GOMP2043.GRP Interim Group Report

Project Title:
Webpages monitor robot

Supervisor:
Jiawei Li(Michael)

Author: Team 202219

Team Member:
Ruizhe HUANG(scyrh1)
Yuhao MA(scyym4)
Shujun JIANG(ssysj1)
Chi MA(scycm2)
Haowei GAO(scyhg2)

December 08, 2022

Contents

1 INTRODUCTIONS

Webpages Monitor Robot is a group working project of the module: Software Engineering Group Project. This project seeks to produce a software robot that searches designated webpages, detects changes on the webpages, and reports to the user periodically. To attain this objective, the team will work through a full cycle of software engineering process, where programming languages, SE approaches, the UI design, coding, debugging, and testing will be completed.

1.1 Motivation

Nowadays the majority of websites are being updated continuously with the advanced tools and web technologies. The interests of many users in the updating content on the websites has risen dramatically. However, the majority users are using traditional bookmarks in web browsers to keep track of the websites which wastes a lot of time and effort.

1.2 Aims and Objectives

The project aims to produce a software robot that detects specific changes on designated web pages and reports the retrieved info to the user automatically on a regular basis. Having spoken with the stakeholder, for example:

1. If the user is a researcher and would like to monitor published papers on the website of Google scholar, he/she may input a few keywords to the robot and the robot will search the keywords for new publications on Google scholar without the need of manual intervention.

2. If the user is searching for videos on the website of Bilibili, he/she may access to the latest videos including some related information by adding the keywords to the robot.

3. If the user is interested in the latest news, he/she may enter some keywords and the robot will fetch the news headlines, links, media, upload times, and news content.

4. If the user needs to monitor specific job information, the software can regularly crawls for appropriate jobs based on the user's resume.

1.3 Proposed solution

Our proposed solution aims to integrate four functions, Google Scholar, Sina News, Bilibili, Shixiseng job portal into a single platform. After a group discussion, it is decided to run the software on the Mini program. And it is able to be used on both Android and IOS operating system.

In this report, several parts will be explained including: the background information part, a clarification of the system requirements, UML diagrams and the specification, a prototype of the software, key implementation ideas, and the problems our team has faced or conquered.

2 BACKGROUND INFORMATION & RESEARCH

2.1 Background

In the current context, most users use bookmarks in their browsers to keep track of websites and get the latest updates [1] if they are interested in certain web pages and need to understand when something has changed, and they have to manually refresh the page periodically to see if anything has changed. It would be a time-consuming and tedious process, especially in today's rapidly-paced life [2]. Change Detection and Notification (CDN) systems make it easy for users to get notified about changes that have occurred on webpages, without having to refresh the web page continuously. Montastic [5], Follow That Page [6] are some of the most popular CDN services which are used by numerous users to get updates about content changes that occur on web pages [1]. A web page is a separate file composed of text, graphics, and additional features such as links to scripts and style sheets.[3] While not all search engines work in exactly the same way, they perform at least the following basic operations [4]: 1. Search the Internet for content that matches your keywords. 2. Record the location and index of the matched content. 3. Provide a search service (search server) to help find words or combinations of words in indexed databases. Web crawlers are used to traverse the Internet, visit web pages, download their content for indexing, and improve upon them according to different needs. Figure 1 shows an overview of the web crawling process.

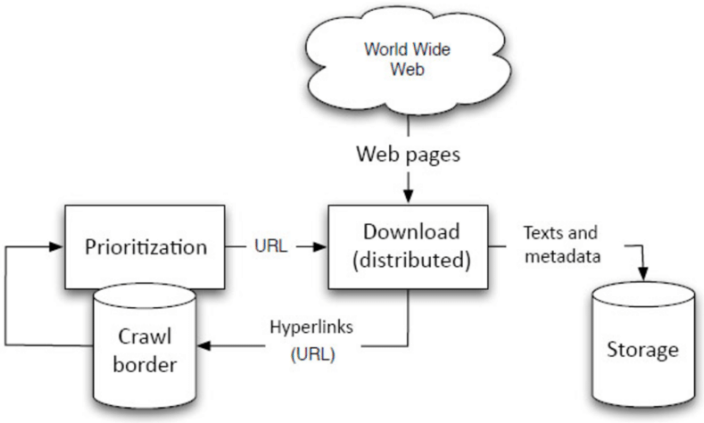

Figure 1: Overview of web crawling process

For each page, it extracts elements considered significant and relevant to form a database of keywords relevant to the page being analyzed [7]. As for our software, it is designed to use a timed crawler, as well as a database, to retrieve several popular websites including Google Sholar, Bilibili, Sina, and ShiXiSeng, and store the information

into the database.

2.2 Competitor Analysis

In today's world, where the Internet is an essential component, there are countless professions that require the ability to monitor webpage updates, and based on this user demand, numerous computer languages are capable of crawling data about website updates. Most of them are able to monitor changes in websites and update queries as frequently as the user needs them. The following is the research on additional related robots.

2.2.1 Visualping

The first is a quite popular website monitoring tool on the market, visualping, which is very simple and greatly versatile, Visualping claims it is utilized by more than 1.5 million clients around the world performing 5 billion checks, it is evident that he has a great deal of use, It has a comprehensive range of features and the frequency of queries can vary from five minutes to a week, with each recent update being sent to you by email or by your preferred method. It also supports multiple modalities, such as visual text or elements. The UI is considerably neat and simple for a portable webpage. Technically, it has multiple proxy options. It can also perform investigations on protected pages. However, as a commercial software, it is slightly more strict about the price, its fees are higher than others, and it has a limit on the number of accounts under one paying company.

2.2.2 Distill web monitor

The next competitor to be introduced is called Distill Web Monitor. In addition to monitoring the entire webpage, the plugin also provides local content monitoring, such as an article, a paragraph of text or even a word, which will be updated whenever it changes, In addition to this, Distill Web Monitor also supports monitoring the feed content of a website (which requires the current page to have a link to a supported feed subscription), thus eliminating the need to use RSS feed tools. The minimum interval between automated shift checks is five seconds, and real-time change alerts can be used on some pages.

