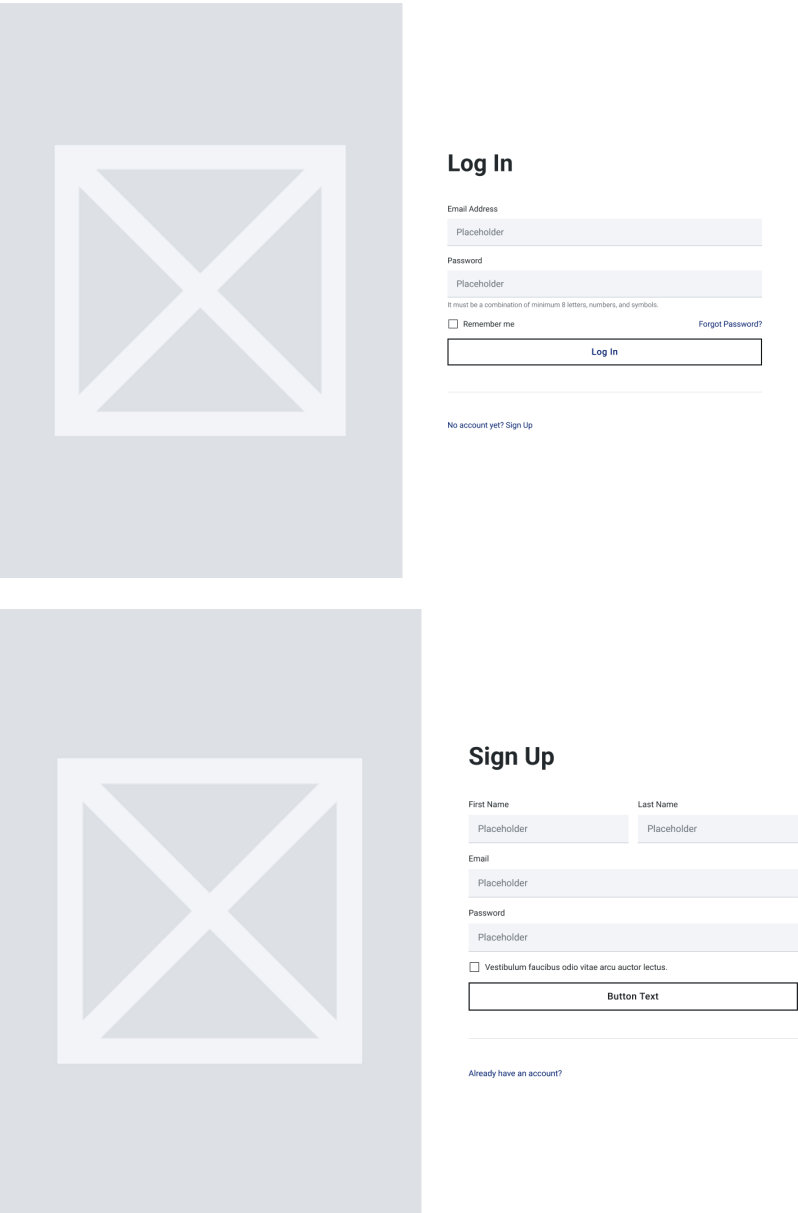


# Lab2

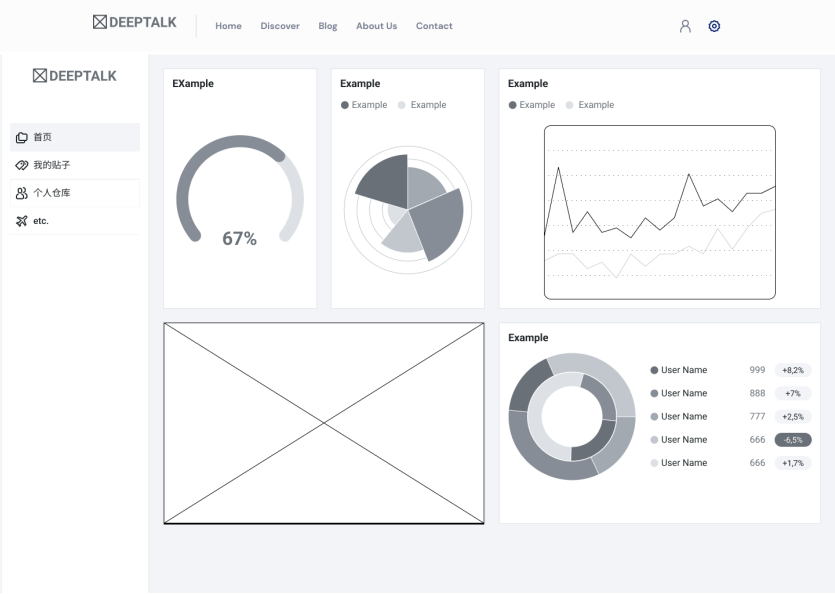
**Q1** Have you designed any paper prototype/wireframe/mock-up/operational prototype for the frontend of your project?

