Practice Module for
Graduate Certificate in
Intelligent Reasoning Systems (IRS)

Find Your Dog

The one-stop app to adopt a dog

Project Report

Team Members

Jonathan Lim Ching Loong	A0261707E
Hu Hang	A0261634H
Fu Jia	A0261661H
Du Qiao	A0261990X

Contents

Executive Summary

Dogs are the most popular pet type among pet owners in Singapore. Currently there are two ways of owning a dog: through Commercial Purchase and through Adoption. While Commercial Purchase is relatively straightforward, the process of adopting a dog is rather tedious.

Find Your Dog is an intelligent reasoning system that aims to reduce the time taken for an aspiring adopter to search for their desired dog. The system uses a recommender system that receives detailed information about a user from a user interface and recommends the dogs that best match the user's preference according to a knowledge base that's built from data obtained from actual shelter organizations.

The system focuses on using Content Based Recommender, which is mainly based on the similarity of the features of the items and recommend items that are similar to what the user has selected. From there, if the user is satisfied with the recommendation, then the user is led to the organisation's platform to proceed to the next step of the organisation.

Background and Objective

Pet ownership is quite common in Singapore. Statistics show that 33% of Singaporeans are pet owners as of January 2022 and 17% used to own one but not anymore. Needless to say, dogs were the most popular pet type among pet owners, with an astounding 58% respondents reported owning dogs.

Currently, there are two ways of owning a dog: through Commercial Purchase and through Adoption. While the commercial purchase route is quite straightforward, there had been some comments that the adoption process can be quite tedious as one would have to go through many procedures to successfully adopt a dog. Among the procedures would include browsing through different websites to find a favourite dog, filling up a long form, contacting the organization, interviews, etc. A rough flow of the procedure can be seen in the diagram below:

Figure 1 Flow of the Procedure of Adoption

It didn't help that even on the websites, most of the profiles required manual clicking into dog profiles to read and consider yourself. Research was largely done manually by the potential adopter and many end up purchasing dogs instead as the process was too time-consuming. Furthermore, it is difficult to ascertain what criteria different shelters consider when it comes to approving adoptions.

As such, this project objective is to be a platform to facilitate the search process for dog adoption in Singapore. The project targets step 1 in the process and aims to reduce the time taken for an aspiring adopter to search for their desired dog by applying intelligent reasoning systems. It also hopes to take into consideration the key approval criteria of each organization to better increase the chances of an aspiring adopter to successfully adopt a dog.

High Level Diagram of Recommender System

Our recommender system contains 5 basic components: Knowledge Base, Recommender, Frontend Interface, Backend Server, and Database. Each of these components are pictured in the diagram below. The diagram shows a typical data flow when recommender system is trying to generate a result for a potential adopter. We will consider each one in detail.

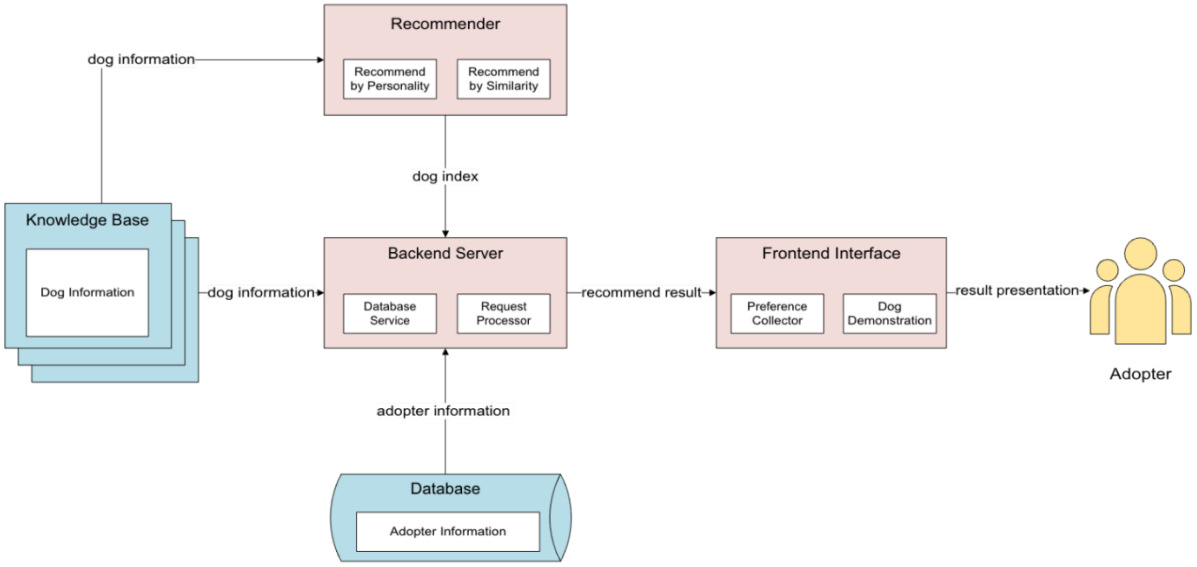


Figure 2 Recommender System Architecture

- Knowledge Base: this component provides information about dogs.
- Recommender: this component processes natural language, calculates dogs' similarity and generates recommendations.
- Frontend Interface: this component is created to provide a graphical user interface to the adopter.
- Backend Server: this component integrates each component together. It directs the flow of data in the system.
- Database: this component stores information about adopters.

Knowledge base

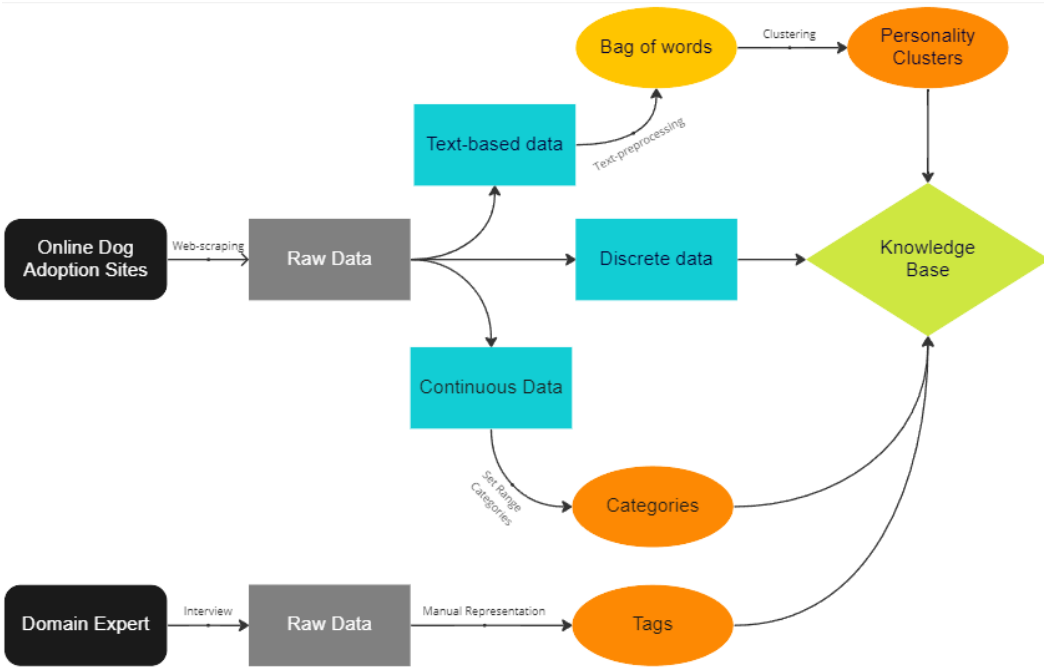


Figure 3 Knowledge Base

Dog profiles were obtained from 17 dog shelter organisations via web-scraping. Information extracted from the sites included:

1) Pictures of the dog
2) Link to webpage
3) Name of dog
4) Estimated DOB or Age
5) HDB Approval status
6) Personality Description and Background

