

1. Specification

- **Description**

Our website is a platform where massive food blogs of users are collected. In every blog, the author will describe his or her true feeling about a restaurant. Therefore, our website can provide users with a large amount of real feedback and offer relative and timely information when users are choosing restaurants. In addition, our website gives every user the opportunity to create their own food blogs when meeting delicious food or special experience for the purpose of sharing or just recording.

So if you are a food lover, just follow our website! Our web address is <http://47.100.46.122/>

- **Define users/target audience**

People who loves food with high quality

People who have an intensive creating desire of food blogs

People who want to explore more restaurants

People who suffer from choice difficulties of restaurants

People who specialized in restaurant market tendency.

- **Product functions**

Without login

1 Users can search and check food blogs by searching key words.

2 User can sort their search results combined with category and average cost.

3 User can see the location distribution of people who also visit this website.

With login

4 Users can add/remove food blogs to his or her favorite list.

5 Users can create food blogs.

6 Users can modify and delete their own articles.

7 Users can log out.

- **Functional/Nonfunctional requirements**

Functional requirements

1 login, register and logout

2 search by key words

3 Create, delete and modify their own articles by users

4 Favorite and not favorite food blogs

5 Check the favorite list

6 Sort the results by category and average cost.

Nonfunctional requirements

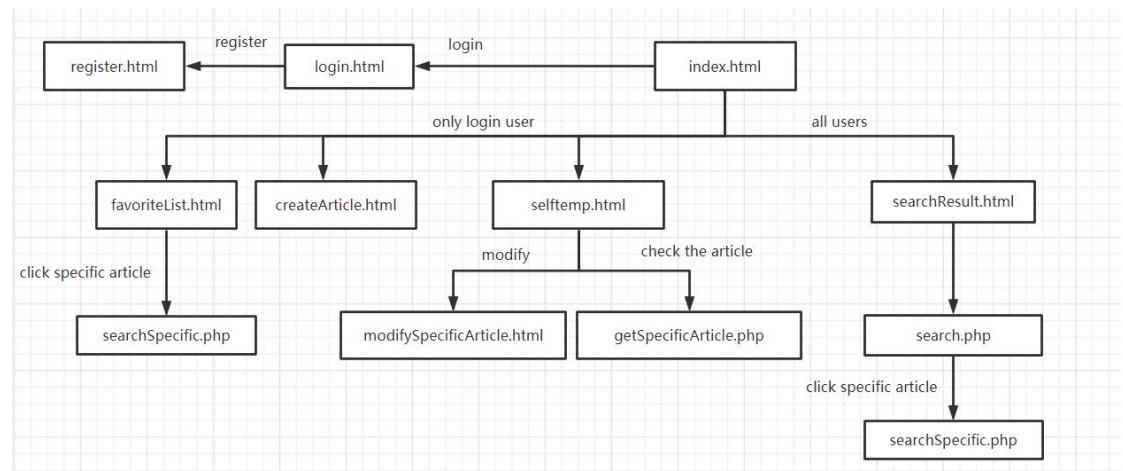
1 Easy extensible for further functions

2 The response time of our website should be within 200ms.

- 3 Number of concurrent users should be at least 50.
- 4 The fully web on load time should within 3s.

2. Structure

- Explain the structure of your web site/app by using a sitemap and wireframes, describe them



When users first open our website, the **index.html** will be shown first. Firstly, users can click login icon to log with their own account on the **login.html**. If they are not registered, they can click the register link on the login.html to register on the **register.html**. Secondly, users can search on the index.html and get the results on **searchResult.html**. When users click the article's name, they can see the whole content.

When users already log in, they can click the create icon on the index.html to create articles on the **createArticle.html**. They can also check or modify their favorite list after clicking the name of the article. Users can also see their self articles list on the **selftemp.html**. When users on the selftemp.html, they can click the modify button to be directed to **modifySpecificArticle.html** to modify all the content of their articles including delete function. They can also click the title of articles to check the content of every article. Furthermore, they can click the delete button on the **selftemp.html** to delete the article completely.

