Software Architecture Document

Architecture design for an AI search engine

Team Number: 29

55170526 马韶泽

55170531 陈　宝

55170533 赵述军

55170535 肖俊义

55170536 陈　宇

55170537 孙一坤

Content

1. Introduction

1.1 Purpose

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the Search Engine system offered by Group 29. It presents a number of different architectural views to describe different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

The structure of this document is based on the “4+1” model view of architecture. The “4+1” View Model allows various stakeholders to find what they need in the software architecture.

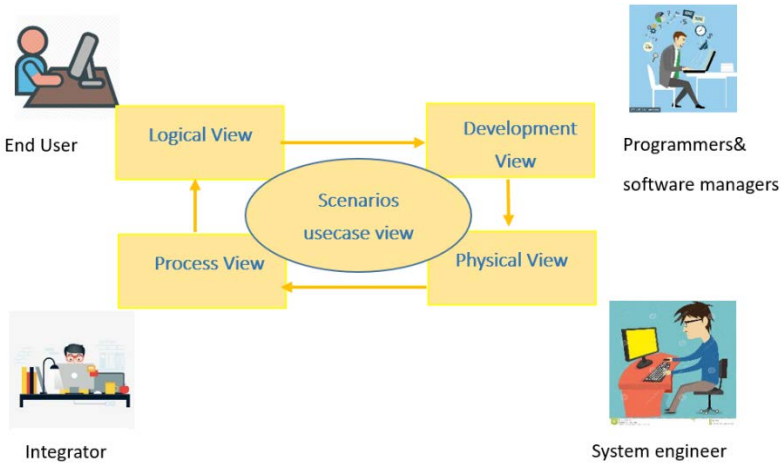

Fig. 1.1 “4+1” View Model

Scope

This Software Architecture Document provides an architectural overview of the AI search engine.It includes these parts:Architectural Goals and Constraints, Use Case View, Logical View, Process View, Deployment View,Physical View, UI Design .

1.2 Definitions, Acronyms and Abbreviations

1. DNN : deep neural network ,It is a neural network with a certain level of complexity, a neural network with more than two layers. Deep neural networks use sophisticated mathematical modeling to process data in complex ways.

2. Click Model:It is Based on the user's history click information, the user's interest and behavior to predict the user's future click behavior and improve the relevance.

3. Memory Database Technology: Place data in a database that operates directly in memory. Storing data in memory can greatly improve the performance of applications compared with accessing data from disk.

4. GAE :Google App Engine .

5. JSP: its full name is “Java Server Pages” advocated by Sun Microsystems. It’s a dynamic web page technology standard established by many companies.

6. HTML: Hypertext markup language, an application of the Standard Generalized Markup Language.

1.3 Artificial Intelligence(DNN) Design

We build a click model based on deep neural network. This framework can make searching result pages not only contain text,but also contain images, audio ,even videos.

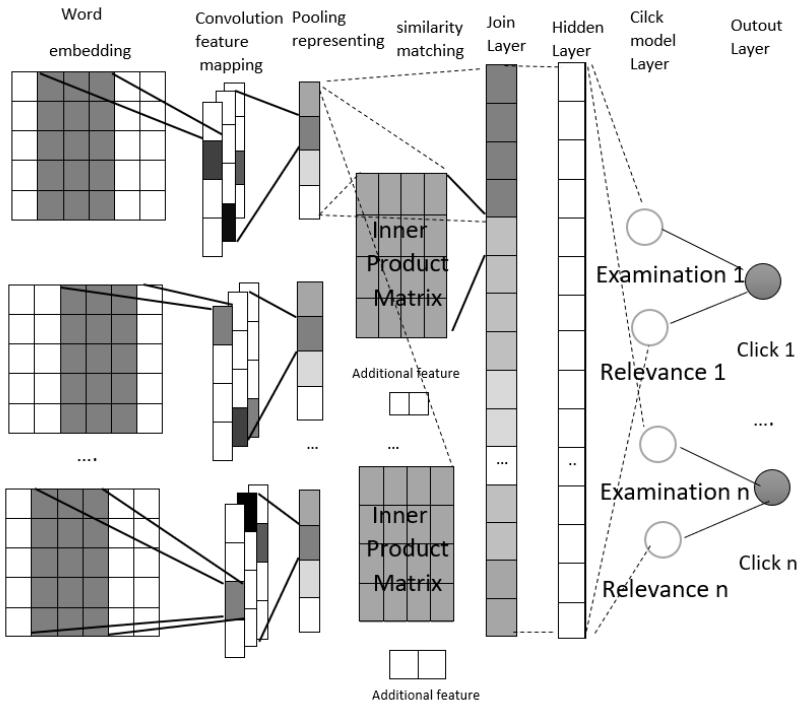

Fig. 1.2 Click model framework based on deep neural network

2. Architectural Goals and Constraints

2.1 Goals

2.1.1 Practicability

AI Search provides users friendly and intuitive interface.Users can put the search results that thwy want to save to favorites.And they can view、synchronize and manage favorites on different devices when logging in.

2.1.2 Reliability

AI Search may handle large volume of data in the world. So Services replicate across many different machines .It automatically detect and hande failures.In addition,it efficiently deals with spam pages and cloaking.

2.1.3 Security

AI Search try to protect user data storage and usage (including history,note,password)and to ensure that the user is in control of their data, rather than we being in charge of us data.Users can ask the us whether their personal data is being processed and used, as well as the purpose of use, the type of data collected and so on.

2.1.4 Compatibility and Extensibility

Users can use different browsers, such as Firefox、 chrome to search as well as can install plug-ins, such as language translation.

2.1.5 Specialty

AI Search should meet with the relevant standards and requirements.

2.1.6 Maintainability

AI Search engine will update periodically and comprehensively the site which has been included in the pages. When updating,it will replace the old page with the new page that was fetched and delete the nonexistent page.

2.2 Constrains

1. The system must ensure that unauthorized access to data is blocked. All remote access is controlled by user authentication.

2. The system will be implemented as a browser-server system. The browser section resides on the PC, and the server part must run on the UNIX server.

3. All performance and loading requirements, as stipulated in the Vision Document and the Supplementary Specification must be taken into consideration as the architecture is being developed.

3. Use-Case View

3.1 Overview

The use case view describes sequences of interactions between objects and between processes. They are used to identify architectural elements and to illustrate and validate the architecture design. In this view, the description of architecture is illustrated by use case diagram.