Based on the type of raw data obtained, different pre-processing methods were applied to generate useful data for the knowledge base. As per diagram above, the raw data can be split into text-based data such as the personality description, discrete data such as gender and HDB approval, and continuous data such as age. The pre-processing methods will be described below:

1) Text-based data (Personality Description)

a. <u>Pre-processing</u>

Natural Language Processing (NLP) methods were applied to extract bags of words for each dog profile. A text pre-processor was built using Spacy to remove stop words, punctuation, symbols, etc. Certain irrelevant words like months, dog names, Singapore town names, etc were also excluded in the bag of words. From this bag of words, the term frequency-inverse document frequency (TFIDF) was calculated and transformed into an array to prepare for clustering

b. Clustering

Based on the TFIDF array obtained, a Hopkins test was performed to identify the cluster tendency of our data. As the score indicates good clusterability, two methods were compared and used to select a good number of clusters for the data. Method 1 was via silhouette score, which showed that 3 and 8 were good number of clusters. Method 2 was via the elbow method which showed that 4 clusters was good. From there, the KMeans method was applied to obtain clusters for k=3, k=4, and k=8.

c. Cluster Analysis

The clusters for k=3, k=4, and k=8 were analysed and labels were assigned. These labels were eventually used as tags for the dogs in the knowledge base to provide additional input for the recommender.
As k=8 clusters didn't display any distinct features between clusters and didn't provide any useful labels, only k=3 and k=4 clusters were used in the end.
The tags obtained from the two cluster sets were as follow:
K=3 Cluster:
Sociable
Socially Intermediate
Socially Challenged

K=4 Cluster:
Medical Problems, Fearful, Aggression, Insecure
Anxious, Quiet, Shy, Low Energy
Takes time to warm up, learning, comfortable
Strong, energetic, independent, and active

2) Continuous Data (Age)

a. Pre-processing

The age information obtained from sites come in multiple formats. In order to generate categories, we standardized the age format to follow the "Years" format (e.g. 1.42 years).

Information in "Estimated DOB" format were converted to "Years" by subtracting this year and month (Sep 2022) from the DOB.

Information in "yr, mth" format is converted to "Years" by converting the months to decimals and added the years.

Incomplete DOBs such as "Mar-2021" or "2021" will be assigned "01-Mar-2021" and "01-Jan-2021" respectively and processed as per "Estimated DOB".

b. Categorisation

After some online research, we decide to follow the classification as follow:
Puppy: 0.75 years and below
Teen/Young Adult: between 0.75 to 4 years
Adult/Senior: 4 years and above

3) Discrete Data (Gender, HDB Approval, etc)

These data are directly input to the knowledge base.

4) Interview Data

a. Interview Input

A domain expert who had experience with multiple dog shelters was interviewed to better understand the thought process of shelters when processing adoption forms and what they consider when approving potential adopters to proceed with the next step.

According to the expert, generally shelters consider information such as working hours, household members (e.g. living alone, with kids, with elderly, etc), whether the household have other pets or are considering getting other pets, the potential adopter's experience with dogs, the hours the adopter spends at home over the week, the type of housing, future delegation of care for the pet (e.g. will the adopter be the main caregiver or other household members), reason for adopting, and in some cases, nationality of the adopter too. The expert confirmed that recently the topic on having window grilles at the adopter's house is trending as there had been many reports of dogs trying to escape through windows due to anxiety or fear.

The expert also suggested we include questions to ask whether the potential adopter are open to adopting older dogs, dogs with medical conditions, or had experience with fearful and aggressive dogs. It should also be noted that for puppies, shelters tend to be stricter as the future personality of the puppy largely depends on how the adopter raise them. As such, they tend to be less keen on allowing less experienced adopters adopt puppies.

b. Interview Outcome

Considering the input from the domain expert, tags were created based on a few of the suggested categories:

i. Interaction with kids
ii. Interaction with elderly
iii. Interaction with other pets
iv. Medical Condition

These tags were generated by analysing the personality description extracted from the online dog adoption sites.

Recommender

1) Introduction of the recommender

The recommender is an important part in the whole system. It receives detailed information about a user and recommends the dogs that match the user's preference most. The commonly used methods for recommendation are Content Based Recommender and User/Item Based Collaborative Filtering. Content Based Recommender is mainly based on the similarity of the features of the items and recommend items that are similar to what the user has selected. Collaborative Filtering is done using large number of users' ratings for items and find similar users. Then the recommender will recommend items that are liked by certain users to the similar users.

For our project, we don't have much user information such as the users' ratings or the popularity of each dog, which means it is not appropriate for us to use Collaborative Filtering. Therefore, we mainly consider using Content Based Recommender.

Our recommender consists of two different parts, which are recommender for new users and for the old ones. For the new users, we need to collect their preference for the dog to build a user's profile to recommend dogs most similar to the profile. For the old users, after they have selected some of the dogs that they like, we recommend dogs that are similar to their previous preferences.

The whole structure of our recommending system is shown below:

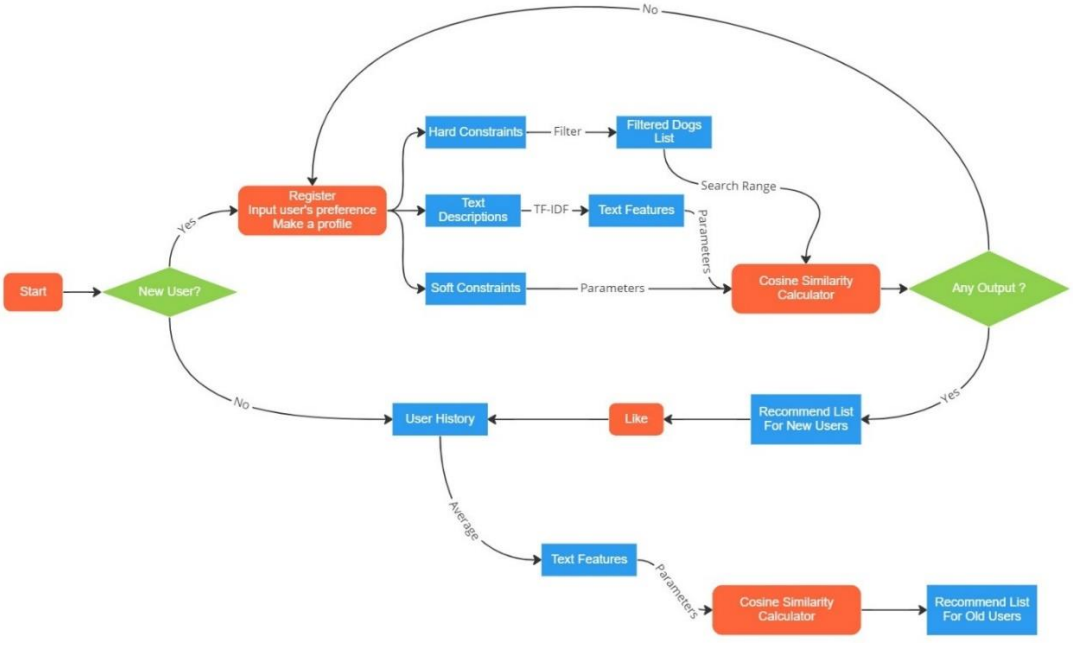


Figure 4 Recommending System

2) Features of the recommender

Based on the information we have collected from the dog shelters' website via web-scraping and the result of the interview with dog adopting expert, we selected the following features for our recommender.

a. Gender
b. Age
c. HDB approval
d. Personality description
e. Interaction with kids
f. Interaction with the elderly
g. Interaction with the other dogs
h. Medical requirement
i. Past ownership requirement
j. Window grills requirement

Depending on their sources and importance in adoption, the above features are evaluated in different degrees. Features a. to d. are obtained from the shelters' websites and are shown directly and explicitly on the websites. These features are very clear and correct. They can be seen as features more considered by the adopter for the dog. We consider these features as hard constraints. Features e. to j. are generated by analysing the personality description extracted from the online dog adoption sites. Therefore, some of the features are not absolutely correct and do not require to be matched completely. We consider these features to be soft constraints.