3. Design

- Describe your design decisions (e.g. reason by referencing usability heuristics, requirements or target groups)

Firstly, we think designing a blog website matches with the requirements which combined with front end and back end. We ourselves always use blog websites like CSDN in China when we meet some problems in programming, so we think blog websites are good place for people to obtain useful information. Therefore,

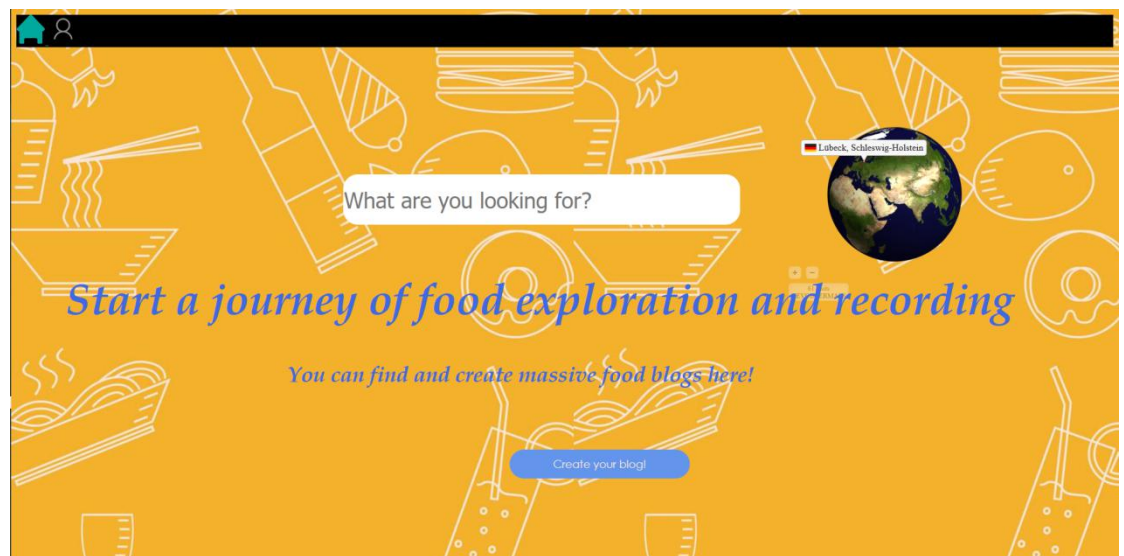
we decide also to develop a blog website to facilitate people.

Secondly, we asked most of the people around us, all of them love food. So it seems like design a food blog website is a nice choice. When taking usability test later, many of our classmates can be the testers.

Last but not least, by referencing usability heuristics, we use some icons to replace specific words, making the interface more concise and conspicuous. By the possible errors, we also add many hints in our website design, thus, making them recover from errors more easily.

- **Include screenshots of final app/website and describe them**

1 Home page before login



In this page before log in, users can search by keywords in the search area.

Users can also click the icon on top left to log in or register.

When user click create your blog button, it will show alert: Please log in.

2 Login & register pages

<p style="text-align: center;">Hello!</p> <p>Hope you find joyiness and satisfication here!</p>	<p>Welcome to FOOD BLOG!Please register.</p> <p>Define your nickname <input type="text"/></p> <p>Set your password <input type="password"/> <small>*Must contain digit,letter and symbols(-!@#5%^&*),length from 8-16</small></p> <p>Please type in the password again <input type="password"/></p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"><button style="background-color: #4a86e8; color: white; padding: 5px 10px; border-radius: 5px;">Register</button><button style="background-color: #4a86e8; color: white; padding: 5px 10px; border-radius: 5px;">Back to homepage</button></div>
--	--

Hello!

Hope you find
joy and
satisfaction
here!

Welcome to FOOD BLOG! Please register.

