Executive Summary

Over the course of the COVID-19 pandemic, pet ownership has shot up in numbers and this leads to an increased the demand on pet stores and shelter adoptions. While there are many benefits to adoption over purchasing a dog, the process to adopt a dog can be relatively more difficult due to the diversity of dogs at shelters and also the manual process of most things.

Looking across various animal adoption sites in Singapore, the process of searching for a dog involves browsing through a catalogue of dogs and reading profile by profile, which is quite tedious and involve a lot of research skills to find a suitable dog. On the other hand, the adoption sites are also not linked, meaning a potential adopter will need to go through multiple sites to research to come to a conclusion.

Our system intends to speed up the research process of adoption by consolidating data from various sites, applying NLP techniques to cluster/categorise dogs by their personalities/background and recommend dogs based on the profile data and search history of the user in our system.

批注 **[JLCL1]:** To be updated once full report is completed.

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:

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.

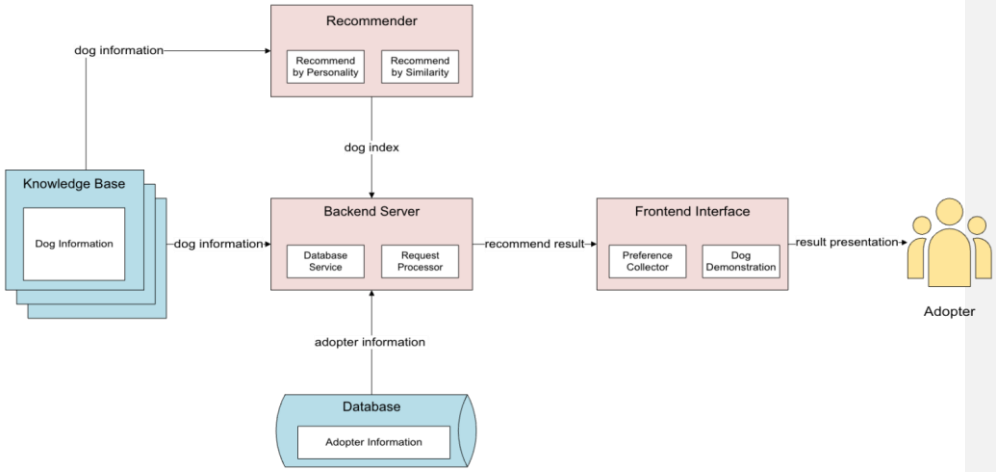
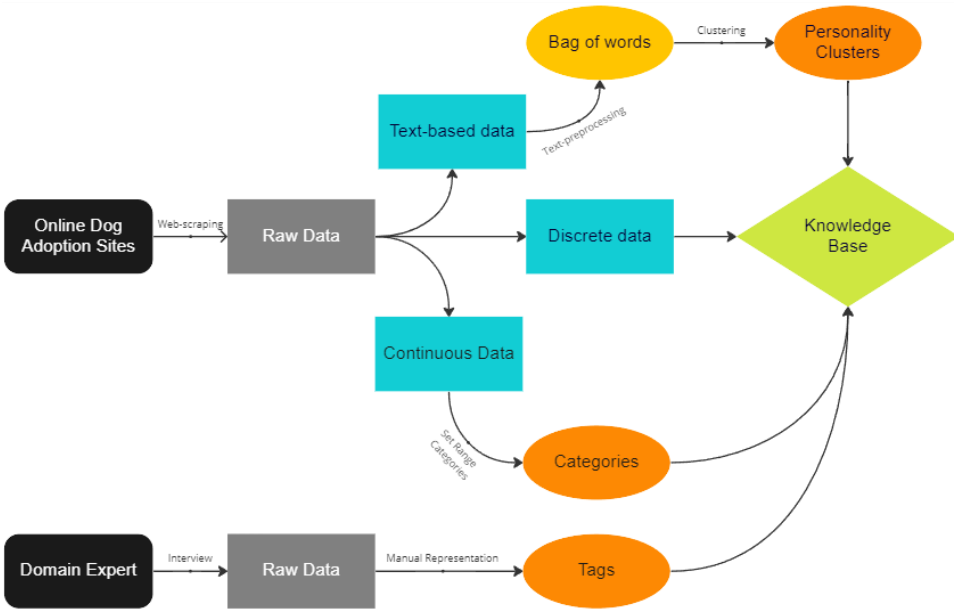


