

BEIJING-DUBLIN INTERNATIONAL COLLEGE

---

## Project Report

---

*Students:*

Jiahe Zhang 20372118 20205722  
Feihe Huang 20372128 20205735  
Tairan Ding 20372106 20205710  
Yun Zhang 20372105 20205709  
Zhaoyu Fu 20372104 20205708

December 8, 2022



# Contents

<b>1 General Overview</b>	<b>3</b>
1.1 Background & market . . . . .	3
1.1.1 Video demand surges under epidemic as industry applications accelerate penetration . . . . .	3
1.1.2 Video Conferencing Industry Overview . . . . .	3
1.1.3 Internet-based video conferencing . . . . .	3
1.1.4 Video conferencing interview software for programmers hollow . . . . .	4
1.2 Objectives & goals . . . . .	4
1.3 Customer Statement . . . . .	5
1.3.1 Preface . . . . .	5
1.3.2 System specification . . . . .	5
<b>2 Glossary of Terms</b>	<b>6</b>
<b>3 System Requirements</b>	<b>7</b>
3.1 Requirements Models . . . . .	7
3.1.1 Application process of requirements engineering . . . . .	7
3.1.2 Requirements analysis methods . . . . .	7
3.1.3 Application of Sprint planning Meeting . . . . .	8
3.1.4 Application of INVEST . . . . .	9
3.1.5 User story cards examples . . . . .	9
3.2 Functional Requirements . . . . .	11
3.3 Non-functional Requirements . . . . .	12
3.4 Functional Requirement Specification . . . . .	12
3.4.1 Stakeholders . . . . .	12
3.4.2 Functional Requirements Specification . . . . .	13
3.5 Fully dressed use cases & use case diagram . . . . .	14
3.5.1 Use Cases . . . . .	14
3.5.2 Use case Diagram . . . . .	16
3.6 Traceability Matrix . . . . .	17
3.6.1 Requirement Traceability Matrix . . . . .	17
3.6.2 Use Case Traceability Matrix . . . . .	17
3.7 Fully dressed description . . . . .	18
3.7.1 Example1 . . . . .	18
3.7.2 Example2 . . . . .	19
3.7.3 Example3 . . . . .	19
3.8 Scenario . . . . .	20
3.8.1 Scenerio 1-Register and fill resume . . . . .	20
3.8.2 Scenerio 2-Join meeting as Interviewee . . . . .	21
3.8.3 Scenerio 3-Internal evaluation meeting for interviewer . . . . .	22
3.8.4 Scenerio 4-Interviewee check interview outcome and feedback . . . . .	23
<b>4 Effort estimation &amp; risk estimation</b>	<b>23</b>
4.1 Scale estimation . . . . .	23
4.1.1 Function Scores Estimation . . . . .	23
4.1.2 Overview . . . . .	25
4.1.3 Effort estimation . . . . .	25

4.1.4	System functions . . . . .	26
4.2	Progress estimation . . . . .	27
4.2.1	COCOMO . . . . .	27
4.2.2	Calculate . . . . .	27
4.2.3	Disadvantages . . . . .	27
4.2.4	Progress Management . . . . .	28
4.3	Risk estimation . . . . .	28
<b>5</b>	<b>Domain model</b>	<b>30</b>
5.1	Domain Model . . . . .	30
5.2	Domain Model Derivation . . . . .	30
<b>6</b>	<b>System architecture &amp; system design</b>	<b>31</b>
6.1	System architecture . . . . .	31
6.1.1	Architectural representations . . . . .	32
6.1.2	Architectural design decision . . . . .	32
6.2	System design . . . . .	32
6.2.1	Interaction diagram & sequence diagram . . . . .	32
6.2.2	Activity diagrams . . . . .	33
6.2.3	State machine diagram . . . . .	34
6.2.4	Database . . . . .	35
<b>7</b>	<b>Design of Tests</b>	<b>37</b>
7.1	Test case . . . . .	37
7.1.1	Website . . . . .	37
7.1.2	Database . . . . .	38
7.1.3	Meeting . . . . .	38
7.1.4	Whiteboard . . . . .	39
7.1.5	Chatroom . . . . .	39
7.2	Test cover . . . . .	39
<b>8</b>	<b>Project management</b>	<b>39</b>
8.1	Personnel management . . . . .	39
8.1.1	Principle of the distribution of duties . . . . .	39
8.1.2	Individual division of labor . . . . .	40
8.1.3	Collaboration in iteration . . . . .	41
8.1.4	Project specific plan . . . . .	41
8.1.5	Team Charter . . . . .	43
8.2	Software process . . . . .	44
8.2.1	Process Activity . . . . .	44
8.3	Process improvement . . . . .	48
<b>9</b>	<b>Contribution</b>	<b>48</b>
9.1	Final Contribution . . . . .	48
9.2	Development Contribution . . . . .	48
9.3	Report Contribution . . . . .	48
<b>10</b>	<b>Docker</b>	<b>49</b>

<b>11 Reference</b>	<b>50</b>
11.1 Agora . . . . .	50
11.2 AngularJS . . . . .	50
11.3 MaterializeCss . . . . .	50
11.4 PubNub . . . . .	50
11.5 Firebase . . . . .	50

# 1 General Overview

## 1.1 Background & market

### 1.1.1 Video demand surges under epidemic as industry applications accelerate penetration

In 2020, the outbreak of the new coronavirus non-Asian epidemic, remote government, remote medical diagnosis, remote education, remote office, remote interview needs and other video-related applications greatly meet the needs of the epidemic prevention and control, reflecting the potential demand for video communication in corporate offices or industry scenarios is very strong, the epidemic has accelerated the process of video conferencing market education. Videoconferencing suppliers are constantly updating their hardware and software, evolving from traditional videoconferencing, which relies on the use of dedicated decoding and transmission equipment, to internet-based software videoconferencing (the focus of this project), and there are now videoconferencing devices with cloud capabilities, analytics software that collects big data for more personalised services, etc. After the global videoconferencing industry entered the explosive period, the global videoconferencing market size was about US\$31.9 billion in 2012, and with the declining cost of broadband and the gradual improvement of the network environment, the global videoconferencing market size is growing rapidly, and the global videoconferencing market size will be about US\$55 billion in 2020.

### 1.1.2 Video Conferencing Industry Overview

Video conferencing refers to two or more parties, face-to-face communication through communication equipment and networks, and live video, video recording and other forms of the biggest difference is actually interactive, the global popularity and development of video conferencing has gone through five stages from the concept to the commercial budding stage, analog video conferencing segment, private network digital video conferencing stage, IP-based network video conferencing stage and cloud video conferencing stage This project focuses on the core theme of the industry's past development, and the combination of technology and application will become the new direction of the industry's development in the future. For technical reasons, this project focuses on the IP-based network videoconferencing stage and focuses on the close integration of videoconferencing-related technologies with the development of the Internet computer industry and its interview process.

In terms of the application areas and application scenarios of video conferencing, video conferencing can be applied to medical, education, finance and other fields, and the application scenarios mainly include morning visitation, sales meetings, task deployment and work coordination, work reporting and regular meetings.

### 1.1.3 Internet-based video conferencing

Internet-based video conferencing refers to the use of new video capture and transmission equipment, which refers to the real-time transmission of moving images, voice and application

data information between two and more points.

The era of cloud video communication has opened up the market space for video conferencing, greatly reducing the threshold for the use of video conferencing and gradually sinking the general-purpose enterprise meeting scene to medium-sized enterprises. Cloud video has a bit on the mobile Xinghe expansion, prompting the integration of video conferencing with more vertical industry application scenarios, expanding the boundaries and latitude of the video conferencing industry.

The current domestic video conferencing industry players can be divided into three categories: traditional hardware video conferencing vendors, video industry newcomers and giant vendors building online office ecology. Traditional hardware video conferencing vendors include Huawei, ZTE, Polycom, Suzhou KODAK, Viewlink Power, etc.; new entrants in the industry include Little Fish Easylink, GoodVision, WillCom, Yilian Network, Two Six Three, etc.; giant vendors building cloud office ecology include Nail, Tencent Conference, Byte Jump's Flying Book, Huawei's WeLink, the addition of giants brings change to the industry pattern on the one hand, and facilitates the industry ecology on the other. On the other hand, it is conducive to the rapid construction of industry ecology and the innovation and accelerated penetration of vertical industry application scenarios.

#### **1.1.4 Video conferencing interview software for programmers hollow**

The interview process for programmers is not yet available on the market with a targeted, appropriate interview software.

Compared to other industries, the interview process for programmers is slightly different, mainly due to the content of the programmer interview. The general programmer interview process consists of both a technical test and a personality test. Generally based on communication, there are generally no major problems with the initial technical test will relatively on the basis of language development and Web direction will do some in-depth communication, the main purpose is to understand the programmer on the language, development and related knowledge of the degree of understanding and the previous projects done to the general understanding of the situation, while from the communication to understand your work ideas and your technical expertise, so as to facilitate the positioning of you This will enable you to position yourself. Technical interview is mainly for the project perspective on the quality of personnel to examine, including the understanding of software projects, for the development of the work of the view and according to their own work experience analysis of the summary. In this part of the interview, candidates need to have a deeper understanding of their past projects and work experience, as well as a basic concept of project and project management. Therefore, in addition to ensuring that users can communicate with each other in an audio-visual way, there are also targeted components such as code handling and whiteboards to assist with negotiations.

However, there is currently no online videoconferencing application on the market for programmers' interviews that allows them to code and communicate together.

## **1.2 Objectives & goals**

This IP network-based online video conferencing platform for the programmer community seeks to break through the barriers in the video conferencing interview process and achieve the true sense of using just this one platform to make video conferencing, communication, idea discussion and sharing, and collaborative code writing and grading accessible at the same time.

In addition to separate meeting rooms and corresponding features such as voice and video, screen sharing, whiteboards, chat rooms and code editing. There should also be complementary

related services to provide users with more fluid booking and other services. The platform also provides two other functional modules, namely the CV module and the schedule planning module.

After the user clicks to open our platform services, the platform can meet the user to register to have their own account to facilitate the user to carry out related operations. Users can access the CV section to edit their CV and have it seen by others on the platform in the future to improve mutual understanding. In the schedule planning section, users can see a good visualisation of their appointments, which can be added and tied to corresponding dates to make them more comfortable for multiple interviews.

The main section revolves around video conferencing. The popular video conferencing software on the market does not have a service for interviews. This is especially true for programmer interviews, which rely heavily on technical skills in coding and communication. For this type of interview, it is important to focus not only on verbal communication, but also on technical skills (coding skills) and the ability to collaborate in the process of writing code. The basic requirement for our meetings is to provide an easy to navigate and comfortable interface design that allows the user to operate freely and to switch between different sections at any time. For the main body of the video conference, the goal was to allow users to switch on and off microphones, cameras, share screens and to allow users to adjust their viewpoint. We wanted users to be able to record video content at any time so that they could better review the meeting, which is particularly important for interviewers who may want to review how different people performed if they are interviewing multiple people at the same time. In the interview room, the whiteboard is used to help both parties sort out and communicate in a more abstract way, and the chat room is used for text communication etc. The code co-editing module supports as many languages as possible to suit the interview environment, and can score and compile the code in time to make the meeting more efficient and to give the interviewer a better and faster understanding of the technical level of the participants.

## 1.3 Customer Statement

### 1.3.1 Preface

This report is designed for the potential users of this display video conferencing platform, i.e. the two categories mentioned above. This report will help you to better understand the progress, internal details, architecture and many other aspects of the project. This will help you to get started and gain a better understanding of the project in the shortest possible time.

### 1.3.2 System specification

In today's increasingly informative world, online services and online offices are starting to be taken seriously by more and more countries and organisations. A number of companies are starting to use online interview platforms to interview and select new employees. Interviewers and interviewees can communicate in real time through the interview platform even if they are in different locations. This not only makes the interview process easier and quicker, it increases the efficiency of the interview and avoids the spread of the epidemic. However, the online interview platforms on the market today are largely multi-functional and not targeted. It cannot better meet the requirements of interviewers interviewing programmers and other technical staff. Therefore, in this project, the team wanted to develop an online interview platform that also has a CV module, video conferencing function, online chat function, online code editing function and online whiteboard writing function.

Firstly, the interview platform allows interviewers and interviewees to register and log in. The system will verify the identity of the person and only users who have completed the login can use the main function of the project (interview function).

The main function of this interview platform is the video conferencing function. The interviewer and the interviewee will enter the same video conference after entering their conference number and password. After obtaining user rights, the user can choose whether or not to turn on the camera and microphone during the meeting. The system also supports users to share their own screen. This facilitates better information exchange between interviewers and interviewees during the interview process.

In addition to video and voice communication, the system also offers text communication to the interviewees. In the interview interface, interviewers and interviewees can use the chat box to type and communicate with each other. Real-time text communication ensures that the interview process runs smoothly.

To facilitate a more systematic assessment of the interviewer's professional competence, the system will provide an online code editor. When opening the editor, the user can select the language of the code to be written. The code compiler supports the interviewer and the interviewee in writing code together and will distinguish between the code written by both. The user can run the code at any time during the coding process.

In addition, interviewers can create their own CV pages. The system supports the interviewer to use the CV template provided by the system to automatically generate a CV after filling in the information. The interviewer can set the permissions for their CV, which can be marked as only visible to them, only visible to them and the interviewer, or public. The interviewer can send their CV to the interviewer in advance for a more targeted interview.

The interview platform also offers a calendar function. Users can record their schedule in the schedule screen. The schedule module allows users to check the availability of interviewers and interviewees so that they can schedule their interviews appropriately. At the same time, the platform supports the appointment function, which allows interviewers and interviewees to book each other's availability for interviews. After a successful appointment is made, the schedule will be added to the interface and the interview time, meeting number and participant password will be displayed.

## 2 Glossary of Terms

**Interviewee:** A person who evaluates the performance of the interviewer and decides whether to enter into an employment relationship with the interviewer.

**Interviewer:** Attend the interview held by the company, show their programming ability, attract the attention of the company, and strive for employees.

**Whiteboard handwriting:** A white page, you can use the tools provided by the system or handwriting to communicate with others.

**Function point:** a standard measurement unit for measuring software scale.

**Workload estimation method based on WBS:** Break down the project and estimate the development days by referring to similar projects online.

