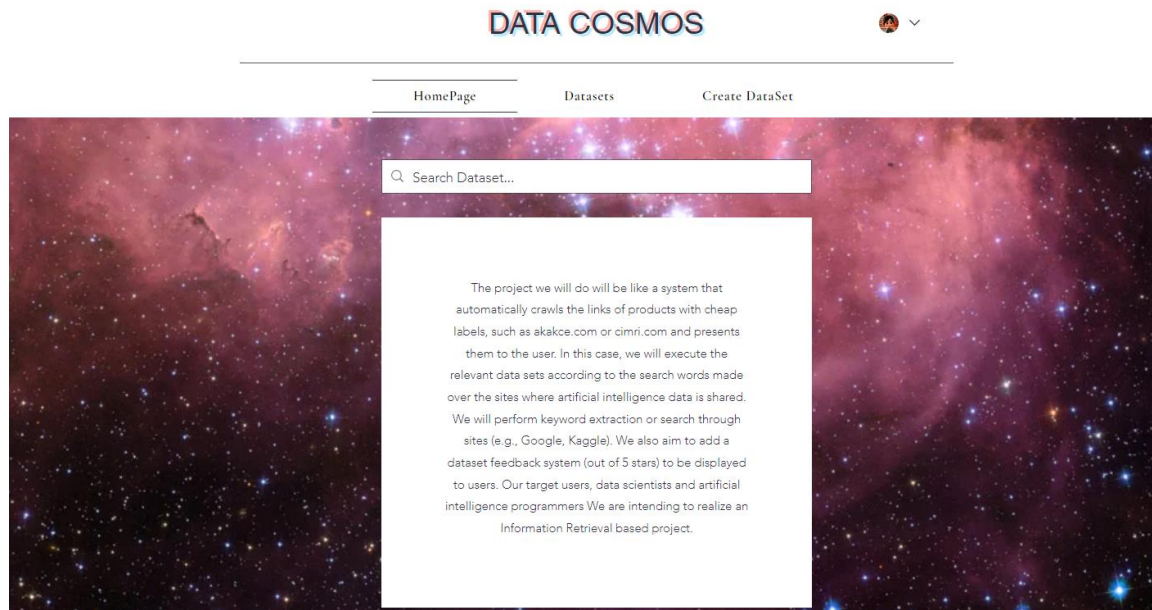




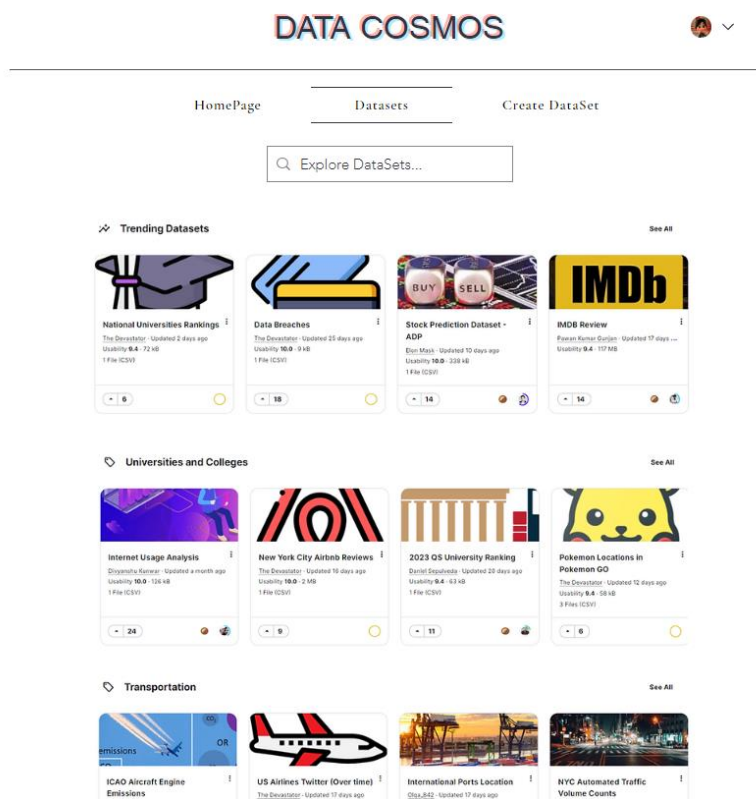


3.5.5 User interface - navigational paths and screen mock-ups

HomePage:





Datasets Page:



Create DataSet Page:

DATA COSMOS


 

HomePage

Datasets

Create DataSet

Enter DataSet Title



Drag and drop files to upload
Consider zipping large directories for faster uploads

or

Browse Files

Create !

