

TsingOverflow

面向清华大学在校生的
线上答疑流程优化设计与实现

课程教师：
刘璘

小组成员：
黎叙锐（化工系）
黄砚星（数学系）
段成业（化工系）
苏宇然（美术学院）



清答疑小程序现行问题多：

- 图片加载异常
- 消息排序错乱 (Client时间做后端记录恶果)
- Mac端内存泄漏
- 附件不能上传
- 不能渲染公式
- 开发者已经毕业 (处于无人维护状态)
-

开发新线上答疑坊呼声大：

- 队委：要做了..... (上学期言)

为什么要做这个项目？

- 因为山就在那里！



图1. 清答疑头像无法加载 (BUG)



图2. 开发新线上答疑坊呼声大

交付目标

我们要做什么？

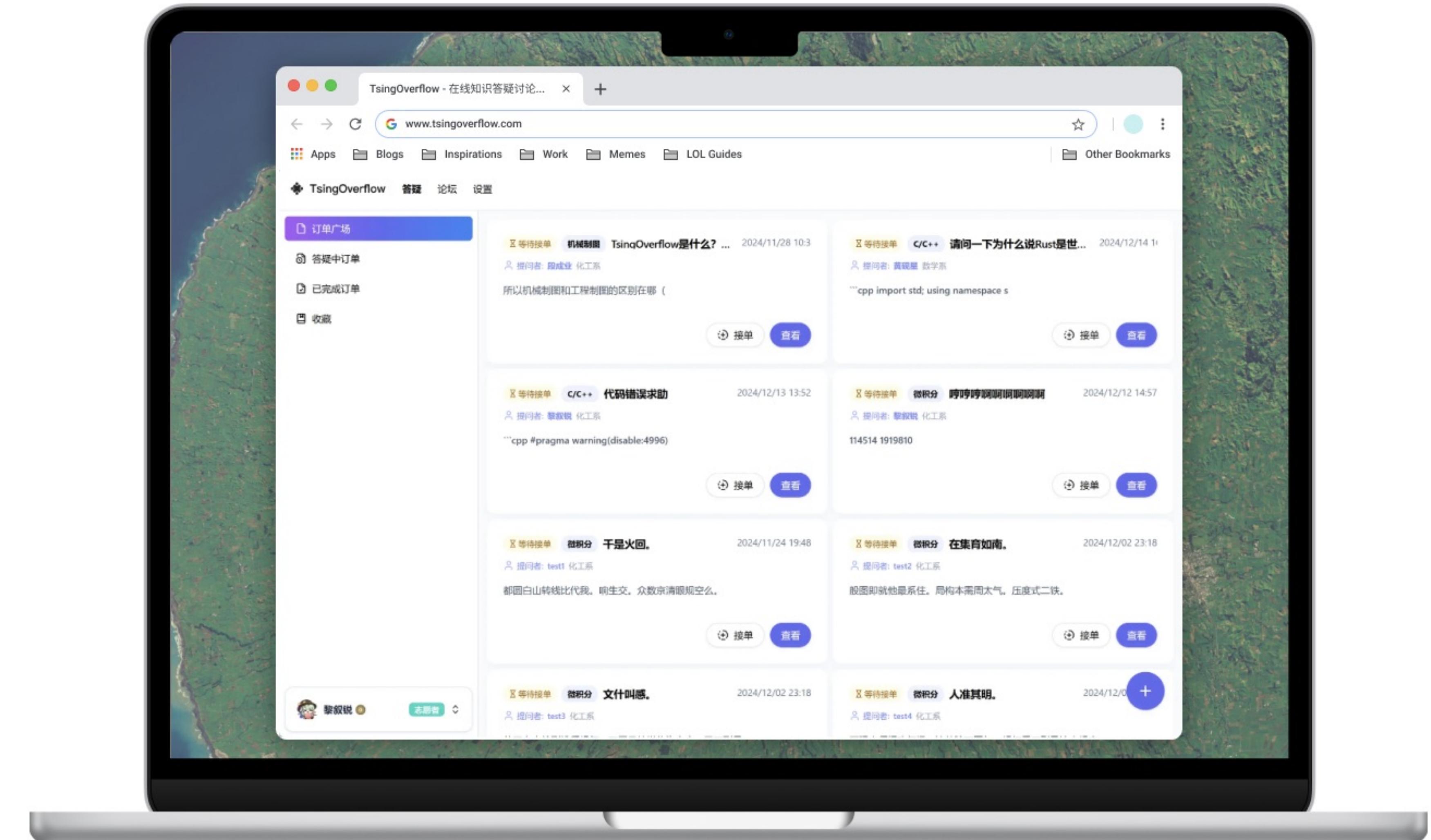
- 做一个新答疑坊！

我们的甲方是谁？

- 答疑坊官方
- 或“五系联合答疑坊”