**Constructive cost model:** It is an accurate, easy-to-use, model-based approach to cost estimation

**Video conferencing:** The face-to-face communication between two or more parties using communication devices and the Internet

**Internet-based video conferencing:** refers to the use of new video acquisition and transmission equipment, refers to the real-time transmission of moving images, voice and application

data information between two and more points.

**MVT:** An architectural pattern used by the django framework in python. M is for models, v is for views, T is for templates.

**User Interface:** The user interface (UI) is the point of human-computer interaction and communication in a device.

**Servers:** Server is a device that provides functionality for other computer programs or Devices.

**Registration:** Customers register using their credentials, which are available in the database. When pre-registering before entering the system, customers are asked to enter their name, gender, etc. By registering, it allows the client enters the system for an interview.

**Interviewer account:** the account of the interviewer have higher level permissions. You can record meetings, create meetings, use chat boxes that are visible only to interviewers, and see real-time manual ratings.

**Database:** Hosted on the website, used to store customer data and parking information.

**Containerization:** Containerization involves bundling an application and data together with all of its related configuration files, libraries and dependencies required for it to run in an efficient way across different computing environments.

**System architect:** the overall designer of the software project, who is the developer and integrator of new products and new technology system of the software organization.

**System analyst:** is the core leader of a large software project. His main responsibilities are to make the overall planning of the software project, analyze the requirements, design the core architecture of the software, guide and lead the project development team to develop and implement the software, and carry out the overall management of the entire project.

## 3 System Requirements

### 3.1 Requirements Models

#### 3.1.1 Application process of requirements engineering

Eliciting requirements, specifying requirements, confirming requirements

We start by identifying some specific requirements through a meeting of the whole team, followed by a study to confirm that the requirements are reasonable and feasible. Questionnaires, on-site interviews, centralised surveys, etc. are used to get more information about the system requirements. This is followed by three or more operations. The questionnaire process does not stop there, we are constantly compiling new data and looking for more available information. We do not finalise the requirements until we have taken into account the customer's experience as far as possible and can actually design the system.

#### 3.1.2 Requirements analysis methods

We started with a group meeting where each group member brainstormed according to the teacher's request. Thinking separately about what requirements could be accomplished to achieve the best use from both the interviewer and interviewer perspectives. All group members' opinions were brought together to create a preliminary summary of the requirements first.

We then used the questionnaire star technique to set 10 multiple-choice and 10 subjective questions and posted them in several large groups on WeChat, inviting students or network engineers to answer the above questions. The data obtained through the survey was analysed according to the big data and the needs that most people considered unnecessary were removed. New needs proposed by some of the participants in the survey were also considered and added.

After that we intercepted some students (about 10 people) who were catching an early class at around 8am in front of the entrance of the 4th Education Department and interviewed them for 5 minutes, asking them what features they would like to have if they were faced with such an interview system, what features they would not like to have, etc.

Finally we invited another group from the same class (5 people in total) to be questioned and surveyed in an empty classroom, where we presented some requirements and talked to them, and used this Focus group to optimise the requirements analysis one last time.

### 3.1.3 Application of Sprint planning Meeting

We have a fortnightly phase. On the Monday of each fortnight, we have a Sprint planning meeting at the Tencent conference or in an empty classroom in the 4th floor, where we decide which features to complete for the sprint based on the product to-do list and a prior estimate of the size and workload of the software. We will then add or subtract work from the sprint plan, taking into account the workload of the other courses in the sprint. Finally, we decide what we plan to accomplish in this sprint. Then we move on to the second stage, where we define the division of labour for each person according to the personnel planning arrangements. We then get more specific. We then get more specific about what each person will be doing at what time. Which function is to be carried out by which person. We also discuss how the work will be done and the group can ask questions and other members of the group will answer their questions or provide useful web pages and books to answer questions. Finally, we create a sprint backlog, in which we record in detail the work to be done in the sprint and the tasks to be completed by each person. This is to facilitate the division of contributions and to make it easier to produce reports and individual documents later.

- example Sprint1: Substitute matters are all about USE cases.

As the semester has just started, the workload for each subject is relatively light. So in this sprint we decided to increase the workload moderately. In sprint 1 we decided to finish building the meeting platform (including turning on the video camera and chatting in the chat box) and to add python and other programming languages to the platform, taking into account the architecture of programming platforms such as Java and python. This was the goal of our sprint: to build the meeting platform and introduce the programming languages. Fu Zhaoyu was responsible for analysing the requirements of the system and producing a detailed list of functions to be completed. He also worked with Zhang Jiahe to complete a preliminary system design framework. Zhang Jiahe was responsible for building a preliminary system and completing the implementation of the conference functions. Zhang Yun was responsible for creating the video function and the chat box as well as the final test product. Huang Feihe was responsible for importing the code into the platform and allowing others in the meeting to edit the code together. Terence Ding was responsible for assisting Zhang Jiahe with the platform build and the final testing part.

First, Fu Zhaoyu played with the requirements analysis and presented it to the team, then Zhang Jiahe and Ding Teran built the platform. Afterwards, Huang Feihe and Zhang Yun added features to the platform. Finally Ding and Zhang tested the platform and the bugs that came out of the test were fed back to the development team so that the development team could make changes to the code.

Afterwards Zhang Yun raised the technical issue of the video function requiring the appropriate permissions and how to import it onto the system. Zhang Jiahe recommended the code on gitlab to help Zhang Yun understand and build the functionality. Terence Ding made a request for the system to be completed as soon as possible as they needed enough

time to test and add features to the system. We revised our plan to complete the system and allow Zhang Yun and Terence Ding to start adding functionality by Friday evening.

Finally, we created a sprint backlog. This shows the tasks to be completed by the sprint.

### 3.1.4 Application of INVEST

We calculated INVEST in six ways to arrive at it.

We control for the independence of the story. We controlled for the size of the story that was maintained. We split the story into several epic stories such as systems. We split the system story into several stories. This ensured that each story was of a more appropriate size and that these were split from the same story thus ensuring that each story was interdependent. As I mentioned above, we conducted a number of workshops, first carefully analysing the requirements from the developer's point of view. This was followed by a questionnaire and a field survey on what the requirements were from the customer's point of view. We also summarised these requirements, taking into account the opinions of the users and accommodating them in case of conflicting requirements. We also negotiate with the difficulties of producing the function, for example by postponing the ddl or arranging for more than one person to deal with the difficulties. The specifics of our system are negotiated between the user and the development team. When making project size projections, the whole team writes down their own estimates of the size of the use case and the time it will take. The results are then averaged to get an average so that we can successfully estimate the estimated size of the project.

For the measurement part, we used the gitlab system to test the code using gitlab. The team rotates through the test and the changes are tested several times by several people. The code is then tested several times by several people after the changes have been made, and is not considered to be complete until there are no problems. This ensures that the tests are successful and that the system is bug-free.

### 3.1.5 User story cards examples

- Story ID1: As an interviewer, I wish I could create and book meetings so that I could interview others. The task is complete when I can create a meeting and invite regular users into it.
- Story ID2: As an interviewer, I want to be able to access meetings so that I can interview candidates. The task is complete when I can access the meeting and show my interviewer status to everyone.
- Story ID3: As an interviewer, I wish I could turn on and off the interviewer's audio so that I could interview them better to find out what they can do as individuals. The task is completed when I can turn off the video and audio of the interviewer on my own page.
- Story ID4: As an interviewer, I want to be able to open a chat box that is only visible to the interviewer so that I can communicate with the other interviewers to determine the final score. Task complete when I can create a chat box that is not visible to the interviewer
- Story ID5: As an interviewer, I would like to be able to rate the interviewers and display the ranking so that I can make a clearer decision on which interviewer to accept. Task complete when I can rate the interviewers and display the ranking.
- Story ID6: As an interviewer, I would like to be able to start or end the recording of a session at any time. This makes it easier for the interviewer to play back and discuss which person to accept. Task complete when I can start and end the recording at any time

- Story ID7: As an interviewer, I would like everyone to exit the meeting as soon as it ends. This ensures that the interview ends properly. When I end the meeting, everyone exits the meeting immediately. Task completed
- Story ID8: As a user, I want to be able to register and log in to the site. This will ensure that I can access the site. Once I have registered my account, I can access the site and enjoy the services it offers. Mission accomplished
- Story ID9: As a user, I would like to be able to edit my CV information. This way I can make my CV more attractive and more likely to attract the attention of the interviewer. When I edit my CV, the revised CV is displayed to other users. , task completed
- Story ID10. As a user, I would like to be able to view all my appointments. This will allow me to attend interviews in a timely manner and prevent myself from missing them. When I click on the appointments screen, my appointments will be displayed to me. Mission accomplished
- StoryID11: As a user, I would like a reminder to be sent to me when the meeting starts. This will ensure that I don't forget to attend the meeting. A notification will be sent five minutes before the meeting starts. Task complete
- StoryID12: As a user, I would like to be able to enter the meeting and have my name displayed properly. This will allow me to attend the interview properly and allow the interviewer to evaluate my performance correctly. When I enter the meeting, my name is displayed correctly and the task is complete.
- StoryID13: As a user, I would like to be able to communicate with the interviewer by video voice and be able to adjust the background of my camera and adjust the volume of my microphone. This would allow me to conduct the interview better. I can talk to someone in a meeting and I can adjust the microphone and camera, task complete.
- StudyID14: As a user, I would like to communicate with others in the chat box. This will allow me to exchange ideas with others during their interviews without interrupting them. I can send messages and emoticons in the chat box during the meeting. Task completed.
- StudyID15: As a user, I would like to be able to use a whiteboard to represent the structure of my code. This would allow me to better represent my code structure with pen and paper. I can open the whiteboard in a meeting and also use the utility to draw a mind map.
- StudyID16: As a user, I would like to be able to use the coding tool and have my code scored so that I can write code to better demonstrate my programming skills in an interview. I can choose the type of code to write in the session and can run the code to see comments, markups etc. After allowing the code you can see the code rating.
- StudyID17: As a user, I would like to be able to see the ratings and rankings at the end. This would give me an idea of my programming weaknesses and an expectation of whether I would be accepted. Eventually I can check the rating and ranking and the task is complete.
- Study ID18: As a user, I would like to be able to withdraw from the session. This will save time during the interview and after the session. Eventually I can exit the meeting and inform everyone that the task is complete.

### 3.2 Functional Requirements

Req	Priority	Description
REQ-1	3	The system allows users to create accounts
REQ-2	4	The system allows users to log in
REQ-3	6	The system allows everyone to edit their own personal information
REQ-4	4	The system provides CV templates for all users to complete their CVs
REQ-5	7	The system generates user CVs based on user template information for others to view
REQ-6	3	The system allows interviewers to book multiple interview sessions at the same time
REQ-7	5	The system allows the interviewer to make adjustments to the interview time
REQ-8	6	The system sends notifications to interviewees when interviewers adjust their interview times
REQ-9	4	The system provides interviewers with a visualisation of the interview schedule in terms of days and months
REQ-10	3	The system will send a notification to the interviewers and interviewees one hour before the interview starts
REQ-11	6	The system keeps a record of each person's attendance, recording the date and length of attendance
REQ-12	7	The system allows the interviewer to play back the entire interview process
REQ-13	7	System allows interviewers to create sessions
REQ-14	4	The system provides an entry page to the conference where the user enters the name and conference number
REQ-15	5	The system ensures that meeting numbers do not overlap at the same time and that previously used meeting numbers are reused
REQ-16	2	The system allows all users to join the conference
REQ-17	6	A pop-up alert will be sent to the user telling them that it is time to enter the meeting
REQ-18	5	Systematic differentiation of interviewer and interviewee positions
REQ-19	7	The system allows all users to turn voice on and off during a conference
REQ-20	5	The system allows all users to turn video on and off during a conference
REQ-21	6	The system supports background substitution for video effects for all users
REQ-22	5	The system allows the meeting creator to end the meeting early
REQ-23	3	The system allows general attendees to withdraw from meetings at any time
REQ-24	8	The system allows meeting creators to record meetings
REQ-25	8	The system allows meeting creators to play back recorded meetings
REQ-26	3	The system allows interviewers to leave messages for general attendees to evaluate their performance
REQ-27	6	The system allows interviewers to evaluate general attendees (visible to other interviewers only)
REQ-28	5	The system automatically summarises the programming languages or other areas in which the programmer excels based on the evaluation results
REQ-29	3	The system will have a chat box where participants can communicate by text

REQ-30	7	The system allows users to use preset emoticons in the chat box
REQ-31	8	A code editing board can be displayed in a meeting opened by the system and is visible to all participants
REQ-32	9	All users can write code using the coding area shown in the session
REQ-33	8	The system allows the user to choose at least two programming languages (java, python, c++, etc...)
REQ-34	6	The system allows users to run their own code in real time
REQ-35	6	The result of the program or the error message can be seen by all participants
REQ-36	8	The system allows attendees to change the code with the permission of the coder
REQ-37	7	The system provides testing of user-entered codes in the coding area and feedback on the evaluation results
REQ-38	5	The system allows the interviewer and interviewee to edit a whiteboard at the same time
REQ-39	7	The system allows the user to draw by hand on the whiteboard
REQ-40	8	The system allows users to build mind maps from materials on the whiteboard

### 3.3 Non-functional Requirements

Req	Priority	Description
REQ-1	3	Internet access is required to use this platform
REQ-2	4	Certain equipment requirements (graphics card; computer screen size .....)
REQ-3	6	The system will be updated regularly without affecting the stored information
REQ-4	4	The platform will pay for any accidental disclosure of personal information of the user
REQ-5	7	Once account registration is complete, customers will need to enter a verification code (sent via SMS)
REQ-6	3	Rapid update of system information (1 minute update)
REQ-7	5	Services are not available to unregistered users
REQ-8	6	The system will be tested to check for potential vulnerabilities
REQ-9	4	The system will be modelled in a way that is realizable for any system
REQ-10	3	The system allows users to upload personal information, CVs, etc. for interviewers to view
REQ-11	6	The system pages are professionally designed and beautifully laid out

### 3.4 Functional Requirement Specification

#### 3.4.1 Stakeholders

Stakeholders are, in the design of this system, those groups of people who may be interested in our software. They are often defined as Level 1 and Level 2 Stakeholders, in our case Level 1 Stakeholders are the interviewers and interview participants in the software engineering field, who are collectively referred to as users and who will use our platform directly to assist them with online meetings, planning meeting schedules, conducting online interviews, collaborative programming, collaborative using whiteboards and discussions. The second level of stakeholder is the interviewer who is more in charge of the interview and enjoys more capabilities than the

interviewer, in addition to being the creator of the meeting, during the interview, which includes the evaluation of the interviewer, scoring and internal review sessions that are only collaborative within the interviewer.

