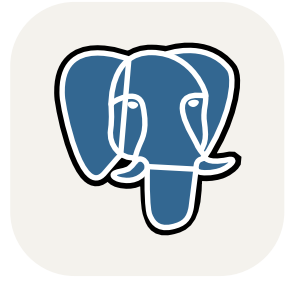
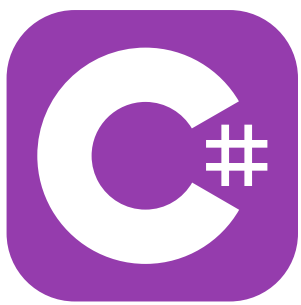


南方财富

Southern Money



一、简介与开发

本质是一个“帖子与交易”的综合性应用

- 社区论坛功能（用户可发布、浏览、管理帖子）
- CS:GO枪械皮肤交易市场（商品分类、详情、发布与管理功能）
- 强大的用户登录认证、个人中心管理及管理员后台审核
- 其它非数据库交互的系统功能（主题颜色、动画时长、横竖屏适配等）



前端：Dart语言 + Flutter框架

后端：C# + ASP.Net + EF Core + Swagger

数据库：SQLite, Postgres

代码管理工具：Git

服务器：8核CPU/8G内存

二、主要功能与数据库设计

帖子功能

Post
PostImage
PostTag
PostLike
PostFavorite

- 支持帖子发布、浏览、点赞、编辑、收藏等功能
- 实现了帖子搜索和分类筛选
- 支持帖子管理

产品功能

(饰品、金融产品等)

