

MSc Computing and IT Management  
School of Computer Science and Informatics, Cardiff University

---

# **A Tutoring Website Aimed at Teenagers in Shanghai**

---



Supervisor:  
**Daniela Tsaneva**

Moderator:  
**Paul L Rosin**

Author:  
**Sicheng Zhou**

Date: 28/09/2021

# Table of content

|                                      |    |
|--------------------------------------|----|
| Table of content .....               |    |
| Abstract.....                        | 1  |
| Acknowledgment .....                 | 2  |
| 1. Introduction.....                 | 3  |
| 2. Aim and Objectives.....           | 5  |
| 3. Problem.....                      | 6  |
| 4. Background.....                   | 8  |
| 4.1 Current Phenomenon .....         | 8  |
| 4.2 Marketing Analysis.....          | 11 |
| 4.3 Web Development tool.....        | 15 |
| 4.3.1 Html5 .....                    | 15 |
| 4.3.2 CSS3 .....                     | 15 |
| 4.3.3 JavaScript.....                | 16 |
| 4.3.4 Python .....                   | 16 |
| 4.3.5 MySQL .....                    | 16 |
| 4.3.6 Flask.....                     | 16 |
| 4.3.7 Vue.js.....                    | 17 |
| 5. Methodology.....                  | 18 |
| 5.1 Process Model.....               | 18 |
| 6. Requirement specification .....   | 20 |
| 6.1 Functional Requirement .....     | 20 |
| 6.2 Non-Functional Requirement ..... | 22 |
| 7. Design.....                       | 23 |
| 7.1 Use case Design .....            | 23 |
| 7.1.1 Use case 1 .....               | 24 |
| 7.1.2 Use case 2 .....               | 26 |
| 7.1.3 Use case 3 .....               | 27 |

|   |    |
|---|----|
| 7.1.4 Use case 4 .....  | 28 |
| 7.1.5 Use case 5 .....  | 29 |
| 7.1.6 Use case 6 .....  | 30 |
| 7.1.7 Use case 8 .....  | 30 |
| 7.1.8 Use case 9 .....  | 31 |
| 7.2 Page Design .....   | 32 |
| 7.2.1 Login and Register Page .....                                   | 32 |
| 7.2.2 Home Page.....  | 33 |
| 8.2.3 Release Information Page &Release Information Details Page..... | 35 |
| 7.2.4 Reset Password Page .....                                       | 36 |
| 7.2.5 Chatroom Window & Generate Order Window .....                   | 37 |
| 7.2.6 Notification Window .....                                       | 38 |
| 7.3 Database Design .....   | 40 |
| 7.3.1 User Table .....  | 40 |
| 7.3.2 Teacher Table.....  | 41 |
| 7.3.3 Parent Table .....  | 41 |
| 7.3.4 Post_Teacher .....  | 42 |
| 7.3.5 Post_Parent Table .....   | 42 |
| 7.3.6 Order Table .....   | 43 |
| 7.3.7 Recommend Table.....  | 44 |
| 7.3.8 Question Table.....   | 44 |
| 7.3.9 Admin Table .....   | 44 |
| 7.3.10 Chat_room Table .....  | 45 |
| 8. Implementation .....   | 46 |
| 8.1 Configuration Environment .....                                   | 46 |
| 8.1.1 Docker .....  | 46 |
| 8.1.2 Redis .....   | 47 |
| 8.2 Register .....  | 47 |
| 8.2.1. Fill in the address .....                                      | 48 |
| 8.3 Login.....  | 51 |
| 8.3.1 Verification code .....   | 52 |
| 8.3.2 Log out.....  | 53 |
| 8.4 Release Information & Looking for a tutor .....                   | 54 |
| 8.4.1 Update &Delete Released Information .....                       | 55 |
| 8.5 Chat Room .....   | 57 |

|  |    |
|--|----|
| 8.6 Generate Order . . . . .   | 60 |
| 8.7 Search Keyword &Filter bar. . . . .  | 62 |
| 8.7.1 Searching Keyword . . . . .  | 62 |
| 8.7.2 Filter bar. . . . .  | 62 |
| 8.8 Chinese and English mode . . . . .   | 64 |
| 9. Analysis. . . . .   | 65 |
| 9.1 Test . . . . .   | 65 |
| 9.2 Aims and objectives retrospective . . . . .  | 74 |
| 9.2.1 Aim - To develop a platform for the distribution of tutors for college students and parents of teenagers in Shanghai . . . . .                       | 74 |
| 9.2.2 Objective - Provide a service platform for users to publish and receive information about tutors . . . . .   | 74 |
| 9.2.3 Objective - Use an advanced Web development technology and from which learn some new knowledge and skills . . . . .                                  | 75 |
| 9.2.4 Objective - Design a simple website interface so that users can quickly and efficiently use the platform's various functions . . . . .               | 75 |
| 9.2.5 Objective - Create a simple management platform that can effectively verify the authenticity of college students . . . . .                           | 75 |
| 9.2.6 Objective - Ensure that the website has great security measures which can protect the personal information that users fill in on the platform. . . . | 77 |
| 9.2.7 Objective - Test the platform continuously to ensure the satisfaction of users when using the website after the completion of project . . . . .      | 77 |
| 10. Conclusion . . . . .   | 78 |
| 10.1 Background Summary . . . . .  | 78 |
| 10.2 Methodology Summary . . . . .   | 79 |
| 10.3 Requirement Specification Summary . . . . .   | 79 |
| 10.4 Design Summary. . . . .   | 79 |
| 10.5 Implementation Summary. . . . .   | 80 |
| 10.6 Future Work . . . . .   | 80 |
| 11. Reflection . . . . .   | 83 |

|                         |    |
|-------------------------|----|
| <i>References</i> ..... | 86 |
|-------------------------|----|

## **Abstract**

The tutoring platform is presented to college students and parents of teenagers in Shanghai, China in the form of web pages.

During the COVID-19 outbreak, college students have lost many opportunities for part-time jobs, and teenagers' studies have also been affected. Every teenager has his weak subject, and every college student has his good subject, which can perfectly be complementary to each other. The best way for this group is to build a platform that benefits both sides.

The main functions of this website are login and registration, parental tutoring information release, view teacher information, recommendation matching, search and filter, chat room, comments and so on. Through various functions, the practicability of the tutoring platform can be improved.

## Acknowledgment

During the study for more than one year, though I was unable to participate in offline teaching in the college due to COVID-19, I was still able to get great help and support from all tutors and students in the college online. As the paper is nearing completion, I would like to thank all Cardiff University staff and my classmates.

First of all, I would like to thank my mentor, Dr.Daniela Tsaneva, who has been constantly giving me suggestions for improvement, providing me with many helpful documents and giving me the greatest encouragement throughout the project. My website development and thesis writing are inseparable from the guidance of Dr.Daniela Tsaneva.

Secondly, I would like to thank all my lecturers who taught me. Through their careful teaching, my computer skills got a breakthrough from A to Z.

Finally, I would like to thank all the staff in Cardiff for providing me with such a convenient online teaching method, which perfectly reappears the offline teaching.

# 1. Introduction

In the current world, the phenomenon of private supplementary tutoring became a strong trend. According to the statement of Hussein (1987), extra help was provided to students and the number was keeping a quick growth around the world (Ewan et al. 2017). This kind of “help” is beyond the regular schooling, and following the definition of Bray (2007), they are the parallel system which involve the curriculum of mainstream education and enhance students’ command of knowledge. Based on the proposition of Murawska and Putkiewicz (2005), it pointed out that the private supplementary tutoring can make students response more efficiently as well as flexibly and was well appraised as the important supplement to the mainstream education system (Silova 2009). That explained the reason why the private supplementary tutoring draw so much attention all over the world.

The phenomenon of private supplementary tutoring is more pervasive in China where both of students and parents show a strong demand for private supplementary tutoring, due to the stress of enrollment and peer pressure. But in the recent year, both of the regular schooling and private supplementary tutoring get a great shock because of the Covid-19. The restriction established to prevent the infection of Covid-19 lead to a hard situation for offline teaching. Many students are forced to stay at home which causes a influence on their performance of study to a certain degree. Meanwhile, the restriction also causes higher unemployment rate, especially for the college students, which will make their debt even worse.

Viewing from these main points, this paper decided to design a platform which can form a win-win relationship between teens and those college students who was unemployed (but not just limited to this group). Given that private supplementary tutoring is necessary for the Chinese students, one of the functions of the platform is



to provide the online teaching website which make the private supplementary tutoring easier to be accessed and will not be constrained by the space and time. That function devoted to solving the problem and demand of teens and their parents for the private supplementary tutoring. In the meantime, considering the current situation of Covid-19 and preparing for the future educational change, which will make the online education to be the mainstream educational model, this platform can give the opportunity for the college students to play the role of teacher for those teen students. According to the statement of Murawska and Putkiewicz (2005), in some extent, tutoring can provide the income for the tutors, because tutoring can increase learning and human capital. This provides the feasibility to decrease the unemployment rate among the college students and relieve the stress of their debt.

The following sections will be included below:

In Chapter 2, I will describe the aims and objectives of the project. In Chapter 3, I will describe the problem of the project. In Chapter 4, I will make a thorough investigation into the background of the tutoring platform, including the background of the tutoring market and the development tools to be used. In chapters 5, 6, and 7, I will determine the Methodology for the project, analyze all requirements, and introduce all design processes. In chapters 8 and 9, I will describe the implementation of the platform and test results in detail. In chapter 10, I will make a summary of the whole paper. In chapter 11, I will summarize my experience in doing this project and reflect on my shortcomings.

## 2. Aim and Objectives

The main purpose of this project is to develop a platform distributing tutors for college students and parents of teenagers in Shanghai. In addition, considering that many college students cannot find suitable part-time jobs due to the epidemic, and the study of teenagers has been restricted (classes have been suspended), thus the platform will not be used for any commercial purposes from the perspective of public welfare.

The following are some of the Objectives of the project:

1. Provide a service platform for users to publish and receive information about tutors.
2. Use an advanced Web development technology and from which learn some new knowledge and skills.
3. Design a simple website interface so that users can quickly and efficiently use the platform's various functions.
4. Create a simple management platform that can effectively verify the authenticity of college students.
5. Ensure that the website has great security measures which can protect the personal information that users fill in on the platform.
6. Test the platform continuously to ensure the satisfaction of users when using the website after the completion of project.

### **3. Problem**

The main problems of this project is to solve the issue that college students in Shanghai can release part-time tutor information through the website and parents can find suitable tutors for teenagers through the website. At present, the problems are divided into the following four points: Market, Design, Development and Test.

#### **Market**

Since the website is used in Shanghai, China, local education policies and the number of users are key issues that need to be solved before the development of the website, which can not only ensure the long-term survival of the website, but also reserve load capacity for the server to be deployed in advance (Manhas, 2017). In addition, how to make the website function complete requires not only self-reflection by developers, but also detailed analysis as well as comparison of local websites and Apps to achieve the effect of learning from each other (Svensson et al. 2010).

#### **Design**

Before the development of the tutor website, we should design the page, function and database of the website. Beautiful website pages can quickly improve users' satisfaction and trust (Nia and Shokouhyar, 2020). The perfection of website functions can improve users' frequency while using (Zhang and Von, 2000). The design of the back-end database determines the speed of the website and the frequency of errors (Harrington, 2016). The three key points of "design problem" are how to present the overall appearance of the web page succinctly to the users, how to meet the main needs of the users with the functions of the web page, and how to create and associate the back-end database of the web page logically.

## **Development**

Since the developers of this project only know a few development tools before studying for the master's degree, the use of various development tools is limited to the basic content taught by the course CMT120-Fundamentals of Programing including Python, Flask, Html, CSS, and JavaScript. The realization of all functions of tutoring website will face difficulties and challenging problems. Resolving this issue allowed the project developer to efficiently complete the code of the project within the deadline.

## **Test**

After the completion of development, web testing is a necessary step. It can not only enable developers to find the defects of the website before the release, but also ensure that users can experience the complete web page functions after the release of the web page and improve the user's experience. How to perform functional testing on the web page after development is also a problem developer must consider.

The above four issues will be mentioned and analyzed in the following paragraphs. These problems will be solved when the project is completed.

## 4. Background

### 4.1 Current Phenomenon

The origin of tutoring can be traced back to the ancient Greek period, and the main purpose is to enable learners to acquire knowledge in the teaching process as well as be able to skillfully use and master it (Gardner et al. 2007). However, in the early 21st century, this new mode of tutoring for college students gradually became popular. College students tutoring is taught by college students in their spare time according to the knowledge they have learned and provides guidance as well as help for teenagers. At the meantime, they get a certain reward from the tutor; Parents also pay less for the tutor than the one in tutoring agencies (Hille et al. 2016).

**Figure 1: Student Enrollment By Various School (1978~2019)**

| Year | Institutions of Higher Education | Secondary School | of which                     |                          |                             |                         | Primary School | Special Education School |
|------|----------------------------------|------------------|------------------------------|--------------------------|-----------------------------|-------------------------|----------------|--------------------------|
|      |                                  |                  | Specialized Secondary School | Regular Secondary School | Secondary Vocational School | Technical Worker School |                |                          |
| 1978 | 14.79                            | 99.97            | 9.95                         | 76.23                    | 9.12                        | 4.67                    | 106.46         | 0.62                     |
| 1996 | 15.38                            | 100.81           | 11.10                        | 74.43                    | 10.29                       | 4.99                    | 102.44         | 0.63                     |
| 1997 | 16.51                            | 102.53           | 12.38                        | 73.85                    | 10.72                       | 5.58                    | 96.14          | 0.52                     |
| 1998 | 18.63                            | 105.66           | 13.06                        | 76.69                    | 10.21                       | 5.70                    | 87.16          | 0.53                     |
| 1999 | 22.68                            | 105.28           | 11.89                        | 79.54                    | 8.48                        | 5.37                    | 78.86          | 0.54                     |
| 2000 | 28.00                            | 104.37           | 12.12                        | 80.23                    | 7.48                        | 4.54                    | 72.28          | 0.48                     |
| 2001 | 33.16                            | 103.59           | 12.66                        | 78.97                    | 7.56                        | 4.40                    | 67.24          | 0.55                     |
| 2002 | 37.85                            | 100.71           | 13.69                        | 75.47                    | 7.04                        | 4.51                    | 64.83          | 0.55                     |
| 2003 | 41.57                            | 106.94           | 14.05                        | 82.78                    | 6.43                        | 3.68                    | 53.74          | 0.54                     |
| 2004 | 44.26                            | 99.24            | 13.67                        | 77.02                    | 5.76                        | 2.79                    | 53.50          | 0.52                     |
| 2005 | 46.63                            | 92.25            | 13.70                        | 71.17                    | 5.33                        | 2.05                    | 53.37          | 0.50                     |
| 2006 | 48.49                            | 85.17            | 12.81                        | 65.60                    | 5.20                        | 1.56                    | 53.33          | 0.50                     |
| 2007 | 50.29                            | 79.97            | 12.08                        | 61.77                    | 4.80                        | 1.32                    | 59.06          | 0.51                     |
| 2008 | 51.28                            | 77.07            | 11.50                        | 60.37                    | 4.14                        | 1.06                    | 67.12          | 0.50                     |
| 2009 | 51.57                            | 75.20            | 10.91                        | 59.44                    | 3.77                        | 1.08                    | 70.16          | 0.50                     |
| 2010 | 51.13                            | 73.96            | 10.22                        | 59.17                    | 3.52                        | 1.05                    | 73.11          | 0.49                     |
| 2011 | 50.66                            | 73.45            | 9.88                         | 59.04                    | 3.55                        | 0.98                    | 76.04          | 0.49                     |
| 2012 | 50.48                            | 72.63            | 9.23                         | 59.35                    | 3.24                        | 0.81                    | 79.25          | 0.47                     |
| 2013 | 50.66                            | 69.67            | 7.74                         | 58.42                    | 2.72                        | 0.79                    | 80.30          | 0.46                     |
| 2014 | 51.16                            | 67.39            | 7.24                         | 57.05                    | 2.32                        | 0.78                    | 79.87          | 0.45                     |
| 2015 | 51.47                            | 66.67            | 6.68                         | 57.11                    | 2.04                        | 0.84                    | 78.97          | 0.43                     |
| 2016 | 51.49                            | 66.15            | 6.31                         | 57.06                    | 1.97                        | 0.81                    | 78.49          | 0.43                     |
| 2017 | 51.78                            | 67.97            | 5.98                         | 59.07                    | 1.99                        | 0.93                    | 80.02          | 0.44                     |
| 2018 | 52.63                            | 69.70            | 5.70                         | 61.04                    | 1.97                        | 0.99                    | 82.63          | 0.48                     |

Note: Quantity of students enrollment of secondary schools in this table doesn't include those of reformatory schools.

The potential user group is the key factor for the sustainable development of Tutoring website. In 2020, the part-time job rate of college students in Shanghai has sharply decreased due to the COVID-19, and many service industries cannot operate normally. Most of the industries that can take part-time jobs have adopted the mode of online work with the rapid development of the Internet. According to the Statistics of Shanghai Statistics Bureau (Figure), in 2019, there were about 530,000 college students in Shanghai, and 1.52 million primary and secondary school students (Shanghai Statistics Department 2020). This is a huge educational resource, namely the potential users of the tutor website, which enables Shanghai college students to tutor primary and secondary school students in various subjects during the epidemic period, and to select the subjects for guidance according to their strengths. Suspended primary and secondary school students can also get meaningful tutoring at home at a relatively affordable price. Though the lockdown caused by COVID-19 is canceled for now in Shanghai, and the part-time jobs rate becomes higher, the COVID-19 is not easy to be controlled. Therefore, online part-time jobs are a wise choice for the situation now and expected to be the future.

It is also necessary for the tutoring website to operate in line with the country's education policy. At present in China, parents' paying great attention to children's education results in a large number of extracurricular education tutoring organization/site showing up. The profit of them is great. But teachers' salaries are a small part of those huge profits, with most of the rest pertaining to investors or people outside the education industry. The influx of capital violates the fundamental principle of the education industry and destroys the ecology of education. At present, private tutoring websites in Shanghai adopt the mode of charging fees. See 4.2 below for detailed Market Analysis. In 2021, China's Ministry of Education issued a "guideline", which includes some regulations: off-campus training institutions shall not make profits; Reduce the learning pressure of young people; Promote public

welfare education. (Chinese Government 2021). Public education is the original intention of the family education website. In addition, college students' tutor only helps students with their homework in school and does not impose additional pressure on students (such as homework outside school), which is in line with the current policies. In China, responding to the government's call is a big factor in the sustainability of websites.