### 3.4.2 Functional Requirements Specification

Actors	Goals	Use Cases
Conference interface	Provides a full suite of key meeting management services to support all users. This includes booking meetings, joining meetings, performing meeting -related interactions (chat, video, voice, whiteboard collaboration, code collaboration, scoring assessments, meeting notes), and exiting meetings.	UC8;UC11;UC12; UC13;UC20;UC21; UC22;UC23;UC24; UC25
Calendar interface	A full suite of collaborative schedule management plug-ins for all users. Includes schedule visualisation, add and remove items, and schedule-based meeting recommendations.	UC6;UC7;UC8; UC9:UC10
Resume interface	All users are provided with a set of CV templates, which are integrated with the information filled in by the user to generate visual CV information and give the user the right to adjust the viewability of the CV.	UC3;UC4;UC5
System	Integration of scoring systems and manual and storage of historical rankings of current meeting rooms	UC16;UC17;UC18; UC19
System	Automatically test lines of code and generate code scores	UC17
System	Automatically create and generate meeting rooms after a user has booked a meeting to store relevant information and remind attendees to attend in advance	UC8;UC9
System	Update user information	UC4
System	Update meeting information	UC10
Security System	The information of registered users is protected from disclosure and all information on the platform is used in a reasonable and lawful manner.	UC27;UC28;UC29
User	Register and login to the website	UC1;UC2
User	Amend or update their CV information	UC3;UC4;UC5
User	View the schedule with interview appointments, including visualization by day and month	UC6;UC10
User	Receive reminders of upcoming meetings from the system	UC11
User	Receive manually issued meeting start reminders	UC9;UC11

User	Access to the conference and the ability to identify yourself as an attendee	UC11
User	During the interview process you can communicate with the interviewer by video and voice, and adjust your own volume level, and your own microphone and camera on/off status, as well as adjust the background of your own video	UC12;UC13;UC14
User	You can use the chat box to communicate during the interview process and send emoticons	UC20;UC21;UC22
User	You can use the whiteboard during the interview process, and you can choose to write by hand when using the whiteboard, or you can choose to use the preset tools for mind mapping	UC23;UC24
User	The interview process allows you to use coding tools, choose the type of code, code and comment, see highlight markers, run the code and see error reports and system scoring	UC25;UC26
User	Rankings and scores can be viewed after the manual scoring results are released	UC18;UC19
Interviewer	Ability to book and create meetings and remind others to attend	UC7;UC8;UC9;UC10
Interviewer	Have access to meetings and be able to identify yourself as an interviewer	UC11
Interviewer	When using the chat box, you can select a chat box that is only visible to the interviewer and communicate and vote in it.	UC16
Interviewer	You can see real-time manual scoring results and scoring rankings	UC17;UC18;UC19
Interviewer	The memoir system can be started or ended at any time	UC15

### 3.5 Fully dressed use cases & use case diagram

#### 3.5.1 Use Cases

- UC1:Registration - both interviewers and interviewees register on this platform as regular users.
- UC2:Login - both interviewers and interviewees log in as regular users on this platform using their registered accounts.
- UC3:Information editing - all users can fill in the content in the CV template provided by the system.
- UC4:CV view - all users can view their CV on the generated CV page.
- UC5:CV permissions - all users have the option to set their CV visibility permissions to self only, public, self only i.e. interviewer.
- UC6:Schedule Display - View a day and month based schedule display for users based on their current appointment status.

- UC7:Suggest an available appointment - check the schedule to suggest a time when both the interviewer and the interviewee are available .
- UC8:Book a video interview - the interviewer selects a suitable time in the schedule screen to book an interview and pulls the appropriate person into the interview appointment.
- UC9:Meeting reminders - any participant can quickly remind others that a meeting is about to start.
- UC10:Flexibility to change meeting times - allows for changes to meeting times if the interviewer or system maintenance etc. is unable to attend the meeting normally .
- UC11:Video interview.
- UC12:Debugging microphones and cameras.
- UC13:Shared screens.
- UC14:Video background replacement - background can be set after or before turning on the camera.
- UC15:Recorded meetings - at any point during the meeting, you can choose to start recording the meeting with a prominent reminder to all participants .
- UC16:Interviewer scoring - interviewers can score interview participants with or without reference to the system evaluation.
- UC17:View system evaluation - you can view the system's consciously summarised score evaluation based on the user's programming performance.
- UC18:Interview participant record storage - allows you to record the performance of historical interview participants up to the end of the room.
- UC19:Interview participant ranking - you can rank the participants in this interview based on performance and evaluation.
- UC20:Open/close the chat box - you can open or close the chat box after entering a meeting.
- UC21:Chat box text exchange - open the chat box and enter text to exchange.
- UC22:Chatbox emoticon exchange - open chatbox and send system preset emoticons.
- UC23:Collaborative Handwriting Whiteboard.
- UC24:Whiteboard mind mapping exchange - open the whiteboard and use the tools given by the system to draw a mind map.
- UC25:Code Co-editing.
- UC26:Assisted coding - can code with the other party as they write their code (synchronised display + differentiation between the two codes).
- UC27:Interview process security - the interview whiteboard, chat box, coding area, chat box content, scoring and ranking are destroyed at the end of the session. The video recording of the meeting is the only record of the interview and is only available to the interviewer.
- UC28:Security of personal information - Users can set their CV permissions to protect the security of their personal information. Only the user has access to the CV information page if it is only visible to him/her.

- UC29: In the event of a system crash, the platform may become unavailable due to various circumstances. In this case, all users of the platform will be automatically logged out in order to protect the safety of the users, and all data will be destroyed directly.

### 3.5.2 Use case Diagram

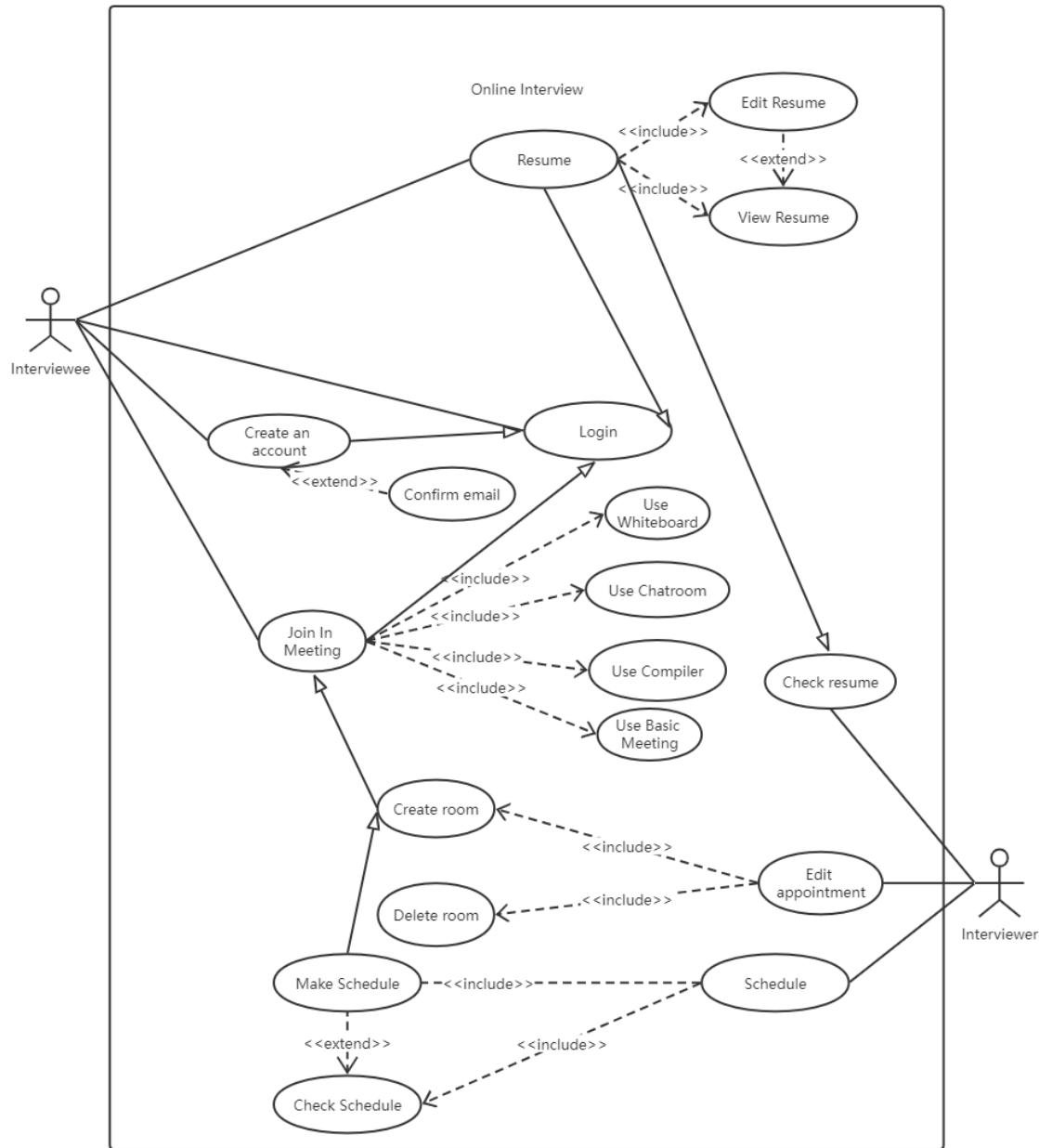


Figure 1: Usecase Diagram

## 3.6 Traceability Matrix

### 3.6.1 Requirement Traceability Matrix

### 3.6.2 Use Case Traceability Matrix

PW	Use Cases	Website	Meeting Interface	Resume Interface	Schedule Interface	Database	Notification Customer	Notification System
2	UC-1	X				X	X	
2	UC-2	X				X	X	
2	UC-3	X		X		X	X	
2	UC-4	X		X		X		
4	UC-5			X			X	X
3	UC-6	X				X X		
7	UC-7					X X		X
7	UC-8	X		X		X X		X
4	UC-9	X		X		X		X
9	UC-10	X		X		X X		X X
6	UC-11	X		X				X X
6	UC-12			X				X
5	UC-13	X		X				X X
7	UC-14			X				X
9	UC-15	X		X		X		X X
2	UC-16	X		X				
9	UC-17	X		X				
4	UC-18			X		X		
3	UC-19	X		X				X
2	UC-20	X		X				X
5	UC-21	X		X				
5	UC-22	X		X				
2	UC-23	X		X				X X
8	UC-24	X		X				X X
2	UC-25	X		X				
10	UC-26			X		X		X X
6	UC-27	X		X				
6	UC-28			X				X X
9	UC-29	X		X		X		

### 3.7 Fully dressed description

#### 3.7.1 Example1

- **Name :**Enter Interview Meeting
- **Id :**UC11
- **Description :**The user has an interview with a particular interviewer, he or she needs to access the meeting by either accepting the link shared by the interviewer or entering the corresponding meeting number. In the meeting room, the details of the participants will be shown
- **Actor :**Interviewer & interviewee
- **Organizational benefits :**The necessary function for actors to correctly access expected meeting room
- **Frequency of use :**Enter interview meeting is the main purpose of the software, the usage of this use case can be very frequent
- **Triggers :**The user selects the option to enter a meeting
- **Precondition :**User enters the meeting page
- **Postcondition :**The user enters the meeting
- **Main course :**
  - The user selects enter room function
  - The system displays a dialog showing the username and a form for entering the expected room number
  - The user enters the number and confirm
  - The system displays the correct room with users information and accesses the user in
- **Alternate course :**
  - The user selects enter room function
  - The system displays a dialog showing the username and a form for entering the expected room number
  - The user enters the number and confirm
  - The system does not find a room with the entered room number, so it creates a new room and access the user to that room
- **Alternate course :**
  - The user selects enter room function
  - The system displays a dialog showing the username and a form for entering the expected room number
  - The user cancels the request
  - The system returns to the main page

### 3.7.2 Example2

- **Name** :Whiteboard handwriting
- **Id** :UC23
- **Description** :During the interview, the interviewer might want the interviewer to draw some simple diagrams to illustrate some points or the interviewer want to use the board to explain an idea, they can open a whiteboard and use their mouse to write on it.
- **Actor** :Interviewer & interviewee
- **Organizational benefits** :The whiteboard function increases the satisfaction of users by providing a tool for them to express ideas
- **Frequency of use** :The majority of interviewees are willing to use the board to illustrate their answers, and interviewees are likely to ask for the interviewer to draw some structural diagrams to check their ability.
- **Triggers** :The users select the whiteboard function
- **Precondition** :The users are already entering a room
- **Postcondition** :The whiteboard is shown and both the interviewer and the interviewee can draw on it
- **Main course** :
  - The user selects the whiteboard function
  - The system displays the whiteboard for users to draw
  - The user uses a mouse to draw on the board
  - The system displays what the user paint
- **Alternate course** :
  - The user selects the whiteboard function
  - The system displays the whiteboard for users to draw
  - The user chooses to exit the whiteboard page
  - The system returns to the main page

### 3.7.3 Example3

- **Name** :Write code
- **Id** :UC25
- **Description** :During the interview, the interviewer might want the interviewer to write some codes to show their technical ability. Therefore, our software provides a browser-based IDE to help candidates easily share their skills
- **Actor** :Interviewer & interviewee
- **Organizational benefits** :Help the interviewee thoroughly test the ability of the interviewer, help the interviewer better present his technical skills

- **Frequency of use** :The majority of interviewees are willing to use the browser-based IDE to show their capacity, and interviewees are likely to ask the interviewer to write some codes to check their ability
- **Triggers** :The users select the write code function
- **Precondition** :The users are already entering a room
- **Postcondition** :The browser-based IDE is shown the interviewer can code on it
- **Main course :**
  - The user selects the code writing function
  - The system displays a dialog for entering which programming language he wants to use
  - The user enters the chosen language
  - The system displays the corresponding browser-based IDE for the user to write code
  - The user writes code on the IDE
  - The system automatically displays highlights and notes
  - The user selects executing the code
  - The system runs the code and gives comments
- **Alternate course :**
  - The user selects the code writing function
  - The system displays a dialog for entering which programming language he wants to use
  - The user enters the chosen language
  - The system displays the corresponding browser-based IDE for the user to write code
  - The user writes code on the IDE
  - The system automatically displays highlights and notes
  - The user selects executing the code
  - The system runs the code and errors occur
  - The system alerts the user errors occur

### 3.8 Scenario

#### 3.8.1 Scenerio 1-Register and fill resume

- **Individual** Lily Yun, senior student, major in computer science
- **Equipment** Mac laptop at school library
- **Scenario**
  - LY open Safari browser
  - LY enter URL to Rainbow Catcher homepage
  - Login & Register page appear
  - LY point Register button
  - Register page show up
  - LY enter personal information

- LY choose “User type” as interviewee and confirm
- Page flash validation result
- Page flash “Register Success”
- LY point Login button
- Back to Login page
- LY fulfill information and confirm
- Page flash validation result
- LY login successfully
- LY enter stakeholders main page
- LY point “Resume” button
- Page of blank resume been displayed in browser
- LY enter personal information and introduction of work experience
- LY point “upload for avatar and certification”
- LY upload corresponding file
- LY set “Browser Permission” of resume as “Only me & interviewer”
- LY point ‘Preview’ button after filled blanks
- Preview of resume displayed on page
- LY pointed confirm
- Resume been saved and show on Resume page.

