

# A Mental Health Dialogue System

ZHANG Yizhe

M.Sc. Practicum Report

Department of Computer Science

Hong Kong Baptist University

2022

# A Mental Health Dialogue System

ZHANG Yizhe

An M.Sc. practicum report submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science

Principal Supervisor: Dr. Li Chen

Department of Computer Science

Hong Kong Baptist University

May 2022

Acceptance of M.Sc. Practicum Report  
Master of Science in Data Analytics and Artificial Intelligence

Department of Computer Science  
Hong Kong Baptist University

We hereby recommend that the M.Sc. Practicum Report submitted by ZHANG Yizhe entitled "A Mental Health Dialogue System" be accepted in partial fulfillment of the requirements for the degree of Master of Science in Data Analytics and Artificial Intelligence.

\_\_\_\_\_  
Dr. Li Chen  
Supervisor

\_\_\_\_\_  
Dr. Martin Choy  
Observer

Date: \_\_\_\_\_

Date: \_\_\_\_\_

## Declaration

I hereby declare that this M.Sc. Practicum Report represents my own work which has been done after registration for the degree of M.Sc. at the Hong Kong Baptist University, and has not been previously included in a project report, thesis or dissertation submitted to this or other institution for a degree, diploma or other qualification.

---

ZHANG Yizhe

Date: \_\_\_\_\_

## Abstract

Recommender systems are popularly applied in various fields now. It can analyze user behaviors or product features to predict personal preferences for a specific user. The development of natural language processing led the recommender systems to grasp the ability to “talk” with users for eliciting preferences more naturally. Compared with the face-to-face psychological counseling process, the conversation recommender system (CRS) is more individualized, private, and convenient. The CRS can help medical institutions with preliminary diagnoses. Those users with mild symptoms may just need to go through it to get the necessary help rather than resort to medical resources. With this report, we developed a mental health platform that contains a conversation recommender system. The chatbot contains a set of dialogue flows. Users can be guided by it to complete a mental assessment. The result will be used to establish the user model and produce recommendations. We describe in detail the principle and mechanisms of mental health assessment, dialogue chatbot, and recommendation system in this platform. Moreover, we demonstrated the front-end and back-end development principles and interactions of all interfaces. Finally, we show the result of a system usability investigation.

Keywords: Conversation Recommender system, Mental Health Assessment.

## Acknowledgments

I would like to express my most gratitude to my supervisor, Dr. Li Chen, for her knowledgeable guidance and supervision in my project work. Through her help, I have learned how to research and implement recommender systems. I would also like to thank my friend Ms. Weiping Chen, whose experience in psychology has given me a lot of advice.

Finally, I am deeply grateful to my parents, for their support, encouragement, and patience.

# Table of Contents

1. Introduction .....	1
1.1. Mental Health Assessment.....	1
1.2. Dialogue Chatbot .....	3
1.3. Recommender System.....	7
1.3.1. Information retrieval (IR) similarity measure.....	7
1.3.2. Term Frequency Inverse Document Frequency (TF-IDF).....	9
2. Front-end and Back-end Server.....	12
2.1. Front-end Design.....	12
2.2. Database .....	14
2.3. Back-end Server .....	16
3. Demonstration .....	21
3.1 Dialogue interaction .....	21
3.2 Users end interaction.....	26
3.3 Researchers end interaction.....	29
4. Investigation .....	32
5. Conclusion and Future work .....	34
Reference .....	35

# 1. Introduction

In this chapter, we demonstrate the core concepts of our projects. The major part is that I will provide the literature review to show the method of mental health assessment called the Kessler Psychological Distress Scale (K10). And how to convert it to the dialogue flow.

## 1.1. Mental Health Assessment

After the end of World War II, some psychologists were beginning to use the dimensional scales to assess the non-specific psychological distress to primary filtrate the potential mental patients before the in-depth clinical assessment [1]. Unfortunately, researchers did not continually utilize the dimension scales on the follow-up surveys. For a long time, it was substituted by structured diagnostic interviews. But the drawback of structured diagnostic is obvious, the governments want to distinguish community cases based on severity rather than purely on diagnosis positive or negative. The dimensional measure of non-specific psychological distress can fulfill the demand.

In our project, we aimed to design a chatbot that can talk with users to guide them to complete an assessment to produce the user model. This model needs to contain the index of cognitive, behavioral, emotional, and psychophysiological symptoms. Therefore, it is not a simple diagnosis result. We need to adopt the dimensional scales of non-specific psychological distress to collect enough information for producing the user model. In addition, because users would not like to spend a long time assessing. Therefore, the time limitation requires us to find a short screening scale to do in the field of non-specific psychological distress. During this time, we found that the US National Health Interview Survey has the same limitation when it tried to collect the population prevalence and trends in the same field. It has severe time constraints, so the process was required to have no more than six to eight questions [2]. To meet demand, Kessler and his team tried to design a short scale. Their conceptualization of this task came



from the research of Dohrenwend and his colleagues. Dohrenwend reviewed the scales of nonspecific psychological distress and found that they commonly contained questions about a heterogeneous set of cognitive, behavioral, and emotional symptoms that are more severe in the set of people who has various mental disorders [1].

Table 1. K10 dimensions and questions [2]

<b>Dimensions</b>	<b>Questions</b> <b>During the last 30 days, about how often did ...*</b>
<b>Tired out</b>	... you feel tired out for no good reason ?
<b>Nervous</b>	... you feel nervous ?
<b>So nervous</b>	... you feel so nervous that nothing could calm you down ?
<b>Hopeless</b>	... you feel hopeless ?
<b>Restless</b>	... you feel restless or fidgety ?
<b>So restless</b>	... you feel so restless that you could not sit still ?
<b>Depressed</b>	... you feel depressed ?
<b>So depressed</b>	... you feel so depressed that nothing could cheer you up
<b>Worthless</b>	... you feel worthless ?
<b>Everything an effort</b>	... you feel that everything was an effort ?

Based on the previous research, Kessler decided to assess the core dimension of non-specific psychological distress on the K10 scale. In the measure beginning, he chose 612 questions to compose a scale from existing screening scales. The new scale was decreased to 235 by removing redundant and unclear questions. This new set focused on the 15 domains represented in the DSM-III-R diagnoses of major depression (MD) and generalized anxiety disorder (GAD) plus the positive affect domain [2]. All the questions were submitted to vote by experts. Only the questions claimed as clear can be retained by all experts consistently (45 questions on 16 dimensions). According to two surveys and the item response theory (IRT) analysis, they set out a couple of conditions for items selection. First, in order to select the item which has maximum sensitivity, each item was required to have 0.8 standardized severity parameters. Another requirement is that all the items need have consistent severity values across sociodemographic subsamples [2]. It can avoid the question biases of a specific population.

After completing all data analysis. The final scale contains 10 questions on 7 dimensions. There are three special dimensions that correspond to two problems respectively (Table 1). This questionnaire uses a five-category response scale (all, most, some, a little, never of the time).

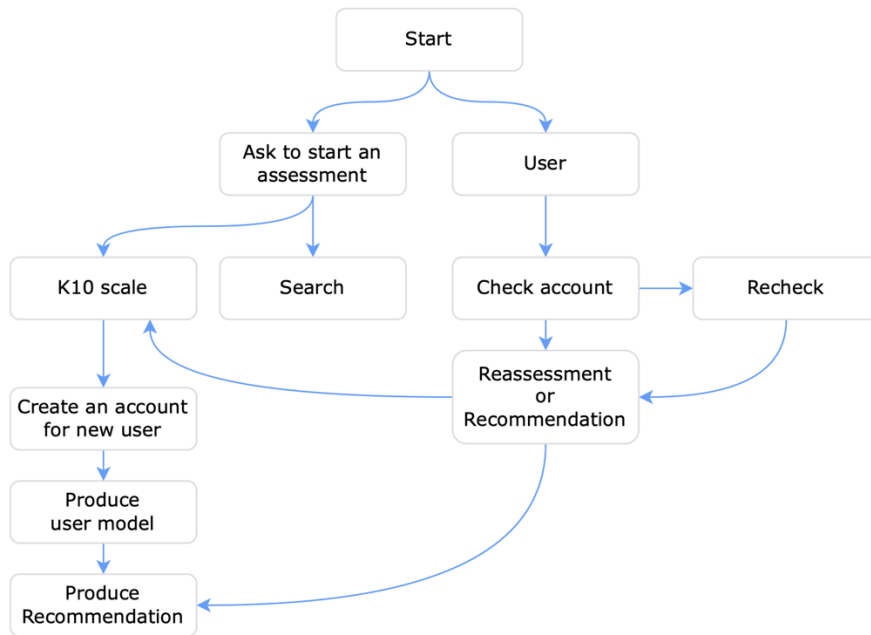
After the user completed the survey, it sums the number of items as the result to show the severity of non-specific psychological distress.

In our project, we utilized the K10 questionnaire to establish the user model. It contains ten dimensions that correspond to the questions of K10. Also, before each article is stored in the database it will be manually tagged with the primary dimensions through the content. When a user completes the assessment, the system will produce the related user model according to the K10 items scores. Finally, the user model will be compared with the article database by the cosine similarity algorithm to generate the recommender content list. Because the K10 scale reflects a period of mental situation. Users can be assessed regularly to update their model. This advantage can help the recommender system adjust the result to relative the newest user preference.