3.2 Use Case Diagram

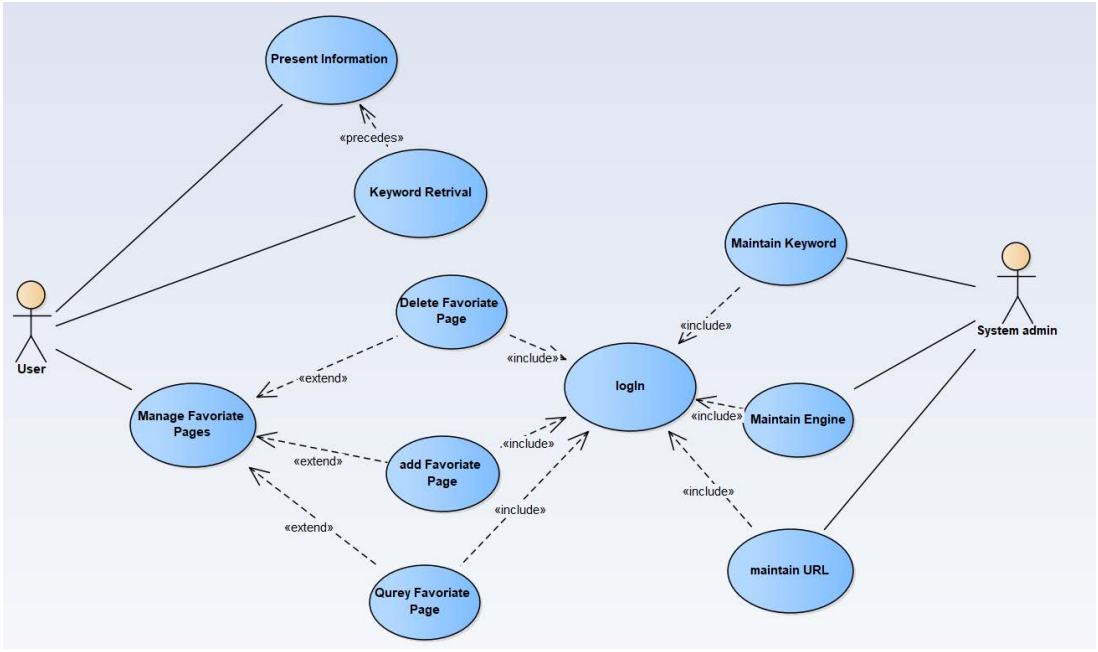

Fig. 3.1 Use Case Diagram

4. Logical View

4.1 Overview

The logical view is concerned with the functionality that the system provides to end-users. Here we use class diagram, sequence diagram to represent the logical view.

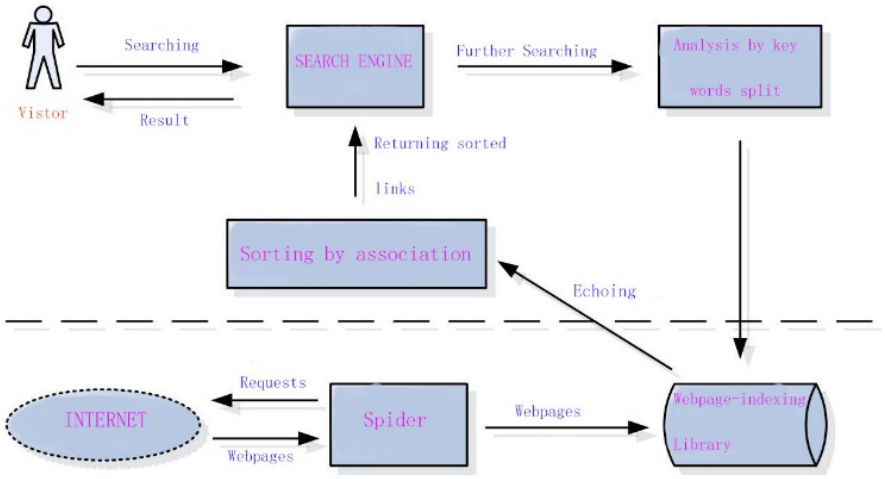

Fig. 4.1 sketch of Key Words Retrieval

4.2 Class Diagram

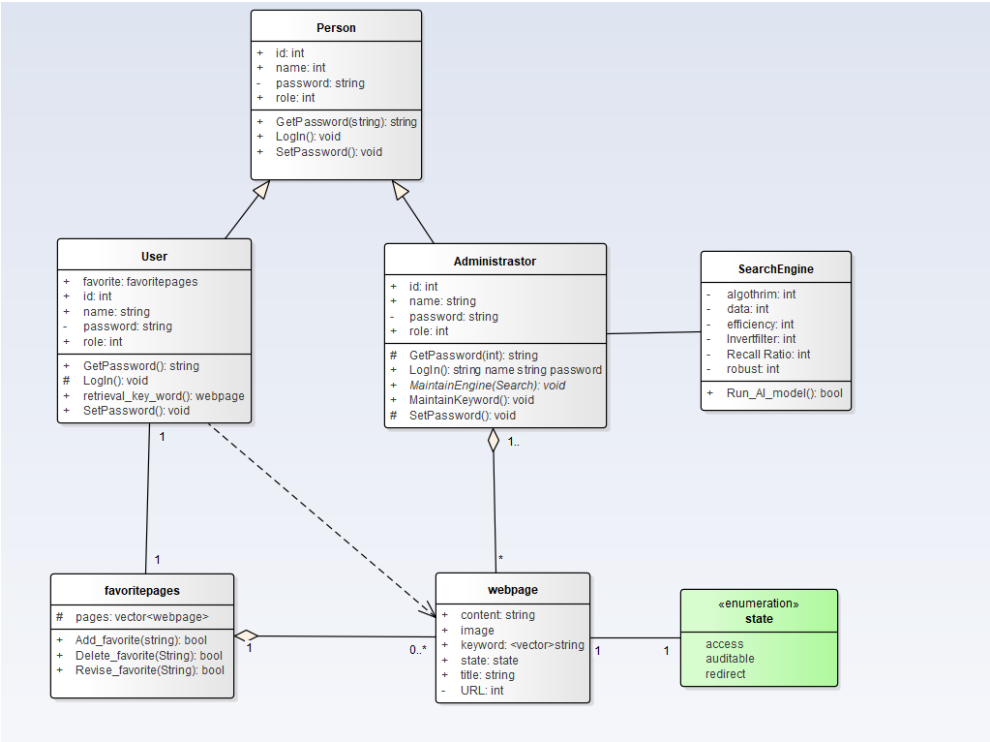

Fig. 4.2 Class Diagram

4.3 Sequence Diagram

Actors of the Search Engine

User: people who can search for information and register to have more permissions

Administrator: people who audit the web page and maintain the system, etc.

4.3.1 Use Case for User

Users have limited authorities. They can use the search engine by inputting key words for some specific information. But they can also become a web provider.