## 4.2 Marketing Analysis

**New Oriental** <http://www.xdf.cn/>

New Oriental is a top education website in China. It has a professional one-to-one tutoring section, featuring personalized and customized programs, personalized teaching plans for students, and also has the head teachers to follow up the learning situation at any time. Their courses are taught online. New Oriental has its unique brand effect and can attract a large number of old and new customers. Compared with letting a large number of teachers enter, it pays more attention to elite teachers.

**TTgood** <https://www.ttgood.com/shanghai/>

TTgood Tutor Website is an excellent tutoring website in China. It features a large number of teachers and students to attract customers. Customers can screen their favorite tutors according to various conditions. But the site is not equipped with recommendation functions, so users have to look for by themselves.

**Educase** <https://sh.eduease.com/>

Educase is also an old tutoring website in China, which also has a large number of teachers and students. But due to the long history of the website, the website picture is relatively simple. It has no online chat, nor recommendation. Users have to search by themselves, and the operation is rather complex.

**JiaJiao114** <https://www.jiajiao114.com/>

Tutoring 114 is a recently emerging tutoring website with exquisite pictures, attracting many users. At the same time, it can make an appointment with customer service placed on the website according to the needs of users, so as to save the trouble of users and be fast and convenient. What is in the ointment is that it has no comment function, so parents cannot check the teacher's teaching levels.



**Zhangmen** <https://www.zhangmen.com/>

The Zhangmen one-to-one website is in the leading position in the tutoring industry, so it has a good reputation like New Oriental and attracts a large number of users. Therefore, it also takes private customization as a feature, focusing on elite teachers, and the price will naturally be higher. What is more inconvenient is that there is no way to check the teachers' teaching situation through the comments of other users, and cannot give parents a good reference.

**Yousi** APP Store

Yousi is a novel tutoring website in China. It cooperates with universities to launch work-study programs to help many college students enter the tutoring industry, and the teaching form is online teaching. It can also filter teachers and check out specific information based on different requirements. But Yousi currently only supports APP form, which cannot better serve more customers.

**Figure 2: Function comparison table of other websites**

| Site Name  | Fee | Online Chat | Searching &Filter | Language    | APP/ Website | Comment | View teacher details | Release information | Teaching crowd    |
|------------|-----|-------------|-------------------|-------------|--------------|---------|----------------------|---------------------|-------------------|
| XDF        | ✓   |             | ✓                 | Chinese     | Website &APP |         | ✓                    |                     | Full-time teacher |
| TTgood     | ✓   |             | ✓                 | Chinese     | Website      |         | ✓                    | ✓                   | ALL               |
| Educas     | ✓   |             | ✓                 | Chinese     | Website &APP | ✓       | ✓                    | ✓                   | ALL               |
| JiaJiao114 | ✓   |             |                   | Chinese     | Website      |         | ✓                    | ✓                   | ALL               |
| Xes        | ✓   |             |                   | Chinese     | Website &APP |         |                      |                     | Full-time teacher |
| Zhangmen   | ✓   |             |                   | Chinese     | Website &APP |         |                      |                     | Full-time teacher |
| YouSi      | ✓   |             |                   | Chinese     | APP          |         | ✓                    | ✓                   | College Student   |
| My Website |     | ✓           | ✓                 | Chinese/E N | Website      | ✓       | ✓                    | ✓                   | College Student   |

## Marketing Analysis

For the understanding and exploration of the above seven websites, I will use SWOT analysis method to objectively and accurately analyze the actual situation of the tutor website I will develop.

### Strength

The target audience for my tutor website is clear. The teachers who can tutor through the tutor website must be college students. My tutor website provides both Chinese and English versions. In Shanghai, an international city, more and more overseas students and foreign parents gather here, which not only expands the audience but also provides a professional platform for those overseas people who want to contact with tutors. The platform has an online chat function, which enables users to communicate and leave messages in a timely manner without the need to provide mobile phone numbers. At present, none of the seven websites above has this

function. My platform does not charge any handling fee and does not have any profits. It is purely for public benefit.

### **Weakness**

As the website is a personal development, there are certain limits in technology and funds, and there is no certain plan for the promotion of the website. The website only supports web version, not developed for mobile application software. At the same time, because the choice of the website teachers are targeted at college students, the teacher force will be relatively weak and their tutors are still in the exploratory stage. Nowadays, there are many tutoring centers in the market causing a fierce competition. It is difficult to find a breakthrough for website promotion at the beginning.

### **Opportunities**

The Chinese government in 2021 has just issued a new policy —"about to further reduce the compulsory education stage student's opinions on homework burden and off-campus training"— to carry on the strict specification for those for-profit education institutions (Chinese Government 2021). Two-thirds of Shanghai education institutions and the site has been closed because of the policy consolidation or direct collapse. Under this policy, my public welfare website has not been affected. Due to the departure of a large number of competitors, the market will retain a large number of users, which is the best opportunity to expand the user group.

### **Threats**

With a large number of new users using the website I developed, it is a very difficult task to maintain the website and audit college students in the later stage. In the complex Internet, the protection of users' personal privacy is also another big problem that needs to be conquered at present. Meanwhile, as the environment changes and more non-profit tutoring sites flood in, how to retain old users and attract new ones will be an ongoing problem.

Overall, the tutor website I have developed has a prominent advantage, but also has certain insufficiency. Developers must first use the advantage of website, win the fame in the specific audience, not be shaken by nowadays tutor market, establish the initial goal, at the same time avoid the possible faults. The developers should not only improve the technology, keep up with the times and make out the mobile applications, but also excavate the elites among existing teachers and reduce the disadvantage of the lack of elite teachers. In the current non-profit environment, tutoring website is a good start. However, if developers want to go further, they need to grasp the opportunity and make sufficient and effective promotion. Finally, for every potential threat, web developers should always be on high alert to improve their site, keep in touch with society, and find the unique appeal of the site to keep new users using it.

## **4.3 Web Development tool**

### **4.3.1 Html5**

As a language description method for building Web content, HTML5 is the most commonly used HTML version in the world at present. Due to the popularity of mobile devices, HTML5 can be more adaptable to the vast majority of mobile browsers, in this way, functions can be implemented not just on computers. It has also been added with semantic tags, allowing developers to write more specific specifications and helping developers who review code to understand more easily the semantics.

### **4.3.2 CSS3**

CSS3's full name is Cascading Style Sheets 3, and its main function is to decorate HTML5's rudimentary pages and format elements in a variety of scripting languages.

It is an updated version of CSS, added with a lot of contemporary design styles and effects.

### **4.3.3 JavaScript**

As a scripting language, JavaScript is widely used in Web application development around the world. It can transform HTML5 pages from static to dynamic, making them smoother and more beautiful when presented to users.

### **4.3.4 Python**

Python is a computer programming language that provides simple and effective object-oriented programming while offering efficient and comprehensive high-level data structures. Its complete lightweight syntax and dynamic typing, as well as the interpreted language characteristics, make it a favorite for many people to quickly develop applications or write scripts. As it continues to be updated and to be added with new features, it is being used for larger developments nowadays.

### **4.3.5 MySQL**

MySQL is one of the most commonly used open-source relational database management systems. Python can be associated with the system through the API provided by the system. Since the tutor website needs to store and process a large amount of data, MySQL can ensure the integrity and security of the data.

### **4.3.6 Flask**

Flask is a lightweight Web application framework written in Python. Flask keeps the code simple and extensible, allowing developers to quickly implement a website or Web service.

### 4.3.7 Vue.js

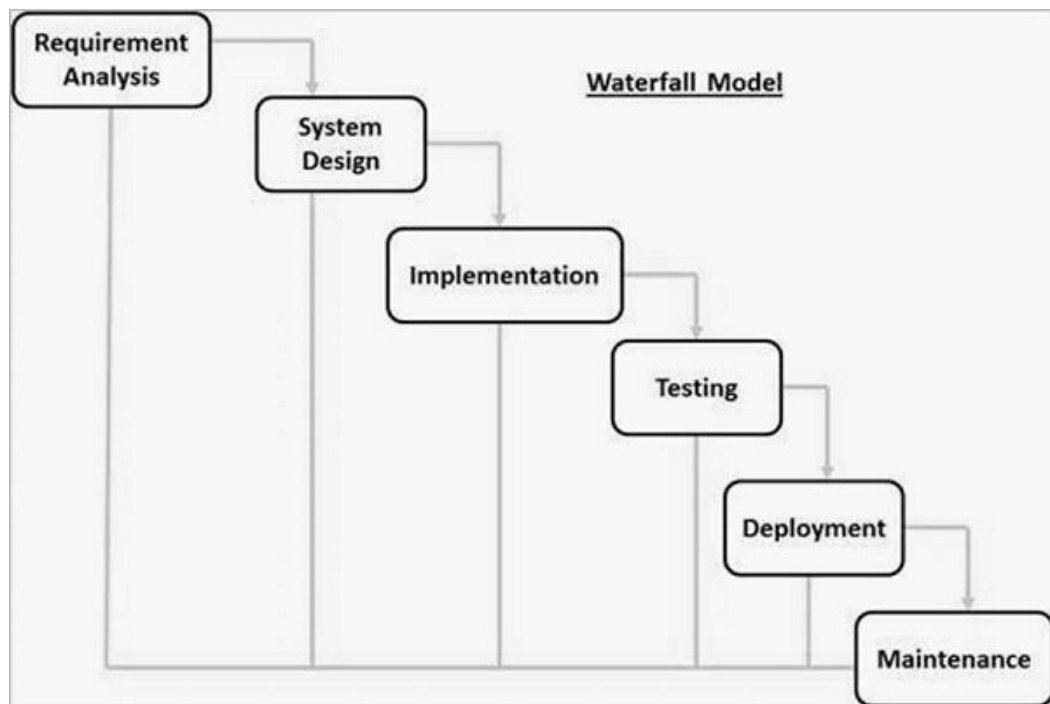
Vue is a set of progressive frameworks for building user interfaces. This means that Vue can be easily embedded as part of a project. Vue has many advanced features of Web development, such as decoupled views and data, reusable components, front-end routing technology, state management, virtual DOM, and more. Unlike other large frameworks, Vue is designed to be applied layer by layer from the bottom up. This means that you can freely add component systems, client-side routing, and large-scale state management to the declarative rendering view template engine. These functions are independent of each other and can be freely selected based on core functions, not necessarily all integrated together.

## 5. Methodology

### 5.1 Process Model

The project selected the Waterfall Model among many software development Process models, which has been widely used since 1970, and according to the survey, Waterfall Model is currently the most commonly used method (Vijayasarathy et al. 2016). Since the developer will develop a website independently at a certain time, an iterative form like Waterfall Model to simplify the process will be needed; In addition, the developer has cleared the requirements of the website according to the market exploration, and there will be no major changes after the completion of the website development. These features happen to be the advantages of Waterfall Model. Compared to a similar development Model, the V Model, it pays more attention to testing at each stage and requires a lot of time to test and modify requirements documents (Balaji and Sundararajan 2012). The Waterfall Model is divided into six steps according to Figure 3:

**Figure 3: SDLC Tutorial – Waterfall Model (2021)**



*Requirement Analysis* - The developers will add and delete the features of their website design by means of questionnaire survey, and the features will be improved.

*System Design* - Through Mockkitt this web page Wireframe drawing software to carry on the front end of the page design and draw the back-end UML diagram as well as the flow chart of each function to complete the back-end code logic design.

*Implementation*-Developers will use Pycharm to write HTML and CSS on the front end and Python-Flask on the back end. Dynamic web database will use Mysql and Redis.

*Testing* - Developers will test each feature developed in the requirements for multiple times, counting bugs and changing the code accordingly.

*Deployment* - The developer's site is targeted at users in Shanghai, China, choosing a more stable Alibaba Cloud server for deployment.

*Maintenance* -The developers will write new requirements through the feedback of users after using the website and simple management of the website through the background management page.



## 6. Requirement specification

### 6.1 Functional Requirement

Functional requirements are offered by the tutor platform to provide users with a variety of operational and interactive functions.

The following functional requirements will be classified as must have, should have, and do not have.

#### **Must-Have**

- Users can register and log in.

Functional Qualification Criteria: The login registration function is divided into two roles: college students and parents. Its functionality is distributed on the platform's entry page. There are two registration pages distinguishing college students from parents; There is only one landing page, and it has graphic verification code function to ensure the security of the website.

- Users can post information about their tutors

Functional Qualification Criteria: Users can successfully edit and send tutor information to the platform.

- Users can view information about their tutors

Functional Qualification Criteria:

Users will be able to successfully view tutoring information posted by other users.

- Users can chat with each other on the site

Functional Qualification Criteria: Parents and college students can successfully do instant chats or leave messages on the site.

- An order can be generated after a consensus reaching between the users.

Functional Qualification Criteria: Parents and college students can mutually determine the conditions of tutor negotiation and generate a simple voucher on the order page of both sides.

- Administrators can review the registration information of college students

Functional Qualification Criteria: Website administrators can review the information of college students registered:

Pass - college students registered successfully.

Fail- college students registered wrongly.

### **Should-Have**

- Users can upload images at registration.
- Users can change the password of account.
- Users can modify the personal information entered at registration.
- Users can evaluate and give a mark to the finished orders.
- Users are able to switch between English and Chinese on the site.
- Users are able to mark precise addresses on the map rather than manually enter them.
- Users are able to view tutoring information posted by other users by keyword searching or conditional filtering.
- Users can log out of the site.
- Users who log in to the site and close the browser does not need to log in again but passes to the homepage directly.

### **Not-Have**

- Platform has no payment capabilities.
- Platform does not install any AD space.

## **6.2 Non-Functional Requirement**

Non-functional requirements are not directly related to the specific functions of the system and are also the most critical requirements of the tutoring website platform.

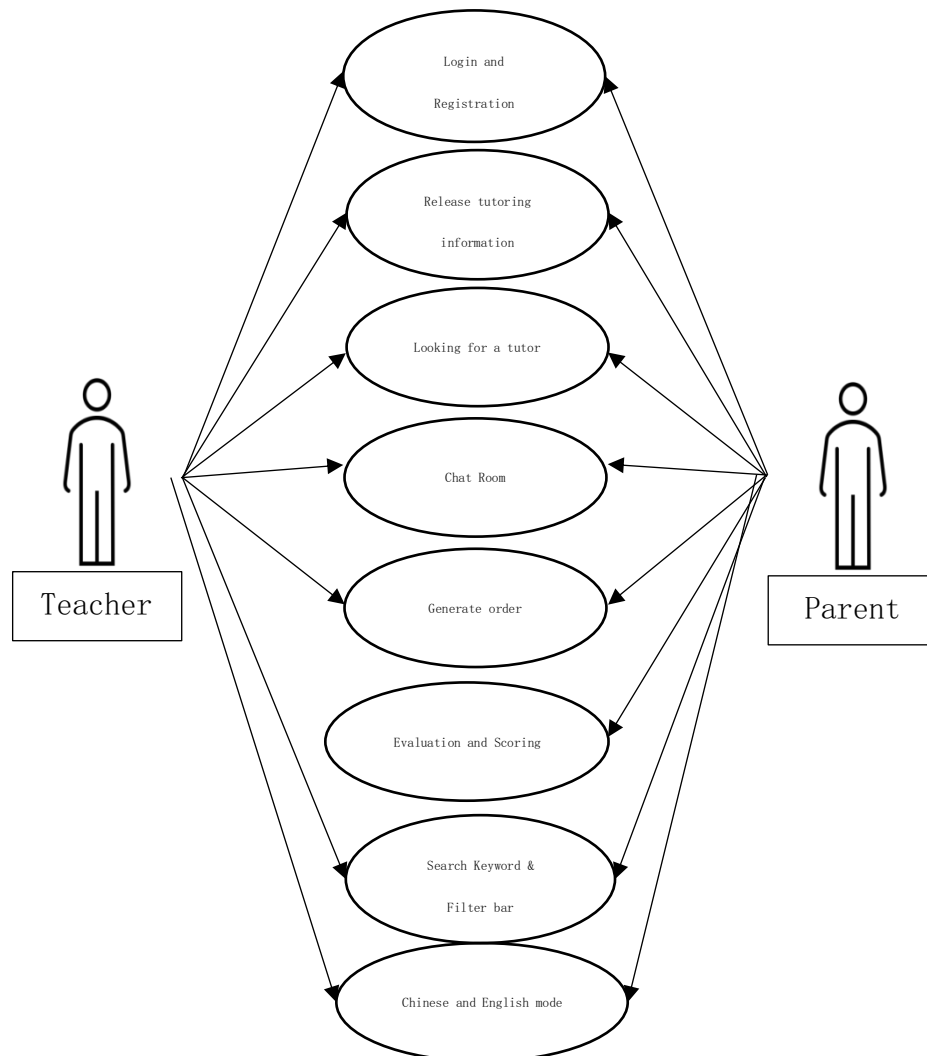
- Platforms are able to pop up an instructional prompt when users' operations are faulty.
- Platforms are open to different browsers and are easily accessible from mobile browsers.
- Platforms are capable of achieving a response time of less than three seconds in 80% of cases.
- Platforms will be weekly maintained.
- Platform servers currently support 100 simultaneous logins.
- Web pages are designed in a brief style.
- Platform code is written to be designed for extensibility.
- The platform supports the uploading of images in different formats.
- Platform protects users' personal information.
- All pages of the platform can be switched between Chinese and English.

## 7. Design

### 7.1 Use case Design

According to the functional requirements of college students and their parents, the uml diagrams of use cases are made. The UML diagram is divided into nine functions, and the roles are divided into teacher and parent, with a small number of different functions between them.

**Figure 4: Use Case Diagram**



### 7.1.1 Use case 1

#### **Registration and Login**

To interact with tutors on the site, users must first register to use the site. The registration page for parents and teachers is different, but the form needs to be completely filled in with the required information. Then the data will be sent to the database table 'user', and the parent and teacher will be identified in the database. If the registration is successful, the platform will prompt the user that the registration is successful. The user can then log in using the username and password, and the forgotten password can be modified by the user's security question. When logging in, you must fill in the random picture verification code, which ensures the security of the website to a certain extent. If the username and password are entered correctly, the user can choose to remember it so that they do not need to log in again next time.

#### *Operation flow:*

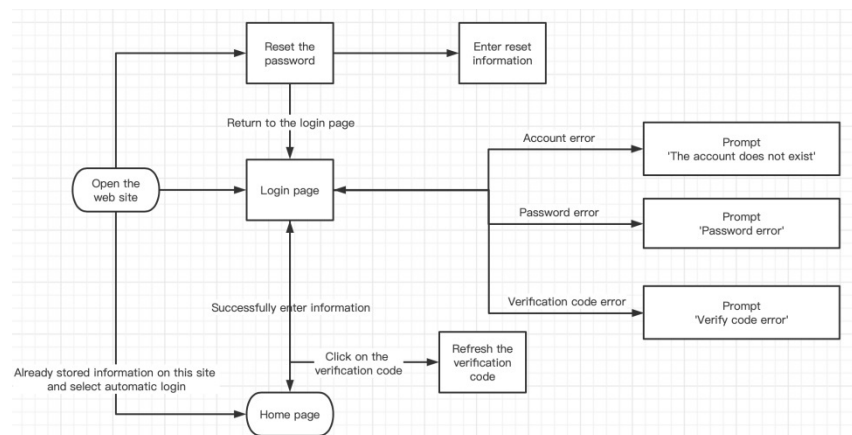
1. Choose 'Register as a Teacher', 'Register as a Parent', or 'Log in' from the navigation bar.
2. If the user chooses to register, the user needs to enter the data into different input boxes according to its id.
3. The user submits data.
4. Select 'login' and enter the correct username and password.
5. Enter the correct random verification code.