## 1.2. Dialogue Chatbot

Because the dialogue chatbot aims to provide a natural conversation with users. It needs to extract the users' intents from the context and then give them responses by the designed chat flows. We utilized the Google Dialogflow engine to design the chat flows. It has a natural language process engine that has been trained by many dialogues. It separates the conversation from one-by-one intents. Before deploying the chatbot. It asked us to design the chat flow in a visual way (Figure 1). The image contains a couple of dialogue flows and consists by a set of rectangles (each rectangle is a page). Because the Dialogflow CX principle is a state machine. It transfers from one state to the other state during the conversation. Each state controls the current progress of the conversation.

Figure 1. Dialogue flow



When a conversation starts, the state machine activates the start page. The start page can handle several intents. Each intent contains two parameters training phrases and parameters. The training phrases are a set of possible inputs from users. When users type content to the chatbot, the engine tries to search the matching intent, Dialogflow scores potential matches with an intent detection confidence.

If the highest-scoring intent is greater than or equal to the threshold of the classification setting, the page which contains this intent will be set as the current page. Then the corresponding content will be returned to users as a response. In addition, each active intent relevant parameter also be kept during the rest of the session. If we want to collect a set of data from one conversation, it can help us to collect all the parameters after the end of the conversation.

In our project, according to the features of the Google Dialogflow CX. Firstly, we separated the original questions into several pages. And transferred the assessed dimensions to natural expressions. Such as tired out is transferred to *'Wonderful! It will only spend a few minutes.'*

*During the last 30 days, about how often did you feel tired out for no good reason?'*. It will make the user more confident that he/she is talking with a human, not a robot.

Figure 2. The K10 Scale Dialogue flow

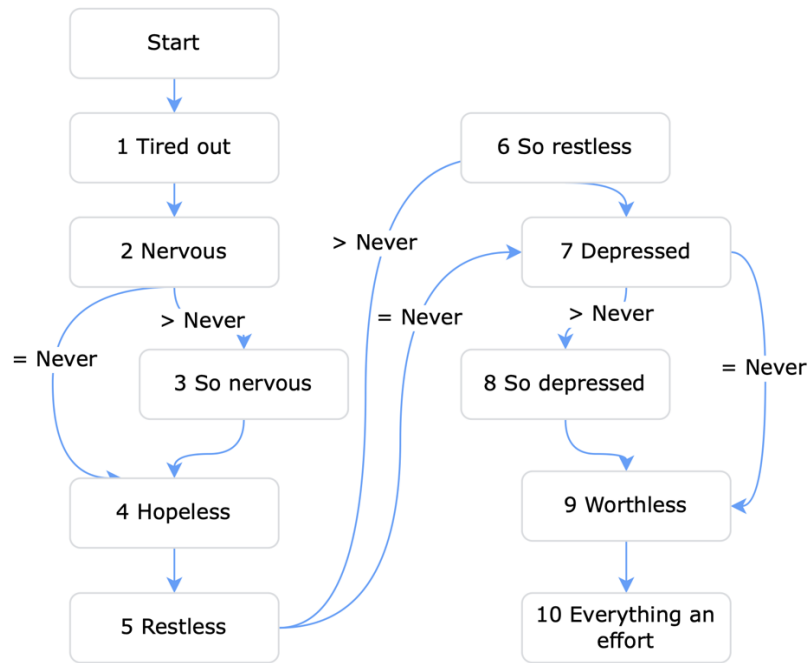


Figure 2 demonstrated the assessment process dialogue flow. When the user greets the chatbot, it enquires the user whether approves to proceed with a series of dialogues to assess their situation. If they refused, the chatbot guides them to input some keywords and search suggestions from the knowledge database. If they say yes. The chat flow will transfer from the welcome page to the assessment page which contains a set of pre-defined dialogues to draw the user model. You can see the whole dialogue flow has 10 questions, but the 2-3, 5-6, and 7-8 are three pairs of questions. The second one is more detailed compared to the first on a pair of questions. If the user responds to the first question is never, it will skip the second one.

In addition, we do not want to make users feel the process is to fill out a questionnaire. Therefore, each answer to the corresponding question has a specific reaction. Table 2 is an example of how the chatbot takes a reaction to the user's answer. The reaction will be selected

by the severity of the question. If a user feels so nervous all the time that nothing can calm him/her down, the chatbot considers that the user may not have enough patience to complete the assessment. Therefore, it will say ‘*I know it is very hard for you how to make yourself calm down. I can give you some articles that contain different ways to try.*’ It makes a very different experience for users compared to the traditional assessment. And we hope the chatbot reaction can encourage users to complete the process.

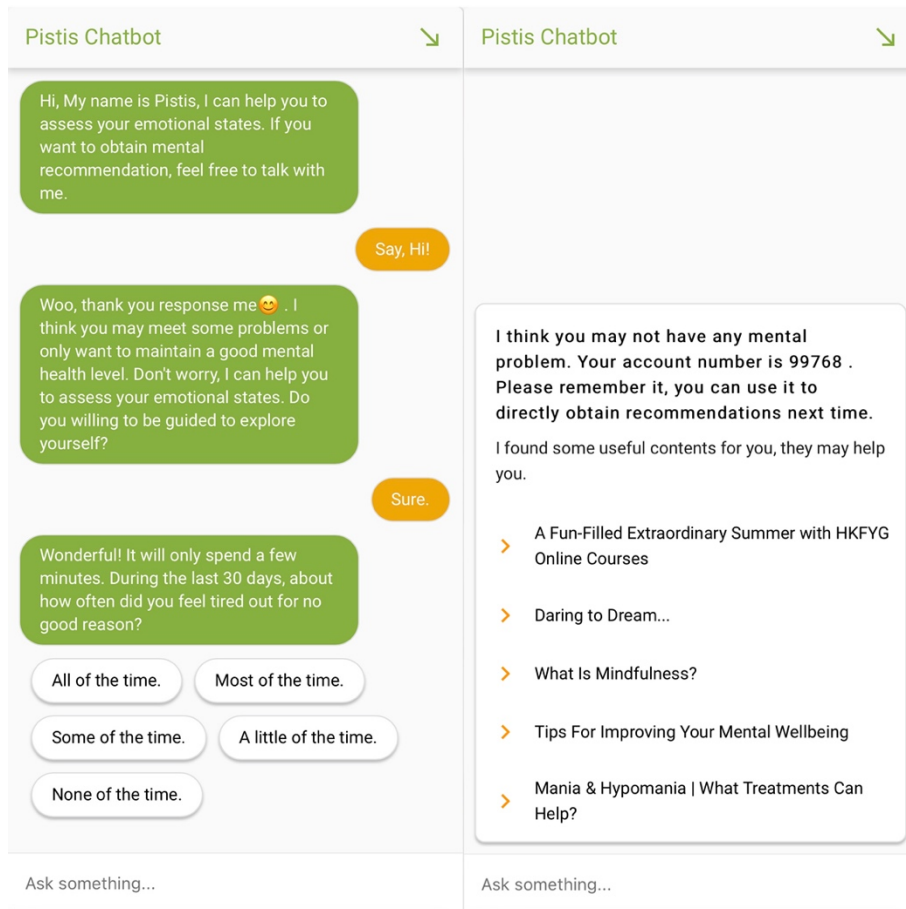
Table 2. An instance of the question reactions

Index	Dimension	Question	Answer	Reaction
3	So nervous	... you feel so nervous that nothing could calm you down?	All of the time	I know it is very hard for you how to make yourself calm down. I can give you some articles that contain different ways to try.
			Most of the time	Yeah, it's hard to relax sometimes. You can overcome it by mental methods.
			Some of the time	Deep breathing may be a good idea. Some people think mindfulness helps them to calm down. I will give you some relative content.
			A little of the time	Got it!
			None of the time	

Another important part of the chatbot is the interface design. To ensure users can complete the whole questionnaire easily, although our chatbot supports the ability of natural language process. We still set a couple of buttons to follow each paragraph to help users provide feedback without typing anything (Figure 3). The main interface color is green to give users a calm conversation environment. When users visit our website, the chatbot will not unfold directly. It was hidden as a small icon at the bottom-right corner of the page. The icon will pop a small chat bubble to greet users to talk with it. If users click the bubble, the chatbot will unfold to a whole conversation box. Users can tell the chatbot whether to start an assessment or log in. Users can click the related respond button or type their answers during the whole progress of the assessment. After the ending of the conversation, the chatbot will send the user’s answer to

the back-end server to analyze the user model. The server will compose the final response after completing the calculation and producing the corresponding recommendation. The right-side image in figure 3 demonstrated the final response. Users can obtain their score and a set of articles about their mental situation.