By dividing the features into hard and soft features, on one hand it shows different degrees of importance for each feature. The clearer and more accurate features are more important. On the other hand, since the number of dogs in our database is limited, we can avoid the situation of having no dogs that matches all the requirements for recommendation.

Text-based feature is based on the personality description of the dog. To calculate the similarity of the text-based feature, we need to do TF-IDF conversion to change the corpus into numerical matrix and use the values of each row of the matrix as text features for each dog. After doing TF-IDF conversion, we get a vector including 1770 values to represent the text feature of one dog. That also means our corpus has 1770 different words in total.

We saved all the features to an csv file and read the data into a pandas data frame to be our knowledge base for the recommender.

3) Calculation of similarity

We need to use our text features and soft constraints to calculate the similarity of dogs and users' profile or preference to give recommendation.

Since our values to be put into the similarity calculator are mostly text-based values, it is better to use cosine similarity to measure the similarity because it ignores the magnitude of the vectors. The formula of cosine similarity of two vectors is shown below:

$$\text{cosine-similarity (A,B)} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

4) Recommending for new users based on their profile

For the new users, when they register for our system, we collect information about their preference. Then we do NLP for the user's input description for his/her preferred dog. After

doing tokenization, stopwords removal and lemmatization, we get a list of pre-processed words. From these words we pick up the ones that appear in our existing corpus, which includes 1770 words, to form a user's text. Because words that don't exist in the in the current corpus won't make much influence on our recommendation, we throw them away. We then join the pre-processed user's text to the existing dogs' descriptions base and generate a new TF-IDF matrix and pick the corresponding row as the text features for the new users. For the existing dogs, there won't be much difference after adding one user text to the TF-IDF matrix. Therefore, we just leave it the same as before so that we don't need to form the whole data frame again. Next, we filter out the dogs that don't match the user's selected gender, age and HDB approval, leaving those that matches all these hard constraints. Among them, we calculate the cosine similarity for each dog between the user's profile.

For the 1770 TF-IDF values, we just simply put them into the formula to do the calculation. For the soft constraints, there are some differences. The numerical inputs of soft constraints are shown below:

Value for the dog (column 1)			Value for the user (column 2)		
Feature	Category	Value	User's profile	Category	Value
Interaction with kids	Good	0	Kid in house	Yes	0
	Bad	1		No	1
Interaction with elderly	Good	0	Elderly in house	Yes	0
	Bad	1		No	1
Interaction with other dogs	Good	0	Other dogs in house	Yes	0
	Bad	1		No	1
Medical Condition	No	0	Open to adopting a dog with medical condition	No	0
	Yes	1		Yes	1
Need ownership experience	No	0	Have ownership experience	No	0
	Yes	1		Yes	1
Need window grills at house	No	0	Have window grills at house	No	0
	Yes	1		Yes	1

Table 1 Numerical Inputs of Soft Constraints

For the value of the dog, if one feature does not matter the recommendation, the value will be 0. Otherwise, it will be 1. For the user, it will be the opposite. Take the window grills as an example. If we simply put this feature into calculation, those adopters with window grills will receive more recommendations on dogs that need window grills. But actually, the adopters with window grills can also adopt dogs that do not need such things at house. Therefore, we add the following rule to apply the soft constraints to the cosine similarity calculator. If the

value of one certain feature in column 1 (for the dog) is 0, or that in column 2 (for the user) is 1, we will ignore the feature in the calculation for cosine similarity by turning both the values into 0 so that it won't have effect in the formula. Because in this case, this feature does not matter the selection of the dog. The reason is either the dog does not need the requirement, or the adopter reaches the requirement already. The only situation that needs to be considered is when the value in column 1 is 1 and that in column 2 is 0, which means the dog needs certain requirement, but the adapter does not reach it. In this case, the cosine similarity calculator takes effect, and the similarity output will be smaller due to a mild increase in the denominator.

After calculating all the similarities, we pick up the five dogs that are the most similar to user's profile. If there is no dog that can match the user's preference, we will also ask the user to renew his/her preference.

5) Recommending for old users based on their previous selections

After we present the recommended dogs to the user. The user can click a link if he/she likes the dog. If a dog is preferred by the user, we will get the index of the dog. We will calculate the average of TF-IDF values of all the dogs liked by the user as the user's history. Then use the average vector to calculate the cosine similarity between the user's history and the other dogs that haven't been viewed. We will show 5 dogs most similar to user's history for the user to pick up. In the recommendation for old user, we only consider text description as our recommending factor. On one hand our total number of dogs is limited. On the other hand, we can also recommend some different kind of dogs in this way for the user to view more dogs.

Backend Server

The backend server needs to integrate each component in the system and direct the flow of data. It's responsible for storing and organizing the information received from frontend, invoking the recommender, and processing adopters and dogs' information in the knowledge base and database.

The backend server is implemented by Flask. It's a micro web framework written in Python. We chose it as our backend server framework because Flask is easy to use, and Recommender is implemented in Python as well.

The frontend interface and backend server use REST APIs to convey requests and responses. For example, API "recommend_by_history" represents one communication that backend received a request to query if the database contains the requested adopter. If so, use the recommend history to invoke recommending by dog index. Otherwise, use the uploaded preferences to invoke recommending by preferences. The details of this API are as follows.

POST /api/recommend/recommend_by_history HTTP/1.1
Host: 127.0.0.1:5002
Content-Type: application/json
Accept: application/json

Request Body:
{
"adopter_name": "Lisa",
"accomodation": "1", // If adopter lives in HDB
"prefer_age_group": "3", // Adopter prefer puppy, adult or elder dogs
"prefer_gender": "1", // Adopter prefer male or female dogs
"personality_preference": "Anxiety", // The natural language of preferences
"Elderly": "1", // If elderly persons at home
"experience": "1", // If adopter have kept dogs before
"kid": "0", // If kids at home
"medical": "1", // If sick dog accepted
"other_dog": "0", // If other dogs at home
"window": "1" // Need window grills at house
}

Response Body:
{
"dog_1":{
"dog_pic_file": "benji /benji.jpg",
"dog_link": "https://sosd.org.sg/wpcproduct/benji/?form=adopt",
"dog_name": "Benji",
"dog_gender": "Male",
"dog_age: "10y",
"dog_home": "1",
"dog_organisation": "SOSD",
"dog_description": "Benji has a bite history and will need a firm owner who is willing to train him to be a good canine citizen. He also has Separation Anxiety and cannot be left at home alone currently, so his potential adopter

must also be ready for this. Due to his aggression, he will NOT be suitable in households with young children. ",
"dog_index": 292
}
}

Table 2 API "recommend_by_history"

Other important APIs and their description are as follows:

API	**Description**
signin	Login to the system
signup	Receive the adopter's preferences, processing and storing them in the Database.
adopter_new_recommend	Processing adopter's preference, invoke the recommender and return a series of dog index.
recommend_by_dog_index	Processing adopter's input dog index, invoke the recommender and return a series of similar dog index.

Table 3 APIs from the Backend Server

The backend server and the recommender are implemented in one project, which makes it easy for backend to invoke recommender's methods. Also, the knowledge base can be shared within both components. Recommender provides 2 methods for backend server which are listed as follows:

Method	recommend_newuser(data, gender, age, accommodation, description, tags, corpus, corpus_words, target_num, filter_hard)	recommend_userhistory(data, dogs_filter_index, history_index, target_num)
Param	- data: dog information from the knowledge base - gender: dog gender preference - age: dog age group preference - accommodation: adopter's accommodation - description: dog personality preference - tags: some filter conditions - corpus: dog corpus - corpus_words: some words to describe a dog - target_num: the required dog index amount - filter_hard: soft constraint or hard constraint	- data: dog information from the knowledge base - dogs_filter_index: filter conditions - history_index: a series of dog index that adopters are interested in - target_num: the required dog index amount
Return	Dog index list	Dog index list
Description	Recommending for a new adopter by his/her preferences	Recommending similar dogs by input dog index

Table 4 Methods from the Recommender

We chose to use two excel files to store dogs' information as our knowledge base, because dog's information is structured data which can be fit in a table, and the data barely changes once the data processing stage is done. Recommender and backend server will load them to memories when required.