要做成什么样子？

- Markdown语法的支持
- LaTeX的支持
- 更优秀的图片管理
- 代码块的嵌入和高亮
- 较为完整的交流平台



前端——样式规范：基于tailwind css框架

◆ TsingOverflow

• 字体字号

文字示例

字号

行高

tailwind类

使用场景

清答疑

20px

28px

text-xl

页面标题，不可大面积使用

清答疑

16px

24px

text-base

分标题、文本文字

清答疑

14px

20px

text-sm

辅助性说明文字

清答疑

12px

16px

text-xs

标签或角标文字

• 色彩系统

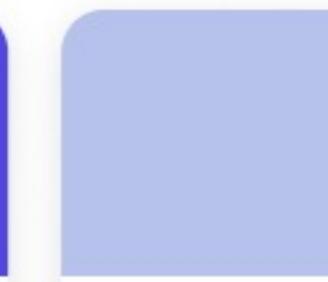
颜色



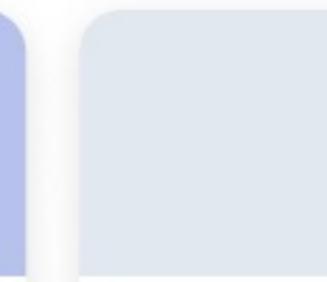
主品牌色



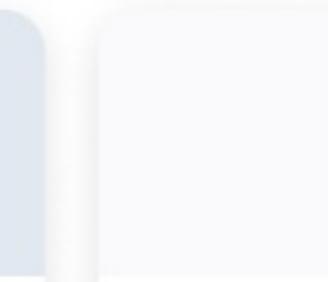
强调色



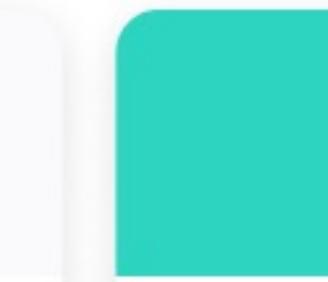
置灰色



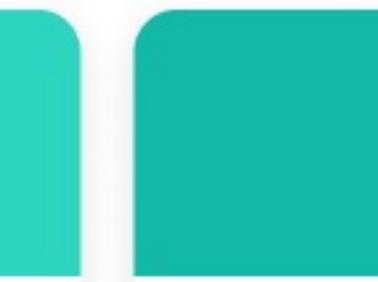
浅边框色



背景色



志愿者色



志愿者强调色

tailwind类

indigo-500

indigo-600

-

slate-200

-

teal-400

teal-500

色值

#5E5CE6

#4F46E5

#B6C1EB

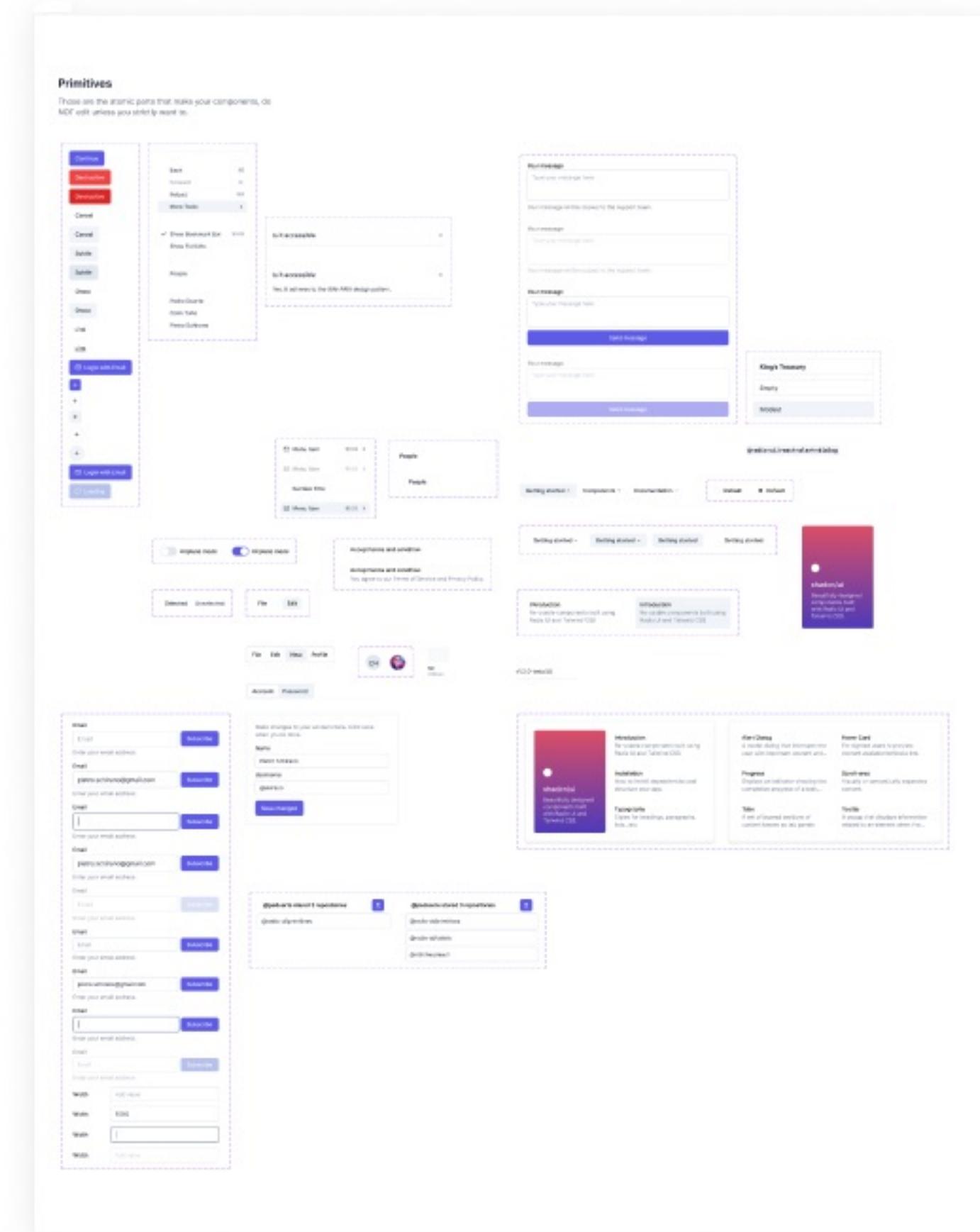
#E2E8F0

#FAFAFC

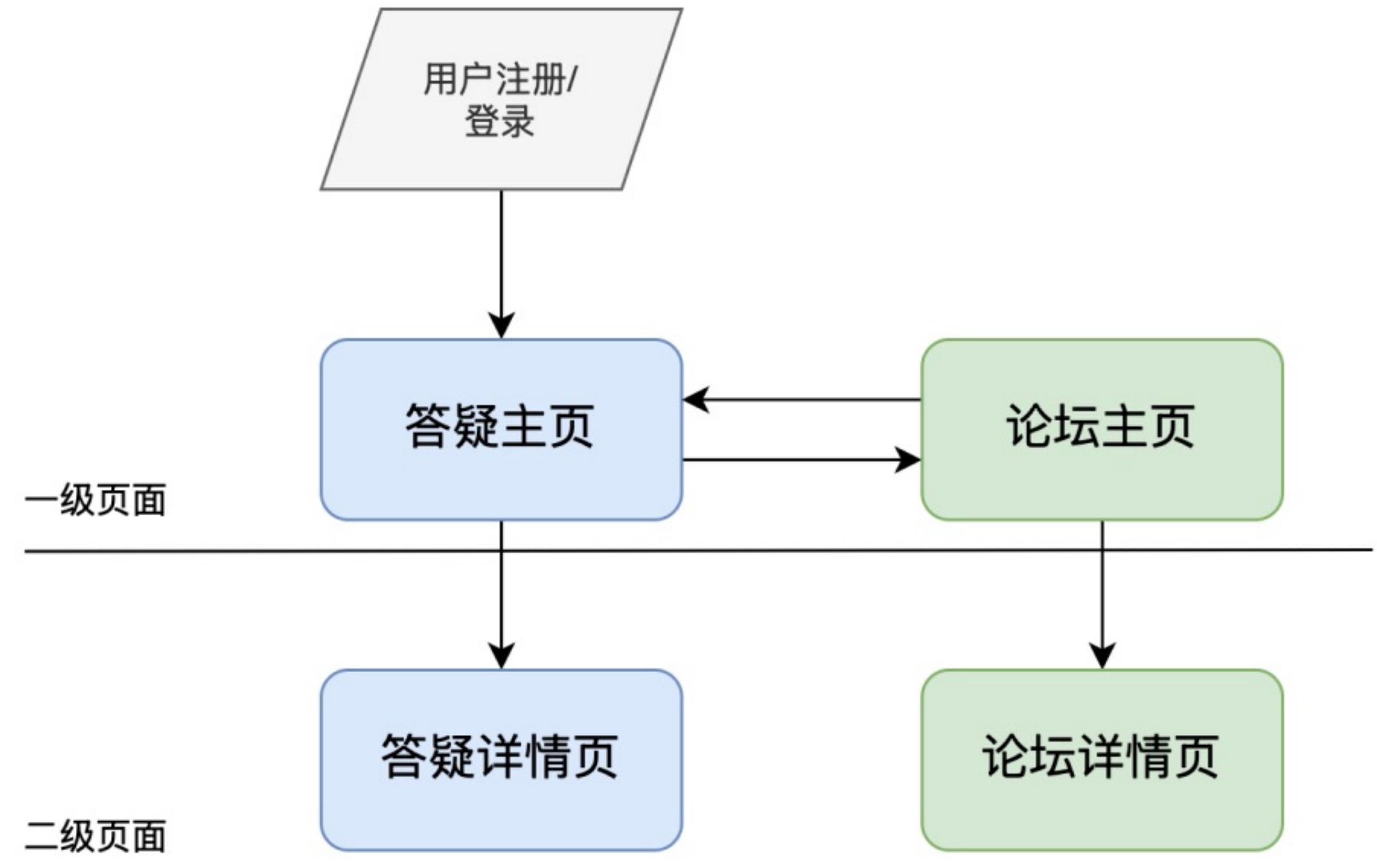
#2DD4BF

#14B8A6

• 组件规范（略）



- 我们前端页面整体采用**三段式框架**, 清晰简洁而不失实用性。
- **用户登录后会首先进入答疑主页**, 同时在主页当中点击选项卡即可进入答疑或论坛的**详情页**。
- 程序前端各个页面之间相互独立, 仅通过local storage当中的**jwt token**共享用户的登录数据, 并各自配有独立的链接, 通过url参数来确定位置, 方便分享与访问
 - 例如: <http://124.221.89.248/dialog?pid=23>