2.2.3 Conclusion

There are actually numerous additional testing tools available for this purpose, for example Uptime Robot and Uptrends, etc. Their functions are more or less the same, the difference may be in the price difference of the charges and the handling of some special cases, and the clientele is slightly different. They are all based on the user's request for a website, and all pay a certain amount of money, The main difference between us and them is that our main target audience is UNNC students and teachers, we have specified the site but not the keywords to be searched, users can search for the keywords as they see fit, we will provide feedback on the updated search results based on the keywords

searched for. At the same time, we develop a job module. Users can fill in the form with your personal information. Robot will recommend the latest internship positions for users based on your personal situation and your needs.

3 REQUIREMENT SPECIFICATION

For the user specification, since our group did not conduct ethical discussion and reporting, we conducted several informal surveys and discussions with our friends in accordance with the rules, and tentatively collected responses from everyone. We will summarize the information from the Internet and our survey.

3.1 Google scholar

First, in terms of the selection of the sites to be crawled. The first is Google Scholar. The site is a must-have piece of software for universities. Nearly every university student facing a dissertation needs to find relevant citations from it. As of January 2018, Google became the world's largest academic search engine. The researchers concerned estimate that the website contains a total of about 389 million documents [8]. The vast majority of respondents to our survey have used Google Scholar for citation purposes. Thus, for a site of this size and number of users, it is essential to create relevant search term updates to help students and professors keep up with the latest academic trends.

3.2 Bilibili

The second is Bilibili, which boldly reached approximately 294 million monthly active users in the first quarter of 2022, according to the research. With 90 percent of users under the age of 25, mainly in the post-90s and post-00s age groups, and a high proportion of university students [9]. UNNCers are no exception. More than 90 percent of our respondents felt that they used B-Sites every day for entertainment or learning, and several students regularly posted videos on Bilibili. For the creator, access to the latest videos in his field is necessary to help improve the quality of his videos and keep up with current events, and for the user, access to the latest videos it needs to keep the user up-to-date with relevant information and improve the user experience.

3.3 Sina News

The third is Sina News, with 146.1 million monthly active users as of August 2021, Sina News has reached the top three in the industry [10]. Sina News updates tens of thousands of news items every day, covering all aspects of news such as politics, military, humanities and sports, and for each category, the updates are just as rapid, For users, if they want to keep abreast of the latest news in a direction, they cannot do without the reminders of the relevant update crawlers, .In response to this question, over 50 percent of respondents expressed a high level of interest in news from all directions,

and more than 70 percent think it is necessary to monitor news websites for real-time keyword updates. Therefore, we decided to crawl the Sina News.

3.4 Shixiseng(To be determined)

The final site is Shixiseng, a Chinese recruitment platform that focuses on internships and school recruitment. Since its inception in 2014, it has been used by more than 17 million college students at home and abroad, according to the website. Currently, Shixiseng has more than 300,000 corporate users and a total of more than 3,000 universities at home and abroad. With 13 million+ APP downloads, it has become the TOP 1 APP download in China's campus workplace category [11]. Internship opportunities are critical for UNNC students and have a direct impact on their future applications for graduate school and employment. According to our informal survey, about 60% of the students knew about the Shixiseng platform, and nearly all of them who had already participated in an internship had experienced posting their CVs on the Shixiseng platform. The main problem is the low response rate after posting a CV. It is thought that the low response rate is due to the fact that some of the advertised jobs have been posted for a long time and there may no longer be a need for trainees for the post. Based on the above two points, we decided to choose a highly representative software like Shixiseng to optimize student usage. Our program will intelligently recommend relevant jobs based on the information filled in about the user, and will also push alerts for jobs that merely go live and fit the user's CV, so that we can achieve more suitable jobs, but also allow the user to be the first to get the latest job push and send in their CV faster.

4 SOFTWARE DESIGN

According to the requirement specification, our team designed the initial version of the web monitoring robot software.

4.1 User Case Introduction

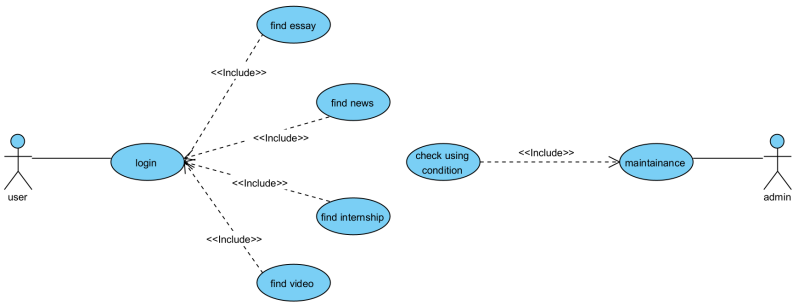

Figure 2: User Case Diagram

The software includes four functions: "Google Scholar" monitor to get the latest paper index keywords; "Sina" monitor to get the latest news index keywords; "Bilibili"

monitor for index keywords of the latest or most popular videos; "Shixiseng" monitor for job postings that users are interested in. Users could select a function through the four modules at the bottom of the software page. The following sections illustrate these functions with activity diagrams.

4.2 Operating principle of Google scholar

The following figure shows the process of a user using "Google Scholar" to obtain a paper. When user first enters the "Google Scholar" page, a new monitor needs to be created to monitor the paper. To activate this function, user needs to set the keyword of the paper, the maximum index range(e.g. papers updated 3 days ago), the number of articles obtained each time, and the time interval of each index. The software will save the relevant information (topic, time, etc.) of the index into the database for the convenience of the administrator to maintain the software. At the same time, it will send a request to "Google Scholar" to obtain the connected web pages. If the request fails, it will show that there are no new papers at this stage. If the request returns normally, the latest paper is displayed on the user interface and an email is sent to notify the user. The user can click on the article to select the PDF file to download. Users can click "Stop" to end the regular monitoring, and then set new information to restart the monitoring, or click "Out" to exit this function. The results of the previous monitoring will be displayed when the subsequent user reenters the page.

