



BEIJING-DUBLIN INTERNATIONAL COLLEGE

COMP3017J SOFTWARE METHODOLOGY PROJECT REPORT

OuterView: An Online Technical Interview Platform

Authors

Jiehongxu Wu (21372323)

Sichen Li (21372309)

Te Qi (21372325)

Tongyu Wu (21372320)

Ziqin Ma (21372329)

December 14, 2023

Contents

1	Preface	3
2	System Specification	4
2.1	Customer Problem Statement	4
2.2	Glossary of Terms	4
2.3	User Requirements Definition	5
2.4	System Requirements Specification	6
3	System Design	13
3.1	User Interface Design	13
4	Project Management	16
4.1	Team Agreement	16
4.2	Division of Work	16

1 Preface

The intended readers of this document are primarily technical interviewing platform developers, product managers, and other stakeholders involved in the design, development, and implementation of collaborative technical interviewing software systems. The purpose of this document is to provide a comprehensive overview of the system requirements and customer problem statement, clearly defining the requirements and capabilities of the platform to guide the development team in creating solutions that meet the needs of interviewers and candidates.

This document has gone through several versions, each improving and expanding on the previous version. The rationale for creating a new version is to incorporate stakeholder feedback, update the document with the latest requirements, and improve overall clarity and organization.

A summary of the changes made in each release is as follows:

- Version 1.0
 1. Initial version of the document. Provided an overview of the system requirements of the technical interview platform.
 2. Identified customer problem statements and software system recommendations to solve problems faced by customers.
- Version 1.1
 1. The description of the problem statement was enhanced to include more examples and details, and stakeholder feedback was incorporated to further refine the content and requirements to illustrate the customer challenges.
 2. Drew use case diagrams and completed use case descriptions to improve the accuracy of project understanding.
 3. Completed the user requirements definition part to make the project requirements clearer.
- Version 1.2
 1. A preface has been added to identify the intended readership and provide context for the document.
 2. A glossary of terms was completed to clarify concepts related to professional terms used in the project.
 3. Completed the UI design and production of the front-end main page, reservation meeting page, and meeting information page.
 4. Implemented the backend logic related to sign up, login and reservation meeting and completed the corresponding database design.
- Version 1.3
 1. Completed system requirements definition.
 2. Updated document structure to improve organization and readability.
 3. Further improvements in the backend to meet the need of frontend.
 4. Completed the rendering and basic logic of the frontend main page, reservation meeting page, and meeting information page.

2 System Specification

2.1 Customer Problem Statement

As a customer, I am facing the problem of conducting technical interviews in an inefficient and time-consuming manner. Traditional interview methods often involve face-to-face meetings, which can be challenging due to time constraints, geographical limitations, and the need for specialized equipment. For instance, scheduling interviews with candidates from different time zones or remote locations can be difficult, and setting up a suitable environment for coding tasks can be expensive and time-consuming. Additionally, it can be difficult to accurately evaluate a candidate's skills and knowledge without a proper platform for live collaboration and code execution.

To address these challenges, I suggest a software system that provides a live, collaborative environment for both interviewers and candidates to write, execute, and debug code together. This system should help candidates easily share their skills and support multiple programming languages, such as Python, JavaScript, and Java. It should also have video and audio interviewing functions, scheduling and management features for interviewers, a virtual whiteboard, and the ability to evaluate a candidate's code with automated test cases. Additionally, the system should allow me to replay interviews for future reference, helping me to better assess candidate performance and make more informed decisions.

By implementing this software system, I believe it will significantly improve the efficiency and effectiveness of technical interviews, enabling me to identify the best candidates more quickly and accurately. This will not only save time and resources but also enhance the overall quality of the interview process, leading to better matches between candidates and employers.

2.2 Glossary of Terms

- **Technical Interview Platform:** A website designed to facilitate live, collaborative technical interviews between interviewers and candidates.
- **Browser-based IDE:** A web-based integrated development environment that allows candidates to share their coding skills during interviews.
- **Programming Language:** A set of rules and conventions used to create computer programs.
- **Video and Audio Interviewing:** Functionality that allows interviewers and candidates to communicate via video and audio during the interview process.
- **Scheduling and Management:** Features that enable interviewers to schedule and manage interviews efficiently.
- **Virtual Whiteboard:** A digital whiteboard that allows interviewers and candidates to visualize their technical conversations during interviews.
- **Automated Test Cases:** Pre-written tests that evaluate a candidate's code for correctness and efficiency.
- **Playback Function:** A feature that allows interviewers to replay the entire interview after it has ended.