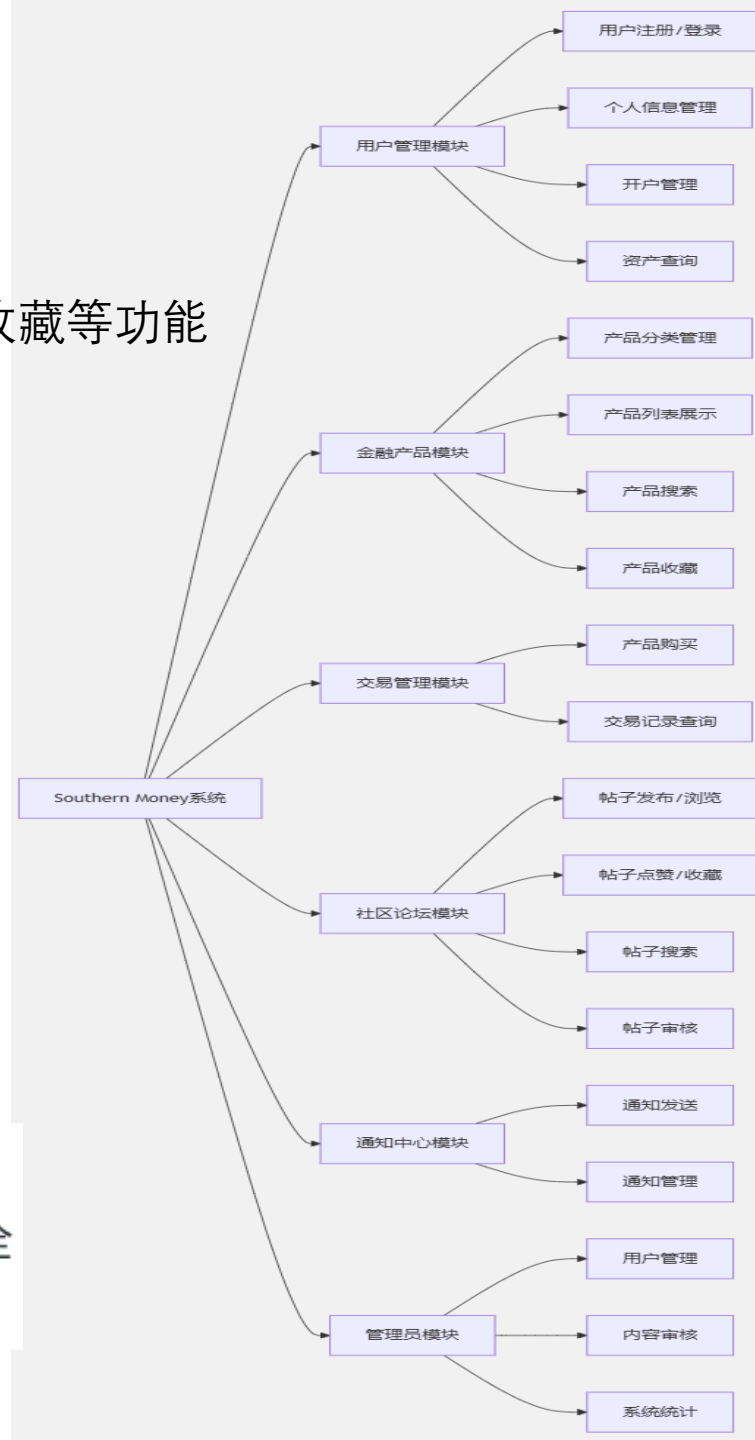
Product
TransactionRecord
UserAsset

- 支持产品浏览、分类查看、搜索等功能
- 实现了产品购买和交易记录管理
- 支持产品收藏和分类收藏

用户功能

User
UserAsset

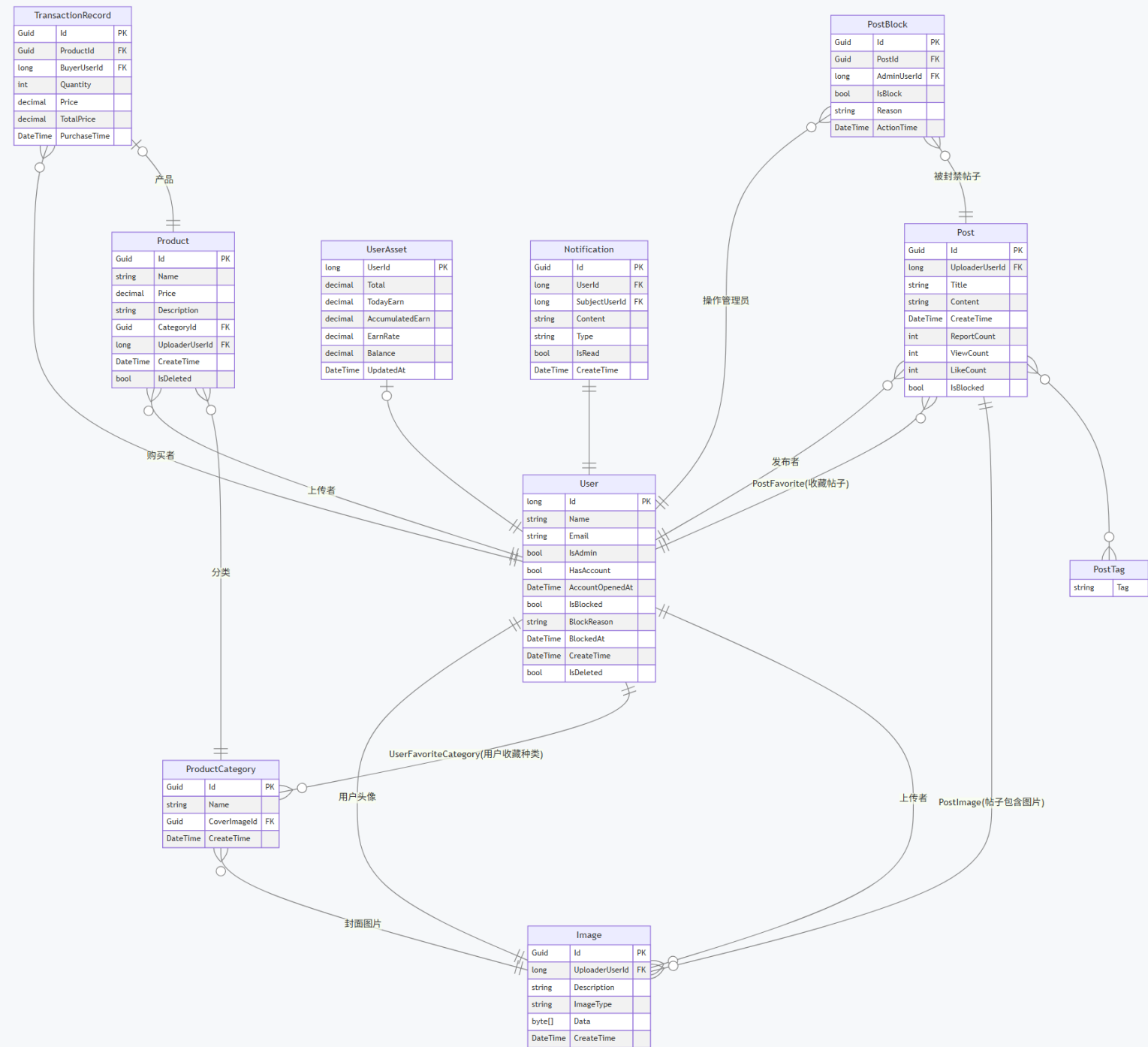
- 支持用户注册、登录、密码修改
- 实现了JWT令牌认证机制，确保API访问安全
- 支持自动登录和令牌刷新