### **3.8.2 Scenerio 2-Join meeting as Interviewee**

- **Individual** Hooper Cannon, software engineer, wait for interview of Microsoft

- **Equipment** Mac laptop at home

- **Scenario**

- HC open Safari browser
- HC enter URL to Rainbow Catcher homepage
- Login & Register page appear
- HC point “Login” button
- HC enter personal information and point “Login”
- Page flash validation result
- HC login successfully
- HC enter Rainbow Catcher main page
- HC point “join Meeting” button
- Room Entrance page display
- HC enter room Id that been sent by Microsoft advanced
- HC enter room and join the meeting
- Interview page display in browser
- Interview page display personal information of interviewee

- HC settle the microphone and camera setting
- Interviewer open microphone to ask HC to introduce himself
- HC open microphone and have an introduction
- Interviewer start Online Coding function
- Online coding area is displayed in page
- Interviewer ask HC to finish code question
- HC use keyboard and start coding
- HC press saved button
- Coding outcome sent to interviewer's account
- Interviewer announce meeting end
- HC press “leave”button and leave meeting

### **3.8.3 Scenerio 3-Internal evaluation meeting for interviewer**

- **Individual** Three interviewer of Microdoft, CHO, after interview of Hooper Cannon
- **Equipment**Lenovo laptop at Microsoft office
- **Scenario**
  - Three interviewer open Safari browser separately
  - Three interviewer enter URL to Rainbow Catcher homepage
  - Login & Register page appear
  - Three interviewer point “Login” button
  - Three interviewer enter personal information and point “Login”
  - Page flash validation result
  - Three interviewer login successfully with user type “Interviewer”
  - Three interviewer login successfully with user type “Interviewer”
  - Three interviewer “join Meeting”button
  - Room Entrance page display
  - HC enter room Id that been sent by Microsoft advanced
  - HC enter room and join the meeting
  - Interview page display in browser
  - Interview page display personal information of interviewee HC
  - Interviewers open microphone to discuss about how to score HC
  - Interviewers display online coding result of HC to meeting
  - Online coding result of HC is displayed in page
  - Interviewers start online score function
  - Interviewers enter score separately
  - Final score generated on screen and saved
  - Interviewers start online comment function
  - Interviewers enter comment separately

- Everyone's comment generated on screen and saved
- Interviewers start online feedback function
- Interviewers enter feedback towards HC separately
- Everyone's feedback generated on screen and saved
- Interviewers make final decision of HC
- Interviewers announce meeting end
- Interviewers press "leave" button and leave meeting

### **3.8.4 Scenerio 4-Interviewee check interview outcome and feedback**

- **Individual** Hooper Cannon, software engineer, wait for interview of Microsoft
- **Equipment** Mac laptop at home
- **Scenario**
  - HC open Safari browser
  - HC enter URL to Rainbow Catcher homepage
  - Login & Register page appear
  - HC point "Login" button
  - HC enter personal information and point "Login"
  - Page flash validation result
  - HC login successfully
  - HC enter Rainbow Catcher main page
  - HC point 'MeetResult'button
  - Meeting Result page displayed
  - Newly updated report of interview of Microsoft been displayed in page
  - HC point the report
  - Meeting result page display in browser
  - Interviewee Id ana meeting Id displayed
  - Final score, feedback from interviewers of HC displayed
  - HC press "back" button and back to main page

## **4 Effort estimation & risk estimation**

### **4.1 Scale estimation**

#### **4.1.1 Function Scores Estimation**

- UC1:Registration - both interviewers and interviewers register on this platform as regular users.(2)
- UC2:Login - both interviewers and interviewers log in as regular users on this platform using their registered accounts.(2)
- UC3:Information editing - all users can fill in the content in the CV template provided by the system.(2)

- UC4:CV view - all users can view their CV on the generated CV page.(2)
- UC5:CV permissions - all users have the option to set their CV visibility permissions to self only, public, self only i.e. interviewer.(4)
- UC6:Schedule Display - View a day and month based schedule display for users based on their current appointment status.(3)
- UC7:Suggest an available appointment - check the schedule to suggest a time when both the interviewer and the interviewer are available .(7)
- UC8:Book a video interview - the interviewer selects a suitable time in the schedule screen to book an interview and pulls the appropriate person into the interview appointment.(7)
- UC9:Meeting reminders - any participant can quickly remind others that a meeting is about to start.(4)
- UC10:Flexibility to change meeting times - allows for changes to meeting times if the interviewer or system maintenance etc. is unable to attend the meeting normally .(9)
- UC11:Video interview.(6)
- UC12:Debugging microphones and cameras.(6)
- UC13:Shared screens.(5)
- UC14:Video background replacement - background can be set after or before turning on the camera.(7)
- UC15:Recorded meetings - at any point during the meeting, you can choose to start recording the meeting with a prominent reminder to all participants .(9)
- UC16:Interviewer scoring - interviewers can score interview participants with or without reference to the system evaluation.(2)
- UC17:View system evaluation - you can view the system's consciously summarised score evaluation based on the user's programming performance.(9)
- UC18:Interview participant record storage - allows you to record the performance of historical interview participants up to the end of the room.(4)
- UC19:Interview participant ranking - you can rank the participants in this interview based on performance and evaluation.(3)
- UC20:Open/close the chat box - you can open or close the chat box after entering a meeting.(2)
- UC21:Chat box text exchange - open the chat box and enter text to exchange.(5)
- UC22:Chatbox emoticon exchange - open chatbox and send system preset emoticons.(5)
- UC23:Collaborative Handwriting Whiteboard.(2)
- UC24:Whiteboard mind mapping exchange - open the whiteboard and use the tools given by the system to draw a mind map.(8)
- UC25:Code Co-editing.(2)

- UC26: Assisted coding - can code with the other party as they write their code (synchronised display + differentiation between the two codes).(10)
- UC27: Interview process security - the interview whiteboard, chat box, coding area, chat box content, scoring and ranking are destroyed at the end of the session. The video recording of the meeting is the only record of the interview and is only available to the interviewer.(6)
- UC28: Security of personal information - Users can set their CV permissions to protect the security of their personal information. Only the user has access to the CV information page if it is only visible to him/her.(6)
- UC29: In the event of a system crash, the platform may become unavailable due to various circumstances. In this case, all users of the platform will be automatically logged out in order to protect the safety of the users, and all data will be destroyed directly.(9)
- Basic: 4
- Resume: 9
- Schedule: 30
- Meeting: 51
- Chatroom: 12
- Whiteboard: 10
- Code Compilier: 12
- System: 21
- Total: 149

#### **4.1.2 Overview**

We estimated the size of each use case according to its size. One person in the meeting wrote down their own size estimate for each use case on a piece of paper. We then averaged the size estimates and added them to the functions according to the use case to get the function points.

#### **4.1.3 Effort estimation**

WBS-based workload estimation method

- Perform WBS decomposition and break down the tasks of the entire project to the best of your ability.
- Estimate the workload for each type of activity in the WBS using an analogous or expert method with reference to data from similar projects.
- Aggregate all the data to estimate the estimated number of days spent.

#### 4.1.4 System functions

- Provides a full suite of key meeting management services to support all users. This includes booking meetings, joining meetings, performing meeting-related interactions (chat, video, voice, whiteboard collaboration, code collaboration, scoring assessments, meeting notes) and exiting meetings: 10 days.
- A full suite of collaborative schedule management plug-ins for all users. Includes schedule visualisation, add and remove items, schedule based meeting scheduling recommendations: 2 days.
- Providing all users with a set of CV templates, integrating and generating visual CV information using the information filled in by the user, and giving the user the right to adjust the viewability of the CV: 1 day.
- Integration of a scoring system and manual and storage of historical rankings of current meeting rooms 0.5 days.
- Automatic testing of lines of code and generation of code scores 5 days.
- Automatic formation and generation of meeting rooms after a user has booked a meeting to store relevant information and to remind participants in advance to attend and update user information 1 day.
- Protect registered users' information from disclosure and all information on the platform will be used reasonably and legally for 5 days.
- Interviewers: interviewers create appointments for meetings; enter meetings and identify themselves; have permission to turn on and off the interviewer's microphone; use chat boxes to select chat boxes visible only to the interviewer and to communicate and rate votes in them; see real-time manual scoring results and scoring rankings; start or end the recording of a meeting at any time; when exiting or ending a meeting, all participants are immediately withdrawn from the meeting: 3 days
- Users: register and login to the website (3); modify or update their biographical information (3); view a schedule containing interview appointments, including visualisation on a daily and monthly basis (3); receive reminders from the system that a meeting is about to start (2); receive reminders from the start of meetings issued manually (2); have access to meetings and be able to identify themselves as participants (3); during the interview process can communicate with Video and voice communication with the interviewer and the ability to adjust your volume level and the status of your microphone and camera on/off, as well as the background of your video (5); the ability to use the chat box to communicate and send emoticons during the interview process (4); adds up to 7 days.
- Access to a whiteboard during the interview process, with the option to write by hand or to use a preset tool for mind mapping (4 days); access to a coding tool during the interview process, the option to choose the type of code, to code and comment, to see highlight marks, to run the code and see error reports and to be scored by the system (3.5 days).
- The ability to see rankings and scores after the manual scoring results are posted and the user's right to exit the session and notify other participants in the session (1 day).
- Total: 43 days

## 4.2 Progress estimation

### 4.2.1 COCOMO

The cost estimating section draws to some extent on COCOMOII, known as the constructive cost model, which is an accurate, easy to use, model-based cost estimating method first introduced by Boehm in 1981. It is essentially a parametric approach to project estimation, where certain characteristics of the project are used as parameters to build a numerical model to predict project costs.

The code-line analysis approach as a metric estimation method was developed extensively in the 1980s and 1990s, and many more parametric models for estimating workload and schedule were developed in the industry, the best known of which is the COCOMO model, the latest version of which is the COCOMO II model.

COCOMO uses 3 different levels of models to reflect different levels of complexity, they are:

- Basic Model. A static univariate model that calculates the software development effort using a function with the estimated number of lines of source code (LOC) as the independent variable.
- Intermediate Model. This is a function of the LOC as the independent variable to calculate the software development workload, and then adjusts the workload estimate with influences involving product, hardware, people, project and other attributes.
- The Detailed Model includes all the characteristics of the intermediate COCOMO model, but when adjusting the workload estimate using the above influences, the impact on the analysis and design steps of the software engineering process is also taken into account.

### 4.2.2 Calculate

- Work required ( $E$ ) =  $ab(KLOC)^{bb}$  [ in persons x months]
- Development time ( $D$ ) =  $cb(Effort\ Applied)^{db}$  [ in months]
- Number of people required ( $P$ ) =  $E/D$  [in units of persons]

The results of the calculations brought into the project are  
Videoconferencing software.

$E$  = target code line 13.2 KLOC, which is small and semi-independent, thus  $a = 3.0$ ,  $b = 1.12$ ,  $c = 2.5$ ,  $d = 0.35$

$$E = 3.0 * (13.2)^{1.12} = 54 \text{ PM}$$

$$D = 2.5 * (54)^{0.35} = 10 \text{ (months)}$$

$$\text{Number of participants in the project } N = E/D = 54/10 = 5.4 \text{ (persons)}$$

### 4.2.3 Disadvantages

COCOMO has some serious drawbacks, the input to the analysis is prioritised, it cannot handle unexpected environmental changes, the data obtained is often not directly usable and needs to be calibrated, only a summary of the past situation is available and it cannot be calibrated for future situations.

#### 4.2.4 Progress Management

The purpose of software project schedule management is to ensure that software projects are completed on schedule and within the specified time. A software project can usually be divided into several sub-projects and tasks, which have certain relationships with each other. The task of project management is to define all project tasks and the dependencies between them, to develop the project schedule, to plan the amount of work and duration required for each task and to track adjustments during development.

Principles:

- Delineation. That is, the decomposition of products and processes.
- Interdependence. The relationship between activities or tasks since the division must be clear.
- Time allocation. A certain number of work units must be assigned to each task being scheduled, and a start and end time must be set for each task.
- Workload confirmation. A predetermined number of people are involved in each project.
- Identify responsibilities.
- Clear outputs. Each task for which a progress plan is scheduled should have a clear output.
- Define milestones.