2.3 User Requirements Definition

Requirements List		
Priority	Type	Description
10	Functional Requirements	Both interviewers and interviewees can see the meeting arrangement in chronological order downwards at the bottom of the homepage.
10	Functional Requirements	Interviewers can initiate audio and video calls or pure audio calls with other designated users.
10	Functional Requirements	Interviewers can cancel a meeting
5	Functional Requirements	Both interviewers and interviewees can set a reminder for their meetings.
10	Functional Requirements	Both interviewers and interviewees can end the meeting session.
8	Functional Requirements	Both interviewers and interviewees can re-enter the meeting session they leave as long as the other users in the meeting remain in the session.
5	Functional Requirements	Both interviewers and interviewees can choose which microphone and camera they use for meetings.
5	Functional Requirements	Both interviewers and interviewees can choose different programming languages for coding in the meeting.
8	Functional Requirements	Both interviewers and interviewees can run the code in the editor and debug them..
5	Functional Requirements	Both interviewers and interviewees can comment and change the code in the editor.
8	Functional Requirements	Interviewers can set the questions and send them to the interviewees..
5	Functional Requirements	Interviewers can send answers to the interviewees.
3	Non-Functional Requirements	Interviewers can set a timer for the session and each question.
10	User Interface Requirements	Both interviewers and interviewees have buttons for booking meetings and joining meetings on the homepage.
8	User Interface Requirements	Interviewers have a button for canceling or changing a meeting in the meeting list on the homepage.
8	User Interface Requirements	Both interviewers and interviewees can see tags containing the name, start time, organizer for each booked meeting in the list on the homepage.
3	User Interface Requirements	Both interviewers and interviewees can see the reminder they set for the meeting in the list on the homepage.
3	User Interface Requirements	Interviewers have buttons for canceling and changing the meeting in the meeting session.

10	User Interface Requirements	Both interviewers and interviewees have a button for exiting from the current page in the meeting session.
10	User Interface Requirements	Both interviewers and interviewees can see the ID and title of the meeting and the username and position of all the users in the meeting session.
8	User Interface Requirements	Both interviewers and interviewees can see the viewing frame of their camera and the other sides' and change the size of all these frames.

2.4 System Requirements Specification

Stakeholders

The stakeholders of the system are the interviewers and interviewees who need to conduct the programming interview. In our case, the interviewer and the interviewee can set up a meeting, so that the interviewer can see the interviewee's programming situation in real time, and make comments and code compilation in real time. The interviewee will also be able to see the interviewer's comments in real time and will be able to communicate via video.

Actors and Goals

Actors and Goals List	
Actors	Goals
Interviewers	Registration and login
Interviewers	Schedule or cancel a meeting
Interviewers	Join or exit a meetingn
Interviewers	Ability to compile and run code
Interviewers	Ability to write comments
Interviewers	Turn on/off microphone and camera
Interviewees	Registration and login
Interviewees	Join or exit a meeting
Interviewees	Ability to compile and run code
Interviewees	Ability to write comments
Interviewees	Turn on/off microphone and camera
Interviewees	Ability to choose programming language

User Cases

User Case Diagram:

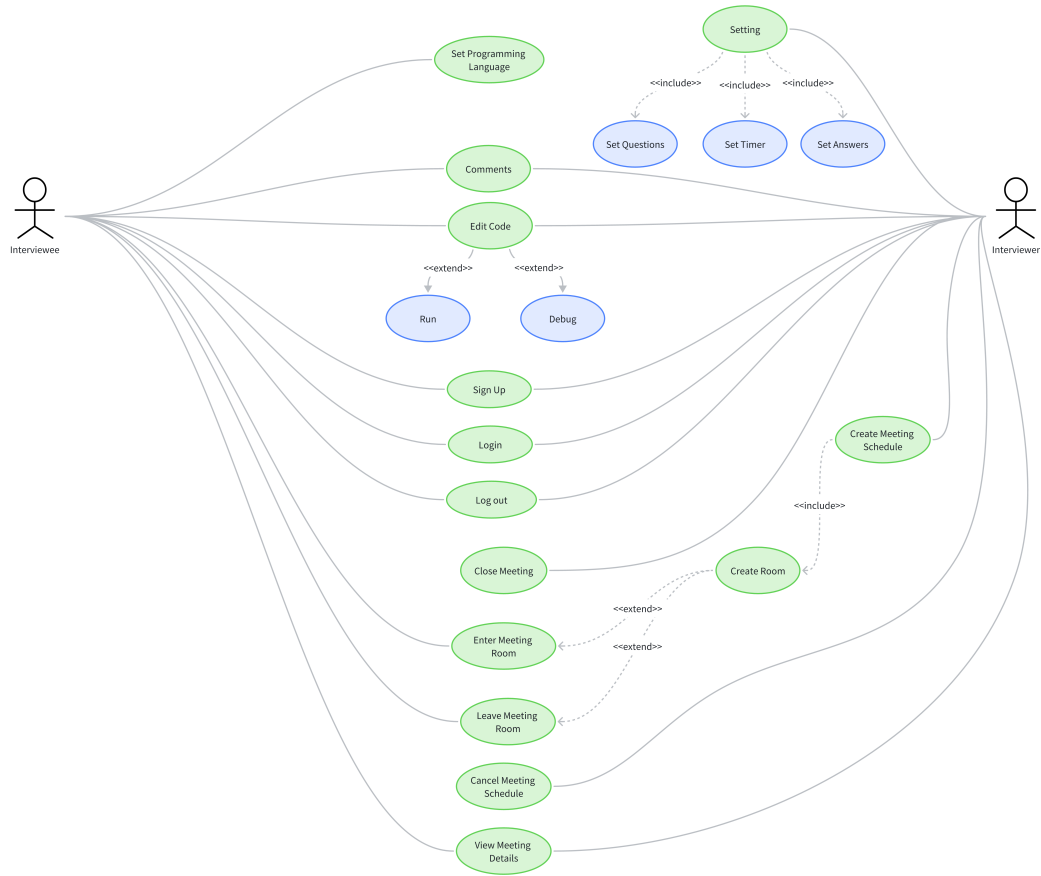


Figure 1: Use Case Diagram

Casual Description:

- **UC1:**
Sign Up: Users can create new accounts.
- **UC2:**
Login: Users can login using the registered account.
- **UC3:**
Log out: Users can exit the current account.
- **UC4:**
Enter Meeting Room: Users can enter the meeting.
- **UC5:**
Leave Meeting Room: Users can leave the meeting.
- **UC6:**
Create Meeting Schedule: Interviewers can create new meeting schedule.
- **UC7:**
Cancel Meeting Schedule: Interviewers can cancel a meeting schedule.
- **UC8:**
View Meeting Details: Users can see the details of a scheduled meeting.
- **UC9:**
Create Room: Interviewers can create new meeting rooms.
- **UC10:**
Close Meeting: Interviewers can end the meeting rooms.
- **UC11:**
Set Programming Language: Interviewees can choose a programming language.
- **UC12:**
Setting: Interviewers are able to set interview questions, answers, answer times, etc.
- **UC13:**
Edit Code: Interviewees can edit code, run and debug code.
- **UC14:**
Comments: Users can write comments.
- **UC15:**
Set Questions: Interviewers can set questions.
- **UC16:**
Set Timer: Interviewers can set a timer for the interviewees in a timed question.

- **UC17:**

Set Answers: Interviewers can set answers for the questions they set.

Detailed User Case Descriptions:

Use Case UC1: Sign Up
Summary: Describes how the user successfully goes to the home page.
Precondition: The Sign Up page must be open in a browser meeting the system requirements.
Trigger: mouse click
Event Process: <ol style="list-style-type: none"> 1. The user will be prompted to enter an user name, user id, email and password. 2. The user inputs the information. 3. The user clicks "Sign Up Now" button to register an account.
Use Case UC4: Enter Meeting Room
Summary: The user (interviewer and/or interviewee) can join the meeting room, start the interview, choose to turn on/off the camera and microphone, open the shared IDE, and exit the meeting room
Precondition: The user is added to the meeting room by the interviewer. The meeting room is available on the user's home page.
Trigger: mouse click
Event Process: <ol style="list-style-type: none"> 1. The user clicks the "Enter Meeting Room" button next to the meeting room to connect to the meeting room. 2. Users enable the camera and microphone (off by default). 3. The screen displays the screen status, microphone status, name and other information of other users in the meeting room. 4. Start the interview. 5. Click Share IDE to share the IDE with others in the meeting room.