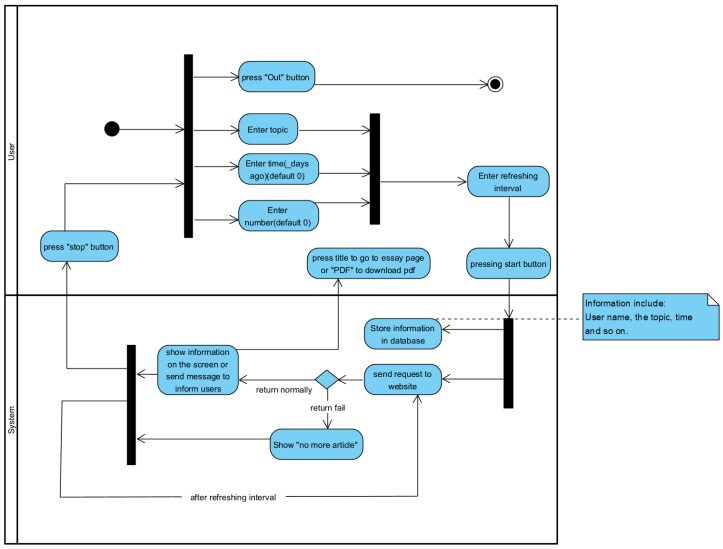

Figure 3: Google Scholar Activity Diagram

4.3 Operating principle of Sina News

The following activity diagram shows the process of a user using "Sina" to get news. As with the other features, when a user first enters the "Sina" page, a new monitor needs to be created to detect the news he or she want to follow. Unlike "Google Scholars", the user only need to set the keyword of news, the number of articles to be retrieved at each time, and the interval between regular searches. The software will also save

index information (subject, time, etc.) in a database for the administrator to maintain the software. At the same time, it will send a request to "Sina" to get a connected page. If the request fails (usually because the page is being updated or maintained at this time and the data is not available), it will show that there are no new messages at this stage. If the request returns normal, the received data is compared to the previous data. If there is no news updated, the user page will not be updated and still display the previous query. If there is an update, the latest message is displayed on the user interface. The methods for suspending, resetting, and ending the monitor are the same as "Google Scholars" monitor.

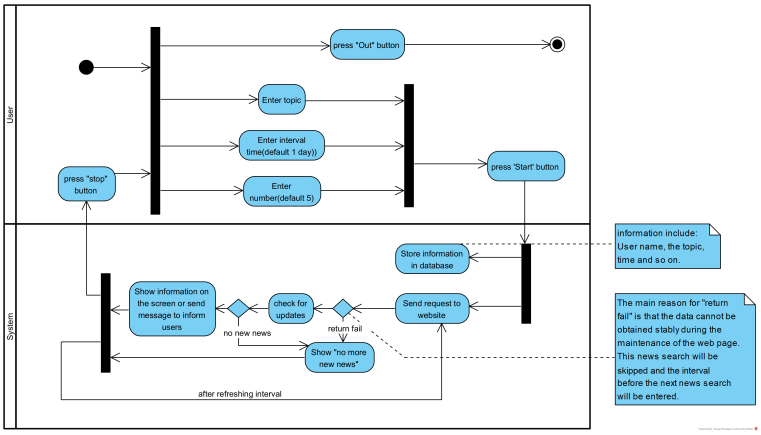

Figure 4: Sina News Activity Diagram

4.4 Operating principle of Bilibili

The following activity diagram shows the process of users using "Bilibili" to monitor videos. When the user first uses the "Bilibili" monitor, the user needs to fill out the same form as the "Sina" monitor to determine the keyword, quantity, and update time. The system will crawl the videos according to the form set by the user. If the update time is not reached, the system will go to wait. The crawling will start only after the update time is reached. If there is an update in the crawled video, it will display it on the page and send an email to notify the user that "the video you follow has been updated". If the data is not updated, the page will still display the last crawl result (no email will be sent to the user). Users can click "stop" or "delete" to stop or delete the monitor. The results of the previous monitoring will be displayed when the subsequent user reenters the page.

4.5 Operating principle of Shixiseng

The following activity diagram shows a user using "Shixiseng" to get jobs. Unlike other monitors, when users first use the monitor, they need to fill in not only search frequency, but also a resume, desired position, and expected salary. The software will also put these information into the database for administrator maintenance. After the user completes the necessary information collection, the software can start to match

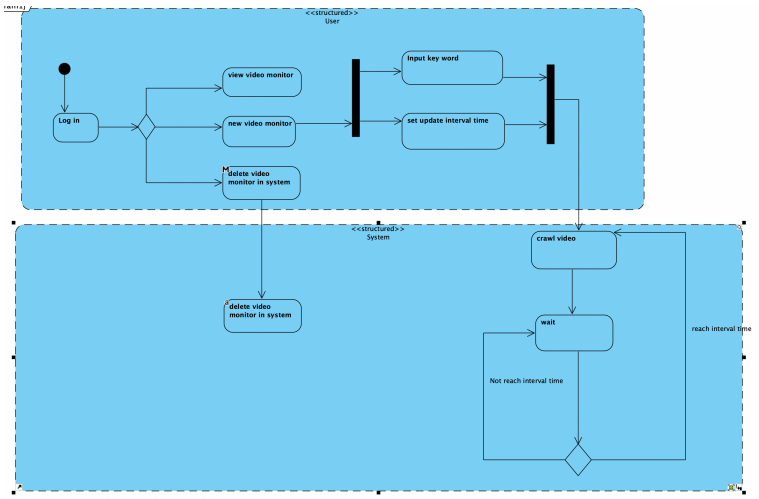

Figure 5: Bilibili Video Activity Diagram

and recommend jobs with similar conditions to the user regularly according to the information filled in. If there is No new job after matching, "No more new jobs" will be displayed. If there are new job recommendations in the next search, the software will send an email to the user or display the job information on the page to notify the user. Users can click on the jobs the software finds for them to be redirected to the corresponding "Shixiseng" resume posting page. The methods for suspending, resetting, and ending the monitor are the same as "Bilibili" monitor. When the controller is deleted, the software will also delete the user's resume from the database to ensure the security of personal information.