### 4.3 Risk estimation

Risk	Affects	Description	Likelihood	Impact	Risk Score
Staff changes	Project	During the course of a project, key personnel with a certain level of experience leave or are unable to participate properly in the progress of the project	0.1	7	0.7
Management changes	Project	Changes in management during the course of the project and the resulting changes in the responsibilities and competencies of different members	0.3	4	1.2
Hardware anomalies	Project	Hardware or equipment that is critical to the development of a program is not functioning properly, making it impossible to run the project smoothly or to verify that certain functions are implemented	0.2	8	1.6

Software exceptions	Project	Software errors or exceptions that may occur during the development of the program that are not related to the code itself, including but not limited to faults that occur during the configuration of the environment	0.6	7	4.2
Demand changes	Project and Product	During the course of the project, the product requirements may change somewhat frequently according to our requirements or in conjunction with problems encountered during project development	0.9	7	6.3
Progress delays	Project	At a certain stage or at a certain point, for various reasons, some of the tasks for that stage were not completed in time for the specified period	0.4	4	1.6
Underestimation, miscalculation of workload	Project	Difficulties encountered in the course of work due to the lack of a well-integrated and appropriate methodology for the proper evaluation of a project before a phase or an entire project is carried out	0.4	5	2.0
Technology stack changes	Project	Changes in external resources and technical support on which the project relies for various reasons	0.4	7	2.8
External platform termination of service support	Project and Product	The Agora SDK, and Agora's accompanying service tools, support the video conferencing theme of the project, and some functional components (audio and video, whiteboard, etc.) may experience service exceptions or discontinue web-specific services	0.1	9	0.9
Projected completion date ahead of schedule	Project and Product	Advancement of the estimated submission deadline for the project before the project is closed	0.1	6	0.6

## 5 Domain model

### 5.1 Domain Model

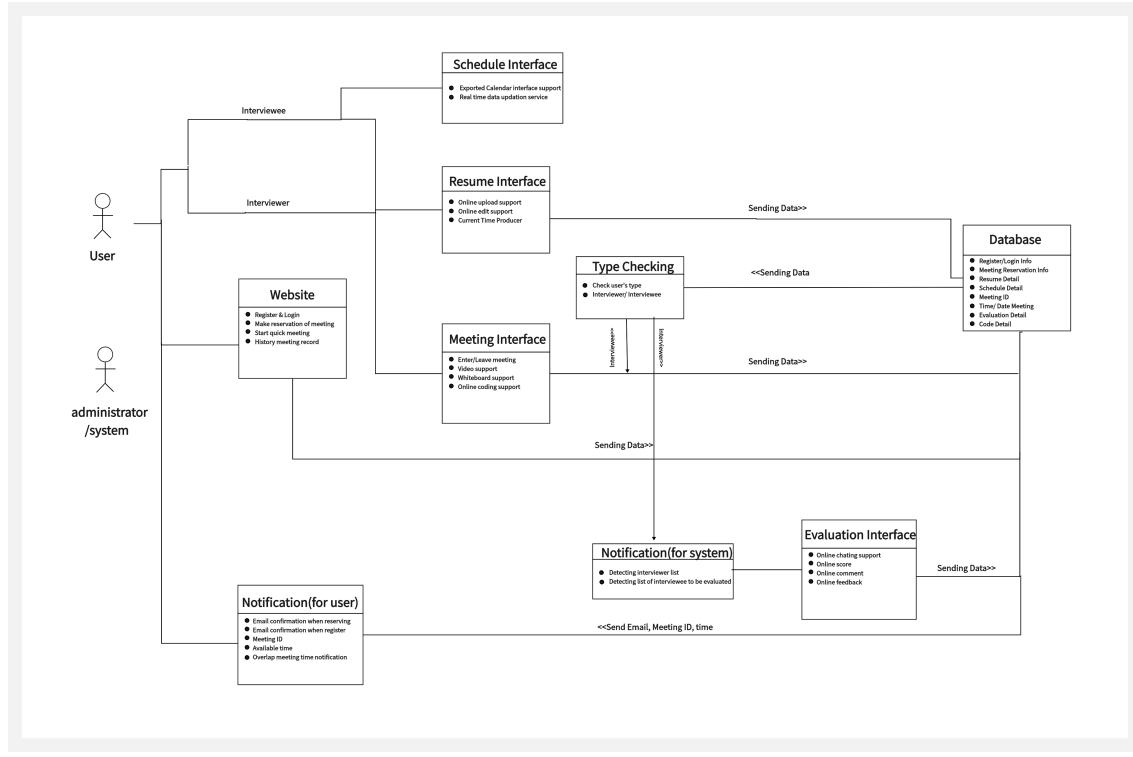


Figure 2: Domain Model

### 5.2 Domain Model Derivation

The domain model is derived by the use cases and functional requirements with the highest priorities. The system allows users to register (UC1) and login (UC2).

- **The Resume Interface:** After login, users can edit their resume, upload file (UC3), and preview their resume (UC4-5) before saved it to browser. User need to set the browse permission to determine who can witness their resume.
- **The Schedule interface:** Schedule will show the upcoming meeting of a user, the detail of meetings are produced by databases. Interviewer can set meetings by viewing their schedule for the available time first. After the time checking, interviewer can make a reservation of the meeting (UC7-UC11). Each time a meeting is been reserved, the meeting ID will automatically be stored in database, and alert the interviewee for the new meeting (UC12).
- **Meeting interface:** Since the meeting ID have been pre-sent to interviewee and interviewer (UC-12), they can enter meeting with the ID or be invited. The meeting UI is differentiate according to type of user, for interviewee, meeting UI will show up microphone, video, online chatroom, whiteboard, online code, but for interviewer, apart from these, will also having online score, commenting and feedback functions, which considered as the final result of a interviewee performance (UC 14-25). The feedback of interview will later be posted by interviewers and interviewee can check up them in their personal page (UC26). During the meeting, every user has the right to leave the meeting in anytime (UC28-29).

- **Online Chat Interface:** Users can communicate in words and emoji messages (UC30-32). The White Board can be used by moving mouse to drawing stuff. The white board also offers a toolbar for user to draw UML diagram (UC 33 - 35).
- **Online Code Interface:** Users are available to select coding language, then after clicking the coding area, user can start coding with others interactive way. After the code is been submitted, the system will score the code and give a score automatically (UC36 -42).
- **Notification System:** When user make a reservation, the system will send a meeting ID and updated time schedule to users, if the newly added meeting are overlap with other meeting, system will alert the user. When user doing register, the system will send email confirmation (UC 11, UC 13).

## 6 System architecture & system design

### 6.1 System architecture

We use the MVT(Model-View-Template) architecture.

- **Model:** view gets the data from the model. Models are database tables. When we create a model with certain data we are creating database tables for that data. Therefore, the model contains the necessary data for a specific request. The view gets the necessary data from its corresponding model. For a request, we will write a model that will have the necessary data, such as interviewers, meeting times, CV information and so on. We have written this in models.py.
- **View:** When a request is sent, the system will search for a view for the request. If there is no view for the request, it will not respond. view is python methods. view gets data from the model or renders a template. view is the connection point between the model and the template.
- **Template:** The view provides a template for a specific request. A template is the front-end component of the system. It contains the static HTML output of the web page as well as dynamic information. When a request is accepted, the system will retrieve the corresponding template file. view will load the template file and dynamically display the data from the model inside the template. We place all HTML templates in a directory called templates.

### 6.1.1 Architectural representations

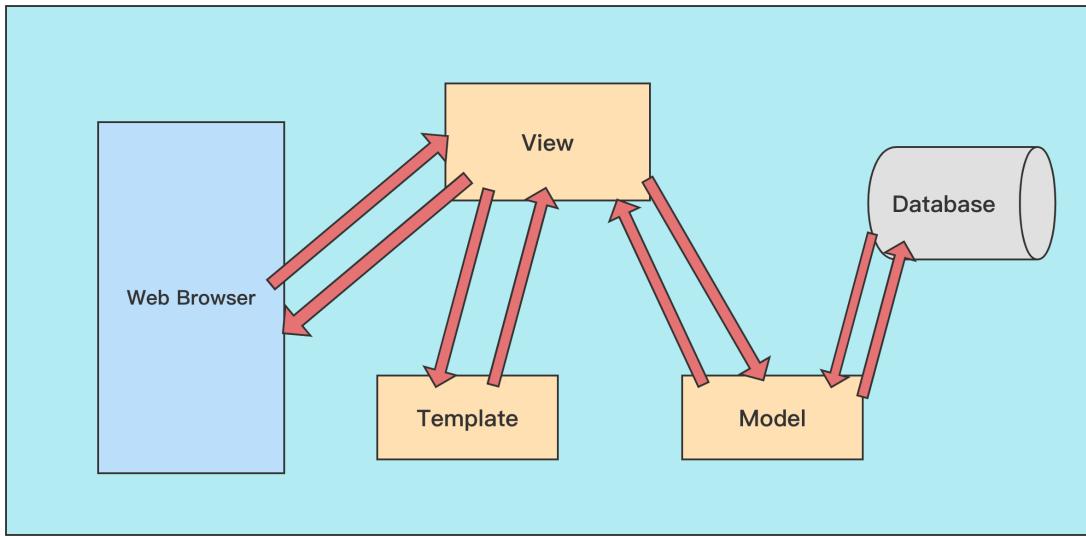


Figure 3: Architectural representation

- **View:** Receive requests, process them and return results.
- **Template:** Encapsulates html, css, JavaScript and so on.
- **Model:** Interact with the database.
- **Web Browser:** Send requests, receive results, render html and css, executes JavaScript.

### 6.1.2 Architectural design decision

The reasons why we choose the MVT (Model-View-Template) are as follows.

- **High cohesion and low coupling for easy development:**

In the actual development process, the original version needs to be iterated according to different requirements. MVT separates the database operations, request processing and page layout, which facilitates the update of the code, without the need to move the whole body, and makes it easy to maintain the code.

- **Improve development efficiency:**

The separation of the front and back ends makes it easy to integrate the code. Convenient for users to divide the work. It also improves reusability and reuse of similar parts, which improves development efficiency.

## 6.2 System design

### 6.2.1 Interaction diagram & sequence diagram

- tool: <https://sequencediagram.org/>

### Video Interview - Basic course of events

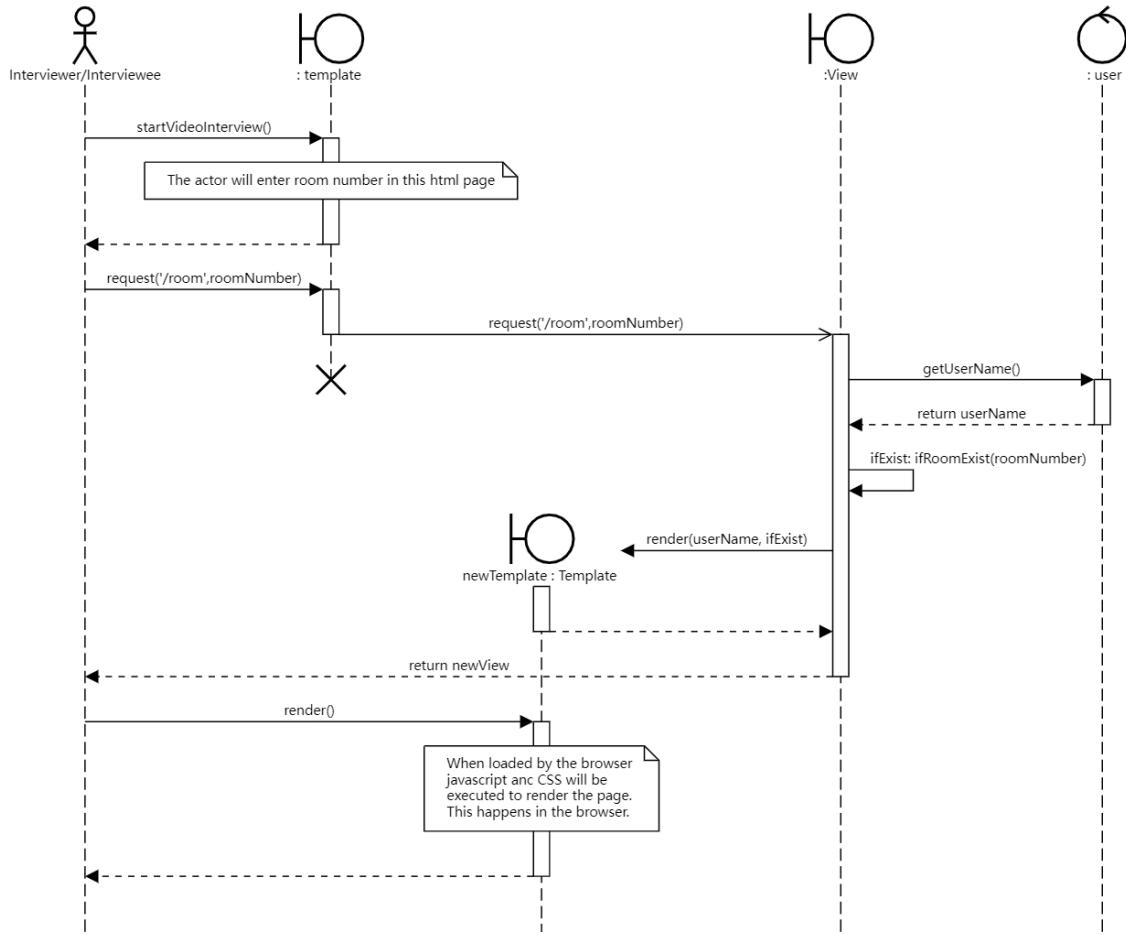


Figure 4: Sequence Diagram

Sequence diagram to show the sequence of the meeting function.

#### 6.2.2 Activity diagrams

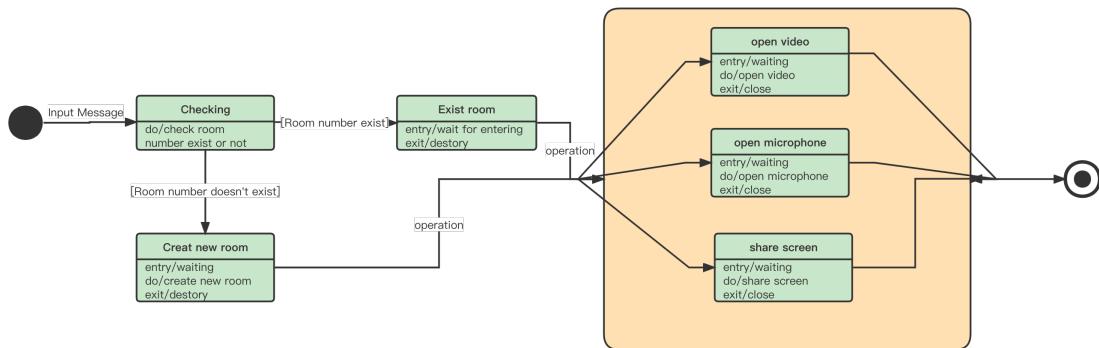


Figure 5: Activity Diagram

Activity diagram to show the activities involved in the meeting process.