Field	**Type**	**Description**
index	Integer	The column order that marking the dog position in knowledge base excel
name	String	Dog's name
age	Integer	Dog's age
gender	Integer	Dog's gender
personality	String	The description of the dog's character, the major factor for recommendation
HDB_approved	Integer	If this dog is allowed to be kept in HDB
picture file	String	The picture file name
organization	String	The shelter organization that the dog comes from
link	String	The shelter organization's link for this dog

Table 5 Major Dog Information from Knowledge Base

Database

We chose SQLite3 to stores adopters' information, because SQLite3 is a small, fast, full-feature database engine. All the adopter information is stored in a database file. When an adopter register to our system, his/her name and preferences will be recorded. They can use this information to login and get recommendations. If an adopter is not satisfied about current recommend result, he/she can update their profile and generate recommendations again. The UML diagram below shows the fields and methods that an Adopter object contains.

Adopter
+ id: INTEGER, PRIMARY KEY + adopter_name: VARCHAR + password: VARCHAR + accomodation: INTEGER + prefer_age_group: INTEGER + prefer_gender: INTEGER + personality_preference:TEXT + recommend_dog_index: VARCHAR
+ insert_adopter_to_db(adopter) + update_preference_to_db(adopter) + update_recommend_dog_index_to_db(adopter) + is_adopter_in_db(adopter_name, password) + query_adopter_from_db(adopter_name)

Table 6 Adopter UML class diagram for web application

Frontend Interface

Our frontend interface is build based on Vue, which is a JavaScript framework for building user interfaces. It builds on top of standard HTML, CSS and JavaScript, and provides a declarative and component-based programming model that helps you efficiently develop user interfaces, be it simple or complex.

Information Interaction Management

We build a whole information interaction management, including user registration, user login verification, basic recommendation, user preference profile updating and similar dog recommendation. In general, our website will record every key behaviour from users and use Axios (which can easily communicate with server through the HTTP protocol) to request backend server to call SQL methods and execute relevant instructions. Besides, Axios can also help to deal with the backend server response such as data verification results and recommendation results, then we refresh the elements that should be displayed on webpages based on response information. The picture below shows how this process works.

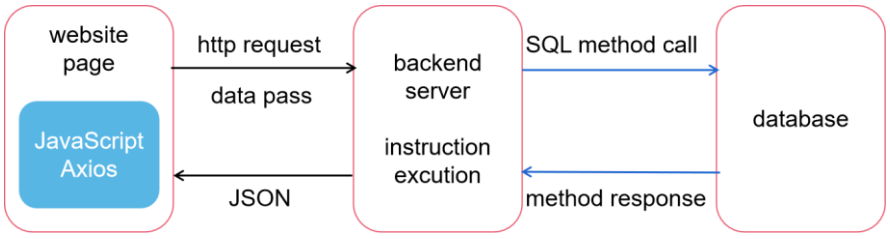


Figure 5 information interaction process

The following picture shows the detailed process of information transmission in one of these functions

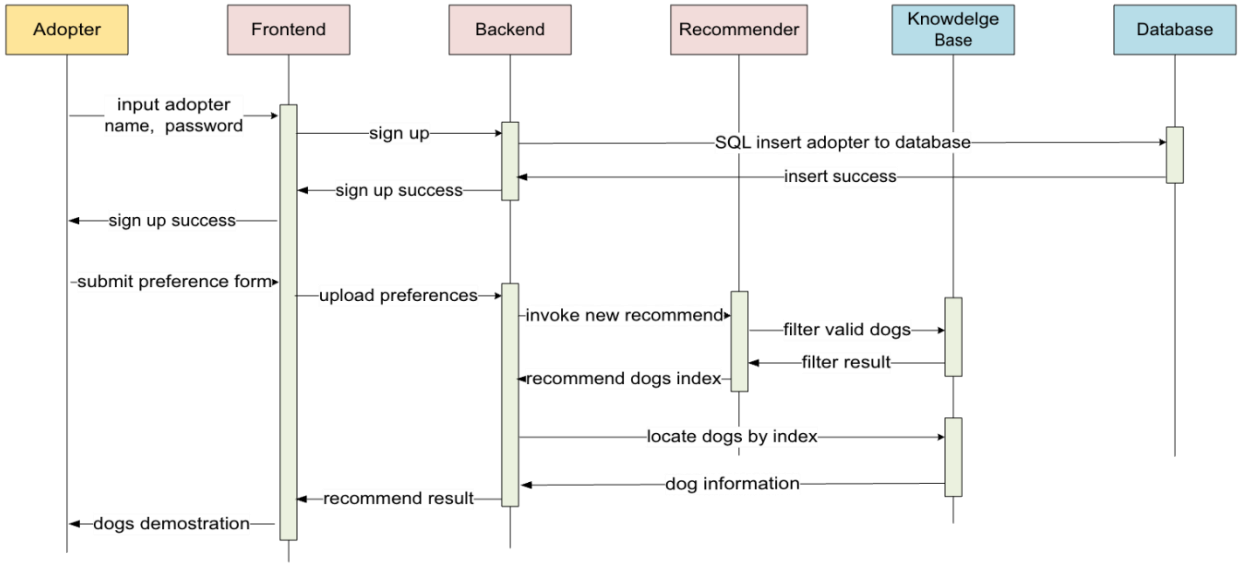


Figure 6 A new adopter recommend process

Frontend User Experience

To save time on development, we chose to use Vue framework templates provided by Element-ui. Because all Vue templates are syntactically valid HTML that can be parsed by spec-compliant browsers and HTML parsers, this makes it easier to present all static content we want to display on the webpages. And we use v-bind directive to bind element's attribute in sync with the component's property which makes us possible to record information from user.

As long as we receive data from backend, every updated information will be stored in SessionStorage of windows as temporary variable. After receiving instruction, the value of elements will be replaced with the variable property from the corresponding component instance via text interpolation using the “Mustache” syntax. This is very fast and will bring good user experience.

We want our website to be clear and concise, thus we set the user login page as the default page. Each functional page after login in can only be triggered by the button of previous page, and user can choose to go back or log out to reset information on most of the pages as well. In this way, user can easily be guided to use our website. The below table shows elements and function of each page component.

Page Component	Element	Function
Login.vue	Username/password input box Confirm button To register button	Website entrance, pass user login information to back-end to do authentication, guide user to homepage
Register.vue	Username/password input box Adoption preference option/input boxes User profile option boxes Confirm/Form reset button To sign button	User registration interface; pass user basic information/dog preference to back-end and store. After clicking confirm will lead user back to Login page
HomeView.vue	Welcome Information/website introduction Dog recommendation button Sign out button	A short introduction about what is our website doing, lead user to the recommendation page
Recommendation.vue	Recommended dogs’ information (default number is five, show empty picture if there has not enough recommended dogs “Yes I want it” button Preference change button Back button/sign out button	Display recommended result from back-end, including dog basic information and its picture; user can use “Yes I want it” button to go to the original dog adoption page; user can click any other dog’s picture to view clicked dog’s details; user can also change profile to query a new round of recommendation
Preference.vue	Similar as register.vue (without username/password input box) Confirm/Form reset button	When user doesn’t like recommendation result or we don’t recommend enough dogs, user can change profile to view more recommend possibility; confirm button will directly lead user to recommendation page

dogdetail.vue	Detail information of dog which is clicked by user "I'm interested" button "Back to previous" button	User can view details of dog the user clicks. If user is interested, click "I'm interested" button to call the backend server to do similar dog recommendation, then user can view more similar dogs as the user chose