#### *Alternative flow:*

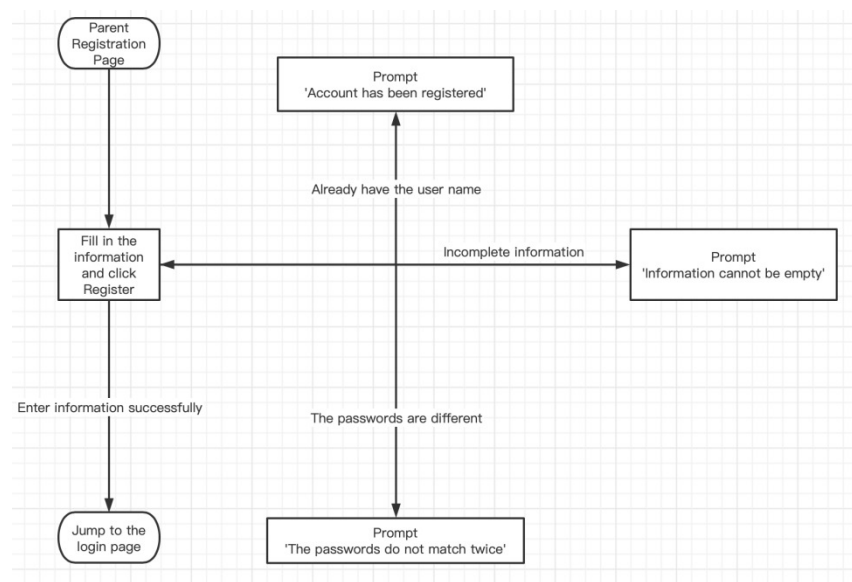
1. If you forget your password on the login page, you can click the "Forget Password" button and change your password by answering the password question.
2. If the user does not enter the correct username or password, a message will be displayed indicating that the user has entered an incorrect username or password.
3. Every page on the platform has a "logout" button that users can click for returning to the login page.

4. After a user selects the "Keep me as signed in" button to log in to the web page, the user can return to the web page without logging in again if the page is closed accidentally.
5. When filling in registration information, if the information is slightly missed and the user has clicked submit, the prompt box will accurately indicate which column is not filled in or is filled in incorrectly.

**Figure 5: Flowchart-Login**



**Figure 6: Flowchart-Registration**



### 7.1.2 Use case 2

#### Release tutoring information

The user can be a parent or a teacher, which depends on the user's mark in the database when logging in. Both users can click the publish button to fill in their relevant information to publish, so that the published information can be viewed by users on the tutor website, which is a fast and convenient way to find a tutor.

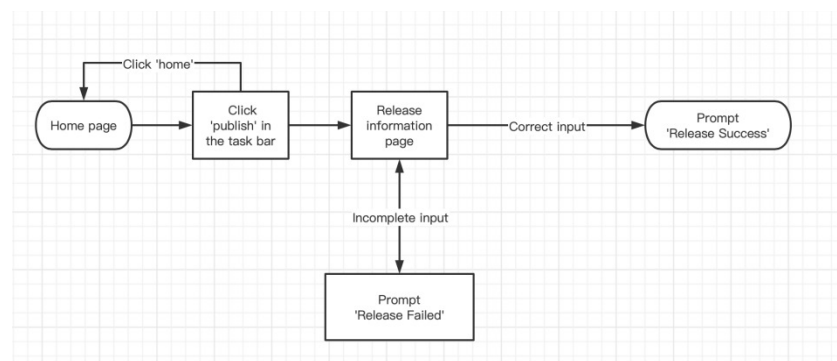
#### *Operation flow:*

1. The user clicks the Publish button after successfully logging in to the page.
2. Input and select the details of tutor information.
3. After editing the content, click the "Submit" button to submit the content. The page will prompt the user "You have successfully published the tutor" and return to the home page.

#### *Alternative flow:*

1. If the user does not enter all the data and click publish, the prompt bar will prompt 'publish failed'.

**Figure 7: Flowchart-Release Information**



### 7.1.3 Use case 3

#### Looking for a tutor

The user can be either a parent or a teacher, depending on the user's mark in the database when logging in. Both can find the tutor information that meets their expectations according to different situations. When they find a mentor they are interested in, they can click on the order title for more specific details or contact the user directly. Then the users will decide together whether to establish an order.

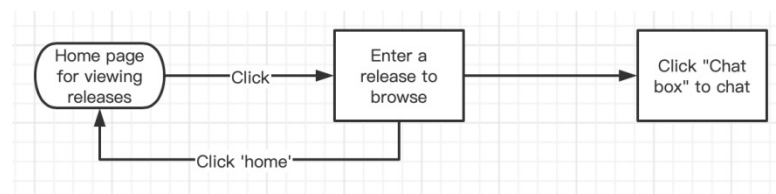
#### *Operation flow:*

1. After successfully logging in to the page, the user can see the information published by the corresponding user (if the login user is a teacher, the information published by parents is displayed) on the main page.
2. Users can click the "More Information" of each tutor to obtain more detailed information (if the user is a parent, the teacher can see the comments and scores of parents who have taught before).
3. If the user is interested in a tutor or a student, they can click the chat button in the home page or details to communicate or leave a message.

#### *Alternative flow:*

User can use the filter function to filter quickly.

**Figure 8: Flowchart-Looking for a tutor**





#### 7.1.4 Use case 4

##### Chat Room

The chat room function can only be used after the user releases the tutor information, the function allows two users to chat privately, which is the only way to communicate in the web page. This will ensure the user's other communication tools not being leaked and is also the key link to generate tutor orders. Users can make instant talks when both users are online. If a user is not online, the online user can use the message function to send messages. When another user logs in, a message will be displayed on the web page. After both sides reach a consensus, there will be a button in the chat room to initiate an order.

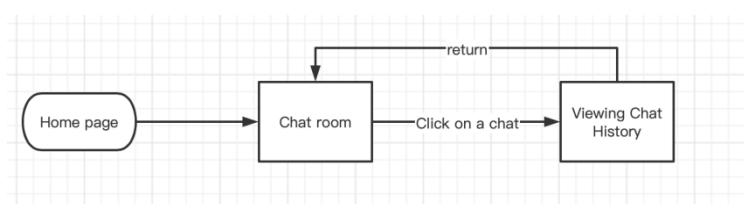
##### *Operation flow:*

1. When users are interested in a tutor, they can click the chat button below the published information to get in touch with the selected user. The system then sets up a chat room for them.
2. Users can enter text in the chat room.
3. Users can click the "Send" button to send messages.
4. Users can click "confirm" or "modify" to make operations on the order in the chat room.

##### *Alternative flow:*

1. Users can click the order button to pop up a window for filling in the order information.
2. Users can press the "exit" to exit the chat room at any time.

**Figure 9: Flowchart-Chat Room**



### 7.1.5 Use case 5

#### Generate order

After successful communication between parents and teachers, users can initiate an order, which is an online credential. The chat room has a button to jump to the order filling page. When the user fills in the order, the detailed information of the order will be displayed in the chat room of both parties. The other user must confirm the order in the chat room before the order can be generated. The finished order will be permanently saved in the order bar. If another user has objections to the order initiated, he can click 'modify order', then the order originator becomes the order modifier, and the logic after that is the same as before.

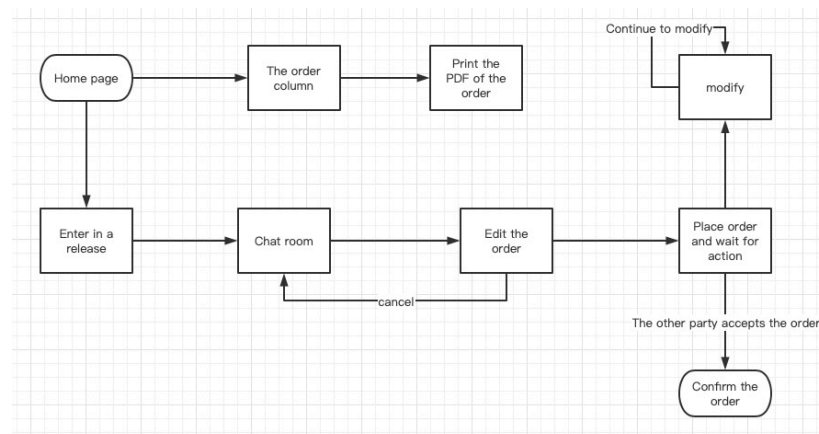
#### Operation flow:

1. The user clicks the "Generate order" button in the chat room.
2. On the order details page, fill in the order details and confirm the order.
3. The person who initiates the order clicks "confirm".
4. If both parties agree to the order, the order details page of both parties will save it.

#### Alternative flow:

1. The order recipient can click the "Modify" button to modify the order information and resend the order.
2. Click 'Print' to download the PDF file of the order.

**Figure 10: Generate order**



### 7.1.6 Use case 6

#### Evaluation and Scoring

When parents and teachers successfully generate an order, parents can evaluate and score the order. The evaluation will be displayed on the release information details page of the teacher, and the score will be accumulated to an average value and be displayed on the same page.

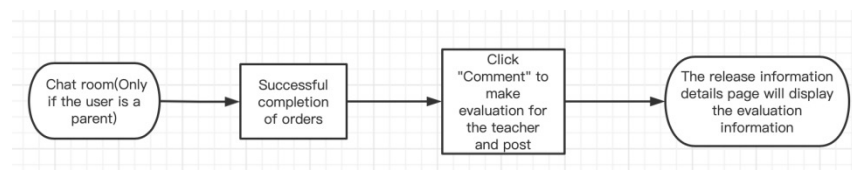
#### *Operation flow:*

1. Parents and teachers successfully generated orders.
2. Parents can click the comment button in the order message in the chat room.
3. After the Evaluation and Scoring window popping up, the parent fills in the relevant information, then clicks Submit.
4. The score and evaluation will be recorded in the database, and the tutor information details page released by the teacher will be updated with the newly added score and evaluation.

#### *Alternative flow:*

1. Parents can choose not to click the comment button without any consequences.

**Figure 11: Flowchart-Evaluation and Scoring**



### 7.1.7 Use case 8

#### Search Keyword & Filter bar

When there is too much information posted by tutors, it is difficult for users to find the tutors they want in the list. They can filter out published content that contains only search terms or filter options by filling in their own required keywords in the search

bar or checking the filter bar for filtering. This will reduce the time for users to rummage through tutoring information.

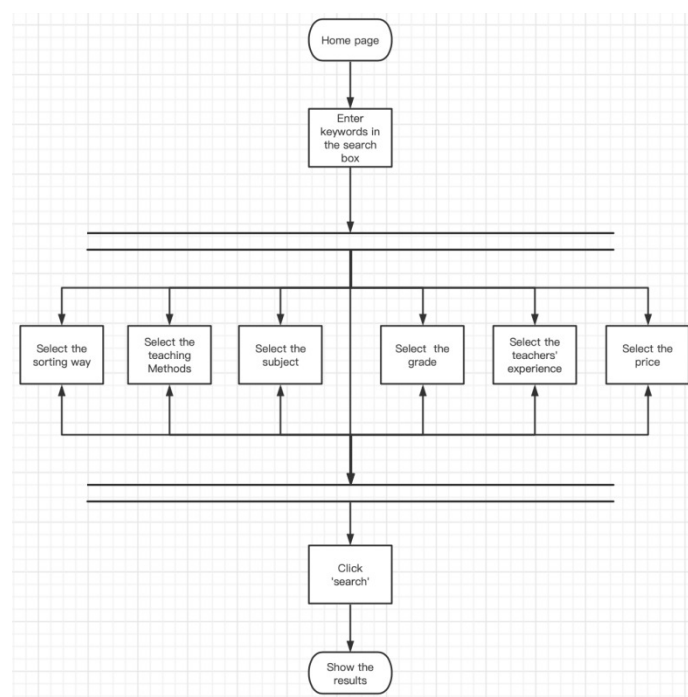
*Operation flow:*

1. Log in to the home page.
2. Enter search terms and click the search button or select Sorting options.
3. Tutor information list will be refreshed to display the filtered information list.

*Alternative flow:*

1. This function is not required.

**Figure 12: Flowchart-Searching & Filter bar**



### 7.1.8 Use case 9

#### Chinese and English mode

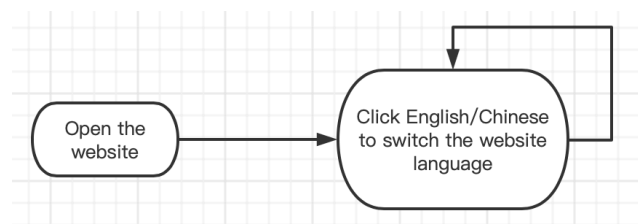
Since the website is targeted at Shanghai, and there are a lot of overseas students in Shanghai, in order to enable overseas students to use the tutor website, the mode of Switching between Chinese and English is added, and all the information on the website

can be switched between Chinese and English.

*Operation flow:*

1. Click English or Chinese in the upper right corner of any page.
2. The page will be refreshed, and the text on the page will be converted to the selected language mode.

**Figure 13: Flowchart-Chinese & English Mode**



## 7.2 Page Design

In the design of front-end page, Bootstrap and Vue, the most commonly used front-end frameworks are applied. Buttons on the tutor web page will all be based on Bootstrap making the web page simple and beautiful. Vue implements page refreshes so that users don't have to reload the entire page each time they use it. It also uses Select2 plugins to optimize the look of dropdowns and checkboxes when users register or fill out information. Finally, the prompt box will pop up every time the user operates successfully or incorrectly. The Toastr plug-in is selected to make the prompt box not dull and boring.

### 7.2.1 Login and Register Page

The login and registration pages will use blue as main color to attract users. Blue is a popular color on Educational websites and institutions in China. There are many registered users in the early stage of website development and the website can only be entered after registration, so if the registration and login can be displayed on one page separately by TAB bar, users will register more quickly. The text will be arranged in a

regular order from top to bottom, the 'reset password' button is highlighted by an underscore, and the text spacing is properly reserved to improve the user experience. Below is a wireframing of the page.

**Figure 14: Login and Register Page**

## 7.2.2 Home Page

At the top of the main page is a navigation bar containing various function buttons. The order of function buttons is based on the predicted frequency of using. The central area is for users to view the release details page. Due to the different needs of parents and teachers, the two different types of users present different views and Posting information pages. Considering that the format and size of life photos uploaded by teachers when registering are different, the display area of photos will adapt to the website automatically and display pictures with coordinated proportions in the information release list. The function button is suspended in the lower right corner of the page, and the button is a must for a tutoring website. The navigation bar will be hidden when the user pulls down the web page to search for published information. These buttons will be kept in the lower right corner of the page following the slide of the web page, which is convenient for the user using the web page

quickly. The following two pictures show the homepages of teachers and parents respectively.

Figure 15: Teacher Home Page

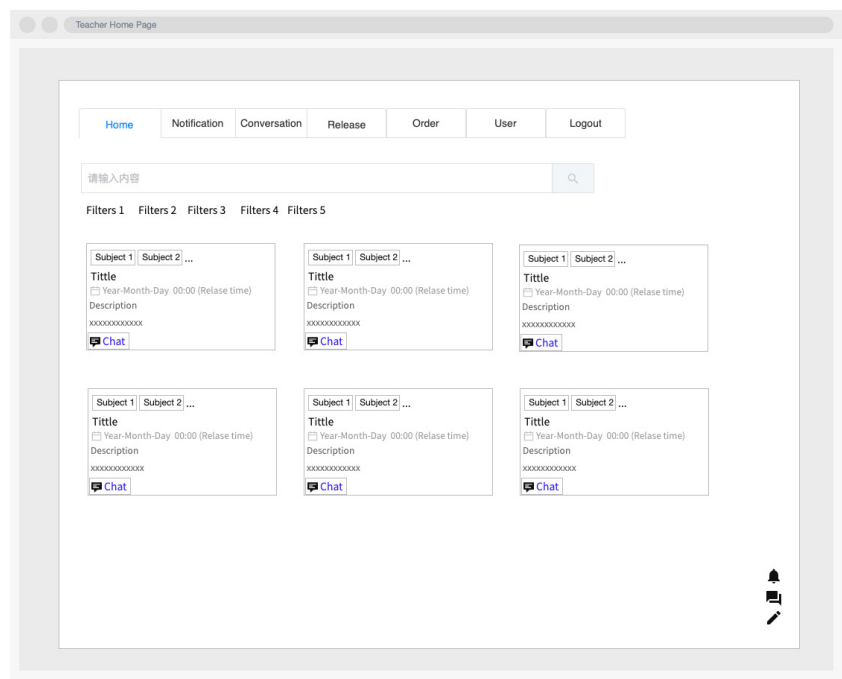
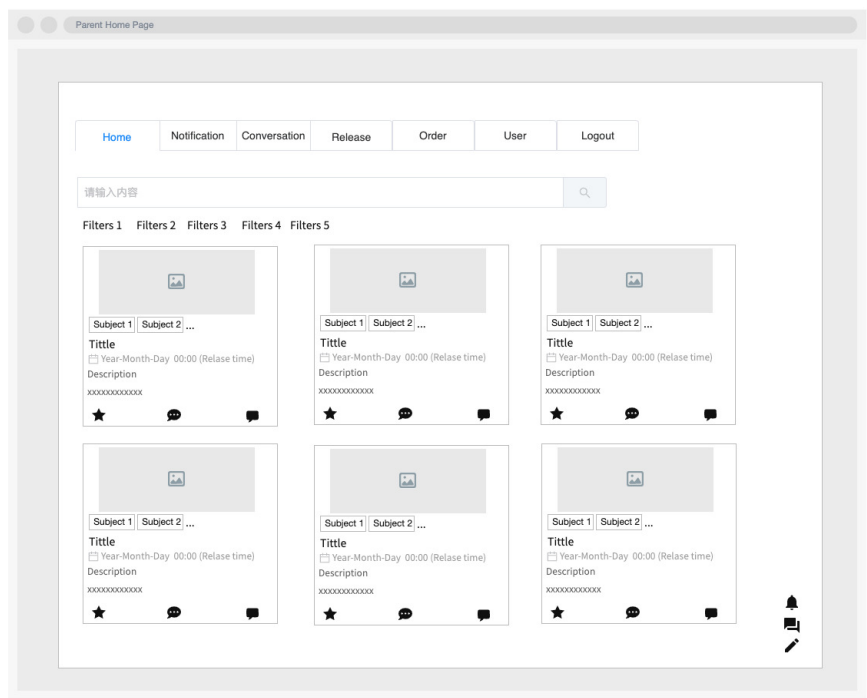


Figure 16: Parent Home page:



### 8.2.3 Release Information Page & Release Information Details Page

Publish information page is entered by clicking the publish button. The page is very simple, using several text box modes, such as multi-check box, check box and drop-down single box. It allows users to quickly and accurately fill in the information they want to release. In the lower right corner of the page, a button for returning to the home page is placed. Users can click to 'return' to the home page at any time and the information will not be released. The layout of the release information details page is basically the same as that of the release information page. In addition to the filled release information, it also displays the pictures and information uploaded by users during registration, enabling users who view details to better understand the details of the user who released the tutor. The following figure respectively shows the details page of teacher release.