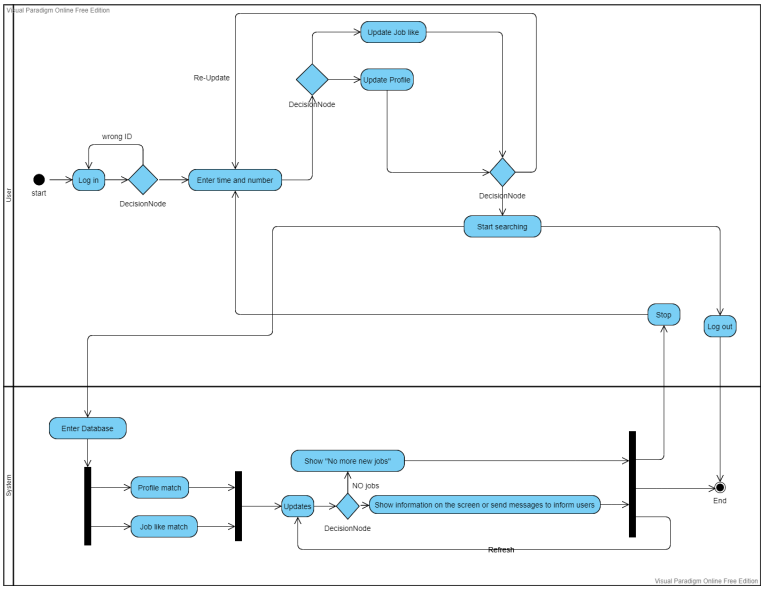

Figure 6: Shixiseng Activity Diagram

5 USER INTERFACE

5.1 Login page

Users are required to enter their username or email and password to log in. If they don't have their own accounts, they need to press the "Register Now" button and create an account by filling the form. They can go back to login page by clicking "I already have an account". After logging in, it will go into "Bilibili" directly.

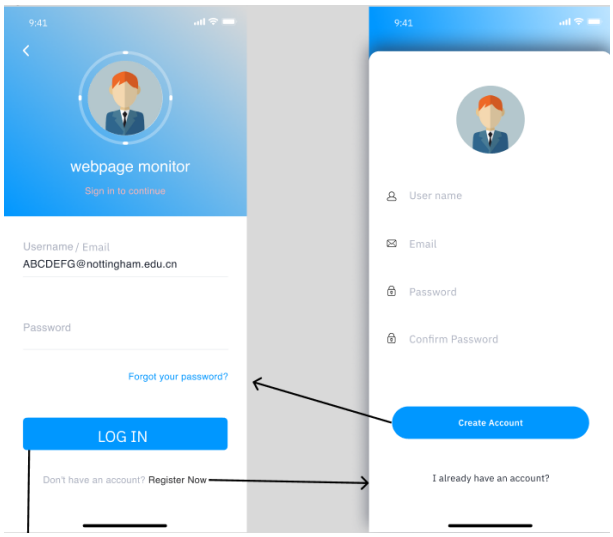

Figure 7: Log In Page

5.2 Main pages

5.2.1 Layout

The layout of main pages will be introduced with "Figure 8" and "Figure 9".

The page on the right[Figure 8] is mainly to control the state of the software and show the result of monitoring. From the top to the bottom, it consists of a button "New monitor" which is to go to the Requirements List(the page on the left), another button on the right which is to start and stop the monitoring function. There are also two text field showing the basic information of monitoring which are topic on the left and refreshing interval on the right. Next, it's the largest area on the page which is to show the result of monitoring and the components of this part in each page have a little different. On the bottom, there are five buttons for users to choose different websites to monitoring.

The page on the left[Figure 9] is the Requirements List which can be reached by pressing "New monitor" on main pages. This page is for users to fill their requirements of monitoring, including topic, the number of contents, refreshing interval and so on, which is different among each website. After filling, users can click the "OK" to go back to the main page and these information will be saved, if user would not like to fill requirements, they can go back by clicking the cross on the left-top corner which means information will be deleted. In addition, the state of Requirements List is read-only when monitor is working and can be changed after monitor is stopped.

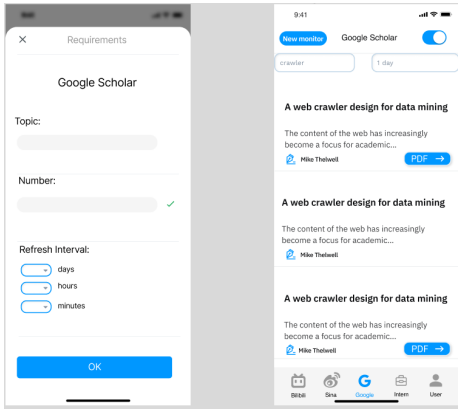

Figure 8: main page: Google Scholar

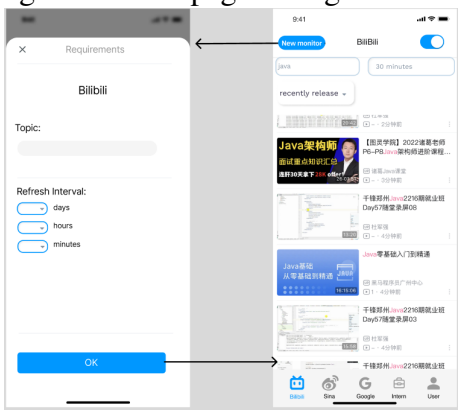

Figure 9: main page: Bilibili

5.2.2 Working flow

The working flow of main pages will be introduced with "Sina"

The page on the left is the main page users go into without monitoring. If they would like to create a new monitor, they should click the "New monitor" and go to the Requirement List(the page in the middle) for them to fill in their requirements and click "OK" to go back to the main page(the page on the left), meanwhile, the "New monitor" is changed to "Edit". Then, they should click the button on the right-top to start monitoring and the content of two text fields will show corresponding text and contents crawled from websites will be shown and refresh according to the refreshing interval(the page on the right). If they would like to change requirements, they should first click the button to stop the monitor and then click the "Edit" to go to Requirements List to change requirements.

