Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 13, No. 1, 2025

The Design and Implementation of Meeting Reminder System Based on uni-app

Yu Qian, Yanping Li

Qinghai Meteorological Information Center, Xining, Qinghai 810000, China

Abstract: In this thesis, the design and implementation process of the meeting reminder system developed based on uni-app was elaborated in detail. By virtue of the cross-platform feature of uni-app and integrating technologies such as Vue.js, Node.js, and JavaScript, a one-stop meeting service system covering meeting information management, intelligent reminders, and multi-terminal adaptation was built for this system. The core functional modules were identified through demand analysis, and the efficient operation of mini-programs and mobile apps was achieved through lightweight deployment on cloud servers and collaborative development with front-end frameworks. In this thesis, the discussion was carried out from the perspectives of system requirement analysis, architecture design, functional module implementation, and test optimization, and the effectiveness of the system in enhancing the efficiency of meeting management and user experience was verified.

Keywords: vue.js; uni-app; Web Development; Meeting Reminder; Cross-platform Application.

1. Introduction

1.1. Research Background and Significance

In the face of todays fast-paced modern working environment, as the core carrier for internal decision-making communication and external collaboration and connection within organizations, the frequency and complexity of conferences have grown exponentially. According to statistics from a workplace efficiency research institute, white-collar workers in first-tier cities attend an average of 5 to 8 meetings per week, among which large-scale meetings that cross departments and levels account for more than 35%. However, the drawbacks of the traditional meeting management model are becoming increasingly prominent: firstly; as the transmission of meeting information relies on decentralized channels such as emails and instant messaging tools, it is easy to cause information omissions; according to a certain survey results, approximately 28% of people in the workplace have missed important meetings due to the failure to receive meeting notifications in time; secondly, frequent time conflicts and the difficulty in real-time synchronization and manual recording of schedules have led to 15% of meetings being rescheduled due to time conflicts among participants; thirdly, the supporting services for large-scale meetings (such as document search, seat positioning, and transportation arrangement) lack a unified access point, which requires participants to switch between multiple applications and results in a relatively poor experience.

It is of significant practical significance to develop an efficient meeting reminder system in such context. uni-app, as a cross-platform framework based on Vue.js, has the feature of "one-time development and multi-terminal operation", which becomes the key to solving the above problems. Compared with the traditional multi-terminal development model (which requires writing code separately for WeChat mini-programs, Android apps, and iOS apps), uni-app can reduce the repetitive development workload by more than 60%, and can increase the code reuse rate to over 80% through the componentization concept of Vue.js[1]. The meeting reminder system built based on this framework can not only integrate the entire process of meeting services (from information release to post-meeting material archiving), but also ensure a consistent experience for users in scenarios such as wechat mini-programs and mobile apps through multi-terminal adaptation, effectively breaking down "information isolated island" and "terminal barriers".

From the perspective of practical application value, the system can significantly enhance the efficiency of three types of users: enabling administrators to centrally manage and batch operate meeting information, and reducing manual maintenance costs by 40%; enabling participants to reduce the meeting preparation time by 50% through intelligent reminders and one-stop services; enabling meeting organizers to manage the entire meeting process through data, providing a basis for decision-making optimization.

1.2. Current Research Status at Home and Abroad

The current meeting management applications mainly fall into two technical routes, but both have obvious limitations:

Although native development Apps, such as Microsoft Teams and Zoom, support full-feature meeting management, the development cycle for those adopting the native development models (Android is based on Java/Kotlin, and iOS is based on Swift) can last for 6 to 12 months and have high maintenance cost (it needs to be iterated separately for the two platforms). According to the data from a certain enterprise, the cross-version compatibility issue of native Apps has led to 10% of users being unable to use core functions due to system version mismatch.

uni-app has taken an important position in the cross-platform framework application field by virtue of the advantages of the Vue ecosystem. The translation program developed based on uni-app has successfully achieved seamless adaptation between WeChat mini-programs and the H5 terminal, verifying the reliability of the framework in multi-terminal compatibility [1]; the system response speed was increased by 40% through the integration of uni-app and cloud services[2]. In contrast, although other cross-platform frameworks (such as Flutter and React Native) have performance advantages, uni-app has lower learning costs in

enterprises where Vue technology stacks are popular (developers do not need to switch programming languages).