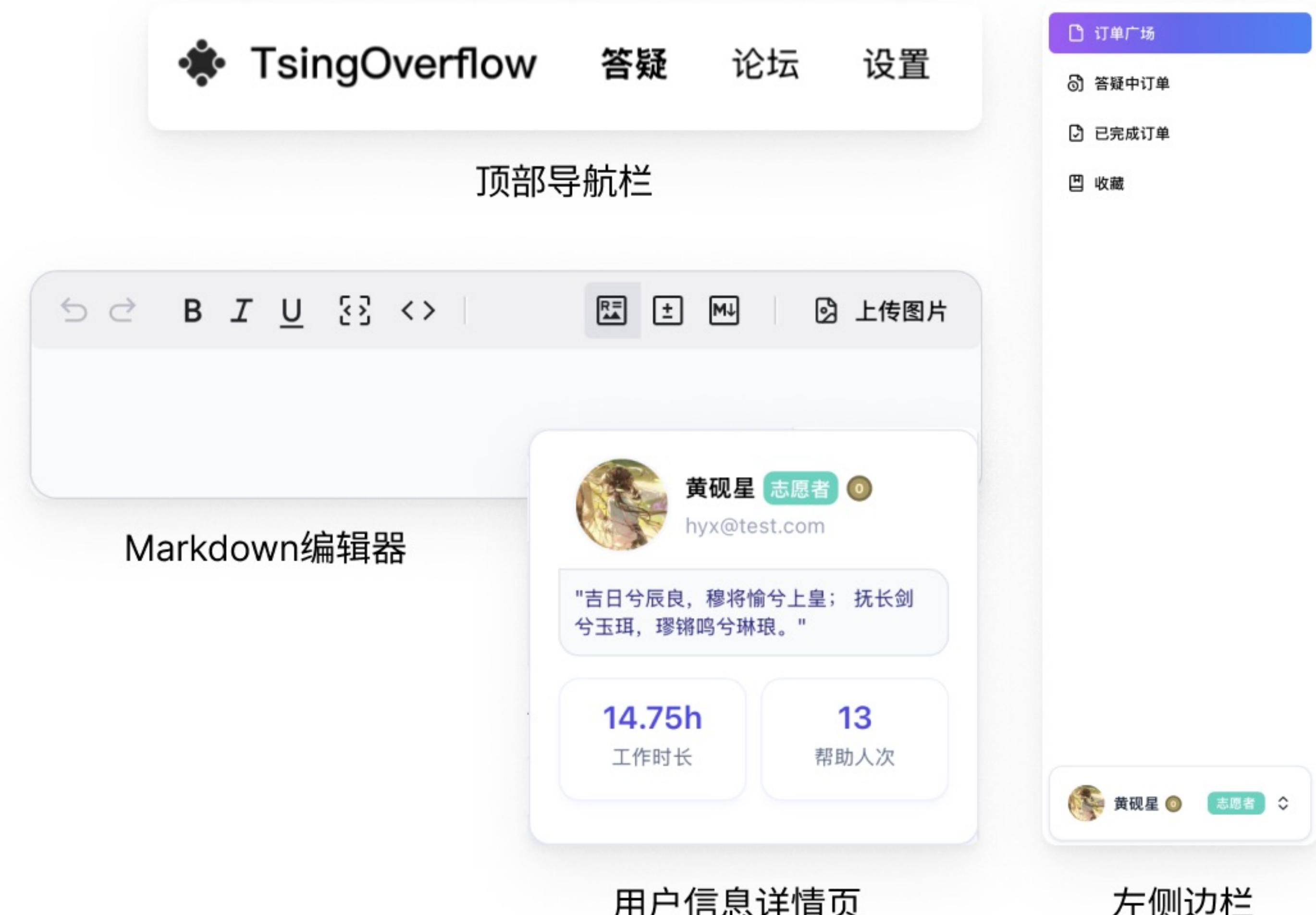


前端页面整体设计图

前端——组件设计

◆ TsingOverflow

- 我们秉持组件化与代码复用的原则，各个页面都拆分为数个组件并高度复用。
- 组件化的页面设计有如下多项优势：
 - 易于修改维护
 - 更高的开发效率
 - 统一的页面风格
 -
- 右图是我们所使用的部分组件示例。