✧ Key Words Retrieval

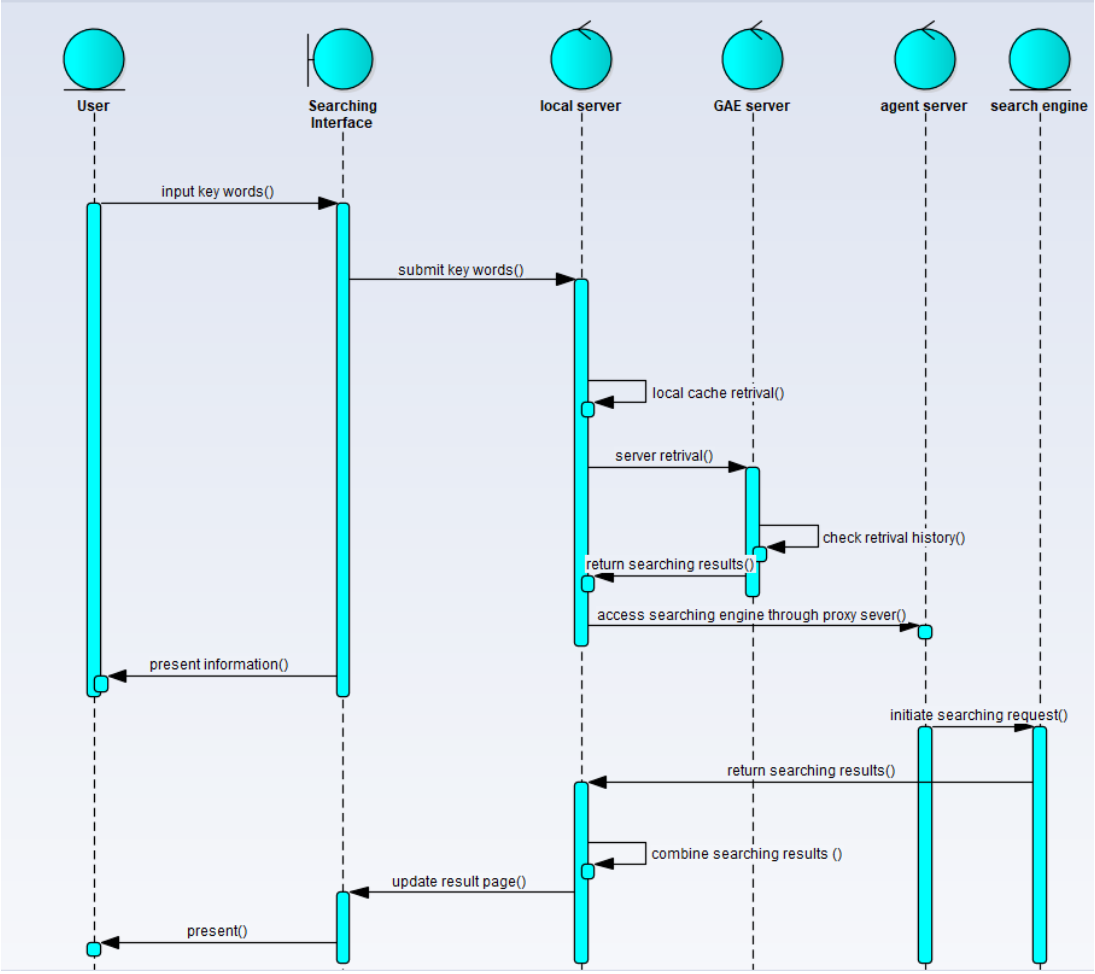

Fig. 4.3 Sequence Diagram of Key Words Retrieval

Use Case: key words retrieval

Participants: Users

a) Brief description: In this use case, the users can retrieve for information based on the search query and self-interests.

Main flow:

1. Users enter the search interface.
2. Users input the query.
3. The system returns a list of web pages to users.

Alternate flow: When the query sentences don't conform to specification, the system return feedback that what the user input is error.

✧ Add to Favorite Web Pages

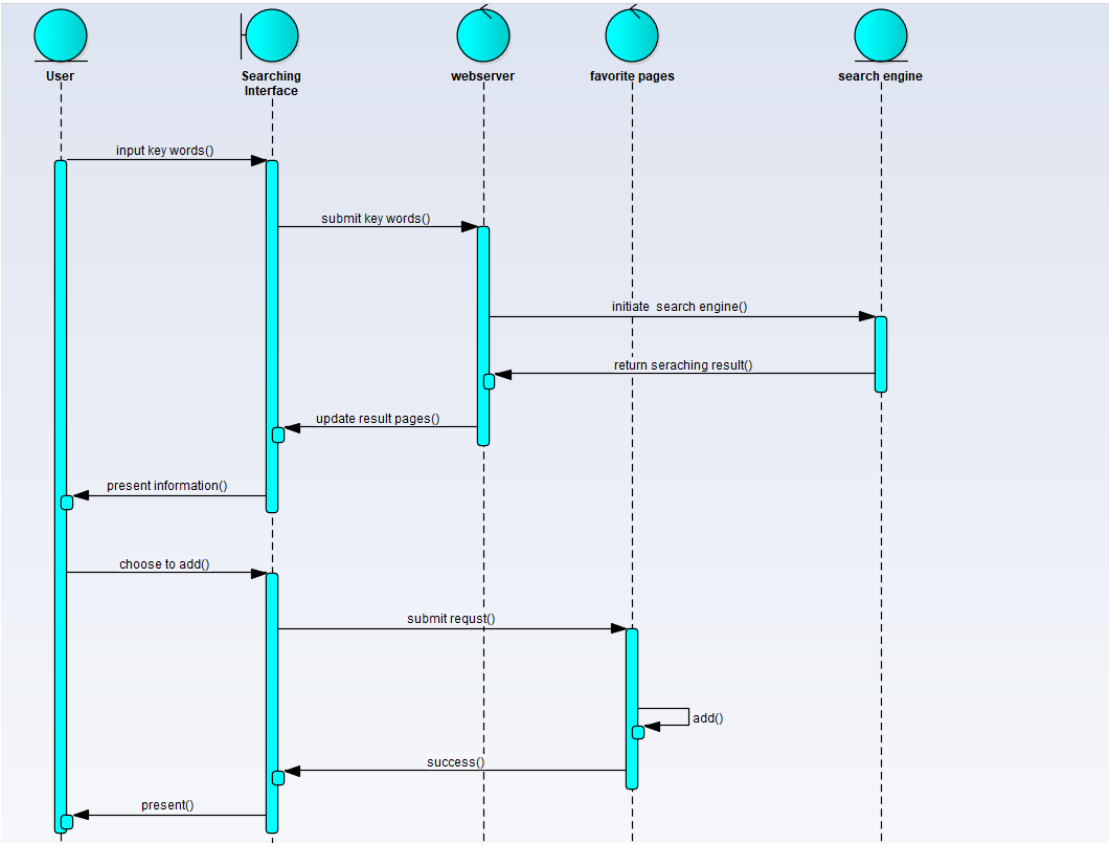

Fig. 4.4 Sequence Diagram of adding to favorite web pages

Use Case: add to favorite web pages

Participants: Users

a) Brief description: In this case, the users can put the web pages they like to the web collection.

Main flow:

1. Users login.
2. Users use key word retrieval.

3. Users enter the favorite web pages management interface.
4. If the web page is liked, Users add it to the web collection

Alternate flow: when the web page is not liked, Users may sign out from the web page.