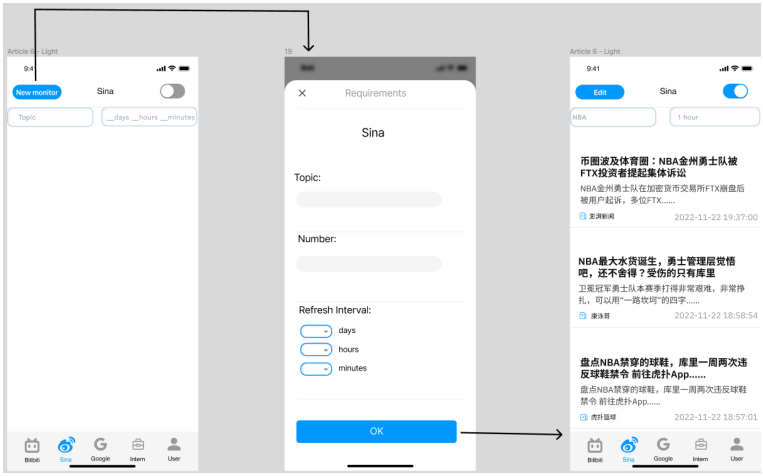

Figure 10: Working Flow

5.3 User menu

The user menu is to mainly to show users' user names, monitoring histories and so on. Users are able to search their monitoring histories, modifying their personal information, setting the user interface and contact the team members if having any questions.

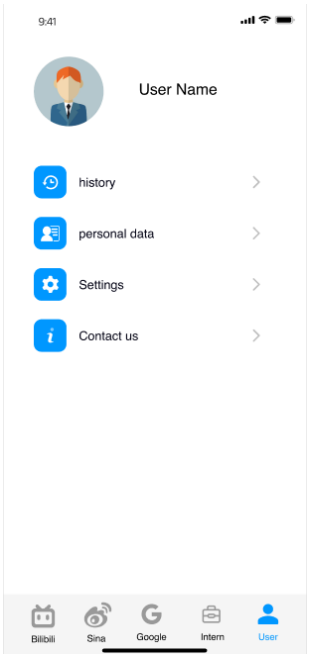

Figure 11: User Menu

6 UNDERLYING CODE

6.1 Bilibili monitor

This is a demo code to monitor bilibili. It will crawl the specified content regularly based on the keywords entered by the user. It will parse the video title, link, introduction, uploader, view counts, pop-ups, upload time and video images, which will be displayed on our front-end.

```python
def bilibili_monitor():
    userSeach = 'music'
    page = 1
    viedoNum = 0
    mainUrl = 'https://search.bilibili.com/all?keyword=' + userSeach
    mainSoup = BeautifulSoup(requests.get(mainUrl).text, "html.parser")
    pages = mainSoup.find('li', class_="page-item last")
    now_date = datetime.datetime.now()
    flag = False
    if (pages):
        pages = int(pages.text)
    else:
        pages = 1
    while page <= pages:
        mainUrl = 'https://search.bilibili.com/all?keyword=' + userSeach + '&order=pubdate' + '&page=' + page.__str__()
        mainSoup = BeautifulSoup(requests.get(mainUrl).text, "html.parser")
        for item in mainSoup.find_all('li', class_="video-item matrix"):
            viedoNum += 1
            val = item.find('a', class_="img-anchor")
            subUrl = val["href"]
            sleep(0.5)
            subSoup = BeautifulSoup(requests.get('https:' + subUrl).text.strip(), "html.parser")
            upload_time = subSoup.find('span', class_="pudate-text").text.strip()
            strftime = datetime.datetime.strptime(upload_time, "%Y-%m-%d %H:%M:%S")
            title = val["title"]
            url = 'https:' + val["href"]
            introduction = item.find('div', class_="des hide").text.strip()
            uploader = item.find('a', class_="up-name").text.strip()
            view_counts = item.find('span', title='观看').text.strip()
            pop_ups = item.find('span', title='弹幕').text.strip()
            image = subSoup.find(itemprop="image")["content"]
            if (now_date - relativedelta(seconds=50) < strftime):
                print('第' + viedoNum.__str__() + '个视频:')
                print('title:' + title)
                print('url:' + url)
                print('introduction:' +introduction)
                print('uploader:' + uploader)
```

Figure 12: bilibili monitor demo1

```python
            if (now_date - relativedelta(seconds=50) < strftime):
                print('第' + viedoNum.__str__() + '个视频:')
                print('title:' + title)
                print('url:' + url)
                print('introduction:' +introduction)
                print('uploader:' + uploader)
                print('view_counts:' +view_counts)
                print('pop_ups:' + pop_ups)
                print("upload_time:  " + upload_time)
                print('image:' + image)
            else:
                flag = True
                break

        page += 1
        if flag:
            break
    timer = threading.Timer(50, bilibili_monitor)
    print("下一次爬取...")
    timer.start()
```

Figure 13: bilibili monitor demo2

6.2 Sina monitor

This is a demo code for monitoring Sina. It will periodically fetch the specified content based on the keywords entered by the user. It will parse news headlines, links, media, upload times, and news content that will be displayed on our front-end.

Figure 14: Sina monitor demo1

The above code shows how to simulate headers messages so that user requests are not blocked by the site's "anti-crawler" technology.

```python
def gettimevalue(date):
    return datetime.datetime.strptime(date,"%Y-%m-%d %H:%M:%S").timestamp()

def getArticle(url):
    url = urlopen(url)
    soup = BeautifulSoup(url,"html.parser")
    head = soup.select(".main-title")[0].text
    date = soup.select(".date")[0].text
    article = []  # 定义列表
    if(len(soup.select("#article p")[:-1])!=0):
        for p in soup.select("#article p")[:-1]:
            article.append(p.text.strip())
    else:
        for p in soup.select("#article")[:-1]:
            article.append(p.text.strip())
    article = '\n\n'.join(article)
    return (head.rjust(60)+"\n"+date.rjust(60) + ' ' + "\t\n"+article)

def CrawlerSina(KeyWord):
    news = []
    for i in range(1,3):
        news=getData(KeyWord,i,news)
    mynews = copy.deepcopy(news)
    for i in news:
        if("http" not in i["url"] or i["intro"]==''):
            mynews.remove(i)
    mynews=mynews[:10]
    for i in mynews:
        temptext = getArticle(i["url"])
```

Figure 15: Sina monitor demo2

The functions in above figure enable the software to obtain a collection of news links containing keywords in the web interface, and then obtain a single news information (title, release time, release media, news content) from each link.