注册/登录页及新用户初始化

◆ TsingOverflow

• 用户注册页



• 登录页-一般用户



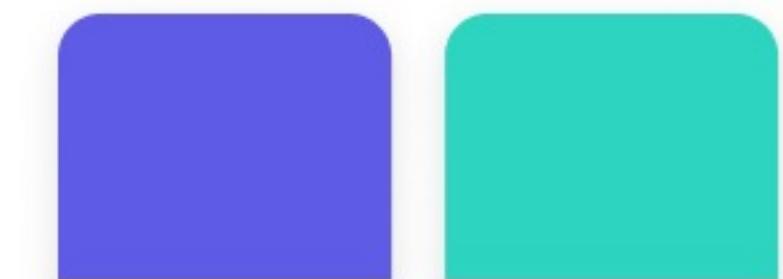
• 登录页-志愿者用户



切换登录方式组件

一般登录 志愿者登录

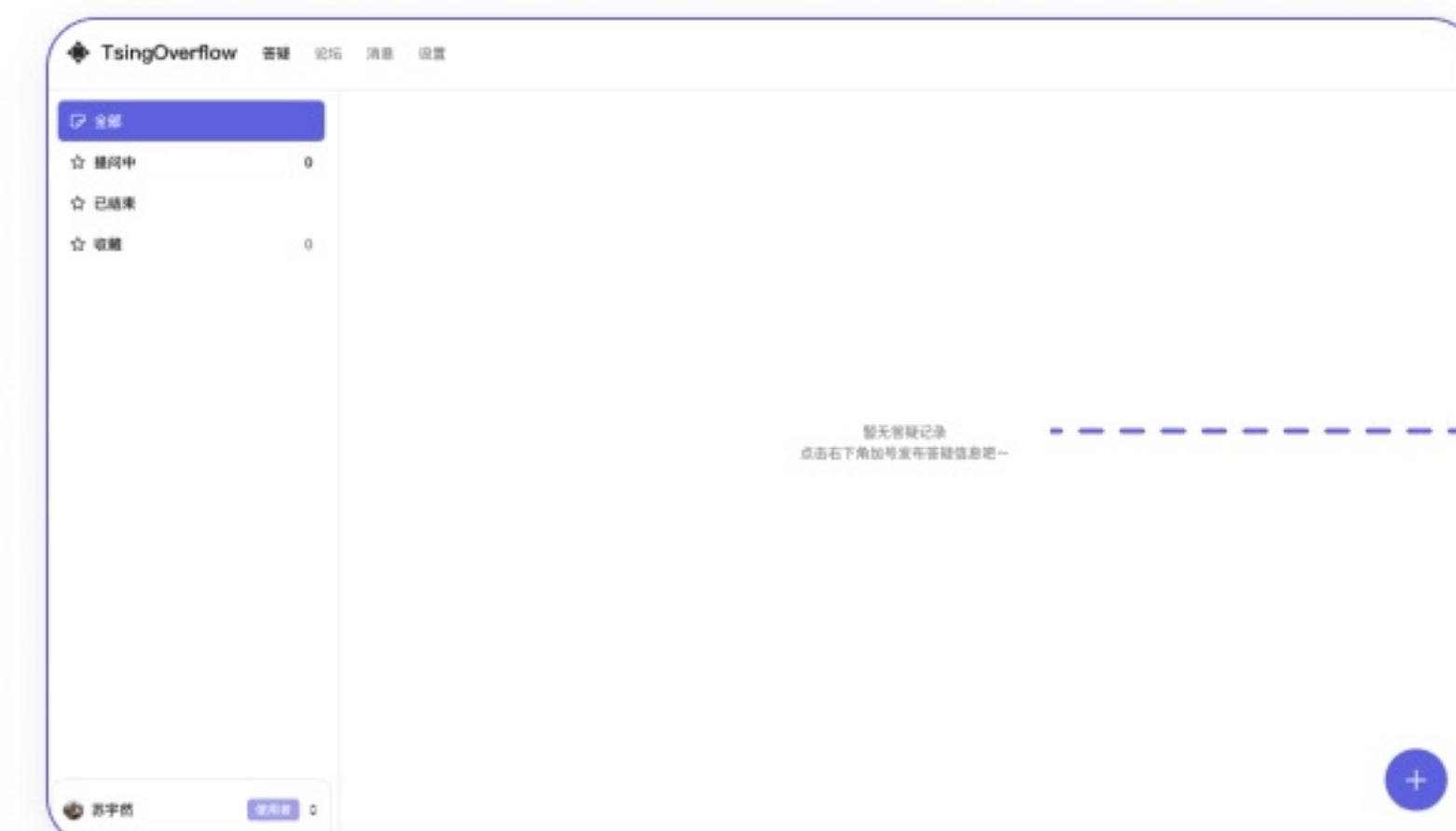
志愿者主题色，与一般模式区分



• 新用户关键词设置



• 新用户初始页面 (答疑)



初次注册文字教学

暂无答疑记录

点击右下角加号发布答疑信息吧~

旧版小程序

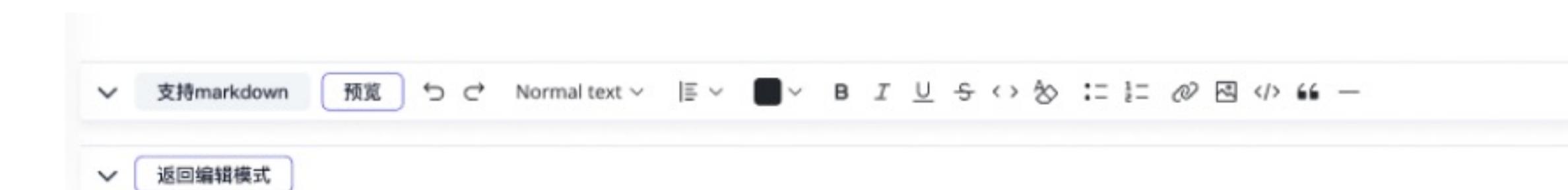
- 图片加载异常
- 消息排序错乱
- 附件不能上传
- 不能渲染公式.....



Ours 全答疑流程的可视化小组件



markdown输入框支持图片、公式渲染



原型设计——答疑页面（一般用户视角）

◆ TsingOverflow

1. 答疑主页

The screenshot shows the TsingOverflow homepage with a sidebar on the left containing '我的提问' (My Questions), '提问中' (In Progress), and '已结束' (Completed). Below this is a search bar and a '发布提问' (Post Question) button. The main area displays a grid of question cards. Each card includes a title, a snippet of the question text, a rating (e.g., 5.0, 15分钟), and a status (e.g., 已完成). A dashed blue arrow labeled '答疑卡组件' (Answer Card Component) points from the bottom right of the card grid towards the '创建提问' (Create Question) section of the next page.

2. 新建答疑页

This screenshot shows the 'Create Question' page. It features a sidebar with '我的提问' (My Questions), '提问中' (In Progress), and '已结束' (Completed). The main form has fields for '标题' (Title) and '内容' (Content). The 'Content' field contains a text area with placeholder text: '求问第四题解题思路，谢谢！'. To the right of the content area is a sidebar for '评分选择' (Scoring Selection) with '一级标签' (Primary Label) and '二级标签' (Secondary Label) dropdowns, both set to '离散数学'. Below this is a '提交' (Submit) button. A dashed blue arrow points from the '我的提问' section of the previous page to this screen.

评分成功后在侧边栏显示工时和评分

3. 答疑详情页1 等待接单

This screenshot shows the 'Waiting for Acceptance' detail page. It includes a sidebar with '我的提问' (My Questions), '提问中' (In Progress), and '已结束' (Completed). The main content area shows a question card with a title, snippet, rating, and status. Below the card is a message: '已经推送14个答疑志愿者正在等待接单...' (14 answer volunteers have been pushed and are waiting for acceptance). A dashed blue arrow points from the '我的提问' section of the previous page to this screen.

4. 答疑详情页2 答疑中

This screenshot shows the 'Answering' detail page. It has a sidebar with '我的提问' (My Questions), '提问中' (In Progress), and '已结束' (Completed). The main content area shows a question card with a title, snippet, rating, and status. Below the card is a message: '已经被志愿者 黎叙锐 接单！正在等待回复...' (Accepted by volunteer Li Xuyue! Waiting for response...). A dashed blue arrow points from the '我的提问' section of the previous page to this screen.

5. 答疑详情页3 答疑结束

This screenshot shows the 'Answered' detail page. It has a sidebar with '我的提问' (My Questions), '提问中' (In Progress), and '已结束' (Completed). The main content area shows a question card with a title, snippet, rating, and status. Below the card is a message: '感谢您的帮助!' (Thank you for your help!). At the bottom, there is a '评分' (Rating) section with a 5-star rating and a '提交' (Submit) button. A dashed blue arrow points from the '我的提问' section of the previous page to this screen.

评分成功

感谢评分~

确定

答疑页面 (志愿者用户视角)

TsingOverflow

志愿者个人信息组件

1. 答疑接单主页

The screenshot shows a list of pending requests on the 'Answering Requests' page. Each request card includes the user's name, a summary of the question, a due date (e.g., 22 月工时), and a 'Claim Request' button.

接单二次确认提示框
账户生命中仅出现一次

• 答疑预览页

The screenshot shows a preview of a request from '黎毅锐'. It includes the question text and a confirmation dialog box asking '确定要接单吗?' (Do you want to claim the request?).