Table 7 Elements and function of each page component

Besides, we also have obvious alert such as "*username has already been used"* or "*Unable to find a match, please consider changing preferences or typing more details in the Personality Description box"* when backend returns relative unsuccessful messages. The whole interactive logic is in picture below

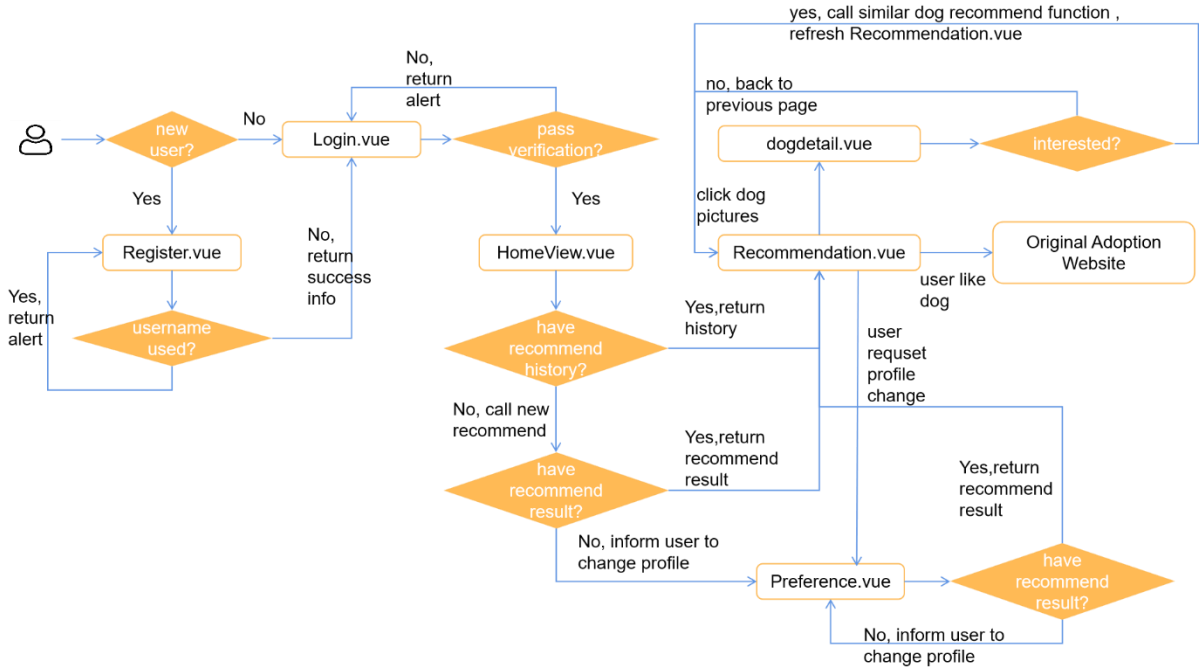


Figure 7 Interactive logic

Running Examples

We tried to input different dog personality descriptions and get some interesting results

1) New user recommending (user without recommending record)

User Input	Recommend Result (highest cosine similarity)
Dog_gender: male&female Dog_age: puppy&teen/young adult Personality description: smart dog with beautiful outlook, which is outgoing and likes to play games Dog_accommodation: HDB No kids, no elderly at home	We have OliveBlessing Female 3 years 2 months HDB_approved She has Outgoing and fearless personality. Gel herself into the shelter pack on the night she was rescued and slept among them so comfortably. Intelligent. Olive means peace. 
We can see that OliveBlessing perfectly matches with our requirements, she is outgoing and beautiful	
Dog_gender: male Dog_age: puppy&teen/young adult&adult/senior Personality description: be friendly with kids, wonderful walking partner Dog_accommodation: Condo Have kids, no elderly, no other pets at home. Don't have experience with dogs	We have Ron Male 8 years 8 months HDB_approved Ron is gentle and a little shy but friendly towards people. He enjoys walks but needs coaxing sometimes when on the leash. 
The characteristic of Ron is highly related to what we want for a dog, he's friendly to people and likes walking which makes him a good walking partner	

Table 8 New User Recommending

2) Similar dog recommendation (based on user profile and dog's user shows interest)

We use the same user profile from the above table and add dog's information to the recommend system, and this is what we got.

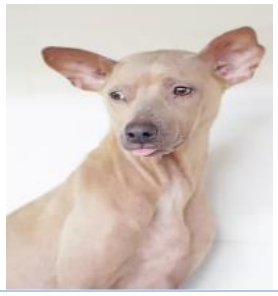
User Input	Recommend Result (highest cosine similarity)
Dog_gender: male&female Dog_age: puppy&teen/young adult Personality description: smart dog with beautiful outlook, which is outgoing and likes to play games Dog_accommodation: HDB No kids, no elderly at home Interested dog: Amber Blessing Amber is a good mummy who keeps her eyes on her child Olive. Smart and observant too. Amber means endurance. 	We have Starz Blessing Male 3 years 9 months HDB_approved Starz was rescued on the night of 26 May 2019 following repeated sightings by feeders that this bunch of puppies is extremely skinny and skittish. Starz is a smart puppy, observant and quick learner. Sociable with other dogs too . 
From the description, Amber and Starz are very similar, they are both smart dogs. Besides, Starz is very outgoing, which is the exact dog we are looking for.	
Dog_gender: male Dog_age: puppy&teen/young adult&adult/senior Personality description: be friendly with kids, wonderful walking partner Dog_accommodation: Condo Have kids, no elderly, no other pets at home. Don't have experience with dogs Interested dog: Nugget Nugget is an effervescent, excitable and friendly boy who can sometimes get over excited. He enjoys his walks but would need to be guided on leash. 	We have Martell Male 8 years 10 months HDB_approved Martell loves attention and daily walks. He is generally friendly with people, but he needs time to warm up to strangers and show his affectionate side. He is energetic and sometimes, when he becomes too excited, he might pull on the leash strongly. Hence, an experienced handler/owner is preferred. 
Nugget and Martell are both friendly, energetic and enjoys walking!	

Table 9 Similar Dog Recommendation

Discussion and Future Improvements

A test environment of the system was launched and we gathered some users to test try the environment. Below are some of the feedback received and the action we took to address them:

1) One comment was that it wasn't clear what can be clicked on the page. We have noted this but due to time constraints, we are only able add the sentence "click picture for more information" to make things clearer.
2) Another comment indicated that the recommended dogs should show some brief information too so that they do not need to click into it to view. We added a preview page for the recommended dogs to accommodate to it.
3) Another comment was for us to provide an option to reject the recommended dog and to provide an option to manual browse through all the dogs in the database. Unfortunately due to time constraints, we were unable to accommodate to this comment and would add this to the list of functions for future improvements for the system.

Apart from that, for future improvements, we would like to review our recommending algorithm. Our recommending algorithm is a bit limited due to our sources of data. As we have no prior user data at the beginning, such as user's ratings for a dog, we can only apply Content-Based Recommender for our system. If we have run the system for some time, we can get more information regarding the user's preference of the dogs. We can analyse which dogs are more preferred by which kind of users. Then we can apply more recommending algorithms, such as Collaborative Filtering to optimize our recommender.

We also hope to be able to interview representatives from each organization to understand what they are looking for in adopters so that we may provide better matches for the adopters, maybe even discovering organization-specific criteria along the way.

If possible, it would be good if our system can link to their sites to provide the latest updated information on the knowledge base, e.g. removing dog profiles for dogs that are newly adopted or updating dog profiles when information is updated. As of currently, our system is unable to automatically update the latest change and some of the dogs in the knowledge base may have already been removed from the website, causing the "Yes I want it" link to not work properly.

