



系统分析与设计项目终期汇报

鸟鸣涧

涧

第11组

组员:

2051278 仝峰铭

2052635 孙泽

2052637 刘文超

2022.6.15

目录

一
系统架构

二
接口设计

三
机制设计

四
原型开发

澜

当前进展

架构层面上进行了重新设计，包括逻辑架构、开发架构与部署架构。

为子系统添加接口，并提供了相应的说明以及处理。

采用了数据持久化以及一些其他机制以提升项目的安全性、可靠性以及灵活性。

工人闲桂花落



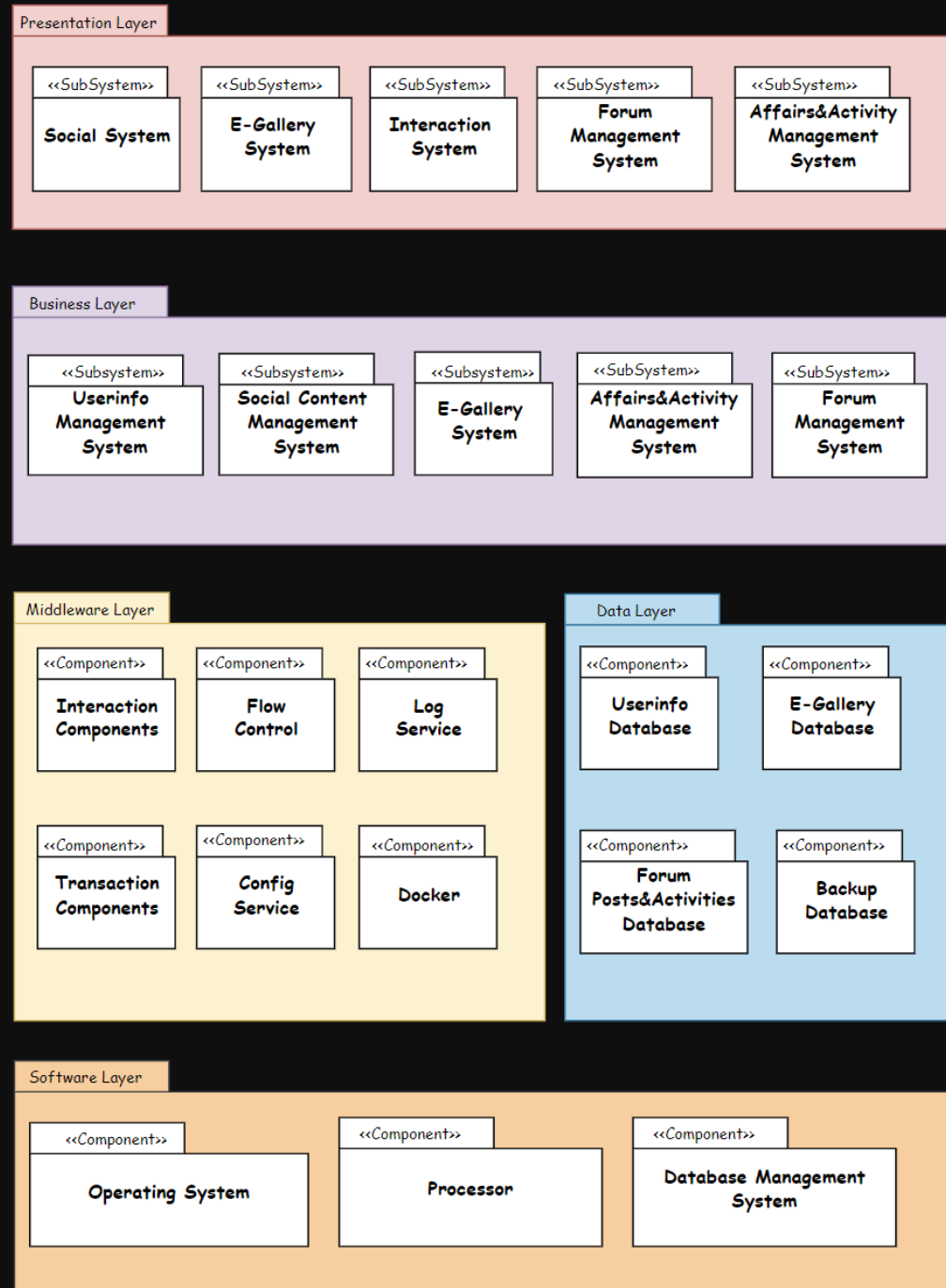
系统架构



逻辑架构

逻辑架构上进行了部分重新设计：

1. 优化对子系统的划分后，对 Presentation Layer 和 Business Layer 的部分进行了更新。
2. 对于用户各种数据的存储新增了 Data Layer 以完整保存关键数据。
3. 对于Middleware Layer增加了更多的新组件以完善服务。