Figure 3. The chatbot interface



## 1.3. Recommender System

### 1.3.1. Information retrieval (IR) similarity measure

Through the research from Dohrenwend and his team, we know that the scales of non-specific psychological distress theory are to detect the anomaly index in the cognitive, behavioral, emotional, and psychophysiological symptoms [1]. Therefore, Kessler designed the K10 scale by this principle to extract the most important symptoms through several surveys. Each

dimension of the K10 scale can effectively represent a separate aspect of the people's psychological indexes. Therefore, we considered utilizing their research fundamental to establish the user model. This user model contains 10 items from 7 dimensions. Including tired out, nervous, hopeless, restless, depressed, worthless, and everything an effort. These seven symptoms can be used to describe the psychological articles as well. Therefore, we decided to use them to tag each article manually on our database (all articles were collected by the Internet, we just use it to analyze the user preference, the original content will not be provided to users by our website). We use an IR similarity measure to match the user model and corresponding articles.

Figure 4. The example of Information Retrieval similarity measure

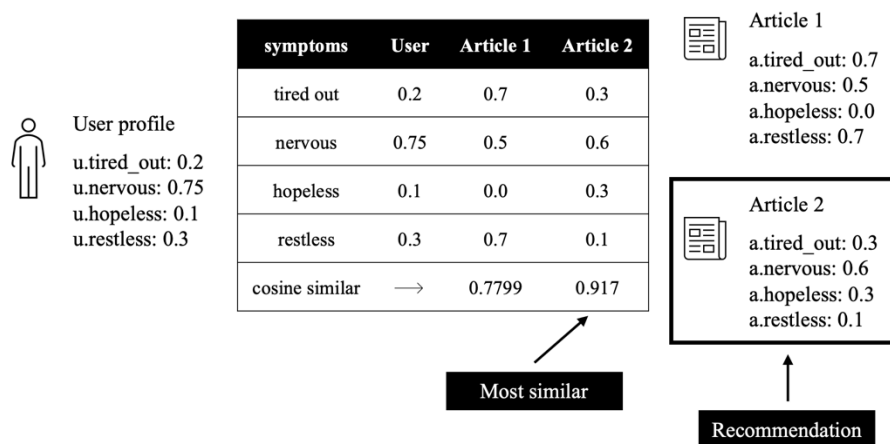


Figure 4 demonstrated this procedure. Both user models and articles have consistent dimensions. Therefore, we can adopt the cosine similar algorithm to find the most similar one. In the meantime, if we want to show a set of articles to users, we can sort the similar values and select the top-K to create an article list. According to this way, we successfully overcome the cold-start problem of the recommender system. After users completed the assessment, the server calculates the IR values and supplies a couple of articles to users (Figure 3) as well as an account number. The user can utilize the account to log in to our platform without re-assessment.

After we invited a few bit participants to experience the first stage of the project, we collected the advice about the articles' dimensions does not very suitable for their real situation. Therefore, we adjusted the browser to supply a couple of feedback buttons to users. If they like the article, they can click the like button to provide their own taste. Then we will average all users' dimension values who have similar taste in the same article. These new dimension values will substitute the original values of the article. The average dimension values will only contain the user who provided the feedback later than the date of their own assessment. If they reassessed their dimension values, their previous feedback will not be included in the calculation cause their user model is not suitable to relate to the previous preference. If users have a high level of a specific dimension, they would like to seek more content about this dimension. Therefore, according to this way. The dimension values of articles will be adjusted to suit the population's taste.

### 1.3.2. Term Frequency Inverse Document Frequency (TF-IDF)

Although the IR similar measure can help users to obtain the primary recommendation, it still has limitations. Because it only relies on the scores of symptoms, the articles which have the same dimensions but belong to different topics will cross appear in the same list. Users may only prefer a particular topic. Therefore, our recommender system is a hybrid system. It is not only containing the IR similarity measure but also contains TF-IDF as the auxiliary method to give users more precision selection.

TF-IDF is an efficient and simple mathematical function for seeking the articles that are related to the user's query. Its process is to determine the relative frequency of words in a particular article compared to the inverse proportion of that word over the entire article corpus [3]. By contrast, the traditional way is the vector-based method for query retrieval. Berry and his team utilized the SVD algorithm to extract the word vector which grasps an n-dimensional representation of a set of articles [4]. When a query is initiated, its numerical representation



will be used to calculate the cosine distance to the word vector. The researchers found this algorithm was highly effective in query retrieval by the experiment [4]. But we found an important drawback to this method. It cannot filter the common words automatically. Such as we consider a set of articles on the medical topic, A lot of medical terminologies will normally appear in a large percentage of this dataset. They will disturb the process of searching. Therefore, the word vector needs to omit a lot of high-frequency common words manually. Term Frequency Inverse Document Frequency (TF-IDF) proposed a new algorithm that can avoid the above drawback for the frequency calculation.

TF-IDF works by calculating the relative frequency of words in a particular article compared to the inverse proportion of that word over the whole corpus. If a word has a high TF-IDF value, it has a high frequency on a specific article as well as it only appears on a small number of articles from the entire dataset. This way can ignore the common words automatically.

TF-IDF has some varieties that are minor differences. The below expression is our project's version. Given a word  $W$ , a collection of articles of  $C$ , a specific article  $A$  that belongs to the  $C$  collection. And a particular word  $P$  has the most appearing frequency in the article  $A$ .

$$TF - IDF(W, A) = \frac{\frac{|W|}{|P|} \times \log_2 \frac{|G|}{|G(w)|}}{\sqrt{\sum_{i=1}^{|G|} TF - IDF(W, G_i)}}$$

$|W|$  and  $|P|$  are the number of words  $W$  and  $P$  appear in article  $A$  respectively.  $|G|$  is the size of the corpus, and the  $|G(w)|$  is the size of the sub-corpus that all articles contain the word  $W$ .

Assume that the word 'Classmate', highly appears on the campus topic of the articles. Therefore, if the user clicks the like button on an article that talks about school bullying. The TF-IDF algorithm can compare this article to the entire corpus for seeking similar content of articles.

By combining the two different recommender methods, the system can efficiently provide articles to suit user preferences. Users can not only reassess their user models whenever, but they can also get the latest articles continuously by providing feedback.

## 2. Front-end and Back-end Server

### 2.1. Front-end Design

For now, more and more people are willing to use multiple devices for browsing the Internet. Our project utilizes a modern responsive webpage framework called Bulma to build the front-end pages. It is developed by packaging Cascading Style Sheets (CSS) framework that contains several components and has the ability of modular. Therefore, we can easily combine an entire webpage without writing any CSS code.

Table 3. Bulma screen specification rules

Mobile Up to 768px	Tablet Between 769px and 1215px	Desktop Between 1024px and 1215px	Widescreen Between 1216px and 1407px	Full-HD 1408px and above
mobile				
	tablet			
	desktop			
	widescreen			
	full-HD			
	table-only			
		desktop-only		
			widescreen-only	
	touch			
	until-widescreen			
	until-full-HD			

In our project, we adopted this framework to build responsive web pages to achieve a consistent experience in the different sizes of screens. The traditional scheme needs to set the pixel or ratio size of each component. By contrast, Bulma separated the screen specifications into five levels, mobile, tablet, desktop, widescreen, and full-HD (Table 3). Each notation will be inserted into the class label of components to indicate the responsive behavior. By default, each component contains the notation of mobile which means it will display two different formats on mobile devices and others respectively. For example, we have a set of cards that needs to display. If the browsing device is a desktop, we consider showing the card list as a horizontal list because it can contain more information. By contrast, if users read the card list on a mobile phone. A horizontal list will squeeze the cards very small and cannot be read comfortably.

Therefore, it will be reordered as a vertical list (Figure 5). Each card will fill the entire screen width to ensure the reading experience. These notations not only can dedicate the arranged direction, but they also can indicate a component whether displayed or not. For instance, on the widescreen devices. Normally, it contains two columns, a small one shows a menu, and a big one shows the main content. But this layout does not satisfy the tablet and mobile phone. The menu should be set with the notation of is-hidden-touch which indicates the menu column will be hidden on the small screens.

Figure 5. Two different layouts by Bulma



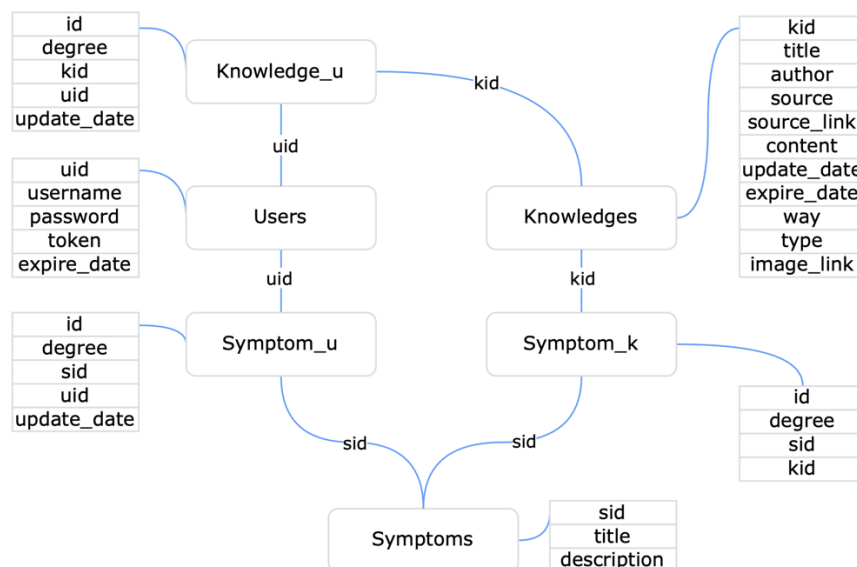
By the ability of screen responsive. Our project contains a couple of display schemes to adapt to the different environments. But it is not enough to solve the layout requirement. It only roughly divided the screen size into 5 types. And for the design requirement, we cannot simply arrange each component's vertical display one by one. The Bulma contains another sub-framework to achieve the flex layout. It is a more precise tool to help us layout the complex design. In the flex layout model, the sub-components have flex sizes corresponding to their

parent component. They will fill the blank space or squeeze themselves to avoid overlapping each other. And they will automatically arrange their arranged direction to horizontal or vertical alignment by the rules. Following the model rules, it separates the parent container into twelve parts. Each sub-container can mark to fill from one part of the parent container to all parts. If the sub-containers do not mark, they will average the twelve parts to fill. But the average number cannot be small than one. If the number of sub-containers is larger than twelve, the rest containers will be arranged in the next row (Figure 5).