部分草图

## 1. 登陆与注册界面



## 2. 主页及部分功能模板



主页 左边菜单栏 右边显示学习进度等功能

# Nibh in dolor bibendum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sapien, est fella, sagittis viverra nulla mattis scelerisque. Eget cras integer.

Recommended

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Best Seller

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

All Product

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

★★★★★ (20)

Chocolate Cheesecake

\$20.99

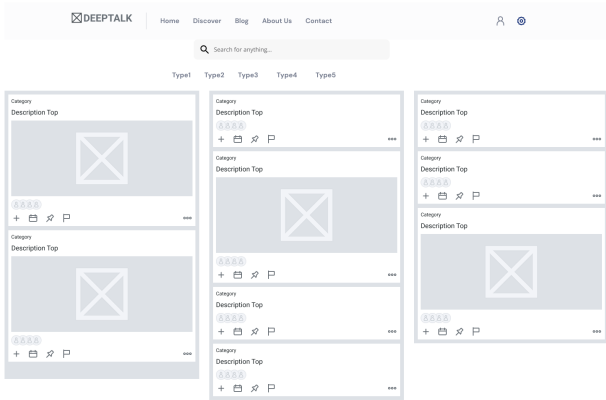
★★★★★ (20)

Chocolate Cheesecake

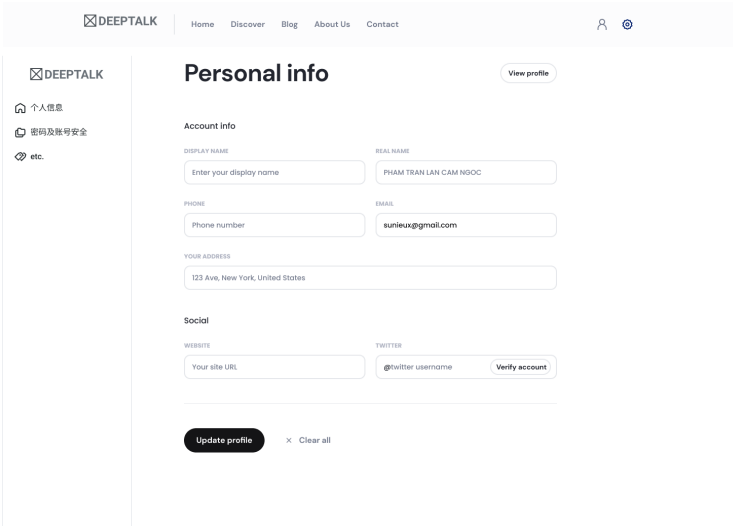
\$20.99

★★★★★ (20)