✧ Delete from Favorite Web Pages

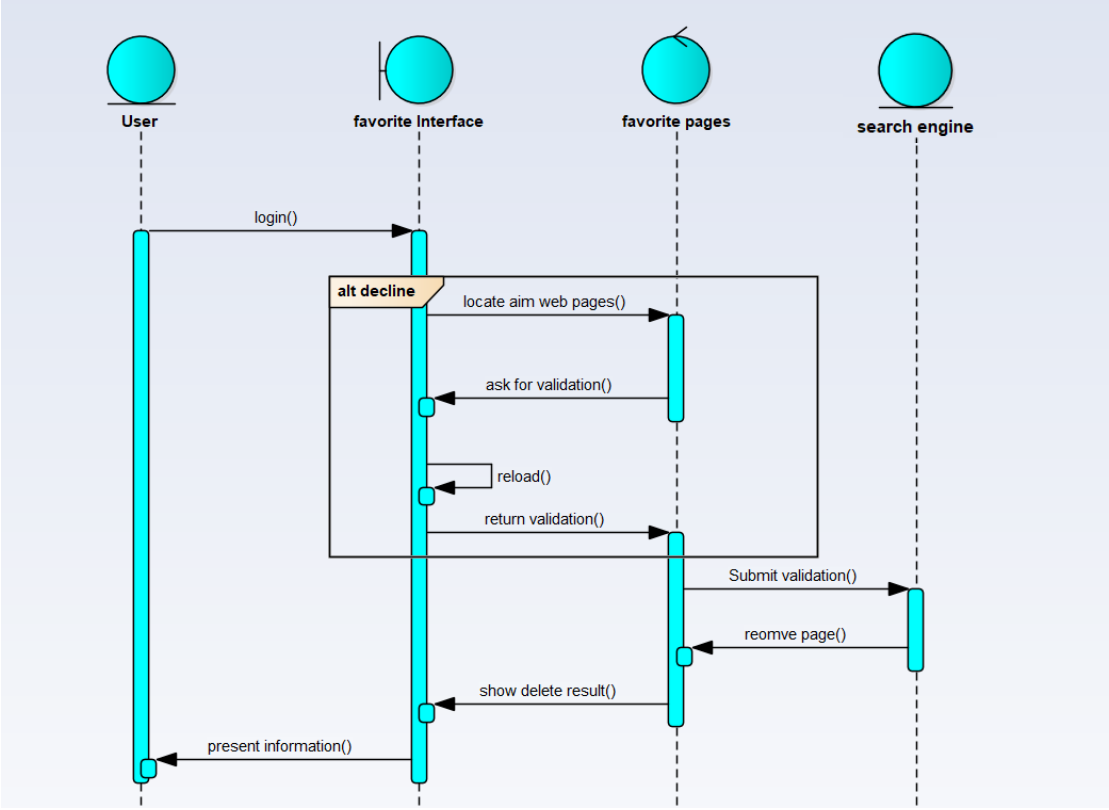

Fig. 4.5 Sequence Diagram of deleting from favorite web pages

Use Case: delete from favorite web pages

Participants: Users

a) Brief description: In this case, the users can delete the web pages they don’ t like any longer from the web collection.

Main flow:

1. Users login.
2. Users enter the favorite web pages management interface.
3. Users flip through the web pages in the collection and locate the aim pages.
4. The system asks the users for validation
5. If the user confirm, the aim pages will be remove from the collection.

Alternate flow: if the user cancels the validation, the users return to the management interface.

✧ Query Favorite Web Pages

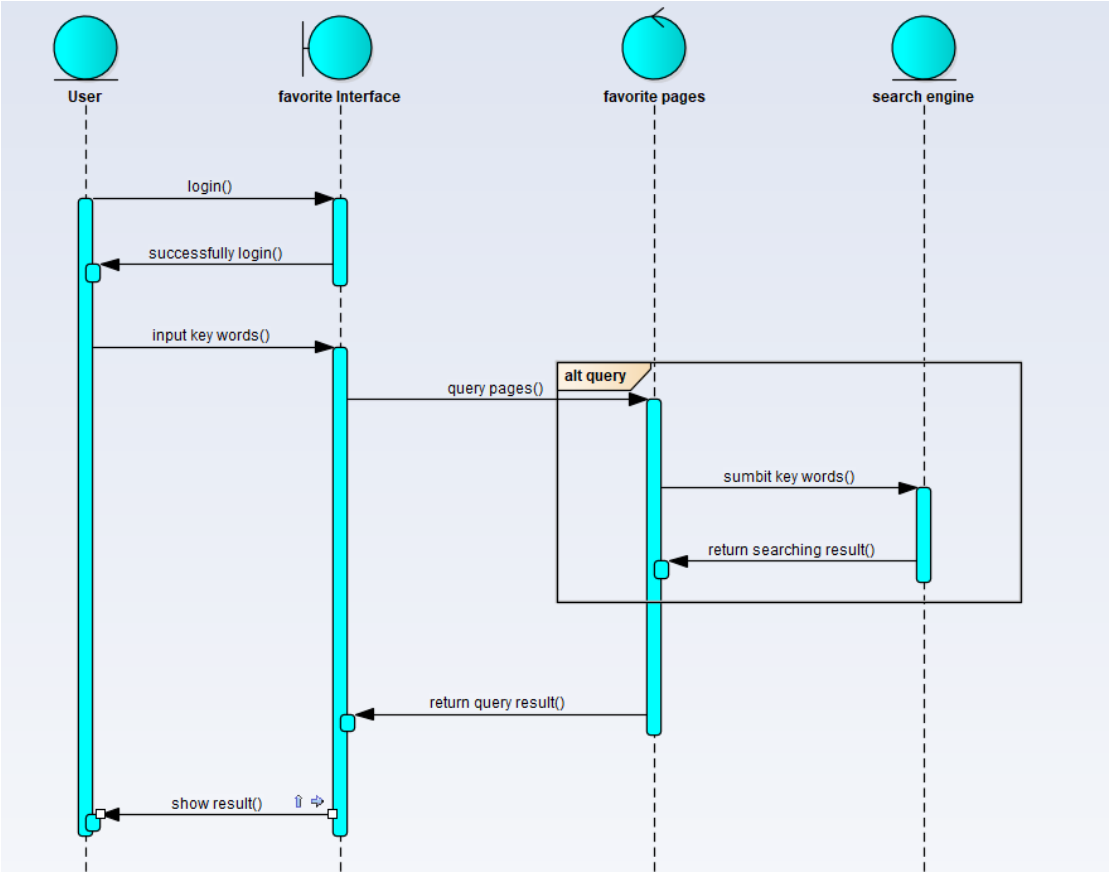

Fig. 4.6 Sequence Diagram of querying favorite web pages

6. Use Case: query favorite web pages
7. Participants: Users
8. Brief description: In this case, the users can query the web pages they have already put in the web pages collection.
9. Main flow:
10. Users login.
11. Users enter the favorite web pages collection management interface.
12. Users input the key words about the web pages.
13. If the web pages relevant exist in the favorite pages, the system will return the web pages.
14. Alternate flow: If the web pages relevant don' t exist in the favorite pages, the system will go to retrieval .

5.Process View

UML process views are often used to describe complex enterprise processes, use case scenarios,

or to model business-specific logic. Assignment of the runtime component instances to processes, threads and address spaces; how they communicate and co - ordinate; how physical resources are allocated to them.