Figure Recommender System Architecture

- 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



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. Pre-processing
    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

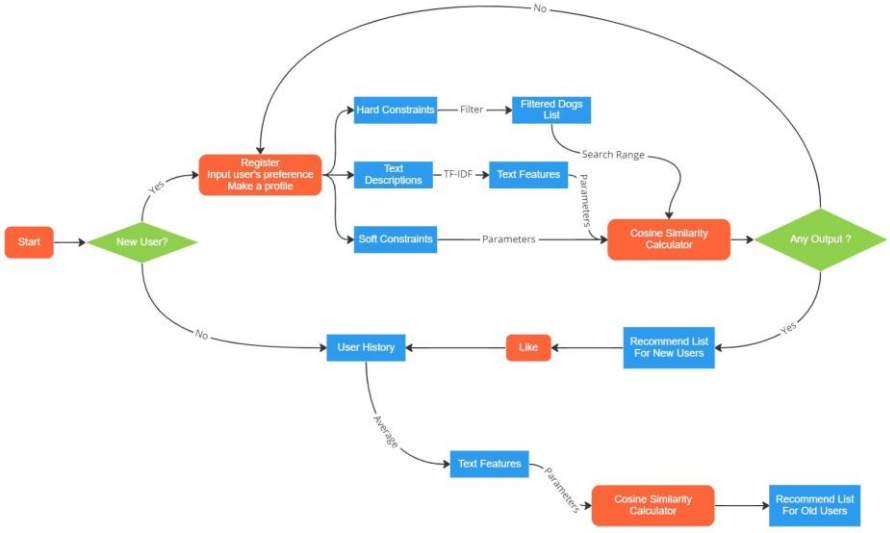
1. Introduction of the recommender

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:



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. The soft constraints are converted to numerical values as below to do the similarity calculation.

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

If one feature does not matter the recommendation, the value will be 0. Otherwise, it will be 1. 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 the number of dogs that match the user's preference does not reach 5, 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 means that

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 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	Invoke Recommender by adopter's preference, return a series of dog index.
recommend_by_dog_index	Invoke Recommender by adopter's preference, return a series of dog index.

Table 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	Description

Table 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 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.

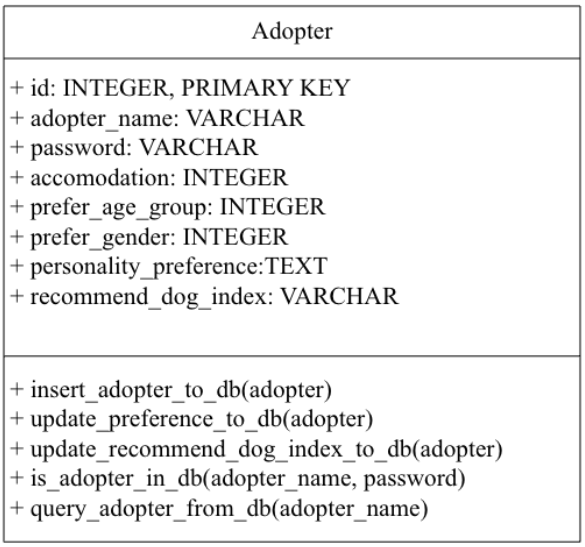


Figure Adopter UML class diagram for web application

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. [6].

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.

