**Script**

**开场介绍**

第一页：

Hello everyone, this is Group 33. Today, I am excited to present our project: the Online Meeting Booking System. And here are the group members.

第二页：

Our presentation covers five sections: Introduction & Background, Architectural Design, Software Implementation, Software Testing, and Project Conclusion.

**Introduction**

第三页：

Let's begin with the Introduction.

Our project aims to provide a high-quality, efficient, user-friendly, and secure meeting room booking system for students and administrators.

第四页：

We designed two access levels:

Staff and students have regular access to book rooms, check real-time availability, view detailed room info with photos, and filter rooms by capacity, name, or location.

Administrators have superior access to manage rooms, IT facilities, approve/reject bookings, handle user accounts, and reply to feedback efficiently.

We identified potential risks: tech stack integration issues, usability challenges (filters/calendar), security concerns (weak sessions, email code rate-limiting), and ethical considerations for handling student data responsibly.

**Architectural Design**

第五页：

Next, the architectural design.

To meet reliability, performance, ease of use, security and continuous improvement requirements, we designed a modular layered architecture. （这里鼠标可以指向旁边的结构）.

Here are the tools, services, and plugins used.

第六页：

Frontend uses HTML, CSS, JavaScript with the Vue framework for unified management, routing, and dependency configuration.

Axios handles backend communication; its interceptors streamline request handling, error management, and loading states.

第七页：

Our backend is built on Spring Boot for robust project foundation and centralized configuration. We leverage MyBatis Plus for simplified database operations with wrapper-based queries that improve SQL readability. Spring AOP for Log management and identity verification.

For performance, Caffeine cache handles high-frequency data with low latency. Qiniu OSS provides cloud storage for files with CDN support. Finally, Knife4j generates interactive API docs for developer testing and integration.

第八页：

Here’s the designed Entity-Relationship (ER) diagram outlining the database design. Our table design follows the three normal forms for efficient management.

**Software Design**

第九页：

Then, we step to software design.

The project is organized into key software modules. As shown, this modular structure ensures clear separation by user role, enhancing scalability and maintainability.

第十页：

The process activity diagrams below showcase the operational workflow of our system, highlighting the different permissions and functionalities for users and administrators. The left side represents the admin activity, the other is for users.

第十一页：

Software Support Services cover various project aspects, offering targeted support for system completeness, efficiency, and long-term maintainability

第十二页：

We adopted the MVC architecture (Model, View, Controller) for modularity and maintainability.

We follow five core principles: **separation of concerns**, **high cohesion/low coupling**, **maintainability**, **testability**, and **RESTful API design**.

第十三页：

And you can see our layered architecture here, each package has its own functionalities. Shared functional objects are extracted to support code reuse and ensure the consistent behavior across the system.

As for **coding conventions**, we adopted unified naming, structured errors, and clear API contracts to keep the project well-organized and easy to maintain.

**Software Testing**

第十四页~第十六页：

Now let’s check out **Software Testing**.  
For unit testing, we followed five basic steps:  
**Identify Target Unit**, **Understand Expected Behavior**, **Isolate the Unit**, **Write Test Case**, and **Run the Test**.  
Here we show several examples, such as to test controller’s ability to query a booking by ID （翻页），obtain user list（翻页）, See reservation records.

第十七页：

Next, let’s take a look at our integration testing.  
We followed six steps:  
**Define Scope**, **Prepare Environment**, **Design Scenarios**, **Execute Tests**, **Verify Results**, and **Report & Fix**. This example tests the lock/unlock and email send function in the AdminUserService.

By conducting this, we ensure that cross-module interactions behave as expected.

第十八页：

Finally, we conducted acceptance testing to ensure the system meets user requirements and is ready for deployment.  
We **defined clear acceptance criteria** based on user needs.  
Then, we **designed test scenarios** that simulate real-world usage.  
Followed by this, we **prepared a realistic test environment and data**.  
Next, we **executed the test cases** by walking through the booking process as an end user.  
At last, we **evaluated the results** to ensure the booking appeared correctly in the user's booking list.

**Conclusion and Future Improvements**

第十九页：

In summary, our group successfully developed a web-based meeting room booking system for universities. Throughout this process, we applied the Scrum framework effectively, gaining hands-on experience in agile development through iterative sprints. We also deepened our understanding of legal, ethical, and security considerations in system design.

Our achievements include a complete set of deliverables covering system specification, design, implementation, and evaluation.

That said, we also encountered some limitations — such as limited frontend styling, inadequate query and loading performance, and the Lack of coping strategies for high-concurrency scenarios.

Looking ahead, we will continue evolving the system as outlined in the timeline below.

演示流程：

用户端：

注册：

这里是我们的登陆页面，如果你是新用户那么可以从注册开始。首先你可以选择一个头像或者直接食用默认头像。然后输入一个不重复的用户名，student id和密码。密码需要满足不少于八位长度，邮箱地址必须为有效邮箱地址。然后发送验证码校验邮箱是否合法。验证码有限流机制，每分钟只能请求一次。收到验证码之后，验证码有效期为3分钟。然后就可以成功注册。

登录：

我们可以使用用户名或邮箱登录，系统会自动判断你的角色，然后进入对应页面。

房间：

用户页面首先可以查看所有房间以及每个房间的详细信息，然后可以根据需求筛选想要预定的房间。

当点击 book 按钮之后可以看到该房间在未来一周内的详细预定情况。然后可以选择空闲的时段发送预定申请。

预定信息：

然后可以前往booking review页面查看自己所有未来的预定信息和过去的预定记录，以及每条预定信息对应的预定状态，并且也搭配了筛选和排序的功能。当然用户可以在会议开始之前更改预定的时间。

反馈：

如果用户在使用的过程中遇到了一些问题，那么他们可以通过反馈模块向管理员发送信息反馈。以及跟进反馈的进度。

用户信息：

最后在用户信息界面，可以查看用户的基本信息，以及更改头像，密码或者邮箱。这里我将展示邮箱的更改。需要先获取到新输入的邮箱接受到的验证码确认新邮箱正确之后，才能更改成功。密码的更改逻辑是类似的。

然后让我们登出当前账户，进入管理员账户。

管理端：

房间管理：

首先，管理员的基本功能和用户端是类似的可以添加预定，但是管理员可以额外修改每个房间的详细信息，或者移除某个房间。

预定信息管理：

管理员同样可以查看本账号的预定记录。

用户管理：

可以根据筛选或者排序来快速定位想要查找的用户。以及可以根据预定的数量来主动封禁某些用户。被封禁的用户只能登录系统，但是无法再申请预定。以及管理员也可以直接删除一些用户。被删除的用户则无法登录系统。

预定管理：

管理员可以查看所有人的预定信息，并且选择接受或者拒绝预定，并提供原因。

反馈：

管理员可以看到所有用户的反馈，并且可以对每个反馈进行回复。

个人信息：

这一部分和普通用户功能相同。