### 6.2.3 State machine diagram

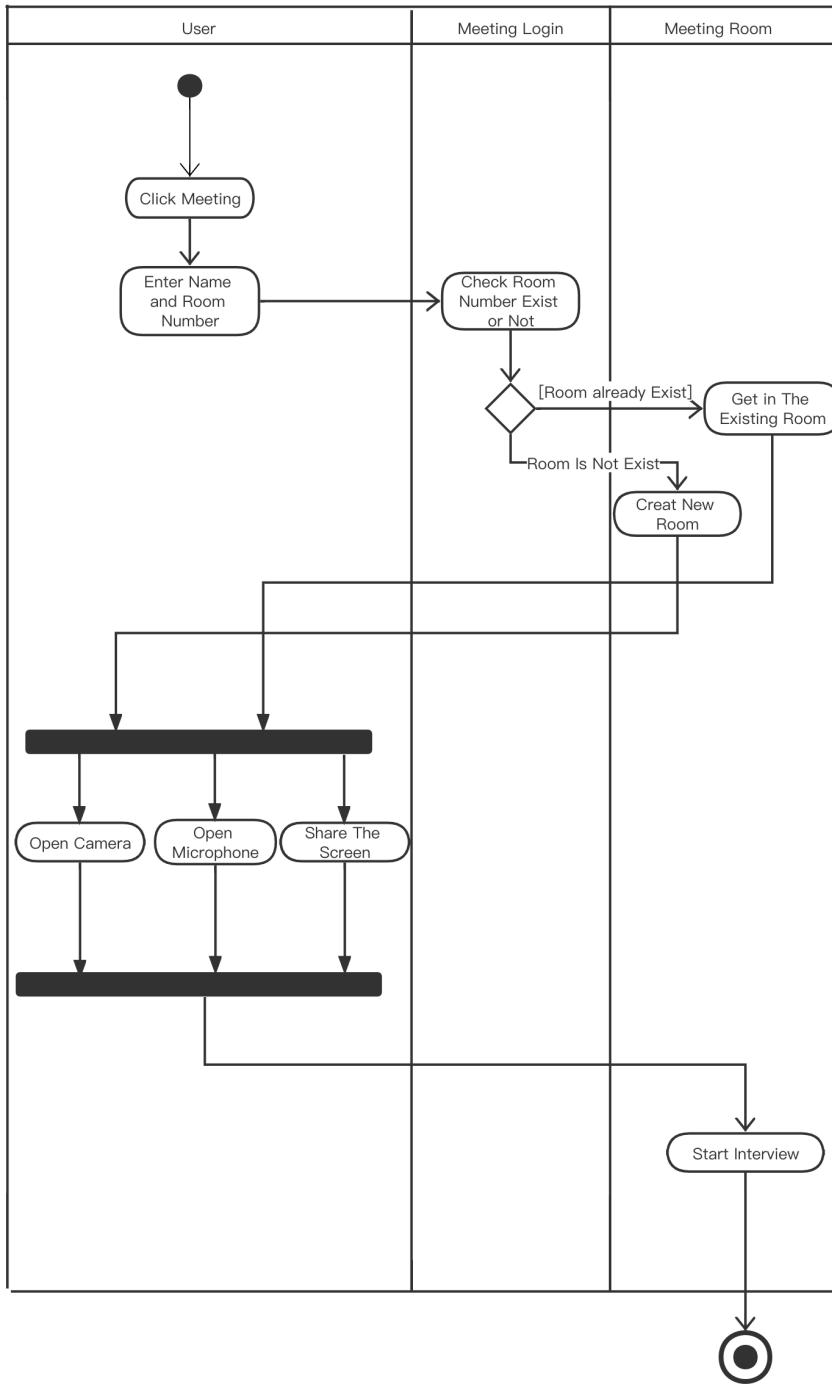


Figure 6: State Machine Diagram

State machine diagram to show how the meeting system reacts to internal and external events.

#### 6.2.4 Database

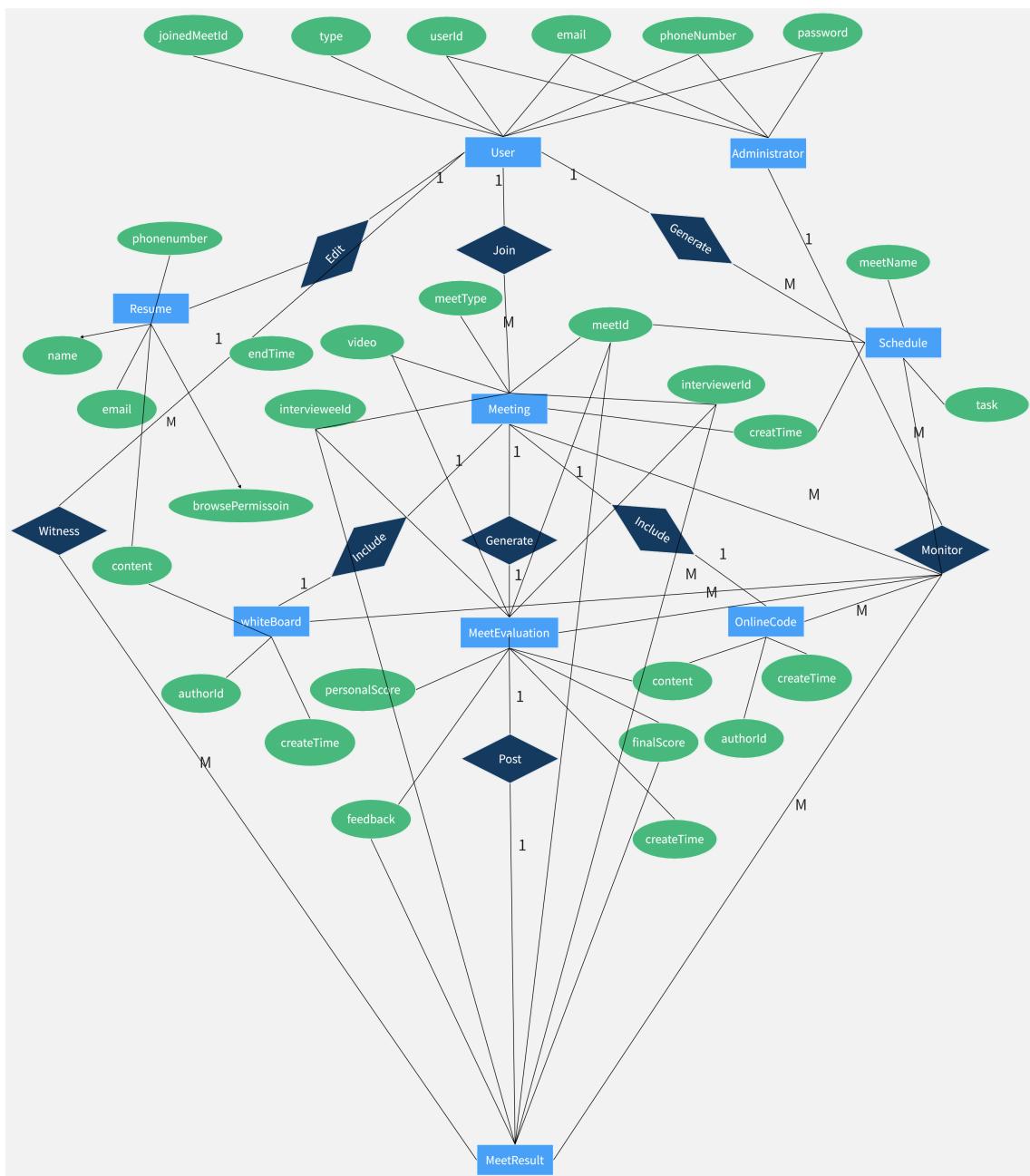


Figure 7: ER Diagram

User	
PK	userId
	email username password phoneNumber type joinedMeetId

whiteBoard	
PK	Id
	authorId (FK) meetId (FK) createTime content

OnlineCode	
PK	Id
	authorId (FK) meetId (FK) createTime content

Administrator	
PK	Id
	email password phoneNumber

Schedule	
PK	Id
	userId (FK) meetId (FK) meetName meetTime task

Resume	
PK	Id
	intervieweeId(FK) name email phonenumber avatar content browsePermission

Meeting	
PK	Id
	intervieweeId(FK) interviewerId(FK) meetType orderType maxNumber video createTime endTime

MeetEvaluation	
PK	Id
	intervieweeId(FK) interviewerId(FK) meetId (FK) personalScore content feedback finalScore createTime orderType video (FK)

MeetResult	
PK	Id
	intervieweeId(FK) interviewerId(FK) meetId (FK) finalScore (FK) feedback (FK)

These tables represent entities from the previous ER diagrams. There are three types of users in the system, interview, interviewer, administrator. The interview and interviewer share the same table User, and differentiate by ‘type’, which 0 = interview, 1 = interviewer. So that in later on process, system check type of user and whether to allow them access internal evaluation meeting only to interviewer or not. Apart from the basic personal information of user such as email, username, password, phoneNumber, the joinedMeetId will remember every meeting id that the user had joined. Besides interview and interviewer, there will also have administrator who can maintain the system, with email, password, phoneNumber stored in the table.

By the joinedMeetId, system can check the detail and corresponding interview, interviewer of every meeting, so that user can track meeting data easily and save them evidence for legal argument or for other security concern. The whiteboard will save the composing authorId, current meetId and the createTime when user save the content. Same logic for OnlineCode, which make it easy to track the author of each line of code. To better aware user the meeting plan, there will be Schedule table to store details of every coming meetings. User can view name and time of meetings and note down the tasks in a specific meeting.

Since we are a system mainly for interview function, there also have a Resume table for interview and interviewer to generate their information in formal form. To be mentioned, each time user generate their resume, they will set the browse permission of resume (0 = Only me, 1 = Public, 2 = Only me + interviewer). For the crucial function meeting, we have three tables in total which are Meeting, MeetEvaluation, MeetResult.

The Meeting table will remember the maximum population in a room, the order type of the meeting (registered or real time), recorded video of meeting if any, and the start time, end time for the meeting. Until now, the tales above are facing to all type of users, but differentiate occurs when meet the function of internal interviewer evaluation towards interviewee and the result of interviewee after each interview.

They stored in MeetEvaluation and MeetResult sperately. Only user whose type is interviewer can join the internal judgement meeting towards interviewee, and they can score the interviewee, write comment, and feedback for interviewee to see. After the evaluation meeting, interviewee can the result of their interview from MeetResult table.

## 7 Design of Tests

### 7.1 Test case

Each unit will be tested before merge into the system. The units that need to be tested are:

- Website
- Database
- Meeting
- Whiteboard
- Chatroom

#### 7.1.1 Website

- Goal: The website section mainly includes the user login and registration, the selection of functional modules, and the display of each functional part of the page, which is a large scope, and it is important for us to focus on whether the page is displayed properly,

whether the css style is correct, whether the page jumping logic is correct, whether the jumping process is coherent, and whether the js interaction logic is normal.

- Example Test Case

- Registration

- \* When you enter the website and click on login, you enter invalid information and are prompted to register your information.
    - \* The user enters the registration information and different messages are reported in real time to indicate whether the user has entered the verification successfully.
    - \* Verify that the user enters the same password twice, etc. to jump to the select function box.
    - \* The user will be prompted to verify the delivery of the email and should check their mailbox to receive the corresponding email.
    - \* Clicking on the link in the email should take you back to the feature selection page in the logged-in state.

#### 7.1.2 Database

- Goal: The database should store users' personal information and related records securely and should be viewable by users on certain pages.

- Example Test Case

- Comparing user information

- \* Detects users entering relevant information when logging in or registering.
    - \* Database retrieval and matching returns the corresponding Jsonify value to the front-end.

#### 7.1.3 Meeting

- Goal: Users can access the conference lobby and create a room by using the function selection box. After entering a user name, they can enter the meeting room and other users can see each other, control the audio and video, and share the screen when they enter the same room using the same link.

- Example Test Case

- Exit the meeting room

- \* When a member leaves, see that it is removed from the domain.
    - \* The user clicks out of the streaming channel and the chat room informs everyone.
    - \* An array with this user ID bound to the streaming pipe appears in the console panel.
    - \* After this information has been transferred to the js file, the relevant function is called to clear the user on the pipeline.
    - \* After closing the page directly or simply exiting the meeting, the user should see that they are no longer in the room when they open the page again.
    - \* Ensure that the user does not see the console's parallel send message in the background of the open page to ensure that the user actually launches inside the meeting room.

#### **7.1.4 Whiteboard**

- Goal: Users can create a room by going to the meeting lobby via the function selection box. Once you have entered your user name you can enter the meeting room where you can click on the button to access the collaborative whiteboard area of the meeting for collaborative drawing.
- Example Test Case
  - Users using the whiteboard together
    - \* A user clicks into the whiteboard area from the meeting room.
    - \* Another user enters the same whiteboard area.
    - \* User A starts drawing using the whiteboard.
    - \* The information is conducted after the pubnub service has authenticated the roomID.
    - \* Calling the test function in User B's back office shows the trajectory and information of the added points typed.

#### **7.1.5 Chatroom**

- Goal: Users can access the conference lobby and create a room through the feature selection box. On the right side of the room there is a chat box that can be refreshed in real time, utilising Agora's webRTM service. Users can post messages and the poster information and posted messages can be seen by everyone.
- Example Test Case
  - Send a message.
    - \* A user clicks into the chat area from the meeting room.
    - \* Another user enters the same chat area.
    - \* User A sends a message.
    - \* User B can see and reply.

## **7.2 Test cover**

- Website: 96
- Database: 96
- Meeting: 99
- Whiteboard: 76
- Chatroom: 96

# **8 Project management**

## **8.1 Personnel management**

### **8.1.1 Principle of the distribution of duties**

For this project we used a 2+2+1 rotating development system, with a node every fortnight and the person in charge adjusted. According to the regulations, the person responsible for the documentation from the beginning to the end of the project is the system architect.

The other four people take on development tasks. One of the four people acts as the system architect, whose role remains the same throughout the project. Two of them are responsible for programming and building the entire system. When these two people have finished writing the program, the other two people on the development team will check the program, fix bugs and add features that they have not implemented. Two people from each node of the four-person development team must be chosen to help write the text.

It must include the system architect, who should communicate and consult with the systems analyst to develop the next steps. The other three rotate as programmers and testers, and one person is selected each week to help write the text. In the process, ensure that the rotation system achieves an even contribution from each person.

Finally, at each stage, four people are responsible for writing the program and three for producing the copy.