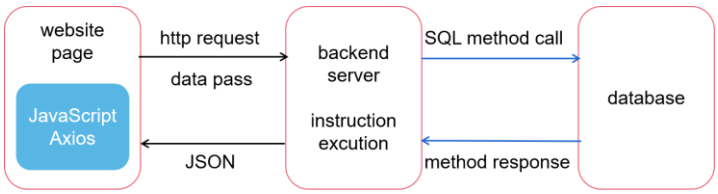


Figure information interaction process

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

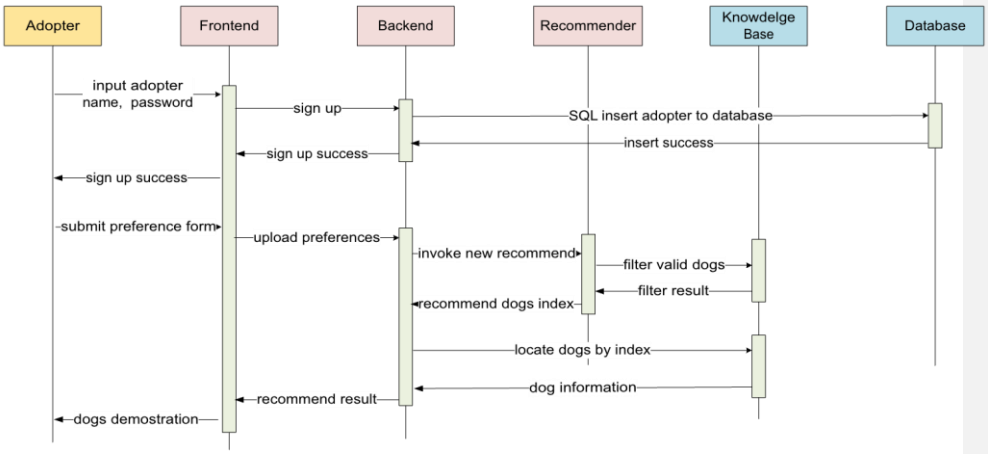


Figure A new adopter recommend process

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 every page 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 more similar dog based on our recommend system; 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 don'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

Table Elements and function of each page component

Besides, we also have obvious alert such as the *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 return relative unsuccessful messages. The whole interactive logic is in picture below

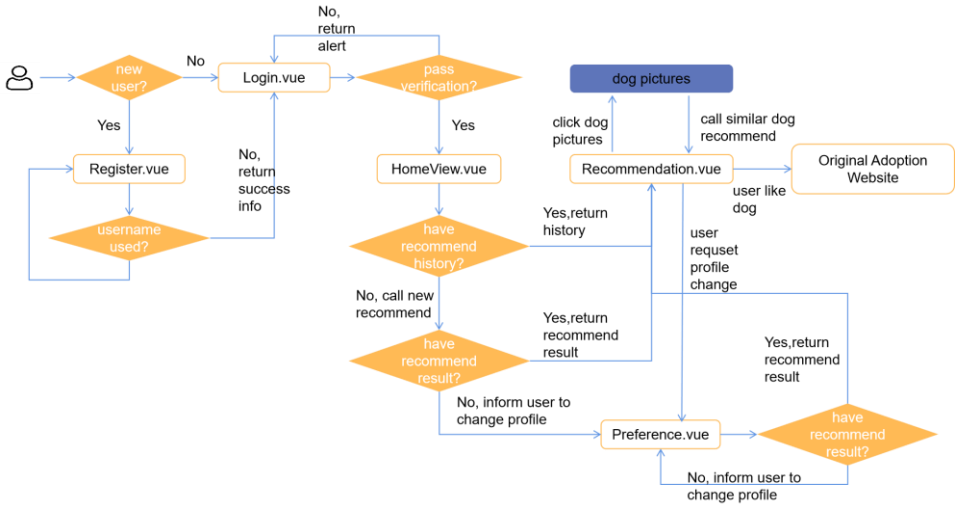


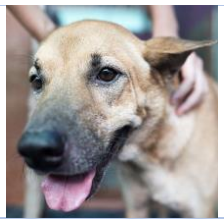
Figure Interactive logic

Running Examples

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

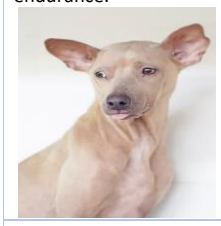
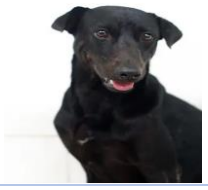
i) new user recommending (user without recommending record)

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	

ii) 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!	

Discussion and Future Improvements

- After we collect a number of user data, we can improve the recommender to use collaborative filtering.
- Organization-specific criteria

Conclusion

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://www.forbes.com/advisor/pet-insurance/survey-78-pet-owners-acquired-pets-during-pandemic/
5) https://www.straitstimes.com/lifestyle/home-design/more-interested-in-adopting-or-fostering-pets-during-covid-19-pandemic-as-they
6) https://vuejs.org/guide/introduction.html#what-is-vue
7) https://en.wikipedia.org/wiki/Flask_(web_framework)
8)