The innovation point of this research is that a "whole-process + cross-platform" meeting service system based on uni-app was built for the first time, and the gap in the existing systems full -scenario services and multi-terminal adaptation was filled by integrating nine major modules such as meeting arrangements and weather inquiries.

2. System Requirement Analysis

2.1. Functional Requirements

This system was designed to meet the core requirements of users throughout the entire life cycle of the meeting. The specific functional module designs are as follows:

User Management Module:

Login approaches: Supporting two types of scenario-based logins.

Code scanning login: Suitable for quick verification at the conference site; users can obtain the website by scanning the code through WeChat.

IP login: For enterprise Intranet users, the IP address resolution (such as the 192.168.1.xx segment) can be performed through the back-end to automatically associate the users department with common meeting types (such as "Meteorological Business Meeting", "Department Regular Meeting"), and display it on the home page.

Permission control: The RBAC (Role-Based Access Control) model is adopted and divided into three levels of roles:

Super administrator: Full permissions (adding, deleting, and modifying meetings, managing user roles);

Meeting administrator: Only meetings created by the user can be managed.

Attendee: Only the meeting information related to the user can be viewed, and there is no editing permission.

Meeting Information Management Module:

Meeting query: Supporting multi-dimensional filtering, including:

Session screening: The screening is performed by meeting level (such as "Plenary Meeting", "Departmental meeting");

Information provision: Supporting keyword matching (e.g., when "meteorology" or "emergency" is entered, it will automatically match the results containing the word in the conference themes, locations, and host names); supporting pinyin initial letter association (e.g., when "qx" is entered, it will match meetings related to "meteorology").

Information maintenance: Administrators can add/edit meeting information through the form (including 12 core fields: meeting ID, theme, time, location, host, participants, agenda, remarks, attachments, seat map, transportation information, and meal arrangement), supporting batch import in Excel (the template includes verification for required fields, and a detailed error log will be returned when the import fails).

Data Management: Supporting PDF, PowerPoint, and Word format files (single file ≤50MB), achieving:

1) Preview: It is possible to open files online (supporting page jumping for PDF files and animation playback for PowerPoint files).

2) Download: Download control can be carried out based on permissions (such as "Download only for attendees" or "Public download"), and download logs (user, time, file) can be recorded.

3) Version Management: Retaining file modification history and supporting returning back to historical versions.

Automatic Push Scenario:

Meeting change: pushing updated content in real time when the meeting time, location, and host are changed (comparison of old values versus new values).

Weather warning: In case of heavy rain, strong wind or other adverse weather conditions at the meeting venue within the next two hours, it will send out warning information and travel suggestions.

Check-in reminder: It sends a "About to Start" prompt (including a quick check-in entry) 10 minutes before the meeting begins.

Additional Service Module:

Basic services: Meeting seats (JPG seat image + zoom in and out function, supporting clicking on the seat to display the name), bus grouping (vehicle number, driver contact information, assembly time and location), meal arrangement (restaurant location + navigation link, meal time, meal type);

Extended services: On-site services (hotel address, front desk tel., check-out time), weather during the meeting (real-time temperature, 7-day forecast, wind force level), and precautions (such as "ID card required", "No photography in the venue").

2.2. Non-Functional Requirements

Performance requirements:

Response speed: The first loading of the page is ≤3 seconds (including resource loading), and the second loading is ≤1.5 seconds (through cache);

Inquiry Efficiency: Single-condition inquiry is ≤1 second, multi-condition combined inquiry is ≤2 seconds;

Concurrent support: There is no lag when the number of concurrent online users is ≥200 and the peak request volume (such as 30 minutes before the meeting starts) is ≥50 times per second.

Compatibility requirements:

Terminal Coverage: WeChat Mini Program (iOS WeChat 8.0+, Android WeChat 7.0+), Android APP (Android 6.0+), iOS APP (iOS 10.0+).

Screen Adaptation: supporting screens from 4.7 inches (such as iPhone SE) to 6.7 inches (such as iPhone 14 Pro Max), and adopting flexible layout (Flex) to ensure element adaptation.