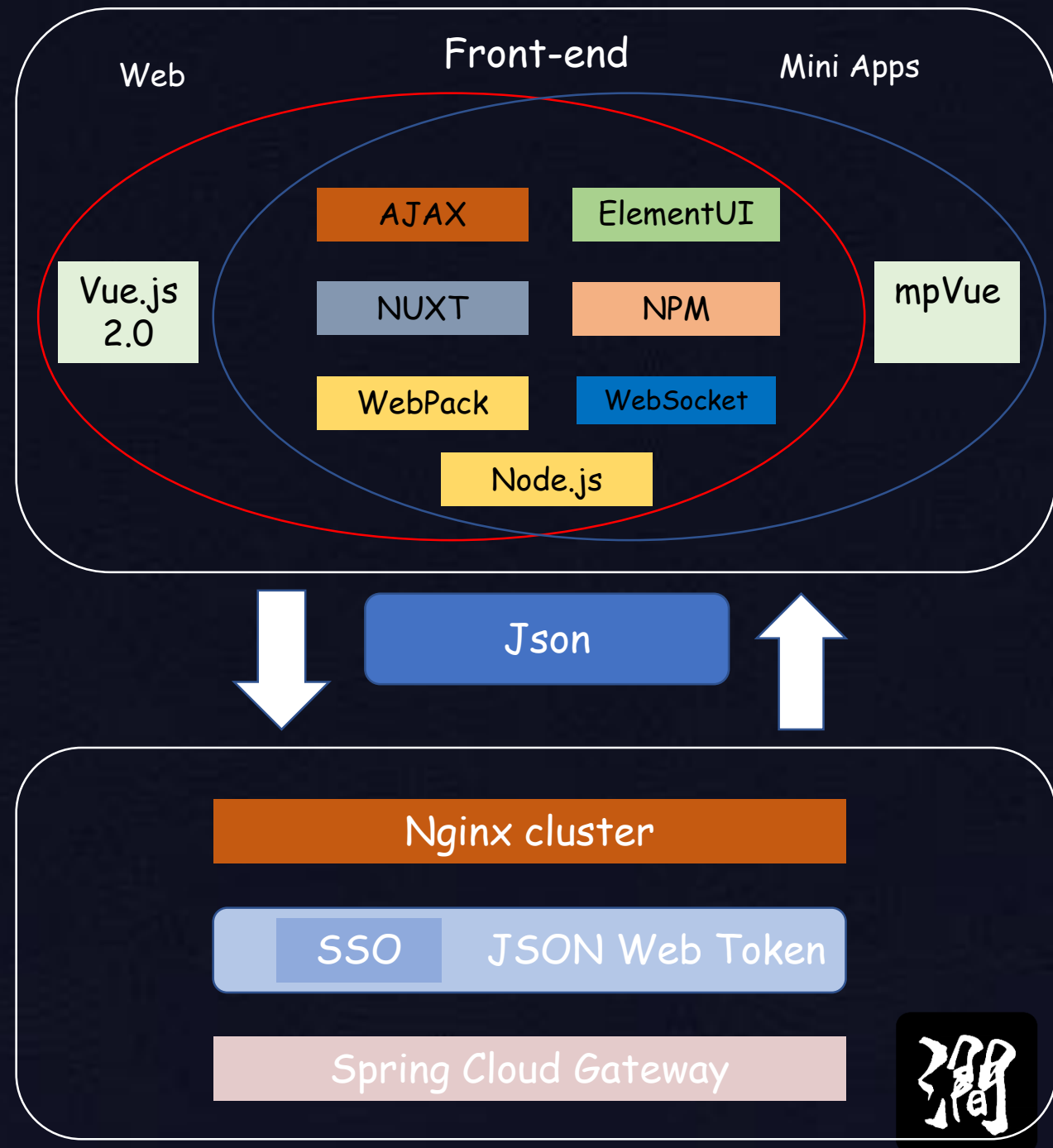


工
夜
静
春
山
空

开发架构

Presentation
Layer

Interface
Layer

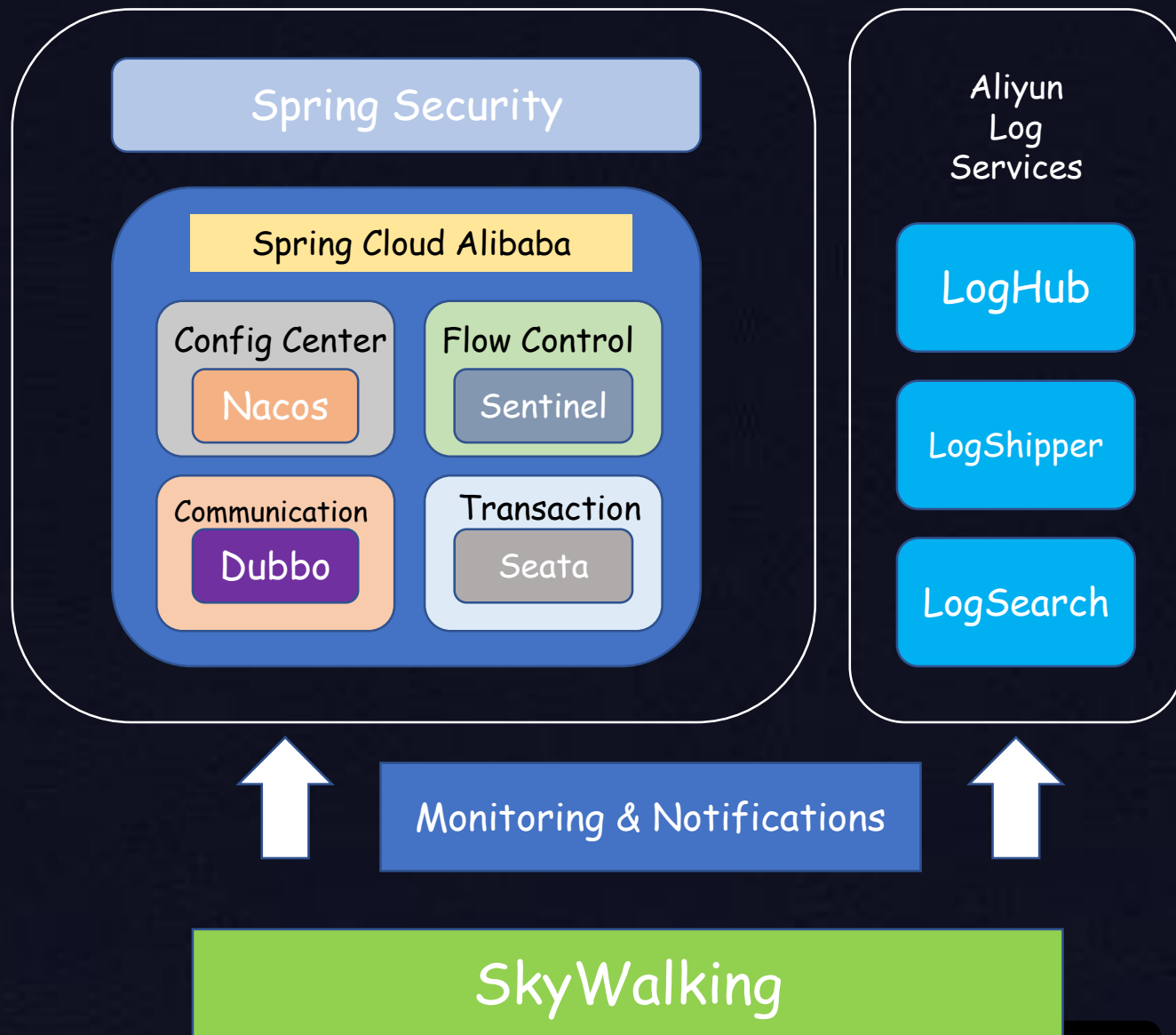


开发架构

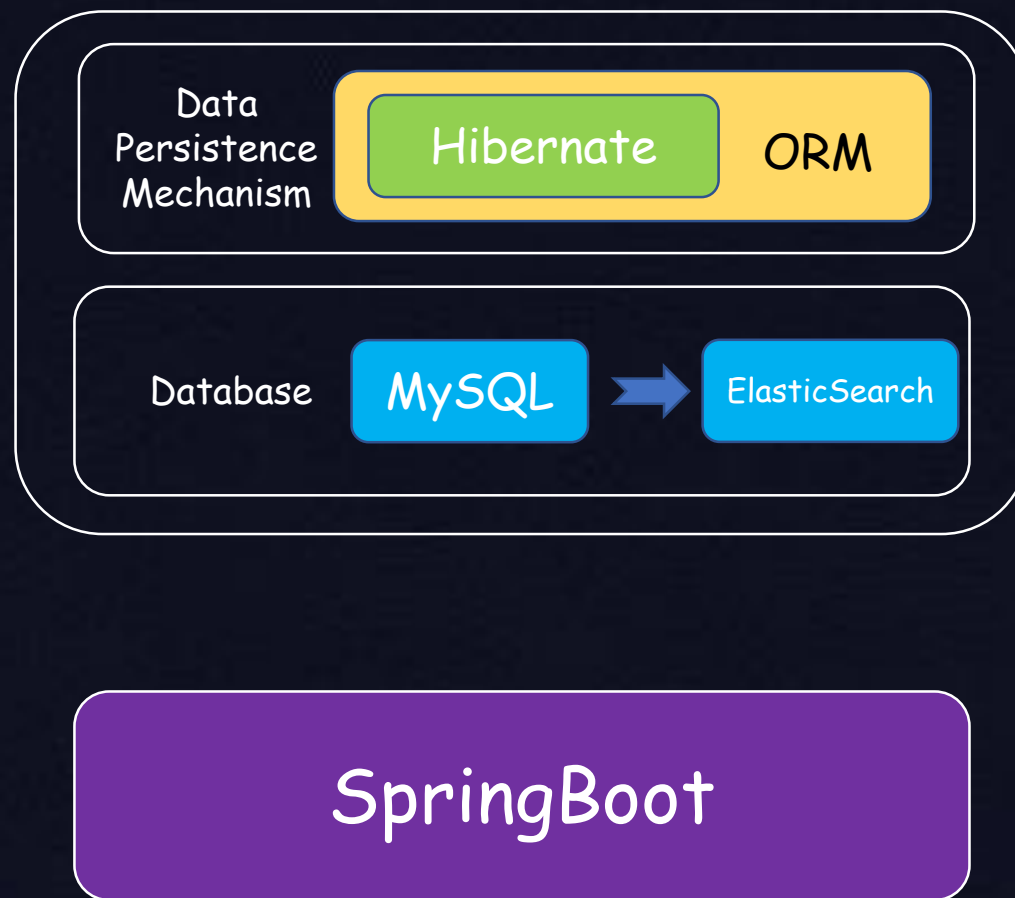
工
时
鸣
春
涧
中



Service
Layer



开发架构



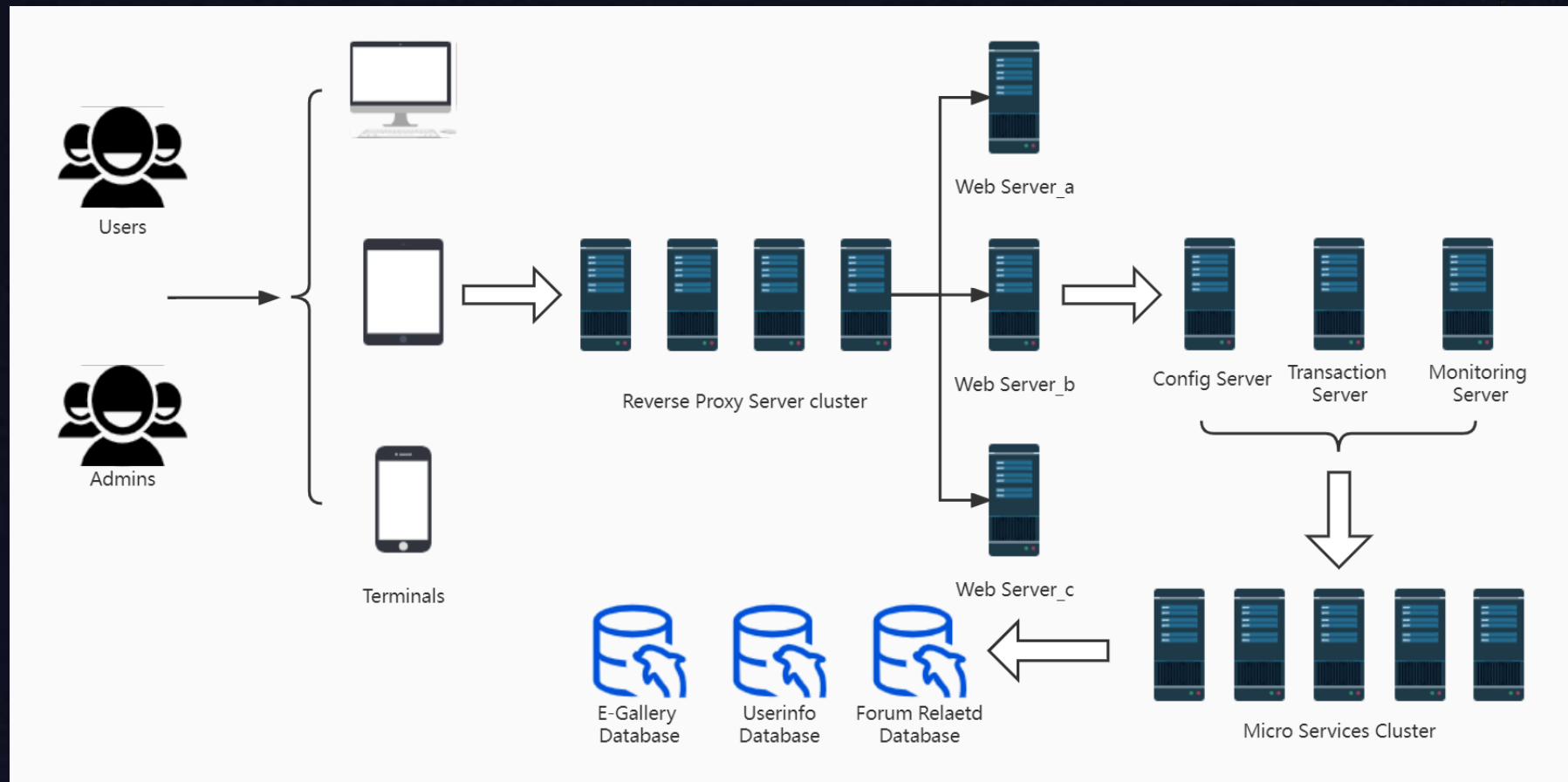
Data
Layer

Server
Layer

春去花还在

部署架构

人来鸟不惊



架构风格

工荆溪白石出



面向对象架构

数据抽象和封装的概念与以对象为中心的设计可以帮助开发人员从属性与方法两个方面把握问题，从而更好地实现系统。

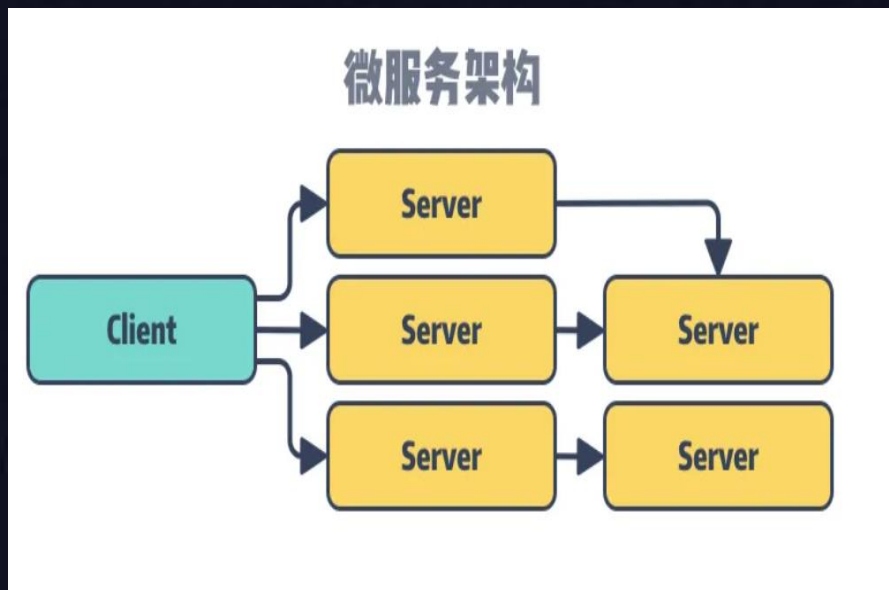
可以实现低耦合、高复用、可维护的系统，使其易于升级，并保护内部的完整性。

可以降低系统冗余，提高系统的整体性能。



架构风格

天寒红叶稀



微服务架构

微服务架构的思路是将应用分解为互相连接的微服务，一个特定的微服务完成一个特定功能。

高内聚、低耦合、不同服务独立部署的微服务架构系统，开发弹性强、迭代快，支持快速更新，可以满足当前系统的需要。