## 2.2. Database

The database is an important fundamental service in our project. It ensures data security and persistence. In addition, the modern database supports maintaining the relationship between tables. It is very useful for avoiding data redundantly.

Figure 6. Pistis Health database structure



Our project contains two different roles of tables. One is a storage table, its responsibility is to store the attributions of entities, such as user information, the detail of articles, and symptoms. Another one is the relationship table. Its responsibility is to link different types of entities, such

as the relationship between users and articles. The users' feedback on articles will be stored in the relationship table.

We adopted MySQL software as the database system. It is an open-source relational database management system. It not only supports storing and managing the data and relationships but also has good compatibility to cooperate with different web server programs. Figure 6 shows the database structure of our project (It does not include the fundamental tables of the Web server). In our project, we have three storage tables to store the entity information. They are the users table, the knowledges table, and the symptoms table. The users table contains the basic personal information, each account has a unique id to identify and link with other tables. The knowledges table stores the detail of articles. Such as title, author, source, content, etc. Each entity has a unique id as well. The third table is called symptoms. It contains a description of each symptom. Users' feedback is saved on the relationship table called *Knowledge\_u*. It will only store the user id, knowledge id, creation date, and the corresponding feedback. Therefore, it can be queried very fast because its size is very small. If we want to analyze all feedback about a specific article, we can filter this table directly without scanning the whole users table and knowledges table. This type of relationship table also exists between the users table and the symptoms table. If we change the description of a symptom, we do not need to change the value one by one for all users. The *Symptom\_u* relationship table maintains the relation by the user id and the relative symptom id. The changes of users' or symptoms' information will not affect it. The *Symptom\_k* relationship table has the same ability as well. According to adopting this method, the relationship between different entities is flexible. We do not need to consider the running performance and the influence to other storage tables when we needed to add or delete an entity.

### 2.3. Back-end Server

Our project has a complex back-end server because it needs to transfer data between different platforms and applications. The programming language of the recommender system is Python. Therefore, we adopted the Django web framework to build. It made our project rapid development, modularization, and easy to maintain. Our project contains several important modules, such as the chatbot, the user's management, the recommender system, the web crawler, etc. Each module is functional independence but data sharing. In addition, all the modules need to be deployed in a common system. If we consider the data transferring between different modules, we will roughly divide all modules into three files named the view file, the model file, and the controller file. Though this way can avoid redundancy of universal functions, it will make the program coding is read not easily and understood. By contrast, Django provides a built-in tool to help developers produce the initial project, and every module will maintain its own Model-View-Controller (MVC) architecture (Figure 7). One module can obtain data from other modules and functions by importing their public models. Figure 7 shows a simple structure of our project (only contains a library module, user module, and several necessary support files). The library module responds to manage all operations and data of the articles dataset. Its models.py python file has two parts the article data structure, and operations. Its urls.py file only contains the route addresses. Its views.py file is the controller layer of the MVC architecture. When the system takes users to visit an article, the system will execute the corresponding function of views.py by the route address from urls.py. The views.py can read the user information by importing the model from the user module. The controller layer of the library utilizes the user information to query feedback of the user and then produces the user preference reading list to create the front-end webpage. Though the user only opens an article detail webpage. But our system asked different modules co-operates for composing the result.

Figure 7. A simple Django file structure

```
server/  
manage.py  
=====  
basic/  
settings.py  
urls.py  
=====  
library/  
models.py  
urls.py  
views.py  
static/  
index.css  
index.js  
templates/  
index.html  
=====  
user/  
models.py  
urls.py  
views.py  
static/  
user.css  
user.js  
templates/  
user.html
```

According to the project management mechanism, we can conveniently separate the entire system into several modules, each module only focuses on its requirement without developing the extended functions to query data from other module tables. All modules are maintained by the settings.py file below the catalog called basic. If you want to temporarily add or delete a module, you can delete the module record of settings.py without permanently deleting the whole module files. In addition, if you want to change a logic coding of a function in a module. You do not need to consider compatibility if you did not change the function name and result. To other modules, it is just a black box.

Another important feature of Django is the database-driven and template engine. Compared with the traditional web server framework, Django packaged all database operations to a set of standard control methods. Our project does not need to control the MySQL database directly, the models.py file of each module contains several classes. Each class represents a type of entity. The parameters are the attributions of the entity.



Figure 8. The class of the user entity

```
class User(models.Model):
    uid = models.AutoField(primary_key=True)
    username = models.CharField(max_length=64)
    password = models.CharField(max_length=32, default="")
    token = models.CharField(max_length=256, default="")
    expire_date = models.DateTimeField('Expire Date')

    symptoms = models.ManyToManyField(Symptom, through='Symptom_u')
    knowledges = models.ManyToManyField(Knowledge, through="Knowledge_u")

    def __str__(self):
        return self.username
```

Figure 8 shows the user entity on our project. The left-hand side of the equals sign is the attribution name. On the right-hand side are the requirements of the corresponding attribution. Such as the *CharField* indicates this attribution is a single line of string. The max character length is 64. In addition, the class does not only contain attributions, but it also includes the relationship with other classes. Because each user includes several symptoms and feedback from articles. The class of users contains the many-to-many relationships with the symptoms table and the knowledges table respectively. After you completed developing all the classes. The Django will scan all the model files of the project to generate the SQL commands for table creation. Then the SQL commands are executed to create the tables by the MySQL database system. If you change some attributions of classes, the Django framework will recreate the commands to update the corresponding tables.

Because the Django framework grasped all data structures of classes. Therefore, it will automatically create the Create, Read, Update, Delete (CRUD) operations by the corresponding notations of the attributions. For instance, if you want to set the current date to the date attribution of a class. You can directly put the data object to this attribution. The framework will transfer the object to the SQL command automatically. On the contrary, if you read an entity from the database. Its attributions will be converted to relative Python objects by the framework. Our project adopted this scheme to easily convert different entities to classes without considering the corresponding database commands. Such as the paginator function, we

want to separate the article list into several pages, and each page only contains ten articles. The web framework can automatically convert the list object to a paginator object. It contains several parameters to indicate the current page number, the number of total pages, and the article list of the current page (Figure 9).

Figure 9. The paginator object

```
def knowledgeList(request, context):
    knowledge_list = Knowledge.objects.values('kid', 'title', 'author', 'source'
, 'expire_date', 'type', 'way',
                                           'status').order_by('-kid')
    paginator = Paginator(knowledge_list, 10)
    page_number = request.GET.get('page')
    page_obj = paginator.get_page(page_number)
    context['list'] = page_obj
    return context
```

The template engine is another core feature of the Django framework. As the front-end development work, the most task is to repetitively build a lot of web pages that only have tiny differences. If we decide to change the web major colors or the navigation bar. These contents are the same on all web pages, but they still need to update one by one. The template engine was created to let the web pages be flexible. It makes the webpage grasp the ability of reuse. The front-end developers create the template webpage. It only contains the page layout without any content. The templates are put into the static files of the web framework. The back-end developers insert the logic coding and the data object attribution names into the template (Figure 10).

Figure 10. An example of template engine

```
<div class="box">
  {% if user_token and user_token != "None" %}
  {% if user_preference %}
  <ul class="menu-list">
  {% for item in user_preference %}
  <li>
  <a href="{% url "rmsystem:interBrowser" %}?kids={{ item.kid }}&current
_page={{ item.kid }}"target="_blank">{{ item.title }}</a>
  </li>
  {% endfor %}
  </ul>
  {% else %}
  You can click 🍌like or 🍌dislike...
  {% endif %}
  {% else %}
  You can type your account...
  {% endif %}
</div>
```

When a webpage is requested. The template engine will first scan all logic coding and attribution names of the template. Then it substitutes the attributions to the real data and executes the logic coding to confirm the final layout. According to these processes, the webpage relays on the newest data to change itself without any manual operation.

