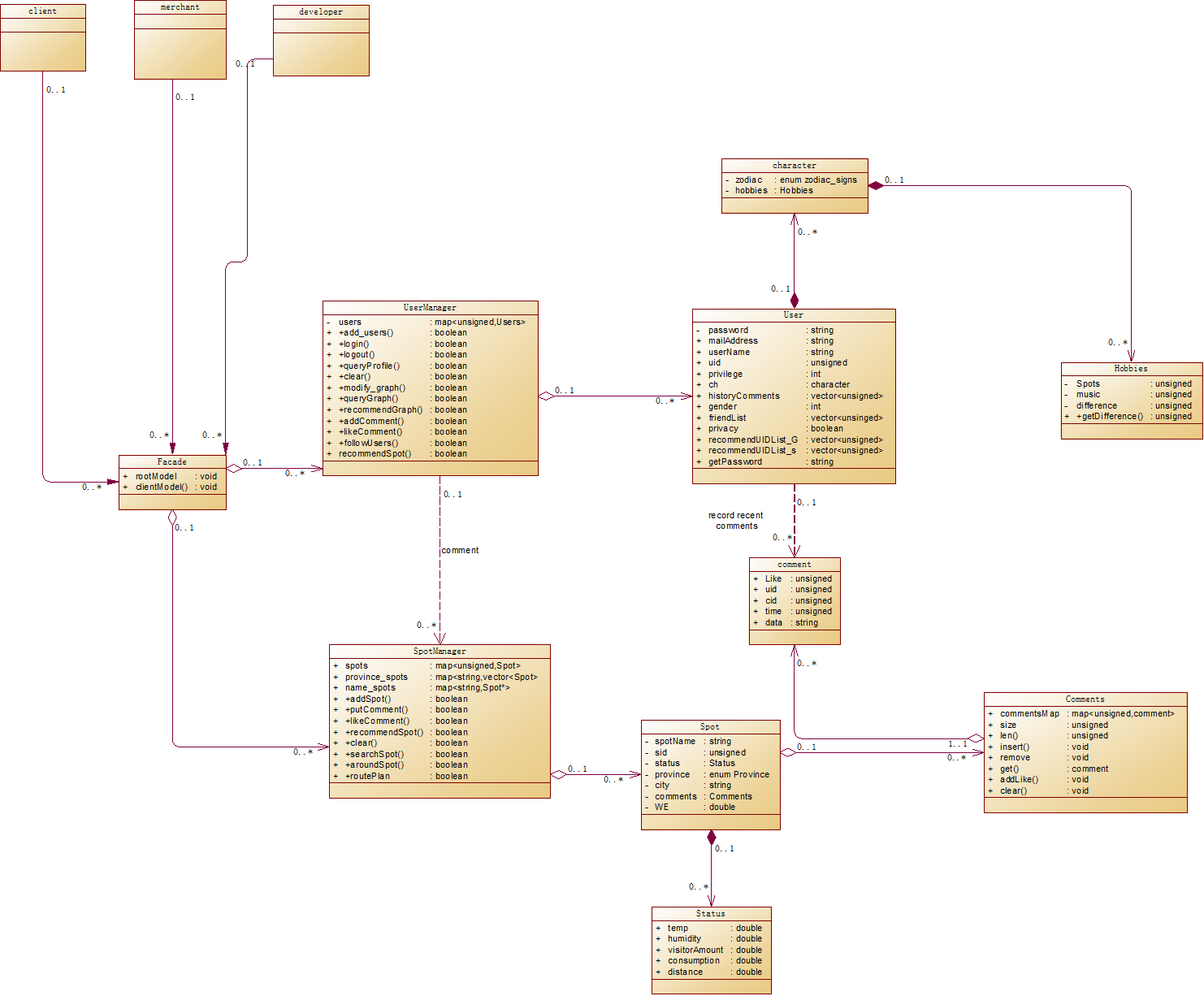
1. Façade：



User（用户类）：定义单个用户的所有信息，包含用户名、密码、邮箱地址、朋友列表、历史评论等。

character（用户特征类）：对用户的习惯信息进行维护的接口

Hobbies（用户习惯类）：定义用户的习惯信息

comment（用户评论类）：对用户的历史评论进行维护的接口

Comments（评论类）：定义单条评论的信息以及对评论进行维护的操作接口

Spot（景点类）：定义单个景点的所有信息，包含景点名、位置、省份、所有相关评论

Status（景点状态类）：定义景点的状态信息，包括湿度、游客流量、距离等

UserManager（用户管理类）：定义所有与用户相关的操作，提供了登录、登出、添加评论等操作的接口。

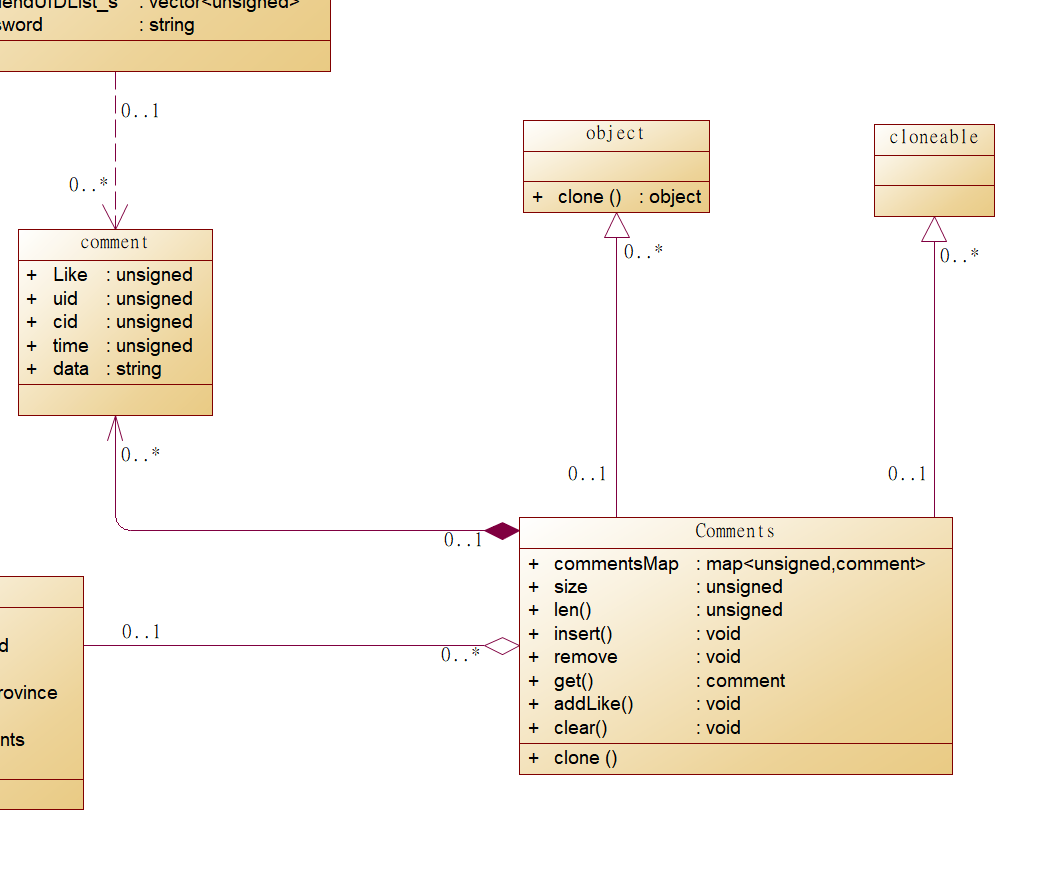
SpotManager（景点管理类）：定义所有与景点相关的操作，提供了添加景点、查找景点、周边景点推荐等操作的接口。

CommandParser（高级指令接口）：分别用一个UserManager和一个SpotManager来配置，是所有允许的操作的高级接口。例如，当用户发表评论时，façade将调用UserManager中的addCommend（）方法；当软件管理员添加新的景点时，façade将调用SpotManager中的addSpot（）方法。

Façade架构优势：

将用户相关操作和景点相关操作分别封装成不同的接口，再通过统一的高级接口来进行调用和维护，方便了用户和管理员直接使用相关接口对系统内部的信息进行操作，而不必关心系统架构和实现过程，同时也方便了软件开发人员对软件内部的相关功能进行更新和调试，而不影响外部的接口使用。

1. Prototype：



comment：用户评论。

object：用户评论集合的抽象类。

comments：用户评论集合。

Prototype架构优势：

使用原型架构来对用户评论进行管理。当用户添加新评论对象时，可以克隆用户的历史评论为当前评论对象，然后只需修改变化的部分。这提高了数据的复用率和软件的处理效率，并简化了用户的操作。