每个模块根据计算复杂量，用户访问量独立进行部署和运行，每个模块都可以独立进行水平向扩展，资源可以充分被利用



接口设计

MVC架构风格

MVC: Model-View-Controller

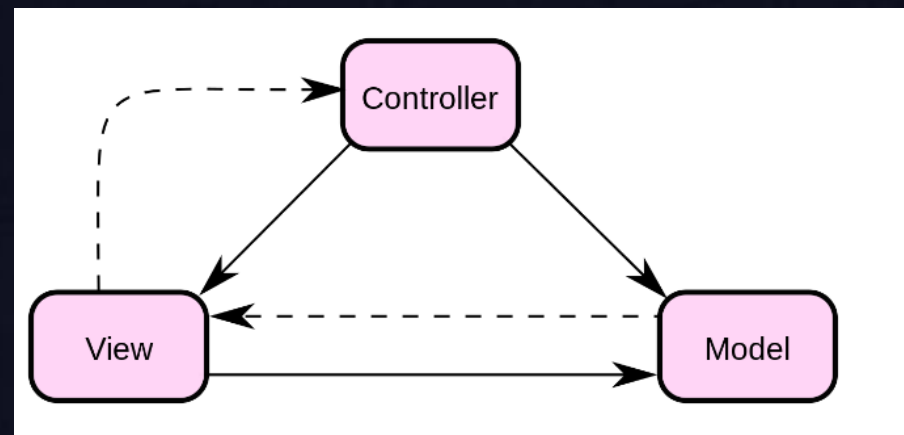
Model: 业务流程的处理过程对其它层来说是黑箱操作, 模型接受视图请求的数据, 并返回最终的处理结果。

View: 代表用户交互界面, 对于Web应用如Html等。

Controller: 从用户接收请求, 将模型与视图匹配在一起, 共同完成用户的请求。

优势:

- 重用性高
- 部署快, 生命周期成本低
- 可维护性高



山路元无雨



潤

子系统

晓看红湿处



部分类表

花重锦官城

UserInfo	+userID: string
	+name: string
	+userKey: string
	+contactMethod: string
	-registerTime: string
	-lastlandTime: string
	-state: enum
PaintingInfo	-privilege
	+userID: string
	+userName: string
	+paintingID: string
	+paintingType: string
	+creationDate: Enum
Comment	+isOriginal: bool
	+userID: string
	+text: string
	+postingTime: string
	+visible: bool
	-commentID: string

接口（模型层）

用户信息数据库

接口类：UserInfoDatabaseSystem

接口函数：

- 1.queryUserInfo(string UID):UserInfo
- 2.updateUserInfo(string UID,UserInfo newUserInfo):void
- 3.queryUserreputation(string UID):Repu
- 4.updateUserreputation(string UID,Repu newRepu):void

画作信息数据库

接口类：paintingInfoDatabaseSystem

接口函数：

- 1.queryPaintingInfo(string VID):void
- 2.storePaintingInfo(PaintingStoration PaintingInfo):void
- 3.updatePaintingInfo(PaintingStoration PaintingInfo):void