Safety requirements:

Data transmission: HTTPS encryption (TLS 1.2 protocol) is adopted for all interfaces to prevent man-in-the-middle attacks.

Identity verification: The validity period of scanning code login is ≤5, After logging in with an IP address, it is bound to a commonly used device (a second verification is required when changing devices).

Storage safety: User password is encrypted and stored using the bcrypt algorithm (salt-hash, not reverse-decrypted). Sensitive meeting information (such as undisclosed agendas) needs to be accessed after logging in.

Usability requirements:

Simplified operation: Core functions (such as querying meetings and setting reminders) can be completed within 3 steps. Button size ≥48px (conforming to mobile touch standards to avoid accidental touches).

Internationalization support: Built-in Chinese-English switching (including date and time format adaptation; e.g. “2024年9月6日” in Chines, and "Sep 6, 2024" in English).

Fault tolerant design: Real-time prompts are given when

input errors occur (such as incorrect date format), and offline cache data is displayed when the network is interrupted.

3. System Design

3.1. Overall Architecture Design

The system adopts a three-layer architecture of "front-end - service - storage" (as shown in Figure 1), and the responsibilities of each layer are clear and efficiently coordinated:

Front-end layer: It is built based on uni-app, and its core technology stack includes:

View layer: Vue.js component-based development (page components, general components such as buttons, forms), and the the interface structure is defined through Template syntax.

Logic layer: JavaScript is used to handle business logic (such as login verification, data requests), and Vuex is used to manage global states (such as user information, current meeting IDs).

Style layer: The interface style is controlled by CSS/SCSS. the conditional compilation provided by uni-app (such as #ifdef MP-WEIXIN) is used to adapt to different terminals (some buttons need to be hidden on the Weixin Mini Program and displayed on the APP terminal).

The front-end communicates with the service layer through a RESTful API and exchanges data in JSON format. The typical request process is: user-triggered operations (such as query meetings) → front-end encapsulation parameters (time range) → sending an HTTP request (GET /api/meeting) → receiving response data → rendering the page.

Service layer: It is deployed on Alibaba Cloud lightweight application server (2-core 4G configuration) and built on Node.js (Express framework), and its core functions are as follows:

API interface: There are 32 core interfaces (login, meeting CRUD, reminder configuration, file upload, etc.), and interface authentication is implemented through JWT (JSON Web Token) (the token is valid for 2 hours and will be automatically refreshed upon expiration).

Business logic: handling data verification (such as meeting time format validation), permission determination (such as ordinary users not being able to delete meetings), third-party service invocation (weather API).

Cache management: It is integrated with Redis (in-memory database), and caches frequently accessed data (such as weather information and popular meeting lists) to reduce the pressure on the database.

Storage layer: It adopts a hybrid storage strategy, considering both data types and access efficiency.

Structured data: User information and basic meeting information (time, location, etc.) are stored in the MySQL database (built into the cloud server, supporting master-slave backup), and the InnoDB engine is used to ensure transaction consistency.

Unstructured data: Meeting materials (PDF and PowerPoint files), seat maps (JPG files) are stored on Alibaba Cloud OSS (Object Storage Service). Its has the following advantages:

1) High availability: 99.99% service availability, supporting disaster recovery and backup;

2) Elastic expansion: Storage capacity can be expanded as needed (initially 100GB, and can be dynamically added);

3) Secure access: controlling file access rights by signing URLs (valid for 1 hour) to prevent link leakage.

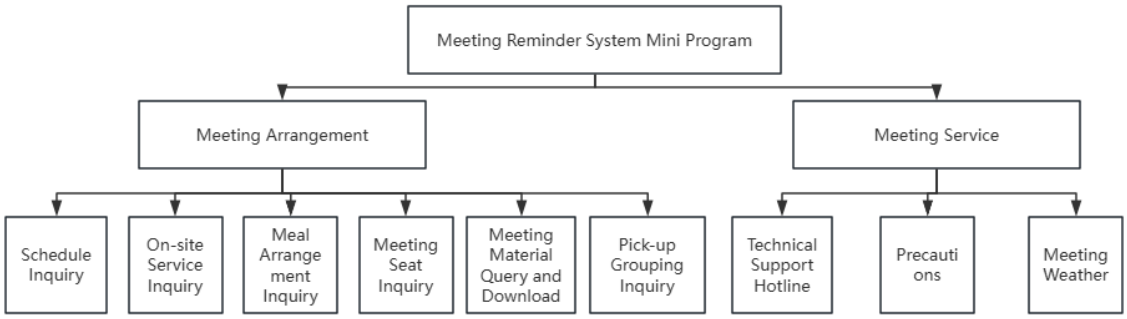

Figure 1. Overall System Architecture Diagram

3.2. Module Design

3.2.1. User Management Module

Login Process:

Scan QR code to log in: suitable for quick verification of conference site, and the steps are as follows:

When using the meeting reminder system mini-program for the first time, users can directly scan the code to log in to the home page. Non-WeChat authorized login ensures the convenience of mini-programs. IP address can also be used for login to provide users with a more personalized service experience.

3.2.2. Meeting Arrangement module

Users can see the meeting poster of the day at the center of the home page, as shown in Figure 3. The middle part of the home page displays the major functional module access points of the program, and users only need to click on the access point of a certain function to enter the corresponding function page. For instance, After the user clicks on the "Daily Schedule", it will enter the inquiry page for the meeting day. The page effect is shown in Figure 4. Attendees can check the meeting arrangements by dimensions such as time, session, and theme through the mini program, and better plan their personal schedules and receive schedule reminders.

3.2.3. Key Technology

Development stage: The mini-program was developed using the uni app framework, the front-end page design was completed, and a simple and intuitive interface was carefully designed based on the principle of user experience. The meeting list, details, reminder settings and other pages were designed to have a reasonable layout, clear operation process, and be compatible with multiple screen sizes, ensuring that users can enjoy a consistent high-quality experience on different devices. The logical development was carried out focusing on the core functions of the meeting reminder system, and the interface connection and other tasks were

accomplished through RESTful API for data interaction. The front-end page can obtain meeting information, weather information, etc. through the calling interface to ensure the accuracy and stability of data transmission.