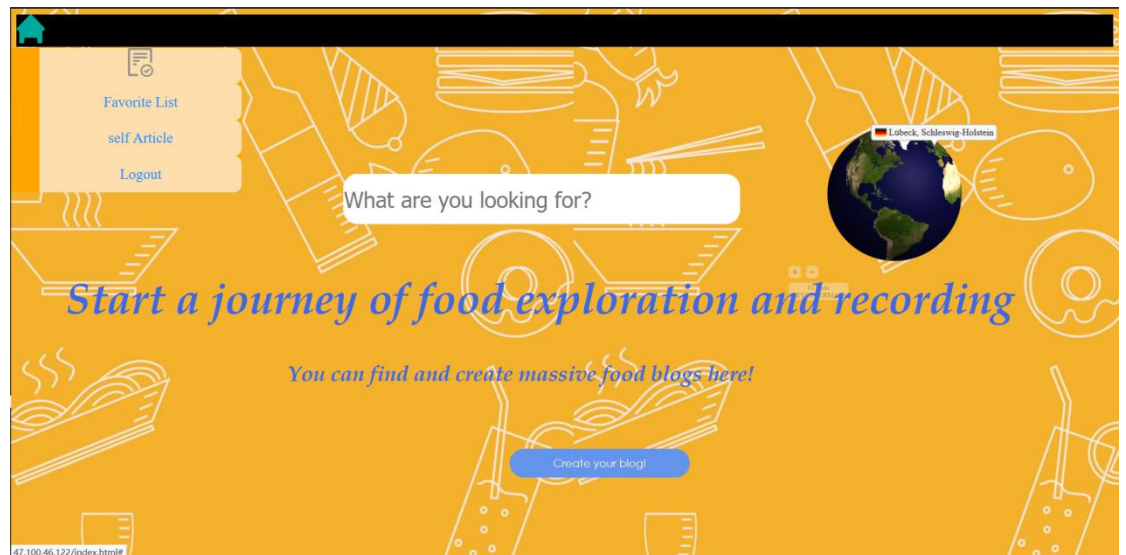
Define your nickname

Set your password *Must contain digit, letter and symbols (~!@#%&^&*), length from 8-16

Please type in the password again

[Register](#) [Back to homepage](#)

3 Home page after login



After successful login, users can see their status turn to log in on the top left, and when users hover around this icon, they can see three more functions: check favorite list, check self articles and logout. Also, when users click create your blog button, users can switch to create article page successfully. When users input the keywords on the search area and tap enter, they can see the search result.

4 Create article page

6 Detail blog page



Users will be guided to this page when they click the titles of search results. In this page, they can see all detailed content including description, category, cost, rating and so on. In addition, they can also add and remove this blog to/from favorite list. P.s. if users want to add into favorite list without log in, website will also hint: please log in.

7 Favorite list page



Users will be guided to this page when they click the favorite list on the main page. Here, users can see the titles of all the articles in their favorite list. Also, they can click the title to see detailed content.

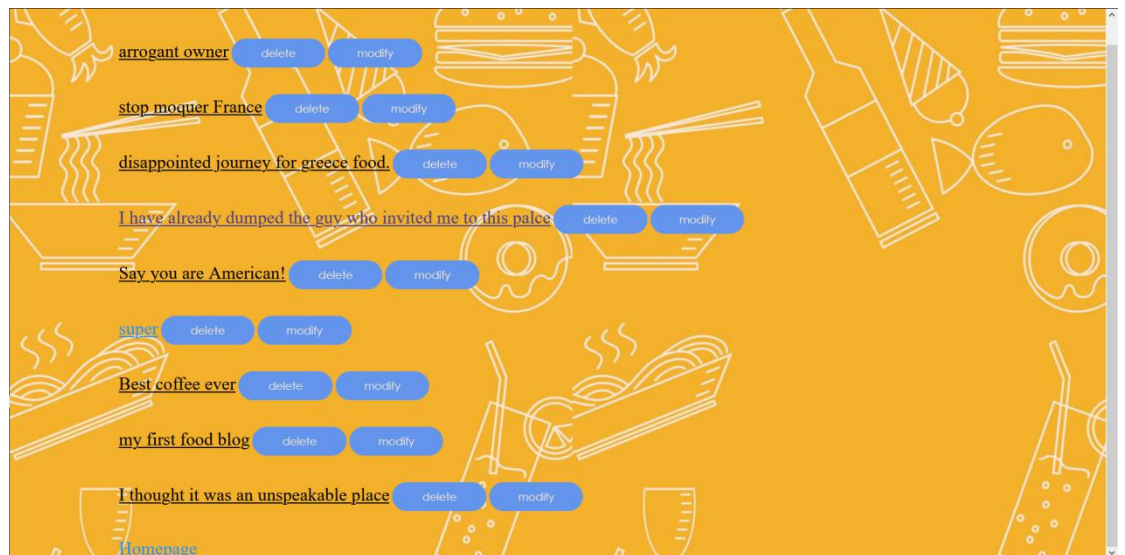
8 Detailed favorite page

Title: I have already dumped the guy who invited me to this place
 Why did some body think that a court theme restaurant would be a good idea, waiters all ware that ridiculous wig
 Category:
[KoreanCuisine](#)
 Average Cost:
 37
 Restaurant and its location:
[Court Yard in 12321,Seoul,Korean](#)
 Rating:
 2

[Remove from favourite list](#) [Back to favorite list](#)

Users will be guided to this page when they click the title of favorite list page.
 Here users can remove this article from favorite list.

