2020-2 Capstone Design(1) Project

Final Report



Professor: Young Bin Kwon

Teacher Assistant: Dr. Syed Farhan Alam Zai

|  |  |
| --- | --- |
| Project Name | Pet Meeting |
| Team Name | First |
| Member Name  And  Student ID | DongMin Shin(20153598)  JaeHyun Shin(20173215)  SungHun Son(20180395) |
| Proposal Document Version  Proposed date and accepted date | 1.2 ( - )  1.3(Sep 13 , Sep 14 No.1 accept)  1.4 (Sep 16, Sep 16) 1.5 (Oct 7, Oct 10)  1.6 (Oct 25, Nov 1) |
| Date | 2020-11-29 |

Table of contents

[1.      Introduction 3](#_Toc57587274)

[1.       Overview 3](#_Toc57587275)

[2.Background 3](#_Toc57587276)

[2. Project Description 4](#_Toc57587277)

[1. Outline of tech stack 4](#_Toc57587278)

[2. Development Details 4](#_Toc57587279)

[3. Database 5](#_Toc57587280)

[3. Goal of the project 5](#_Toc57587281)

[1. Evaluation Metrics 5](#_Toc57587282)

[2. Development Process 5](#_Toc57587283)

[1. Time Schedule 5](#_Toc57587284)

[**2. Comparison Table** 14](#_Toc57587285)

[**3. Member Role** 15](#_Toc57587286)

[**4. Theoretical basis** 16](#_Toc57587287)

[1. Web Socket 16](#_Toc57587288)

[**2. Master and Slave** 17](#_Toc57587289)

[**3. Recommender System** 17](#_Toc57587290)

[4. Jest test 17](#_Toc57587291)

[**5. Open source and Github Activity** 18](#_Toc57587292)

[**1. Open Source** 18](#_Toc57587293)

[**2. Github Activity** 18](#_Toc57587294)

[2. Comments 19](#_Toc57587295)

[6. Comment & Resolution 23](#_Toc57587296)

[1. Comment about proposal 23](#_Toc57587297)

[2. Comment about recommender system 23](#_Toc57587298)

[7. Conclusion 23](#_Toc57587299)

[1. Lessons 23](#_Toc57587300)

[2. Unsatisfactory points 23](#_Toc57587301)

[3. Future plan 23](#_Toc57587302)

[4. Meaning of project 23](#_Toc57587303)

# **1.      Introduction**

### **1.       Overview**

Pet meeting is an online open web platform service and our clients are pet owners and pet sitters.

And this platform enables the pet owners and sitters to fulfill their need for pet entrusting and pet raising by embedding chatting or community inside the platform.

 First, with entrusting function, pet owners can entrust his/her pet to a potential pet sitter who wants to raise the pet during the period that both agreed in advance.

 Second, community service, there are Show off page, Sitting page and Worry page. First, in the Show off page, pet owners can upload their posts that contain pictures or GIF files of their attractive pets with descriptions and users can save, share, and recommend each post to others like Instagram. Next, in the Sitting page, pet owners can upload pet cards that contain pet’s profile and review to entrust their pets, and pet sitters can apply for that. Also in the Worry page, users can share their worries for raising their pets better and vote for good solutions.

 For helping users choose their desirable and suitable posts about pets or information in communities, we introduce our own three hybrid recommender systems, relational search function, suggesting pet function and suitable show-off function. By gathering and analyzing data about what contents each user has clicked, their preferences and review scores, we offer recommendations optimized for pet sitters based on previous activities of them.

### **2.Background**

In Korea, the number of households and people that raise pets have increased.

Concretely, nearly 5.91 million households are raising pets in Korea, and considering the average number of people per household (2.4 people), the number of people raising pets is close to 14.18 million.

And the number of young people who are raising pets also have risen for many reasons, for example national economic difficulties. The 20/30 millennials (born from the early 1980s to the early 2000s) who give up marriage and childbirth and raise pets are increasing while low birthrates have recently emerged as a serious social problem.

Considering these social problems and trends, the necessity for the application for free pet consignment service has risen, but such application does not exist now.

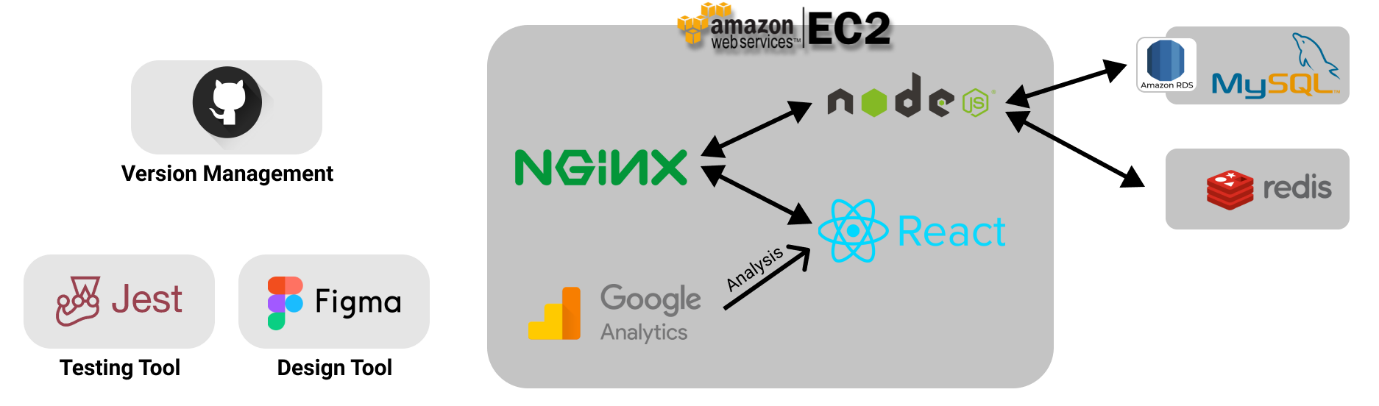
Existing online services only sell their pet consignment service at a burdensome price to young people emphasizing their specialty. If people, especially poor young people, want to leave their pets in other’s care for a while, then they should pay for expensive consignment services in an animal hospital normally. And it must put an economic burden on many current pet owners.

So by providing a web open platform for free pet consignment service, we expect to relieve their burden on expensive pet consignment services directly so contribute to alleviating socio-economic problems.

Since most of the users are pet lovers, active conversation will keep going on community platforms with comments. Pet owners and pet sitters can show off their pets and post the articles.

# 2. Project Description

### 1. Outline of tech stack



### 2. Development Details

* Frontend

1) React - JS Framework

* Backend

1) Nginx - for reverse proxy

2) Node.js - Server application

Socket.io - JS library for real-time communication

Express.js - Web application framework for NodeJS

3) MySQL - RDBMS via Amazon RDS

4) Redis - NoSQL via ‘Amazon ElastiCache’ for real-chatting4

* DevOps, Testing and Monitoring

1) Github - Version Management Service

2) Jest - JS framework for testing

3) Google Analytics - for measuring and analyzing website usage

4) Jest Snapshot - for E2E test

* Design

1. Figma

Before Midterm: Choosing, Main, Home, Entrust, Raise page, Chat room

After Midterm: Show off, worry page

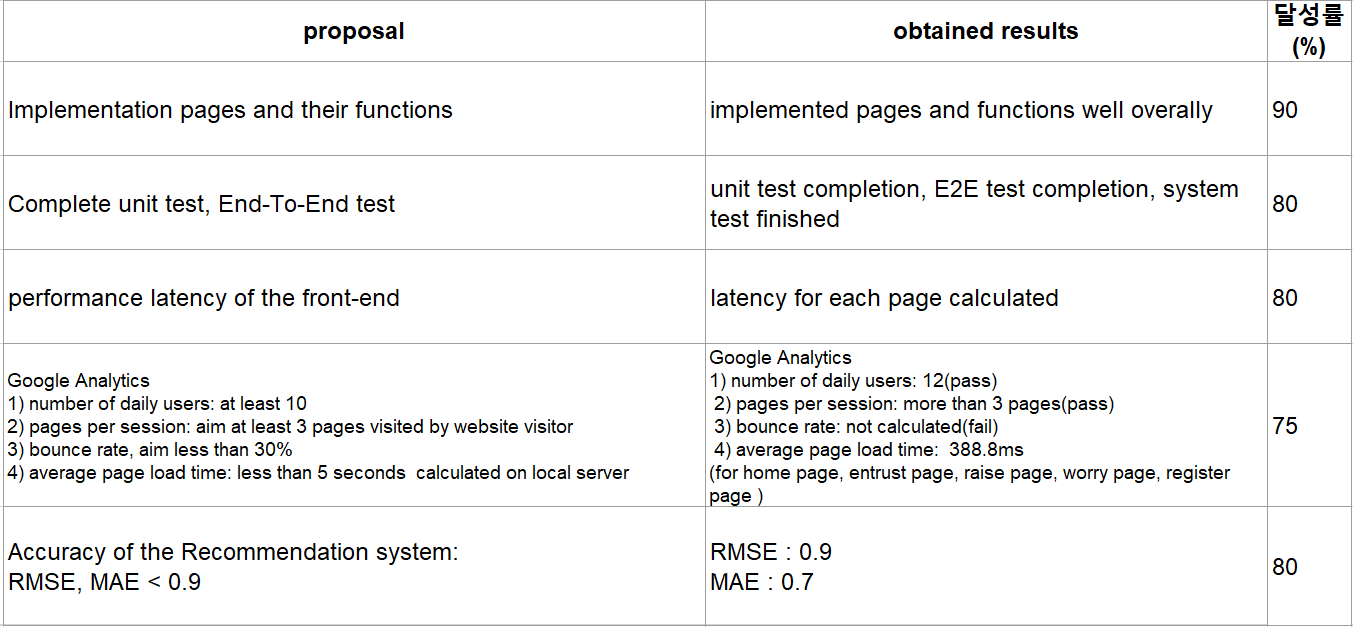
### **3. Database**

We construct our databases into two, MYSQL and Redis. Redis is for saving chatting history. Because users can send their messages by just text,image,video, or even geographical info, and so on. To provide this various types of message, key-value data structure will be used.

We also use a master/slave model to prevent some blackout or either problem. As you know, the master/slave model is easy to implement but it is hard to guarantee consistency of slaves. So we keep track of all history logs of transactions, and monitor whether the DB is consistent or not.