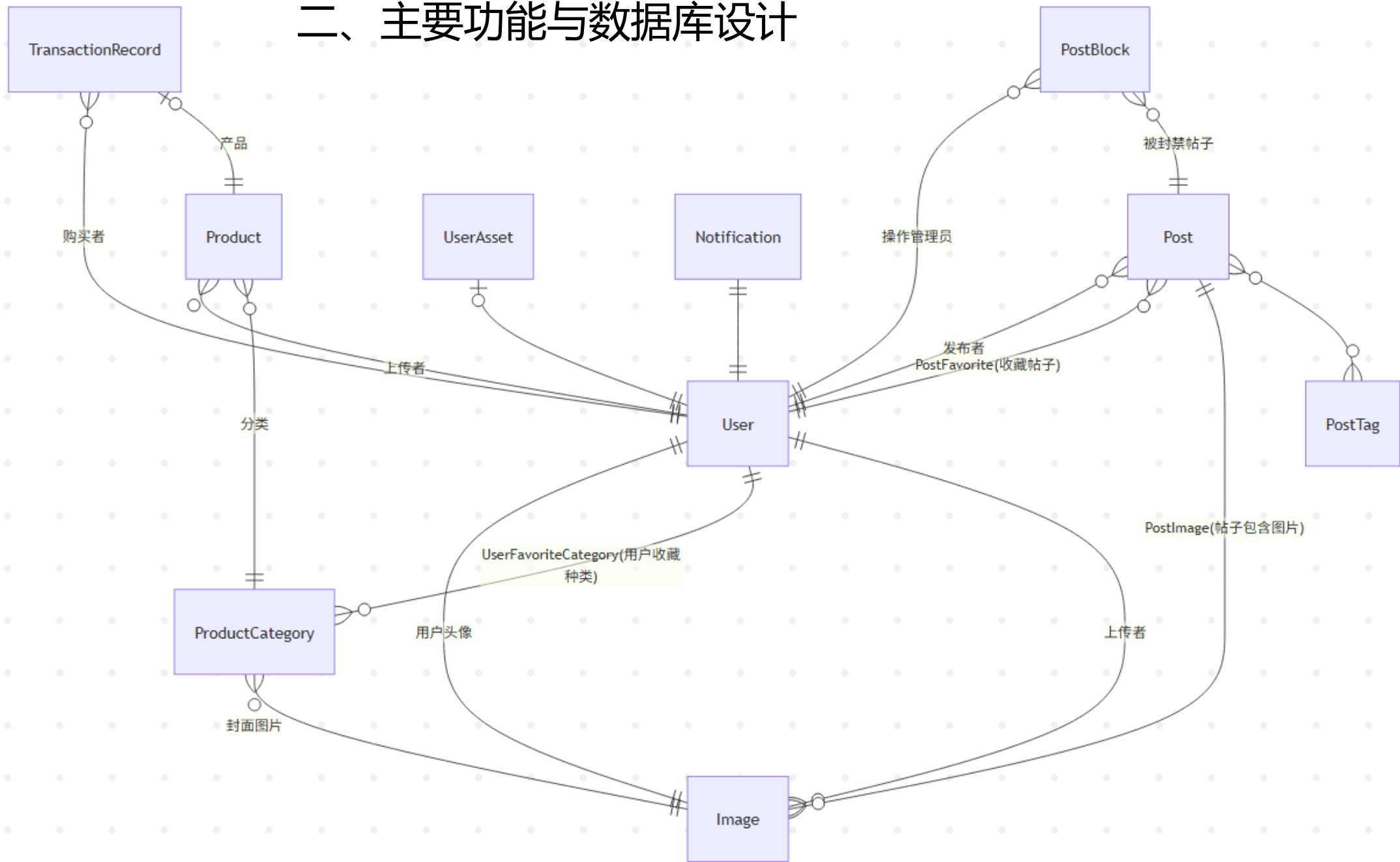
二、主要功能与数据库设计

E-R图

关系示例：
 Post 与 Image 是 1:N 关系
 Post 与 Tag 是 N:N 关系
 UserAsset 与 User 是 1:1 关系



二、主要功能与数据库设计



三、数据库优化策略

分页：使用 `Skip()` 和 `Take()` 实现分页，避免全表扫描和内存溢出；

预加载：使用 `Include()` 和 `ThenInclude()` 加载关联实体，避免 N+1 查询问题，减少数据库查询次数；

延迟执行：使用 `IQueryable` 构建查询，在调用具体方法(如 `CountAsync()`)才执行

条件查询：使用 `Where()` 子句过滤冗余数据，减少返回结果集大小

数据完整性：使用 `IsDeleted` 字段标记已删除记录，而非物理删除-保留数据完整性，支持数据恢复

系统设计：通过存储库模式分离数据访问逻辑与业务逻辑，提高代码可维护性

认证与安全：密码安全和JWT认证

```
.Skip((page - 1) * pageSize)
.Take(pageSize)
```

```
var query = _context.Posts
    .Include(p => p.User)
    .Include(p => p.PostImages)
    .ThenInclude(pi => pi.Image)
```

```
./Database/Repositories
├── ImageRepository.cs
├── NotificationRepository.cs
├── PostRepository.cs
├── ProductCategoryRepository.cs
├── ProductRepository.cs
├── TransactionRepository.cs
├── UserAssetRepository.cs
├── UserFavoriteCategoryRepository.cs
├── UserRepository.cs
├── SouthernMoneyBackend/Controllers
├── AdminController.cs