9 Self articles page



arrogant owner delete modify

stop moquer France delete modify

disappointed journey for greece food. delete modify

I have already dumped the guy who invited me to this place delete modify

Say you are American! delete modify

super delete modify

Best coffee ever delete modify

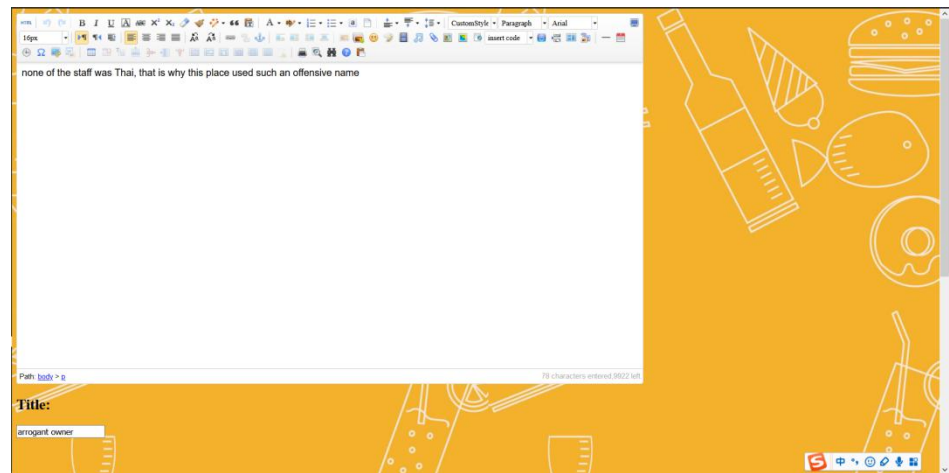
my first food blog delete modify

I thought it was an unspeakable place delete modify

[Homepage](#)

Users will be guided to this page when they click the self articles on the main page. They can see all the blogs they wrote. In addition, when click delete button, users can delete this article, when clicking modify button, users will be guided to the modify page.

10Modifypage



Please input the restaurant and the location:

Please choose the category:

Input average cost per person:

Please rate the restaurant:

Users will be guided to this page when click the modify button on Self articles page. Users can modify and delete their articles here.

4. Implementation

- **Describe structure of your website/application (code-wise, classes, explain architectural decisions) and structure of DB (ER model)**

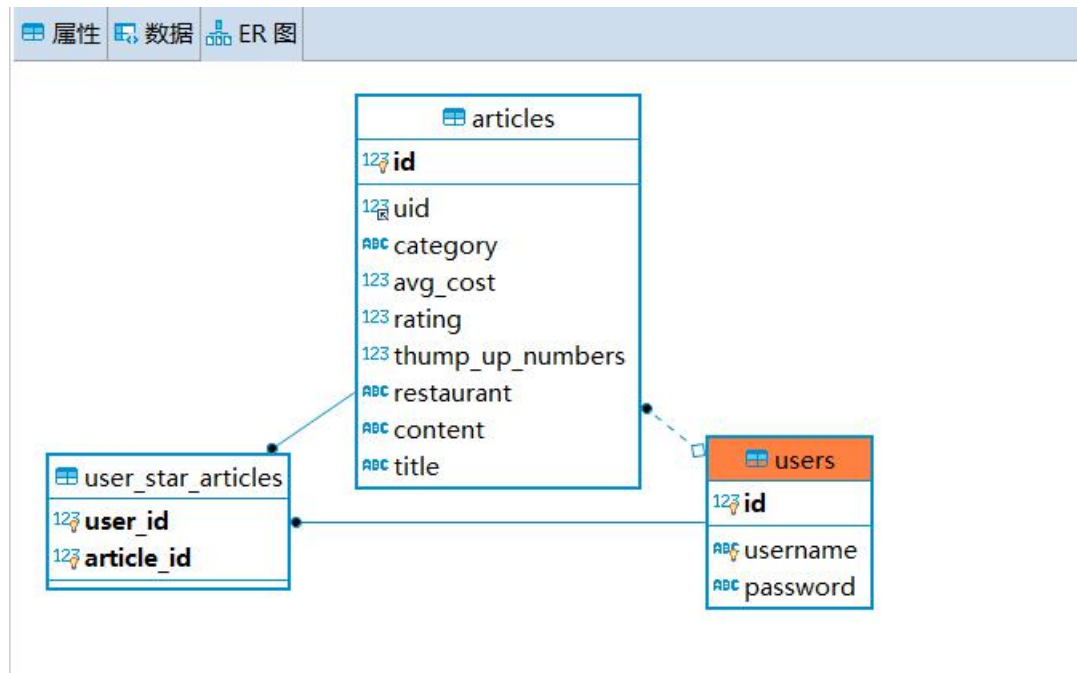
Object oriented: When we connect to the mysql, we use mysqli in the php, which is encapsulate the connection to the database which is very convenient.