6.3 Google Scholar

This is a demo code for monitoring Google Scholar. It will periodically fetch the specified content based on the topics, numbers of essays and publish dates entered by the user. If there's an available PDF, users can click and download.

```python
def google_scholar_crawler(topic, number, daysAgo):
    page = number // 10
    rest = number % 10

    if page == 0:
        url = "https://scholar.google.com/scholar?hl=en&as_sdt=0,5&q={}&scisbd=1".format(topic)
        spider(url, number, daysAgo)
    else:
        try:
            for i in range(0, page):
                url = "https://scholar.google.com/scholar?start={}0&q={}&hl=en&scisbd=1&as_sdt=0,5".forma
                spider(url, 10, daysAgo)
            url = "https://scholar.google.com/scholar?start={}0&q={}&hl=en&scisbd=1&as_sdt=0,5".format(pa
            spider(url, rest, daysAgo)
        except:
            print("No more articles")

    timer = threading.Timer(60, google_scholar_crawler, (topic, number, daysAgo,))
    print("\n\n\nNew Loop")
    timer.start()
```

Figure 16: Google Scholar monitor demo1

```python
def spider(url, number, daysAgo):
    response = requests.get(url)
    selector = etree.HTML(response.text)

    if daysAgo == 0 and number != 0:
        for i in range(1, number + 1):
            a = selector.xpath('//*[@id="gs_res_ccl_mid"]/div[{}]'.format(i))[0]
            title = dealTitle(a)
            if title == "NULL":
                break
            if i == 1:
                LastCrawl[0] = title
    elif daysAgo != 0 and number == 0:
        for i in range(1, 101):
            a = selector.xpath('//*[@id="gs_res_ccl_mid"]/div[{}]'.format(i))[0]
            timeLimit = dealTime(a, daysAgo)
            if timeLimit == False:
                break
            title = dealTitle(a)
            if title == "NULL":
                break
            if i == 1:
                LastCrawl[0] = title
    elif daysAgo != 0 and number != 0:
        for i in range(1, number + 1):
            a = selector.xpath('//*[@id="gs_res_ccl_mid"]/div[{}]'.format(i))[0]
            timeLimit = dealTime(a, daysAgo)
```

Figure 17: Google Scholar monitor demo2

7 Key Implementation Ideas

7.1 Supporting Development Tools

7.1.1 Github - Code hosting service platform

7.1.2 Overleaf - Online collaborative LaTeX editor

7.2 Software Design Ideas

7.2.1 Back-end

- **Programming language - Python**

  1. Simple interface for capturing web pages. Python provides a complete API for accessing web documents compared to other dynamic scripting languages; Python has a simpler interface for capturing web documents compared to other static programming languages.

  2. Powerful third-party libraries. In addition, capturing web pages sometimes requires simulating browser behavior, and many sites prohibit capturing rigid crawlers. In this case, we need to simulate the behavior structure of Useragent, such as simulating user login, simulating the storage and setting of Sesion/Cookie. There are excellent third-party packages in Python to help us do this, such as Requests or Mechanize.

  3. Data processing quickly and easily captured web pages usually need to be processed, such as filtering Html tags, extracting text, etc. Python's Bython provides faster document processing, but most documents can be done with very short languages and tools.

- **Framework - Flask**

  1. Ease of prototyping: There are fewer levels of architectural abstraction, allowing for rapid prototyping.

  2. Small code base: The base library code is small, effectively reducing the size of the application.

  3. Flexibility: Developers can add external elements to the project on demand. The architecture has no strict design pattern, protocol, and database requirements. Each component can be replaced flexibly.

7.2.2 Front-end

We have tentatively decided to use WeChat applets as our front-end and use Taro+React to develop, which may change in the future.

- **WeChat applets**

  1.The development of native App involves multiple platforms of Android/iOS, development tools, development languages, adaptation to different devices, etc.;

while applets only need to develop one to run on different devices of different platforms such as Android/iOS.

2.WeChat applets do not need to be downloaded and installed, and they have small memory, do not take up much space on your phone, and run fast. When you need to use it, you can search for the corresponding app in WeChat and click into it, and when you don't need to use it, you can directly exit. Compared with traditional apps that need to be downloaded, installed and then registered, applets save time and effort and are much more convenient.

- **React - A JavaScript library for building user interfaces**
- **Taro - an open cross-platform and cross-framework solution that supports the use of frameworks such as React/Vue/Nerv to develop WeChat / Jingdong / Baidu / Alipay / ByteDance / QQmini program / H5 / RN and other applications.**

8 PROBLEMS and SOLUTION

8.1 Technical

This section will review the technical challenges which have been solved and facing now.

8.1.1 Development language

At the beginning, whether java or python should be our underlying code language cannot be decided since some of us are familiar with java and others are familiar with python. Finally, python was chosen to be the main language of underlying code. The most vital reason was that our technical lead is skilled in python who can assist us to solve complicated problems. In addition, as a dynamic language, python is more efficient than java and more suitable for a small development team.

8.1.2 Website selection

Some controversies happened on which academic website to monitor, "Google Scholar" or "ACM". The advantage of the first one is that it updates more frequently and the number of essays is considerably larger. However, unlike the second one, the structure of the first one is extremely complicated, making it difficult to crawl detailed data. Our testing manager suggested us to choose "Google Scholar", for the reason that updating frequently meaning the difficulty of testing is low which has benefits for our subsequent development.

The choice of news sites has also been controversial. Our team discussed "BBC", "CNN", "Global Times" and "Sina", and finally chose "Sina". Compared to the first two options, "Sina" is more stable and convenient for development and testing. Sina's anti-crawling technology is easier to crack and contains more news than the third option.

After the final discussion, our team finally selected "Sina" as the monitoring object of the news section.