├── ImageBedController.cs
├── LoginController.cs
├── NotificationController.cs
├── PostController.cs
├── StoreController.cs
├── TestController.cs
├── TransactionController.cs
├── UserController.cs
```

```
postsQuery = postsQuery
    .OrderByDescending(p => p.ReportCount)
    .ThenByDescending(p => p.CreateTime);
var totalCount = await postsQuery.CountAsync();
var posts = await postsQuery
    .Skip((page - 1) * pageSize)
    .Take(pageSize)
    .ToListAsync();
```

```
return await _context.PostFavorites
    .Where(pf => postIds.Contains(pf.PostId) && pf.UserId == userId)
    .Select(pf => pf.PostId)
    .ToHashSetAsync();
```

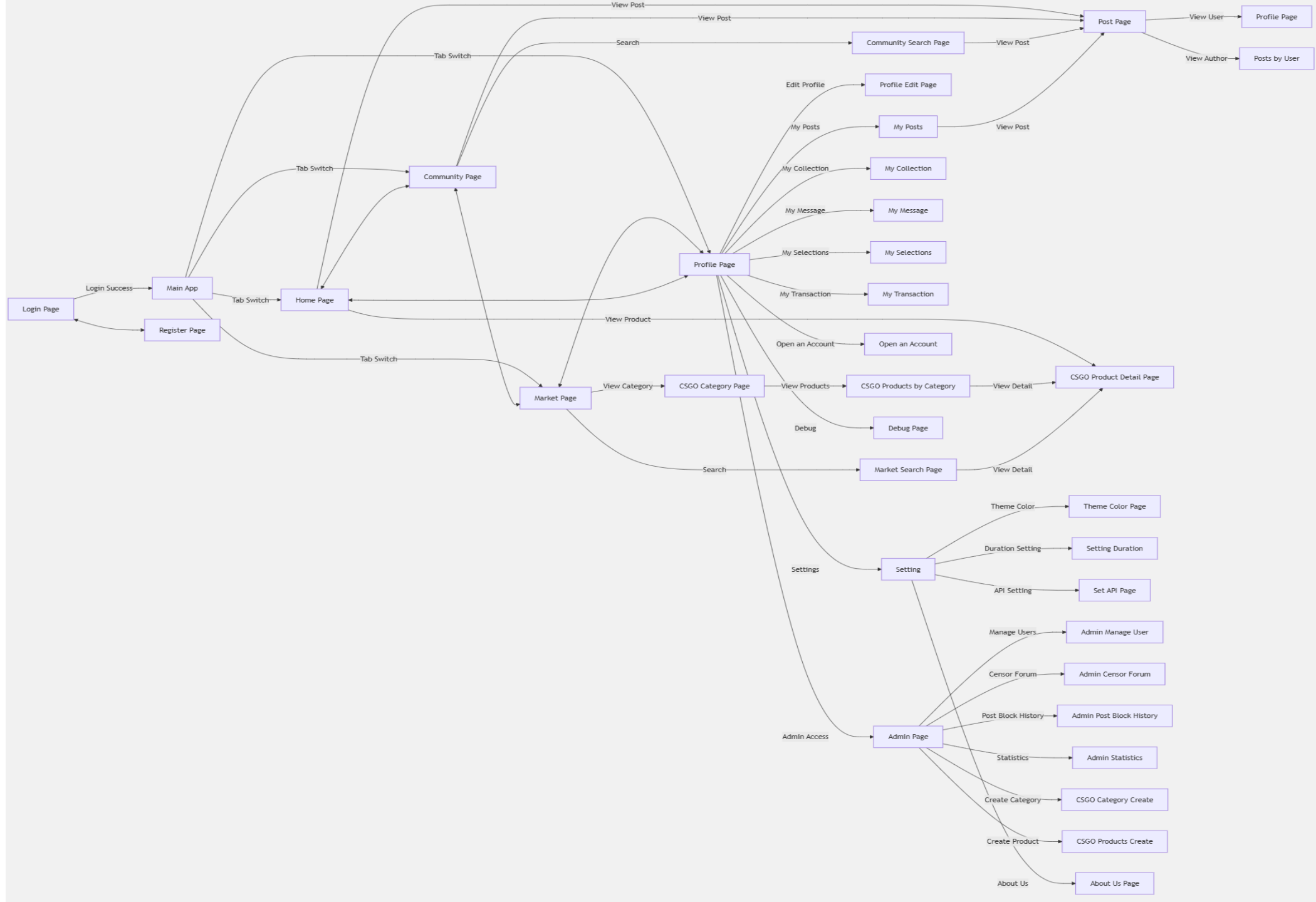
```
public bool IsDeleted { get; set; } = false;
```

```
public async Task<long> RegisterUser(Database.User user, bool isGuest)
{
    user.Password = Utils.HashPassword(user.Password);
```

```
9 public static class JwtUtils
```

```
136 public static ClaimsPrincipal? ValidateToken(string token)
```