Due to the differences in adoption processes across organizations, we also decided to link the user to the organisation's website if they are interested in our recommended dog. This also addresses the issue of being unable to provide the latest update to the user. As such, the user will get to access the latest information before they make their final decision. We consider the purpose of the app as achieved when recommending the right dog and by clicking on the "Yes I want it" link, they are directed to the next stage of the adoption process. Unfortunately, due to the nature of the next step, we are unable to provide an offline platform for this.

Appendix 1 Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)

PRACTICE MODULE: Project Proposal

Date of proposal: 25 September 2022
Project Title: Find Your Dog - the one stop app to adopt a dog
Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email:zhan.gu@nus.edu.sg
Background/Aims/Objectives: **Background:** Pet ownership is quite common in Singapore. Statistics show that 33% of Singaporeans are pet owners as of January 2022 and 17% used to own one but not anymore. Needless to say, dogs were the most popular pet type among pet owners, with an astounding 58% respondents reported owning dogs. Currently, there are two ways of owning a dog: through Commercial Purchase and through Adoption. While the commercial purchase route is quite straightforward, there had been some comments that the adoption process can be quite tedious as one would have to go through many procedures to successfully adopt a dog. Among the procedures would include browsing through different websites to find a favourite dog, filling up a long form, contacting the organization, interviews, etc. It didn't help that even on the websites, most of the profiles required manual clicking into dog profiles to read and consider yourself. Research was largely done manually by the potential adopter and many end up purchasing dogs instead as the process was too time-consuming. Furthermore, it is difficult to ascertain what criteria different shelters consider when it comes to approving adoptions. **Aims/Objectives:** This project objective is to be a platform to facilitate the search process for dog adoption in Singapore. The project targets step 1 in the process and aims to reduce the time taken for an aspiring adopter to search for their desired dog by applying intelligent reasoning systems. It also hopes to take into consideration the key approval criteria of each organization to better increase the chances of an aspiring adopter to successfully adopt a dog.
Requirements Overview: Information acquisition ability (web-scraping, interviews, information searching) Programming ability (recommender, frontend, backend) Data processing ablility System integration ability
Resource Requirements (please list Hardware, Software and any other resources) **Hardware proposed for consideration:** Laptop or desktop computer **Software proposed for consideration:** Virtual environments, e.g., Anaconda environment

Developing tools like Jupyter Notebook or Google Colab
Virtual machines, e.g., Virtual Box
Windows or Linux systems
Python and necessary packages
Frontend UI editor

Number of Learner Interns required: (Please specify their tasks if possible)
A group of 4 students

Methods and Standards:
Description of the system:
Input: User profile containing user's preference for dogs
Output: Dogs that are similar to the user's preference
Basic components: Knowledge Base, Recommender, Frontend Interface, Backend Server, and Database
Methods:

System Components	Objective	Main work
Knowledge Base	Provides information about dogs. Get data from different shelter websites via web-scraping.	Web-scraping Interview NLP Clustering
Recommender	Processes natural language calculates dogs' similarity and generates recommendations.	Define features Filter the dogs Process input data Calculate similarities
Frontend Interface	Provide a graphical user interface to the adopter.	Use Vue to develop the interface Record behaviours from users Give requests to backend
Backend Server	Integrates each component together and directs the flow of data in the system.	Define APIs Process information from knowledge base and database Direct the flow of data
Database	Stores information about adopters.	Use SQLite3 to store user's information

Team Formation & Registration

Team Name: Group 12 of Intelligent Systems

Project Title: Find Your Dog - the one stop app to adopt a dog

System Name: Find Your Dog

Team Member 1 Name: Jonathan Lim Ching Loong

Team Member 1 Matriculation Number: A0261707E

Team Member 1 Contact (Mobile/Email): e0983101@u.nus.edu

Team Member 2 Name: Hu Hang
Team Member 2 Matriculation Number: A0261634H
Team Member 2 Contact (Mobile/Email): e0983028@u.nus.edu
Team Member 3 Name: Fu Jia
Team Member 3 Matriculation Number: A0261661H
Team Member 3 Contact (Mobile/Email): e0983055@u.nus.edu
Team Member 4 Name: Du Qiao
Team Member 4 Matriculation Number: A0261990X
Team Member 4 Contact (Mobile/Email): e0983384@u.nus.edu

Appendix 2 Mapped System Functionalities against knowledge, techniques and skills of modular courses: MR, RS, CGS

Courses	Knowledge Used
Machine Reasoning	We build up our knowledge base to store the data and information for our dogs. During this process, we use web-scraping and manual elicitation way such as interview to do the knowledge acquisition.
Reasoning Systems	We use the recommendation algorithm learnt at this course to build our recommender to recommend dogs for adoption.
Cognitive Systems	We use NLP and TF-IDF indexing learnt at this course to process the descriptions of the dogs and the descriptions given by the user to do the recommendation based on text features.

Appendix 3 Installation and User Guide

Preparatory work

1) Having an appropriate web browser, recommend Google Chrome
   Download link: https://www.google.com/chrome/
2) Clone or download project source code from GitHub
   Path:https://github.com/dq04171877/IRS-PM-2022-10-30-ISY5001-Grp12-Find-your-dog
3) Python or Anaconda, make sure you have installed all required libraries as stated in requirements.txt in your environment

✓ numpy
✓ pandas
✓ scikit-learn
✓ spacy (also need to download en_core_web_sm)
✓ flask-cors
✓ openpyxl

Back-end server development guide

1) Open the console and set the working path to the path where you put your local copy of the GitHub repository and change path to \SystemCode\backend
2) Install required libraries using commands below:
   - pip install -r requirements.txt
   - python -m spacy download en_core_web_sm
3) Run main.py in console to start the back-end. The following lines in console means your back-end server is running successfully.

```
C:\Users\12947\Downloads\Backend>python main.py
[('adopter',), ('sqlite_sequence',)]
 * Serving Flask app 'SURVEY_API'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5002
Press CTRL+C to quit
```

Important notice:
Keep the back-end server running for all the time you using our system
Python versions 3.7, 3.8, 3.9 and 3.10 have been tested for the back-end program and can run successfully as long as the necessary packages have been installed and the version of the packages can match the Python version.

4) Follow the front-end file install guide to start the system.

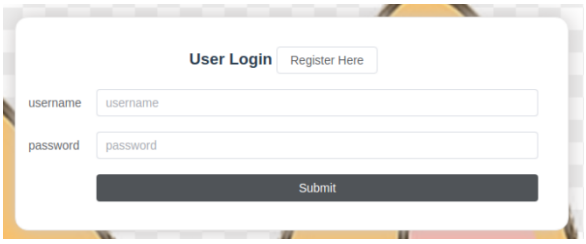
Front-end file install

1) Under the path of your local copy of the GitHub repository, change path to \SystemCode\frontend

2) Open index.html using web browser to view the default page and do further operation

Open commands in console with default browser:

- Windows: start index.html
- Linux: see index.html
- Mac: open index.html



First use guide

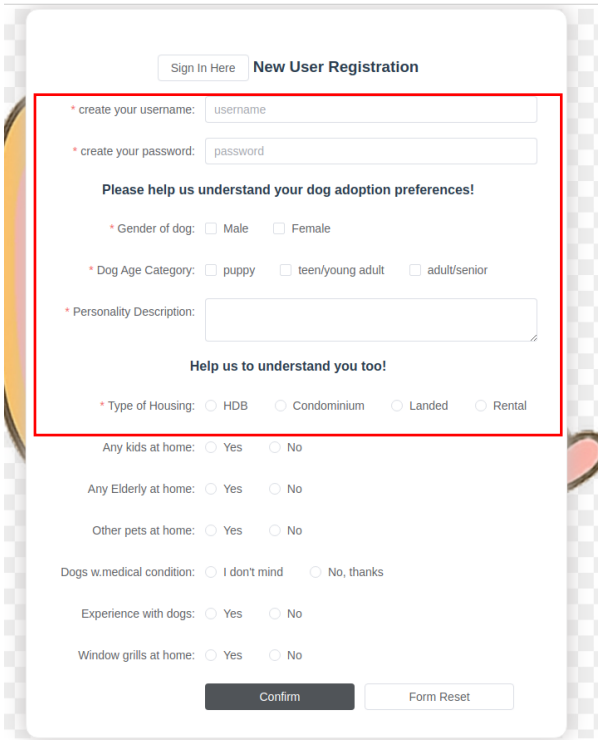
Register

For new user, you have to sign up on our website before login. You have to fill in the part where the red box is circled in below picture. You have option to answer or not answer the remaining questions based on your willing.