# **3. Goal of the project**

### **1. Evaluation Metrics**

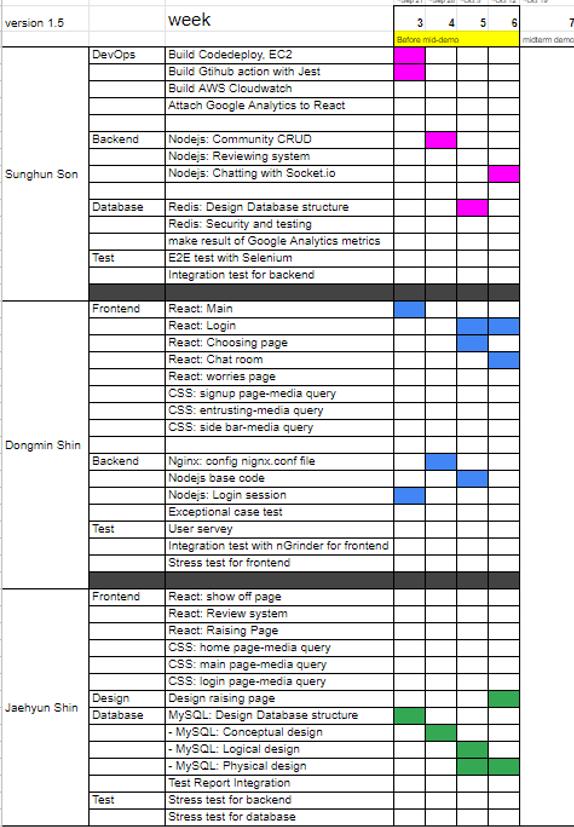


### **2. Development Process**

### 1. Time Schedule

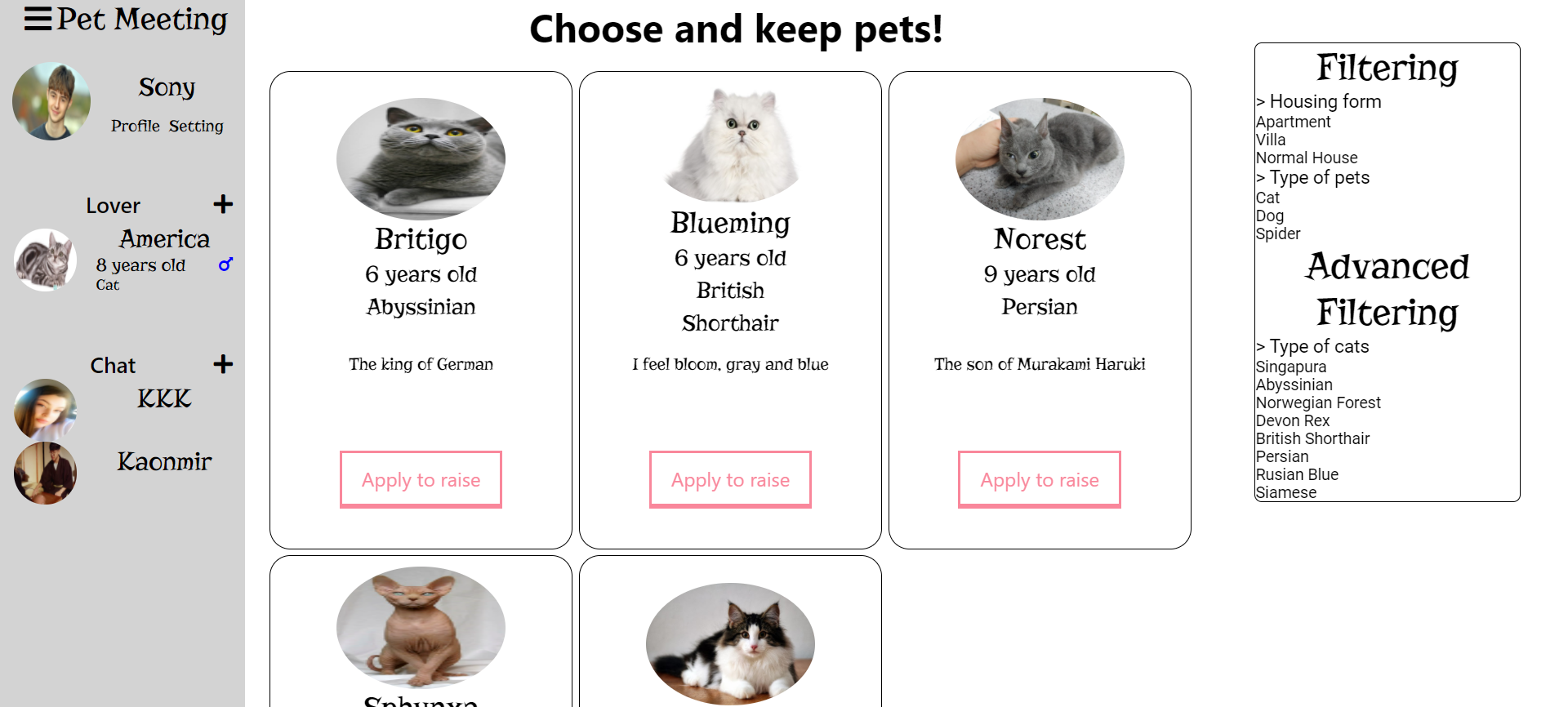
1) Before Midterm Demo

I will list up the works that our team proposed as a project goal before midterm demo and evaluate degree of the task completion based on self-reflection.



1) implemented 5 crucial pages for main function of our web service .

a. choosing page



Users who are sitter can check which pets are listed to be entrusted. Recommender system gives some candidate pets that are likely to fit each user's preference. If a user clicks that red button which ‘Apply to entrust’ is labeled, the system asks the pet owner to choose yes or no. Before the choice, the owner can check the sitter’s profile and reviews of past activity.

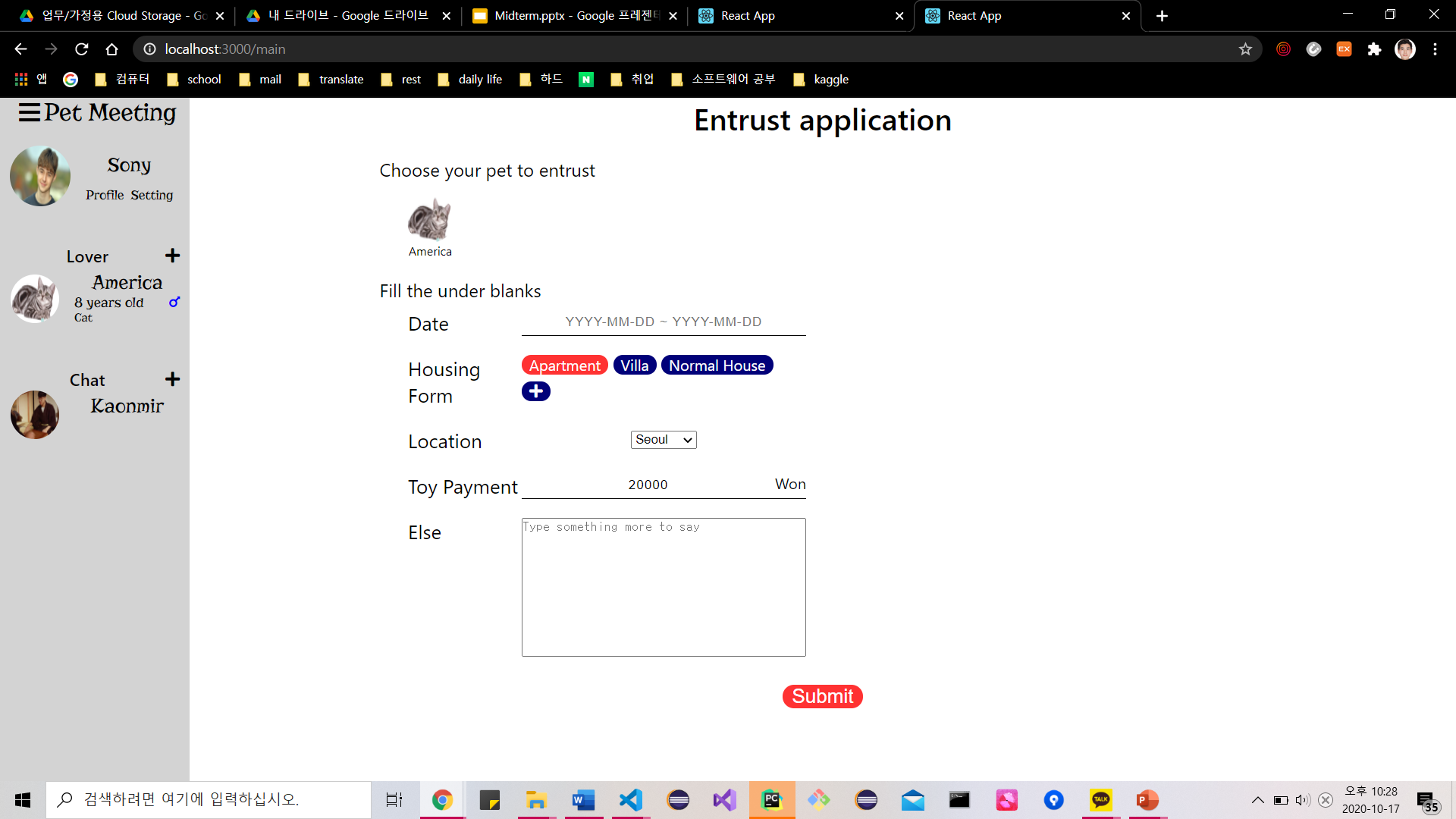
b. Main page

텍스트이(가) 표시된 사진

자동 생성된 설명

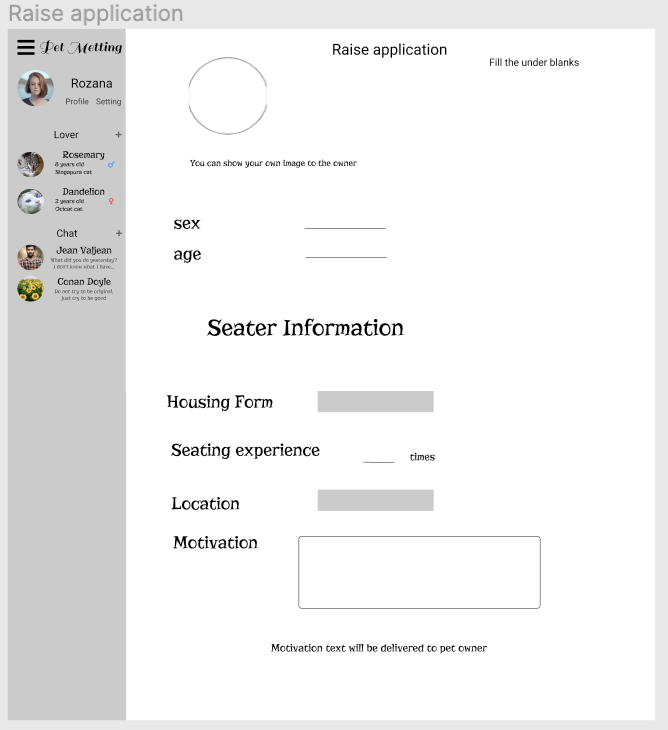
All users, pet owner and pet sitter, will first see this page after they sign in. users can handle most parts of our service on this page. Users can check their pets and people they chat with.

c. Entrust page



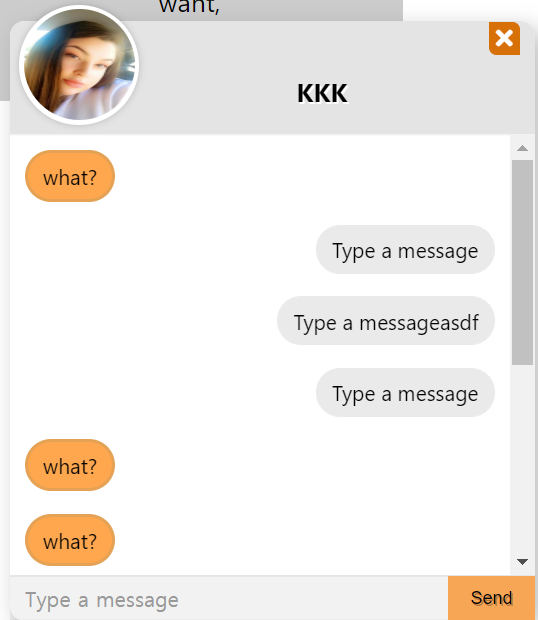
Pet owners can entrust their pet to a pet sitter for free. Owners make an entrust application, so sitters can know about the owner's pet and owner. After applying, the sitter can check the owner's review and personal ‘show off page’, and the others.

d. Raise application



Pet sitter can send a request for raising his desirable pet to the pet owner by filling out this Raise application on the choosing page.

e. chat room



Chat is a floating page of a website, so its shape is quite similar to phone UI. Users can chat with sitters who are sitting with their pet, or just a private friend.