接口（视图层）

山中相送罢

上传画作系统

接口类：PaintingUpload System

接口函数：

1.sendUpload():void //上传画作

2.updatePaintingdatabase():void //更新画作数据库

社交系统

接口类：Social System

接口函数：

1.editMessage():void

2.editComment():void

3.leaveComment(Content new commnet):bool

//通过审核成功发布后返回true

事务管理系统

接口类：AffairManagement System

接口函数：

1.submitLogfile(file):void

2.editNotification():void

3.authorize(string UID):void//授予用户管理员权限

画作信息系统

接口类：IPaintingsInfoSystem

接口函数：

1.displayPaintingsInfo(string paintingType):PaintingInfo list

2.sharePainting():void

接口（控制层）

日暮掩柴扉

个人信息系统

接口类： IUserInfoSystem

接口函数：

1.getUserInfo(string UID):userInfo

2.updateUserInfo(sting UID,UserInfo newUserInfo):void

3.getFilteredPersonallInfo(FiteredCondition condition):PersonInfo List

内容管理系统

接口类： ISocialContentManagetSystem

接口函数：

1.auditContent(Content newContent):bool

2.editMessage():void

3.editComment():void

平台管理系统

接口类： Platform Management System

接口函数：

1.organizeActivity():void

2.reportInappropriateContent():void

3.deleteInappropriateContent():void

专栏管理

接口类： Column Management System

接口函数：

1.createColumn():void

2.authorizeModerator(UID):void//授权成为专栏版主



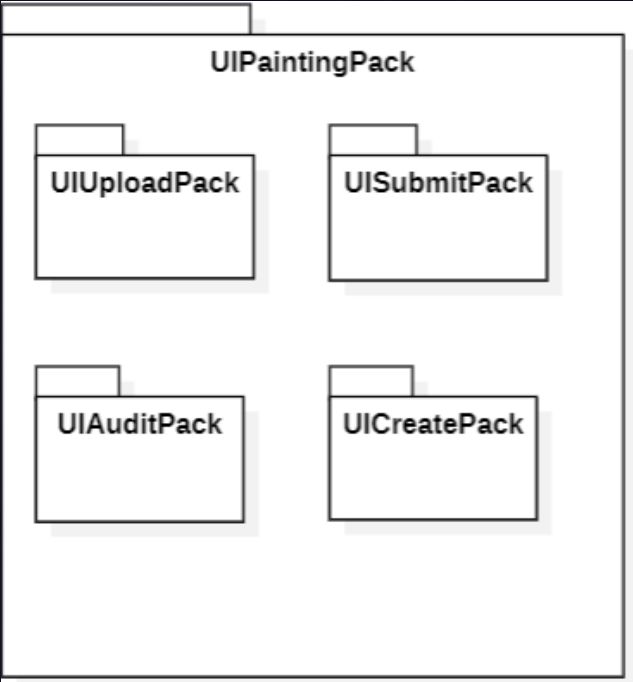
画作系统API规范

接口设计

画作信息接口

木末芙蓉花

«interface» IPaintingInfo
+appId: string +res_body: Json +res_code: int +res_error: string -token: string -url: string
+init(string appId, string token): void +uploadPainting(string url, string type, var data): bool +getPaintinginfo(string list tid): Json



画作系统API规范

参数列表

涧户寂无人

参数名称	类型	默认值	示例值	必要性	描述
painting_id	string			是	数据源
tags	string		山水画	是	标签，用于画作归类
pic_address	string		http://www.nmj.com/uploads/allimg/190103/1-1Z103130012163.jpg	是	图片的网络地址
markers	string		作者: xx 创作时间:xx ...	否	标注
uploader_id	string		10272	是	上传者id
quality	enum	1	1	否	1: 普通图; 2.高清图; 3.压缩图
size	int		800*800	是	图片宽度*图片高度

参数名称	类型	默认值	示例值	必要性	描述
code	int		0	是	0为上传成功，其他值为失败错误代码
msg	string		"pic_address不存在"	否	失败的原因，code不为0时，才有此字段
picurl	string		http://test.nmj.com/1a8b1.pg3	否	上传成功后生成的图片网络地址。code为0时，才有此字段

返回的JSON内容

提交系统API规范

参数列表

纷纷开且落

参数名称	类型	默认值	示例值	必要性	描述
act	string	submit	submit	是	固定值submit; 代表提交申请操作
type	enum		1	是	申请的类型; 1: 创建专栏; 2: 举办活动; 3: 人员变动
submission_info	string		本申请简要信息为...	是	申请的简介
fileitem	File		activity.docx	是	项目企划文件
uploader_id	string		10272	是	申请者id
token	string			是	用于用户登录状态的验证

参数名称	类型	默认值	示例值	必要性	描述
code	int		0	是	0为上传成功, 其他值为失败错误代码
msg	string		"pic_address不存在"	否	失败的原因, code不为0时, 才有此字段

返回的JSON内容

查看账号信息API规范

参数列表

空山不见人

参数名称	类型	默认值	示例值	必要性	描述
act	string	query	query	是	固定值: query; 代表查询请求
user_id	string		10272	是	用户id
user_key	string		12345abc	是	用户的账号密码

参数名称	类型	默认值	示例值	必要性	描述
code	int		0	是	0为提交成功, 其他值为失败错误代码
msg	string		"用户信息查询失败"	否	失败的原因, code不为0时, 才有此字段
user_id	string		10272	是	用户id
user_key	string		12345abc	是	用户的账号密码
name	string		test123	是	用户昵称 返回的JSON内容
contactMethod	string		123@126.com	否	绑定的联系方式
registerTime	string		2022-06-04 22:08:10	是	注册时间
lastlandtime	string		2022-06-04 22:08:10	是	最近登录时间
state	enum	1	1	是	账号状态; 1: 正常; 2: 限制; 3: 封禁
privilege	bool	0	0	是	账号权限; 0: 普通用户; 1: 版主

查看画作API规范

参数列表

但闻人语响

参数名称	类型	默认值	示例值	必要性	描述
pic	string		pic	是	固定值pic; 代表数据为画作
picurl	string		http://test.nmj.com/1a8b1.pg3	是	图片的网络地址
user_id	string		10272	是	请求查看用户id
token	string			是	用于验证当前请求发起者的登录状态