Figure 11. A partial coding of the Regular expression

```
linkRequest = requests.get(link)
articleLink = re.search('<iframe src=".*?.html', linkRequest.text).group().replace("<iframe src=\"", "")
articleRequest = requests.get(articleLink)

image = re.search('<td class="mcnImageCardBottomImageContent" [\s\S]*?</td>', articleRequest.text).group()
image = re.search('src=".*?jpg', image).group().replace("src=\"", "")
```

Because our project needs to collect the up-to-date articles from the Internet. Therefore, we developed a web crawler to regularly collect information from the sources. We analyzed their hypertext markup language (HTML) files. And extract the core information (Such as authors, images, date, etc.) by the regular expression. A regular expression is a query notation language. It has a set of rules to indicate how to match the specific sub-content from a long string. We can extract information in this way if we found the specific characters around the information from the HTML files. Our web crawler contains a set of regular expressions to automatically scan the target sources and extract the information to store (Figure 11).

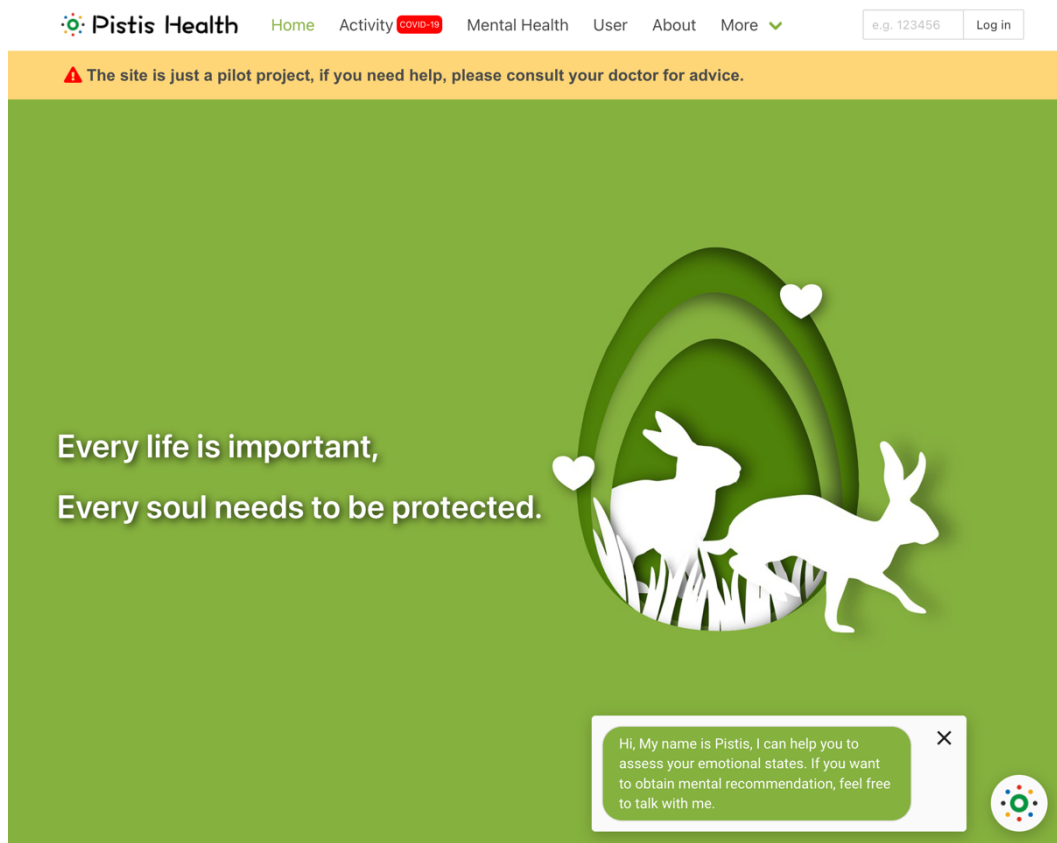
### 3. Demonstration

This section demonstrated the entire interaction of our project. It has three parts, dialogue, users end, and researcher end respectively. We will show the interfaces and the execution logic behind them.

#### 3.1 Dialogue interaction

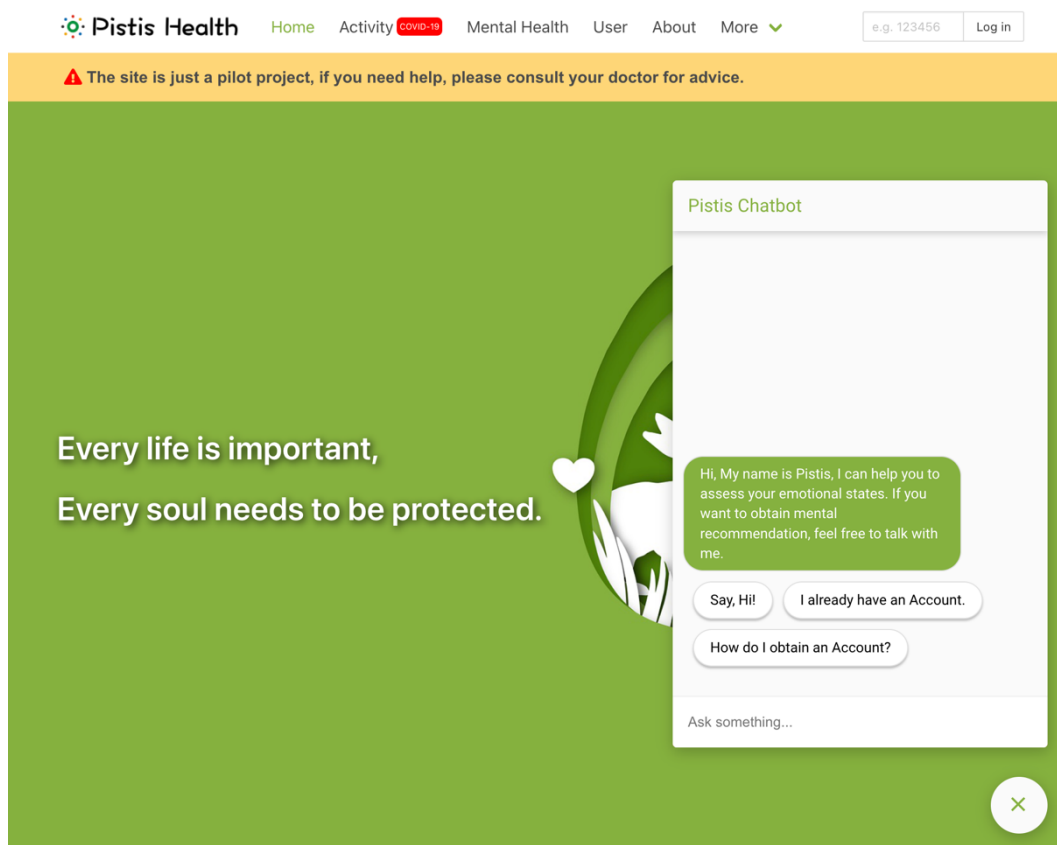
When the users visit our website, they will find a small button has been put in the right-bottom corner. This is the entry of the chatbot. To ensure users can be attracted to talk with it, after the end loading of the home page, it will pop a chat bubble (Figure 12). This chat bubble will not automatically be activated when the device has a small size of the screen because it will overlap too much space.

Figure 12. The home page



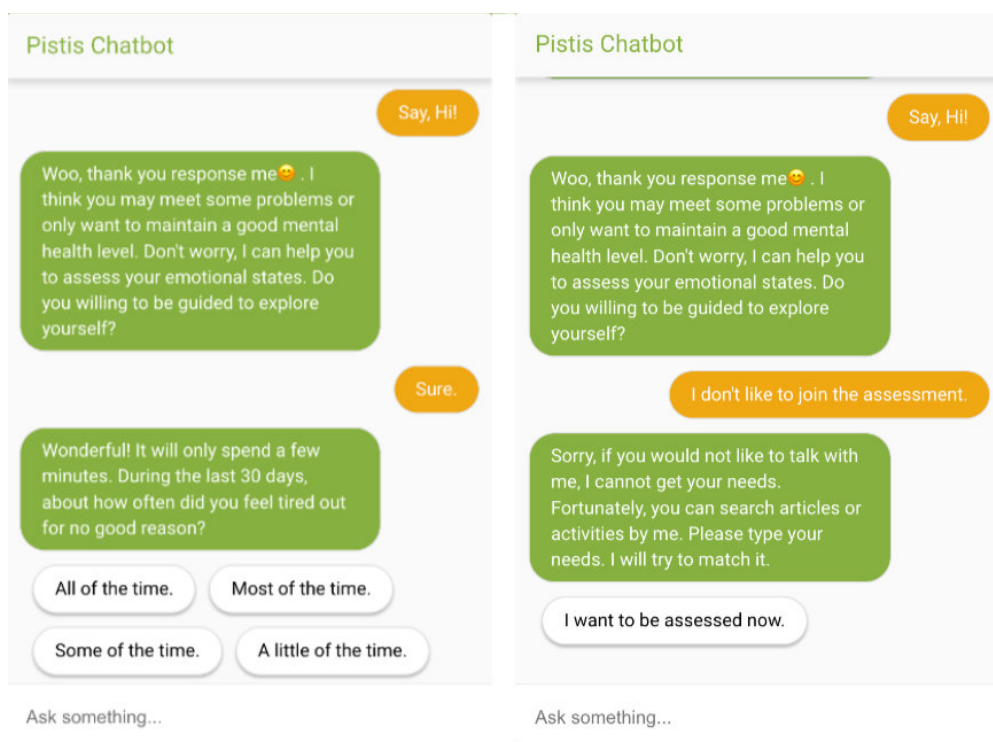
If users click the chat bubble, it will unfold into a full functions chat box that contains several interaction buttons and an input box (Figure 13). Because users need to follow our pre-designed dialogue flows to converse with the chatbot. Therefore, each chat bubble of the chatbot will follow several response buttons. Each button is a hint message. Users can click a button to send the message directly, or they can type a message with a similar meaning as a hit message, the natural language process engine will try to match the most possible dialogue flow to the response. If it cannot match any suitable intent, the chatbot will tell users it cannot understand, then it will guide users to provide a correct response.