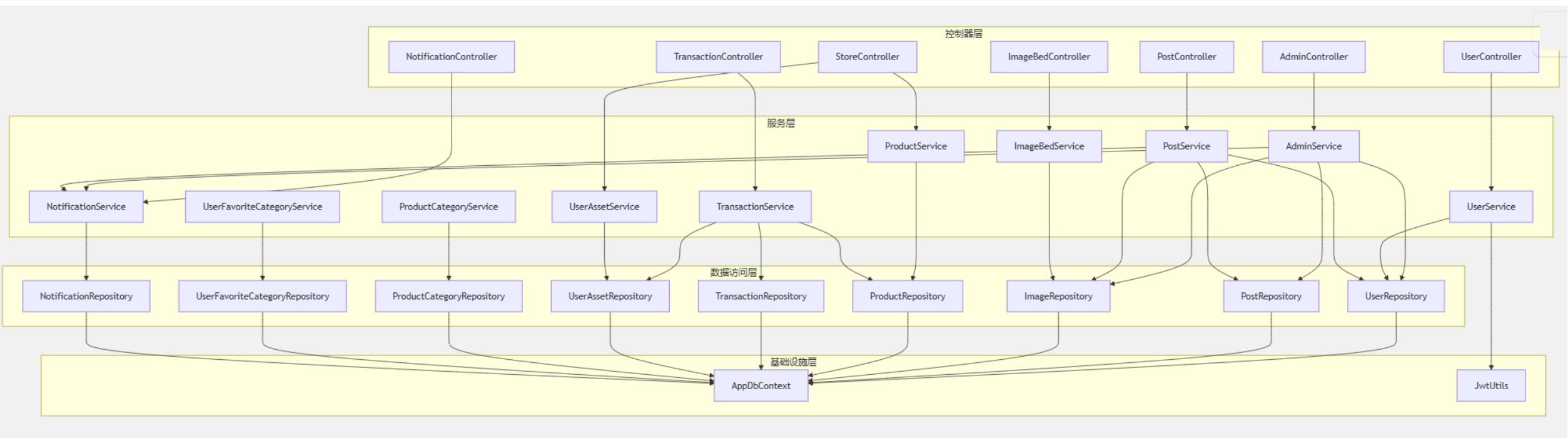
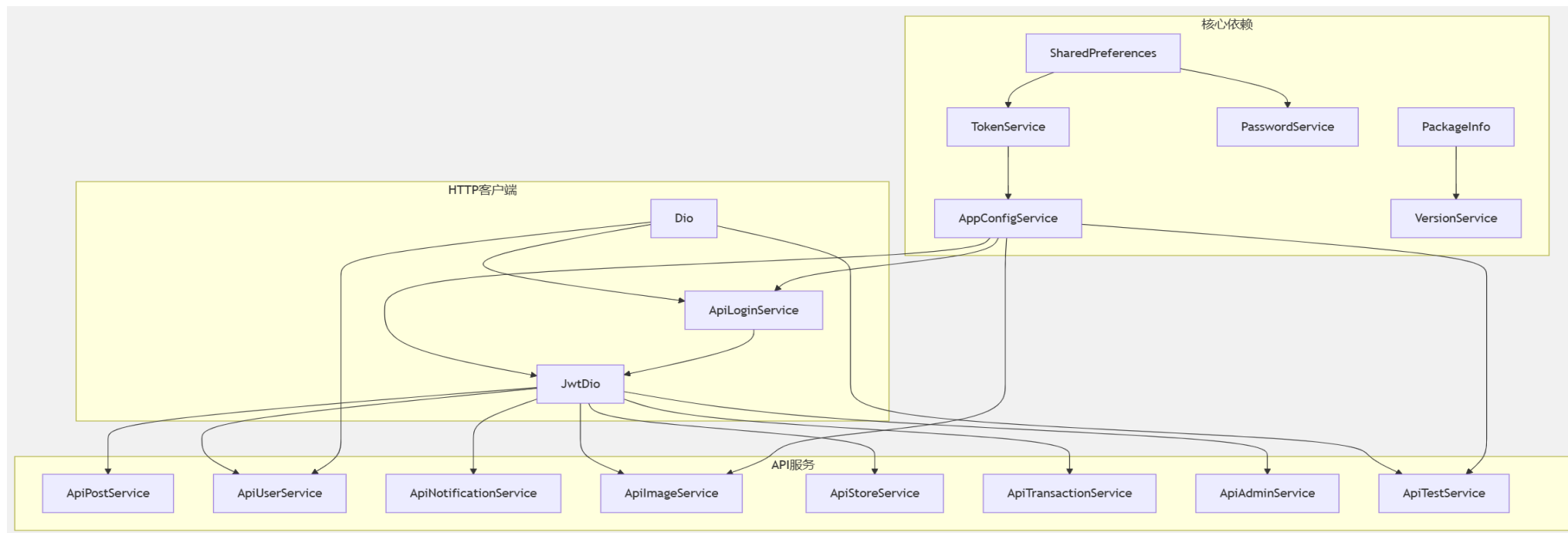
四、前端 页面架构