参数名称	类型	默认值	示例值	必要性	描述
code	int		0	是	0为上传成功, 其他值为失败错误代码
msg	string		"pic不存在"	否	失败的原因, code不为0时, 才有此字段
pic_id	string		20345	是	图片的id
pic_info	string		作者: xx 创作时间:xx ...	是	图片的基本信息
add_time	string		2022-6-1	是	上传时间
isDeleted	bool	false	false	是	图片是否被删除
altText	string	图片无法显示	图片不存在	否	在图片无法正常加载时显示的替换文字
altPic_url	string	http://xxxxxxx.pg3	http://test.nmj.com/1a8b1.pg3	否	在图片无法正常加载时显示的替换图片

返回的JSON内容

子系统接口规范

IPaintingUploadSystem

UserInfo	PaintingInfo
<div>-userID: string -userPassword: string -userName: string -gender: bool -credit: float</div>	<div>+userID: string +userName: string +paintingID: string +paintingType: string +creationDate: Enum +isOriginal: bool</div>
<div>«interface» IPaintingUploadSystem</div>	
<div>+PaintingUploadInfo: PaintingInfo +UploadorInfo: UserInfo</div>	
<div>+paintingUpload(UserInfo PaintingInfo): void +updatePaintingdb(PaintingInfo): void +getPaintinginfo(string list tid): Json</div>	

返景入深林



接口实现示例

在用户的登录过程中，设计将Sa-Token应用于其中：

基本流程：

当用户在未登录状态下访问一些需要登录后才能访问的功能；用户需登录后“持令牌”通过验证并访问对应功能。

```
public class LoginController
{
    // 测试登录示例，实际通过数据库比对
    // URL----http://birdsunggulley/xxx/doLogin?name=zhangsan&pwd=20149248
    @RequestMapping("doLogin")
    public Result doLogin(String name, String pwd)
    {
        if("zhangsan".equals(name) && "20149248".equals(pwd))
        {
            StpUtil.login(10001);
            return SaResult.ok("登录成功");
        }
        return SaResult.error("登录失败");
    }
    // 查询登录状态 ---- http://birdsunggulley/xxx/isLogin
    @RequestMapping("isLogin")
    public Result isLogin()
    {
        return Result.ok("是否登录: " + StpUtil.isLogin());
    }
}
```



接口实现示例

对于“记住我”模式：

勾选“记住我”按钮时：在浏览器写入一个持久Cookie储存Token，此时用户即使重启浏览器Token依然有效

不勾选“记住我”按钮时：在浏览器写入一个临时Cookie储存Token，此时用户在重启浏览器后Token便会消失，导致会话失效

```
// 使用 localStorage 保存token, 达到 [持久Cookie] 的效果
localStorage.setItem("satoken", "xxxx-xxxx-xxxx-xxxx-xxx");
// 使用 sessionStorage 保存token, 达到 [临时Cookie] 的效果
sessionStorage.setItem("satoken", "xxxx-xxxx-xxxx-xxxx-xxx");
//示例:
StpUtil.login(10001, new SaLoginModel()
    .setDevice("PC") // 此次登录的客户端设备类型,
    .setIsLastingCookie(true) // 是否为持久Cookie
    // .setTimeout(60 * 60 * 24 * 7) // 指定此次登录token的有效期
    .setToken("xxxx-xxxx-xxxx-xxxx") // 预定此次登录的生成的Token
);
```