Figure 13. The full functions of chat box



If users are willing to talk with the chatbot, they can respond to the chatbot greeting. The chatbot will tell them it wants to start an assessment of users for qualifying their emotional states. Users can type YES to start the assessment conversation. Otherwise, they can type NO if they don't like to join the assessment. The chatbot will convert to the search engine. Users can type their needs to search the whole articles database (Figure 14). They can ask the robot to start evaluating at any time as well.

Figure 14. Different responses of greeting



While users are joining an assessment conversation, the chatbot activates the K10 scale dialogue flow. It contains 10 pre-defined questions and over 30 pre-defined responses that correspond to different answers. Table 4 demonstrated the entire dialogue flow of the assessment conversation. The third, sixth, and eighth questions are detailed questions respectively related to the previous one. If users type never as the answer to the second, fifth, or seventh question. The chatbot will not show the related detailed question.

Table 4. The responses of the K10 scale

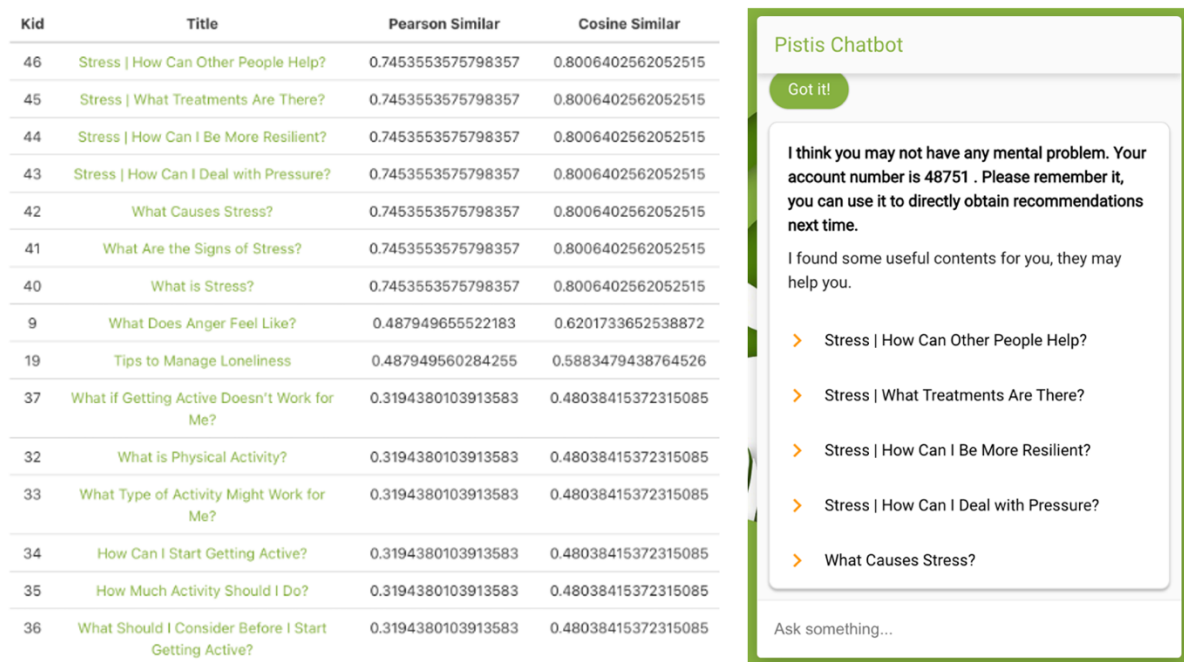
Index	Dimensions	Problems	Answer index	Answers	Response
1	Tired out	Wonderful! It will only spend a few minutes. During the last 30 days, about how often did you feel tired out for no good reason?	5	All of the time	It sounds not good, but don't worry. Please follow through with the test.
			4	Most of the time	I know that will affect you to do a complicated thing.
			3	Some of the time	It does not serious, you can browse some advice after completing the assessment.
			2	A little of the time	Got it!
			1	None of the time	
2	Nervous	Thank you, :D. During the last 30 days, about how often did you feel nervous?	5	All of the time	I know it made you cannot trust yourself. Some mental suggestions can help you to overcome it.
			4	Most of the time	The frequency may a little bit high, you can read some articles to adjust yourself.
			3	Some of the time	Don't worry, everyone always meets some situation to feel nervous.
			2	A little of the time	Got it!
			1	None of the time	
3	So nervous	During the last 30 days, about how often did you feel so nervous that nothing could calm you down?	5	All of the time	I know it is very hard for you how to make yourself calm down. I can give you some articles that contain different ways to try.
			4	Most of the time	Yeah, it's hard to relax sometimes. You can overcome it by mental methods.
			3	Some of the time	Deep breathing may be a good idea. Some people think mindfulness helps them to calm down. I will give you some relative content.
			2	A little of the time	Got it!
			1	None of the time	
4	Hopeless	During the last 30 days, about how often did you feel hopeless?	5	All of the time	It is difficult to overcome. But I will accompany you to find some advice.
			4	Most of the time	I know your feeling. I'll be there for you through this.
			3	Some of the time	Don't worry too much, we can try different new things to build hope.
			2	A little of the time	Got it!
			1	None of the time	
5	Restless	During the last 30 days, about how often did you feel restless or fidgety?	5	All of the time	Insecurity makes you feel bad, I will give you some activities information. You can meet new friends to accompany you.
			4	Most of the time	I will find some relative suggestions for you for meeting the situation.
			3	Some of the time	I have some knowledge about it if you are willing to reading some articles about this situation.
			2	A little of the time	Got it!
			1	None of the time	

Index	Dimensions	Problems	Answer index	Answers	Response
6	So restless	During the last 30 days, about how often did you feel so restless you could not sit still?	5	All of the time	I know you are hard to relax. After completing the assessment, let me help you.
			4	Most of the time	I know the feeling when I meet some important things. Don't be afraid to trust yourself.
			3	Some of the time	Everyone cannot calm down sometimes. Don't worry about your situation.
			2	A little of the time	Got it!
			1	None of the time	
7	Depressed	During the last 30 days, about how often did you feel depressed?	5	All of the time	It sounds not good, but don't give up. I grasp a lot of knowledge to help you overcome a difficult time.
			4	Most of the time	I know you may feel not good when you are experiencing it. Let me know how to help you.
			3	Some of the time	The frequency is not high, you can try some mental health ways to solve it.
			2	A little of the time	Got it!
			1	None of the time	
8	So depressed	During the last 30 days, about how often did you feel so depressed that nothing could cheer you up?	5	All of the time	I am sorry to hear it, I hope I can help you to overcome the depressed emotion.
			4	Most of the time	I know you are experiencing some bad things. Don't worry, I can help you.
			3	Some of the time	You have to trust yourself, you can try some new activities to distract yourself.
			2	A little of the time	Got it!
			1	None of the time	
9	Worthless	There are two last questions! During the last 30 days, about how often did you feel worthless?	5	All of the time	Don't think that, Every life is important, Every soul needs to be protect. I can help you!
			4	Most of the time	You are very important to the people closest to you. I will introduce some articles to help you refind yourself.
			3	Some of the time	Follow my advice, I'll help you fight that emotion.
			2	A little of the time	Got it!
			1	None of the time	
10	Everything an effort	During the last 30 days, about how often did you feel that everything was an effort?	5	All of the time	It may influence your life, you need to spend more energy to complete things. But I can help you to know how to solve it.
			4	Most of the time	You can't recharge yourself have enough energy when you feel tired. Some nature activities can help you to rest your soul. Don't refuse to try.
			3	Some of the time	It is not a serious situation, Adjust your mood and do something relaxing.
			2	A little of the time	Got it!
			1	None of the time	



After the ending of once assessment, the scores will be sent to the back-end server for analyzing the user model. If users only feel nervous and cannot calm down, the articles which talk about how to decrease stress will be matched to produce the recommendation. The reason for the recommendation is all articles are tagged by their content. The articles which talk about stress are mostly focused on the nervous dimension in our database. Therefore, they will get a high cosine similar score (Figure 15). On the right side of figure 15 is the chatbot recommendation result. It contains two parts. The first part is the severity of non-specific psychological distress calculated by the K10 scale. The second part is the article list which is the top five cosine similarity articles corresponding to the user model.