1. set server setting for database and web application

Setting server was done in week2. However the server spec we used in amazon ec2 is a free-tier version, so sometimes, errors occurred.

1. Implement node.js server code for following front pages

Node Js server code proceeded throughout week 3, 4,5, 6 as time schedule, but node js server code for community system was not completed before midterm demo. Also, code restructuring was necessary for stable backend infrastructure. So we self-evaluated as 80% of the overall server code process.

1. logical design of database schema & MYSQL physical design

designed database and inserted into MySQL workbench.

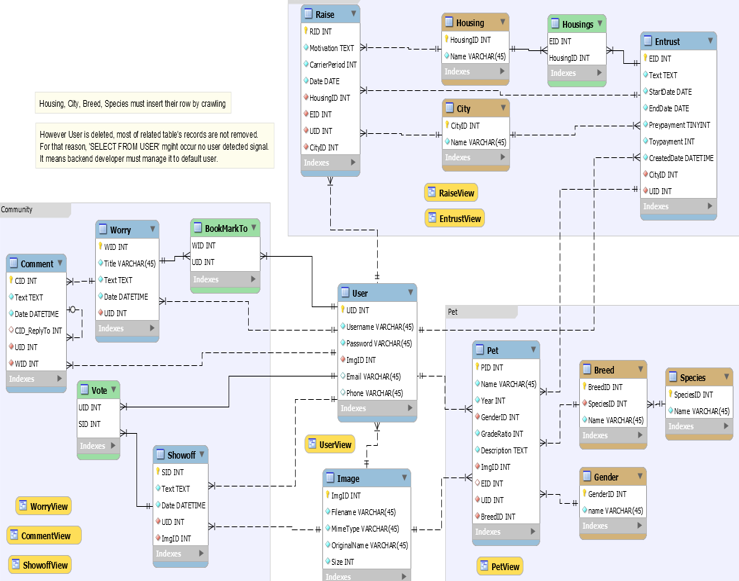
Below image is a figure of Pet-Metting’s physical database.

In the design process, we separated our database into 4 parts; User, Pet, Entrust, and Community.

The tables of blue color are the essential table which we call it usually.

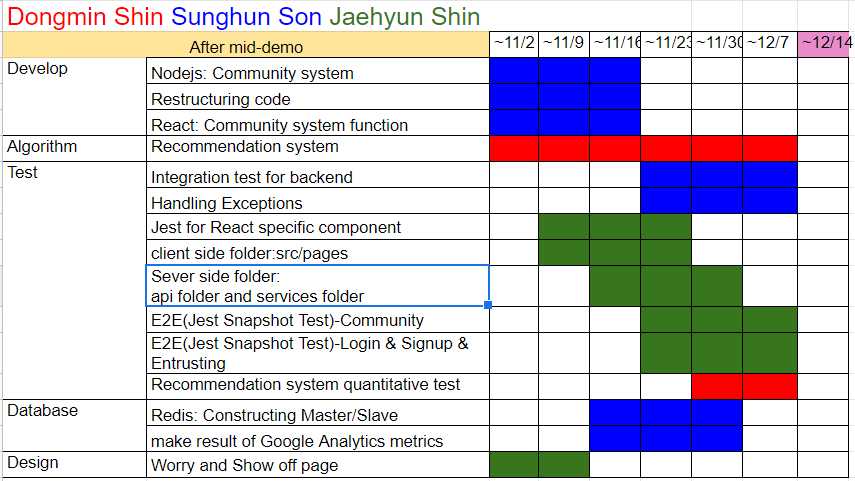
The green color is the weak relationship table that you saw above.

And orange one, the administrator has a responsibility to fill it using crawling or DB schema

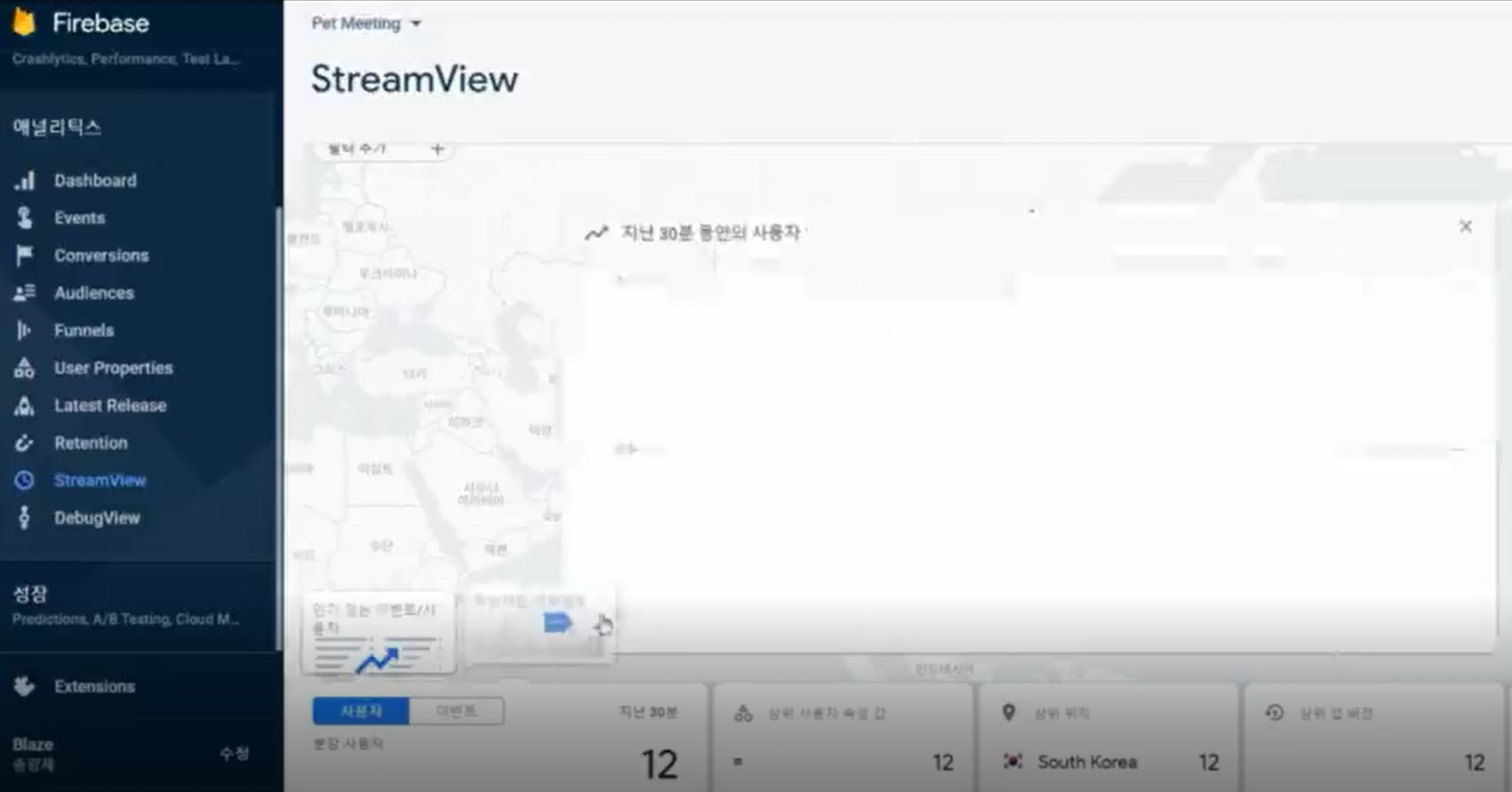


Above 4 goals are the goals proposed before the midterm demo.

2) after midterm demo until final

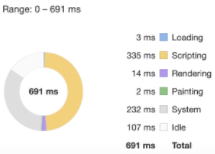


1. Attach Google Analytics to React



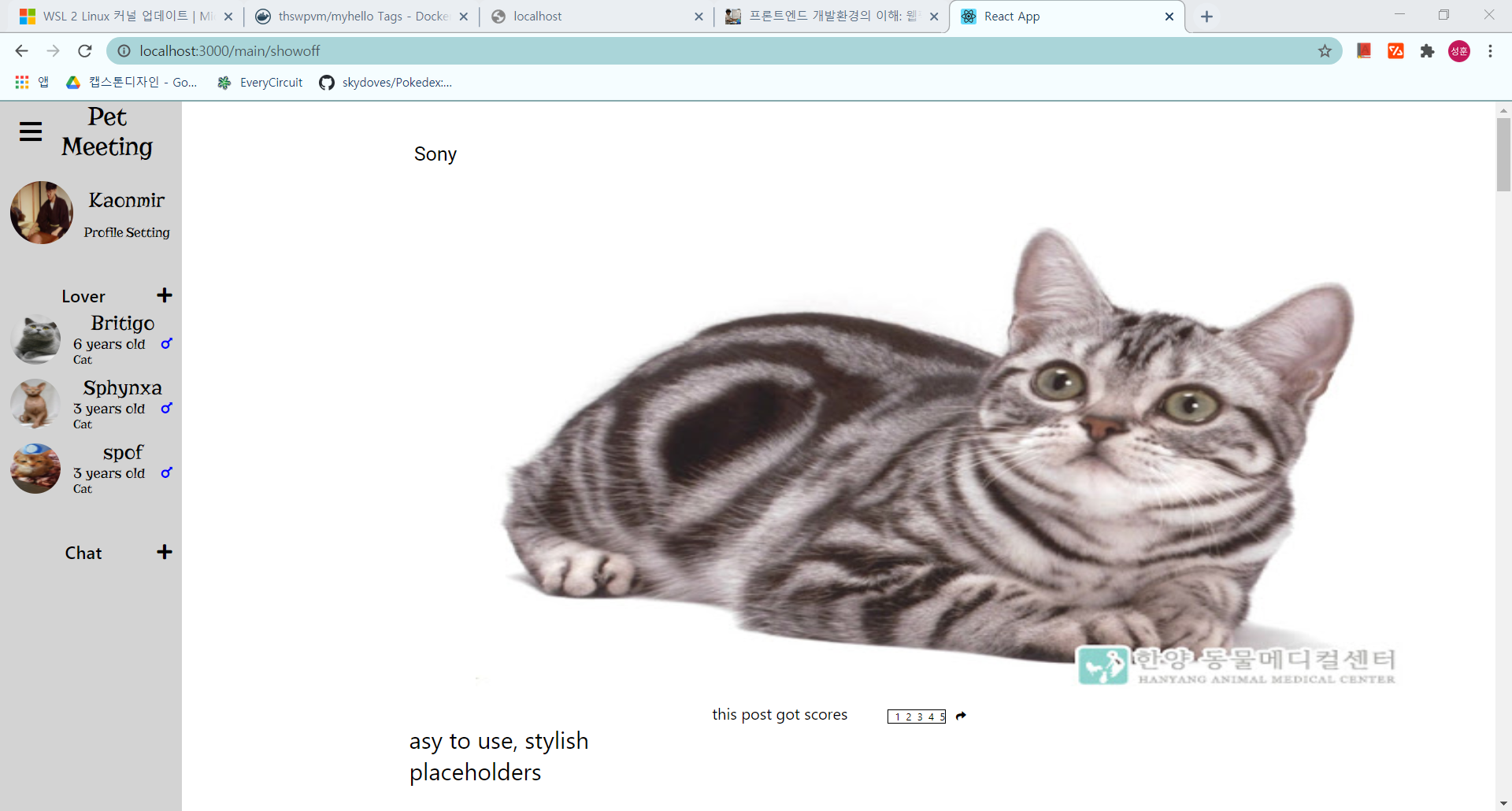
Key performance Indicator is the analytics that measure web traffic and visitor behavior, which is crucial to measure the web site’s efficiency. Following 4 metrics belong to KPI.