2. 答疑详情页1 编辑答疑

The screenshot shows the 'Edit Answer' page for a request from '黎毅锐'. It includes a rich text editor with a 'Markdown' button, a preview window, and a 'Return to Editing Mode' button.

输入框富文本编辑器，支持md

3. 答疑详情页2 等待提问人回复

The screenshot shows the 'Waiting for Response' page for a request from '黎毅锐'. It includes the question text and a message: '正在等待提问人 苏宇然 回复' (Waiting for the asker Su Yuzhan to reply).

等待提问人回复 提问人已采纳

4. 答疑详情页3 答疑结束-提问人已评分

The screenshot shows the 'Answering Ended - Questioner Scored' page for a request from '黎毅锐'. It includes the question text, a 'Score' section with a rating of '+60min', and a 'Publish Forum' button.

旧版小程序（知识库）

埋藏位置奇怪（在“通知中心”里？）
只能由志愿者上传答疑记录



Ours

单独开设为一级页面，与通知和消息分开



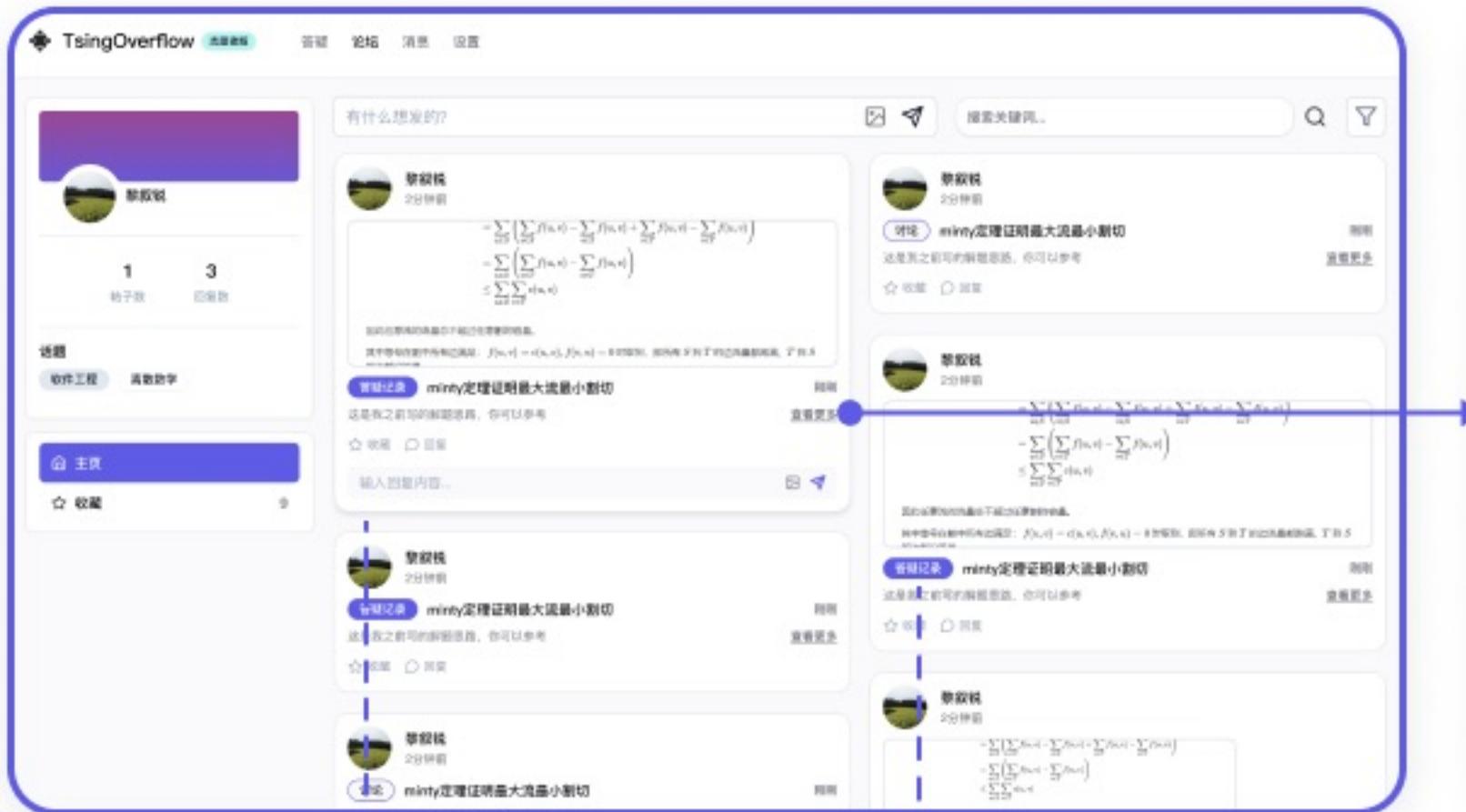
除了上传答疑记录，也支持在论坛直接发布帖子、回复



原型设计——论坛页面

◆ TsingOverflow

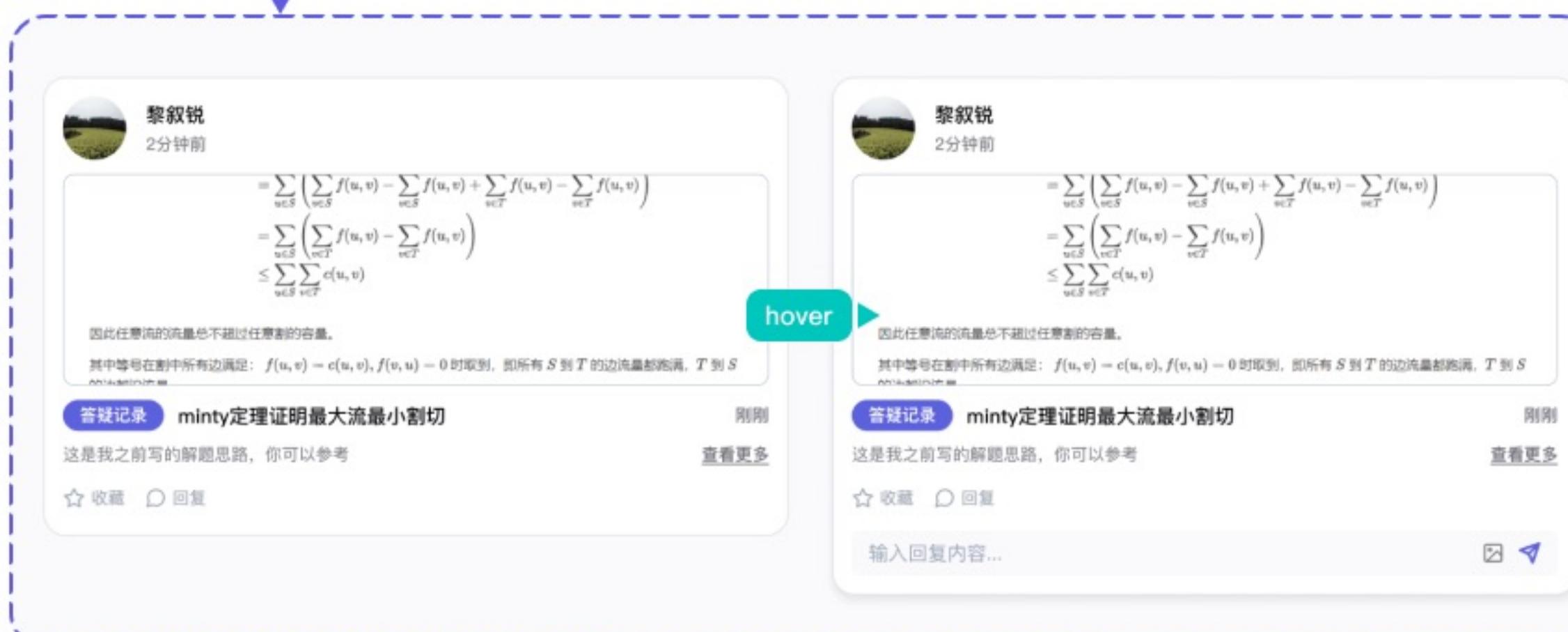
1. 论坛主页



2. 帖子详情页



帖子组件



→ 帖子来源分类标签组件

- 答疑记录为从答疑详情页中直接发布到论坛的，已经结束的答疑的文字记录，记录中提问者匿名。
- 讨论为直接在论坛中发布的帖子。

答疑记录

讨论

3. 帖子详情页 - 回复



回复帖子



页面响应

该截图展示了桌面端的订单广场界面。顶部导航栏包含“TsingOverflow”、“答疑”、“论坛”和“设置”选项。左侧侧边栏有“订单广场”（紫色背景）、“答疑中订单”、“已完成订单”和“收藏”按钮。右侧主体区域显示了三个帖子列表：

- 帖子4：状态“等待接单”，标签“工程制图”，提问者“苏宇然 美术学院”，时间“2024/11/24 16:44”。内容：“这是测试内容4”。操作按钮：“接单”、“查看”。
- 帖子5：状态“等待接单”，标签“R语言”，提问者“苏宇然 美术学院”，时间“2024/11/24 16:44”。内容：“这是测试内容5”。操作按钮：“接单”、“查看”。
- 帖子6：状态“等待接单”，标签“物理化学”，提问者“段成业 化工系”，时间“2024/11/28 11:02”。内容：“记成 哟就聊列出框”。操作按钮：“接单”、“查看”。

底部状态栏显示用户“黄砚星”（志愿者身份）以及一个蓝色的“+”号按钮。

桌面端主页截图

该截图展示了移动端的订单广场界面，整体布局与桌面端相似。

- 帖子4：状态“等待接单”，标签“工程制图”，提问者“苏宇然 美术学院”，时间“2024/11/24 16:44”。内容：“这是测试内容4”。操作按钮：“接单”、“查看”。
- 帖子5：状态“等待接单”，标签“R语言”，提问者“苏宇然 美术学院”，时间“2024/11/24 16:44”。内容：“这是测试内容5”。操作按钮：“接单”、“查看”。
- 帖子6：状态“等待接单”，标签“物理化学”，提问者“段成业 化工系”，时间“2024/11/28 11:02”。内容：“记成 哟就聊列出框”。操作按钮：“接单”、“查看”。

底部状态栏显示用户“黄砚星”（志愿者身份）以及一个蓝色的“+”号按钮。

手机端主页截图

前端重点——页面响应

◆ TsingOverflow