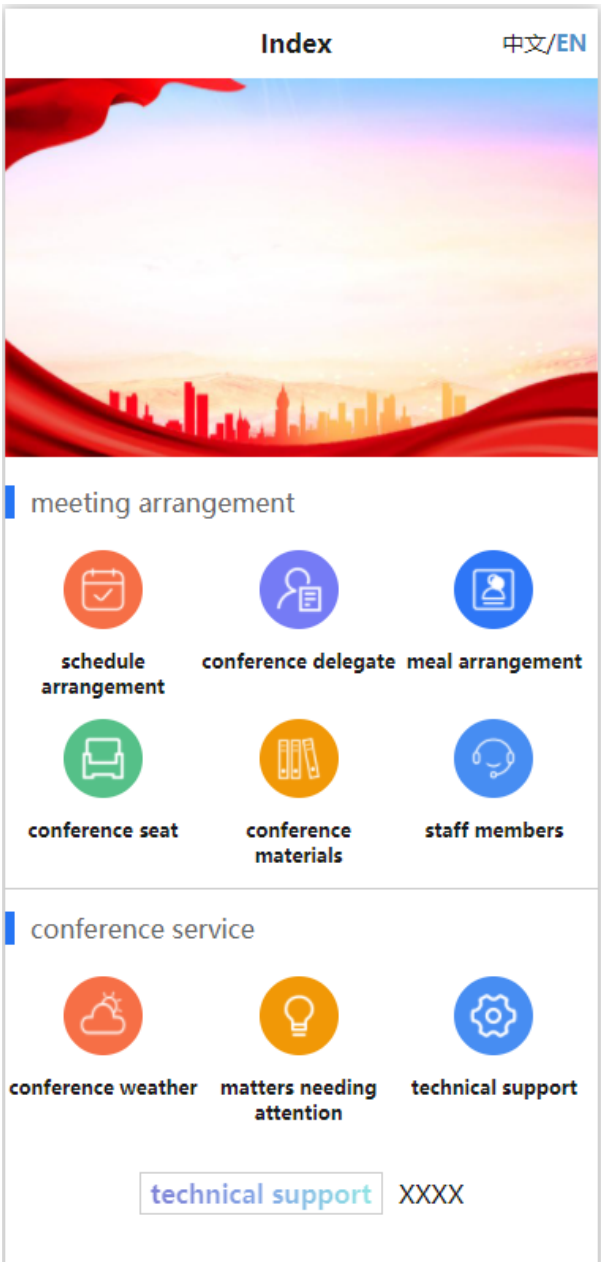

Figure 2. Home Page Design

Figure 3. Meeting Arrangement Module Design

Deployment stage: the developed mini-program was deployed to the cloud server, the appropriate cloud server was selected based on the user volume and data processing requirements of the mini-program, and the installation of the operating system, network environment configuration, firewall Settings, etc. were completed to ensure the performance and security of the server. The domain name was bound, enabling the user to conveniently access the mini-program through the domain name resolution. In addition, the server configuration and other operations were completed [4].

Function expansion and integration: a real-time weather interface was integrated to provide users with real-time weather information based on the meeting location, facilitating attendees to make travel preparations in advance. Meanwhile, it provides a function for downloading meeting materials, allowing users to conveniently obtain relevant meeting documents within the mini-program, further enhancing the practicality and user experience of the mini-program. As shown in the Figure 4.

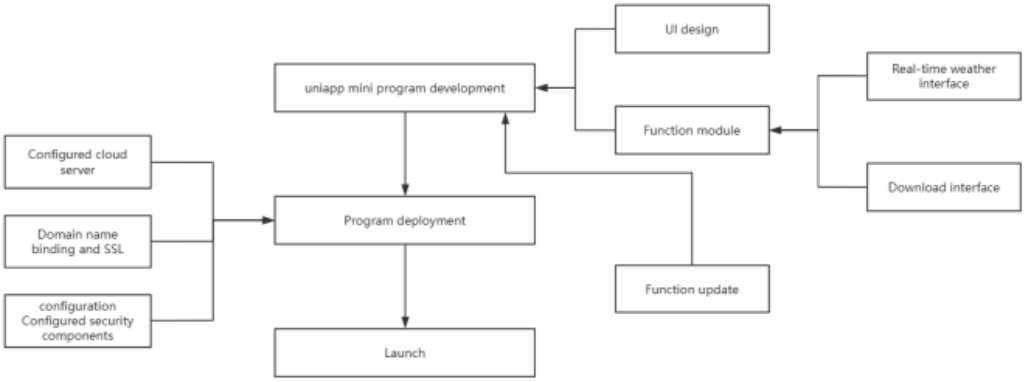

Figure 4. Deployment Process

4. System Implementation

4.1. Implementation of Meeting Materials Management

Meeting materials can be previewed or downloaded online on the meeting information page. The meeting information page supports multiple file formats, such as PDF, PPT, Word documents, etc. Online preview requires the use of a third-party library or service, such as PDF.js for the PDF preview function. Files need to be stored in a cloud storage service such as Alibaba Cloud OSS or AWS S3[2], and then provided to users for download by generating temporary access links.The page effect is shown in Figure 5.