Tips:

Try to describe as much as detail as possible what kind of personality you are looking for a dog to get precise recommendations.

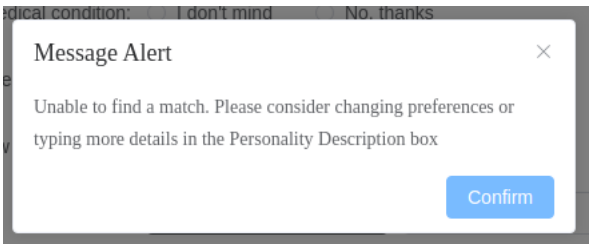
If you are unable to view the whole page, you can adjust the window ratio to fit



Click “Confirm” button will automatically directly guide you to the login page

Dog recommendation

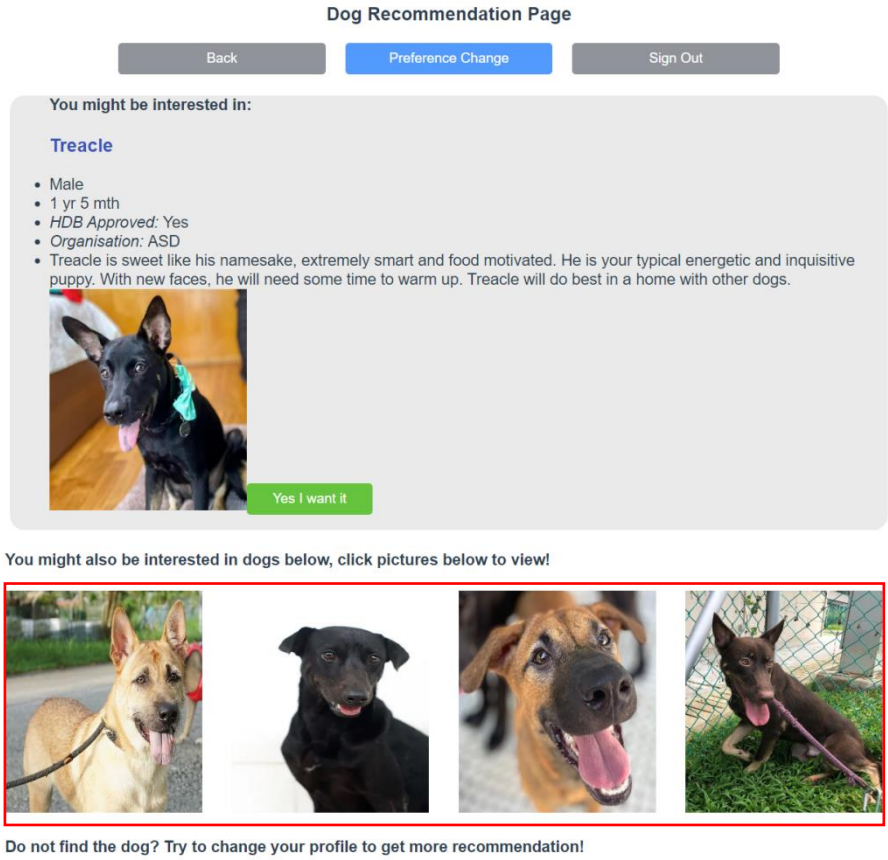
Click “Dog recommendation” button to start recommending process. There are chances you may meet a pop-up window popping up as below. That means we didn’t match any dog based on your given information, and you have to refill your preference form.



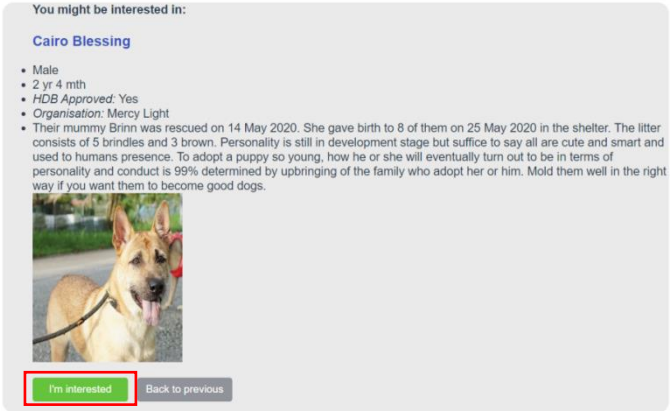
When the back-end successfully returns the recommendation result, we will automatically guide you the the dog information view page, where you can choose your favorite dog.

Dog recommendation result view

In this page, you can view our recommendation result, including basic information of the dog that best matches your profile and other dogs’ picture you might be interested.

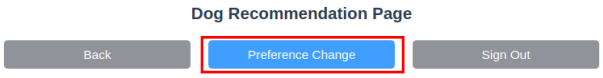


You can click any picture in the red box to view the dog's detail and see if you have interest.

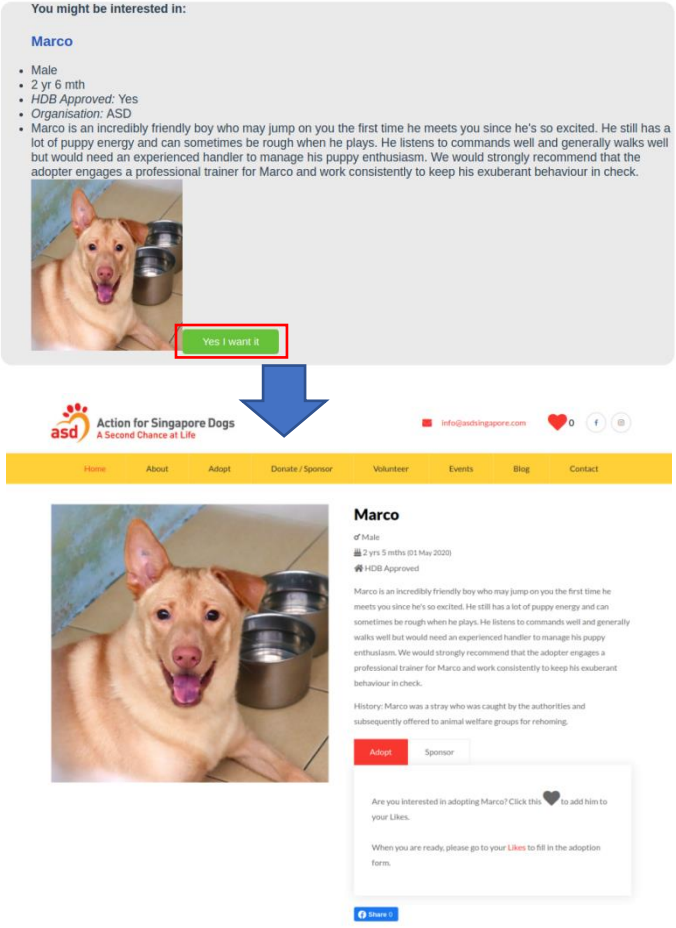


If you like this dog, click "I'm interested", our recommend system will find more similar dogs for you to choose and refresh the dog recommendation page.

You can also change your preference if you have no interest in any of dog we recommend for you by clicking this button.



When you find the your favorite, just click "Yes I want it", then we will guide you to the original organization website to see how to adopt it.