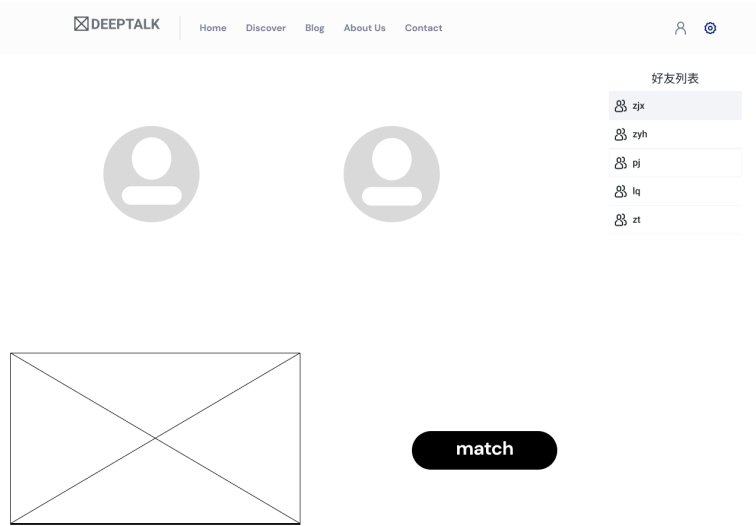
Load more 100+



帖子

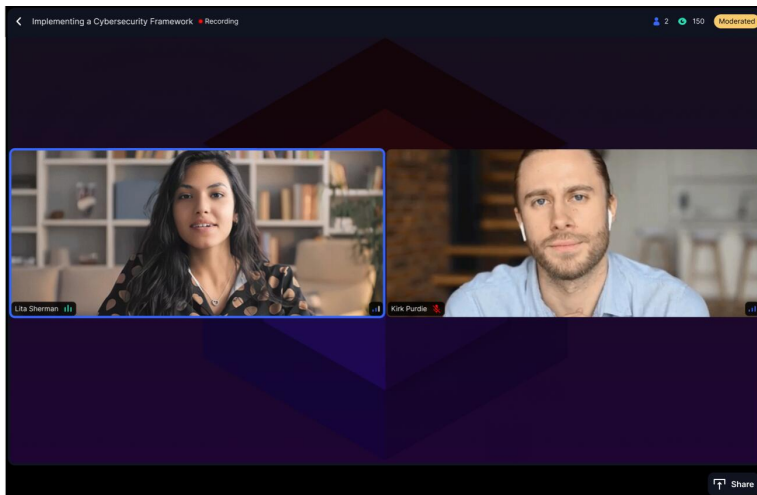
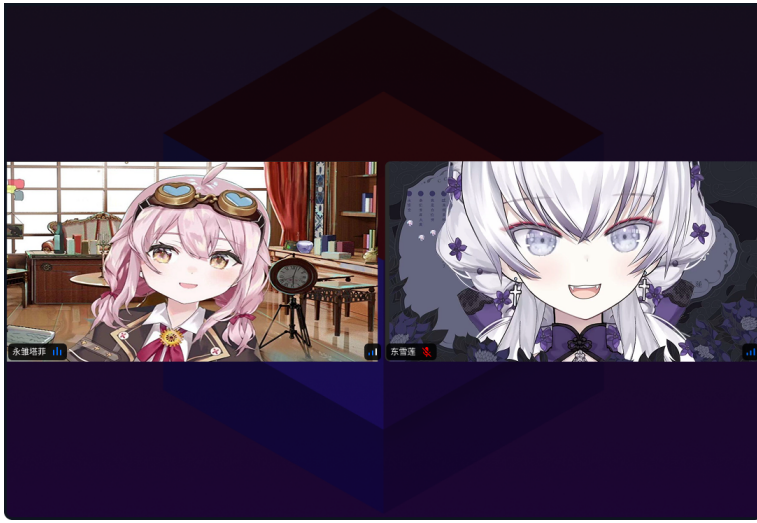


个人信息 到时候一些类似帖子发布之类的功能也会和这个类似

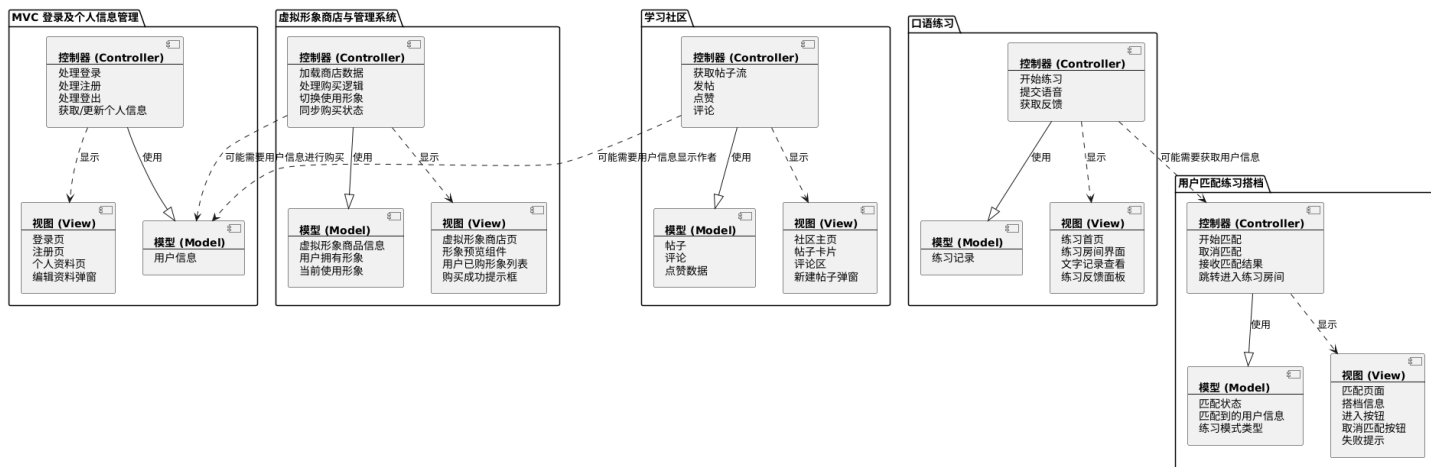


匹配界面 左下角聊天区域 右侧为好友列表

3. 双人交流界面



**Q2** What are the MVC components (if any) of your frontend?



## 1. 登录及个人信息管理

- **模型 (Model)**：用户信息，如用户名、邮箱、头像、登录状态、Token 等。
- **视图 (View)**：登录页、注册页、个人资料页、编辑资料弹窗。
- **控制器 (Controller)**：处理登录、注册、登出、获取/更新个人信息的逻辑。