8.1.3 User requirement

Requirement investigations with investigation technique such as interviewing or questionnaire are not conducted to avoid unnecessary difficulty, resulting that agreement on some detailed parts of our design cannot be reached. In order to precisely defined detailed functional and non-functional requirements, the aims of our software was corresponded with the results of investigations found online and summarizing our own requirements under the agreement of our supervisor.

8.1.4 Prototype design

1) Function of the software

Our initial design was to add a search page where users can directly search topics they want, in order to increase the usage of our software. However, the idea was denied by our technical lead for the reason that it will raise the complication of our software, making our main function: monitoring being less outstanding. After discussion, his ideas were accepted.

2) Design of prototype

Although all of the components were decided, our prototype cannot be made more exquisite because we have inability to design impressive patterns by ourselves. Thus, the models of prototype were bought from a online shop and suitable models were chosen and adjusted to fit our software.

8.1.5 Sharing Databases

Sharing databases can be assessed only when the host computer and computers who would like to connect to the sharing databases are connected to WIFI "edurom". After discussion, we decided to apply the databases with official VPN provided by university.

8.2 Interpersonal

8.2.1 Formal meeting

After reporting our process, problems were asked separately, lacking of unified records. As a result, awkward pauses happened repeatedly, wasting time of all of us. Thus, at the end of the second formal meeting, our supervisor advised us to take turns to be the host of the meeting who should take responsibility to collect problems, summarize the achievement of this week and the plan for next week, to make meetings more efficient.

8.2.2 Email address

We have a discussion on how to inform users that websites has updated. Finally, we decided to do this by sending emails. However, we cannot make sure that is we can collect users' email addresses without writing the ethic form. Our supervisor answered us that collecting users' email address is allowed.

9 FUTURE WORK

After discussion, four of our team members will engage in developing front-end code, they will spend two to three weeks to learn how to design user interfaces with React. The other team member will engage in developing back-end code, he will spend two to three weeks to learn how to implement underlying code with flask structure and complete the back-end code. After that, our plan is to test and maintain our code.

In the result of the COVID-19 virus, we have to delay the deadline completing our code. In order to complete our tasks on time, we plan to compress the time spending in testing, documenting and packing, meaning that we have to achieve these goal with higher efficiency.

The new planning time line is shown in the table [Table 1].

Task	Start Date	Dead Line
Completing front-end and back-end code	26/12/2022	05/02/2023
Test	06/02/2023	26/02/2023
Maintance	27/02/2023	14/03/2023
Final Group Report	27/02/2023	06/04/2023
Documentation and Packing	15/03/2023	18/04/2023

Table 1: Time Line

10 REFERENCE

[1] Mallawaarachchi, V., et al. (2021). "Change Detection and Notification of Web Pages." ACM Computing Surveys 53(1): 1-35.

[2] Meegahapola, L., et al. (2017). Detection of change frequency in web pages to optimize server-based scheduling. 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer).

[3] B. E. Smith. 2008. Creating Web Pages For Dummies (9th ed.). John Wiley&Sons Inc, New Jersey, USA.

[4] Technical Report : Monica P., Kamyar D. (2005) How search engines work and a web crawler application.

[5] montastic. Retrieved December 7,2022, from https://montastic.io/

[6] distill. Retrieved December 7, 2022, from https://distill.io/about

[7] Diallo, M. S., et al. (2018). "A search engine dedicated to the environment around the Congo basin area: Model and Architecture." ICST Transactions on Ambient Systems 6.

[8] Gusenbauer, Michael (2018-11-10). "Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases". Scientometrics. 118: 177–214. doi:10.1007/s11192-018-2958-5. ISSN 0138-9130. S2CID 53249161.

[9] bilibili. Retrieved December 9, 2022, from https://www.bilibili.com/read/cv18496411/

[10] Sina Science and Technology. Retrieved December 9, 2022, from https://finance.sina.com.cn/tech/2021-09-23/doc-iktzscyx5808785.shtml?_zbs_baidu_bk

[11] Baidu. Retrieved December 9, 2022, from https://baike.baidu.com/item/%E5%AE%9E%E4%B9%A0%E5%83%A7/22895722

11 Minutes

11.1 Summary of Team18 First Formal Meeting

Date (MM/DD/YY):	10/21/22
Present member:	All
Absent Member and Reason:	
Content:	language： python Web(initial)：Douban AMS(https://journals.ametsoc.org/view/journals/aies/aies-overview.xml), google scholar design: divide into several modules
Task and deadline:	Decide which website to choose (10/24)
If supervisor attend:	Y

Question	Answer
Ethic form	No ethic problem
how to deploy our software	any platform is OK
which language should we choose	any language
which website can we choose	free to choose and supervisor will discuss with us if it's suitable

11.2 Summary of Team18 Second Formal Meeting

Date (MM/DD/YY):	10/28/22
Present member:	Yuhao MA, Chi MA, Shujun JIANG, Haowei GAO
Absent Member and Reason:	Ruizhe Huang was absent from the meeting because he went home.
Content:	Web(v1): news: http://english.sina.com/ essay: https://scholar.google.com video: https://www.bilibili.com economy: http://www.stcn.com/article/list/djjd.html (alternative offer) Role: Technical lead: Yuhao MA Implementer: Chi MA Testing: Shujun JIANG Documentation: Haowei GAO Project manager: Ruizhe HUANG Every minute should have a moderator to gather the topics to be discussed before the meeting and controls the meeting pace. The moderator also needs to report on the progress of the previous week's work. Task1: Collect relevant papers and literatures about network monitoring robots and crawls. Task2: Look for projects or successful commercial products in development related to network monitoring robots and get background information about the technologies they use. Collect information about the market situation, prospects, etc of this product. Task3: About the crawler function development experiment on Google Scholar. Task4: About the crawler function development experiment on Sina English news Web crawler.
Task and deadline:	1: Task1 primarily responsible —— Chi MA 2: Task2 primarily responsible —— Haowei GAO 3: Task3 primarily responsible —— Yuhao MA 3: Task4 primarily responsible —— Shujun JIANG
If supervisor attend:	Y