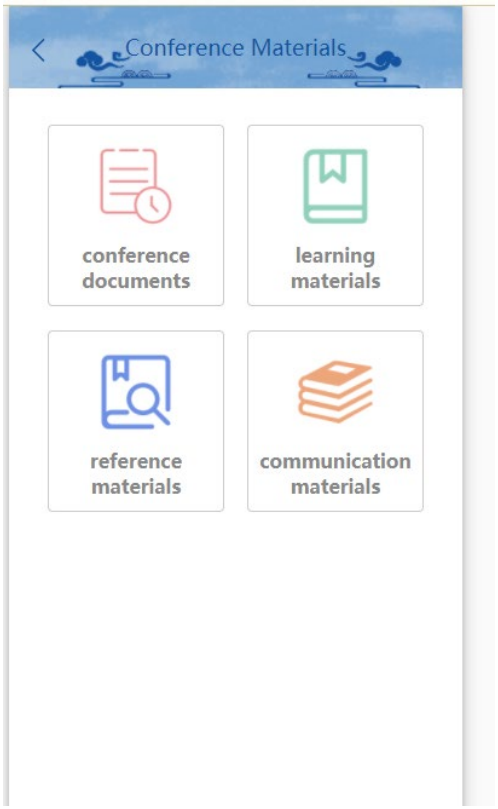

Figure 5. Design of Meeting Material Management Module

The codes are as follows:

```
<view class="title">conference manual</view>
</view>
<view class="right">
<image src="../../static/imgs/index/xz.png" mode="" @click="doDown('http://sipff4c3a.hn-bkt.clouddn.com/%E7.pdf','a.pptx')"></image>
<image src="../../static/imgs/index/gd.png" mode="" @click="previewFile('http://sipff4c3a.hn-bkt.clouddn.com/%E7.pdf')"></image>
</view>
</view>
<view class="item">
<view class="left">
<view class="no">2</view>
<view class="title">Main Event Work Manual</view>
</view>
<view class="right">
<image src="../../static/imgs/index/xz.png" mode="" @click="doDown('http://code.gqqf.xyz/doc.docx','a.pptx')"></image>
<image src="../../static/imgs/index/gd.png" mode="" @click="previewFile('http://code.gqqf.xyz/doc.docx')"></image>
</view>
```

4.2. Inquiry of Weather for the Meetings

uni.getLocation was used to obtain the users location and then call the weather API. The issue of location permission was handled, the weather API was called, the weather data was cached, and Redis was used to store the most recently requested weather data [3], thus reducing the number of API calls and improving performance. Real-time warning information was added to display the forecast for the next 7 days. The page effect is shown in Figure 6.