**Figure 17: Teacher Release Information| Figure 18: Parents Release Information Page**

The image displays two side-by-side screenshots of web forms for releasing information. The left form is titled "Teacher Release Information" and the right form is titled "Parents Release Information". Both forms have a "Save & Post" button at the bottom and a home icon in the bottom right corner.

**Teacher Release Information**  
Post the subjects and strengths you're good at

Self-Introduction  
[Text Box]

Select grade  
Choose Grade [Dropdown]

Select Subject  
Mathematics [Checkbox] English [Checkbox]

Select Teaching way  
☐ Online ☐ Offline

Save & Post

**Parents Release Information**  
Post lessons child wants to catch up on

Child's current learning situation  
[Text Box]

Select child's grade  
Choose Grade [Dropdown]

Select child's Subject  
Mathematics [Checkbox] English [Checkbox]

Select Teaching way  
☐ Online ☐ Offline

Save & Post

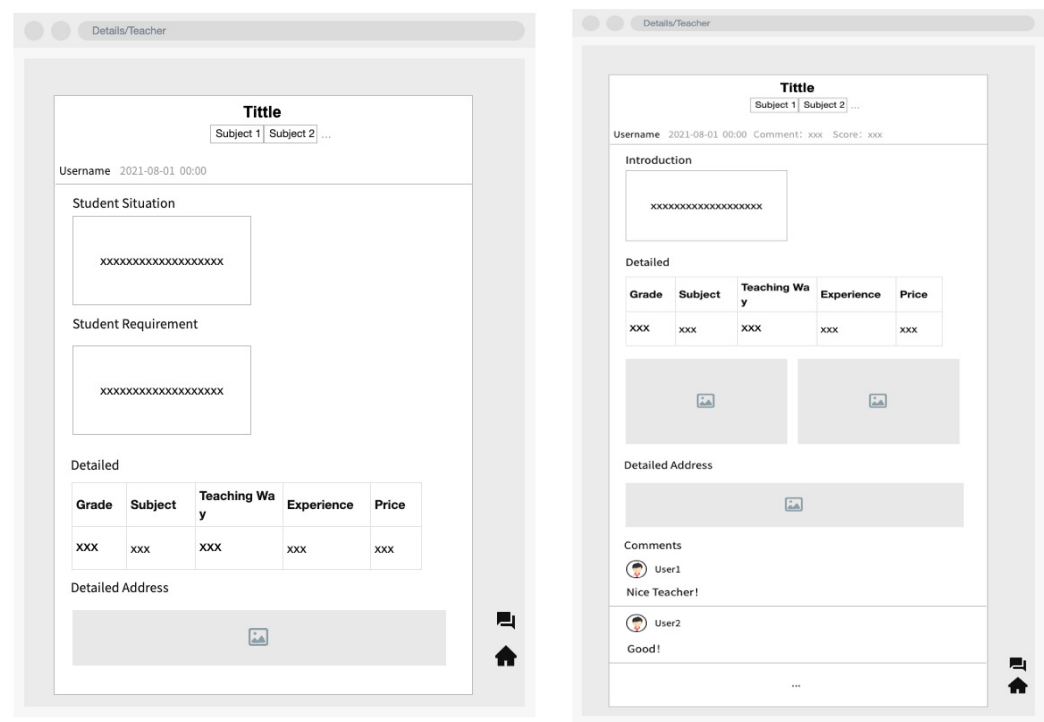


Figure 19: Parents Release Information|

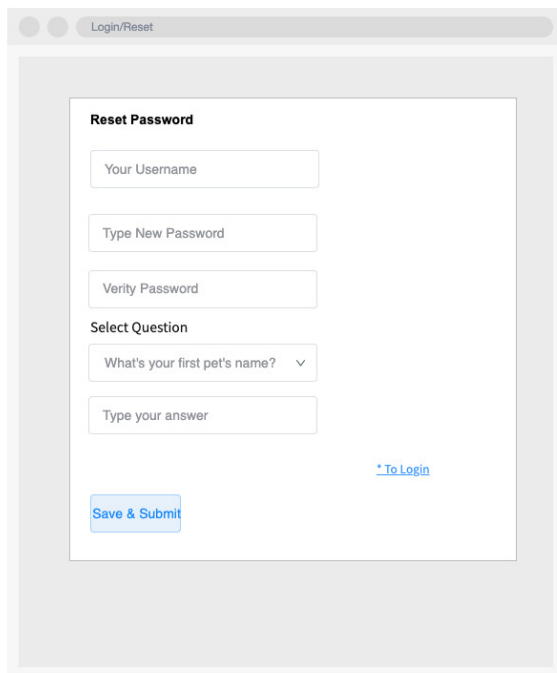
Figure 20: Teacher Release Information

Details Page

Details Page



**Figure 21: Reset Password Page**

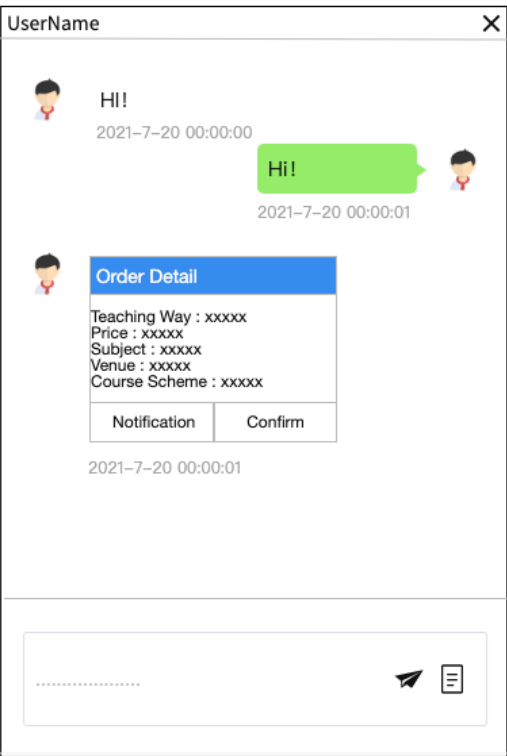


The screenshot shows a web browser window with the title 'Login/Reset'. Inside the browser, there is a 'Reset Password' form. The form contains the following elements: a text input field labeled 'Your Username', a text input field labeled 'Type New Password', a text input field labeled 'Verify Password', a section titled 'Select Question' with a dropdown menu showing 'What's your first pet's name?' and a downward arrow, and a text input field labeled 'Type your answer'. Below the dropdown menu is a blue link that says '\* To Login'. At the bottom left of the form is a blue button labeled 'Save & Submit'.

### **7.2.5 Chatroom Window & Generate Order Window**

Chat window in the case of already login, can be popped up at any time in any page of the tutor website. Users can communicate with each other or leave messages immediately. The time of sending a message is recorded in the chat window. There is also a button to generate an order next to the send button. Only through the 'Order Edit window' button or the 'Modify' button in the chat can pop up the modify order interface. The generated order is also displayed in the chat window. The following figure shows the chat window and the order generation window.

**Figure 22: Chatroom Window & Generate Order Window**



**7.2.6 Notification Window**

The notification window is accessed by clicking a button on the main page and can accept messages and orders from other users. Unread messages are indicated with red dots and numbers. The order message is presented in the form of text in the notification bar, marked with square brackets. Messages are sorted according to the time when users send messages from late to early. The following figure shows the notification window.

Figure 23&24: Notification Window

Notification

Test 1

Hello!

xxxx-xx-xx 00:00:00

1

Test 2

[Order Message]

xxxx-xx-xx 00:00:00

1

Test 3

xxxxxxxxxxxxxxxxxxxx

xxxx-xx-xx 00:00:00

Test 4

xxxxxxxxxxxxxxxxxxxx

xxxx-xx-xx 00:00:00

Test 5

xxxxxxxxxxxxxxxxxxxx

xxxx-xx-xx 00:00:00

Test 6

xxxxxxxxxxxxxxxxxxxx

xxxx-xx-xx 00:00:00

Edit the order

Teaching Way

Online

Price

...

Subject

Mathematical

Teaching Address

...

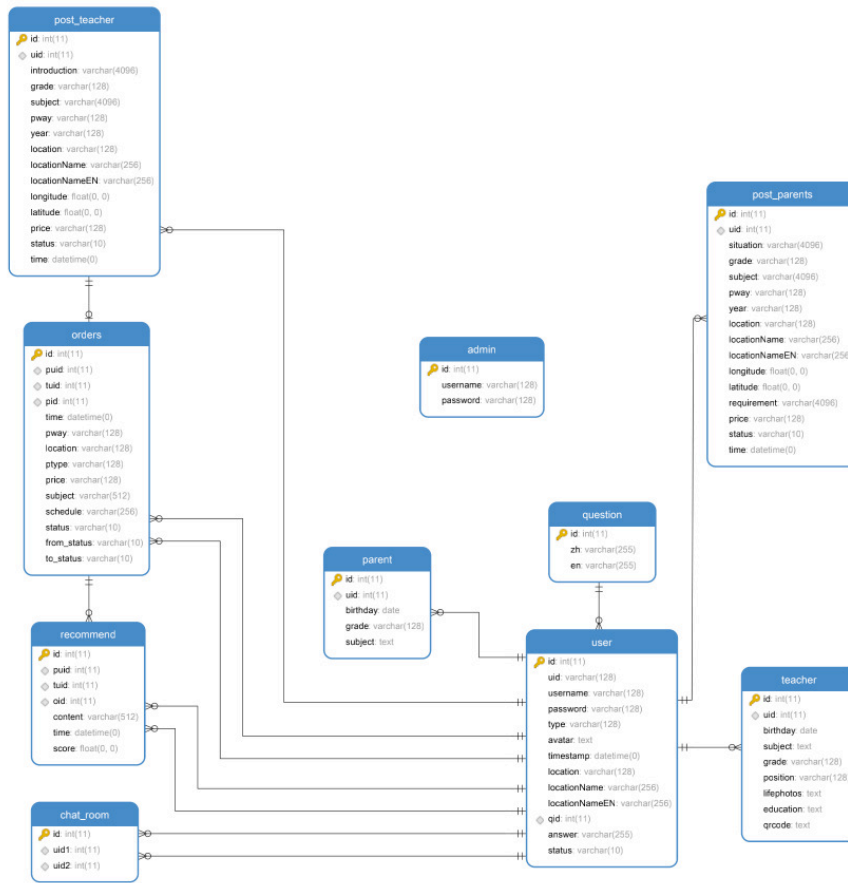
Course Scheme

...

Submit&Send

## 7.3 Database Design

Figure 25: Database structure diagram



### 7.3.1 User Table

The user table of tutor platform is used to store all information of teachers and parents when they register.

|   | Name     | Data format  | PK/FK | Data type | Note  |
|---|----------|--------------|-------|-----------|---|
| 1 | Id       | Int(10)      | PK    | String    |   |
| 2 | uid      | Varchar(128) |       | String    | User ID                                       |
| 3 | username | Varchar(128) |       | String    | Username                                      |
| 4 | Password | Varchar(128) |       | String    | Password                                      |
| 5 | type     | Varchar(128) |       | String    | User Type<br>1 Admin<br>2 Parent<br>3 Teacher |
| 6 | Avatar   | Text         |       | String    | Avatar  |

|    |                |              |  |        |                                  |
|----|----------------|--------------|--|--------|----------------------------------|
|    |                |              |  |        | Base64                           |
| 7  | timestamp      | datetime     |  | String | Register Time                    |
| 8  | location       | Varchar(128) |  | String | Latitude&<br>longitude           |
| 9  | locationName   | Varchar(256) |  | String | Address Name<br>(CN)             |
| 10 | locationNameEN | Varchar(256) |  | String | Address Name<br>(EN)             |
| 11 | qid            | Varchar(128) |  | String | Password Question<br>ID          |
| 12 | answer         | Varchar(255) |  | String | Password Question                |
| 13 | status         | Varchar(10)  |  | String | User Status<br>1 normal<br>2 ban |

### 7.3.2 Teacher Table

The teacher information table of tutor platform is used to store the specific teaching information of teachers.

|   | Name      | Data format  | PK/FK | Data type | Note                             |
|---|-----------|--------------|-------|-----------|----------------------------------|
| 1 | Id        | Int(10)      | PK    | String    |                                  |
| 2 | uid       | Varchar(128) |       | String    | User ID                          |
| 3 | subject   | text         |       | String    | Good at Subject                  |
| 4 | grade     | Varchar(128) |       | String    | Good at Grade                    |
| 5 | lifepotos | text         |       | String    | Life photo                       |
| 6 | education | text         |       | String    | Education<br>Certificate Picture |

### 7.3.3 Parent Table

The student information table of tutor platform is used to store students' specific teaching information.

|   | Name    | Data format  | PK/FK | Data type | Note               |
|---|---------|--------------|-------|-----------|--------------------|
| 1 | Id      | Int(10)      | PK    | String    |                    |
| 2 | uid     | Varchar(128) |       | String    | User ID            |
| 3 | grade   | Varchar(128) |       | String    | Want which Subject |
| 4 | subject | text         |       | String    | Want which Grade   |

### 7.3.4 Post\_Teacher

The teacher posting information table of tutor platform is used to store the teaching information posted by teachers.

|    | Name           | Data format   | PK/FK | Data type | Note                                  |
|----|----------------|---------------|-------|-----------|---------------------------------------|
| 1  | Id             | Int(10)       | PK    | String    |                                       |
| 2  | uid            | Varchar(128)  |       | String    | User ID                               |
| 3  | introduction   | Varchar(4096) |       | String    | Detailed introduction                 |
| 4  | grade          | Varchar(128)  |       | String    | Good at Subject                       |
| 5  | subject        | Varchar(4096) |       | String    | Good at Grade                         |
| 6  | pway           | Varchar(128)  |       | String    | Online or Offline                     |
| 7  | location       | Varchar(128)  |       | String    | Tutoring Address                      |
| 8  | locationName   | Varchar(256)  |       | String    | Address Name (CN)                     |
| 9  | locationNameEN | Varchar(256)  |       | String    | Address Name (EN)                     |
| 10 | longitude      | Float         |       | String    | Longitude                             |
| 11 | latitude       | Float         |       | String    | Latitude                              |
| 8  | requirement    | Varchar(128)  |       | String    | Hope the price                        |
| 9  | price          | Varchar(128)  |       | String    | 1 Deleted<br>2 Online<br>3 Have order |
| 10 | time           | datetime      |       | String    | Release time                          |

### 7.3.5 Post\_Parent Table

The parents posting information table of tutor platform is used to store the teaching information posted by parents.

|    | Name           | Data format   | PK/FK | Data type | Note                  |
|----|----------------|---------------|-------|-----------|-----------------------|
| 1  | Id             | Int(10)       | PK    | String    |                       |
| 2  | uid            | Varchar(128)  |       | String    | User ID               |
| 3  | situation      | Varchar(4096) |       | String    | Detailed introduction |
| 4  | grade          | Varchar(128)  |       | String    | Want which Grade      |
| 5  | subject        | Varchar(4096) |       | String    | Want which Grade      |
| 6  | pway           | Varchar(128)  |       | String    | Online or Offline     |
| 7  | location       | Varchar(128)  |       | String    | Tutoring Address      |
| 8  | locationName   | Varchar(256)  |       | String    | Address Name (CN)     |
| 9  | locationNameEN | Varchar(256)  |       | String    | Address Name (EN)     |
| 10 | longitude      | Float         |       | String    | Longitude             |
| 11 | latitude       | Float         |       | String    | Latitude              |

|    |             |               |  |        |                                       |
|----|-------------|---------------|--|--------|---------------------------------------|
| 12 | requirement | Varchar(4096) |  | String | Requirement                           |
| 13 | price       | Varchar(128)  |  | String | Hope the price                        |
| 15 | status      | Varchar(10)   |  | String | 1 Deleted<br>2 Online<br>3 Have order |
| 16 | time        | datetime      |  | String | Release time                          |

### 7.3.6 Order Table

The order information table of tutor platform is used to store the order information generated after the agreement between teachers and parents is reached.

|    | Name        | Data format  | PK/FK | Data type | Note   |
|----|-------------|--------------|-------|-----------|--|
| 1  | Id          | Int(10)      | PK    | String    |  |
| 2  | puid        | Varchar(128) |       | String    | Parent User ID   |
| 3  | tuid        | Varchar(128) |       | String    | Teacher User ID  |
| 4  | pid         | Varchar(128) |       | String    | Post   |
| 5  | time        | datetime     |       | String    | Order time   |
| 6  | pway        | Varchar(128) |       | String    | Online or Offline  |
| 7  | location    | Varchar(128) |       | String    | Tutoring Address   |
| 8  | ptype       | Varchar(128) |       | String    | 2 Parent initiate order<br>3 Teacher initiate order                                |
| 9  | price       | Varchar(128) |       | String    | Negotiated price   |
| 10 | Subject     | Varchar(512) |       | String    | Negotiated subject   |
| 11 | schedule    | Varchar(256) |       | String    | Negotiated date  |
| 12 | status      | Varchar(10)  |       | String    | 1 Order unconfirmed<br>2 Order confirmed   |
| 13 | from_status | Varchar(10)  |       | String    | Confirmation status of the originator of the order<br>1 Confirmed<br>2 Unconfirmed |
| 14 | to_status   | Varchar(10)  |       | String    | Confirmation status of the order taker<br>1 Confirmed<br>2 Unconfirmed             |



### 7.3.7 Recommend Table

The evaluation form for orders of tutor platform is used to store the evaluation and scoring information of orders.