### 8.1.2 Individual division of labor

- Jiahe Zhang: Works as a system architect in the team and is responsible for communicating with the project analysts. It fully understands and adds to the requirements analysed and translates them into a startable draft system together with the system analysts. At the same time, it takes on some of the programmer's work. Strong communication skills and programming language skills, task-oriented and interaction-oriented
- Zhaoyu Fu: Worked as a systems analyst in the team and speculated on the various requirements and features to be implemented. They also shared with team members, listened to their additional input, understood their opinions and created a sound development plan. Ensuring that everyone was satisfied with the plan. At the end of the week, listen to the developer's description and produce a good copy. It is interaction-oriented.
- Feihe Huang: Work as a developer and tester in a team. Good work ethic and communication skills. Always able to work within the development team to complete relevant code writing plans in a quality and timely manner, get to know team members and negotiate adjustments to the plan in a timely manner to ensure the development plan runs smoothly. It plays a vital role in the development team. It is interaction oriented
- Tairan Ding: Very strong personal coding skills and a good ability to learn. Always able to complete my tasks as quickly as possible and the quality of the completed code is extremely high. I have a natural interest in code. I am always using new techniques and writing methods to complete tasks in the hope of honing my coding skills. He does beautiful work, the code is easy to understand and he can add some amazing new features to the existing ones. He is the self-directed type.
- Yun Zhang: Adaptable at work, able to understand unfamiliar parts and able to get to work quickly and solve problems efficiently. No matter how difficult the task is, even when faced with unfamiliar territory, she will do her best to complete it. Always able to complete the tasks assigned by the team perfectly. She is a task-oriented person.

Our group includes three members with different personality types in the development team. This ensures that development tasks are executed properly while ensuring maximum efficiency. The team member who is a good communicator is placed as the systems analyst. Because this role requires good communication skills, it needs to be held all the time. This is because the person in the braking programme cannot be replaced for a long period of time, otherwise the programme may break down, which can be disruptive to development. The same goes for the

system architect. If the designed system is faulty, the results of later development may be nil. It is therefore essential that the project is well analysed and designed from the source. The other three people work in shifts, fixing the code as much as possible and checking for all errors in the code.

#### **8.1.3 Collaboration in iteration**

- All team members will participate in the first phase of the discussion. Each person will give their opinion and analyse all the requirements of the system as thoroughly as possible to help the system analyst.
- The Systems Analyst will consult with the Systems Architect to determine the final system design architecture.
- The remaining three members of the development team will work interchangeably at each stage, examining the code, identifying appropriate issues and raising them in a timely manner. We will also check each other's code and correct inefficiencies or errors.
- Every Monday, the copywriting team will hand over the system design to the development team in time for the development team to complete the relevant code and deliver it to the team on Friday. The copywriting team will test the code again at the end of the week. Make sure the code will work properly with multiple tests.
- Application of pair programming
  - Every Thursday morning, team members will meet in an empty classroom in the fourth school building. Present the problems you have encountered and show the team members what you have done. The rest of the team will check the code on his computer and solve the problems encountered by the team members through manual programming. He will also evaluate the code he has written and make some suggestions about it. We have created projects on Microsoft and gitlab. Each person will push the results into the project when they have finished their work. Everyone in the group can make changes.

#### **8.1.4 Project specific plan**

Gantt Chart

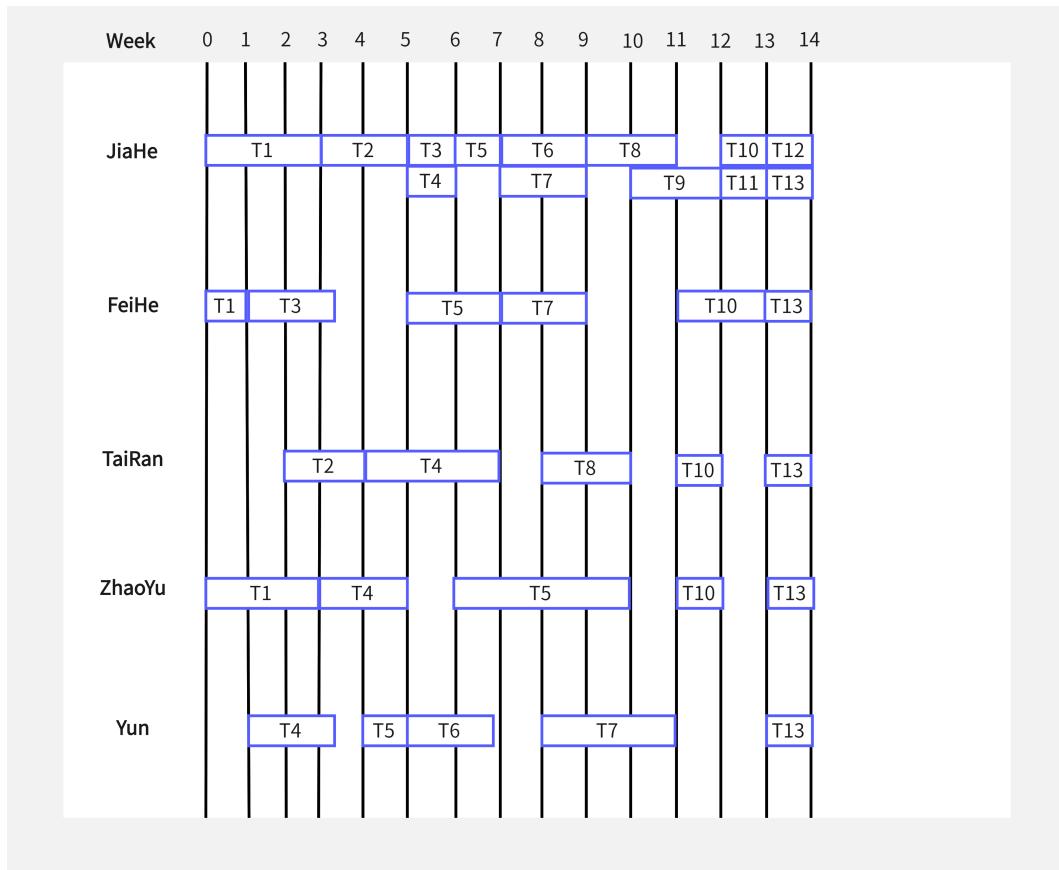


Figure 9: Gunt Chart

### Staff Allocation Chart



Figure 10: Staff Allocation Chart

### **8.1.5 Team Charter**

- **Team Goals**

- Complete the development and testing of the project (video interview sessions) within the specified time frame and as early as possible.
- Ensure that the completed project is of high quality, i.e. that the software has good ease of use and user-friendliness; and that the code is robust and maintainable.
- Each group member will learn practical development techniques and gain an appreciation of software engineering during the development of the project.

- **Team Code**

- Each team member is expected to be proactive in completing their own tasks and not to delay the progress of the project. If there is a need to adjust the schedule because the task is too busy, you may consult with the team.
- Each team member is expected to adhere strictly to the code specifications negotiated by the team during the development process in order to ensure the quality of the code.
- Each team member is expected to attend team meetings and centralised development on time, without being late, and to inform other team members in advance if they are unable to attend for any reason.
- Each team member should actively engage in self and mutual learning, encourage technical exchanges with other team members, and share learned techniques, methods or their own insights with teammates in the group.

- **Team meetings and focused development**

- A discussion meeting for group members will be held every Sunday at 9:00am. The discussion should be as short and efficient as possible. During the discussion, the team summarises the progress of the previous week's project, checks whether the team members have completed their scheduled plans and draws up a development plan for the following week. It is also a good opportunity to discuss any major technical issues that have arisen during the development process, so that the team can use its collective strength to solve problems and move the development process forward.
- Considering that a weekly meeting cannot cope with emergencies, the team leader can always organise a 10-minute stand-up meeting to discuss solutions to problems encountered in development.
- The frequency of centralised development depends on the situation. If centralised development is required, the time and place for group development should be set at the last group or stand-up meeting, normally on a Thursday.

- **Team Building**

- Team members are expected to follow the team code and if frequent non-compliance is observed, fruit or snacks should be purchased for the team, or by the team leader if the team as a whole has followed the team code well.
- The completion of each major iteration can be celebrated with a get-together, for example, to provide motivation and replenish energy for the work to come.
- During the development process, the team has a clear division of labour but remains flexible. If any team member feels that the current division of labour is unreasonable, this can be raised in a meeting and resolved by negotiation.

## 8.2 Software process

### 8.2.1 Process Activity

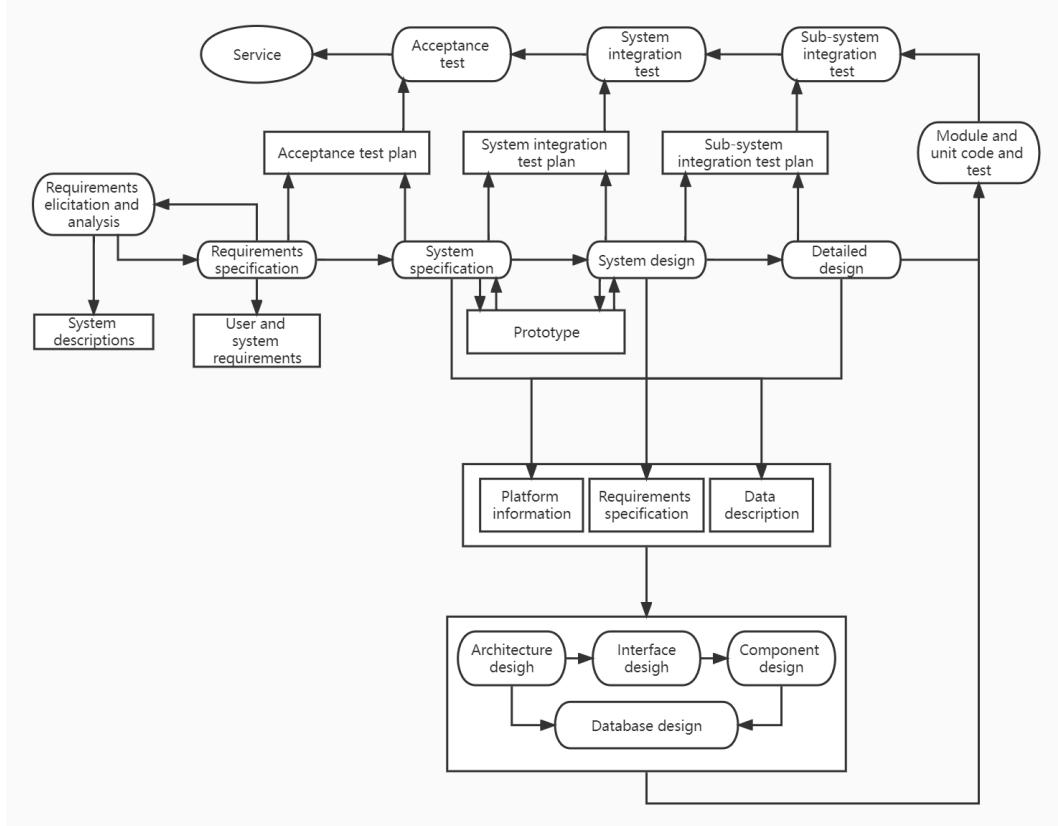


Figure 11: Overview Process Activity

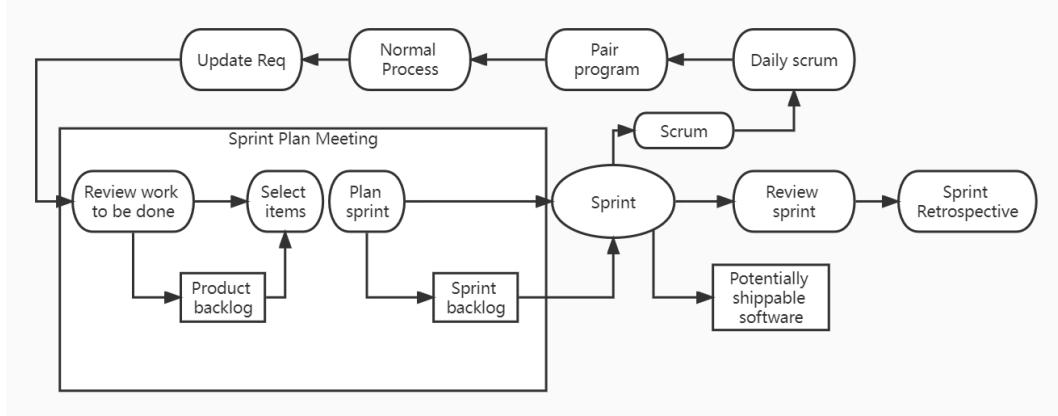


Figure 12: In Sprint Process Activity

The software process model of our group combines the agile method and prototype model, which not only possesses the quick reaction of changes and adjustments from agile development, but also possesses the high quality UI development and efficient code management from prototype model. On the basis of traditional scrum management architecture, we add the process to review and analyze the previous work to acquire more efficiency from the lessons and experience in

the last phase. Additionally, our model includes all four fundamental elements of the whole software develop (specification, design and implementation, validation, evoluation) to assure the completeness and success of our whole developing process.