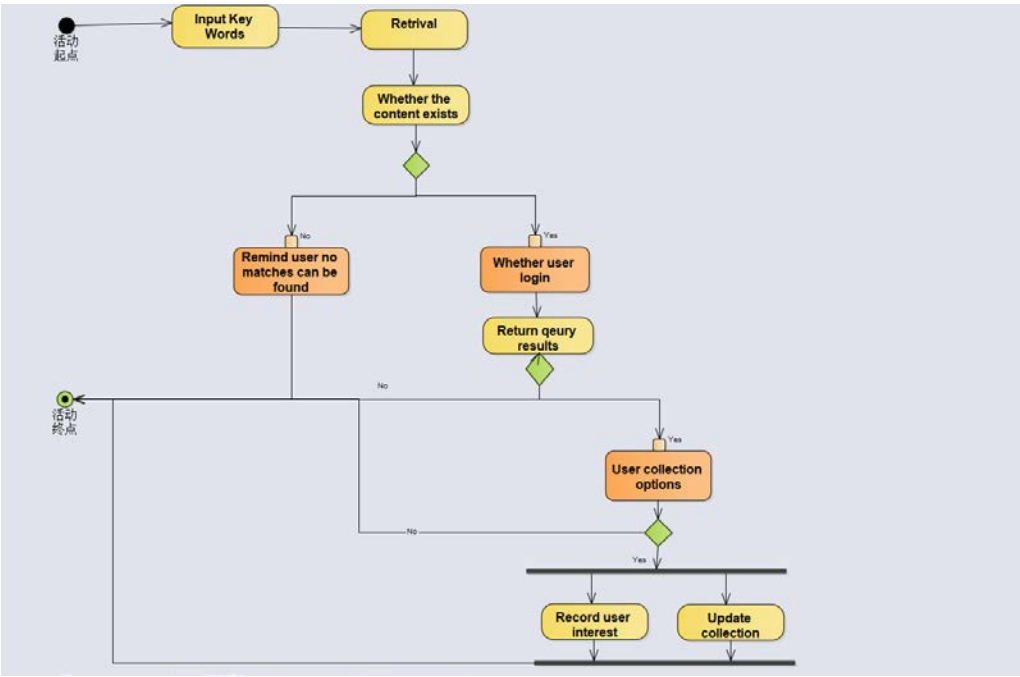

Fig. 5.1 State Diagram

6. Development View

6.1 Overview

The development view illustrates a system from a programmer's perspective and is concerned with software management. It uses the component diagram to describe system components.

6.2 component diagram

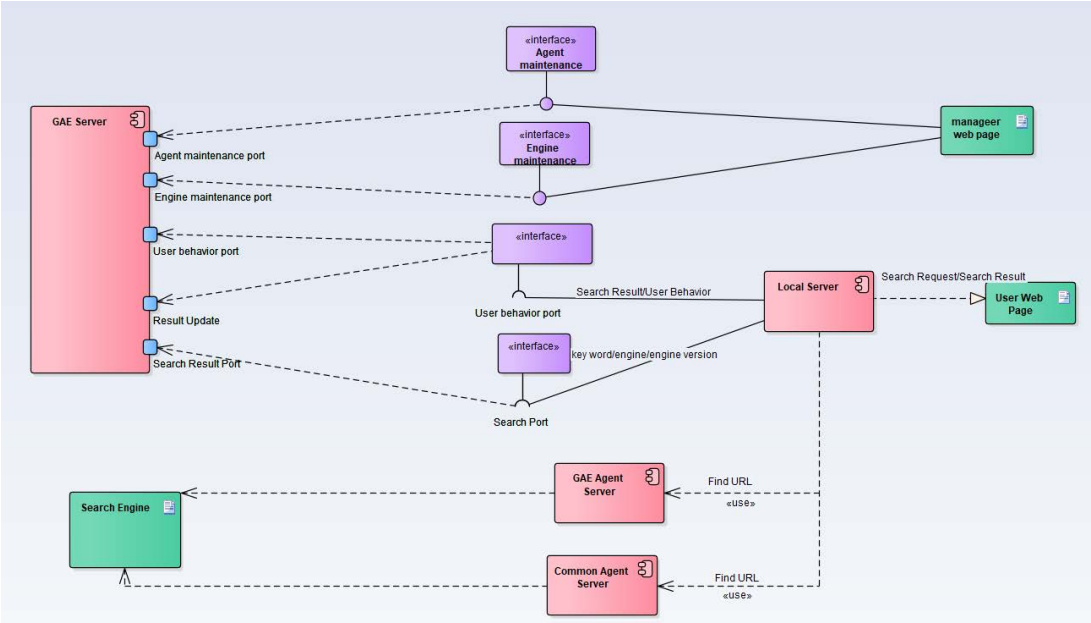

Fig. 6.1 component diagram

7. Physical View

7.1 Overview

The physical view is concerned with the topology of software components on the physical layer as well as the physical connections between these components. Deployment diagram is often used to represent this view.

7.2 Deployment diagram

A description of the deployment view of the architecture describes the various physical nodes for the most typical platform configurations.

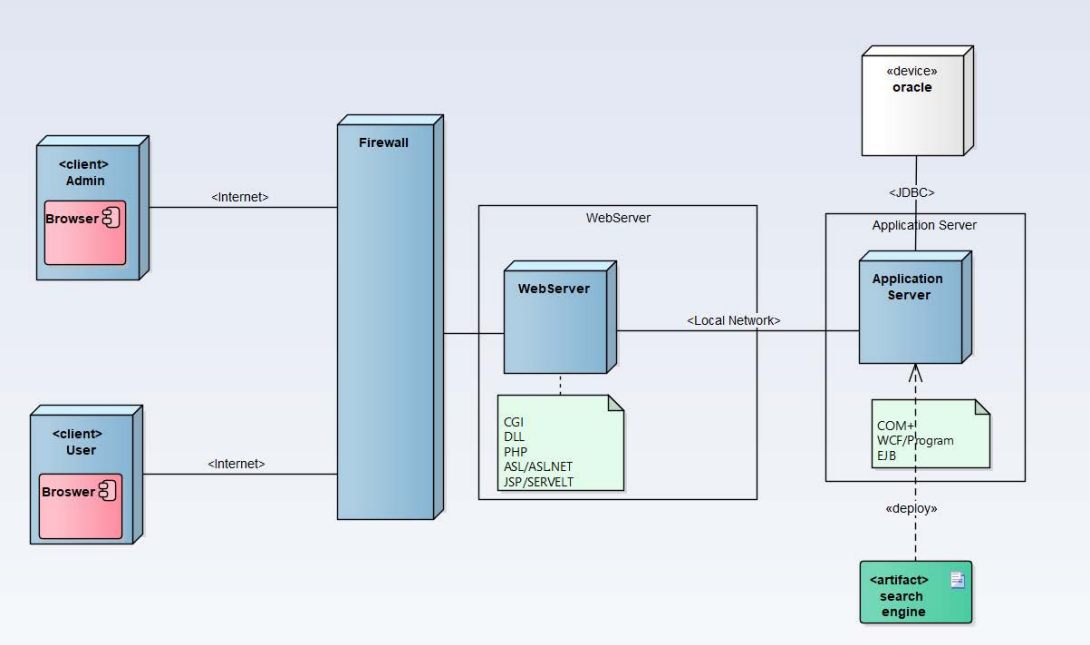

Fig. 7.1 Deployment diagram

8. Data View

8.1 data storage approach

we would like to use both local memory database and cloud Google App Engine(GAE) to store to store data information.In this way users don't have to worry about losing data, and they just need to log in to check their history when using another computer.

