一、后端

1.1、环境要求

- 使用Golang 1.18及以上版本
- 数据库操作推荐GORM，尽可能少用原生SQL操作
- 系统需兼容Linux、Mac、Windows平台

1.2、资源并发

- 凡涉及竞争的资源，应做好并发处理，如次卡扣减、产品库存、人次限购等场景
- 开发时应考虑集群产生的资源竞争问题，使用分布锁非单机锁

1.3、代码规范

- 参照Effective Golang，统一风格（重要）

- 关键业务逻辑应合理添加备注

- 所有涉及状态相关，需统一定义与引用，如定单状态：

```
const (
        OrderStatusInit    = 1
        OrderStatusInit = 2
        ...
)

# 允许方式
db.Where("order_status IN(?)",[]int{OrderStatusInit,OrderStatusInit})
# 不允许方式
db.Where("order_status IN(?)",[]int{1,2})
db.Where("order_status IN(1, 2)")
```

1.4、API文档

- API文档使用swagger，请求及返回参数有明确注解

二、前端

2.1、开发语言

- 使用js开发，运管管理系统、商户管理系统采用react web框架

2.2、UI组件库

- Arco Design

2.3、开发规范

- 整体参考clean-code-javascript
- 图片命名
  以模块划分命名，以下划线连接，例

  ```
  home_
  home_bg.png
  home_bg_1.png
  home_bg_2.png
  ```

  在js中引入时以驼峰命名：

  ```
  import homeBg1 from 'images/home_bg_1.png'
  import homeBanner2 from 'images/home_banner_1.png'
  ```

- 文档注释规范
  页面组件注释

  ```
  /**
   * @name 登录
   * @description
   */
  export default function Login() {
      /* todo */
  }
  ```

  函数注释

  ```
      /**
      * @description 提交登录表单
      * @param string email  ''
      * @param string password  ''
      * @return null
      */
  ```

```
function submit(email,password){
    /* todo */
}
```

三、APP及小程序

3.1、开发语言

- 使用uniapp及vue3框架进行开发

四、运维

4.1 文档要求

- 有完整的项目架构说明、安装、部署流程说明
- 有依赖第三方收费服务需要显著注明
- 有依赖闭包非开源组件请提前沟通

五、数据库

5.1、数据库类型

- 关系型数据库存储采用MySQL 5.7及以上版本
- Redis
- Mongodb [可选]
- Elasticsearch [可选]

5.2、字段约束

- 勿在数据库层使用字段强约束
- 多表数据操作请遵循ACID原则