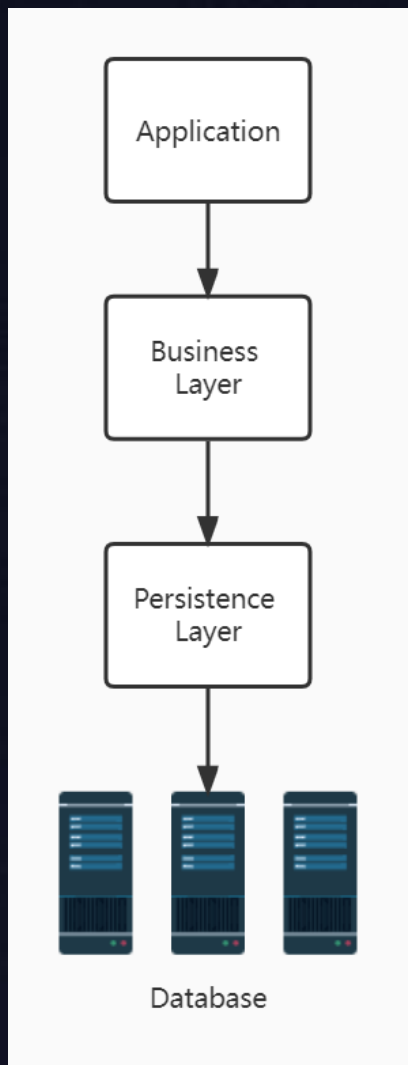


机制设计



机制设计

泉听咽危石



持久化机制

本系统中的大量数据需要被储存在关系型数据库中，用于核心功能的实现。

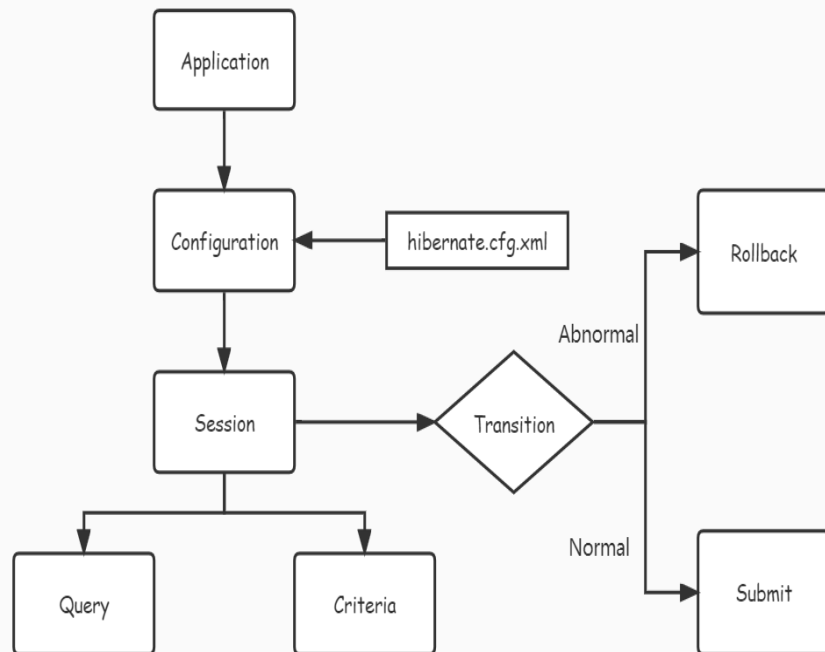
在业务层和数据库之间添加了持久化层，协助并自动将对象存储到数据库中。

在这样一层应用程序和数据库之间的抽象下，程序将通过会话（session）间接访问数据库。



机制设计

日色冷青松



Hibernate

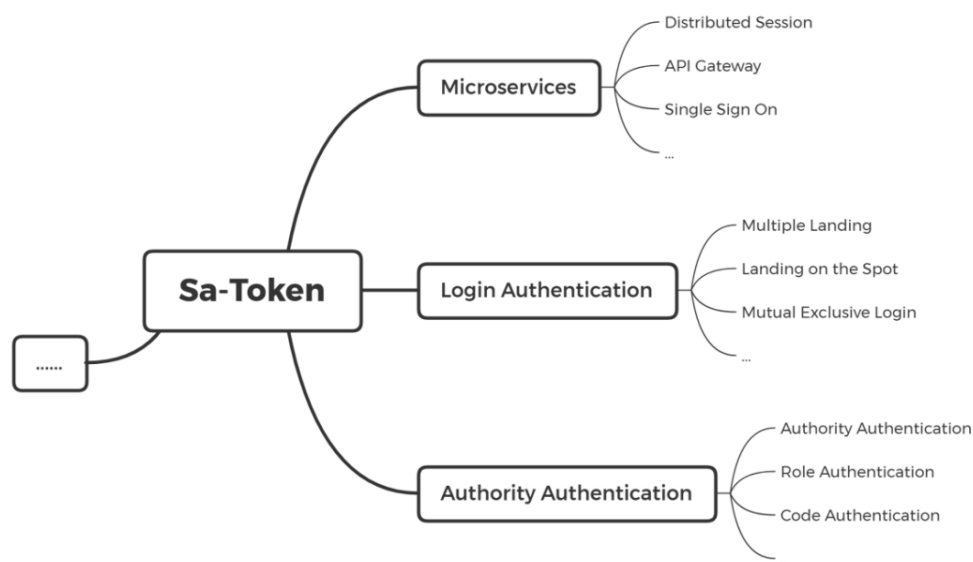
Hibernate是一个开源的、全自动的对象关系映射，即ORM框架。对JDBC进行了轻量级的封装，可以自动生成SQL语句并执行。