Because we use ueditor(rich text editor) in the program. There're several reasons to use Ajax to post the requests to the backend(PHP), than just use<form> and submit in the html to call the php:

1. We use localStorage to store some information of user, such as

selftemp.html and searchResult.html, login.html. Index.html is controlled by main.js.

1. After login.html is open, users can log in, the value will be got by login.js and be validated by login.php. If users do not register before, users can open register.html. We also create getUserId.php here to facilitate the further querying, because we store user_id in the articles table
 - 1.1 After register.html is open, the registration information will be got by register.js and be validated by register.php.
2. After favoriteList.html is open, the content and action of this page are controlled by favoriteList.js and favoriteList.php. In the favoriteList.html, by sending request to searchSpecific.php, searchSpecific.php will echo part of html code to show the detailed content of one blog. By sending request to delete.php, users can remove one blog from favorite list.
3. After create.html is open, the created content will be stored to database by create.php, we use createArticle.js to send the ajax request.
4. After selftemp.html is open, the self-written articles of users will be shown by requesting the selfArticle.php via selftemp.js
 - 4.1 Users can see the detailed content when clicking the title of an article, getSpecificArticle.php will echo the detailed content.
 - 4.2 The modifySpecificArticle.html will be open by clicking the modify button on selftemp.html by requesting the modifySpecificArticle.php via modifySpecificArticle.js. Users can delete this article by requesting the delete SpecificArticle.php via modifySpecificArticle.js. Users can save the modified content by requesting the saveModify.php via modifySpecificArticle.js
 - 4.3 Users can directly delete the article on selftemp.html by requesting the deleteSpecificArticle.php via selftemp.js.
5. After searchResult.html is open, users can see the all the titles of the results by requesting search.php via searchResult.js.
 - 5.1 When click the title of an article, users can see the detailed content by requesting searchSpecific.php. Also, users can remove/add it from/to the favourite list by requesting the star.php.



The ER model of our database is like this. First is the users table. We use username and password to identify one user.

The articles table initially has nine columns, but we have no enough time to realize the thumb up functions, so this column actually is equal to 0. The author of this article is recognized by the uid as a foreign key by referencing users table. The third table is user_star_articles table which is used to store all the articles which are in the favorite list of users. Both user_id and article_id are foreign keys.

○ When using other technologies:

- Describe which technologies/frameworks you considered and which ones you have chosen and WHY!

1 We use rich text editor (RTE) in our project. Because when users writing articles, they always want to make the article attractive and emphasize some important stuffs, so we apply the RTE technology to make users set many kinds of styles and implement the" what you see is what you get" function. One category of RTE named u editor (UE) was chosen because the source of UE is freely available, its functions are very comprehensive and the access process is simple.

2 We use AJAX technology to interact with PHP in our project because AJAX can exchange data with the server and update part of the web page content without reloading the entire page.

3 We use Toastr plug-in to make our prompt box more beautiful. Because the original alert is not very pretty, so we use toastr to improve

it.

- Explain which technologies you have used for which functionality of your project

We use RTE for the create article and modify article functions. Users will create and modify article on the u editor interface.