User Case UC6: Create Meeting Schedule:
Summary: Users can create meetings.
Precondition: The main window must be open in a browser meeting the system requirements.
Trigger: mouse click
Event Process: <ol style="list-style-type: none">1. If the user wants to create a meeting, click "Create Meeting Schedule".2. Users need to fill in more information for this meeting in the display section. Information to fill in may includes:<ul style="list-style-type: none">• Meeting name (default name is username)• Start time and end time• Duration of the meeting (30min, 45min and 60min for option)• Participants of the meeting3. Users invited to the meeting. This field requires an ID.4. The user clicks "Complete appointment" below the display section to finish filling the information.5. After completion, the user will return to the main page and see the newly scheduled meeting in the right scheduling table.

User Case UC8: View Meeting Details:
Summary: The user can see the details about a meeting.
Precondition: The user has the meeting he or she wants to attend scheduled.
Trigger: mouse click
Event Process: <ol style="list-style-type: none">1. In the configuration table on the right, click the meeting for which the user wants to view specific information.2. The page will provide a pop-up window to show the details of the meeting. The information displayed may include:<ul style="list-style-type: none">• Meeting name (default name is username)• Start time and end time• Duration of the meeting (30min, 45min and 60min for option)• Participants of the meeting3. The user clicks "Enter Meeting" in the pop-up window to complete the request; or click the "Back" button in the upper left corner to close the popup.4. The "Cancel Meeting" and "Edit Meeting" buttons at the bottom of the pop-up window are only eligible for the people who create the meeting schedule to click; the invitees cannot click either button. After clicking "Cancel Meeting", the meeting will be deleted; after clicking "Edit Meeting", the page will jump to the information filling page in the scheduled meeting where the user can modify it.
User Case UC13: Edit Code
Summary: Participants can edit code, run and debug code.
Precondition: The participant enters the meeting and the interviewee selects the language to enter the corresponding IDE
Trigger: mouse click
Event Process: <ol style="list-style-type: none">1. The interviewee chooses the programming language.2. The interviewee enters the code3. The interviewer comments in real time4. The interviewee runs or debugs the code5. The interviewer see the results and communicate by camera or comments.

Acceptance Test Cases:

Use Case: Edit Code
Precondition: The participant enters the meeting and the interviewee selects the language to enter the corresponding IDE
Expected Result: Attendees can edit code, run and debug code in real time.
Steps: <ol style="list-style-type: none"> 1. The participant enters the meeting and the interviewee selects the language to enter the corresponding IDE 2. Participants enter the code 3. Participants click Run or Debug
Actual Result: Attendees can edit code, run and debug code in real time.
Use Case: Create Room
Precondition: The meeting window must be open in a browser meeting the system requirements.
Expected Result: The interviewer can invite the interviewee and create the meeting, and the interviewee can see the invited meeting and accept or decline the invitation.
Steps: <ol style="list-style-type: none"> 1. The interviewer creates the meeting 2. The interviewer chooses the invitee 3. The interviewer sets the time and content of the meeting 4. Interviewers release meetings 5. Interviewers release meetings
Actual Result: The interviewer can invite the interviewee and create the meeting, and the interviewee can see the invited meeting and accept or decline the invitation.
Use Case: Login and Sign Up
Precondition: The user enters the login and registration page
Expected Result: The user successfully registered, the information was saved in the database, and the user was able to log in
Steps: <ol style="list-style-type: none"> 1. Open the registration or login page 2. Enter the username and password 3. Click the Sign Up or sign in button
Actual Result: The user successfully registered, the information was saved in the database, and the user was able to log in

3 System Design

3.1 User Interface Design

User Interface Prototype

User Interface Description

The user interface is divided into four main parts: *Sign Up/Login*, *Joining a Meeting*, *Booking a Meeting*, and a *Meeting Schedule*. The following descriptions provide a brief overview of each feature included and offer preliminary models of some key features.

- **Joining a Meeting:** After clicking to join the meeting, the meeting details will be displayed in a pop-up window.
- **Meeting Schedule:** The meeting schedule contains the meetings that users will participate in in order of meeting start time. After clicking on any of the meetings, the meeting details will appear and the meeting details will be displayed in a pop-up window.
- **Meeting Details:** The content in this pop-up window includes: *meeting name*, *meeting duration*, *start time and end time of the meeting*, *initiator of this meeting*, *canceling the meeting*, *editing the meeting* and *entering the meeting*. The user clicks *Enter Meeting* in the pop-up window to enter the meeting room; or clicks the *Return* button in the upper left corner to close the pop-up window. Only the meeting initiator is allowed to click the *Cancel Meeting* and *Edit Meeting* buttons below the pop-up window; invitees cannot click these two buttons. After clicking *Cancel Meeting*, the meeting will be deleted; after clicking *Edit Meeting*, the page will jump to the information filling page in the *appointed meeting* for modification.
- **Booking a Meeting:** After clicking to schedule a meeting, the user needs to fill in the following information for this meeting in a new page: *meeting name* (the default name is "username's scheduled meeting"), *start time*, *duration of the meeting* (options are: 30min, 45min, 60min) and *the invitees*. This column requires an ID. The user clicks *Complete Appointment* below the page to end information filling. After completed, the user will return to the main page and see the newly scheduled meeting in the *Meeting Schedule* on the right side of the page.