Figure 15. An example of assessment result



### 3.2 Users end interaction

After users completed the assessment, they can start browsing the recommendation by the inter-browser (Figure 16). Because all articles are collected on the Internet. Therefore, the original webpage is loaded to the user's browser directly to respect the original website copyright. On

the left toolbar, the list has been divided into several parts. The first part is produced by the IR similarity measure. If users visit an article directly without the chatbot. The first part will only contain the title of the current article. After users have read the article yet, they can click the top buttons for providing feedback either like or dislike. This behavior is recorded to create the content-based recommender model. The TF-IDF algorithm has been used to establish the content matrix. So, when the system tries to find the user preference, it averages the TF-IDF vectors of the user's liked articles and then will compare them with the rest to look up the top-n associated content. The second list is established in this way. If they did not provide any feedback, it encourages users to read articles and click the buttons. The last part is the content related to the current article. They are also produced by the TF-IDF algorithm but only related on the current content.

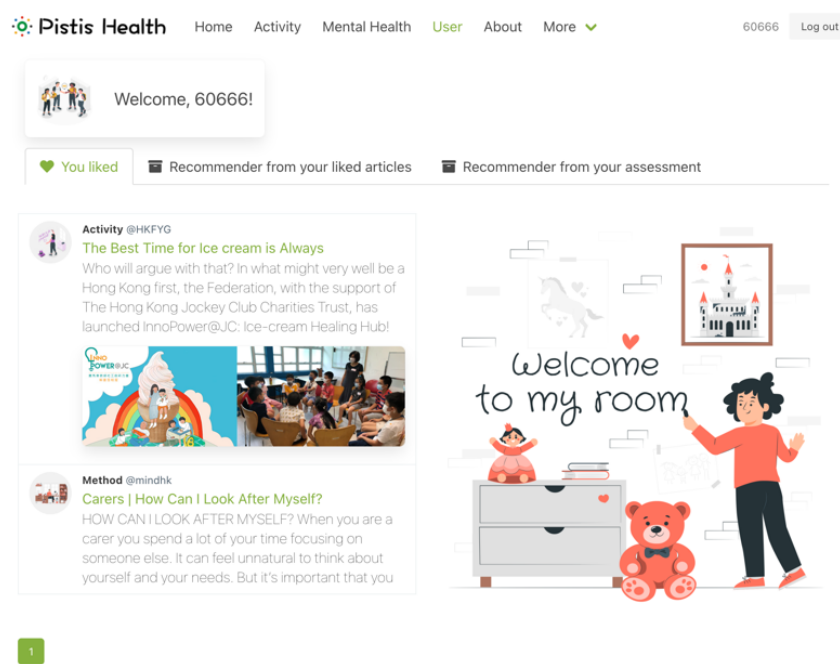
Figure 16. The inter-browser

The screenshot displays the Pistis Health website interface. At the top, the navigation bar includes 'Home', 'Activity', 'Mental Health', 'User', 'About', and 'More'. The user is logged in as '30381' and can click 'Log out'. The main content area features a sidebar with 'CONTENTS' and 'FOR YOUR PREFERENCE' sections. The 'CONTENTS' section lists various articles, with 'What is Depression?' highlighted. The 'FOR YOUR PREFERENCE' section includes a message: 'You can click like or dislike on the bottom to get your own recommendations.' Below this, the 'RELATED TO CONTENTS YOU'VE VIEWED' section lists related articles such as 'CBT | What Types of Problems Can CBT Help With?' and 'Bipolar Disorder | What Treatments Can Help?'. The main article, 'What is Depression?' by mindhk, has 'Dislike' and 'Like' buttons at the top. The article content includes a 'Table of content' section with links to 'What is depression?', 'What are the symptoms of depression?', 'What causes depression?', and 'What treatments are there for depression?'. The main text of the article defines depression as a low mood that lasts for a long time and affects everyday life, and notes that in its mildest form, it can mean just being in low spirits, while in its most severe form, it can be life-threatening.

The reason for developing the inter browser is we need to collect the users' feedback to improve the manifestation of the recommender system. As we mentioned before, the classification of each article was tagged by the researchers. Therefore, it could be biased by researchers' oversight. We hope the dimensional scores of each article can be dynamically changed by users' behaviors. When a user clicks the like/dislike button, the server will search for all users who have the same taste. Then it averages each dimensional score of the K10 questionnaire to update the dimensions of the article. If an article has both the highest and lowest scores, its final dimension score will tend to be in the middle.

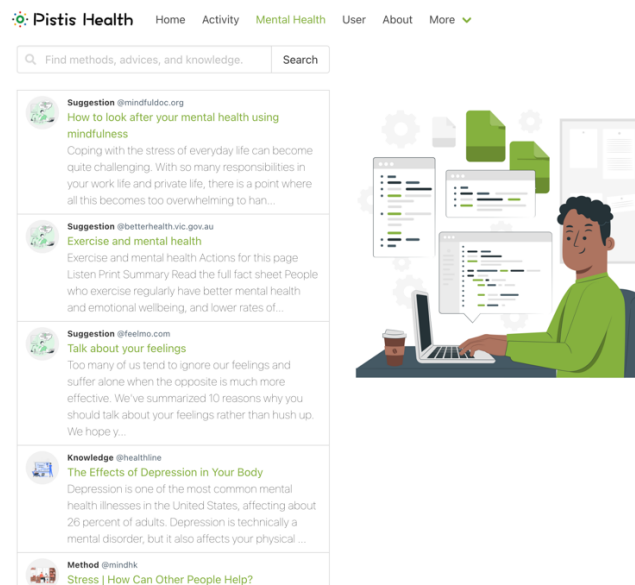
Users not only can get the recommendation from the chatbot. If they have an account, they also can visit the user page to browse directly (Figure 17). This page contains three subpages, the user liked articles, the TF-IDF recommendation by the liked articles, and the IR similarity measure by the dimensional scores. If users do not need to reassess their mental situation, they can browse this webpage regularly because the lists will automatically update when a new article satisfies the user's dimensional scores.

Figure 17. The user page



The activity page and the mental health include the up-to-date activities in Hong Kong and all the articles respectively (Figure 18). Both pages have a search engine to help users search for their needs.

Figure 18. The mental health page



### 3.3 Researchers end interaction

To conveniently manage and analyze the articles, symptoms, and users. Our project developed the management system. The knowledge page shows the articles list that contains the id, source, expiration date, type, collection way, and status (Figure 19). You can click the title of an article to browse the detail. The export button provides the function to extract the whole dataset to a CSV file. In addition, the modification page can change all attributions of the articles. It lists all dimensional scores of the articles, you can reset it to ensure the recommender precision (Figure 20).

Figure 19. The knowledge research page

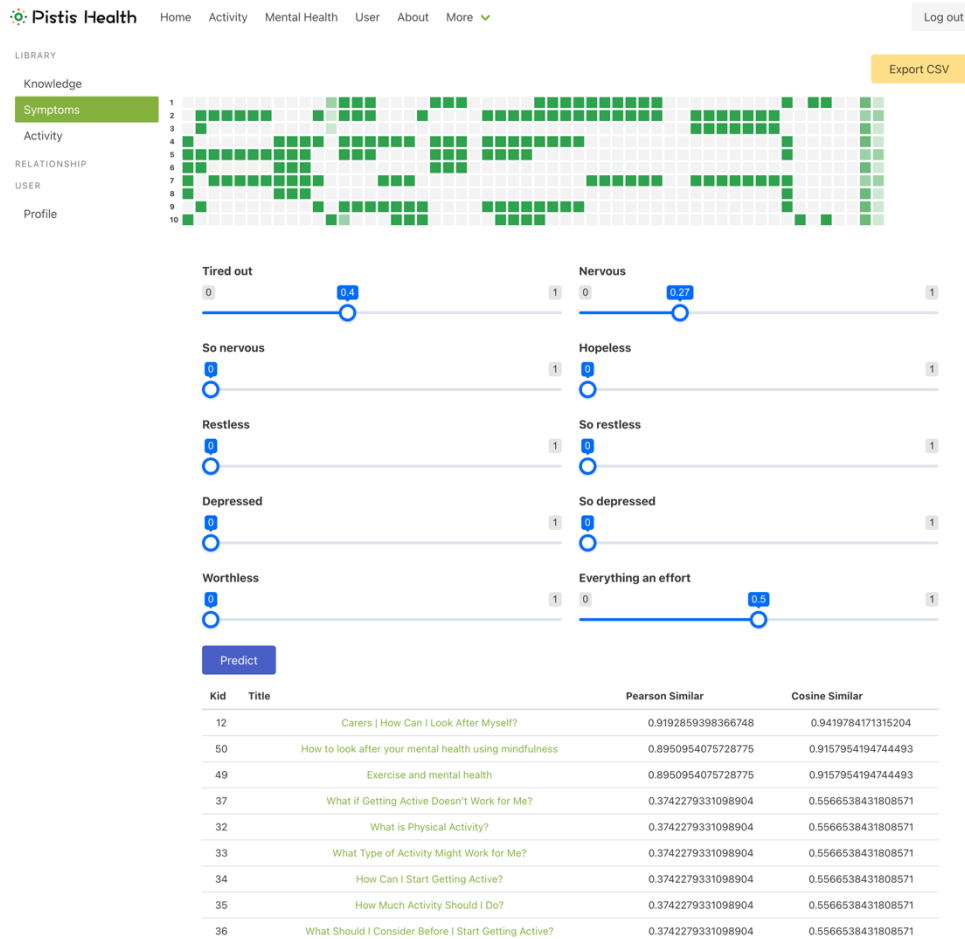
id	title	Author	Source	Expire Date	Type	Way	Status
68	Building Hope and Harmony	HKFYG	HKFYG	April 18, 2023, 8:37 p.m.	ACT	AUT	DIS
67	Growing Up Drug Free	HKFYG	HKFYG	April 10, 2023, 4:50 p.m.	ACT	AUT	DIS
66	A Fun-Filled Extraordinary Summer with HKFYG Online Courses	HKFYG	HKFYG	March 18, 2023, 3:18 p.m.	ACT	AUT	DIS
65	Caring to Call - Calling to Care	HKFYG	HKFYG	March 13, 2023, 1:12 p.m.	ACT	AUT	DIS
64	Building a Sustainable Future	HKFYG	HKFYG	March 13, 2023, 1:12 p.m.	ACT	AUT	DIS
63	Building a Borderless Neighbourhood!	HKFYG	HKFYG	March 13, 2023, 1:12 p.m.	ACT	AUT	DIS
62	Daring to Dream...	HKFYG	HKFYG	March 13, 2023, 1:12 p.m.	ACT	AUT	DIS
61	The Age of the Slasher!	HKFYG	HKFYG	March 13, 2023, 1:12 p.m.	ACT	AUT	DIS
60	Have a Merry Tai O Christmas!	HKFYG	HKFYG	March 13, 2023, 1:12 p.m.	ACT	AUT	DIS
59	Plastic-Free Future	HKFYG	HKFYG	March 13, 2023, 1:12 p.m.	ACT	AUT	DIS