1. (Real) 12 daily users > (Goal) 10 daily users. (pass)
2. Pages per session more than 3. (pass)
3. We didn’t find out the bounce rate in google analytics. (fail)
4. average page load time is less than 5 seconds. (pass)



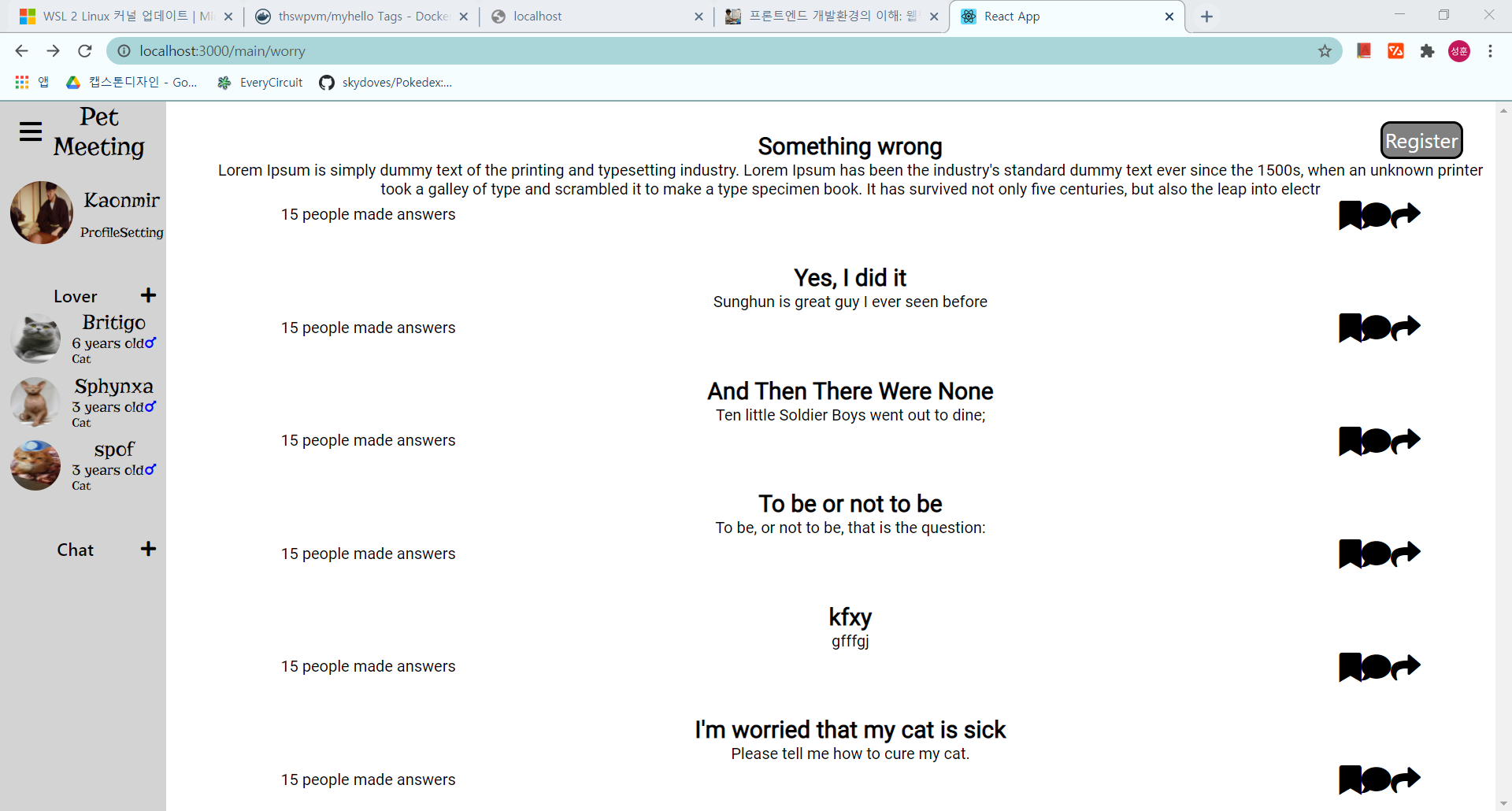
2) implement community service

1. show off page



Users can show off their pets through this page. Like instagram, users can upload their pictures of pets, and some descriptions. Also users can save, share, and recommend each post to others. Recommender systems make users find the best suitable post by learning their patterns.

1. worry page



If users have trouble with raising a pet, they can ask the community to solve it. Users can also answer each problem. They also vote, share each post. Especially, this page is anonymous for all.

3) Implement Recommendation system

We made pet card recommendation system on choosing page

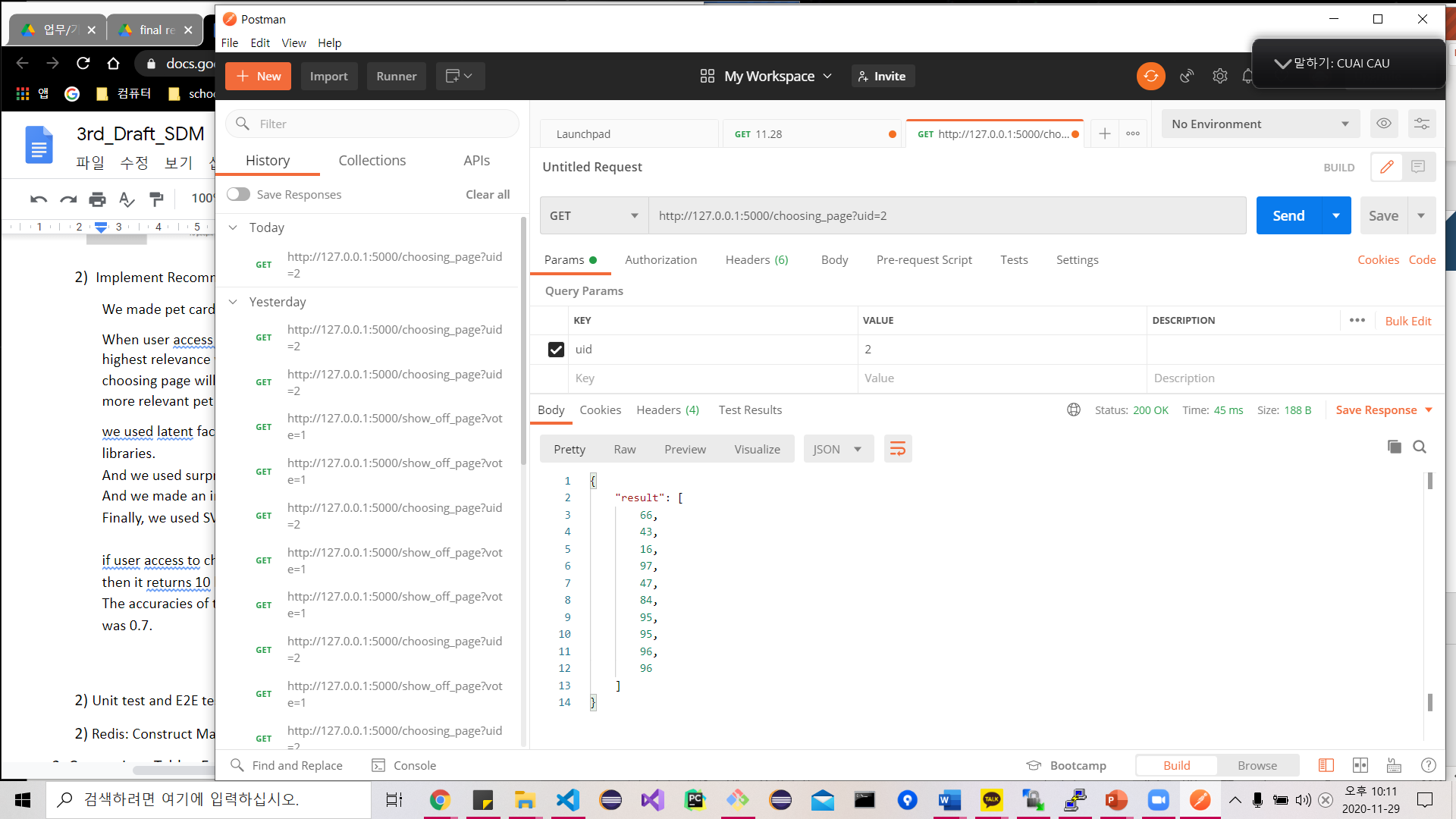
When a user accesses a choosing page, it functions to produce 10 pet’s id lists that have highest relevance with that user’s id considering previous all rating data, and then choosing a page will be filled with pet posts of those pet id lists. Then, users can choose more relevant pet posts.

We used a latent factor based collaborative filtering system, and surprise, pymysql libraries.

And we used surprise, pymysql libraries.

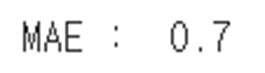
And we made an independent WAS of this ML system using the Flask framework.

Finally, we used SVD++ algorithm for better recommendation accuracy.