## 2. 口语练习

- **模型 (Model)**：练习记录（包括模式、搭档类型、音频、反馈等）。
- **视图 (View)**：练习首页、练习房间界面、文字记录查看、练习反馈面板。
- **控制器 (Controller)**：开始练习、提交语音、获取反馈等功能。

## 3. 虚拟形象商店与管理系统

- **模型 (Model)**：存储虚拟形象商品信息（如名称、价格、是否已购买、是否正在使用），以及用户当前拥有和正在使用的形象。
- **视图 (View)**：包括虚拟形象商店页（展示所有可购买形象）、形象预览组件、用户已购形象列表、购买成功提示框等界面。
- **控制器 (Controller)**：负责加载商店数据、处理购买逻辑、切换当前使用的形象、同步用户的购买状态。

## 4. 学习社区

- **模型 (Model)**：帖子、评论、点赞数据。
- **视图 (View)**：社区主页、帖子卡片、评论区、新建帖子的弹窗。
- **控制器 (Controller)**：获取帖子流、发帖、点赞、评论等操作。

## 5. 用户匹配练习搭档

- **模型 (Model)**：保存匹配状态（匹配中、成功、失败）、匹配到的用户信息、练习模式类型。

- **视图 (View)**：匹配页面，显示动画、搭档信息、进入按钮、取消匹配按钮、失败提示。
- **控制器 (Controller)**：控制开始匹配、取消匹配、接收匹配结果、跳转进入练习房间。

## Q3 What is your plan for evaluating/testing the user experience of your project?

### 1. ui相关测试与评估

- A/B测试 (界面与交互优化, 功能有效性验证)**：针对关键模块（如虚拟形象交互、社区动态发布流程）设计不同版本，通过用户行为数据（点击率、停留时长）对比优化界面设计。
- 数据挖掘 (用户行为分析, 学习路径优化)**：分析用户与界面交互日志，获得用户行为记录，分析改进ui设计。绘制用户从注册到核心功能使用的完整路径，识别关键环节（如首次使用AI对练）的流失点，针对性优化引导流程
- 用户直接测试**：让非前端设计组的同学参与软件体验，密切关注用户使用时的受挫点、耗时处、难以完成的任务和用户主动提出问题的部分，并再后续的开发测试中持续发布问卷跟踪用户体验，改进相关设计。