Figure 20. The article modification page

Another important page of the management system is the symptoms analysis page. It demonstrates the graph map of the whole dimensional scores. You can visually view the dimensional distribution of all articles. Below of the graph map. You can utilize the slide button to adjust the value of a single dimension for simulating a real user model. When you click the predict button, the system will give you the cosine similar corresponding to the current user

model. Researchers can use it to explain why the system produces the specific article list to the user.

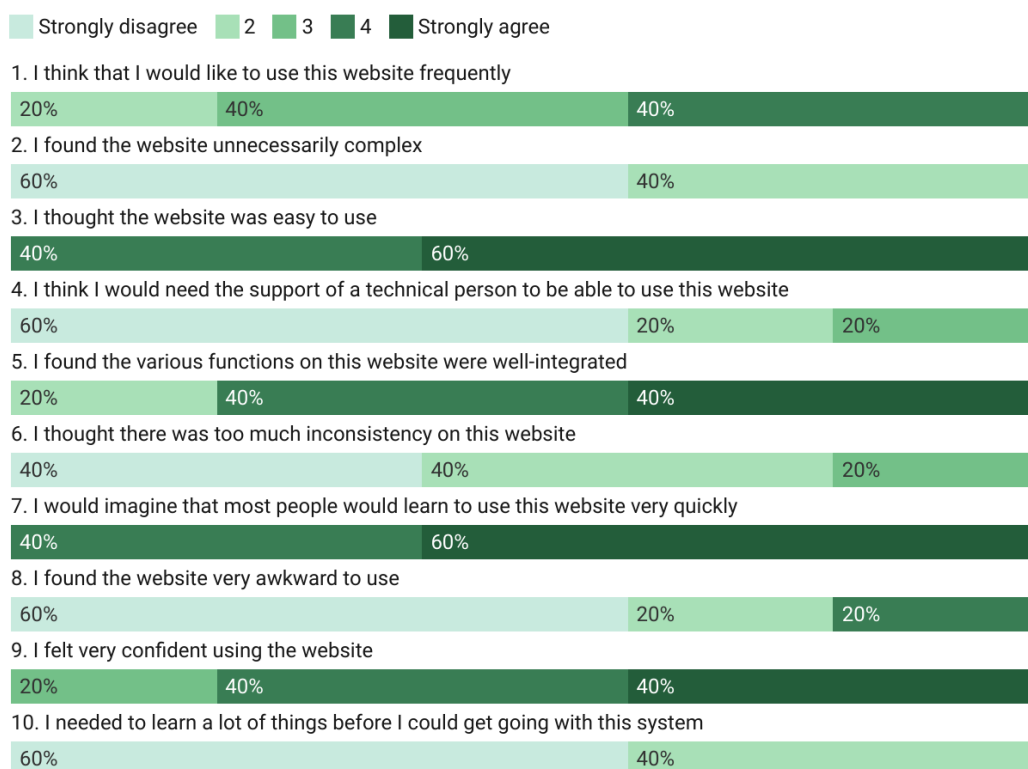
Figure 21. The symptoms analysis page



## 4. Investigation

After we completed the major development of the project. We adopt the System Usability Scale (SUS) to collect feedback from users about the interfaces. This scale is proposed by Brooke in 1996. It was designed to conveniently assess the usability of a given product or service [5]. This scale contains ten questions about the users' experience on a 5-point scale. And the statements alternate between the positive and negative. Bangor and his team declared that the SUS has proven itself a valuable and robust tool in helping assess the quality of the user interfaces by a set of evaluations [6].

Figure 22. The System Usability Scale Result



In our project, we slightly adjusted the original SUS statements to suit our project needs. And invited five participants to experience our website. They have different educational backgrounds, but none of them works in software development. They first talked with the chatbot, got recommendations, and browsed the articles. Then they completed the SUS scale

and provide their own comments as an addition. Figure 22 shows the scoring distributions. The singular questions are positive, and the rest are negative. Through the figure, we can see that most feedback is positive, they thought that the website can be easily used to access and obtain the recommendations.



## 5. Conclusion and Future work

At the beginning of our project, we aimed to design and develop a mental dialogue system to talk with users to obtain their mental situation to provide the corresponding psychological treatments for solving their problems. But after we talked with several psychological consultants, we thought this scheme is infeasible. We cannot only rely on several short dialogues to diagnose psychological diseases and their causes. Therefore, we adjust our goal to assess nonspecific psychological distress. Because in this field, researchers declared that the several special physical and psychological symptoms have differences between psychiatric patients and nonpatients [1]. According to this theory, Kessler and his team produce the Kessler Psychological Distress Scale (K10) to measure psychological distress. This scale is very short and only has 10 questions on 7 dimensions. We adopted this scale to establish the user model. Each dimension of the user model represents a specific symptom. It made the chatbot grasp the ability to assess users' psychological distress. The symptoms do not only help users to know the severity of the nonspecific psychological diseases, but it also helps us to establish the Information retrieval (IR) similarity measure method to articles. Users will get their fundamental recommendations through this process. To collect the user's preferences, we developed the inter-browser. It provides a good reading environment to users as well as a set of buttons to record feedback. Users' preferences are used to supply content-based recommendations. It helps users to focus on the specific topic to avoid the confusion of several topics having the same dimension. Through our system, users can talk with the chatbot to create their own user model about nonspecific psychological distress and obtain mental health advice. Then they can utilize the inter-browser to read articles and provide feedback. The feedback will affect the dimensions of articles as well as the preference record of users. Users can read an endless stream of their favorite topics. If their mental situation changed, they could reassess by

the chatbot to adjust their user model and get up-to-date advice. This process is the major functions of our mental health dialogue system.

In future, we hope the database can store various types of recommendations, such as videos and audio. We want to extend the volume of our dataset to provide a rich variety of suggestions. Meanwhile, our chatbot still needs to follow the specific dialogue flow to respond to user requests. We want to build a machine learning model to process and produce responses without any pre-defined statement and flow. Moreover, we hope the website can contain a user community, we can use the user-based recommender system to let users help each other.

## Reference

- [1] B. P. Dohrenwend, P. E. Shrout, G. Egri, and F. S. Mendelsohn, “Nonspecific Psychological Distress and Other Dimensions of Psychopathology Measures for Use in the General Population,” *Arch Gen Psychiatry*, vol. 37, no. 11, pp. 1229–1236, 1980.
- [2] R. C. Kessler *et al.*, “Short screening scales to monitor population prevalences and trends in non-specific psychological distress,” *Psychological Medicine*, vol. 32, pp. 959–976, 2002, doi: 10.1017/S0033291702006074.
- [3] J. Ramos, “Using TF-IDF to Determine Word Relevance in Document Queries,” in *Proceedings of the first instructional conference on machine learning*, 2003, vol. 242, no. 1, pp. 29–48.
- [4] Michael W. Berry, Susan T. Dumais, and Gavin W. O’Brien, “Using Linear Algebra for Intelligent Information Retrieval,” *SIAM Review*, vol. 37, No. 4, pp. 573–595, 1995.
- [5] P. W. Jordan, B. Thomas, I. L. McClelland, and B. Weerdmeester, *Usability evaluation in industry*. CRC Press, 1996.
- [6] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574–594, Aug. 2008, doi: 10.1080/10447310802205776.