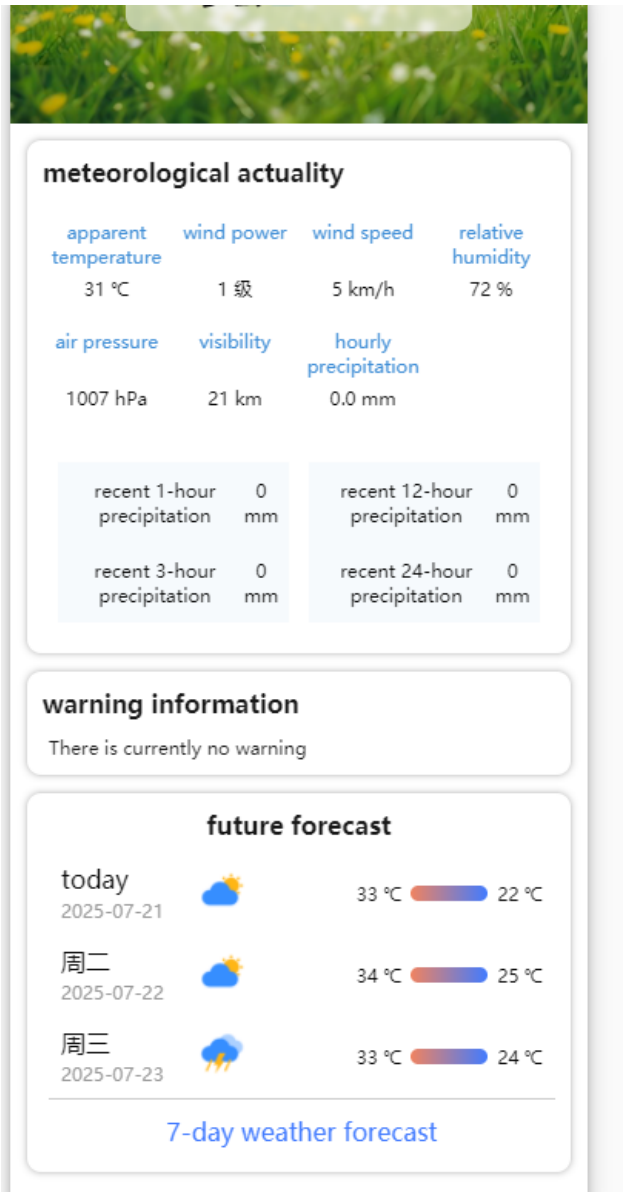

Figure 6. Design of Weather for Meetings Module

The codes are as follows:

```
<template>
<view class="index">
<view class="head">
<image src="../../static/imgs/back.png" mode="" @click="goback" class="back"></image>
<!-- <image src="../../static/imgs/weather.jpg"
```

```
mode="aspectFill"></image> -->

<view class="area">
<view class="name">{{ addrInfo.name }}
{{ addrInfo.country }}</view>
<view class="ssd">
{{ nowInfo.text }}   
  <!-- <i :class="qi-+nowInfo.icon" style="padding: 0
16rpx 0 12rpx;"></i>    -->
<!-- <image
class="icon" :src="https://codermoyv.gitee.io/coder-
moyv/assets/images/wechat/weather_custom/+nowInfo.icon
+.png"></image> -->

<image
class="icon" :src="https://images.weserv.nl/?url=https://a.he
cdn.net/img/common/icon/202106d/+nowInfo.icon+.png">
</image>
{{ nowInfo.temp }} ℃
</view>
```

4.3. Other Services and Inquiries for the Meetings

Users can switch between Chinese and English display at the center of the home page. Under the home page image, highlighted meeting prompts are provided, along with detailed inquiries about on-site services and meal and transportation arrangements. JPG format meeting seats are also available for users to view. The page effect is shown in Figure 7.

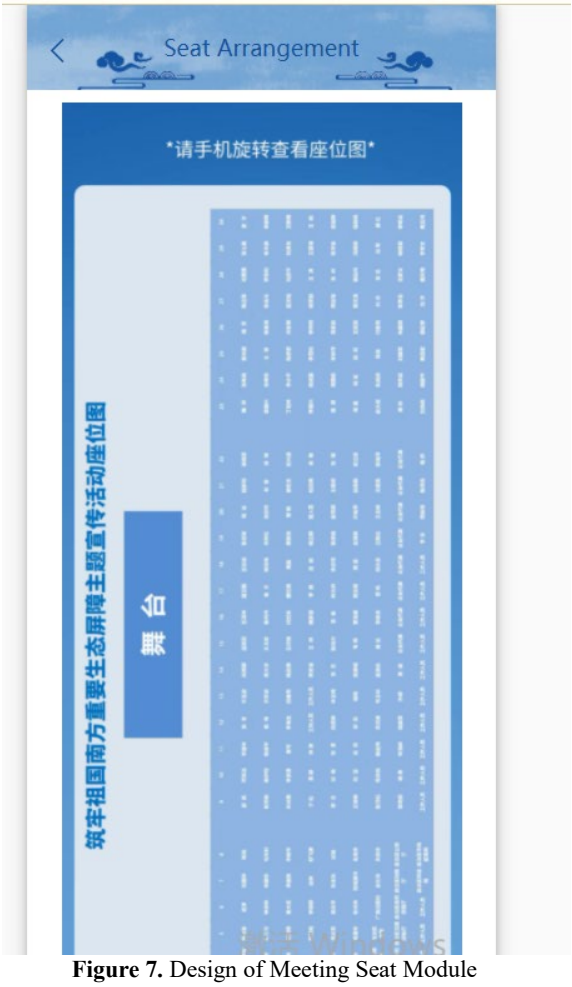

Figure 7. Design of Meeting Seat Module

5. System Testing

5.1. Functional Testing

Fifty test cases were designed using the black-box testing method, covering core functional scenarios. Some key use cases are as follows:

Module - Test Cases - Input - Expected Output - Test Results

Login Module - Scan the code to log in - (valid QR code) (QR code containing the correct meetId and user identification)- Login successful, jump to the meeting details page - Pass

Login Module - Scan the code to log in (invalid QR code) (QR code with random string content)- Prompt "QR code format error" - Pass

Data Management - Download Permission Verification (for non-participants) - Non-participants click the download button - Prompt "No download permission" - Pass

Weather Inquiry - User refuses location permission - Click "Do not Allow" location - Display the default weather of the meeting location (Xining) - Pass

The test results showed that all 50 use cases have passed and there are no defects in the core functions.