11.3 Summary of Team18 Informal Meeting

Date (MM/DD/YY):	11/1/2022
Present member:	All
Absent Member and Reason:	
Content:	build project (gitlab, private) prepare a new website (https://www.shixiseng.com/)
Task and deadline:	obtain all the required data, combine the code with flask (Y.MA JIANG HUANG) (ddl: 11/8) read DMS ppt and discuss UML diagrams (HUANG C.MA GAO)(11/4) look up for comparing algorithm (to deal with new web)(Y.MA JIANG)
If supervisor attend:	N

11.4 Summary of Team18 Third Formal Meeting

Date (MM/DD/YY):	11/4/2022
Present member:	ALL
Absent Member and Reason:	
Content:	build project (gitlab, private) prepare a new website (https://www.shixiseng.com/) Y.MA: Bilibili. Obtain name, link. Sort as title, amount of play, time, number JIANG: Xinlang，Obtain title, content except picture. Sort as title, time, number HUANG: Google Scholar, Obtain title, PDF(link). Sort as title, time,number C.MA: relevant, software, essay and summarize them as a report GAO: web robot, analysis for competitive products analysis for competitive products What to do next：Combine code to FLASK frame Think about the final presentation function of the software Analyze the example of interim report and final report
Task and deadline:	obtain all the required data, combine the code with flask (Y.MA JIANG HUANG) (ddl: 11/8) read DMS ppt and discuss UML diagrams (HUANG C.MA GAO)(11/4) look up for comparing algorithm (to deal with new web)(Y.MA JIANG) JIANG, HUANG: change the loop to non-blocking (ddl:11/8) JIANG, HUANG, Y.MA: ignore repeated content(ddl: 11/8) ALL:read GRPHandbook about interim report (All)(ddl:11/8)
If supervisor attend:	Y

Question	Answer
Which UML diagram should be drawn	it should be as detailed as possible
Interim report	Should contain most of the functions of our software

11.5 Summary of Team18 Informal Meeting

Date (MM/DD/YY):	11/8/2022
Present member:	All
Absent Member and Reason:	
Content:	1. Read the example report given by supervisor and learn how to write report
Task and deadline:	Find resource online to complete the “Background” part(C.MA, GAO) Think about technical aspect for the interim report(Y.MA) Think problems so far(HUANG) FOR ALL: Draw draft about UML Diagram
If supervisor attend:	N

11.6 Summary of Team18 Fourth Formal Meeting

Date (MM/DD/YY):	11/12
Present member:	All
Absent Member and Reason:	
Content:	HUANG: non-blocking loop，separating the time and number JIANG: non-blocking loop Y.MA: monitoring the website，avoid obtain repeated contents C.MA: find essay about crawler GAO: improve analysis for competitive products
Task and deadline:	HUANG: google scholar activity diagram, complete user case diagram C.MA GAO: Shixiseng activity diagram Y.MA: bilibili activity diagram JIANG: xinlang activity diagram (ddl: 11/15) Interim report: HUANG: 1 8 JIANG: 7 GAO: 2 3 Y.MA: 2 3 C.MA: 5 4,6 write after creating prototype 8 appending meeting records
If supervisor attend:	N

Q	A
Can we choose database which is not provided by university	Yes
Can we write our report with markdown	No, we have to use latex, because .md file have to be opened with other softwares
How to deal with user requirement	We can find it online, but need to correspond to our project, if we cannot find proper one,

	we can draw up by ourselves(e.q. informal talk)

11.7 Summary of Team18 Fifth Formal Meeting

Date (MM/DD/YY):	11/18
Present member:	All
Absent Member and Reason:	No
Content:	HUANG: prototype draft part1 Interim reporting Framework JIANG:Sina UML diagram Y.MA: bilibili UML diagram C.MA: Internship UML diagram GAO: prototype draft part2
Task and deadline:	Interim report plan: HUANG: 1 8 JIANG: 7 GAO: 2 3 Y.MA: 2 3 C.MA: 5 4,6 write after creating prototype 8 appending meeting records (ddl: 11.25) Prototype (ddl: 11.22)
If supervisor attend:	Y

Q	A
Do images in prototype need to be approved by the copyright holder?	Yes required
Write in the interim report whether the final presentation, e.g. a web page or applet, can be changed at a later stage of design?	Yes, you can include it in the interim report and add the option requirement
The aim of the interim report	As an aeeseement, Summarize the current progress. Not everything in the report can be changed at will

11.8 Summary of Team18 Sixth Formal Meeting

Date (MM/DD/YY):	11/25
Present member:	ALL
Absent Member and Reason:	No
Content:	report our process: final version of Prototype Interim report
Task and deadline:	The first edition of interim report(ddl: 11/29)
If supervisor attend:	Y

Q	A
If we can collect uses' email address for us to send them emails when uploading	Use free, don't need to fill the ethic form

11.9 Summary of Team18 Informal Meeting

Date (MM/DD/YY):	11/29
Present member:	ALL
Absent Member and Reason:	No
Content:	Introduction: Added content and fixed syntax errors User Interface: Added the user mailbox login page
Task and deadline:	
If supervisor attend:	

11.10 Summary of Team18 Seventh Formal Meeting

Date (MM/DD/YY):	12/9
Present member:	ALL
Absent Member and Reason:	
Content:	1. Telling supervisor that the first edition of the interim group report has been finished and ask him to give us some advice 2. Discussing members who are in charge of front-end code and back-end code **Result: front-end : Haowei GAO, Yuhao MA, Shujun JIANG, Chi MA** **back-end: Ruizhe HUANG, (Assist: Yuhao MA, Shujun JIANG)** 3. constructor of front-end: React back-end: Flask
Task and deadline:	1. Think and write **individual plan** (ddl: 12/16) 2. study how to develop software with particular constructor
If supervisor attend:	Y

11.11 Summary of Team18 Eighth Formal Meeting

Date (MM/DD/YY):	12/16
Present member:	ALL
Absent Member and Reason:	
Content:	The problems of the Interim Group Report: 1. retract 2. grammar 3. tense 4. index of figures 5. name of figures 6. figures should correspond with texts
Task and deadline:	
If supervisor attend:	