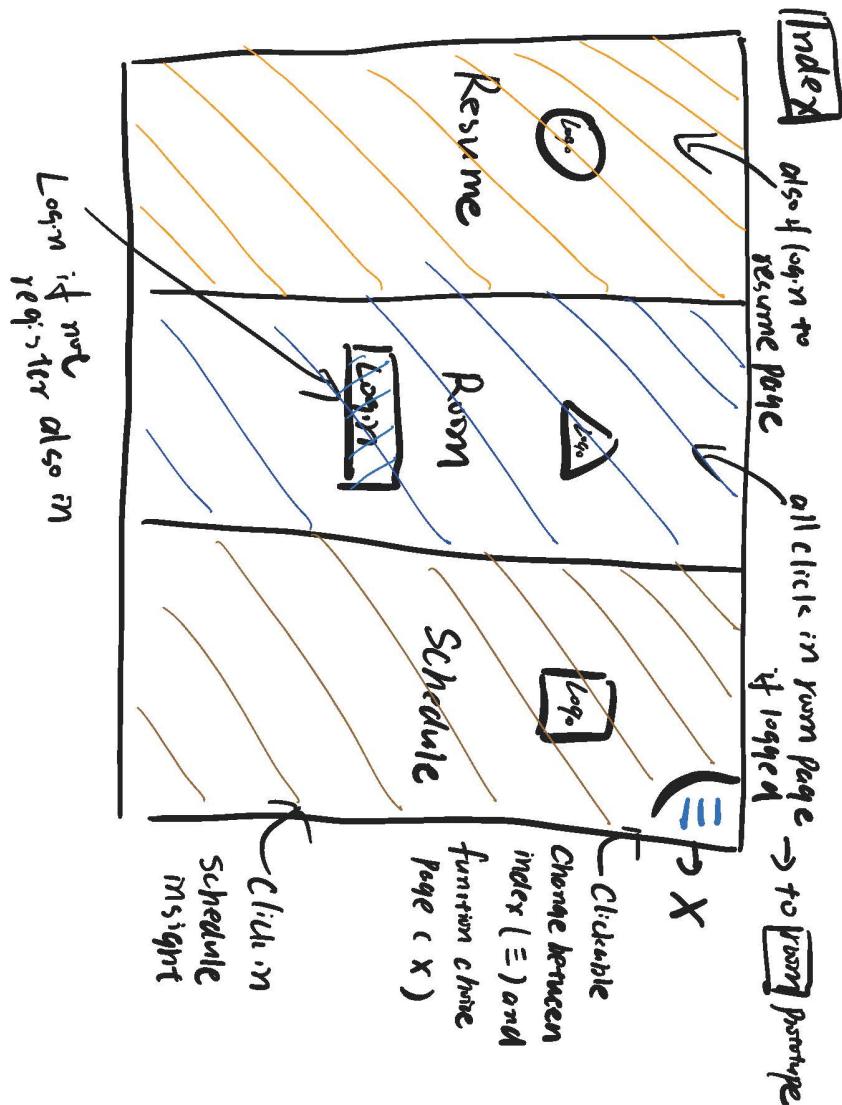


Figure 13: Index Prototype Example

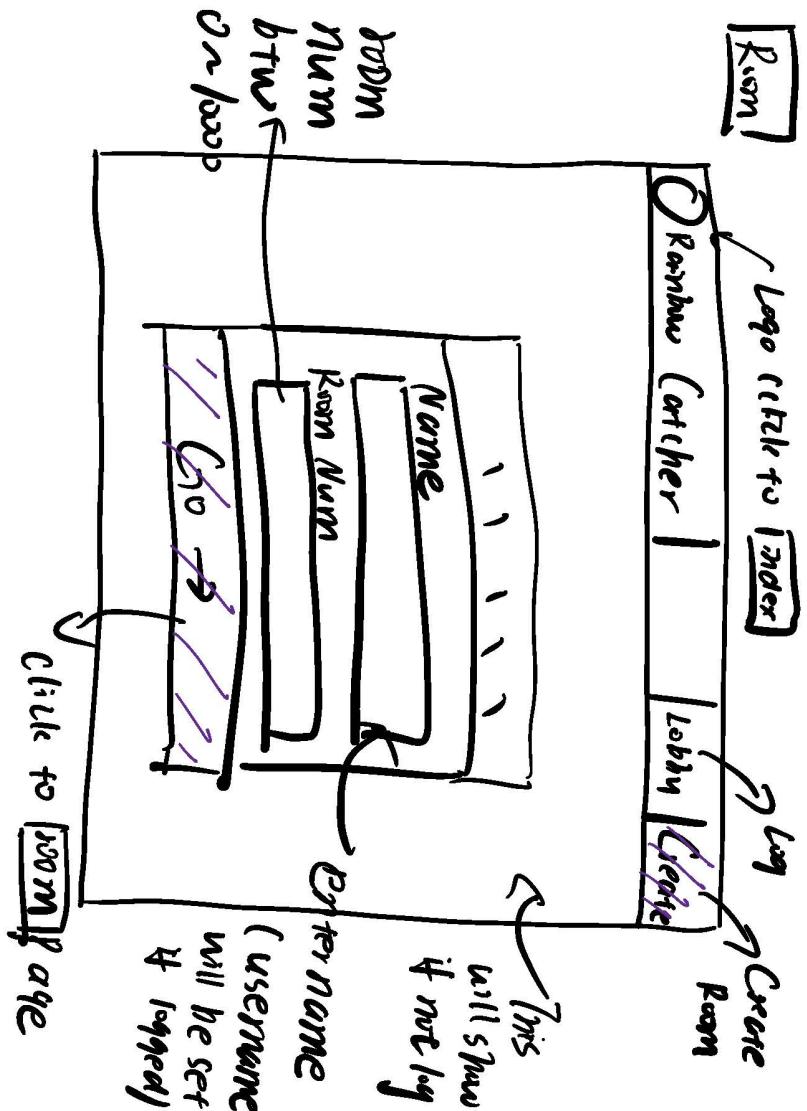


Figure 14: Lobby Prototype Example

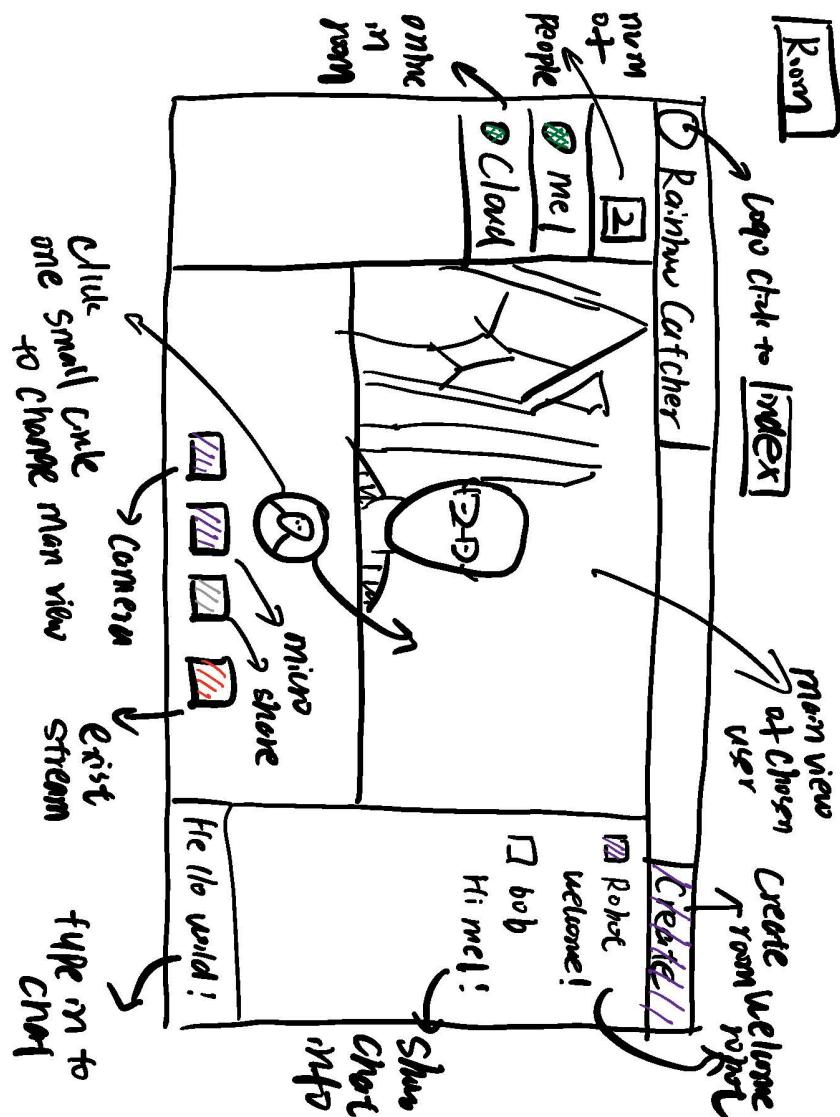


Figure 15: Room Prototype Example

### 8.3 Process improvement

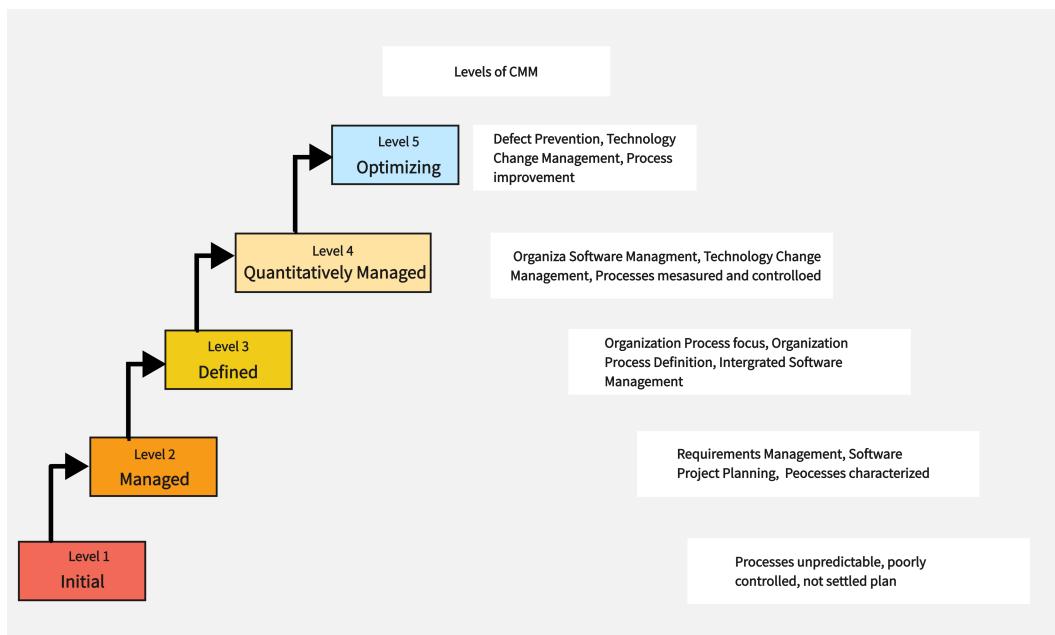


Figure 16: Process Improvement

## 9 Contribution

### 9.1 Final Contribution

- Jiahe Zhang 20372118: 32.5%
- Feihe Huang 20372128: 27.5%
- Yun Zhang 20372105: 20%
- Tairan Ding 20372106: 10%
- Zhaoyu Fu 20372104: 10%

### 9.2 Development Contribution

	Basic	Meeting	Whiteboard	Resume	Chatroom	Final
Feihe Huang	50%	0%	20%	0%	30%	30%
Jiahe Zhang	50%	100%	50%	0%	50%	40%
Tairan Ding	0%	0%	15%	50%	10%	15%
Yun Zhang	0%	0%	15%	50%	10%	15%
Zhaoyu Fu	0%	0%	0%	0%	0%	0%

### 9.3 Report Contribution

Part of Report	1	2	3	4	5	6	7	8	Latex Integration	Final
Feihe Huang	50%	0%	0%	0%	0%	30%	0%	0%	40%	30%
Jiahe Zhang	50%	0%	33%	50%	0%	30%	100%	0%	60%	35%
Tairan Ding	0%	0%	33%	0%	0%	20%	0%	0%	0%	10%
Yun Zhang	0%	0%	33%	0%	100%	20%	0%	50%	0%	15%
Zhaoyu Fu	0%	100%	0%	50%	0%	0%	0%	50%	0%	10%

10 Docker

docker file and description see in Readme

The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal content is as follows:

```
readme.md - epicengine (SSH: 192.168.65.240) - Visual Studio Code

① readme.md x
① readme.md > # 运行
1 # 构建
2 docker build -t epicengine:latest .
3
4 # 运行
5 docker run -itd -p 80:5000 epicengine:latest

(base) root@ub20a:~/epicengine# docker build -t epicengine:latest .
Sending build context to Docker daemon 98.37MB
Step 1/6 : FROM python:3.8
Step 2/6 : WORKDIR /app
--> Using cache
--> 17ce7246ab2f
Step 3/6 : COPY epicengine /app
--> Using cache
--> 0396f547fa00
Step 4/6 : RUN python -m pip install --upgrade pip -i https://pypi.tuna.tsinghua.edu.cn/simple && pip install --no-cache-dir -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
--> Using cache
--> 17c116f9e6f7
Step 5/6 : EXPOSE 5000
--> Using cache
--> 5954df4d0a014
Step 6/6 : ENTRYPOINT ["python","manage.py"]
--> Using cache
--> 14c29ef1f55
Successfully built 14c29ef1f55
Successfully tagged epicengine:latest
(base) root@ub20a:~/epicengine# docker run -itd -p 80:5000 epicengine:latest
5cbdd684066fe4010660d4da98abb42c705fb9891f9f89c925338ec49dc88f
(base) root@ub20a:~/epicengine# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5cbdd6840666 epicengine:latest "python manage.py" 31 seconds ago Up 29 seconds 0.0.0.0:80->5000/tcp, ::80->5000/tcp nostalgic_hopper
(base) root@ub20a:~/epicengine#
```

Figure 17: Docker

```

readme.md ×
① README.md > # 运行
1 # 构建
2 docker build -t epicengine:latest .
3
4 🚀 运行
5 docker run -itd -p 80:5000 epicengine:latest

(base) root@ub20a:~/epicengine# docker build -t epicengine:latest .
Sending build context to Docker daemon 98.37MB
Step 1/6 : FROM python:3.8
--> b2477bd071d6
Step 2/6 : WORKDIR /app
--> 330534a6e0c8
--> 17ec7246ab2f
Step 3/6 : COPY epicengine /app
--> Using cache
--> 206f94039fa0
Step 4/6 : RUN python -m pip install --upgrade pip -i https://pypi.tuna.tsinghua.edu.cn/simple && pip install --no-cache-dir -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
--> Using cache
--> 17c116f9e6f7
Step 5/6 : EXPOSE 5000
--> Using cache
--> 5954df42da914
Step 6/6 : CMD ["python", "manage.py"]
--> Using cache
--> 14c29ef1f55
Successfully built 14c29ef1f55
Successfully tagged epicengine:latest
● (base) root@ub20a:~/epicengine# docker run -itd -p 80:5000 epicengine:latest
5cbdd6840666fd40196664dd88bab042c705fb9891f9f89c925338ec49dc88f
● (base) root@ub20a:~/epicengine# 

```

Figure 18: Docker

## 11 Reference

### 11.1 Agora

Link: <https://www.agora.io/en/>

### 11.2 AngularJS

Link: <https://angularjs.org/>

### 11.3 MaterializeCss

Link: <https://materializecss.com/>

### 11.4 PubNub

Link: <https://www.pubnub.com/>

### 11.5 Firebase

Link: <https://firebase.google.com/>