The screenshot shows a mobile application interface. At the top, there is a navigation bar with icons for TsingOverflow, 答疑 (Answer), 论坛 (Forum), and 设置 (Settings). Below the navigation bar, the main content area has a header with a back arrow, the category "c/c++", the title "请问一下为什么说Rust是世界上最好的语言?", and a message "1条消息记录 等待接单中".

The main content area contains a code block in C/C++:

```
import std;
using namespace std;

int main() {
    std::println("你们说的rust是这个语言嘛？");
    return 0;
}
```

Below the code block, there is a reply from a user named 黎叙锐, posted on 2024/12/18 14:51. The reply title is "The derivation of Boltzmann equation" and contains a bullet point: "玻尔兹曼分布的推导: $\frac{N_i}{N} = \frac{\alpha e^{-\beta E_i}}{\sum e^{-\beta E_i}} = \frac{e^{-\beta E_i}}{\sum e^{-\beta E_i}}$ ".

On the right side of the screen, there is a large block of code representing the implementation of a Markdown component. The code uses JSX syntax and includes imports for `react`, `reactMarkdown`, `remark`, `rehype`, and `highlight.js`. It defines a `Markdown` component that handles class names, remark and rehype plugins, components, and a `code` prop for syntax highlighting.

```
<Markdown
  className="prose leading-4 text-gray-800 pb-2 px-3 max-w-[60vw]"
  children={reply.ct}
  remarkPlugins={[remarkGfm, remarkMath]}
  rehypePlugins={[rehypeKatex]}
  components={{
    code(props) {
      const {children, className, node, ...rest} = props
      const match = /language-(\w+)/.exec(className || '')
      return match ? (
        <SyntaxHighlighter
          {...rest}
          PreTag="div"
          children={String(children).replace(/\n$/,'')}
          language={match[1]}
          style={nightOwl}
          customStyle={{backgroundColor: 'transparent', padding: 0, margin: 0}}
        />
      ) : (
        <code {...rest} className={className}>
          {children}
        </code>
      )
    }
  }}>
```

Markdown格式、代码高亮与LaTeX公式渲染

后端——Model约束

- 我们的数据库结构较为复杂
 - 先画图！有的放矢！
 - 图中每个ForeignKey连接的类实际是关联它的id成员。
 - Django的ORM会自动处理。
 - ManyToManyField是多对多关系。
 - Django的ORM实际上开了一张新表存索引。