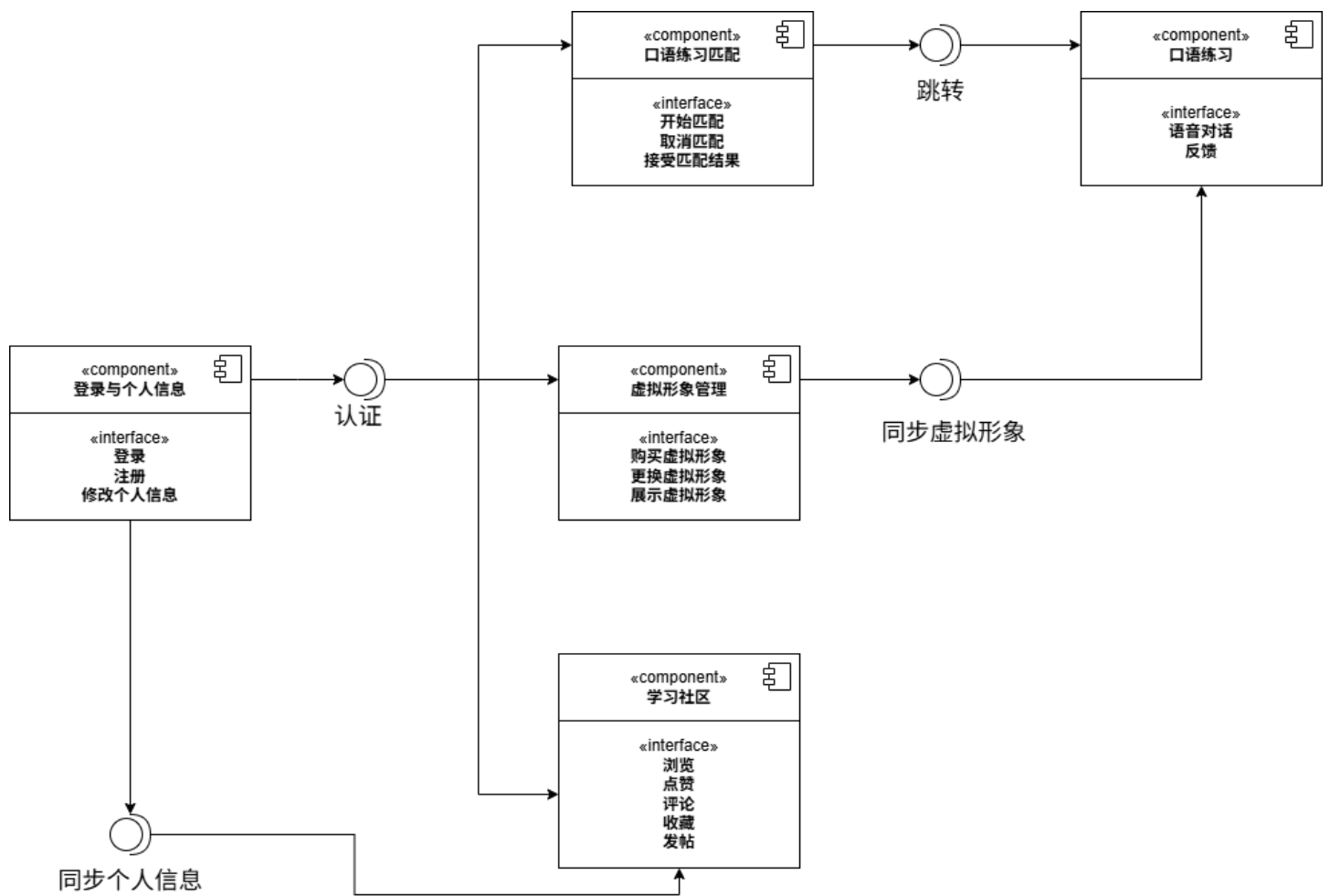
### 2. 性能监测：

- 结合非功能性需求，利用自动化工具（如Jest、Selenium）测试系统响应时间，确保音视频通话延迟 $\leq 300\text{ms}$ 、AI评分反馈生成时间 $\leq 8\text{秒}$ ，并在高并发场景下（如1000人同时在线）进行压力测试。

