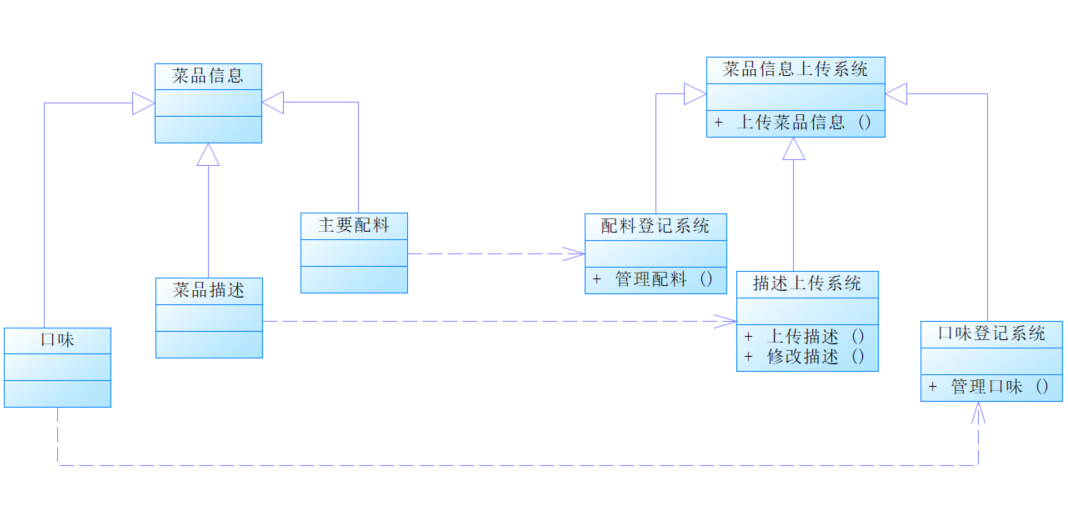
设计模式

1. 工厂方法模式

上传菜品涉及到产品信息管理系统的创建和添加新产品对象的工作。因此，可以使用工厂方法模式来实现产品信息管理系统的创建和管理。在该模式中，抽象产品类菜品信息类表示菜品对象，具有一些常见的操作，如获取描述和获取食谱等方法。抽象工厂类菜品上传系统表示工厂接口，用于创建产品对象。其子类上传描述，配料登记和上传口味信息各自实现了该工厂方法以创建不同的产品。

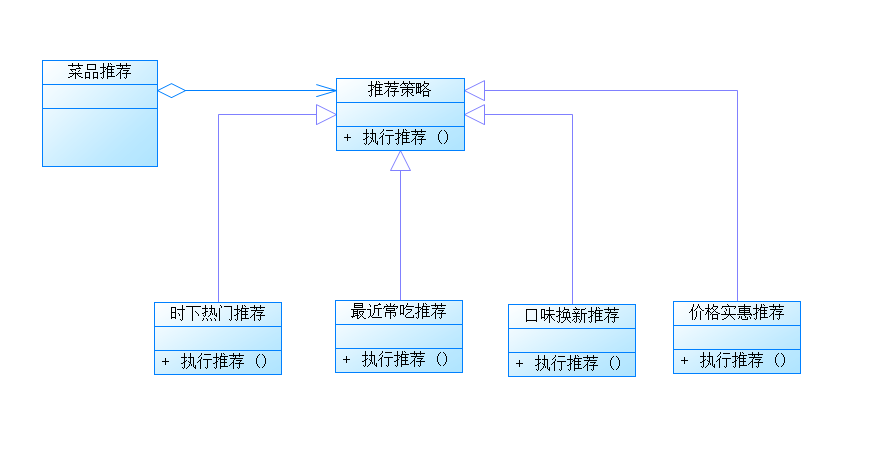


1. 策略模式

菜品推荐功能可以考虑使用策略模式来实现。策略模式允许在运行时动态地选择算法，并且能够将算法的实现与客户代码分离，从而提高代码复用性和维护性。

在该模式中，菜品推荐类表示上下文类，负责维护一个指向推荐策略对象的引用。而推荐策略则表示策略抽象类，定义了一个执行推荐()方法用于实现具体的算法。其子类时下热门、等则分别实现了具体的算法。

对于菜品推荐功能，可以将不同的推荐算法实现为不同的策略类，例如基于用户浏览历史的推荐算法、基于用户上传历史的推荐算法等。当需要执行菜品推荐功能时，客户端代码可以根据需要选择相应的策略，并将其传递给菜品推荐类进行执行。



1. 观察者模式

评论相关功能和社交相关功能经常需要进行状态更新和通知，因此可以使用观察者模式来实现这些功能。在该模式中，Subject表示被观察者类，具有attach、detach和notifyObservers等方法，其中attach和detach用于添加和移除观察者对象，notifyObservers用于通知所有的观察者对象进行状态更新。而Observer则表示观察者类，定义了update方法用于接收被观察者通知并进行处理。具体地，在大作业中，Subject可以表示评论管理系统或社交系统，Observer可以表示UI框架中的各个控件。

其中，Subject是用户类，定义了attach、detach和notifyObservers等方法。其子类用户评论类是具体实现的被观察者对象，包含一些状态信息，并能够修改这些状态信息。Observer是管理员类，定义了update方法用于接收被观察者通知并处理相应事件。