Hibernate可移植性更强，在对象的维护和缓存方面更具优势，更加稳定。



机制设计

古木无人径



权限认证：Sa-Token

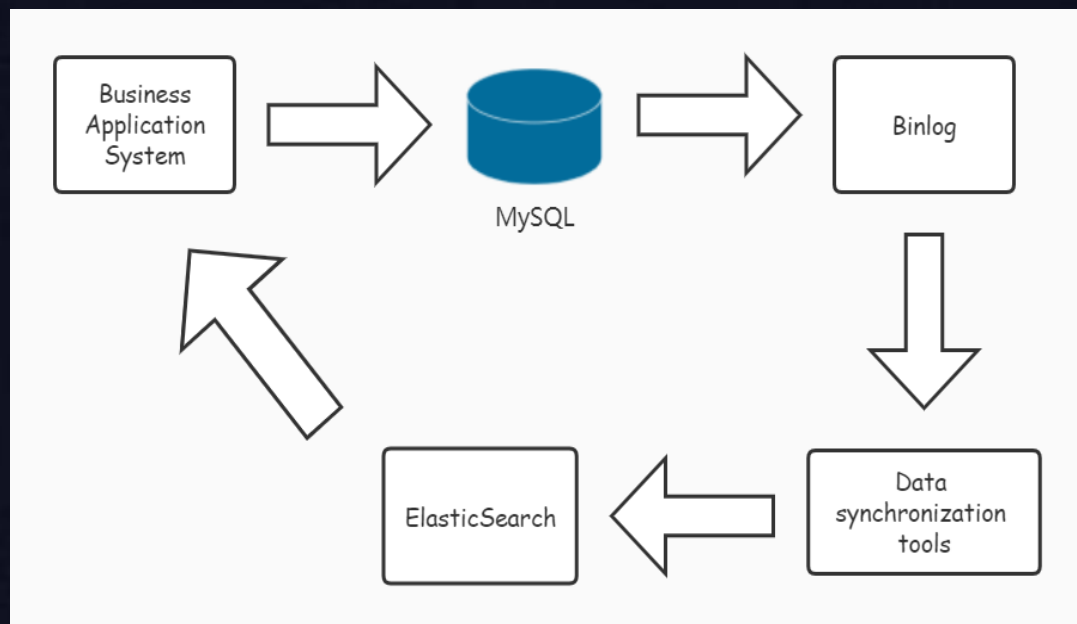
Sa-Token是一个轻量级的、支持前后端分离的权限认证框架。

其集合了微服务架构模式下的登录认证，权限认证等基本功能，API调用简单易行，权限认证安全可靠。



机制设计

深山何处钟



读写分离

考虑到用户时常需要进行查询操作，数据库的读写操作的使用频率是存在较大差异。

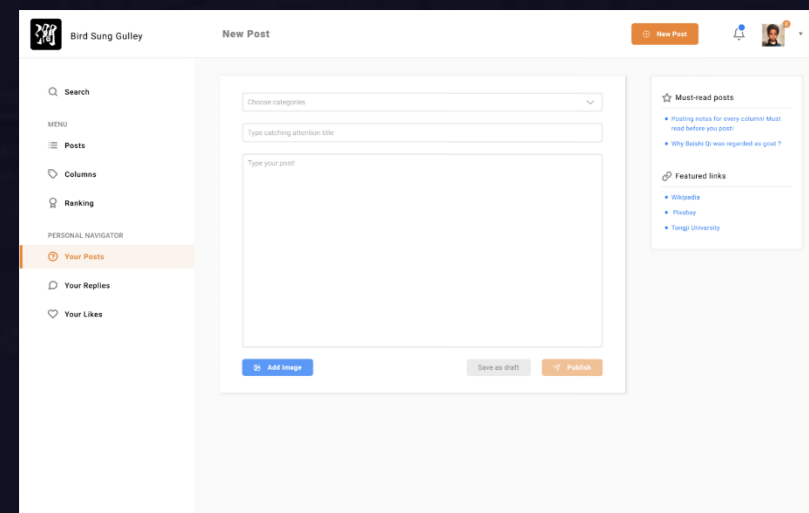
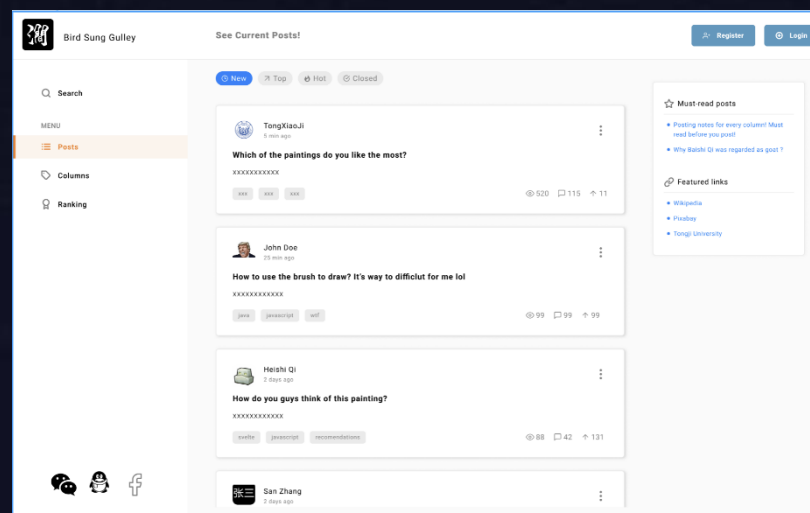
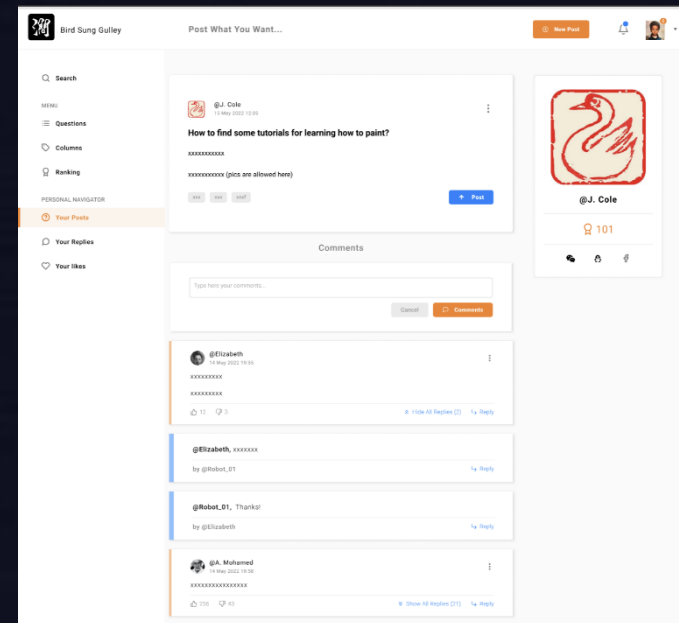
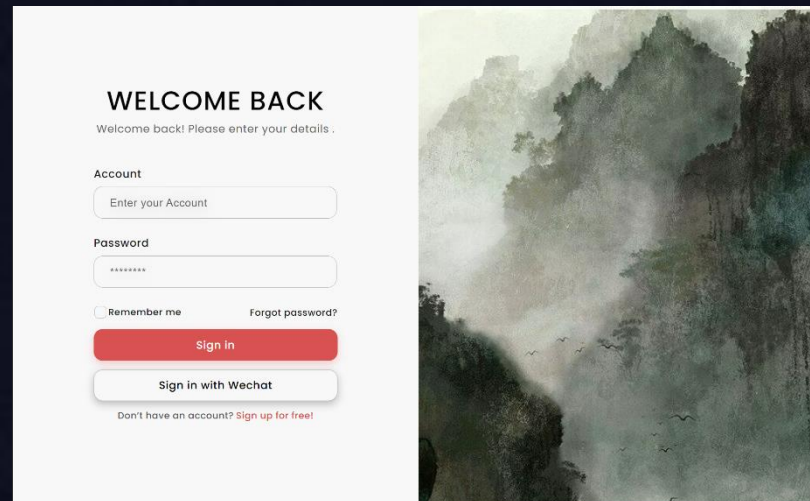
使用MySQL数据库用来进行写操作，ElasticSearch用来进行读操作。为了保证MySQL和ElasticSearch数据同步，我们使用Binlog技术



原型开发

前端原型

独坐幽篁里



后端原型

弹琴复长啸



```
events
{
    worker_connections 1024;
}

http
{
    include mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user[$time_local] "request" '
        '$status $body_bytes_sent "$http_referer"'
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log logs/access.log main;
    sendfile on;
    tcp_nopush on;

    gzip on;
    upstream tomcat_proxy
    {
        server 192.168.3.63:8080 max_fails = 3 weight = 1 fail_timeout = 60;
        server 192.168.3.64:8080 max_fails = 3 weight = 1 fail_timeout = 60;
    }
}
```

```
@Service
public class LoginServiceImpl implements LoginService
{
    @Autowired
    private UserMapper UserMapper;

    @Override
    public Result login(LoginDTO loginDTO)
    {
        if (StringUtils.isEmpty(loginDTO.getLoginName()))
            return new Result(400, "You haven't enter your account!", "");
        if (StringUtils.isEmpty(loginDTO.getPassword()))
            return new Result(400, "You haven't enter your password!", "");

        QueryWrapper<User> wrapper = new QueryWrapper();
        wrapper.eq("login_name", loginDTO.getLoginName());
        User user = UserMapper.SelectOne(wrapper);
        if (user != null && user.getPassword().equals(loginDTO.getPassword()))
        {
            LoginVO loginVO = new LoginVO();
            loginVO.SetID(user.GetID());
            loginVO.SetToken(UUID.randomUUID().toString());
            loginVO.SetUser(user);
            return new Result(200, "", loginVO);
        }
        return new Result(401, "Login Failed", "");
    }
}
```

涧



THANKS!