We use AJAX when the front end needs the data in database, like the CRUD and login/register functions.

We use Toastr to show all the errors and success hints.

5. Test

- **What did you test and how did you test it?**

1 Our group use the Use case test to test the whole functional requirements of the website.

For example, in the login interface, our test step as follows

Scenario	Description of steps
Successfully log in.	1 User input the username and password
	2 Website validates the inputs
	3 Website allows the user to access .
Exception 1: user inputs the wrong username	The website hints that username or password wrong. Then use case ends
Exception 2: user inputs the wrong password	The website hints that username or password wrong. Use case ends
Exception 3: user does not register	The website hints that username or password wrong. Use case ends

Test cases

Test condition1: Input a right pair of username and password

The website validates the inputs.

The website shows the user to log in successfully and switch to the home page.

Test condition2: Enter a wrong username

The website hints that username or password wrong.

Test condition3: Enter a wrong password.

The website hints that username or password wrong.

Test condition4: User log in without registration.

The website hints that username or password wrong.

The second test we use is equivalence partition for the function register,

The password format in our website is that it must include both digits and letters, the length is from 8-16.

In our test because the input requirement belongs to formatted, therefore:

Valid EC V1= {password| password=yhr99520}

Invalid EC V2= {password| password=12345AB}

These two values will be used to test in the register interface. When type in yhr99520, this is an acceptable password, but If the user wants to set the password into 1, the website will show the message: the password must contain both digits and letters, the length is 8-16.

The third test we use is usability test for the whole website.

1 User analysis and profiles: the users are broken into 4 categories including: young people with a strong creating desire of food blogs; young people who loves food of high quality; young people who are familiar with the use of blogs; young people with choice difficulties.

2 Decide what to test:

Select tasks for different categories. For example: for young people with a strong creating desire of food blogs, the task is to create a food blog and modify it after the first submitting on the website.

Select performance objectives. Still in the above example, the performance objectives include how long to complete the task, how long to find the login, create and modify button, if the user find the modify button successfully? If the task has been performed successfully and so on.

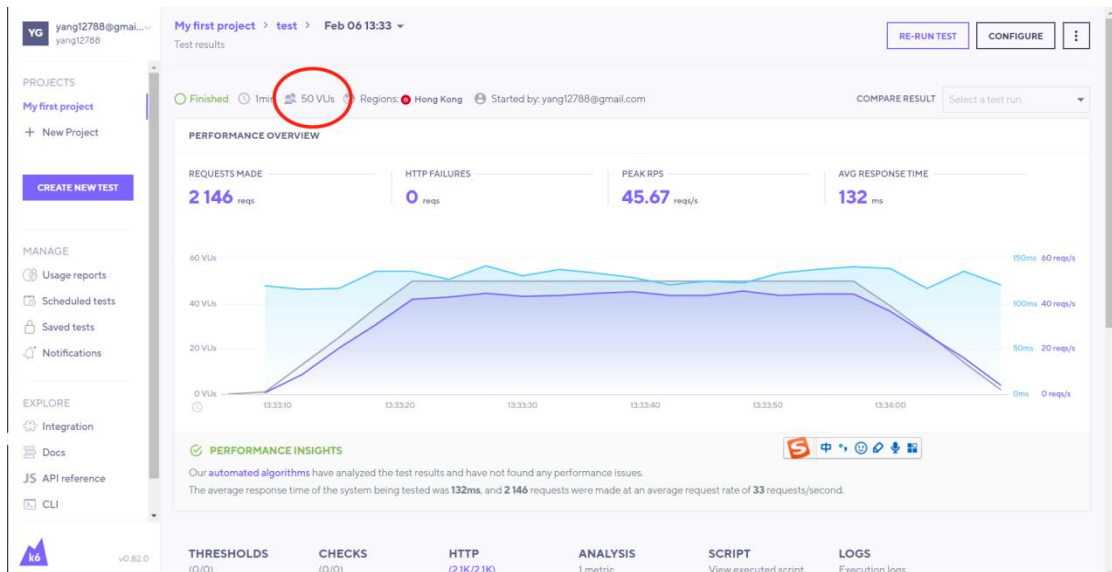
3 Prepare for test

We choose tasks according to each category of users, then we welcome them, paying our recruits with many delicious snacks, Haoran is the facilitator, Rui yang is the note taker, Tianshi is the pilot test user because he also have passion for create food blogs. Then we clean the desk and adjust the screen height to make the tester feel more comfortable.