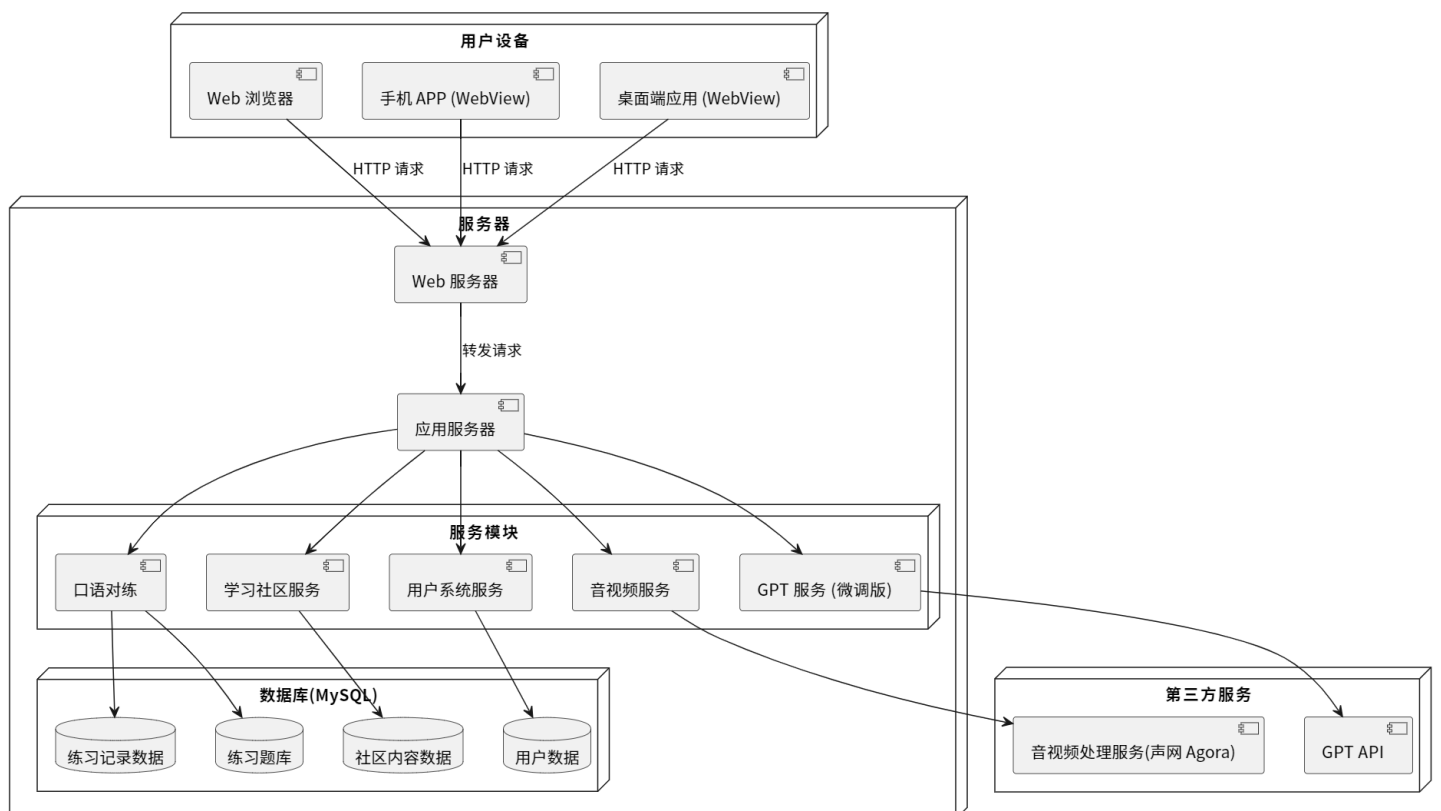
### 3. 认知走查与专家评审：

- 邀请UX专家对关键页面（如评分报告页）进行认知走查，评估信息呈现的清晰度与操作直觉性。

## Q4 What is the component diagram of your project like?

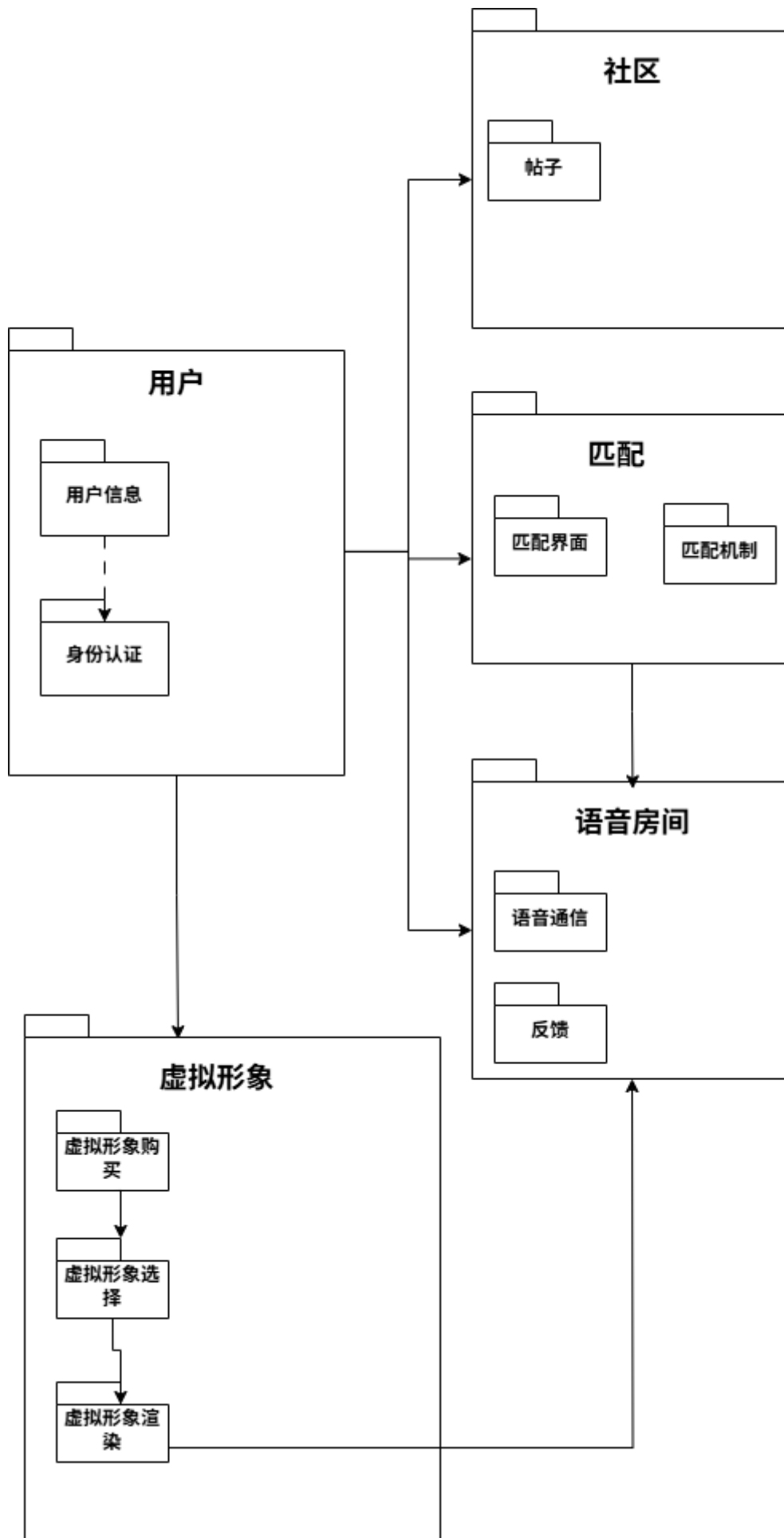


**Q5** What is the deployment diagram of your project like?



**Q6** What is the package diagram of your project like?





**Q7** Any plan of using virtualization technologies?

## 1. Docker 容器化技术

- 开发环境一致性：通过使用Docker，可以创建包含项目所需所有依赖项和环境配置的Docker镜像。开发人员在本地拉取镜像后即可运行与生产环境一致的开发环境，避免了“在我机器上可以运行”的问题。例如，前端开发需要特定版本的依赖库，后端开发需要特定的Java环境和数据库驱动，都可以通过Docker镜像进行封装，确保每个开发人员的工作环境一致。
- 测试环境隔离：在进行功能测试时，可以为每个测试任务创建独立的Docker容器。这样，不同的测试用例可以在隔离的环境中运行，不会相互干扰。例如，测试四六级口语练习功能和测试学习社区的博客发布功能可以分别在不同的容器中进行。如果某个测试环境出现问题，可以快速重建容器而不会影响其他测试工作。
- 持续集成/持续部署（CI/CD）：Docker可以无缝集成到CI/CD流程中。在代码提交后，自动构建Docker镜像并进行测试，确保代码在容器化环境中正常运行。一旦镜像通过测试，可以将其推送到容器仓库，然后在生产环境中快速部署。

## 2. Kubernetes 容器编排技术（可选）

- 多容器应用管理：当我们的项目发展到一定规模，涉及多个容器化应用（如前端服务、后端服务、数据库服务等）时，Kubernetes可以帮助我们进行高效的容器编排和管理。它可以自动部署、扩展和管理容器化应用，确保应用在不同节点上正常运行。在我们的DeepTalk项目中，随着功能模块的不断增加和用户规模的扩大，我们引入了多个容器化应用，Kubernetes在管理这些容器化应用方面发挥了关键作用。它不仅能自动部署和扩展这些服务，还能确保它们稳定运行。