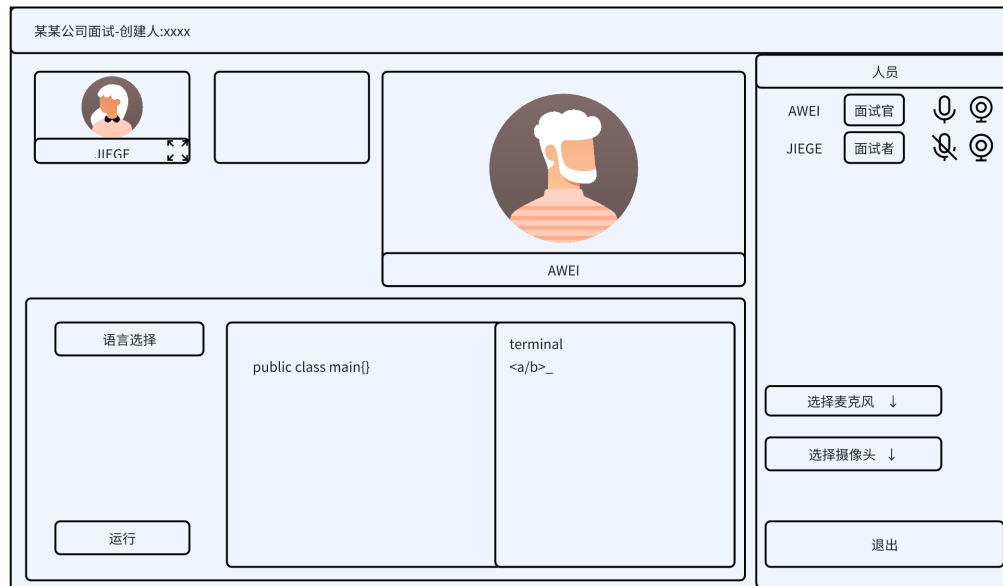


Figure 2: UI Prototype 1

主页面



The main page features a light blue background. On the left side, there is a vertical navigation bar. At the top, it includes a profile icon and a label 'ID' with a link '账户设置' (Account Settings). Below this are two large, rounded buttons: '加入会议' (Join Meeting) and '预约会议' (Book Meeting). At the bottom of the bar is a button with a left-pointing arrow labeled '返回登陆 (退出) 账号' (Return Login (Exit) Account). The main content area on the right is titled '时间' (Time) and contains two rows of meeting information. Each row has a table with columns for '时间' (Time), '会议号' (Meeting ID), '状态' (Status), and a right-pointing arrow. The first row shows '会议名称一' (Meeting Name 1) and '预约一' (Book 1). The second row shows '会议名称二' (Meeting Name 2) and '预约二' (Book 2).

