# Distribution of Work Document

## Group: Tomato killer

URL of Github：<https://github.com/vriusc/Intelligent-Tutoring-System>

|  |  |  |
| --- | --- | --- |
| Name | Distribution | Github |
| Yongru Qian | 19% | vriusc |
| Xiaochi Li | 19% | xiaochili, ubuntu |
| Alvaro | 19% | AlvaroPatty UCD |
| Jiahao Liu | 15% | everEvergolw |
| Chih-Yu Huang | 15% | BOBOeternal |
| Weimin Sun | 13% | RichMillions1 |

## Contributions

### Yongru Qian

I work as a Team Management，product manager，Back-end JAVA developer on the project.

Team Management: In this part my primary responsibility is to ensure smooth communication and collaboration within the team. I establish and oversee the project timeline, ensuring each task is completed on schedule. I'm in charge of delegating work assignments to team members and scheduling and organizing meetings based on project needs. My goal is to guarantee efficient team operations while meeting all project requirements.

Product manager: In this project, my primary responsibilities include analyzing and determining the direction and functionalities of the project. I streamline and formulate project planning, design its features, and create initial product prototypes. Collaborating with frontend, backend, and CICD teams, we establish a preliminary implementation plan. By taking feedback from within the team, I refine and upgrade the prototype post-implementation. Additionally, in partnership with the product testing team, I ensure the pqrogress and quality of product development, coordinate resources from all sides, and guarantee the project progresses as planned.

Back-end JAVA developer: In this project, I specialize in crafting a robust backend framework utilizing Springboot and MyBatis. Springboot is employed to seamlessly manage frontend requests, while MyBatis facilitates efficient interactions with the database, ensuring smooth data relay back to the frontend. In collaboration with the frontend team, we have standardized API interfaces and data transmissions, and I leverage Postman for meticulous API testing. Engaging with the CICD team, I guarantee precise port forwarding, which is pivotal for the seamless deployment of both backend and frontend components to our servers. Additionally, I collaborate with our database specialists to cohesively design and strategize the database layout, ensuring optimal data formats and tables are in place. Together with our testing team, I remain vigilant in addressing and rectifying any code discrepancies promptly.

### Xiaochi Li

I worked as a DevOps Engineer on the project and was responsible for the full infrastructure and continuous integration continuous delivery.

CI/CD:In this part, my job has two components. First: Deploying Jenkins tool to the server. Second: Writing Jenkins files for each service block. When we click on the build button, Jenkins will go to the corresponding GitHub branch to find the Jenkins file and automatically execute the process in the Jenkins file. After the Jenkins deployment is complete, the time cost of updating our service is greatly reduced.

Docker: Most of the work is automated by Jenkins. We put a well-written Dockerfile into GitHub, and Jenkins performs builds based on the Dockerfile and uploads images to DockerHub.Docker Hub repository name: xiaochili123.

Kubernetes: This part of the work includes infrastructure build and architecture design. We use Kops to manage the Kubernetes cluster itself and helm to manage the services running in the kubernetes cluster. After Jenkins uploads the image to DockerHub, Jenkins will control the Kops node to run the helm command to update the service. In addition, in order to use kubernetes, we also need to write a lot of kubernetes files, we mainly use the following kinds of files: ingress controller file: responsible for managing the ingress traffic distribution, service file: responsible for the corresponding service is assigned a port within the cluster to access, deployment file: responsible for setting the number of pods to run in the cluster, ensuring that pods can be restarted automatically, and the HPA file: responsible for automatic scaling.

AWS: I manage our AWS cloud server independently, using the following features: EC2 instance, IAM, S3 bucket and Route53. The EC2 instance is the server we are running on, IAM is responsible for administrative privileges, S3 bucket is responsible for storing the learning materials, and Route53 is responsible for binding our domain to the kubernetes cluster.

### Alvaro Mauricio Patty Cruz

I have worked as a Front-end developer, responsible on the design, architecture and development of the UI Interface and web application.  
Design & implementation: On this stage I have designed and choose the programming languages that will have to be implemented, with the help of NPM as a package manager, I have started building the project adding different libraries such ReactJS, Axios, i18Next, etc. And after that starting with the coding of the UI Interface.  
Axios & Postman: During the development stage I had to be in constant communication with the services and the way of testing the new API was using Postman and later for the implementation on the front-end Axios. This was an important library because of the easy accessibility and fast debugging of the API requests.  
I18Next: it was the implementation of this library that made the User interface interactive for different languages, I had the responsibility on the translation of different labels, titles and text to other languages.  
Collaboration: As a developer I had constant collaboration with other parts of the work distribution such as Back-end, AI, Devops, etc. This happened on the development of the project which we had constant communication around the different features to be implemented.

### Jiahao Liu

As the project's Python engineer, I built intelligent API services using Flask, integrating OpenAI GPT and Google Cloud Vision API. This allowed a personalized learning experience for students, enhancing feedback through facial recognition.

With the OpenAI GPT integration, I set up API routes for various tasks like writing feedback and question answering. Through custom OpenAI\_GPT\_Tutor and OpenAI\_Utilities classes, I optimized OpenAI's configurations and feedback generation. Personalized feedback was crafted based on student scores to suit individual learning styles. I also ensured data security by using encryption keys and configuration files for API requests.

Leveraging the Google Cloud Vision API, I designed a real-time facial expression recognition service. To provide instant results, I integrated SocketIO for seamless frontend-backend interaction. I also managed error scenarios, like undetected faces or data conversion issues.

### Chih-Yu Huang

I've served as a database administrator and quality assurance engineer, overseeing database optimization and ensuring our application meets quality standards.

Database Management: As the Database Administrator, I handle the project's data storage, retrieval, and management. I design, implement, and maintain the database architecture and collaborate with developers to meld database functionalities with our app. My role is pivotal in optimizing performance, safeguarding data integrity, and devising backup strategies.

Unit Testing: In my capacity as a JUnit Tester, I verify the software's reliability and functionality. I create and run JUnit test cases to validate software components, ensuring early detection and rectification of errors. My feedback assists developers in refining the codebase to boost software quality.

User Testing: Undertaking User Testing, I guarantee the software aligns with user expectations. I organize tests gauging usability, functionality, and user experience, utilizing real-world scenarios to determine software adaptability. By interpreting user feedback, I highlight improvement areas and work with design and development teams to refine the user interface, heightening overall user satisfaction.

### Weimin Sun

I spearheaded early-stage product design, backend Python development, product testing, and report planning, including user experience and stress testing.

Early-stage product design: I oversaw prototype design, functionality mapping, and the drafting of User Stories, User Case Diagrams, and Descriptions.

User experience testing: I implemented Tree Tests via Optimal Workshop to refine the website sitemap, ensuring intuitive user interaction with the product.

Reporting planning: Managed the creation and structure of interim and final reports and crafted specific PowerPoint presentations.

Python backend: Developed Python services with Flask that interface with the GPT API, listening for requests on a port, interacting with ChatGPT, and returning results.

Stress testing: I headed stress testing, writing json test files, executing server tests, and analyzing the resultant data.

### One more thing

The development of our various features is stored separately in different branches. Following is a description of the branches：

1. Document: evidence of teamwork and related documentation.
2. Backend-GPT: Code for GTP section
3. Database-Dev: code For mysql
4. qianbackground：Code for Java back-end and Junit test
5. Front-End-Dep：Code for Front-end
6. Helm：Kubernetes file
7. Backend-FaceRec：Code For face recognition(on going)
8. Stress-Testing: Code for Stress Testing