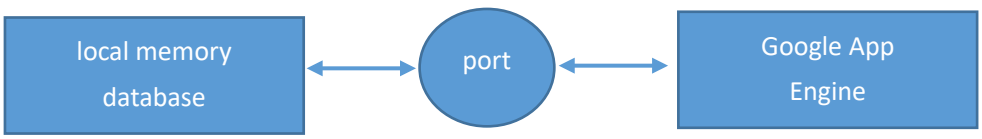

Fig. 8.1 data storage approach

8.2 Entity-Relationship Diagram

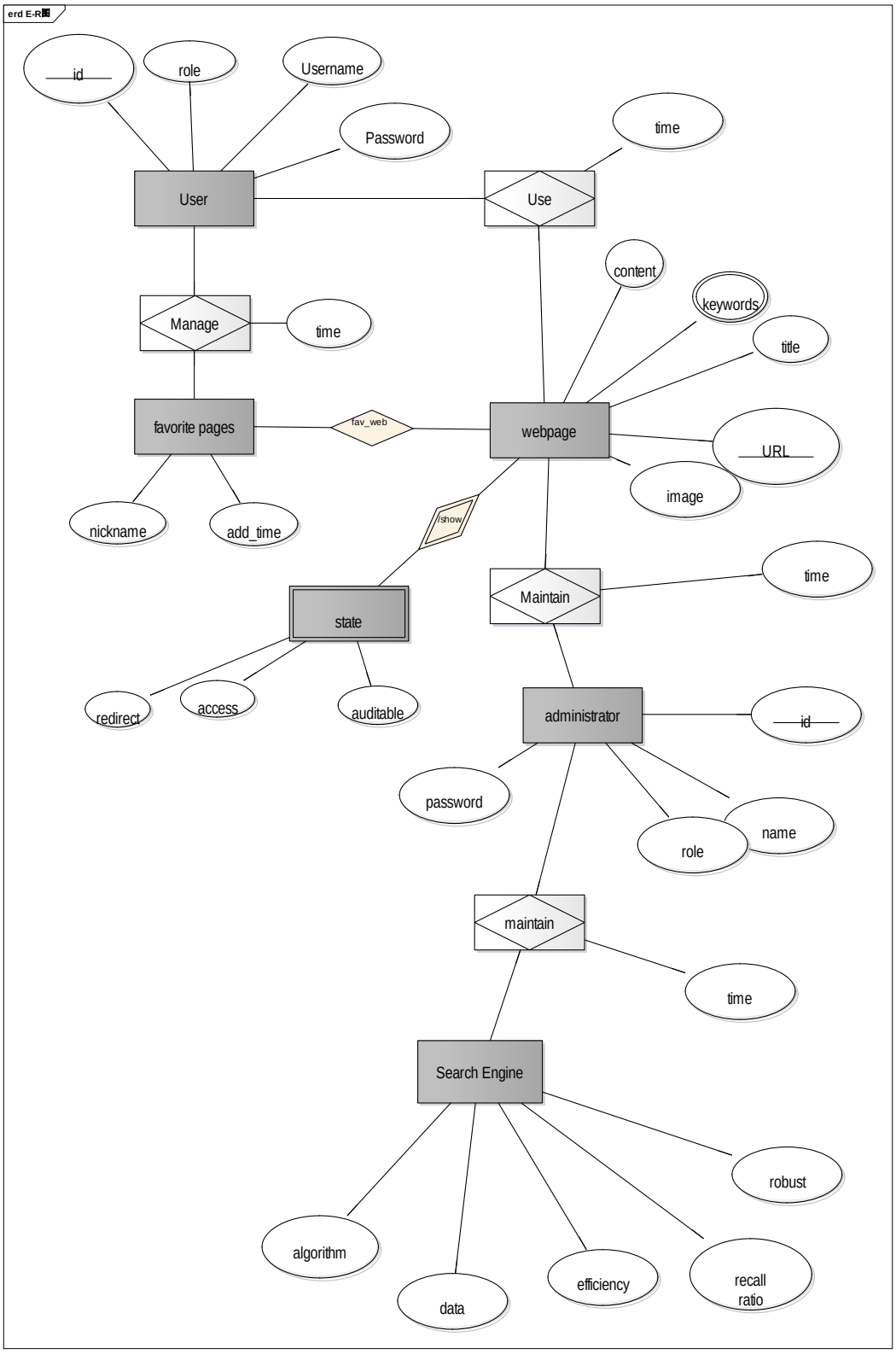

Fig.8.2 Entity-Relationship Diagram

9. UI Design

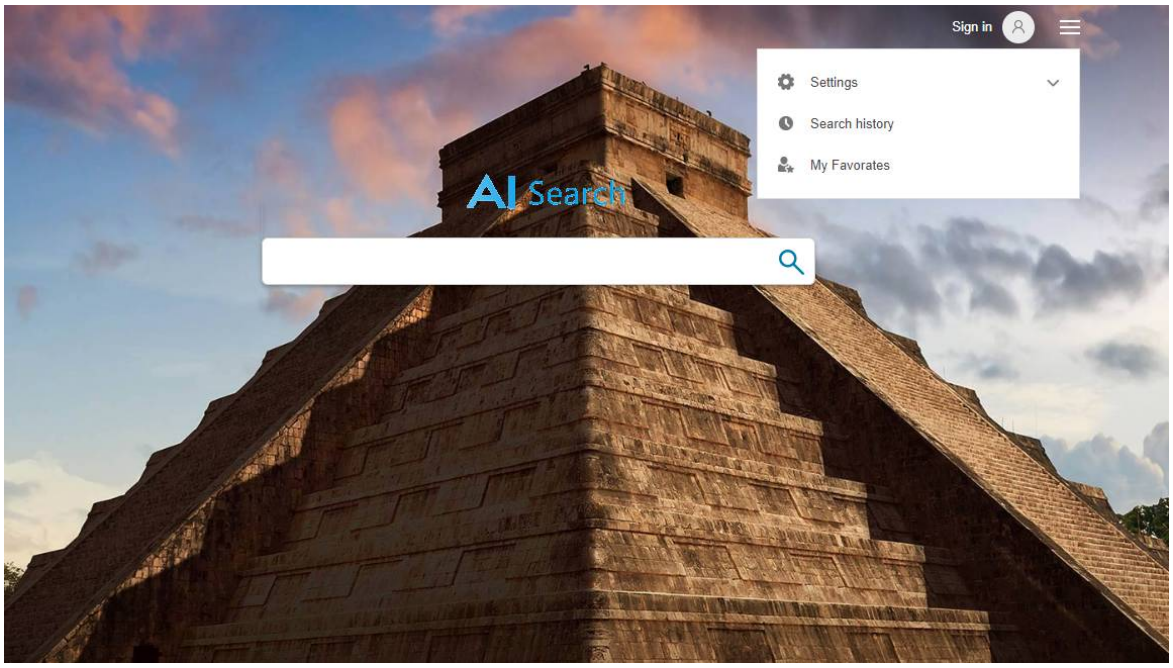

Fig.9.1 **Homepage**. User can query information by typing keywords.