|   | Name    | Data format  | PK/FK | Data type | Note                                    |
|---|---------|--------------|-------|-----------|---|
| 1 | Id      | Int(10)      | PK    | String    |   |
| 2 | puid    | Varchar(128) |       | String    | Parent User ID                          |
| 3 | tuid    | Varchar(128) |       | String    | Teacher User ID                         |
| 4 | oid     | Varchar(128) |       | String    | Order ID                                |
| 5 | content | Varchar(512) |       | String    | Comment on the content                  |
| 6 | time    | datetime     |       | String    | Comment time                            |
| 7 | score   | float        |       | String    | The score that parents give the teacher |

### 7.3.8 Question Table

The question table is used to store the secret protection questions during user registration or password change.

|   | Name | Data format  | PK/FK | Data type | Note                              |
|---|------|--------------|-------|-----------|-----------------------------------|
| 1 | Id   | Int(10)      | PK    | String    |                                   |
| 2 | zh   | Varchar(255) |       | String    | Chinese secret protection problem |
| 3 | en   | Varchar(255) |       | String    | English secret protection problem |

### 7.3.9 Admin Table

The admin form is used to store the user name and password for logging in to the admin page.

|   | Name     | Data format  | PK/FK | Data type | Note            |
|---|----------|--------------|-------|-----------|-----------------|
| 1 | Id       | Int(10)      | PK    | String    |                 |
| 2 | username | Varchar(128) |       | String    | Admin User Name |
| 3 | Password | Varchar(128) |       | String    | Admin Password  |

### 7.3.10 Chat\_room Table

The Chat\_room table is used to store the UserID of the two users who entered the chat room.

|   | Name | Data format  | PK/FK | Data type | Note    |
|---|------|--------------|-------|-----------|---------|
| 1 | Id   | Int(10)      | PK    | String    |         |
| 2 | uid1 | Varchar(128) |       | String    | User ID |
| 3 | uid2 | Varchar(128) |       | String    | User ID |

## 8. Implementation

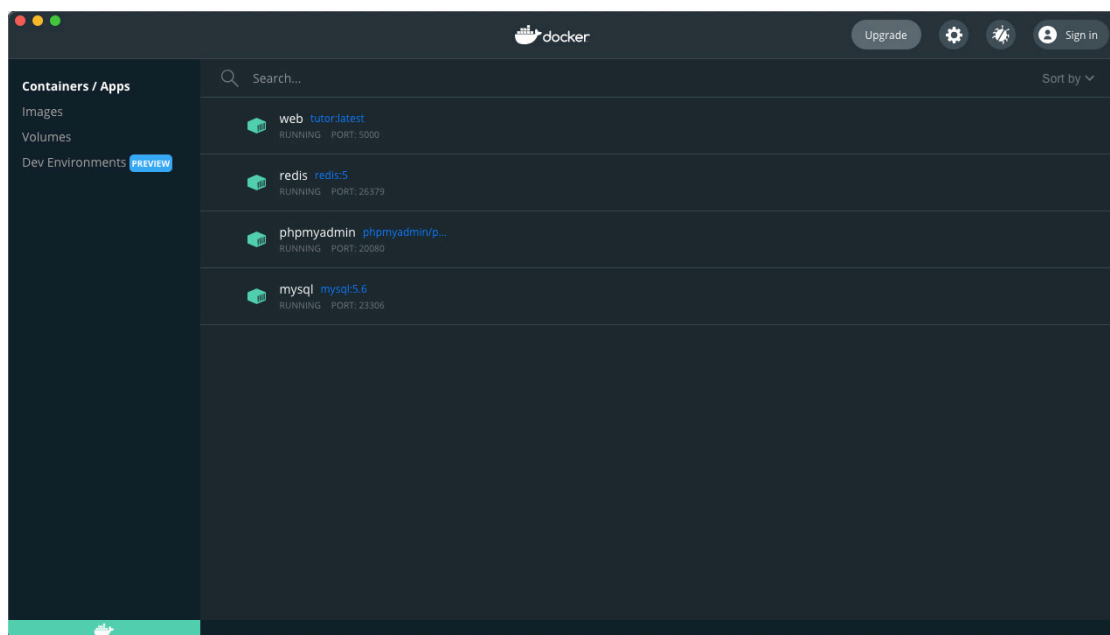
Link: <http://tutor.ecfun.cc:5001/>

### 8.1 Configuration Environment

#### 8.1.1 Docker

Docker is a virtual environment container, which can package the development environment, code, configuration files and so on into this container, and publish and apply to any server. Docker is similar to Git and can be changed to the previous version at any time, which is convenient for developers to modify during development. The deployment of tutor website in Docker is more convenient for the maintenance and modification of code and database in the later period. The following figure shows the files deployed in Docker:

**Figure 26: The Docker interface**



### **8.1.2 Redis**

Redis (Remote Dictionary Server) is an open-source key-value database written in ANSI C language, providing multiple languages. (Redis, 2021). Redis not only maintains the simple and fast characteristics of key-value database, but also absorbs some advantages of relational database. This places it between the relational and key-value databases. The website will use Redis to store the contents of chats and read/unread status of messages.

## **8.2 Register**

Since the website divides users into parents and teachers, some personal details need to be filled in when registering, and the information required to be filled in is slightly different. The collected information is divided into two types and stored in Teacher Table and Parent Table respectively in the back end. The rest of the information is stored in a common User Table. The purpose of this step is to easily obtain the corresponding information when the order details page displays the information later. Since the logic of the codes for registration of teachers and parents is basically the same, the only difference between the codes is the information obtained. Therefore, Figure 27 only shows the back-end logic of registration of parents:

Figure 27:

```
def register_parents():
    if session.get('is_login'):...
    else:
        # Get the form request from the front end 从前端获取表单请求
        username = request.args.get('username', default=None, type=str)
        password = request.args.get('password', default=None, type=str)
        password2 = request.args.get('password2', default=None, type=str)
        birthday = request.args.get('birthday', default=None, type=str)
        grade = request.args.get('grade', default=None, type=str)
        subject = request.args.get('subject', default=None, type=str)
        location = request.args.get('location', default=None, type=str)
        locationName = request.args.get('locationName', default=None, type=str)
        locationNameEN = request.args.get('locationNameEN', default=None, type=str)
        qid = request.args.get('qid', default=None, type=str)
        answer = request.args.get('answer', default=None, type=str)

        # Determine whether the front end has committed data 判断前端是否提交了数据
        if not all([username, password, password2, birthday, grade, subject, location, qid, answer, locationName, locationNameEN]):
            return jsonify({'code': StatusCode.LOGIN_INFO_EMPTY[0], 'msg': StatusCode.LOGIN_INFO_EMPTY[1]})
        # Verify the password twice 验证两次密码是否输入一致
        if password != password2:
            return jsonify(
                {'code': StatusCode.REGISTER_PASSWORD_NOT_MATCH[0], 'msg': StatusCode.REGISTER_PASSWORD_NOT_MATCH[1]})
        if not validate(birthday):
            return jsonify({'code': StatusCode.TIME_FORMAT_ERROR[0], 'msg': StatusCode.TIME_FORMAT_ERROR[1]})
        # Find if the username exists 查找这个用户名是否存在
        user = User.query.filter_by(username=username).first()
        # Exist, indicating registration 存在, 提示注册过了
        if user:
            return jsonify(
                {'code': StatusCode.REGISTER_ACCOUNT_EXISTS[0], 'msg': StatusCode.REGISTER_ACCOUNT_EXISTS[1]})
        else:
            uid = uuid.uuid4().hex[:8]
            data1 = User(uid=uid,
                        username=username,
                        password=password,
                        type='2',
                        location=location,
                        qid=qid,
                        answer=answer,
                        locationName=locationName,
                        locationNameEN=locationNameEN,
                        avatar=STUDENT_AVATAR,
                        timestamp=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime()))
            data2 = Parent(uid=uid,
                          birthday=birthday,
                          grade=grade,
                          subject=subject)

            db.session.add(data1)
            db.session.add(data2)
            db.session.flush()
        return jsonify({'code': StatusCode.REGISTER_SUCCESS[0], 'msg': StatusCode.REGISTER_SUCCESS[1]})
```

### 8.2.1. Fill in the address

Considering that the following pages need to be sorted according to the distance of geographical location, when filling in the address of the registration page, I chose the API of Google Map, where users can mark positions on the map to generate latitude and longitude. See Figure 28 for the code part.

Figure 28:

```
function initMap() {
  var map = new google.maps.Map(document.getElementById('map'), {
    center: LAND,
    restriction: {
      latLngBounds: BOUNDS,
      strictBounds: false,
    },
    zoom: zoom
  });
  google.maps.event.addListener(map, 'click', function (e) {
    for (var i = 0; i < markers.length; i++) {
      markers[i].setMap(null);
    }
    $("#location_msg" + index).val("{ _(\"Selected\") }");
    var latlng = e.latLng.toString().replace("(", "").replace(")", "").split(",");
    $("#location" + index).val(e.latLng)
    var myLatLng = {lat: parseFloat(latlng[0]), lng: parseFloat(latlng[1])};
    my_gaode_geocoder(myLatLng)
    var marker = new google.maps.Marker({
      position: e.latLng,
      map: map,
    });
    markers.push(marker);
  });
}

function openMap(id) {
  index = id;
  $("#btnMap").click();
}
</script>
<script async defer
  src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDwn6zZtvdIsNLC-ofFa7dA6PWNKvd0LU&callback=initMap">
```

Since the website is only for Shanghai at present, I have made the map display only in Shanghai. However, in the code, I have considered that the function of switching regions will be implemented in the later update iteration, so the function of switching regions is also put into the code but not enabled. See Figure 29 for the code.

**Figure 29:**

```
// Only the Shanghai area is retained 只保留上海地区
{#var area_data = "{ request.cookies.get('locale') }"};#}
var area_data = "zh";
var index = 1;
if (area_data == "en") {
  // Map selection range 地图选点范围
  var BOUNDS = {
    north: 59,
    south: 50,
    west: -7,
    east: 1,
  };
  var LAND = {lat: 54, lng: -2};
  var zoom = 6;
} else {
  // Map selection range 地图选点范围
  var BOUNDS = {
    north: 31.8,
    south: 29.5,
    west: 120.5,
    east: 123,
  };
  var LAND = {lat: 31.214562, lng: 121.483398};
  var zoom = 9;
}
var markers = [];
```

Considering that the user needs to see the address text and not the coordinates on the front end, the developers used the GaoDe and Youdao apis, which convert coordinates into addresses and translate the addresses into English. See Figure 30 & Figure 31 for the code.

**Figure 30:**

```
<script>
function my_gaode_geocoder(myLatLng) {
  AMap.plugin('AMap.Geocoder', function () {
    new AMap.Geocoder().getAddress([myLatLng.lng, myLatLng.lat], function (status, result) {
      if (status === 'complete' && result.info === 'OK') {
        // 'result' indicates the detailed information of the corresponding geographical location. result为对应的地理位置详细信息
        $("#locationName").val(result.geocode.formattedAddress);
        // transcript 翻译
        trans(result.geocode.formattedAddress);
      }
    })
  })
}
</script>
<script src="https://webapi.amap.com/maps?v=1.4.15&key=8534f7015d5ba34fb222fdc1520eb143"></script>
```

Figure 31:

```
function trans(content) {
    var appKey = '4f79786445085e87';
    var key = 'x9l0WolUarohXWuYlBaKv0PifnL0mQ6q';
    var salt = (new Date()).getTime();
    var curtime = Math.round(new Date().getTime() / 1000);
    var query = content;

    var from = 'zh-CHS';
    var to = 'en';
    var str1 = appKey + truncate(query) + salt + curtime + key;
    var sign = CryptoJS.SHA256(str1).toString(CryptoJS.enc.Hex);
    $.ajax({
        url: 'http://openapi.youdao.com/api',
        type: 'post',
        dataType: 'jsonp',
        data: {
            q: query,
            appKey: appKey,
            salt: salt,
            from: from,
            to: to,
            sign: sign,
            signType: "v3",
            curtime: curtime,
        },
        success: function (data) {
            if(data.errorCode == "0"){
                $("#locationNameEN").val(data.translation[0]);
            }else{
                $("#locationNameEN").val();
            }
            console.log(data);
        }
    });
}

function truncate(q) {
    var len = q.length;
    if (len <= 20) return q;
    return q.substring(0, 10) + len + q.substring(len - 10, len);
}
```

## 8.3 Login

Users can log in after registration. Users not only need to fill in the username and password registered at that time, but also need to fill in the graphic verification code, which can effectively ensure the security of the website. In addition, you can also



check whether login remains or not. This function is mainly to enable users to directly jump to the homepage of the website without logging in again after closing the browser. The code is shown in Figure 32:

**Figure 32:**

```
def login():
    # Check whether you have logged in 判断是否已经登陆
    if session.get('is_login'):
        # Have logged in jump to the home page 已经登陆过 跳转到首页
        return jsonify({'code': StatusCode.LOGIN_SUCCESS[0], 'msg': StatusCode.LOGIN_SUCCESS[1]})
    else:
        # Get the form request from the front end 从前端获取表单请求
        username = request.args.get('username', default=None, type=str)
        password = request.args.get('password', default=None, type=str)
        remember = request.args.get('remember', default=None, type=str)
        verifycode = request.args.get('verifycode', default=None, type=str)

        if not all([username, password, remember, verifycode]):
            return jsonify({'code': StatusCode.LOGIN_INFO_EMPTY[0], 'msg': StatusCode.LOGIN_INFO_EMPTY[1]})
        # Check whether the verification code is correct and convert it to lowercase 判断验证码是否输入正确, 转换为小写
        if session.get('image'):
            if session.get('image').lower() != verifycode.lower():
                return jsonify({'code': StatusCode.LOGIN_VERIFY_FAIL[0], 'msg': StatusCode.LOGIN_VERIFY_FAIL[1]})
        else:
            return jsonify(
                {'code': StatusCode.LOGIN_VERIFY_IMAGE_FAIL[0], 'msg': StatusCode.LOGIN_VERIFY_IMAGE_FAIL[1]})

        # Check whether the User exists by using the User Table 用户表, 数据表实体类, 查找用户是否存在
        user = User.query.filter_by(username=username).first()
        if not user:
            return jsonify(
                {'code': StatusCode.LOGIN_ACCOUNT_NOT_FOUND[0], 'msg': StatusCode.LOGIN_ACCOUNT_NOT_FOUND[1]})
        else:
            # The user exists, verifies the password, and reads the user password of the database 用户存在, 验证密码, 读取数据库的用户密码
            if user.password == password:
                if user.status == '1':
                    # Session global variable. session会话持久化, 全局变量
                    session.permanent = True
                    session['username'] = user.username
                    session['uid'] = user.uid
                    session['type'] = user.type
                    session['is_login'] = '1'
                    if remember == '1':
                        session['remember_username'] = user.username
                        session['remember_password'] = user.password
                        session['remember'] = 1
                    else:
                        session['remember_username'] = ""
                        session['remember_password'] = ""
                        session['remember'] = 0
                else:
                    return jsonify(
                        {'code': StatusCode.ACCOUNT_EXCEPTION[0], 'msg': StatusCode.ACCOUNT_EXCEPTION[1]})
            else:
                return jsonify({'code': StatusCode.LOGIN_PASSWORD_ERROR[0], 'msg': StatusCode.LOGIN_PASSWORD_ERROR[1]})
    return jsonify(
        {'code': StatusCode.LOGIN_SUCCESS[0], 'msg': StatusCode.LOGIN_SUCCESS[1], 'data': session.get('uid')})
```

### 8.3.1 Verification code

Figure 33 is the part of the code that generates the image verification code. Figure 34 stores the verification code in String format in the back-end Session:

Figure 33:

```
def generate_code_image():
    code = ''.join(random.sample(string.ascii_letters + string.digits, 4))
    width, height = 120, 50
    im = Image.new('RGB', (width, height), 'white')
    font = ImageFont.truetype('static/assets/fonts/arial.ttf', 40)
    draw = ImageDraw.Draw(im)
    for item in range(4):
        draw.text((5 + random.randint(-3, 3) + 23 * item, 5 + random.randint(-3, 3)),
                  text=code[item], fill=(random.randint(32, 127), random.randint(32, 127), random.randint(32, 127)),
                  font=font)
    for i in range(2):
        x1 = random.randint(0, width / 2)
        y1 = random.randint(0, height / 2)
        x2 = random.randint(0, width)
        y2 = random.randint(height / 2, height)
        draw.line((x1, y1), (x2, y2), fill='black', width=1)
    im = im.filter(ImageFilter.GaussianBlur(radius=1.5))
    return im, code
```

Figure 34:

```
def get_code_image():
    # Give the front end picture and reserve the back end picture content string 给前端的图片和保留在后端的图片内容字符串
    image, code = generate_code_image()
    # Images are written in binary form 图片以二进制形式写入
    buf = BytesIO()
    image.save(buf, 'jpeg')
    buf_str = buf.getvalue()
    # Return buf_str as response and set the header field 把buf_str作为response返回前端，并设置首部字段
    response = make_response(buf_str)
    response.headers['Content-Type'] = 'image/gif'
    # Store the captcha string in the session 将验证码字符串储存在session中
    session['image'] = code
    return response
```

### 8.3.2 Log out

When the user wants to log out from the home page, click the 'Log out' button to return to the login interface. For those users who previously chose the "Keep me as signed in" button, after they click 'Log out', the page will automatically fill in the username and password in the login page. Since the username and password entered by the user who previously selected "Keep me as signed in" are stored in the Session, this operation is implemented through the Session. The code can be seen in Figure 35.

**Figure 35:**

```
def logout():
    # Operations on the session 对session的操作
    # Save the remembered user name and password 先对记住的用户名密码保存
    try:
        username = session['remember_username']
        password = session['remember_password']
        remember = session['remember']
        # Clear all records 清除全部的记录
        session.clear()
        # Remember the user name and password 记住用户名密码
        session['remember_username'] = username
        session['remember_password'] = password
        session['remember'] = remember
        return jsonify({'code': StatusCode.LOGOUT_SUCCESS[0], 'msg': StatusCode.LOGOUT_SUCCESS[1]})
    except Exception as e:
        return jsonify({'code': StatusCode.LOGOUT_SUCCESS[0], 'msg': StatusCode.LOGOUT_SUCCESS[1]})
```

## 8.4 Release Information & Looking for a tutor

The main purpose for users to enter the tutoring website is to release tutoring information or find tutoring. Because the website divides users into teachers and parents, the page they enter after logging in is different. For example, the information released by parents is displayed after the teacher logging in. I stored the information filled by the two users in the release information page in two database tables respectively for convenience of later call. Since fetching addresses is an external API, a default address information is set to prevent data loss. The following figure shows the parent Posting information and the back-end code that stores the Posting information for display in the front end (no longer shown due to the similar logic of the parent Posting information back-end code).