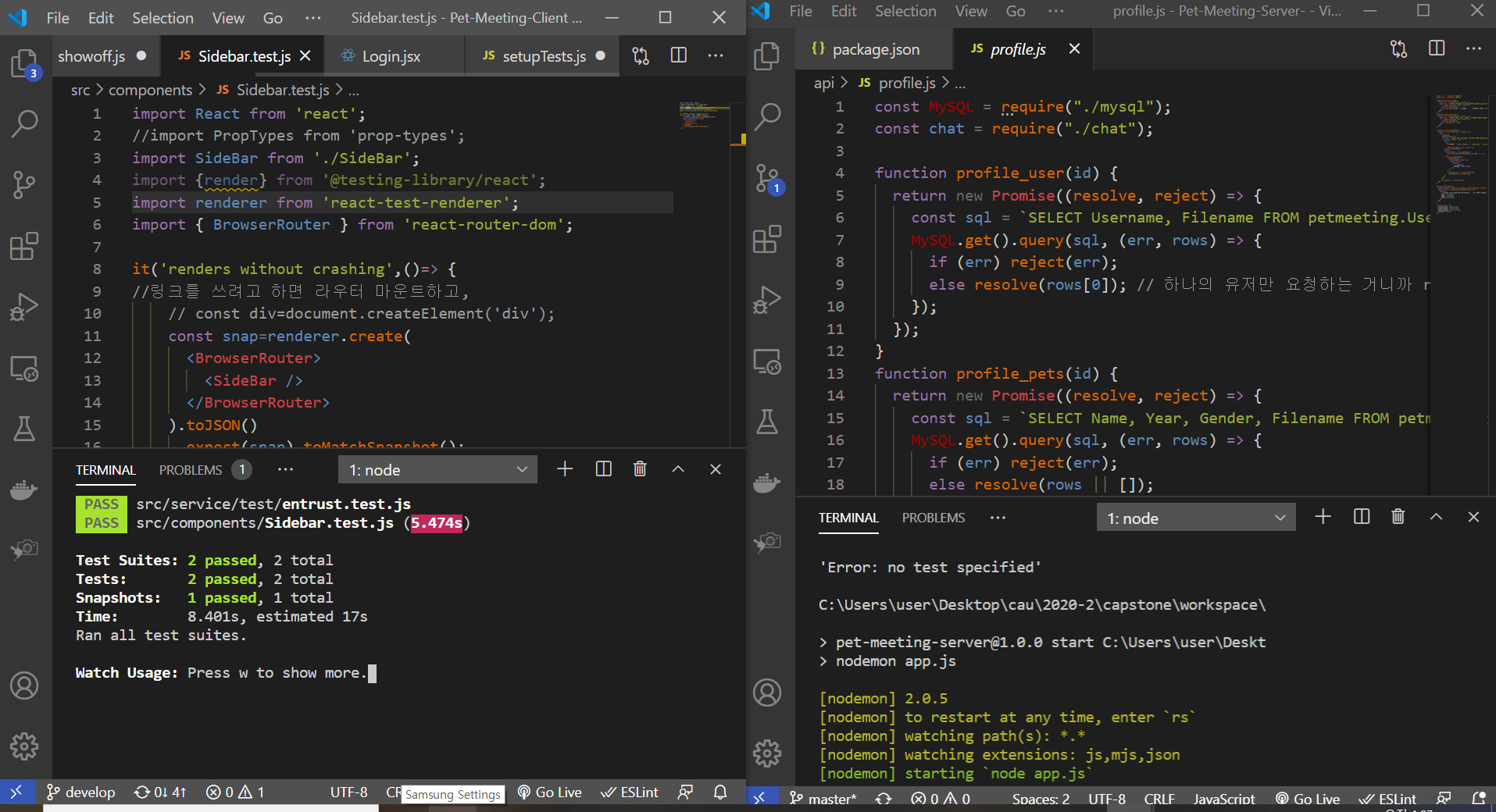
if the user accesses the choosing page, then his user\_id is given to the ML model.

then it returns the 10 highest related pet id lists based on a preference dataset.



The accuracies of this were not more than 0.9. Concretely, RMSE was 0.9, and MAE was 0.7.

4) Unit test and E2E test for service



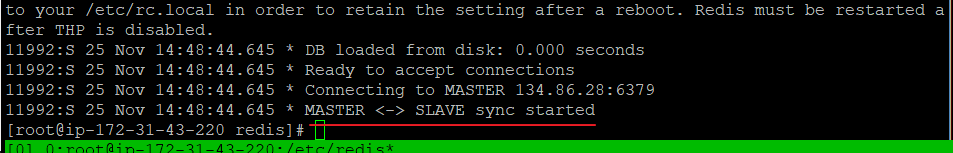
Unit Test of Client side and Server side passed.

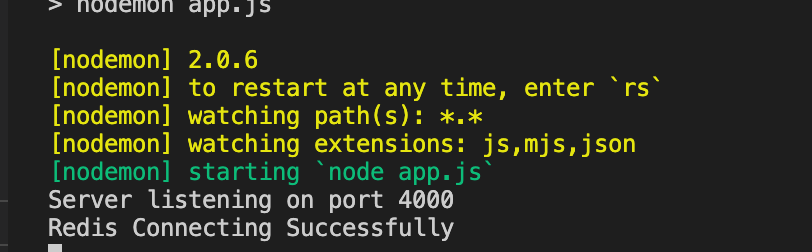
-Client side: entrust.test.js

-Server side: function testing UserLogin

End to End test for client side passed. ex) Sidebar.test.js

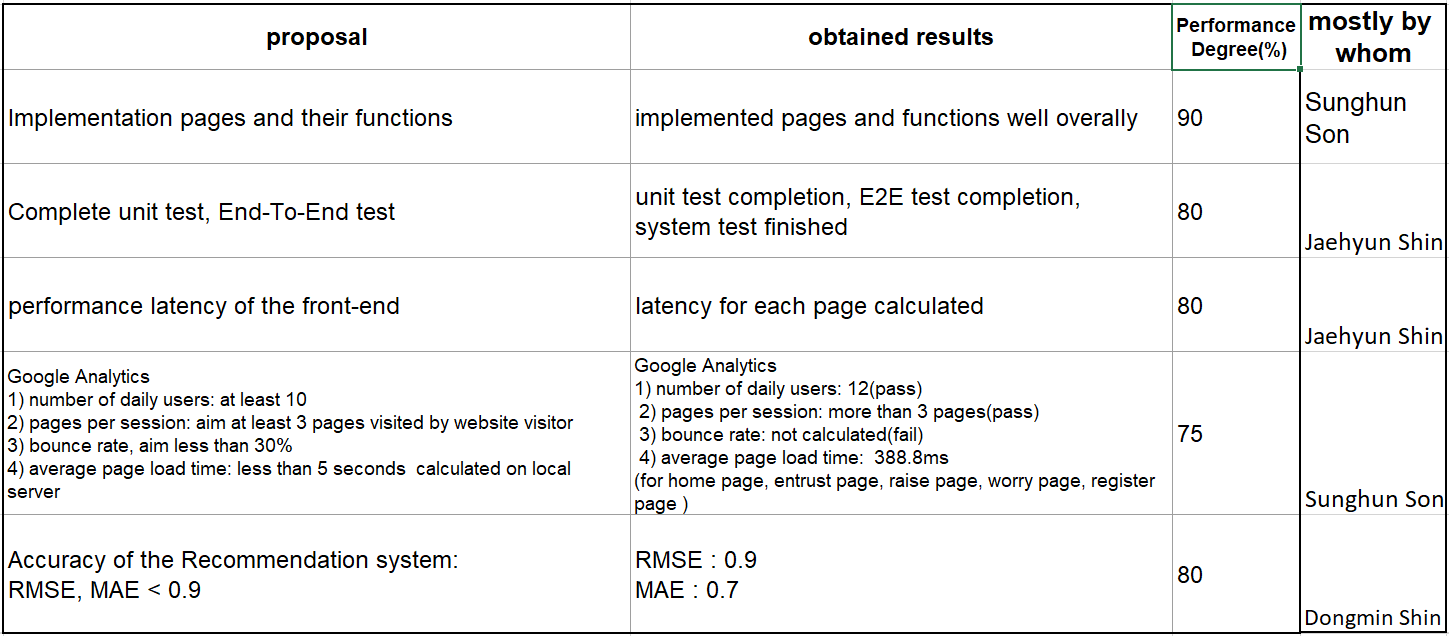
5) Redis: Construct Master/Slave structure





### **2. Comparison Table**

Expected goal and achieved performance expressed with numeric value



Caution) Performance latency of the front-end and average page load time are the same concept.

We expected google analytics to calculate page load time, however after numerous research of web development guide, we figured out chrome devtool is much better and frontend latency speed can be referred to as page load time.

* 1. Skipped profile page, designed low-quality css pages.
  2. Unit test executed in server and client code, but consists of simple logic tests.

Test coverage is really important in web development, but our test code’s covers only simple logic.

* 1. Our team didn’t aim for a specific number for the latency speed of the page in the proposal. . We executed the test, however, the average rendering speed is 388.8ms. Though there are many image files to be rendered and server spec is not good, this is not very high speed in objective perspective.
  2. Model’s RMSE(Root Mean Square Error) should be less than 0.9, but the machine learning model’s accuracy is equal to 0.9. In case of mean absolute error, mean absolute error is less than 0.9.

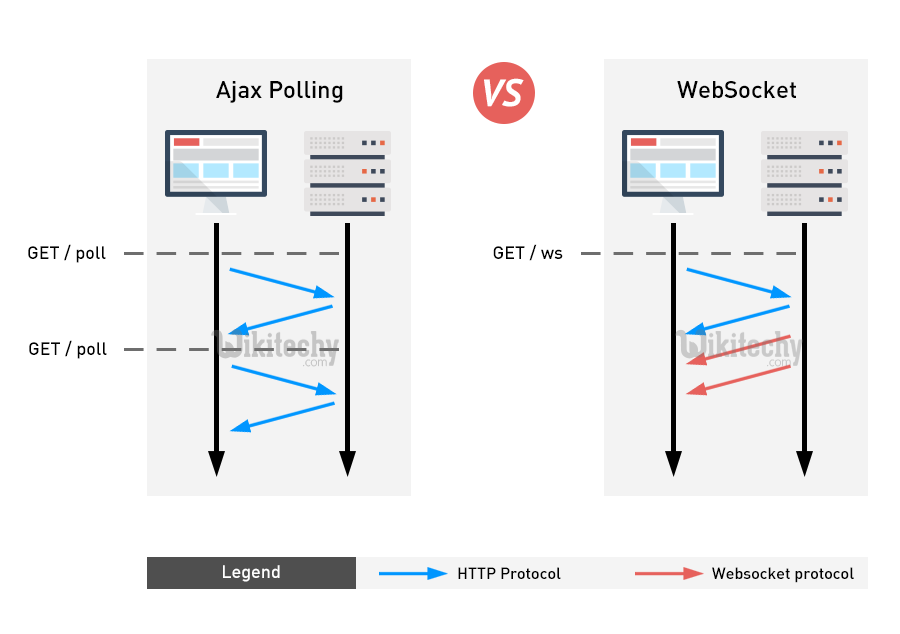
### **3. Member Role**

|  |  |
| --- | --- |
| Sunghun Son | -Frontend React code  -Database  -Devops |
| Dongmin Shin | -Frontend react code  -Design Pages with Figma  -Recommendation algorithm |
| Jaehyun Shin | --Backend node.js code  - Administer Database  - End to End Test using Jest Snapshot  - Unit Test using Jest |

# **4. Theoretical basis**

### 1. Web Socket

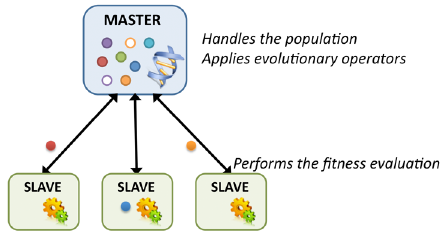
Web Socket technology is necessary for real-time chatting systems on our Pet-Meeting service. Web socket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. WebSocket and HTTP are all located at layer 5 in the OSI model and depend on TCP at layer 4. However WebSocket, defined on RFC 6455, is designed to work over HTTP ports 443 and 80 as well as to support HTTP proxies and intermediaries.