- 自动伸缩：Kubernetes可以根据实际的负载情况自动调整容器的数量，以满足应用的需求。在高并发场景下，通过Kubernetes的自动扩缩容功能，系统能够根据实时负载动态调整各服务的实例数量，从而有效应对流量高峰，提升用户体验。同时，其负载均衡技术可以合理分配流量至各个健康实例，避免单点过载。
- 滚动更新与回滚：在部署新版本的应用时，Kubernetes支持滚动更新策略，可以逐步替换旧版本的容器实例，避免整个系统同时更新导致的服务中断。如果新版本出现问题，还可以快速回滚到之前的稳定版本。这在测试新功能或修复bug时非常有用，可以降低更新带来的风险，确保系统的稳定运行。

## Q8 Any possible security risk in your project? How will you minimize them?

### 1. 隐私与数据安全风险：

- **数据加密**：用户录音文件采用AES-256加密存储，数据库敏感字段（如密码）使用bcrypt哈希加密。

- **合规管理**：明确数据保留期限与删除机制，遵循《数据安全法》等法规，获取用户明确授权后使用数据。
- **访问控制**：基于RBAC（角色权限控制）限制数据访问，仅授权人员可操作敏感数据（如录音文件下载）。
- **定期审计**：利用阿里云安全中心扫描系统漏洞。
- **应对SQL注入**：
  - 1、参数化查询（Prepared Statements）：避免直接拼接SQL语句，使用参数化查询（如Node.js的 `pg` 模块），确保用户输入被作为数据而非代码解析。实现方式：后端（Hibernate/JPA）使用ORM（Hibernate/JPA）自动生成参数化查询；手动编写SQL时，严格使用占位符（如 `%s` 或 `?`）传递参数。
  - 2、输入验证与白名单过滤：对用户输入的数据进行格式校验（如邮箱、手机号的正则匹配），拒绝非法字符（如 `'`、`;`、`--`）。
  - 3、最小权限原则：数据库账号仅分配必要权限（如禁止 `DROP TABLE`、`FILE` 权限）。
- **应对XSS攻击**：
  - 1、输出转义（Output Encoding）：在数据输出到HTML、JavaScript或URL前，对特殊字符（如 `<`，`>`，`&`，`"`）进行转义。
  - 2、内容安全策略（CSP）：通过HTTP头限制脚本来源，阻止内联脚本和外部恶意资源加载。