Figure 36:

```
def teacher():
    # 判断是否已经登陆
    if not session.get('is_login'):
        # 未登陆
        return jsonify({'code': StatusCode.AUTHENTICATION_FAILED[0], 'msg': StatusCode.AUTHENTICATION_FAILED[1]})
    else:
        # 从前端获取表单请求
        introduction = request.args.get('introduction', default=None, type=str)
        grade = request.args.get('grade', default=None, type=str)
        subject = request.args.get('subject', default=None, type=str)
        pway = request.args.get('pway', default=None, type=str)
        year = request.args.get('year', default=None, type=str)
        price = request.args.get('price', default=None, type=str)

        if not all([introduction, grade, subject, pway, year]):
            return jsonify({'code': StatusCode.LOGIN_INFO_EMPTY[0], 'msg': StatusCode.LOGIN_INFO_EMPTY[1]})

        try:
            uid = session.get('uid')
            user = User.query.filter(User.uid == uid).first()
            if user:
                location = user.location
                locationName = user.locationName
                locationNameEN = user.locationNameEN
            else:
                location = "(31.18404711618375, 121.48052978515625)"
                locationName = "上海市徐汇区枫林路街道东安一村(东安路)"
                locationNameEN = "Dong 'an Yi Village (Dong 'an Road), Fenglin Road, Xuhui District, Shanghai"
            location_array = location.replace("(", "").replace(")", "").split(",")
            data = PostTeacher(uid=uid,
                              introduction=introduction,
                              grade=grade,
                              subject=subject,
                              pway=pway,
                              location=location,
                              locationName=locationName,
                              locationNameEN=locationNameEN,
                              longitude=float(location_array[1]),
                              latitude=float(location_array[0]),
                              year=year,
                              price=price,
                              time=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime()))
            db.session.add(data)
            db.session.flush()
            return jsonify({'code': StatusCode.POST_SUCCESS[0], 'msg': StatusCode.POST_SUCCESS[1]})
        except Exception as e:
            return jsonify({'code': StatusCode.POST_FAILED[0], 'msg': StatusCode.POST_FAILED[1]})
```

### 8.4.1 Update & Delete Released Information

Considering that users need to modify or even delete the published information, the functions of deleting and updating are simply added in the back end. Since the logic of releasing information is different between teachers and parents in information processing, two logic of updating information are written respectively. The following three Figure show the back-end implementation:

Figure 37:

```
def delete():
    # Check whether you have logged in 判断是否已经登陆
    if not session.get('is_login'):
        # Not log in 未登陆
        return jsonify({'code': StatusCode.AUTHENTICATION_FAILED[0], 'msg': StatusCode.AUTHENTICATION_FAILED[1]})
    else:
        uid = session.get('uid')
        type = session.get('type')
        # Get the form request from the front end 从前端获取表单请求
        id = request.args.get('id', default=None, type=int)

        if type == '2':
            PostParent.query.filter(PostParent.id == id).update({'status': '0'})
        elif type == '3':
            PostTeacher.query.filter(PostTeacher.id == id).update({'status': '0'})
        return jsonify({'code': StatusCode.DELETE_SUCCESS[0], 'msg': StatusCode.DELETE_SUCCESS[1]})
```

Figure 38:

```
def update_parents():
    # Check whether you have logged in 判断是否已经登陆
    if not session.get('is_login'):
        # Not log in 未登陆
        return jsonify({'code': StatusCode.AUTHENTICATION_FAILED[0], 'msg': StatusCode.AUTHENTICATION_FAILED[1]})
    else:
        # Get the form request from the front end 从前端获取表单请求
        id = request.args.get('id', default=None, type=int)
        situation = request.args.get('situation', default=None, type=str)
        grade = request.args.get('grade', default=None, type=str)
        subject = request.args.get('subject', default=None, type=str)
        pway = request.args.get('pway', default=None, type=str)
        year = request.args.get('year', default=None, type=str)
        requirement = request.args.get('requirement', default=None, type=str)
        price = request.args.get('price', default=None, type=str)

        if not all([id, situation, grade, subject, pway, requirement]):
            return jsonify({'code': StatusCode.LOGIN_INFO_EMPTY[0], 'msg': StatusCode.LOGIN_INFO_EMPTY[1]})

        try:
            PostParent.query.filter(PostParent.id == id).update({
                'situation': situation,
                'grade': grade,
                'subject': subject,
                'pway': pway,
                'year': year,
                'requirement': requirement,
                'price': price
            })
            return jsonify({'code': StatusCode.UPDATE_SUCCESS[0], 'msg': StatusCode.UPDATE_SUCCESS[1]})
        except Exception as e:
            return jsonify({'code': StatusCode.UPDATE_FAILED[0], 'msg': StatusCode.UPDATE_FAILED[1]})
```

Figure 39:

```
def update_teacher():
    # Check whether you have logged in 判断是否已经登陆
    if not session.get('is_login'):
        # Not log in 未登陆
        return jsonify({'code': StatusCode.AUTHENTICATION_FAILED[0], 'msg': StatusCode.AUTHENTICATION_FAILED[1]})
    else:
        # Get the form request from the front end 从前端获取表单请求
        id = request.args.get('id', default=None, type=int)
        introduction = request.args.get('introduction', default=None, type=str)
        grade = request.args.get('grade', default=None, type=str)
        subject = request.args.get('subject', default=None, type=str)
        pway = request.args.get('pway', default=None, type=str)
        year = request.args.get('year', default=None, type=str)
        price = request.args.get('price', default=None, type=str)

        if not all([id, introduction, grade, subject, pway, year]):
            return jsonify({'code': StatusCode.LOGIN_INFO_EMPTY[0], 'msg': StatusCode.LOGIN_INFO_EMPTY[1]})

        try:
            PostTeacher.query.filter(PostTeacher.id == id).update({
                'introduction': introduction,
                'grade': grade,
                'subject': subject,
                'pway': pway,
                'year': year,
                'price': price
            })
            return jsonify({'code': StatusCode.UPDATE_SUCCESS[0], 'msg': StatusCode.UPDATE_SUCCESS[1]})
        except Exception as e:
            return jsonify({'code': StatusCode.UPDATE_FAILED[0], 'msg': StatusCode.UPDATE_FAILED[1]})
```

## 8.5 Chat Room

Web online chat room is one of the most difficult parts in the development process. The chat room uses flask-Socketio at the back end and socket.io.js at the front end. After initiating a chat, it needs to check whether the two parties have established a chat room. The information is stored in the MySQL database. If a chatroom has not been created before, the chatroom is created first and the user ids (uid) of the communication parties are stored in the chatroom table. Then flask-Socketio chat channels are set up to build rooms. Flask-socketio can only receive instant messages, that is, messages cannot be received when the other party is not online. Therefore, the developers introduced Redis database for the storage of chat records. The Redis database is selected because of its high read and write performance and is suitable for high-frequency read and write scenarios of chats. When storing chat records, the CHATRoom ID is used as the key, and the UID, message content, and message time



are stored as values in the form of List data. This allows it to query chat history by room ID. At the same time, the room ID is exclusive to the communication parties, so there will be no data error problem. In this way, when initiating a chat, if both parties have already established a chatroom, the chat records will be read in Redis database directly according to the chatroom id, and drawn in the front end, scrolling to the bottom of the chat data. This solves the problem of not being able to receive offline messages. When both communication parties are in chat rooms, instant communication can be carried out through flask-Socketio and messages can be stored in Redis at the same time, and the front-end uses Vue to dynamically update the chat messages. Figure 40 shows the code for the above back-end.

**Figure 40:**

```
# Join a chat room 加入聊天室
@socketio.on('join', namespace='/chat')
def on_join(message):
    room_id = int(message['room_id'])
    session['room_id'] = room_id
    join_room(room_id)

# Send a message 发送消息
@socketio.on('send', namespace='/chat')
def test_message(message):
    # Insert the chat information into the Redis cache database 将聊天信息插入Redis缓存数据库
    # Sender, message, time 发送者, 信息, 时间
    r.rpush(session['room_id'],
            json.dumps({
                'uid': session.get('uid'),
                'content': message['content'],
                'time': message['time'],
                'status': message['status'] if 'status' in message else '0'
            }))
    # Set the number of unread messages 设置未读消息条数
    msg_count_key = 'count_room_{uid}'.format(str(session['room_id']), session['uid'])
    msg_count = r.get(msg_count_key)
    if msg_count:
        msg_count = msg_count.decode()
        r.set(msg_count_key, int(msg_count)+1)
    else:
        r.set(msg_count_key, 0)

    order = None
    if "##86#" in message['content']:
        id = int(message['content'].replace("##86###", ""))
        orders = Order.query.filter(Order.id == id).first()
        order = {
            "id": orders.id,
            "puid": orders.puid,
            "tuid": orders.tuid,
            "pid": orders.pid,
            "time": orders.time.strftime('%Y-%m-%d %H:%M:%S'),
            "pway": "Online" if orders.pway == "1" else "Offline",
            "location": orders.location,
            "ptype": orders.ptype,
            "price": orders.price,
            "subject": orders.subject,
            "schedule": orders.schedule,
            "status": orders.status,
            "from_status": orders.from_status,
            "to_status": orders.to_status
        }

    emit('my_response', {'data': message, 'order': order, 'room': session['room_id']})

# Exit chat room 退出聊天室
@socketio.on('leave', namespace='/chat')
def on_leave(message):
    room_id = int(message['room_id'])
    leave_room(room_id)

# Close the page and disconnect the socket 关闭页面 断开socket连接
@socketio.on('disconnect', namespace='/chat')
def test_disconnect():
    uid = session.get('uid')
    r.delete('status_uid.' + uid)
    print('Client disconnected')
```

According to the function of chat room, the developers also developed message notification and display unread message function. This feature stores the number of unread messages in the Redis database, where the key is `count_room_*_uid_*`, the first `*` indicates the room ID of the chat room, and the second `*` indicates the user ID. Therefore, in flask-Socketio message sending, the record UID is the SENDER's UID, that is, a user sends N messages, and the value stored in Redis is N. When the offline user opens the message notification, it reads the number of messages of the UID of the other party to be communicated. When clicking to get chat history, update the number of unread messages to 0. For example, user UID A communicates with user UID B, and when USER B is offline, user A sends A message to user B, creating A chat room with id 10000. When A sends the first message to B, the number of unread messages is saved in Redis. The key is `count_room_10000_uid_A` and the value is 1, that is, an unread message. In this case, A sends another message to B, and the value of `count_room_10000_uid_A` changes to 2. When B goes online, open the message notification, read the UID of all users who have established chat rooms with B in Mysql, and query the chat records in Redis by the chat room ID, that is, key is the chat room ID. Query the number of unread messages by uid, that is, `count_room_{chat room ID}_uid_{user ID}`, that is, `count_room_10000_uid_A`. When user B clicks to view the message, change the value of `count_room_10000_uid_A` to 0. See Figure 39 for the back-end code.



Figure 41:

```
def friend():
    # Check whether you have logged in 判断是否已经登陆
    if not session.get('is_login'):
        # Not log in 未登陆
        return jsonify({'code': StatusCode.AUTHENTICATION_FAILED[0], 'msg': StatusCode.AUTHENTICATION_FAILED[1]})
    else:
        room = ChatRoom.query.filter(
            or_(ChatRoom.uid1 == session.get('uid'), ChatRoom.uid2 == session.get('uid'))).all()
        chatlist = []
        for index, room_ in enumerate(room):
            redis_message = r.lrange(room_.id, 0, -1)
            if redis_message:
                msg = json.loads(r.lrange(room_.id, 0, -1)[-1])
            else:
                msg = None

            if room_.uid1 == session.get('uid'):
                uid = room_.uid2
            elif room_.uid2 == session.get('uid'):
                uid = room_.uid1
            user_to = User.query.filter(User.uid == uid).first()
            if msg and user_to:
                room[index].msg = msg
                room[index].to = user_to
                chatlist.append(room[index])
        chatlist.sort(key=sortTime, reverse=True)

        data = []
        for item in chatlist:
            unread = r.get('count_room_{0}_uid_{1}'.format(str(item.id), item.to.uid))
            data.append({
                'user': {
                    'id': item.to.id,
                    'uid': item.to.uid,
                    'username': item.to.username,
                    'type': item.to.type,
                    'avatar': item.to.avatar,
                    'timestamp': item.to.timestamp
                },
                'msg': {
                    'uid': item.msg['uid'],
                    'content': _("[Order message]") if "*#06#" in item.msg['content'] else item.msg['content'],
                    'time': item.msg['time']
                },
                # 查询未读消息条数
                'unread': int(unread.decode()) if unread else 0
            })
        return jsonify({'code': StatusCode.GET_SUCCESS[0],
                        'msg': StatusCode.GET_SUCCESS[1],
                        'data': data})
```

## 8.6 Generate Order

Order generation is only possible in the chat room. When the edited order is inserted into the database, the ID of the order is obtained which will be sent to the other party in the form of chat messages. In this way, the order can be sent in the chat box. How to draw a text message as an order confirmation box is determined according to the custom character. In the code, `*#06#` is used as the prefix of the order confirmation message and `@@@` is used as the separator, that is, an order message is `*#06#`

@@@OrderID. That is to say, when the code prefix is encountered, the message is transformed into an order message, the message is segmented with '@@@', the order message is queried in the database with OrderID, and then it is drawn in the front end. As shown in Figure 42.

**Figure 42:**

```
orders_dict = {}
for item in orders:
    orders_dict[item.id] = item

message_list = []
for item in message:
    orders = None
    if "*#06#" in item['content']:
        id = int(item['content'].replace("#06#@", ""))
        if orders_dict[id].pway == "1":
            pway = "Online" if request.cookies.get('locale') == 'en' else "线上"
        else:
            pway = "Offline" if request.cookies.get('locale') == 'en' else "线下"
        orders = {
            "id": orders_dict[id].id,
            "puid": orders_dict[id].puid,
            "tuid": orders_dict[id].tuid,
            "pid": orders_dict[id].pid,
            "time": orders_dict[id].time.strftime('%Y-%m-%d %H:%M:%S'),
            "pway": pway,
            "location": orders_dict[id].location,
            "ptype": orders_dict[id].ptype,
            "price": orders_dict[id].price,
            "subject": orders_dict[id].subject,
            "schedule": orders_dict[id].schedule,
            "status": orders_dict[id].status,
            "from_status": orders_dict[id].from_status,
            "to_status": orders_dict[id].to_status
        }
    message_list.append({
        'uid': item['uid'],
        'content': item['content'],
        'time': item['time'],
        'orders': orders,
        'status': item['status'] if 'status' in item else '0'
    })
```

In order to confirm and modify the order status of both parties, the status field is added in the chat message and stored in Redis, which is used to identify the order status. Therefore, a piece of chat data has the following format:

### Ordinary message

```
{'uid': 'A', 'content': 'hello', 'time': '2021-09-10 10:10:10', 'status': '0'}
```

### Order message

```
{'uid': 'A', 'content': '*#06#@@@1212', 'time': '2021-09-10 10:10:10', 'status': '1'}
```

**The status flag has the following three states. Different order buttons are displayed according to this parameter:**

- 1 is ordering.
- 2 indicates modification.
- 3 is consent.

## 8.7 Search Keyword & Filter bar

### 8.7.1 Searching Keyword

Users can fill in keywords in the search bar and click “search”. Fuzzy matching of the published information will be displayed in the published information list. There are two types of users and the two types of published information content are stored in different database tables, so there will be two types of back-end logic in search. Fuzzy matching function basically depends on 'like' in sqlalchemy to achieve. In particular, when releasing information, the subject options are multi-selected, and the storage form of back-end database is split by "@@@" (as a Split character), so a separate logic is written when doing the fuzzy matching. Since the back-end logic of parents and teachers is basically the same, Figure 43 only shows the part of the search logic code of teachers:

**Figure 43:**

```
teacher_list = PostTeacher.query \
    .filter(or_(*[PostTeacher.subject.like("%" + item + "%") for item in subject.split("@@@")])) \
    .filter(and_(PostTeacher.introduction.like("%" + introduction + "%"),
        PostTeacher.grade.like("%" + grade + "%"),
        PostTeacher.pway.like("%" + pway + "%"),
        PostTeacher.year.like("%" + year + "%"),
        PostTeacher.price.like("%" + price + "%"),
        PostTeacher.status == '1')) \
```

### 8.7.2 Filter bar

After the user selecting the option in the filter bar, the publish information list will display the appropriate information based on the filter result. Search and filtering can

be done simultaneously. There are 5 screening items in total, among which 'distance' is the one with relatively complicated logic. Only after selecting the "Distance" filter, the secondary filter will become a similar range of "1km" or "5km", while the other filter items except "Distance" only have positive and reverse order, so the developer wrote "distance" as a separate judgment condition. The 'distance' function invokes the latitude and longitude stored in the database table of both users at registration, and then calculates the distance between them. Figure 44 is the part of the back-end logic code that the teacher uses when using the filter bar (which also contains the search):

**Figure 44:**

```

if type == '2': # "2" The type in the database table User is teacher
    dict_word = ['time', 'grade', 'year', 'price', 'distance']
    if len(dict_word) <= sortBy:
        return jsonify({'code': StatusCode.ORDER_BY_PARAM_ERROR[0], 'msg': StatusCode.ORDER_BY_PARAM_ERROR[1]})
    recommend_teacher_list = []
    # Query Information List 查询信息列表
    if sortBy == 4: # '4' = distance
        if not longitude or not latitude:
            return jsonify({'code': StatusCode.MISS_PARAMETERS[0], 'msg': StatusCode.MISS_PARAMETERS[1]})
        # Calculate how many meters near latitude and longitude sort 计算经纬度 附近多少米 排序
        latlng = func.acos(func.sin(func.radians(
            float(latitude))) * func.sin(
            func.radians(PostTeacher.latitude)) + func.cos(
            func.radians(float(latitude))) * func.cos(
            func.radians(PostTeacher.latitude)) * func.cos(
            func.radians(PostTeacher.longitude) - (func.radians(
            float(longitude))))) * 6371
        teacher_list = PostTeacher.query \
            .filter(or_(*[PostTeacher.subject.like("%" + item + "%") for item in subject.split("@@@")])) \
            .filter(and_(PostTeacher.introduction.like("%" + introduction + "%"),
                PostTeacher.grade.like("%" + grade + "%"),
                PostTeacher.pway.like("%" + pway + "%"),
                PostTeacher.year.like("%" + year + "%"),
                PostTeacher.price.like("%" + price + "%"),
                PostTeacher.status == '1')) \
            .filter(latlng <= sortOrder) \
            .order_by(asc(latlng)) \
            .paginate(curPage, pageSize, error_out=False)
    else:
        attr = getattr(PostTeacher, dict_word[sortBy]) # Get sort properties 获取排序属性
        teacher_list = PostTeacher.query \
            .filter(or_(*[PostTeacher.subject.like("%" + item + "%") for item in subject.split("@@@")])) \
            .filter(and_(PostTeacher.introduction.like("%" + introduction + "%"),
                PostTeacher.grade.like("%" + grade + "%"),
                PostTeacher.pway.like("%" + pway + "%"),
                PostTeacher.year.like("%" + year + "%"),
                PostTeacher.price.like("%" + price + "%"),
                PostTeacher.status == '1')) \
            .order_by(attr.asc() if sortOrder == 1 else attr.desc()) \
            .paginate(curPage, pageSize, error_out=False)
    paginate = Paginate(teacher_list)

```

## 8.8 Chinese and English mode

The website page realizes the function of Chinese-English conversion with the Flask-babel extension tool which is used to mark the text content to be switched in the form of 'gettext()' in the front end, then the .pot file is generated by the command of back-end Babel, and the content to be translated is manually input inside the file to generate a .mo file. At this time, the translation function of the configured 'en' and 'zh' in the front-end page is realized.

