/MEETING MINUTES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Meeting/Project Name:** | | Daily Meeting | | |
| **Date of Meeting:** | | April 10, 2022 | **Time:** | 15:00– 16:00 |
| **Minutes Prepared By:** | | Mu Di | **Location:** | X9318 |
| **1. Meeting Objective** | | | | |
| Front-end group showing and discussing the style of the Mini Program venue details page and reserving page. The back-end group discussed the work progress and problems. | | | | |
| **2. Attendees** | | | | |
| Chen Runsheng, Cheng Weishi, Fan Shiqing, Li Peishuo, Mu Di, Sheng Junjie | | | | |
| **3. Review** | | | | |
| **Topic** | **Discussion** | | | |
| **Front-end Group Report** | The front-end group completed the page layout and content of the venue display and reserve. They also completed the list display of venues, facilities and activities in the management system. It was rendered correctly with suitable data. Corrections for submitting data forms were also successfully completed. All members agreed that the current page meets the requirements. | | | |
| **Back-end Group Report** | The "Orders" class is partially complete, implementing the definition of the entity and the writing of the interface. In addition, all orders can be queried through Prediction class combined with Pageble. And unit tests against the code that has been written so far. The API documentation section completes the writing of User and Venue and is ready to dock with the front end. | | | |
| **Summarizing on issues** | All team members approved of the current styling of the front-end group and decided that the next plan was to focus on the core function of reservation. | | | |
| **Front-end Group Discussion** | The front-end group discussed what to do after the page style was completed; the Mini Program is now able to display venue listing information and has access to the booking page, the next step is to interface with the booking function completed by the back-end group.  At present, the management system has completed all venues, facilities and activities mainly related to reservations. In the next step, the front-end group will design the user interface and forms for orders and user management and discuss with the back-end group the API interface about its functionality. | | | |
| **Back-end Group Discussion** | The team decided that it was time to extend the API further and design the Orders class further, at a minimum, with reservation, payment, and refund functionality. The payment and refund functions are implemented by the Flag field. We believe that orders need to be divided into 5 categories: failed, unpaid, paid, refund and used. Different Settings should be set for different payment methods. If the user pays in cash, the order flag should be set to pay, changing from background to used or paid. The user can make a refund at any time through Cancel, and if the user makes a wrong reservation (for example, at the same time), it becomes a failure. If payment is made online, it is converted to paid. | | | |

|  |  |
| --- | --- |
| **4. Plan** | |
|  | **Action** |
| 1 | Further enrich the API documentation |
| 2 | Front-end group and back-end group should work together to complete the connection of both ends with reserve function of activities and facilities, as well as the test on related functions. |
| 3 | According to the existing ideas to achieve the order booking refund and other functions |