Socket.io enables real-time, bidirectional and event-based communication. It works on every platform, browser or device, focusing equally on reliability and speed. Socket.IO is not a WebSocket implementation, although it can communicate with each other. Socket.IO provides powerful reliability which can support proxies and load balancers. And if there is some fault while connecting, Socket.IO makes auto-reconnection and periodic disconnection detection is supported. And also to create separation of concerns, it allows us to create several namespaces to separate communication channels but will share the same underlying connection. Web socket technology is necessary for real-time chatting systems on our pet platform.

### **2. Master and Slave**

Master and slave or Master/Slave is a model of asymmetric communication or control where one device or process controls one or more other devices or processes and serves as their communication hub.  We concern only master/slave models of Database, especially MySQL and Redis. Master/Slave is one solution of replication to prevent some unexpected faults of the data center. Master keeps the replica updated by sending a stream of commands to the replica, in order to replicate the effects on the dataset happening in the master side. When link between the master and the replica breaks, for network issues or because a timeout is sensed in the master or the replica, the replica reconnects and attempts to proceed with a partial resynchronization. When a partial resynchronization is not possible, the replica will ask for a full resynchronization. This will involve a more complex process in which the master needs to create a snapshot of all its data, send it to the replica, and then continue sending the stream of commands as the dataset changes



**Master/Slave model structure**

### **3. Recommender System**

Recommender systems are the information filtering systems that find the proper items for each specific user using the rating or preference of users to an item.

Collaborative filtering predicts the interests of a user by collecting preferences or taste information from many users. This filtering can cause cold start problems for new users or new items as there will be insufficient data on these new entries to work accurately. To make appropriate recommendations, the system must first learn the user’s preferences by analyzing past voting or rating activities.

### 4. Jest test

Jest is a resting framework with a focus on simplicity. It supports unit testing and mocking. Jest test is used to test a component in isolation and verify the DOM output of a component.

# **5. Open source and Github Activity**

### **1. Open Source**

1. Front-end : React JS https://ko.reactjs.org/
2. Back-end : Node JS https://nodejs.org/ko/
3. DB : Mysql, Redis

MySQL https://www.mysql.com/

Redis https://redis.io/

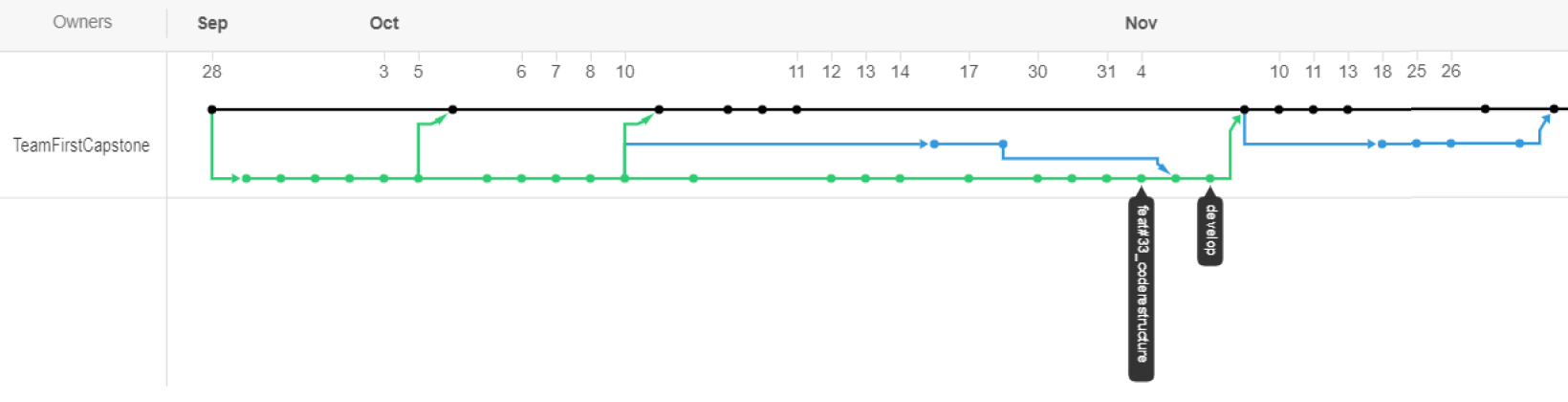
1. Cloud : AWS ec2 <https://github.com/aws/aws-codedeploy-agent>
2. Test : JEST https://jestjs.io/
3. ML: scikit-learn, surprise

### **2. Github Activity**

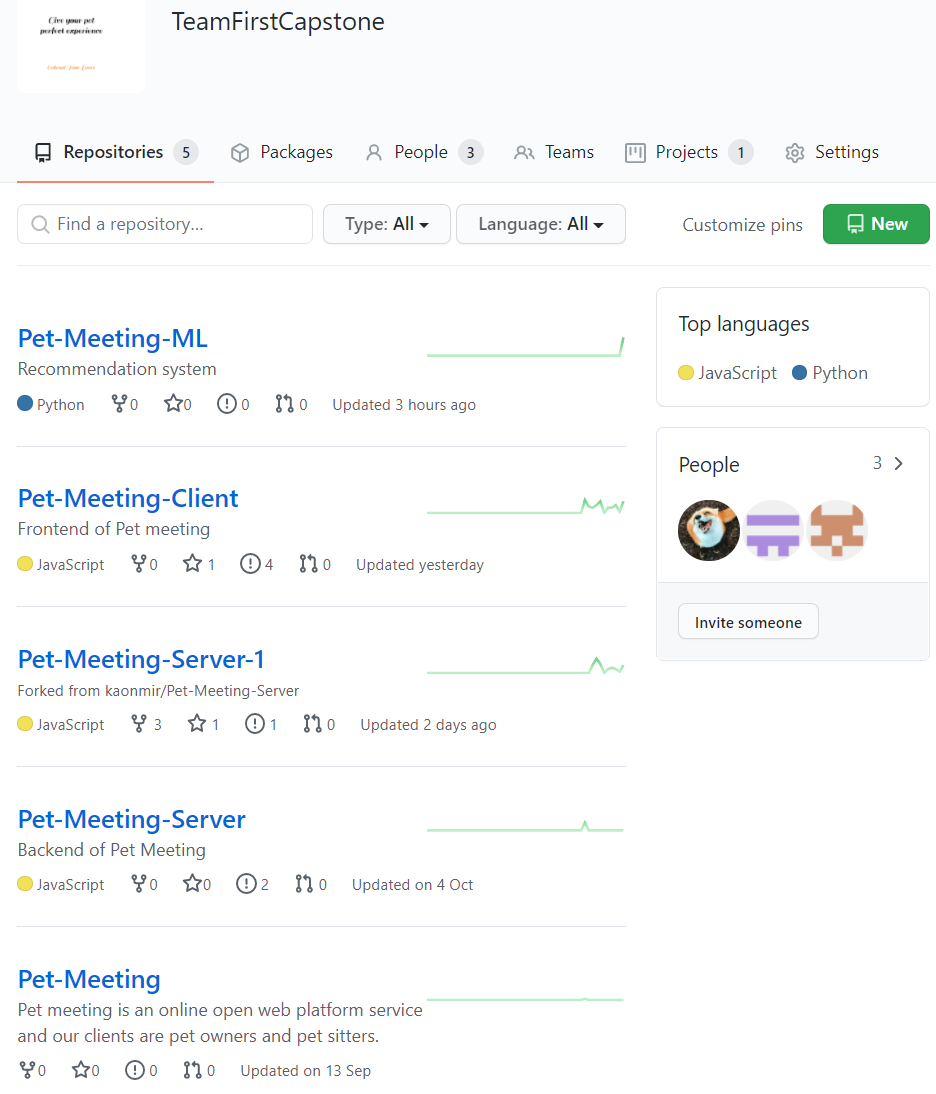
We use github as a version control and collaboration tool. We used team organization

TeamFirstCapstone.

* Github Repository: [https://github.com/TeamFirstCapstone /Capstone-Pet-Meeting](https://github.com/TeamFirstCapstone%20/Capstone-Pet-Meeting)
* Issue Page: ToDo, Doing, Done Label used as scheduler
* For effective project management, we plan to merge personal development branches into the branch named "Dev" through pull requests, and the master branch keeps a stable serviceable project.
* Planning request for feedback from developers who are expertise in software development such as one working in Google or Microsoft.



Above image is network branch image of github server repository. We used squash merge to make network branch clean.



Jaehyun Shin’s github ID: harrywinks

Sunghun Son’s github ID: kaonmir

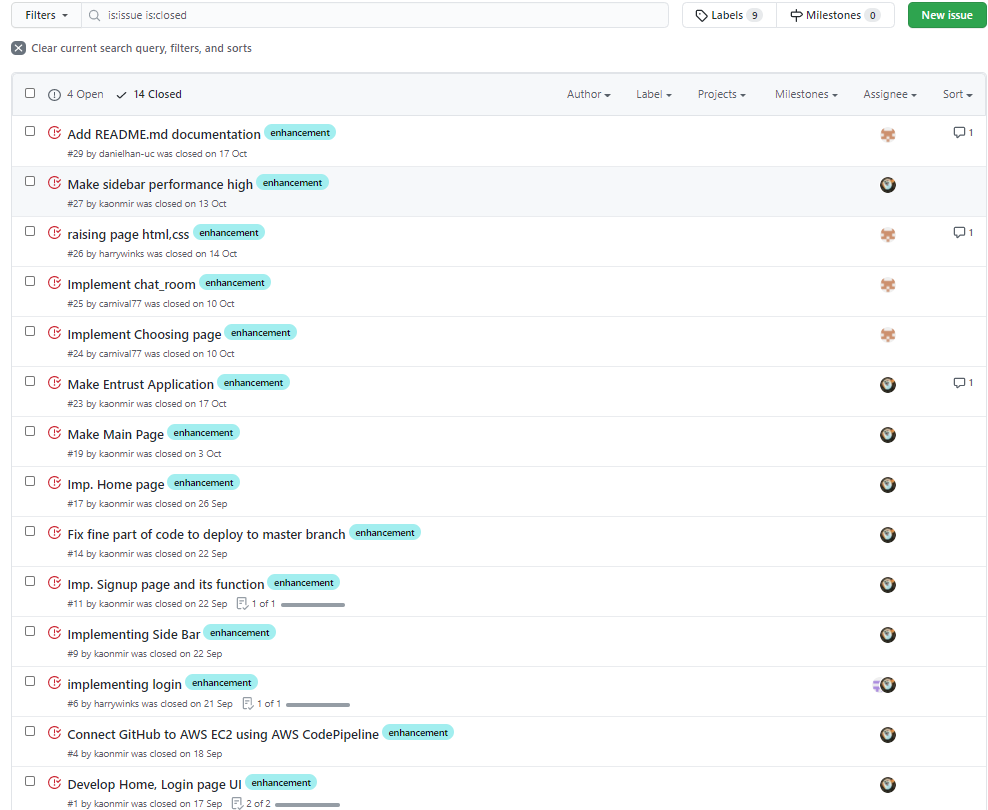
Dongmin Shin’s github ID: carnival77

Above is capture of team First’s github repository.