## 2. 内容审核风险：

- **AI预审+人工复核**：对用户生成内容（如社区动态）进行敏感词过滤（正则表达式），并通过人工团队定期抽查，并对于不良行为较多的用户予以封禁等处理。
- **AI输出监控**：对GPT生成的对话内容设置硬编码黑名单，避免不当建议。

## 3. 技术实现风险（可能采用其他相关易于部署的技术）：

- **音视频成熟技术**：
  - i. **腾讯云 TRTC（Tencent Real-Time Communication）**
    - **免费额度**：
      - 每月赠送 **10,000 分钟** 音视频时长（新用户首月额外赠送 30,000 分钟）。
      - 支持实时音视频、屏幕共享、低延迟互动（<300ms）。
    - **教育支持**：
      - 腾讯云提供 **学生开发者计划**（需认证），可申请云资源代金券，覆盖 TRTC 和云服务器费用。
    - **特点**：

- 集成简单，提供 Android/iOS/Web SDK。
- 支持虚拟背景、美颜、AI 降噪等增强功能。

## ii. Agora 声网（开发者免费版）

- **免费额度：**
  - 每月 **10,000 分钟** 音视频时长（需注册认证）。
  - 支持实时音视频、录制、跨平台 SDK（Web/移动端/桌面）。
- **学生友好性：**
  - 提供详细文档和社区支持，适合学习实时音视频开发。
  - 可申请教育合作计划（需学校或导师推荐）。
- **特点：**
  - 全球节点覆盖，抗弱网能力强（支持 QoS 优化）。
  - 可直接与 GPT 等 AI 服务集成（如语音评分）。

## iii. Daily.co（免费层）

- **免费额度：**
  - 每月 **2,000 分钟** 音视频时长（支持最多 50 人同时在线）。
  - 提供预构建 UI 组件（如视频会议界面）。
- **特点：**
  - 无需服务器搭建，10 分钟快速集成。
  - 支持录制、虚拟背景、屏幕共享。

## 4. 运营风险：

- **纯AI模式备用：**若真人匹配质量不稳定，可切换至纯AI陪练模式，保障用户体验。
- **数据备份：**每日增量备份用户数据至阿里云OSS，确保灾难场景下快速恢复。

## Q9 What will the peak demand/capacity of your product be in the future?

在极端条件下，可能会有以下高资源需求功能同时运行：

### 1. 口语练习：实时对话/形象系统

这一系统要求**低延迟**的实时对话功能，同时需要**充足的网络带宽**用于传输虚拟形象数据/视频数据。如果采用虚拟形象，则对客户端的视频处理能力提出挑战（需要通过视觉方法识别面部特

征)。

## 2. 数据库系统

口语练习、社区分享等功能涉及到对服务端数据库的访问，在进行其他任务的同时，访问数据库会对服务器的IO性能和CPU性能提出考验。

## 3. AI系统

引入AI评分和AI对话时，对AI的响应速度有一定要求，同时可能消耗大量的token。

同时，多人同时在线的平台本身也对服务端的并发性能提出挑战。

**Q10** Based on the potential demand/capacity, any performance bottleneck in your project? How will you improve them?

### 1. 网络带宽消耗量大

改进方法：

- a. 提高服务器本身的吞吐能力
- b. 采用压缩技术，对音视频流进行压缩，减小单任务的带宽占用
- c. 限制同时进行的任务（练习、观战）数，采用排队机制，同时可以引入付费高级用户，提供排队优先或专属通道功能，而高级用户的收入可以用于补贴其造成的网络/服务器开销
- d. 由于虚拟形象只需发送分析得到的特征信息，优先鼓励用户使用虚拟形象功能，减少带宽占用

### 2. 延迟要求较高

改进方法：

- a. 采用压缩技术，减少实时音频/信息/视频流的数据大小，降低传输的时间开销
- b. 鼓励使用数据量更小的虚拟形象功能
- c. 若资源有限，则优先向核心功能（口语对话）倾斜

### 3. AI的响应速度

改进方法：

- a. 采用性能更好的AI计算服务提供商
- b. 优化AI模型自身的性能，如使用蒸馏微调的模型等.

### 4. 客户端的处理能力

改进方法：

- a. 减少客户端不必要的动画效果、特效等
- b. 对于虚拟形象的采集，若客户端不能提供充足的性能，可以减小采样率，降低运算开支

## 5. 服务器的并发/CPU/IO性能

改进方法：

- a. 提高服务器本身的性能
- b. 优化算法，降低服务器负载
- c. 采用异步处理机制，将一些非实时的任务放到后台异步处理，避免阻塞主线程