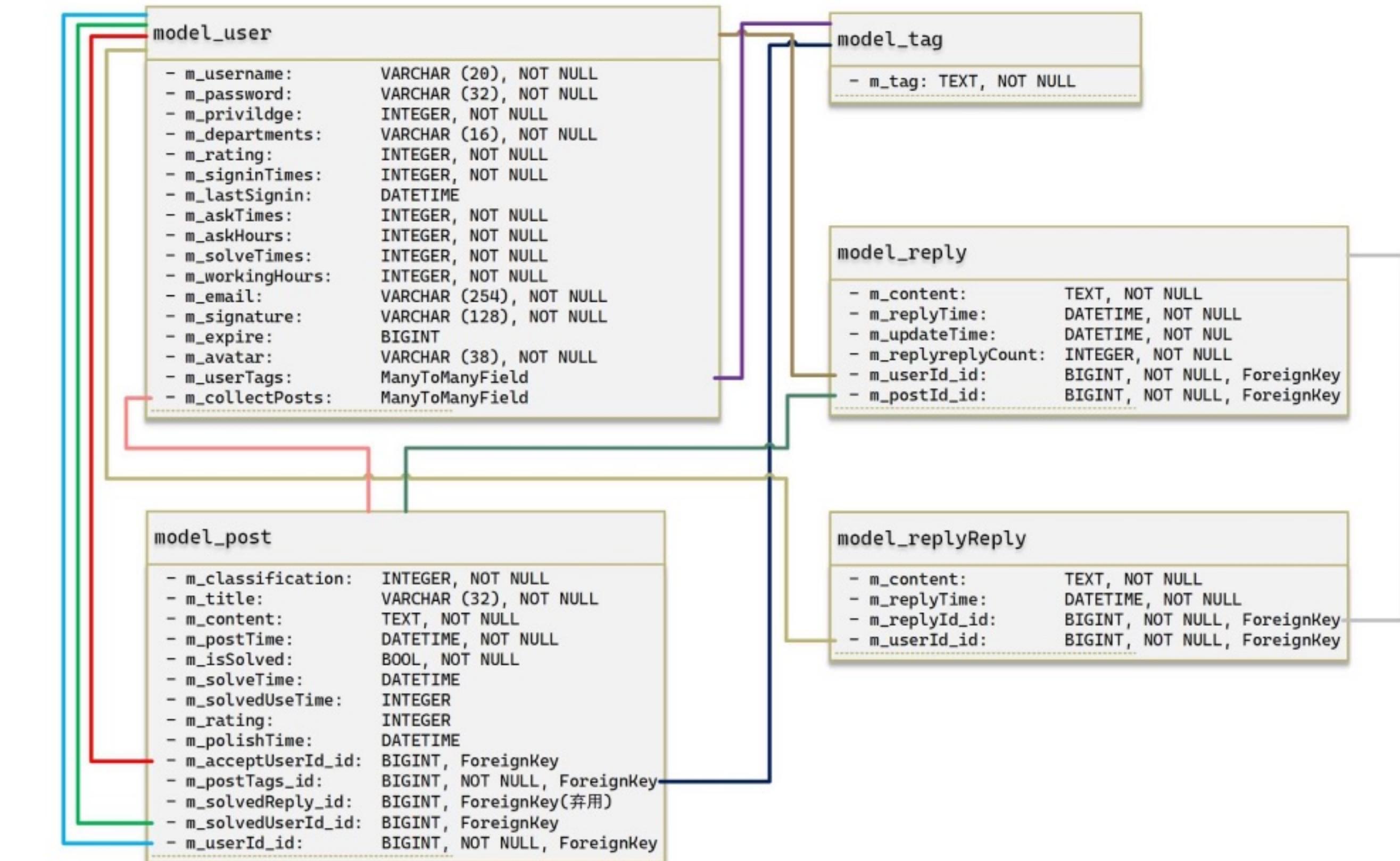
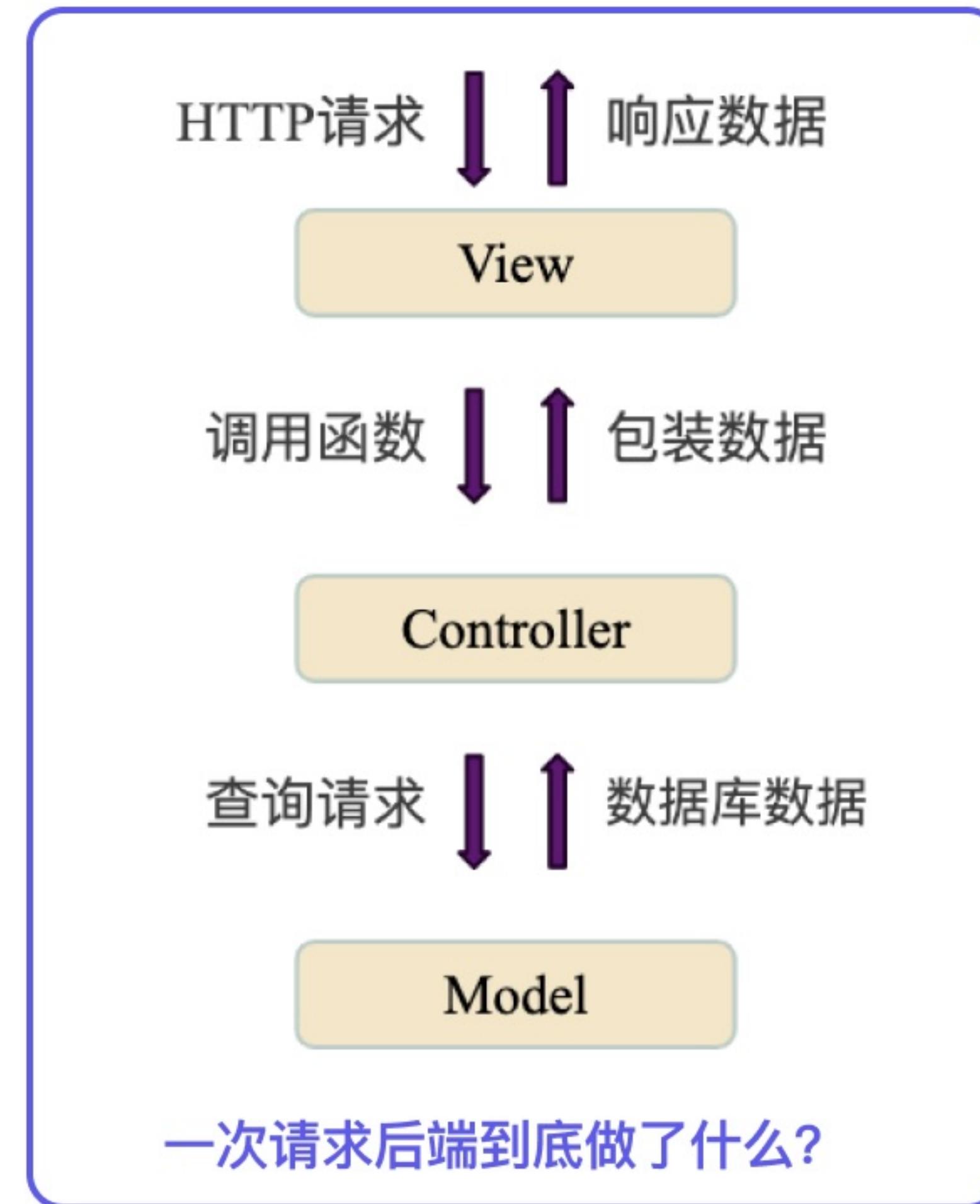


图. 数据库结构示例图

- 我们采用类Model–View–Controller (MVC) 的结构。
 - 借鉴思想，但是和标准的MVC不完全相同。
- Model
 - 只允许定义数据库的ORM模型，不实现其他任何功能。
- Controller
 - 只允许在Model的基础上实现程序行为，只负责业务逻辑的处理和数据的流转。
- View:
 - 负责实现后端API的数据规范处理，一般只允许使用Controller的接口。
 - 做好“包装”，让每个函数尽量只做好一件事。



- 回顾计算机中Cache的定位?
 - 在多维度中寻找平衡。
- 我们使用Redis缓存的位置:
 - 校验JWT和获取用户信息时。
- 我们使用的原则:
 - 不是数据库的上位或下位替代。
 - 不滥用。
 - 考虑开发难度的平衡。