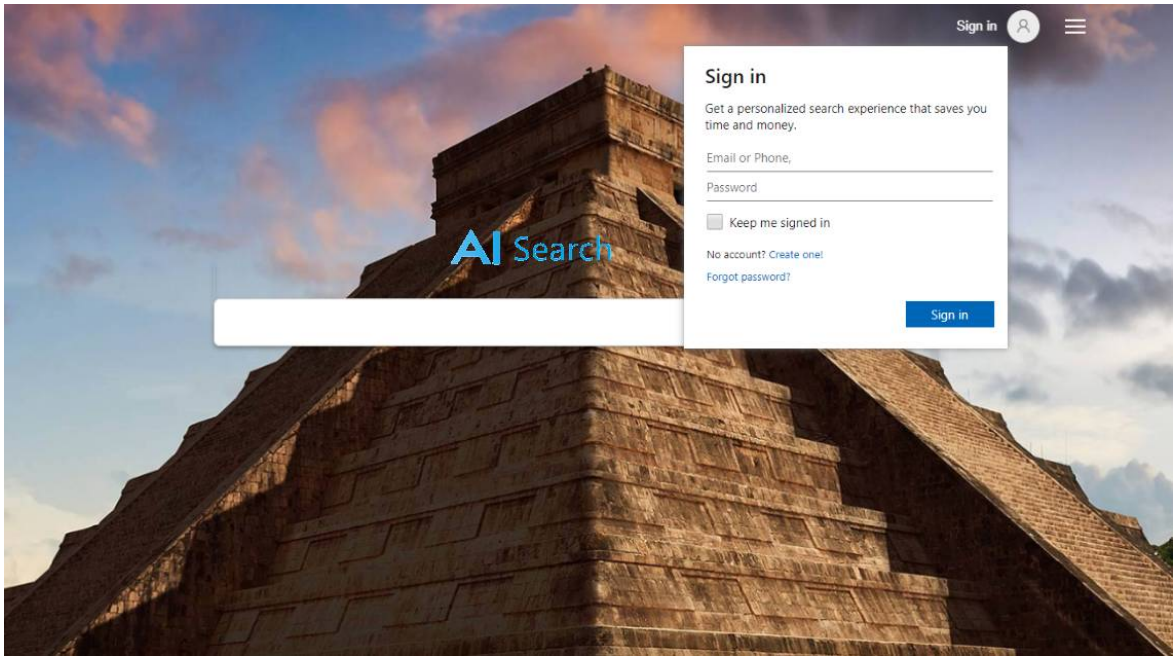

Fig.9.2 **Login Page**. User can enter their username and password to log in the system and they can manage their favorate pages.

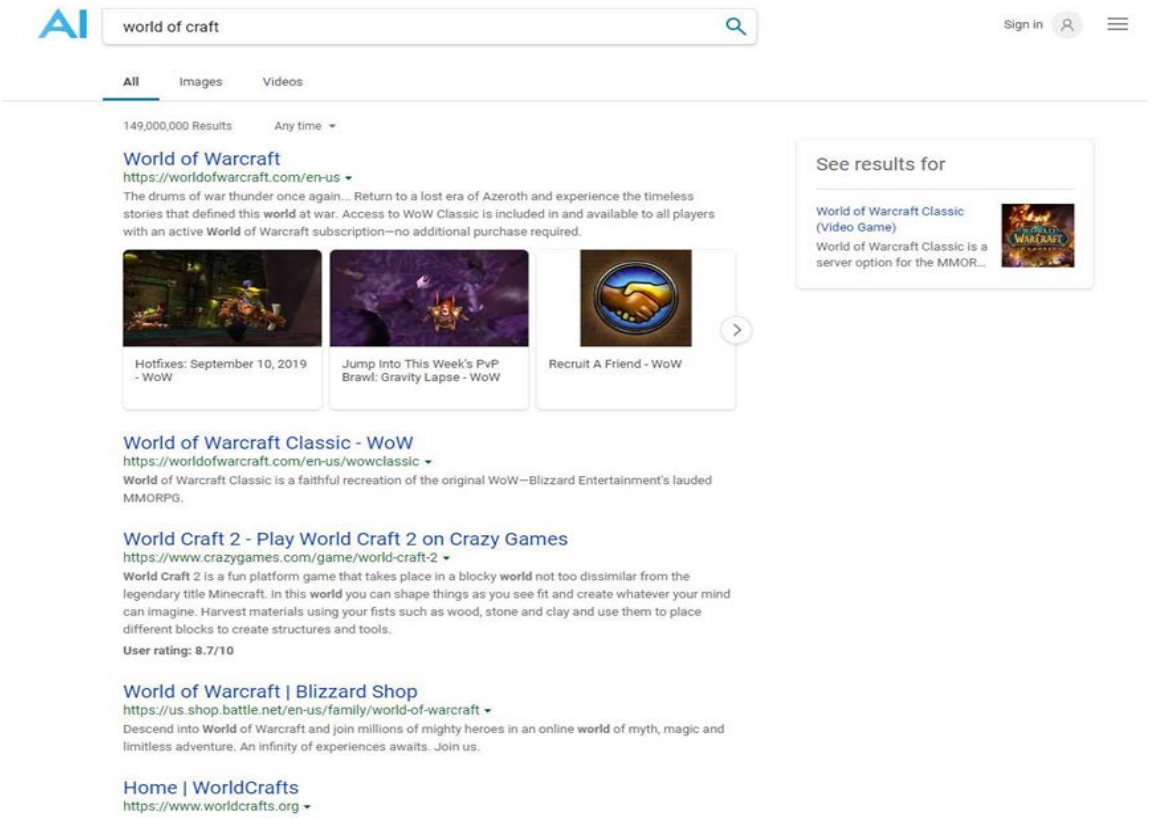

Fig.9.3 **Search Result**. The system returns matching information according to the keyword input by the user.