五、系统架构

前端层次架构 (右)

后端层次架构 (下)

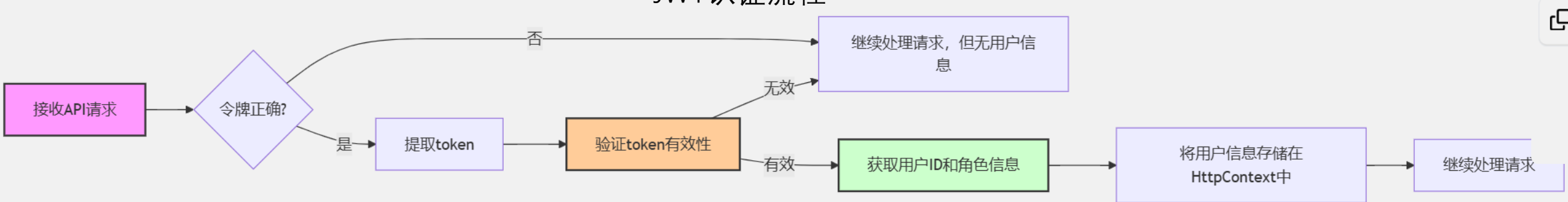


五、系统架构

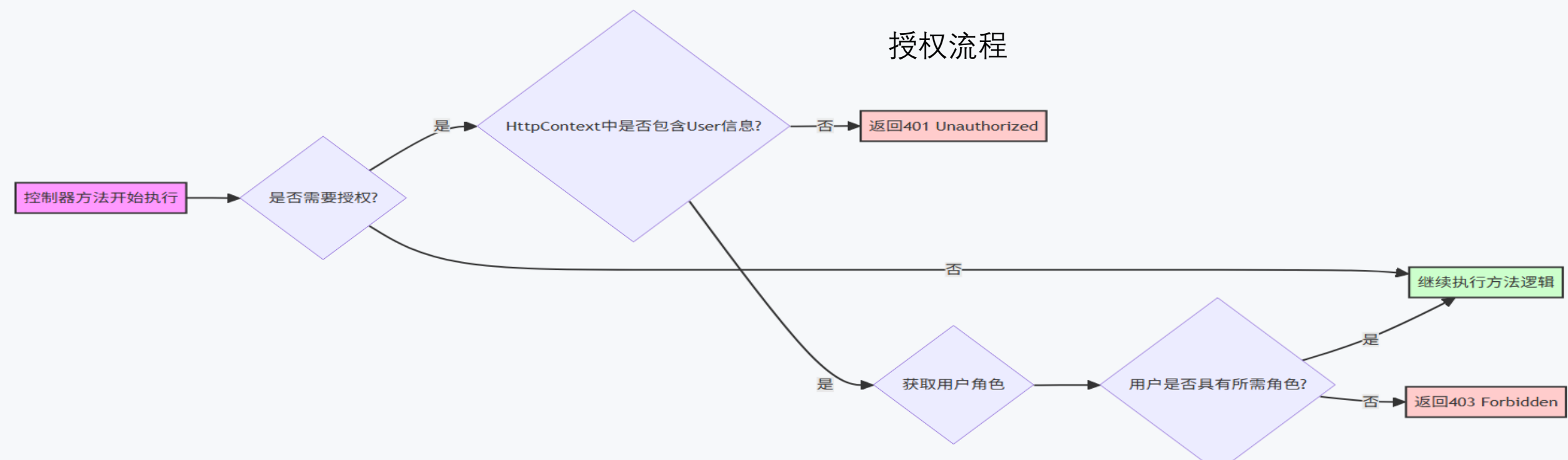
后端架构



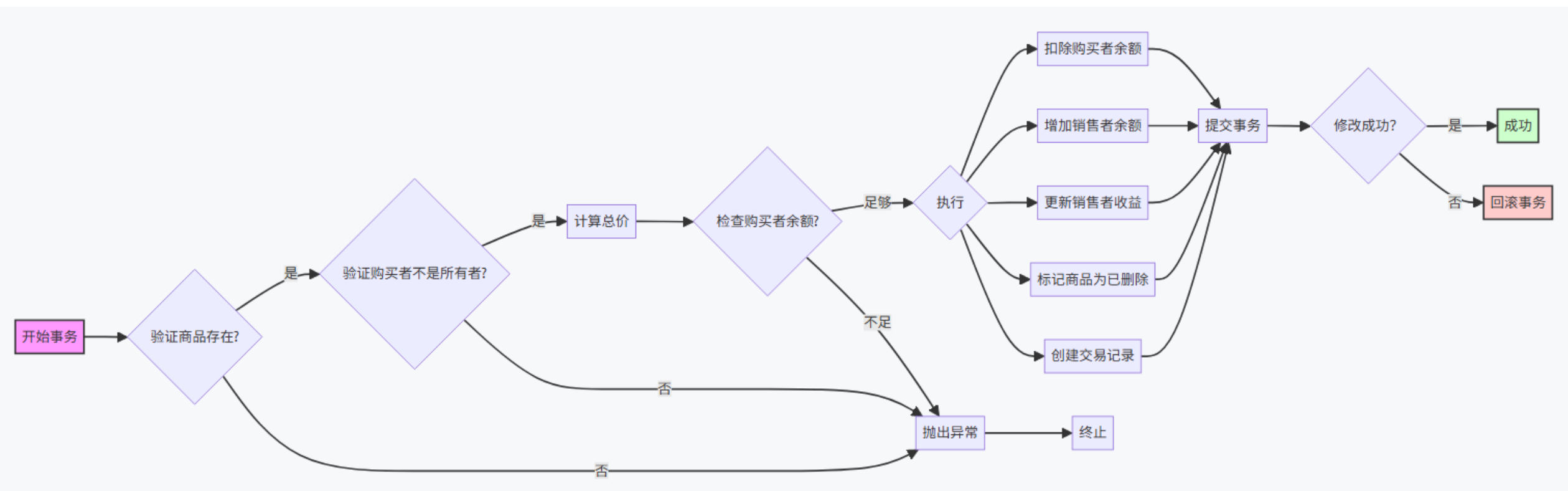
JWT认证流程



授权流程

