**开心农场项目管理系统—设计开发**

**团队项目软件系统设计说明书**

**项目承担小组：软件工程“萌新入坑”小组**

**小组编号：202017**

**完成日期：2020.6.4**

**项目名称：开心农场**

**版次：1.0**

1.软件系统总体结构：

经过详细的讨论，我们的开心农场软件比较适合使用策略模式，策略模式是定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换，且算法的变化不会影响到使用算法的客户。

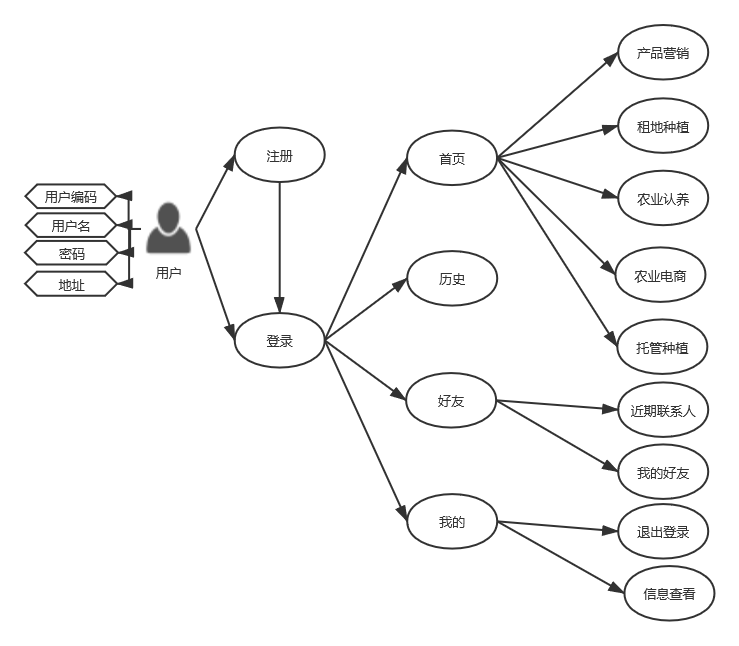
选择策论模式的理由：

(1)策略模式提供了管理相关的算法族的办法。策略类的等级结构定义了一个算法或行为族。恰当使用继承可以把公共的代码转移到父类里面，从而避免重复的代码。

(2)策略模式提供了可以替换继承关系的办法。继承可以处理多种算法或行为。如果不是用策略模式，那么使用算法或行为的环境类就可能会有一些子类，每一个子类提供一个不同的算法或行为。但是，这样一来算法或行为的使用者就和算法或行为本身混在一起。决定使用哪一种算法或采取哪一种行为的逻辑就和算法或行为的逻辑混合在一起，从而不可能再独立演化。继承使得动态改变算法或行为变得不可能。

(3)使用策略模式可以避免使用多重条件转移语句。多重转移语句不易维护，它把采取哪一种算法或采取哪一种行为的逻辑与算法或行为的逻辑混合在一起，统统列在一个多重转移语句里面，比使用继承的办法还要原始和落后。

2.系统数据库逻辑结构：



3.软件重用方案：

　首先我们讨论了关于什么代码应该重用？一方面是通用的，每个产品都能用到的代码，肯定要入库；另一方面是具有专业性的算法，比如图形算法等等，这种代码的通用性仅限于那个领域，不是所有产品都能用上的，但也要建立，因为他的技术含量高，避免二次反复开发。  
　　对于客户而言，我们是一家人，是软件的开发和提供方，所以公司的代码一般不分服务与客户，代码大家都能看到，如果任其自由复制修改应用，那么虽然也叫重用但没办法统一，也就无法管理了。所以我主张，公司的重用库一般情况下最好实现封装，普通项目开发人员一般也得不到它的代码。他们要做的就是上报可重用的代码、查询库中已有的代码、直接调用。即使测试过程中发现是重用库的Bug，也和开发人员没有关系，由管理重用代码库的组织来解决，找到代码的作者，由其修改，新库一发布，就行了。所以说公司内部建立这种请求－－服务的关系的话，各自的职责便明确化了。  
4.关键类：

　 本系统所有关键类如下所示：

普通用户类：定义了普通用户的基本属性

管理员类：定义了管理员的基本属性

用户功能类：定义与用户相关的功能实现基本方法

管理员管理类：定义与管理员相关的管理功能实现基本方法

用户控制类：控制用户的相关服务

管理员控制类：控制管理员的相关服务