阅读教材6.1后，经过小组讨论，我们小组认为自己的工程设计方法为自顶向下与自底向上结合的设计方法。

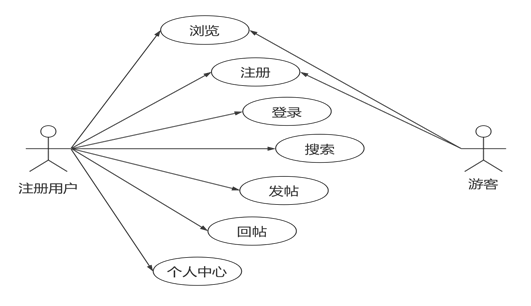
### 自顶向下：

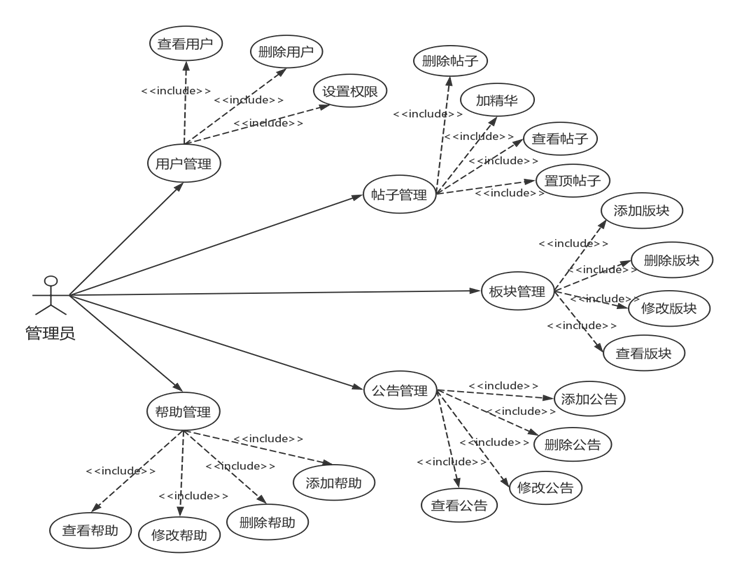
自顶向下设计是指根据给定的问题，递归地将给定问题分析成若干小的基本问题，自下而上是一种相反的设计方法，是指根据已有的组件（基本问题），通过一定的排列组合，最终解决需要解决的问题

在讨论项目主题的初期，我们小组对开发怎样的项目展开了激烈的探讨，最终决定,项目的主题定为论坛，一个专为全球计算机科学爱好者设计的交流论坛,这便是我们项目架构的“顶”。最初，我们项目的设计是自顶向下的，从确定的项目主题开始逐渐细化，从而先确定项目的整体框架，只有当框架确定下来，我们在今后的开发才有所依据。

主题的下一层，便是要实现的功能。根据我们的主题，我们将功能分类为登录、注册模块，个人中心模块，帖子模块，评论模块，搜索模块等，确定了工程大概的工作量和推进顺序，并且有助于接下来的分工和小组协作。

对于每一个功能，我们又讨论进行了设计，具体设计见需求规格说明书，这里只列出部分示例。





### 自顶向下：

采用自底向上的方法时, 往往先考虑系统的细节或其中一部分, 然后对多个部分进行抽象归纳合成一个大模块. 在此Web应用的开发过程中, 有很多这样的过程, 具体地, 模型层中Comment类的建立过程就是一个自底向上的过程.

Commet类的建立: 提到评论, 往往首先想到用户会添加评论(回帖), 其次是用户和用户之间在评论区交谈, 后者属于前者的子集, 因为评论本质上也是一个帖子, 这时我们会把这个处理过程和用户绑定在一起; 在开发过程中, 发现其他用户(甚至包括游客)都必须能看到评论, 只要该评论不是私有的, 这样上面的绑定就不可行了, 但是仍然可以通过和这个帖子关联的用户列表来访问评论; 再进一步, 帖子的预览应该显示评论数, 但是不需要显示内容, 这样如果还去查询就非常耗时间了, 此时评论数应该被升级为一个字段, 字段必须要有保存的地方, 此时不能确定保存位置, 但评论列表, 当前页数, 主题…都应该被升级成字段时, 应该建立Comment类来归纳这些字段和方法.

不仅仅限于Model层, 其他层也会有类似过程, 自底向上融合进了开发过程中.