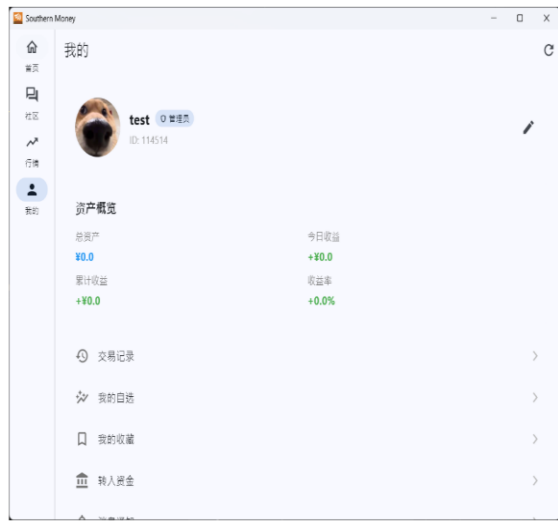


商品购买流程



六、项目演示

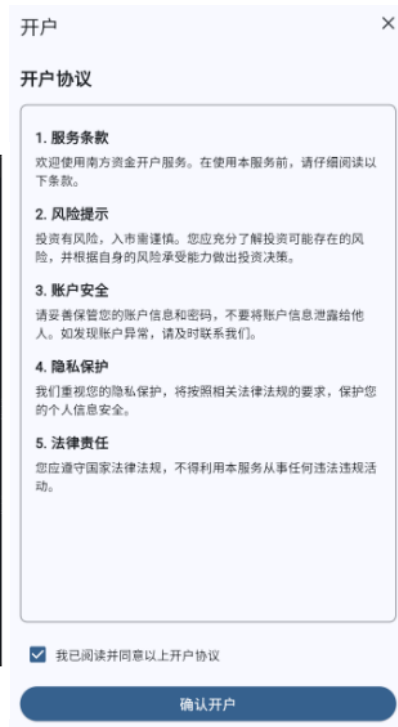
- 南方财富



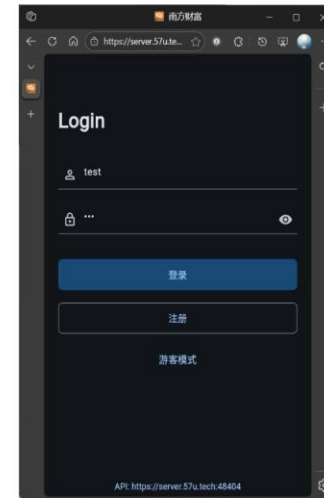
亮色主题



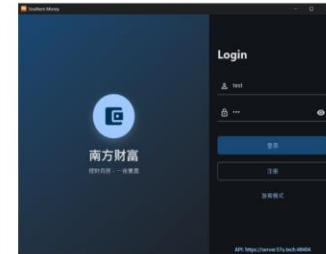