**Figure: 45. Babel gets the functions and configuration items of the current language options**

```
def get_locale():
    cookie = request.cookies.get('locale')
    if cookie in ['zh', 'en']:
        return cookie
    return request.accept_languages.best_match(app.config.get('BABEL_DEFAULT_LOCALE'))

BABEL_DEFAULT_LOCALE = 'en'

BABEL_DEFAULT_TIMEZONE = 'UTC'
```

**Figure: 45. Partial translation content of .po file.**

```
#: api/chat.py:213
msgid "[Order message]"
msgstr "[订单信息]"

#: templates/chat.html:61 templates/index.html:468
msgid "Write your text here....."
msgstr "输入信息"

#: templates/edit.html:17 templates/edit.html:59
msgid "Update User Info"
msgstr "更新用户信息"

#: templates/edit.html:21 templates/edit.html:63
msgid "UserName"
msgstr "用户名"

#: templates/edit.html:25 templates/edit.html:67 templates/login.html:114
#: templates/login.html:161
msgid "Your Child's Birth Day"
msgstr "孩子生日"
```

## 9. Analysis

The goal of this project is to develop a tutoring platform to help college students find part-time tutoring. The results of the functional test and the specific implementation of Aim and Objective of the project will be analyzed below.

### 9.1 Test

The tutor platform will pass the functional test before release and rely on developers to test the process of each function. The number of every test is set to 10. As long as one test fails, the form will be recorded as 'Fail '. The test function will be tested according to 7.1 Use Case Design of this project. If the test is 'Fail ', the code will be iterated accordingly. The browser used in this test is Google Chrome Version 93.0.4577.63 and the operating system is MacOS Big Sur 11.0.1. The following 11 tables are detailed test results for developers.

| Test Case 1 - Register   |  |  |           |
|--------------------------|--|--|-----------|
| <b>Success criteria:</b> | Users can successfully register for a platform account; Distinguish between teacher and parent user types.                                       |  |           |
| <b>Preconditions:</b>    | Be able to successfully connect to the Internet.   |  |           |
| Step Number              | Procedure  | Response   | Pass/Fail |
| 1                        | Enter the web address in the browser.  | The login page is displayed. By default, the login page is selected in the TAB bar.  | Pass      |
| 2                        | Switch TAB bar to Teacher Registration/Student Registration  | Successfully jump from the login page to the registration page on the same page.   | Pass      |
| 3                        | Parent/Teacher Registration page<br>Fill in the following details<br>Username:<br>Password:<br>Confirm password:<br>Grade:<br>Subject Selection: | If the form is filled in with complete information, it automatically switches to the login TAB.<br><br>When the correct photo is uploaded, the page displays the image. (Teacher registration page only) | Pass      |

|                            |   |  |      |
|----------------------------|---|--|------|
|                            | Upload academic:<br>certificate :(registered for teachers<br>only)<br>Upload life photos :(registered for<br>teachers only)<br>Click register complete.       |  |      |
| 4                          | If there is an unfilled item or more in the form of the registration page, click the registration complete button   | A prompt box pops up on the web page, and the text is "Information cannot be Empty".                         | Pass |
| 5                          | Enter the registered username in the username of the registration page, and fill in the details of other forms, click the button to complete registration.    | A prompt box pops up on the web page, and the copy is "Account has been registered".                         | Pass |
| 6                          | The password on the registration page is not the same as the confirm password, and other forms are filled in details, click the registration complete button. | A prompt box appears on The web page with the text "The Passwords do not match twice"                        | Pass |
| 7                          | If the correct photo cannot be uploaded on the teacher registration page  | A prompt box appears on the web page with the text “The uploaded image is not in the correct format or size’ | Pass |
| <b>Comment: None</b>       |   |  |      |
| <b>Related Tests: None</b> |   |  |      |
| <b>Author:Sicheng Zhou</b> |   | <b>Checker:Sicheng Zhou</b>  |      |

| Test Case 2 – Login      |  |   |           |
|--------------------------|--|---|-----------|
| <b>Success criteria:</b> | You can log in to the web page using the username and password                     |   |           |
| <b>Preconditions:</b>    | Registered   |   |           |
| Step Number              | Procedure  | Response  | Pass/Fail |
| 1                        | Enter the correct username, password and authentication and click the login button | Enter the next page, and parents and teachers jump to different pages | Pass      |
| 2                        | Click the image verification code  | Successfully switch text on the picture.                              | Pass      |
| 3                        | Parents and teachers jump to different pages                                       | If the form is filled in with complete information, it                | Pass      |



|  |  |  |      |
|--|--|--|------|
|  |  | automatically switches to the login TAB.   |      |
| 4  | Check the "Keep me as signed in" option  | After successful login, close the browser and open it again to directly access the main page. The account is the one you have logged in to before. | Pass |
| 5  | Click the reset password button  | Skip to the corresponding page   | Pass |
| 6  | If the entered username does not exist and the verification code is incorrect, click Login.        | A dialog box is displayed stating "The Account does not exist". The verification code is refreshed.  | Fail |
| 7  | Enter the correct username, incorrect password, and correct verification code, and click Login.    | A dialog box is displayed indicating Password error. The verification code is refreshed.   | Fail |
| 8  | Enter the correct username, incorrect password, and incorrect verification code, and click Login.  | A dialog box is displayed indicating Password error. The verification code is refreshed.   | Fail |
| 9  | The user enters the correct username, correct password, wrong verification code, and clicks login. | A dialog box is displayed, stating Verify code Error. The verification code is refreshed.  | Fail |
| <b>Comment: The main reasons for the failure are incorrect priority of the prompt and missing small functions.</b> |  |  |      |
| <b>Related Tests: Test Case 1</b>  |  |  |      |
| <b>Author:Sicheng Zhou</b>   |  | <b>Checker:Sicheng Zhou</b>  |      |

| Test Case 3 - Release tutoring information |  |   |           |
|--|--|---|-----------|
| <b>Success criteria:</b>                   | Teachers/parents can successfully post tutoring information  |   |           |
| <b>Preconditions:</b>                      | Is logged in   |   |           |
| Step Number                                | Procedure  | Response  | Pass/Fail |
| 1  | Teachers/parents click icon in the lower right corner or click on the upper Tab bar to publish                     | Skip to the information release page  | Pass      |
| 2  | Teachers/parents fill in the corresponding form on the complete release information page and click Save to submit. | Skip to view, and above pops up the message 'The publication of the information was successful' copy. | Fail      |
| 3  | The form for teacher/student to fill in the release information page has   | A prompt box is displayed on the web page. The text is "Information                                   | Fail      |



|                                   |  |  |      |
|-----------------------------------|--|--|------|
|                                   | one or more places left unfilled, click Save submit. | Cannot be Empty", and the page is not refreshed.                       |      |
| 5                                 | Click the back button                                | Skip to the home page, the previously filled information is not saved. | Pass |
| <b>Comment: None</b>              |  |  |      |
| <b>Related Tests: Test Case 2</b> |  |  |      |
| <b>Author:Sicheng Zhou</b>        |  | <b>Checker:Sicheng Zhou</b>  |      |

| Test Case 4 - Looking for a tutor |   |   |           |
|-----------------------------------|---|---|-----------|
| <b>Success criteria:</b>          | Parents can view information posted by teachers and see the details page  |   |           |
| <b>Preconditions:</b>             | Parent login only   |   |           |
| Step Number                       | Procedure   | Response  | Pass/Fail |
| 1                                 | After logging in successfully, parents can browse the brief tutor information published by all teachers on the main page, and each information has an independent box, click the box. | Skip to the detailed tutor information page released by the teacher | Fail      |
| 2                                 | Click the comment button on the page  | You can view information about past parent reviews                  | Pass      |
| 3                                 | Click the chat button on the page   | Pop-up chat box   | Pass      |
| 4                                 | Click the back button   | Skip to the main page   | Pass      |
| <b>Comment: None</b>              |   |   |           |
| <b>Related Tests: Test Case 3</b> |   |   |           |
| <b>Author:Sicheng Zhou</b>        |   | <b>Checker:Sicheng Zhou</b>   |           |

| Test Case 5 - Looking for a student |   |  |           |
|-------------------------------------|---|--|-----------|
| <b>Success criteria:</b>            | Teachers can view information posted by parents and view the details page   |  |           |
| <b>Preconditions:</b>               | Teacher login only  |  |           |
| Step Number                         | Procedure   | Response   | Pass/Fail |
| 1                                   | After the teacher logging in successfully, the brief tutor information published by all parents can be viewed on the main page, | Skip to the detailed tutoring information page released by the parents | Fail      |

|                                   |  |                             |      |
|-----------------------------------|--|-----------------------------|------|
|                                   | and each information has an independent box, click the box |                             |      |
| 2                                 | Click the chat button on the page                          | Pop-up chat box             | Pass |
| 3                                 | Hit the back button  | Skip to the main page       | Pass |
| <b>Comment: None</b>              |  |                             |      |
| <b>Related Tests: Test Case 3</b> |  |                             |      |
| <b>Author:Sicheng Zhou</b>        |  | <b>Checker:Sicheng Zhou</b> |      |

| Test Case 6 – Chat Room                 |   |   |           |
|---|---|---|-----------|
| <b>Success criteria:</b>                | Parents and teachers can communicate with each other online   |   |           |
| <b>Preconditions:</b>                   | You have logged in, and the other user t has published information  |   |           |
| Step Number                             | Procedure   | Response  | Pass/Fail |
| 1                                       | The user clicks on the chat hover button/order bar in the lower right corner  | The chat window is displayed  | Pass      |
| 2                                       | User enters any character in the input field of chat box, click send button   | The characters entered by the user are stored in Redis Database and displayed in the chat box message bar.                                    | Pass      |
| 3                                       | The user who receives the message clicks the chat icon in the lower right corner/the user who has opened the chat box receives the message directly | The corresponding information is received from Redis Database and displayed in the chat box message bar                                       | Pass      |
| 4                                       | Chat records are saved permanently  | When the user closes the chat window and opens it again, the previous chat content will still be displayed in the message bar of the chat box | Pass      |
| <b>Comment: None</b>                    |   |   |           |
| <b>Related Tests: Test Case 4&amp;5</b> |   |   |           |
| <b>Author:Sicheng Zhou</b>              |   | <b>Checker:Sicheng Zhou</b>   |           |

| Test Case 7 - Generate order |   |          |           |
|------------------------------|---|----------|-----------|
| <b>Success criteria:</b>     | After successful communication between parents and teachers, an agreement is generated and the credentials can be left. |          |           |
| <b>Preconditions:</b>        | Logged in and only generated in a chat room.  |          |           |
| Step Number                  | Procedure   | Response | Pass/Fail |

|                                   |  |  |      |
|-----------------------------------|--|--|------|
| 1                                 | Click the Generate order button inside the chat box  | The order editing window is displayed  | Pass |
| 2                                 | After completing the form information, click the Submit button                               | <p>The order details are generated in the chat box message bar of the order originator with a modification button;</p> <p>The order taker's chat box message bar will generate order details with confirmation and modification buttons</p> <p>..</p>  | Pass |
| 3                                 | Click the order details modification button in the chat box message bar                      | The order editing window is displayed, and the information previously filled in is saved in the form by default.   | Pass |
| 4                                 | When the information is modified, click to complete the modification                         | <p>The chat box message bar of the person who modifies the order will generate the order details again, with the modification button attached.</p> <p>The chat box message bar for the recipient of the modified order generates the order details again, with confirmation and modification buttons</p> | Pass |
| 5                                 | The person who accepts or modifies the order clicks the agree button in the chat message bar | The order is generated, the button below the order details is hidden in the chat box and message bar of both parties, and it is changed into the non-clickable copy "completed".   | Pass |
| 6                                 | When both parties have confirmed the order, click to download the PDF.                       | The browser can download a PDF file of the order, which contains the correct order information.  | Pass |
| <b>Comment: None</b>              |  |  |      |
| <b>Related Tests: Test Case 6</b> |  |  |      |
| <b>Author:Sicheng Zhou</b>        |  | <b>Checker:Sicheng Zhou</b>  |      |

| Test Case 8 - Evaluation and Scoring |   |  |           |
|--------------------------------------|---|--|-----------|
| <b>Success criteria:</b>             | Ability to score and evaluate generated orders                                    |  |           |
| <b>Preconditions:</b>                | Only the parents have logged in and the order has been generated                  |  |           |
| Step Number                          | Procedure   | Response   | Pass/Fail |
| 1                                    | When both sides of the order are confirmed, parents click the comment button      | Teachers of this order can be scored and commented (the evaluation can be left blank). After the successful evaluation, a copywriting prompt bar of "successful comment" will pop up | Pass      |
| 2                                    | Parents view the tutor information list and details page click the comment button | At the bottom of the tutor information page, you can view the comments and ratings of the teacher by historical users.   | Pass      |
| <b>Comment: None</b>                 |   |  |           |
| <b>Related Tests: Test Case 7</b>    |   |  |           |
| <b>Author:Sicheng Zhou</b>           |   | <b>Checker:Sicheng Zhou</b>  |           |

| Test Case 9 - Re-edits the personal information |   |  |           |
|---|---|--|-----------|
| <b>Success criteria:</b>                        | You can modify the information you fill in when registering                                     |  |           |
| <b>Preconditions:</b>                           | Already logged in   |  |           |
| Step Number                                     | Procedure   | Response   | Pass/Fail |
| 1   | After logging in successfully, click the user button in the TAB bar at the top of the home page | The user profile window is displayed to view the entered personal information                                    | Pass      |
| 2   | Click the Edit button in the profile window   | Skip to edit profile page  | Pass      |
| 3   | After modifying the information, click Save   | The information filled in during registration can be modified, if successful, then skip to the website home page | Pass      |
| 4   | If one or more of the forms for teacher/student to fill in the release                          | A prompt box is displayed on the web page. The text is "Information  | Pass      |

|   |  |  |  |
|---|--|--|--|
|   | information page are left unfilled, and click Save | Cannot be Empty", and the page is not refreshed. |  |
| <b>Comment: None</b>                    |  |  |  |
| <b>Related Tests: Test Case 1&amp;2</b> |  |  |  |
| <b>Author:Sicheng Zhou</b>              |  | <b>Checker:Sicheng Zhou</b>                      |  |

| Test Case10 - Search Keyword& Sorting |  |   |           |
|---------------------------------------|--|---|-----------|
| <b>Success criteria:</b>              | Be able to successfully screen various types through the screening bar, successfully search corresponding information through the keyword. |   |           |
| <b>Preconditions:</b>                 | Is logged in   |   |           |
| Step Number                           | Procedure  | Response  | Pass/Fail |
| -                                     | When the user selects ascending/descending by time at the filter bar sort (single option)  | The published information list is displayed in ascending order and the page is refreshed immediately  | Pass      |
| -                                     | When the user selects ascending/descending by grade at the filter bar sort (single)  | The release information list will be displayed automatically in order from small to large/from large to small according to the grade filled in by the user release information, and the page will be refreshed immediately          | Pass      |
| -                                     | When the user selects ascending by experience/descending by sorting in the filter bar (single)   | The list of published information is automatically displayed in order from small to large/from large to small according to the teaching experience filled in by users' published information, and the page is refreshed immediately | Pass      |
| -                                     | When the user selects ascending/descending by price at the filter bar sort (single option)   | The list of published information is automatically displayed in order from small to large/from large to small according to the price range filled in by users' published information, and the page is refreshed immediately         | Pass      |
| -                                     | When the user selects the distance at the filter sort and by the distance range (single)   | The list of published information is automatically arranged according to the geographical location filled in by users and the distance between teachers and users from near too   | Pass      |

|  |  |  |      |
|--|--|--|------|
|  |  | far, and the page is refreshed immediately   |      |
| -  | When the user checks online/offline teaching in the screening bar (multiple selection or no selection)                 | The information list will automatically display only online/offline teaching information. If both are selected or not selected, all teaching information will be displayed, and the page will be refreshed immediately | Pass |
| -  | When the user checks the student grade range in the filter bar (single)  | The published information list only shows the information published by teachers in the selected grade range. The page will be refreshed immediately  | Pass |
| -  | When the user checks the number of years of teaching experience of the teacher in the screening bar (single selection) | The published information list only displays the information published by teachers with selected years of teaching experience, and the page will be refreshed immediately  | Pass |
| -  | When the user checks the price range in the filter bar (single)  | The list of published information only shows the information published by teachers in the selected price range, and the page will be refreshed immediately   | Pass |
| -  | When the user checks the desired subject in the filter bar (single)  | The list of published information only shows the information published by teachers of selected subjects, and the page will be refreshed immediately  | Pass |
| <b>Comment: The above tests are disordered and can be operated at will, and the user does not affect each other when using each screened component. The intersection display is formed according to the final screening results.</b> |  |  |      |
| <b>Related Tests: Test Case</b>  |  |  |      |
| <b>Author:Sicheng Zhou</b>   |  | <b>Checker:Sicheng Zhou</b>  |      |

| Test Case 11 - Chinese and English mode |   |
|---|---|
| <b>Success criteria:</b>                | Web pages can be switched between Chinese and English to achieve a bilingual website. |
| <b>Preconditions:</b>                   | Internet Connected  |

| Step Number                     | Procedure  | Response   | Pass/Fail |
|---------------------------------|--|--|-----------|
| 1                               | Click the Chinese button in the upper right corner | The page is refreshed and all copywriting on the page is changed to Chinese. | Fail      |
| 2                               | Click the English button in the upper right corner | The page is refreshed and changed from Chinese to English.                   | Pass      |
| <b>Comment: None</b>            |  |  |           |
| <b>Related Tests: Test Case</b> |  |  |           |
| <b>Author:Sicheng Zhou</b>      |  | <b>Checker:Sicheng Zhou</b>  |           |

## 9.2 Aims and objectives retrospective

Aims and objectives initially established are compared and analyzed with the results of current project realization to determine project integrity.