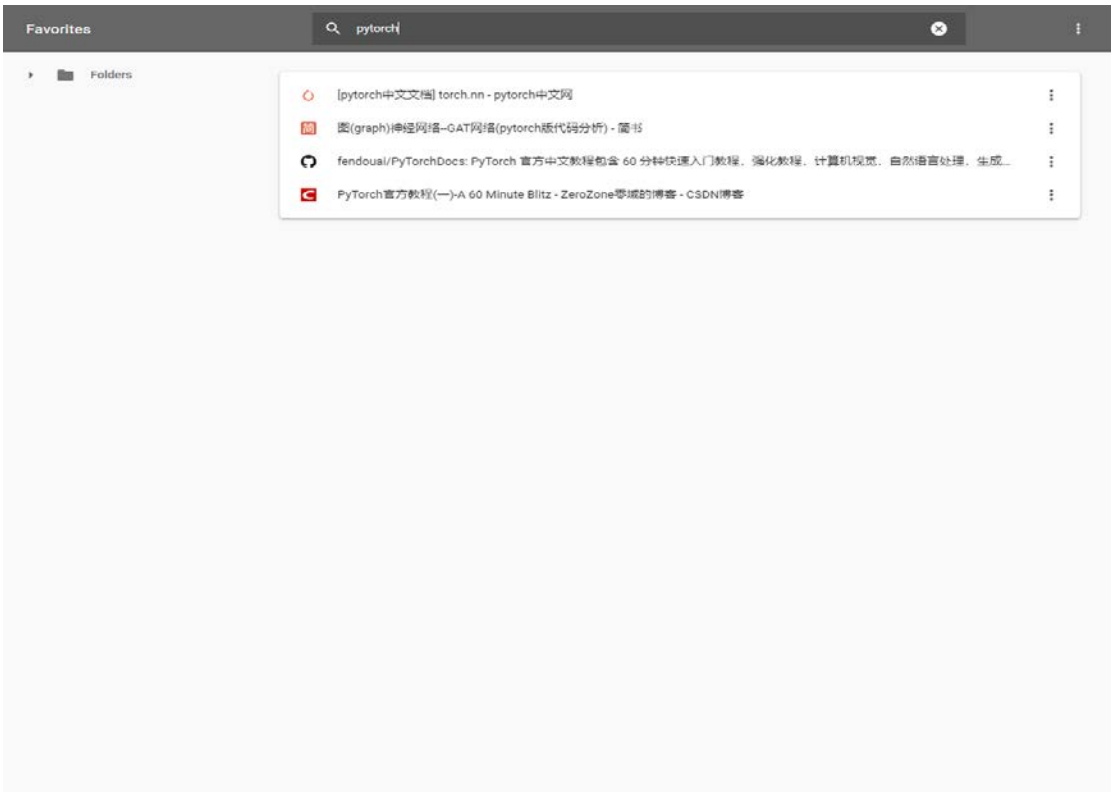

Fig.9.4**Favorate Page Query**. User finds matching pages from their favorite pages.

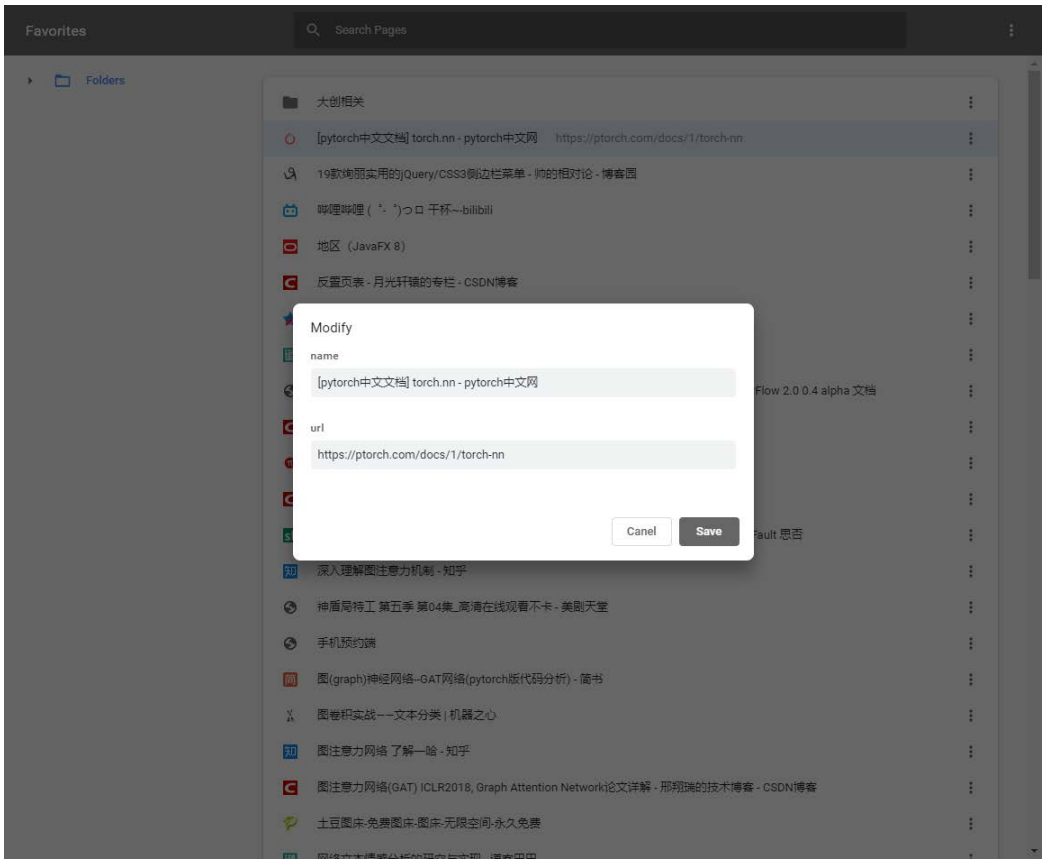

Fig.9.5 **Modify Demo** .User rename or change url for a page

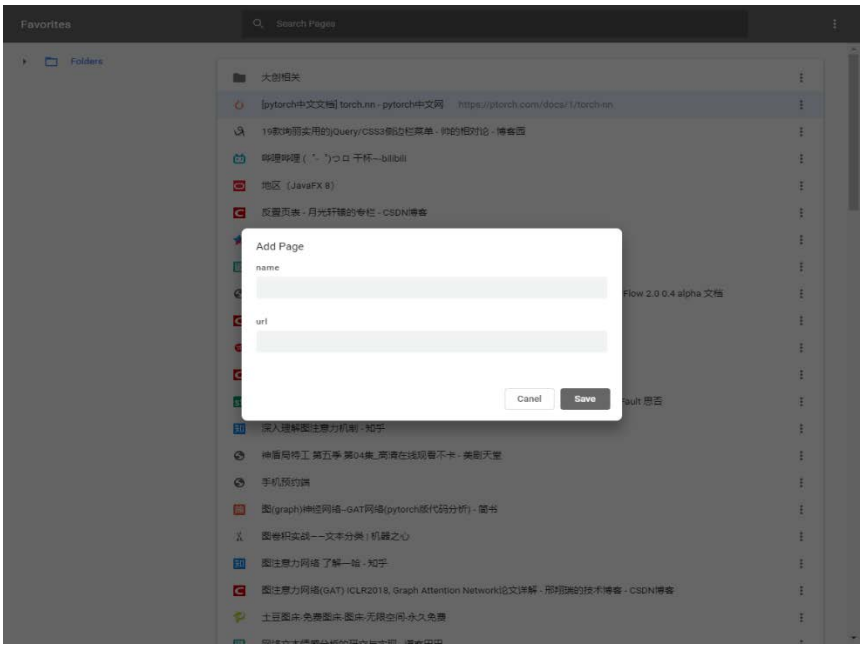

Fig.9.6 **Add Page Demo.** User enter name and url for a page to add it to favorites.