Appendix 4 Individual Reports

Individual Report for Jonathan Lim Ching Loong A0261707E

Personal contribution to group project

For this project, I took on the role as team leader. As such, I chaired meetings and assigned due dates to the assignments. I'm also the main person in building the Knowledge base. I performed web-scraping for SOSD, Purely Adoptions, and Causes for Animals to obtain the dog profiles for the knowledge base, and after consolidating the web-scraping results from the other team members, I performed data pre-processing for the consolidated data. These pre-processing steps include text pre-processing, TF-IDF and KMeans clustering, selection of clusters, and finally assigning labels and tags. For continuous data, I also processed and set the age categories.

Apart from that, I also interviewed the domain expert to generate the manual elicited knowledge and processed the data to form more inputs for the knowledge base. After the UI prototype was created, I conducted the testing sessions with non-related people to gather feedback and proposed improvements that were suggested.

I also contributed by doing audio and video editing to generate the two presentation videos at the end of the project.

What learnt is most useful for me

1. Web Scraping

This is the first time I've performed web-scraping and the process taught me a lot. I've learnt the useful libraries in Python to perform web-scraping as well as how some websites can stop you from web-scraping. I've learnt how to read some html language and how different websites can be built in very different ways and as a result affect the way we perform web-scraping.

2. Text-preprocessing, TF-IDF and Clustering

In order to cluster the dogs based on personality, I created custom text-preprocessing tools with spacy and applied TF-IDF to generate arrays. I've learnt how to understand these arrays better and how to plot useful graphs to analyze the data. I've learnt how to troubleshoot the data and how to select useful clusters after applying the KMeans method. I've actually played around with various clustering strategies and settled with KMeans, which gave me better insights into the differences between different methods.

3. UI Design and Deployment

I've learnt how to make the knowledge we learnt in class into a functional, physical app that can carry out useful functions. One of my test users who is an expert in designing systems also gave me very valuable tips on app design and how to make things more practical for a user. While we did not implement these tips due to time constraints, I learnt a lot from it and will keep in mind for any future apps that I'm involved in.

How to apply the knowledge and skills in other situations or workplaces

Natural language processing is a trending thing now and to gain more experience and knowledge in the basics of it (e.g. text-preprocessing, TF-IDF) is definitely very useful for future situations and in the workplace. Couple this with the knowledge I've picked up in UI design, I believe I contribute to design on NLP apps such as translation or interpreter apps.

Individual Report for Hu Hang A0261634H

Personal contribution to group project

In our project, I took part in developing the recommender system and part of web-scraping. I did web-scraping from a shelter organization website and got information such as name, gender, HDB approval and personality description of the dog. I got the data from the web using Python and the Beautiful Soup package. The data was used to build up our database of the dogs. Apart from participating in the web-scraping, my main contribution to the group project is to develop the recommending system. I analyse our system situation based on the data we have and the result we want and decide to build a recommender that is mainly content based. I made two main functions for the recommender: recommending for new users and for the old users. I did the programming and wrote the functions so that the recommender can use the information received from user and give recommendations of the dogs for the user. I added some rules to the recommender to make it more suitable for some constraints being considered in recommending. I tested and made some improvement to the recommender.

What learnt is most useful for me

1. Web scraping

In the project, I learnt how to do web scraping by myself and use it to get the information we need from the shelter's websites. I learnt the functions of the Python package Beautiful Soup. Also, by doing web scraping, I learnt the basic structure of a website. I learnt how to pick up the information useful for us in the html file of the website.

2. Recommending algorithms

By developing the recommending system, I get better understanding of the theory and algorithm of recommending system. I compare the function and usage of different algorithms and select the one that is suitable for our project. I also learnt made some improvements and add some rules to the recommender for it to make more precise recommendations.

3. The developing process of a system

The process of developing the system consists of many steps. From the early discussion to come out with the idea of what to do, to completing different task of the project, I witnessed and took part in the developing process of a system. I learnt how to develop an app with my teammates. Also, by doing the coding of develop a system, I learnt some good programming skills and habits that can be useful in workplaces.

How to apply the knowledge and skills in other situations or workplaces

The recommending algorithms are very popular in various kinds of apps. In this project, I developed a recommender to recommend dogs for different people according to their preferences and personal conditions. Such techniques can be used to develop other recommending systems to recommend products for people to buy, hotels for people to stay and so on. The recommender in this project also works with NLP processing of texts. Such technique can be used to do recommend news or articles to the readers who are interested in them. I also learnt the skills of doing web-scraping. Information acquisition is an important part of developing systems and apps. With web-scraping we can get information from the internet more efficiently.

Individual Report for Fu Jia A0261661H

Personal contribution to the group project:

Based on my working experience as product assistant, I contributed a lot of ideas about what functions we need for our project and how do we link each part. Fortunately, most of them was adopted by my team and I've promoted these functions to be done in time. I also gave a lot of positive feedback and revision suggestions for the work of my teem members, which made our team working with high efficiency.

Besides, I completed the UI interface and interaction process deign, and implemented the front-end framework. Thus our group lacks experience of front-end developing, I learned online and built a complete user interface based on vue framework. After debugging the front-and-back-end interaction process together with the teammate, we basically accomplished all functions we designed at first. This is very important to set up links between users with our recommend system.

What learned is most useful for me:

This is the first time that I have participated in completing design and implementing of a whole system, I have a clear understanding of what kind of work should be done in each part after this project.

From the start, we all search files online actively to know about what information the Singapore dogs adopt website can give to user. In this process, we found out that we can match dogs' information with users based on users' requirements to reduce users' information searching time. Then we tried to generate domain knowledge through ways like interviewing dog adopters, and form our knowledge base. From these work, I have trained my ability of information searching and knowledge summarize. Through the meeting with teem mates, I also learned how to use knowledge base to build a recommendation system.

For the front-end development which is the part I have focused on the most, is actually brand new to me. But it is very interested to know how is the data transferring through servers, and how could we dynamically display information user might be interested to the web pages. My studying ability was also trained in program debugging.

How can I apply the knowledge and skills in other situations or my workplaces:

After this project, I understand how is an intelligent reasoning system formed. I could use these experience to build other intelligent systems in similar situation which I think will benefit a lot to my working field. For instance, instead of pets recommendation, we could use similar approaches to build a fashion recommend system which I believe will be popular among girls.

Individual Report for Du Qiao A0261990X

Personal contribution to group project

In the topic selection group meeting, I brought up the idea of pets. Then we agreed it was an interesting topic and decided to do a dog adopting website as our group project.

Since I'm the only one in the group who has full-time software engineering experience, I took the responsibility of integrating the whole system and maintaining the project files on GitHub.

For technical selection, I proposed a web application with a client/server architecture and chose the framework. To verify my thought, I built an early version of the web application including frontend, backend, and database. After they connected successfully, I concentrated on the backend server development and brought the recommender, the knowledge base, and the database together. In this process my team members and I were working hard to develop, debug and improve user experience. Sometimes it kept us up late at night.

What learnt is most useful for me

In the preparing stage, we learnt to use beautifulsoup library and scratched dog information from adopting websites, which is useful because it's an efficient way to collect information from web pages.

Apart from machine learning knowledge from class, another thing I learnt is program management. Although I worked as developer before, I never built a web application. In my previous job I used different technical stack and most of time I was working on an existing project rather than creating a new one. This group project gave me a chance to think how those components are working together and implemented it in an organized way.

How to apply the knowledge and skills in other situations or workplaces

I can use the NLP skills to extract people's view towards public issues and capture the trend from huge number of websites.

References

1) https://www.statista.com/statistics/1319910/singapore-pet-ownership-rate/#:~:text=According%20to%20a%20survey%20on,do%20not%20own%20a%20pet.
2) https://www.statista.com/statistics/1001965/singapore-pet-ownership-rate-by-pet-type/
3) https://www.aaha.org/your-pet/pet-owner-education/aaha-guidelines-for-pet-owners/life-stage-canine/
4) https://vuejs.org/guide/introduction.html#what-is-vue
5) https://en.wikipedia.org/wiki/Flask_(web_framework)