暗色主题



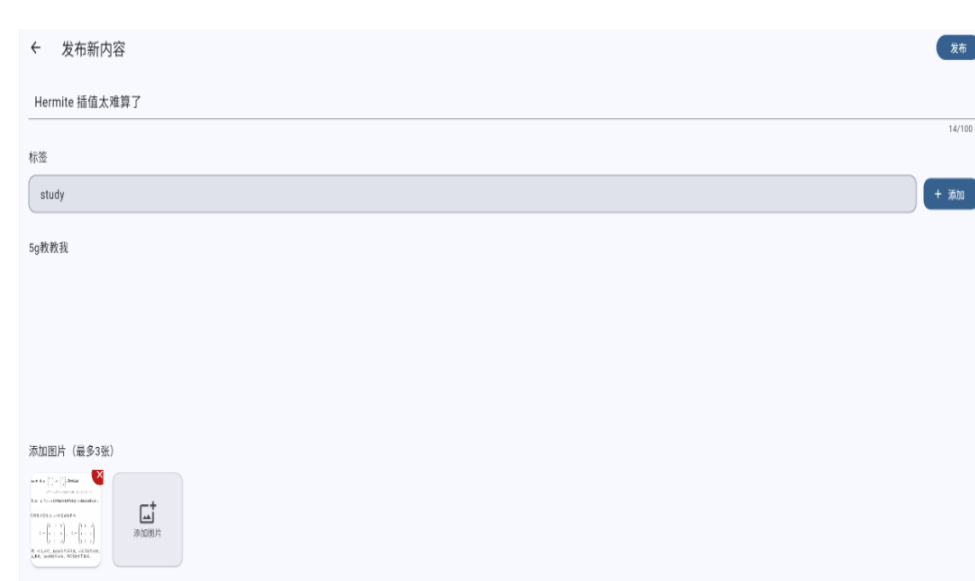
Android平台展示



Web平台展示

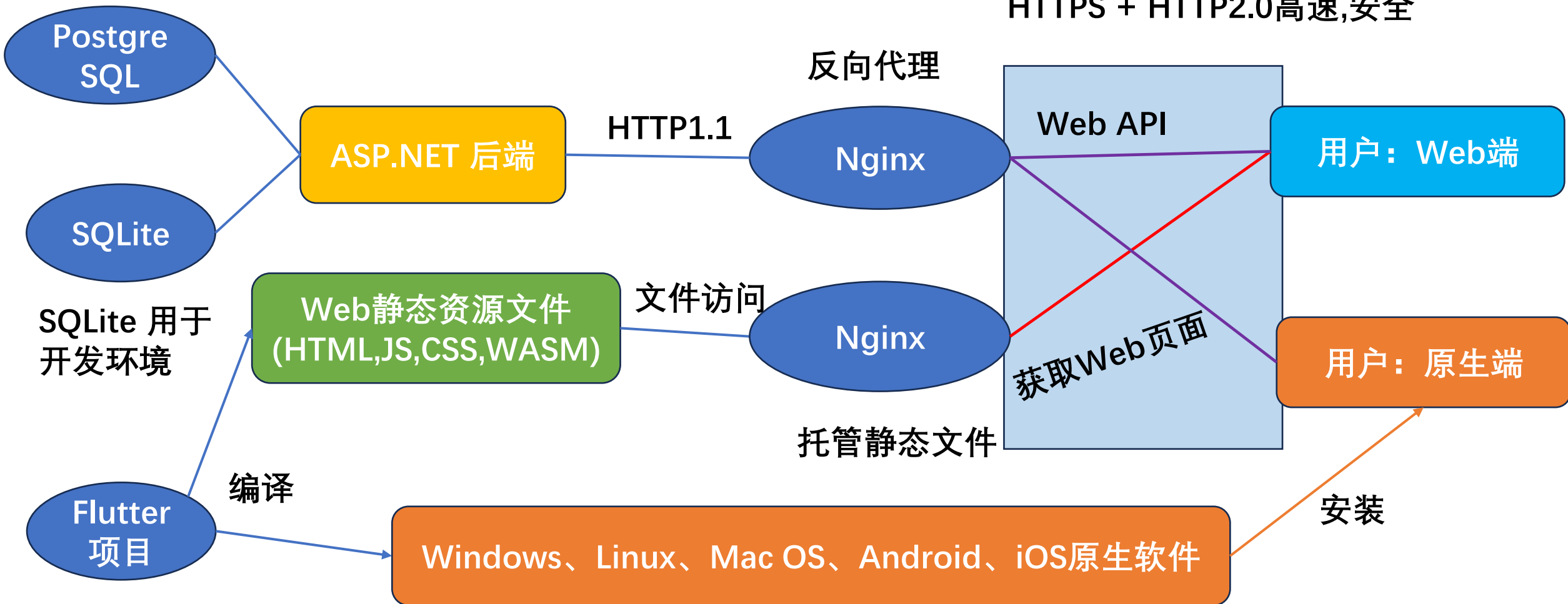


Windows平台展示



部署

PostgreSQL 用于生产环境



Thanks for Your Listening!