1. Pet-Meeting-ML is repository for recommender system. This is connected to web server with flask.

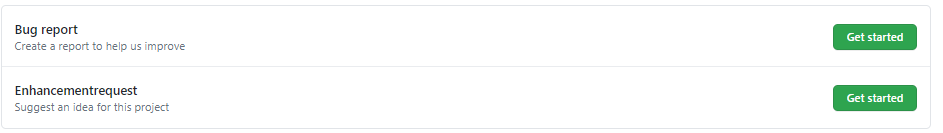
2. Pet-Meeting-Client is repository for client part of web. Deploy version is master branch .

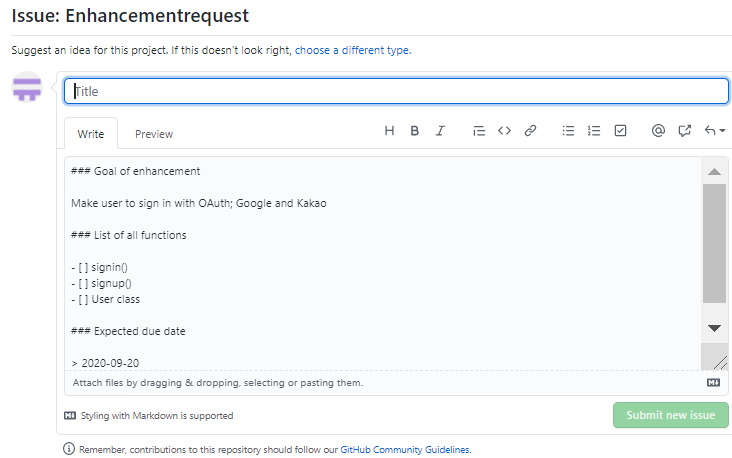
3. Pet-Meeting-Server-1 is repository for server side. Deploy version is master branch.



### **We used github issue to communicate between teammates.**

We made template for issue dashboard as below, to development consistent .





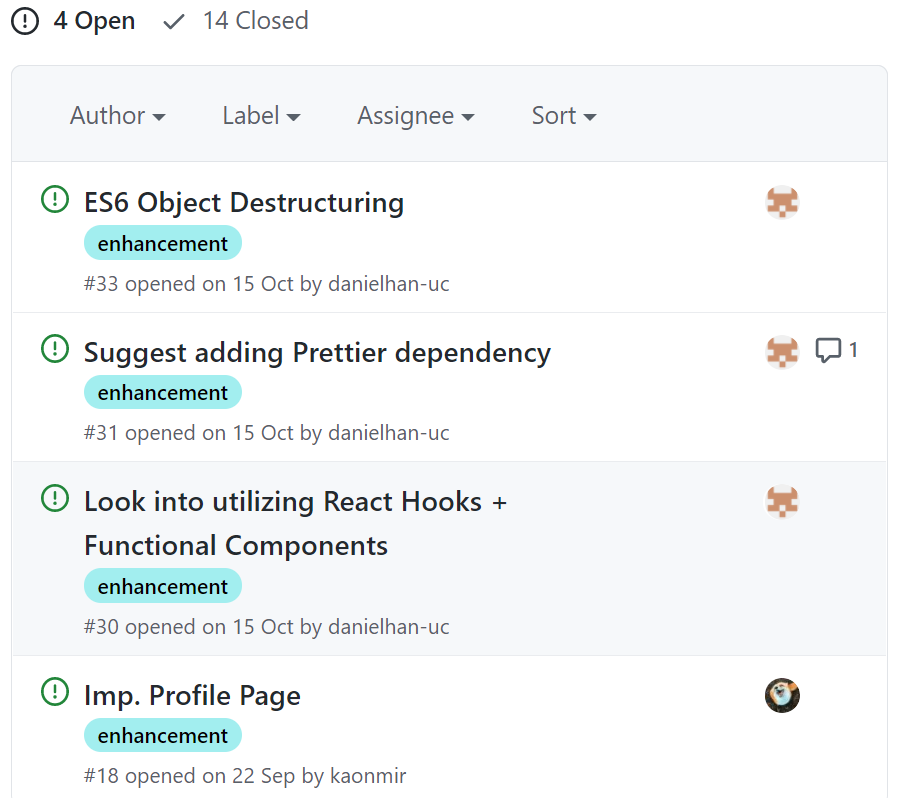
### **2. Communication with other world(United States Software Engineer)**

Two software engineers Daniel Han and David Lim left comments each on client side repository and server side repository. Comment Links are attached below.

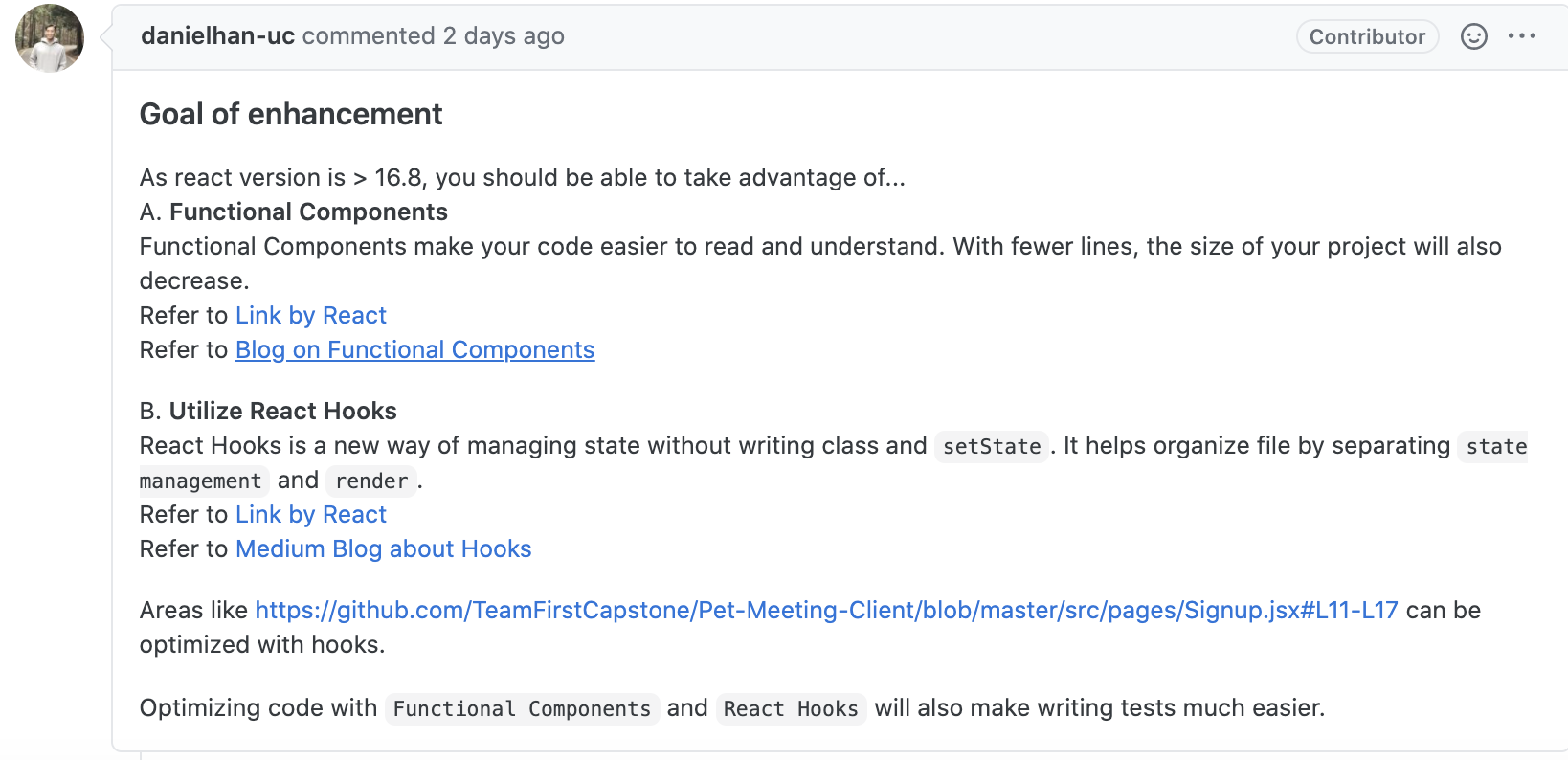
**1st comment**

Daniel Han, in New York, working in UrbanCompass as web developer, left feedback in <https://github.com/TeamFirstCapstone/Pet-Meeting-Client/issues/> .