5.2. Performance Test

Response time test: JMeter was used to simulate concurrent access by 200 users. The key indicators are as follows:

Home page loading: Average 1.8 seconds (demand ≤3 seconds), 90% response time was ≤2.5 seconds;

Meeting inquiry: Average 0.6 seconds (demand ≤1 second), 90% response time was ≤0.8 seconds;

File download (10MB PDF) : Average 8 seconds (affected by network), supporting resume from breakpoint;

Compatibility testing: Verified in mainstream terminal and system versions;

WeChat Mini Program: iOS (wechat 8.0.30), Android (wechat 8.0.28), normal interface adaptation, no functional abnormalities;

Mobile APP: Huawei P50 (Android 12), iPhone 13 (iOS 15), Xiaomi 12 (Android 13). All functions were operating normally, and the styles were adaptive to different screen sizes.

Safety testing:

Interface authentication: The meeting delete interface was accessed through an expired token and then returned to a 401 error (unauthorized);

File access: An invalid OSS signature URL was used to download files and then returned to a 403 error (access prohibited);

Data encryption: User passwords were stored in the database using bcrypt hash (cannot be decrypted in reverse).

5.3. Test Conclusion

The system functions completely covered the requirements and specifications, all performance indicators met the standards (response time, concurrency support), good compatibility on mainstream terminals, the security met the design standards, and the actual deployment conditions were met.

6. Conclusion and Prospects

6.1. Conclusion

This system is built based on the uni-app cross-platform

framework, integrating the form of mini-programs and mobile apps to achieve intelligent management of the entire meeting process. This system provides core services such as meeting query, material download, and weather warning supported by a Node.js+ cloud server architecture and combined the React/Vue front-end technology stack. The system uses Alibaba Cloud OSS to store meeting materials, utilizes Redis to cache weather data for performance optimization, and supports multi-format preview in PDF/PPT/Word. The modular design covers scenarios such as schedule management, on-site services, and bus grouping. The code scanning login and IP positioning functions take into account convenience and personalization. It effectively enhances the efficiency of meeting organization through an intelligent reminder mechanism and cross-device data synchronization, creating a one-stop smart conference experience.

6.2. Prospects

In the future, optimization can be carried out in the following aspects:

Intelligent reminder algorithm: Based on the users historical data (such as the reminder response time of the past 10 meetings), the optimal reminder time is predicted through machine learning (such as logistic regression models) (for instance, if a user is accustomed to preparing 20 minutes in advance, this time is automatically recommended).

Data visualization statistics: the ECharts chart library is used to display data such as the meeting participation rate (on-time attendance/latecomer/absence ratio) and the frequency of departmental meetings, and generate an administrator dashboard.

Extended hardware integration: It connects to smart bracelets (such as Huawei bands) and sends meeting reminders via Bluetooth, covering scenarios where users are not looking at their phones.

AI-assisted functions: It integrates large language models (such as ChatGPT) to automatically generate meeting minutes (based on voice-to-text conversion) and perform real-time translation in multiple languages (supporting participants to communicate in different languages).

Through continuous iteration, the system can Through continuous iteration, the system can evolve from a "meeting reminder tool" to an "intelligent meeting assistant", further releasing the value of technology in improving organizational collaboration efficiency.

References

[1] L. Gong. “The Design and Implementation of Translation Program Based on uni-app Framework” [J]. Computer Programming Skills & Maintenance, 2025, (02): 23-25. DOI: 10.16184/j.cnki.comprg. 2025.02.035.

[2] Q. S. Cui, L. Yang, Y. Zh. Dou, et al. “Design and Implementation of University Network Fault Reporting and Repair System Based on uni-app”[J]. Information Recording Materials, 2024,25 (12): 151-153+180. DOI:10.16009/j. cnki. cn13-1295/tq. 2024.12.039.

[3] H. Min, J. J. Zhang. “The Design and Implementation of Online Course Reservation Mini-Program Based on uni-app and Express” [J]. Modern Information Technology, 2024,8 (17): 100-104+110. DOI: 10.19850/j.cnki. 2096-4706.2024.17.019.

[4] J. Zh. Li, Zh. M. Hu, J. G. Chen, et al. “The Design and Implementation of the Self-study Room Platform Based on SpringBoot and Uni-App Framework” [J]. Journal of Hebei University of Water Resources and Electric Engineering, 2024, 34 (02): 77-82. DOI: 10.16046/j. cnki. issn 2096-5680. 2024. 02.014.