Account Page, Favourite Datasets and Uploaded Datasets:





bensseker

Favourite Datasets Uploaded DataSets Account

Kişisel bilgilerinizi aşağıda görüntüleyin ve düzenleyin.

Hesap

Kişisel bilgilerinizi güncelleyin.

Giriş E-postası:
bensseker@gmail.com
Giriş e-postanız değiştirilemez

Ad

Bensu


Soyad


E-Posta *

bensseker@gmail.com

Sil





Bilgiyi Güncelle



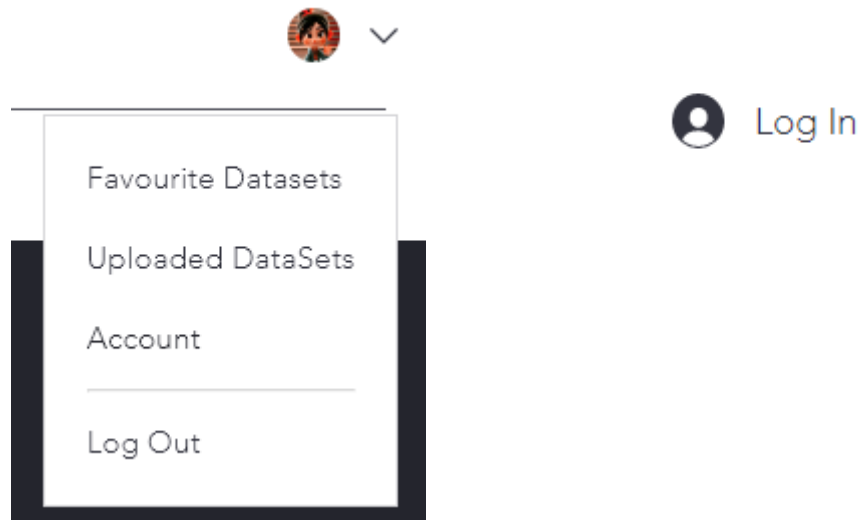


bensseker

Favourite Datasets Uploaded DataSets Account

	Cyber Incidents 2005 to 2020 MymahfL - Updated 2 years ago Usability 7.4 - 1 File (CSV) - 90 kB	25
	TREC (Question Classification) The Devastator - Updated 14 days ago Usability 9.4 - 2 Files (CSV) - 126 kB	1
	Student Study Hour V2 C-SPQ - Updated 2 years ago Usability 10.0 - 1 File (CSV) - 326 B	22 Bronze
	Mixed-type Wafer Defect Datasets ColtTera - Updated 2 years ago Usability 8.8 - 2 Files (other) - 23 MB	19

Log In and Log Out:



4. Glossary

User: They will gather the datasets related to their request in our website.

Resource: Resources are the websites that we are taking datasets and sending them to the users.

API: Application Programming Interface, helps 2 programs to communicate.

User Interface: A website which is easier to use for the user to gather their request from our project.

Query: A query is a request of data from the database.

Database: A systematic collection of data stored on a computer, particularly one that is easily accessible.

Architecture: General design of the project is called architecture. It defines the shape of the project.

Sign up: User gets into system by their unique information to keep their information private and makes it easier for us to keep track of security of our system.

Valid User: The user who successfully logged in our website.

5. References

<https://www.researchgate.net/publication/361696361> Online Testing of RESTful APIs Promises and Challenges

<https://www.researchgate.net/publication/328991978> Automatic Generation of Test Cases for REST APIs A Specification-Based Approach

<https://www.sisense.com/blog/rest-api-testing-strategy-what-exactly-should-you-test/>

<https://www.educba.com/dataset-type/>

<https://betterprogramming.pub/designing-sequence-diagrams-for-api-orchestration-113ac45e16df>

<https://www.geeksforgeeks.org/what-is-information-retrieval/>

[https://documentation.commvault.com/11.24/essential/98645 rest api get navigation.html](https://documentation.commvault.com/11.24/essential/98645-rest-api-get-navigation.html)