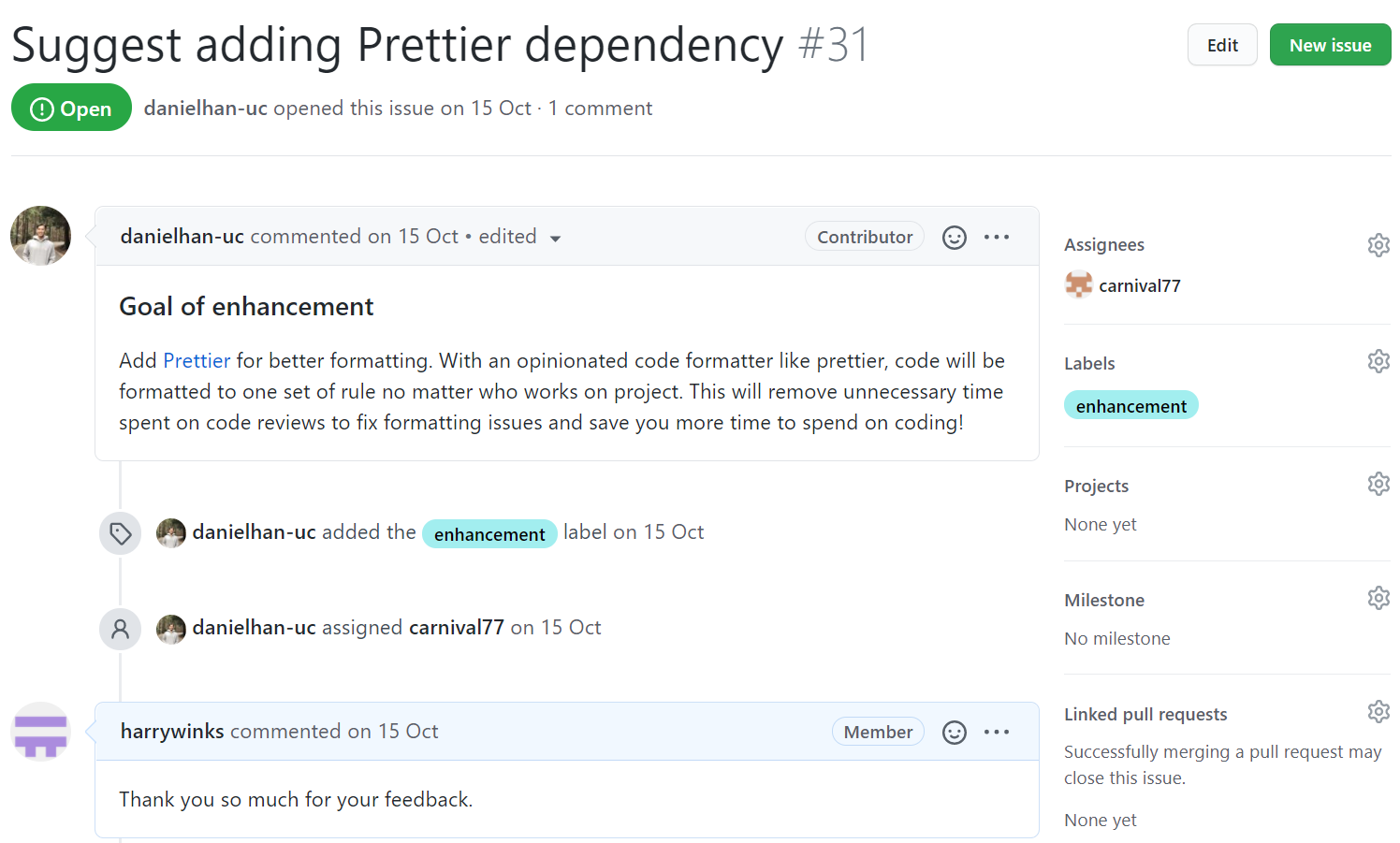
He used github issues to make comments to us.

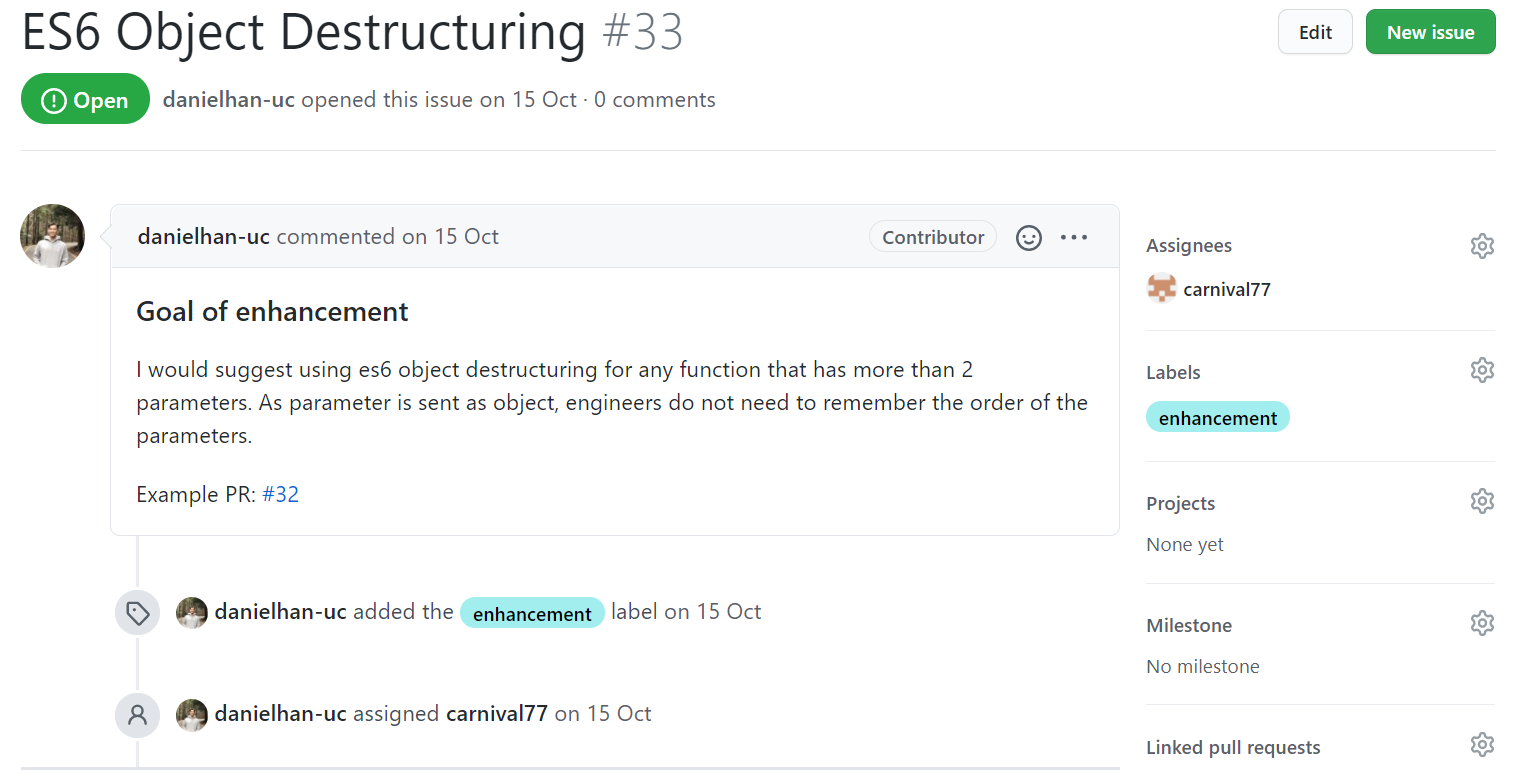


1) ISSUE 30 <https://github.com/TeamFirstCapstone/Pet-Meeting-Client/issues/30>



2) ISSUE 31 https://github.com/TeamFirstCapstone/Pet-Meeting-Client/issues/31



3) ISSUE 33 https://github.com/TeamFirstCapstone/Pet-Meeting-Client/issues/33 

We used React.js as client-side code, so Han left the comment on React hooks and explained the necessity of functional components to make readable React code. So after his feedback, Sunghun tried to make changes in the code following his advice. Also, following Han’s advice, while our team was trying to deploy the client code, we changed commonJS denotation to ES6 denotation.

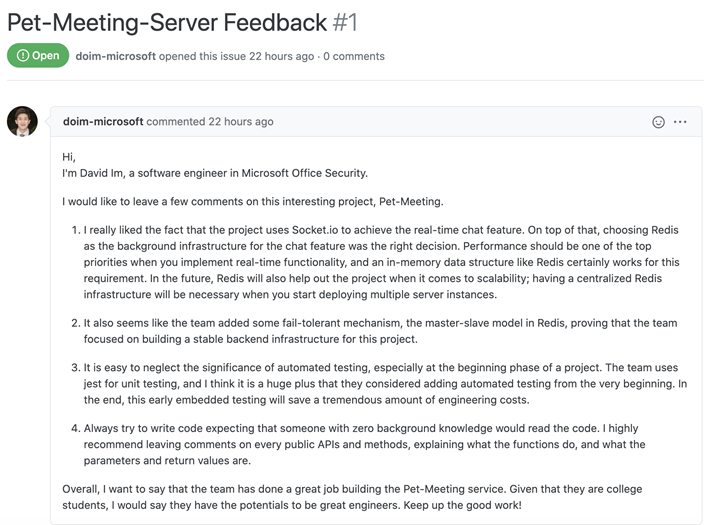


**2nd comment**

David Lim, software engineer working in Microsoft 365 security(Seatle) , left positive feedbacks on our project.

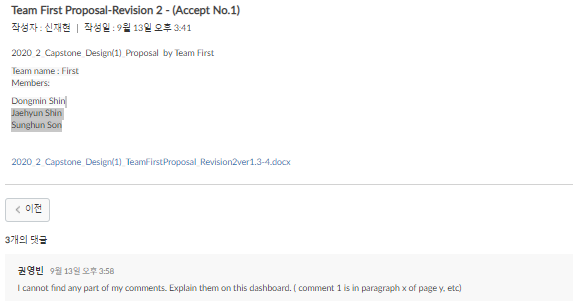
<https://github.com/TeamFirstCapstone/Pet-Meeting-Server-1/issues/1>

His point was that real-time chatting system and using Redis as background infrastructure for the chat feature was the right decision. Also, he commented on the automated test of our project using Jest.



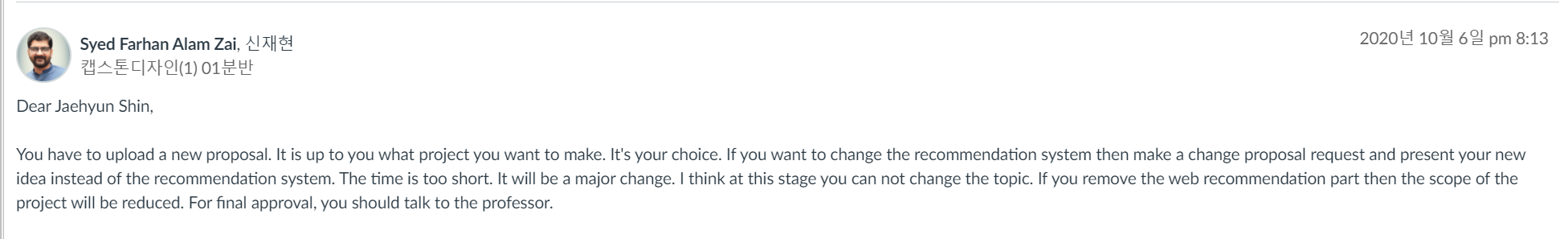
# **Comment & Resolution**

### 1. Comment about proposal



After this feedback from professor, we always denoted the changed version with red letters.

### 2. Comment about recommender system



We decided to put recommender system inside our web platform to differentiate with other pet related webs after JaeHyun Shin got answer from Teach Assistant about the risk of removing recommender system.

# **7. Conclusion**

### 1. Lessons

1. latent based collaborative filtering system implementation
2. Automated testing programming
3. Background infrastructure implementation using Redis

### 2. Unsatisfactory points

1. Skipped profile page, designed low-quality css pages.

2. RMSE is not less than 0.9

### 3. Future plan

1. Improve ML algorithm performance

2. Implement map function

### 4. Meaning of project

1. Satisfying surging demand for pets on these days.

2. Entrusting pet for free is very unique concept and also recommender system makes this web’s user convenience higher.