账户设置

ID

加入会议

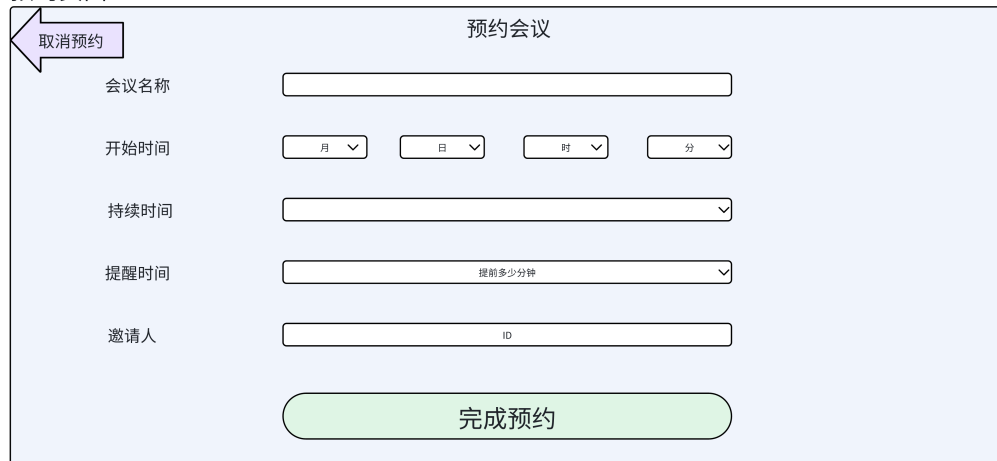
预约会议

返回登陆 (退出) 账号

时间

时间	会议号	状态	
会议名称一		预约一	>
会议名称二		预约二	>

预约页面



The booking page has a light blue background. On the left, there is a button with a left-pointing arrow labeled '取消预约' (Cancel Booking). The main content area is titled '预约会议' (Book Meeting) and contains several input fields: '会议名称' (Meeting Name), '开始时间' (Start Time) with dropdowns for month, day, hour, and minute, '持续时间' (Duration) with a dropdown, '提醒时间' (Reminder Time) with a dropdown labeled '提前多少分钟' (How many minutes in advance), and '邀请人' (Inviter) with a dropdown labeled 'ID'. At the bottom is a large green button labeled '完成预约' (Complete Booking).

取消预约

预约会议

会议名称

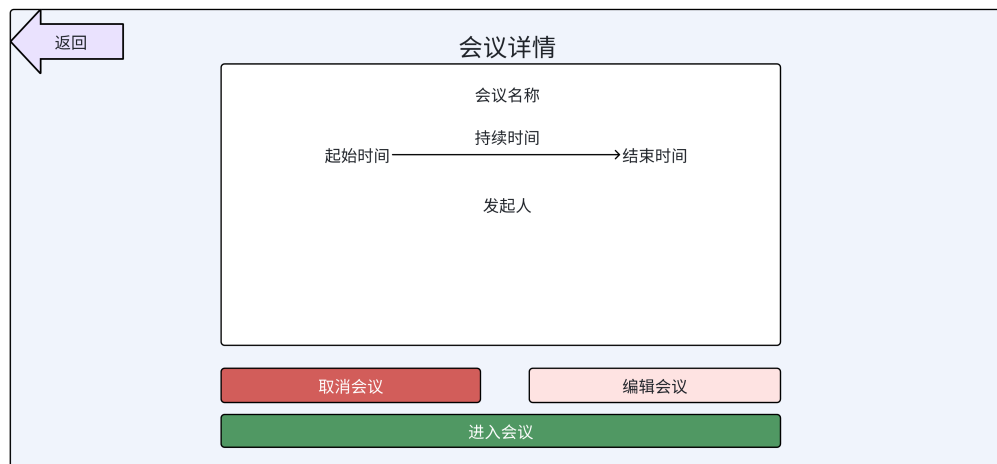
开始时间

持续时间

提醒时间

邀请人

完成预约



The meeting details page has a light blue background. On the left, there is a button with a left-pointing arrow labeled '返回' (Return). The main content area is titled '会议详情' (Meeting Details) and contains a large white box with the following text: '会议名称' (Meeting Name), '持续时间' (Duration), '起始时间' (Start Time) followed by a right-pointing arrow, '结束时间' (End Time), and '发起人' (Initiator). Below this box are three buttons: a red button labeled '取消会议' (Cancel Meeting), a pink button labeled '编辑会议' (Edit Meeting), and a green button labeled '进入会议' (Enter Meeting).

返回

会议详情

会议名称

持续时间

起始时间 → 结束时间

发起人

取消会议

编辑会议

进入会议

Figure 3: UI Prototype 2

4 Project Management

4.1 Team Agreement

- **Meeting Time:** Every Monday at 9pm and every Friday at 5pm.
- **Code Managing:** This project uses git to manage the source code and other related resources. A private repository on GitHub is used for hosting code and team collaboration. Commit message should be clear and follow the convention like
`<type>[optional scope]: <description>`. Any changes that may be deemed destructive should be made on a separate branch and then merged to the main branch after stabilized and being reviewed by at least one other team member.
- **Details for Cooperation:** A clear API document with detailed comments should be provided by the team members who is responsible for the backend. The backend developers and frontend developers should work closely together.

4.2 Division of Work

Frontend:

- Te Qi
- Tongyu Wu
- Jiehongxu Wu

Backend:

- Sichen Li
- Ziqin Ma

**The division of work part will be further elaborated in future version of the document as the project progresses.*