### 9.2.1 Aim - To develop a platform for the distribution of tutors for college students and parents of teenagers in Shanghai

This is the macro-objective of the project. The main function of the project is consistent with the purpose of this goal. In some details, considering the local website service policy, the external interface is matched with the local website service and will not be shielded.

### 9.2.2 Objective - Provide a service platform for users to publish and receive information about tutors

The specific functions of this Objective have been basically realized in the test table of 9.1 above. In the later iteration process, functions such as chat and order generation are added to increase users' satisfaction when using the website. The developer also added some personalized functions such as Chinese and English switching, address selection visualization. The usability of these functions is unknown before the website

is released, the pros and cons of these personalized functions can only be judged by viewing the data from the back end after the website is released.

### **9.2.3 Objective - Use an advanced Web development technology and from which learn some new knowledge and skills**

Developers in the process of website development front-end mainly used Html, vue.js, Json, jQuery. Python-flask is used for the back end. These are the dominant web development tools in the current environment. Due to the need for functionality, the developers also input a number of never-used plug-ins to improve the integrity of functionality, such as Flask. Socketio, Flask. Babel, jQuery.ajax, etc. These development tools and plug-ins are all learned and used by developers through reading books and browsing programming platforms in the process of development to achieve the original Objective.

### **9.2.4 Objective - Design a simple website interface so that users can quickly and efficiently use the platform's various functions**

In order to maintain the consistency and simplicity of the appearance of the web page, the developer adopted the front-end development framework of Bootstrap and basically realized the initial Objective. However, due to the time limitation, there was no further improvement in the processing of some details, such as the floating button, the web adaptive mobile browser, etc.

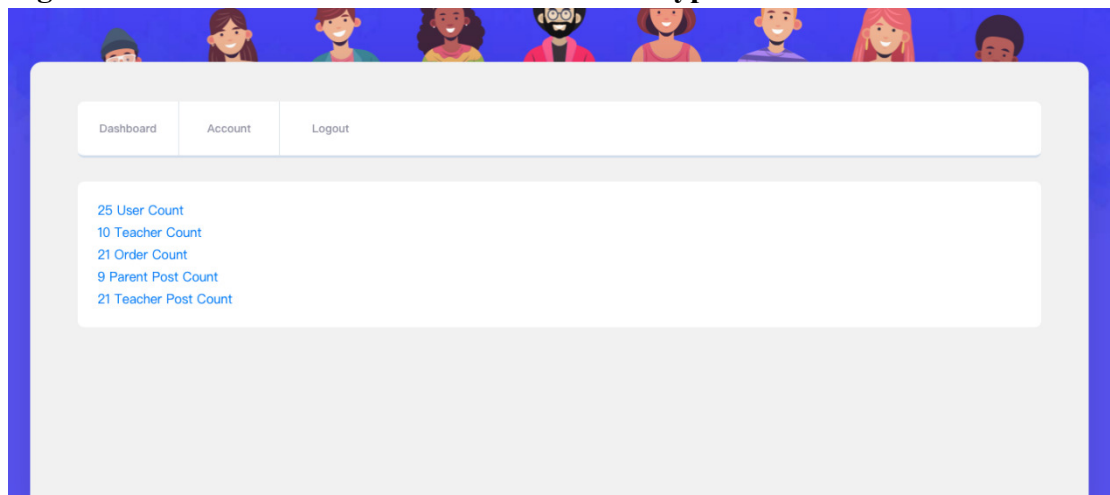
### **9.2.5 Objective - Create a simple management platform that can effectively verify the authenticity of college students**

There was no design and development management page in the first version of the site development (because the developers thought it could be easily managed in a database). As the version of tutor website is constantly iterating and updating, the

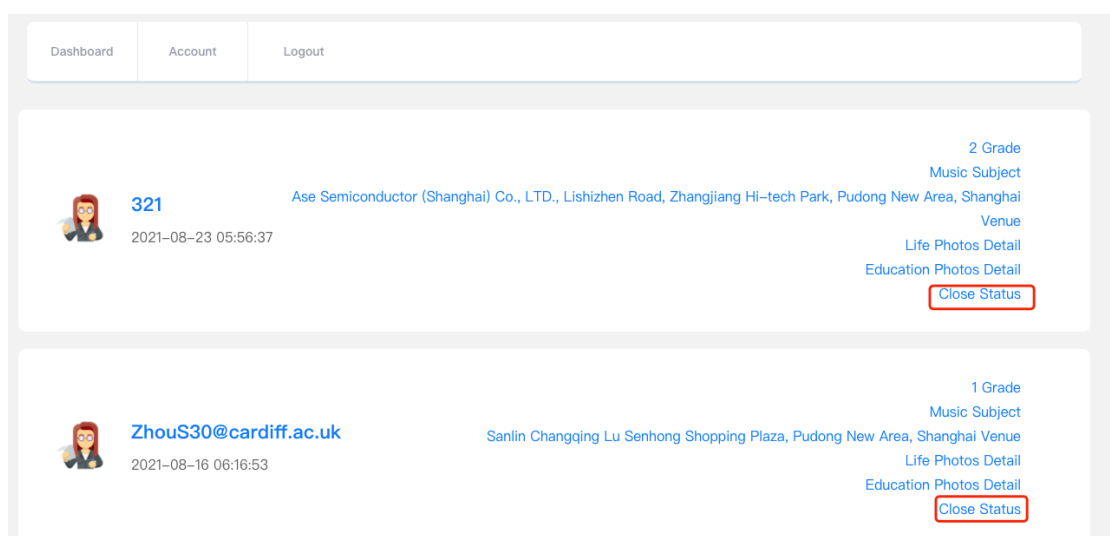


number of back-end database is also beyond the original estimate. Considering the importance and convenience of managing the background, in the final version, the developer developed a separate page to manage the users within the site. This management page can display the specific number of users on the website, view the degree certificates uploaded by teachers, and ban or unseal all accounts. In the end, the developers achieved their initial objectives, but the design and layout of the management page was not very beautiful. The page interface is shown in the following two pictures:

**Figure: 46. Collect statistics on the number and type of users**



**Figure:47. View user information and banned user pages**



### 9.2.6 Objective - Ensure that the website has great security measures

**which can protect the personal information that users fill in on the platform**

Developers have added prerequisites to judge ‘whether to log in or not’ in each function of the back end of the website (see Figure 48 for example) to ensure the security of the routing address of the website. When users log in, the function of graphic verification code is opened to prevent criminals from cracking the password of users inside the website. This basically ensures the security of user information, but more perfect security measures and functions cannot be fully realized in a limited time.

**Figure 48:**

```
def data():  
    if not session.get('is_login'):  
        # Not log in 未登陆  
        return jsonify({'code': StatusCode.AUTHENTICATION_FAILED[0], 'msg': StatusCode.AUTHENTICATION_FAILED[1]})  
    else:
```

### 9.2.7 Objective - Test the platform continuously to ensure the

**satisfaction of users when using the website after the completion of project**

The developer only tested the basic functions of the website, but this was far from enough in the process of product testing. The contents that could not be tested included performance test, security test, compatibility test and stability test. Due to the limitation of development time, the test did not reach the original Objective.

## 10. Conclusion

According to the full text of this article, developers have gone from having a simple plan to considering all the required goals and problems; From my shallow understanding of educational background to the literature and policy of relevant educational background in-depth exploration; From the confusion of website development process to the determination of website development Methodology; From simple design wireframes to detailed page layout modifications; From the design of website functional requirements to the database design and function association; From the lack of code writing ability to the master of the front-end and back-end of various development tools; From the simple development of the website to the final skilled testing and analysis.

Developer tried to develop and improve around the theme of "A Tutoring Website Aimed at Teenagers in Shanghai". The main development and completion of the tutor website functions are:

Login and Register, Release Tutoring Information, Looking for a teacher/teenagers, Generate order, Chat Room, Evaluation and Scoring, Search Keyword and Filter bar, Chinese and English mode.

The whole paper proves that it is feasible to develop a fully functional tutoring website.

### 10.1 Background Summary

Before the development of tutoring website, the developer of this paper first reviewed the history and current situation of the development of tutoring; Secondly, because the website developed is related to the field of education, and the site was released in Shanghai, China, so I investigated and consulted the local education policies to ensure the compliance of various functions of the website. Thirdly, I conducted a detailed

investigation on seven popular tutor websites in China, sorted out and compared the internal functions of the websites, and finally conducted SWOT analysis on these tutor websites, so as to determine the necessity of various functions of the tutor website. Finally, I deeply understand and learn the basic website development tools, and briefly explain the meaning of each website development tool, to consolidate the foundation and efficiency of code development.

## **10.2 Methodology Summary**

In this paper, the developer looked up various development process models. Finally, the development process Model of this project selected Waterfall Model and briefly explained the theory of the Model. According to each step of the model and the goal of tutoring website development, the developer of this paper thinks and explains the general planning and thinking of each step.

## **10.3 Requirement Specification Summary**

Before writing the code of the function of the tutor website, the developer of this paper will first classify the requirements as functional and non-functional. After that, the priority of functional requirements was sorted into 'Must', 'Should' and 'Not'. Within the limited development time, the priority order of development and iteration were clarified. Finally, list the non-functional requirements. The Requirement Specification not only clarifies the development direction of website functions for developers, but also helps the subsequent design process.

## **10.4 Design Summary**

In the design of tutoring website development, the developers of this paper first designed the Use Case diagram in the first part, listing the Use scenarios and main functions of tutoring website, in order to draw the boundaries of tutoring website Use. Then all the functions in the Use Case diagram are designed, the realization goal of

each function is determined, and the flow chart of each function is drawn. In the second part of the design process, the developer of this paper designed the web page using wirefraps, presenting each page in the form of lines and boxes in the mind of the developer, reasonably arranging content and functions on each page, establishing the basic structure of the page for the development stage. The last part of the design is the database design, because in the development of the synchronous creation of the database table will lead to a large degree of complexity of the database, so before the development of the code, the developer will design carefully the database. According to the requirements of each function, and through continuous optimization as well as the addition and deletion of code later, 10 database tables were finally designed, and the structure and annotations in each table were very clear described.

## **10.5 Implementation Summary**

In the implementation part, the developers first understand and choose a virtual environment -Docker, which can not only control the version in the process of project iteration, but also easily deploy to the external server after the completion of development. After that, I studied Redis in depth and described its main functions in order to realize the internal chat function of the website. Finally, the developer wrote the code of each function, and analyzed the main logic code of each function with screenshots.

## **10.6 Future Work**

In this paper, the developer thinks that although the main functions of the tutoring website developed at present have been realized, there are still many deficiencies. If there is enough time and user experience as well as feedback after the release, in the future, developers will make improvements based on the three parts: page aesthetics, functional integrity and website sustainability

- Page aesthetics
  - Synchronize all pages of the site in detail to maintain page consistency.
  - Work with all browsers on the phone.
  - Improve layout based on continuous feedback from users.
  
- Functional integrity
  - Optimize the code for each function in detail.
  - Allow users to use more secure verification codes during login.
  - Be able to limit the number of times each user can register (for example, use email, mobile phone number, etc., as the username, the registration can be successful only after obtaining and filling in the verification code).
  - Limit the length and format of a user's password during registration.
  - Users can modify the profile picture.
  - Users can send emojis or pictures in the chat room.
  - When the order is online, it can provide a live class platform within the website.
  - Enable both parties to sign on the middle line of the order file when generating the order.
  - Add optional subjects in real time according to the current popular subjects in the region.
  - When the user base is large, research a set of appropriate recommendation algorithm to intelligently recommend teachers/students for users.
  - Add a forum page.
  
- Website sustainability
  - Write detection codes for each function of the website and count the usage of users.
  - Quickly solve user feedback information.
  - Expand the geographic coverage of the website.
  - Work with local charity.

- The background can automatically review the degree certificates uploaded by teachers.
- Improve page loading speed.

## 11. Reflection

Education is and will be a central issue for a nation. When some unpredictable disasters strike, such as COVID-19, we need not only to face them calmly, but more importantly, not to stop our steps. For me, what affected me most was that I could not continue my study, but fortunately, thanks to the development of the Internet, I could take classes remotely. But my friends have not been so lucky. They have lost their income and jobs, and the epidemic has forced them to stay indoors. I deeply sympathize with this, and this also formed my original intention to do this tutor website. As a non-profit website, I believe its emergence will help more college students like them.

I will discuss the reflection of this tutor website in three stages. Respectively: before project development, during project development and after project completion. Before the development of the project, I was unable to study in the UK due to COVID-19. As the original theme of the project was website development in Cardiff, I was very worried about the background investigation at the initial stage of the project development. It was not until my mentor Dr.Daniela came along that she agreed with me to change the website development area in my project and gave me great support. There was a long vacation before the project was decided, during which I realized that the skills I had learned in the previous courses were far from enough for the knowledge I needed to finish the project. So, I started to watch a lot of website development videos from websites, which not only consolidated the website development tools I learned in previous courses: Python, Javascript, CSS, Html, Mysql, but also learned vue. js, Redis, and various Flask plug-ins. During the boring process of watching videos and practicing codes, code knowledge was far more difficult than I expected, which made me very troubled and I thought about giving up more than once. However, after I communicated with my classmates through chatting tools, I found that I was not the only one who was studying hard. Most of my classmates had never been exposed to computer science before the course, just like



me. After that, we would discuss the knowledge we learned in the week through text or video conference every week and solve the doubts for each other.

After the project was determined, I conceived simply the functions required by the tutor website. In the second meeting with my tutor, I was recognized by my tutor, and my tutor also planned for me about the functions should be completed first in the code development process, which was very detailed. I think this plan is very useful, and it saves me a lot of time and effort in developing the code, because every feature has a necessary correlation, and arranging it in the wrong order can lead to more code changes. In the in-depth study of the topic background, I first applied the methods taught in the Course of Cardiff Library and used the document manager to quickly and effectively sort and classify the literature I have read, which saved me a lot of time in later paper writing. Despite an efficient software, I still have a lot of deficiencies in the background research. Because I use most of the time in the development and study of link, I only read literature related to education history and policy, and there is still no strong evidence for some arguments, some literature is very old as well.

In the process of project development, the development of each function of the website cost me a lot of time, especially every change of the database will involve a lot of code. This made me realize that the early stage of database design is very critical, and the completeness of consideration determines the efficiency of development.

I was particularly impressed by the function of "arranging the information posted by users according to the distance". The first thing that came to my mind was to calculate the distance between users by using the longitude and latitude of users' addresses. However, when I designed the blank bar for users to fill in their addresses when registering, I only let users input their addresses. Nevertheless, obtaining users' geographical location by IP address not only has a large error, but also involves users' personal privacy. Therefore, obtaining the longitude and latitude of users became my primary goal at that time. After thinking and searching for relevant methods, I decided to use Google Map to select points instead of manual input by users. When I called

the Google Maps API, I found that the Google Maps API can only fetch latitude and longitude, not convert latitude and longitude to address (for a fee), but the front page display needs address. So I used the API of AutoNavi in China to convert the user's latitude and longitude into Chinese address, while converting the Chinese address into an accurate English address, which was another difficult problem. Finally, I used YouDao's API to implement it.

Due to a small feature that resulted in a series of feature changes, I used a total of 3 external apis in this part, all of which were free. I thought the search process was fun and anxious, but when I actually managed to implement these features, I felt an instant boost of accomplishment. This allowed me to explore a different kind of fun in the boring process of code development, and also made me full of confidence and expectation for the future work in this field.

At the tutor meeting after the completion of the project, I demonstrated the specific functions of tutor website. My tutor gave very reasonable suggestions and found some loopholes in the functions. It is worth thinking about how, in the future, if I were an independent software developer, I would give myself effective advice after a project is developed. I think this is a necessary part of the project development, rather than just doing it after the project is finished. This made me realize that for every project I develop or work on, I always need to be skeptical of the project in order for it to be sustainable.

In general, I have basically accomplished the objectives of the project in the short period of three months, and acquired a lot of knowledge and skills, which I think is worthy of pride. But my impetuous mentality and shallow knowledge need to be improved constantly. In my opinion, this project still has a great space for development, and all my friends are very supportive of this project. Though the difficulties it will face in the future will be more onerous, I believe that any project will be successful as long as we carry out the development with the spirit of learning and exploration, and plan and test with a meticulous and questioning attitude.

## References

- Balaji, S. and Sundararajan Murugaiyan, M. 2012. WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. *JITBM & ARF* 2(1), pp. 26-30.
- Bray, M. 2007. The shadow education system: Private tutoring and its implications for planners. *Compare: A Journal of Comparative Education* 36(4), pp. 515-530
- Chinese Government. 2021. *Wide-ranging guideline to rein in tutoring sector*. Available at: [http://english.www.gov.cn/policies/policywatch/202107/26/content\\_WS60fdf199c6d0df57f98dd91b.html](http://english.www.gov.cn/policies/policywatch/202107/26/content_WS60fdf199c6d0df57f98dd91b.html) [Accessed 14 September 2021].
- Harrington, J.L. 2016. *Relational database design and implementation*. Morgan Kaufmann.
- Hille, A., Spieß, C.K. and Staneva, M. 2016. More and more students, especially those from middle-income households, are using private tutoring. *DIW Economic Bulletin* 6(6), pp.63-71
- Hussein, M.G. 1987. Private tutoring: A hidden educational problem. *Educational studies in mathematics* 18(1), pp. 91–96.
- Manhas, J. 2017. Initial framework for website design and development. *International Journal of Information Technology* 9(4), pp.363-375
- Murawska, B. and Putkiewicz, E. 2005. Warsaw: The Institute of Public Affairs. *Private tutoring in Poland [in Polish]*.
- Nia, M.R. and Shokouhyar, S. 2020. Analyzing the effects of visual aesthetic of Web pages on users' responses in online retailing using the VisAWI method. *Journal of Research in Interactive Marketing*.
- Shanghai Statistics Department. 2020. *STUDENTS ENROLLMENT BY VARIOUS SCHOOLS (1978~2019)* Available at: <http://tjj.sh.gov.cn/tjnj/nj20.htm?d1=2020tjnjen/E2103.htm> [Accessed 14 September 2021].
- Svensson, R.B., Sprockel, Y., Regnell, B. and Brinkkemper, S. 2010,. Cost and benefit analysis of quality requirements in competitive software product management: A case study on the QUPER model. *2010 Fourth International Workshop on Software Product Management IEEE*, pp. 40-48.

Vijayasarathy, Leo, R. and Charles, W.B. 2016. Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software* 33(5), pp. 86–94.

Wright, E., Moosung L. and Siyuan F. 2018. Shadowing the International Baccalaureate: Private Supplementary Tutoring for the Diploma Programme in China. *Educational Research for Policy and Practice* 17(2), pp. 127–43.

Zhang, P. and Von Dran, G.M. 2000. Satisfiers and dissatisfiers: A two-factor model for website design and evaluation. *Journal of the American society for information science*, 51(14), pp.1253-1268