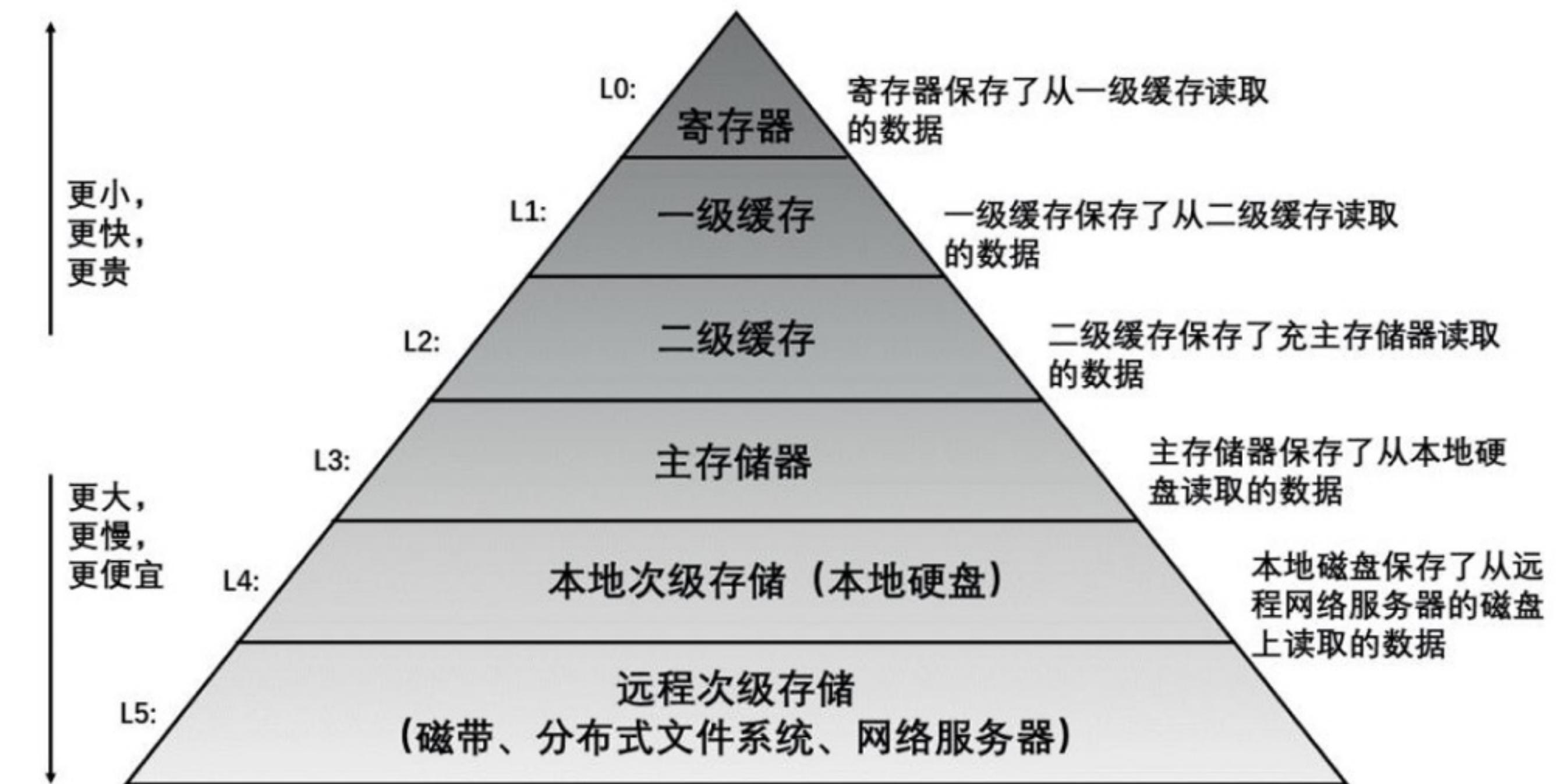
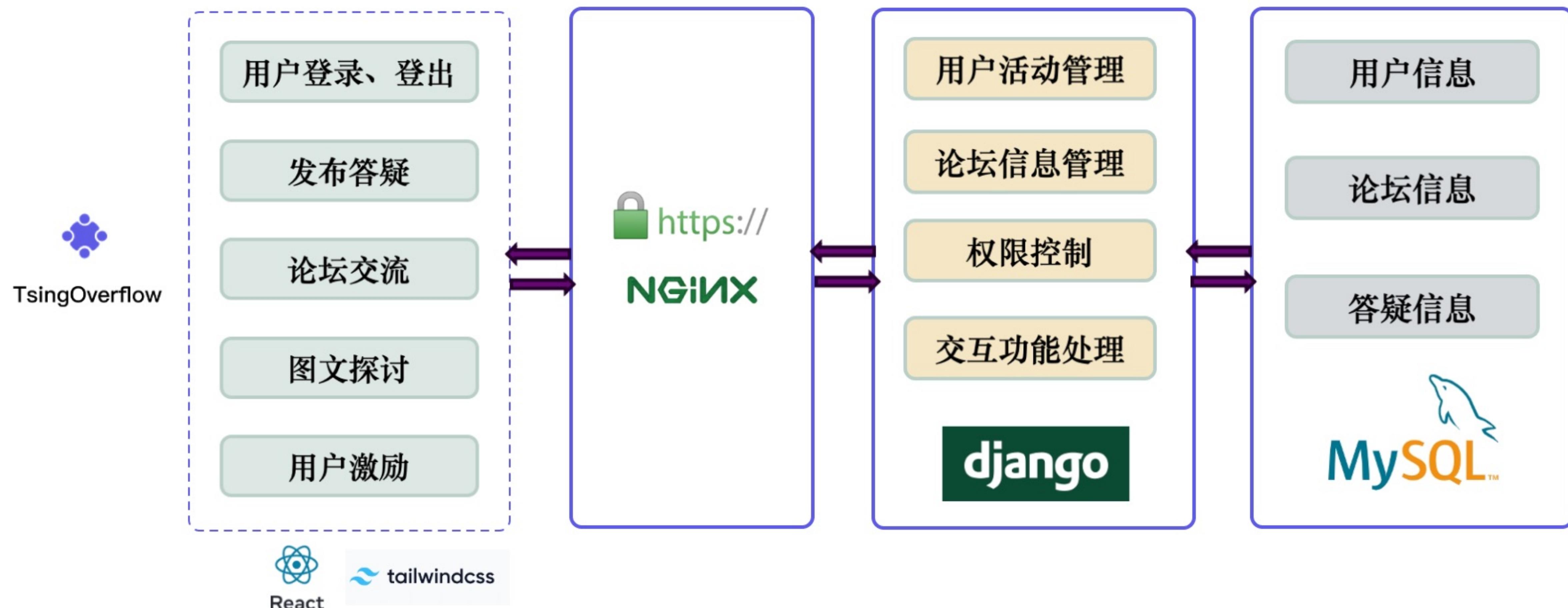


图. 缓存处于的位置 (来源 : CSAPP)

总体架构

◆ TsingOverflow



实际开发甘特图

基础环境搭建

- 基本符合中期预期

功能对接

- 开发困难的发源地
 - 显比预期时间长

公测

- 因不可抗力取消

部署

- 弥散于开发的全过程

Debug

- 需要做好长期斗争

每个环节分工明确

- 均存在一位“话事人”

组长的作用

- 统筹协调每个工作
- 铁腕项目经理

原则

- 谁有能力就做什么
- 谁有空闲时间就来做
- 时间“留余量”
- 体谅每个小组成员！
 - 毕竟我们都有期末、大作业和毕设三座大山。

工作	黎叙锐	黄砚星	段成业	苏宇然
前期调研	40%			60%
原型设计	3%	4%	3%	90%
中期 PPT	30%	15%	10%	45%
需求规格说明书	40%	10%	30%	20%
前端（重点）	5%	45%		50%
后端（重点）	90%			10%
单元测试	15%			85%
性能测试	30%			70%
部署	100%			
交付文档编写	45%	15%	25%	25%
交付演示 PPT	30%	20%	20%	30%