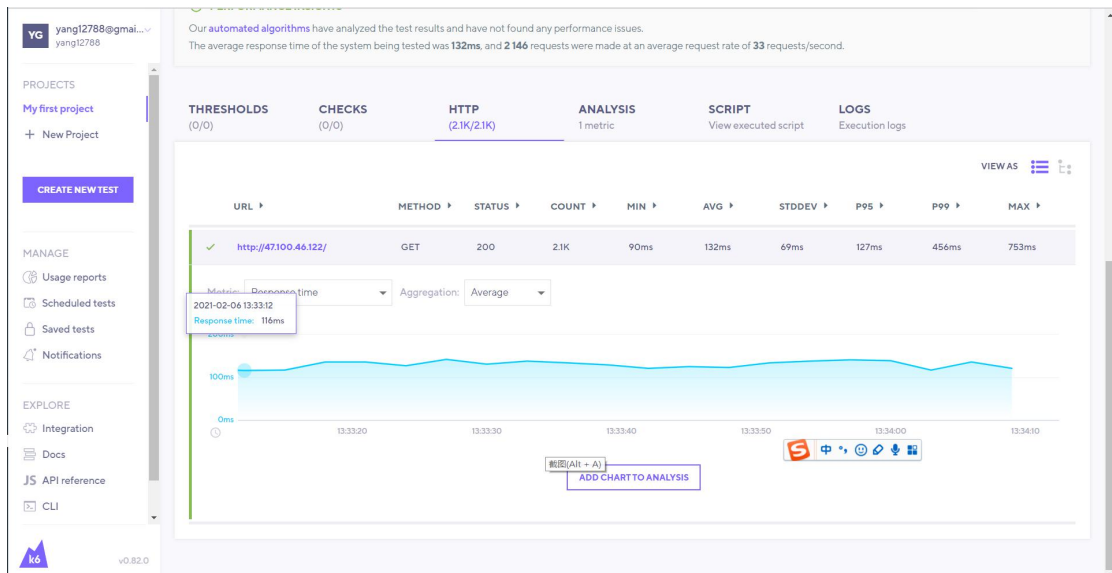
4 Conduct the test

5 Analyze the date: we adopt the buttom-up approach and determine the previous selected performance objectives. Then we adjust our website again and again.

The fourth test we use is online website test.



We set the concurrent user number to 50. The website is fine.



From this picture we can see the average response time is around 100ms.

Browser Timings

Redirect	0ms	Connect	491ms	Backend	239ms
TTFB	0.7s	First Paint	0.8s	DOM Int.	1.2s
DOM Loaded	1.5s	Onload	1.7s	Fully Loaded	2.5s

From this picture we can see the several timings

- **Why have you decided to do this specific kind of testing?**

The reason why we choose use case test is that we can find other bugs when individual parts integrating together, and these bugs cannot be found when testing individual parts.

The reason why we choose equivalence partition is that the test case numbers can be largely decreased.

The reason why we choose usability test is that the website can be more user friendly, thus to attracting more users and expand the data size.

The reason why we choose online test is it can test the non functional requirements more easily.

6. Evaluation

- **Evaluate your work process: what would you do differently next time? Have there been any difficulties?**

Improvements next time:

We think we will choose a consecutive time period to do the project or record what we have done and what we still need to do later, because every time when we have an interval for about 2 days, it is very struggle to remember what we write previously and feel disconnected.

We think next time we should design the code structure more carefully. Because the first two assignments are not that complicate, so we could ignore some code redundancy, but this time the quantity and complexity of code increased a lot, so the code redundancy becomes a bigger problem. We always found that when we want to add a new function, we must change several part of the code.

Difficulties:

We underestimate the complexity of the program, so that we spend a lot of time to even finish the basic function.

When we deploy the program to the cloud server, there existed a lot of problems due to some incompatibility such as the version of PHP, the version of mysql and some security authority problems.

- **Give a short overview of task responsibilities by group members**
Haoran Yang: php and part of javascript, test, database design, writing report
Rui Yang: html, css, part of javascript, test, database design, writing report
Tianshi Feng: Database, test