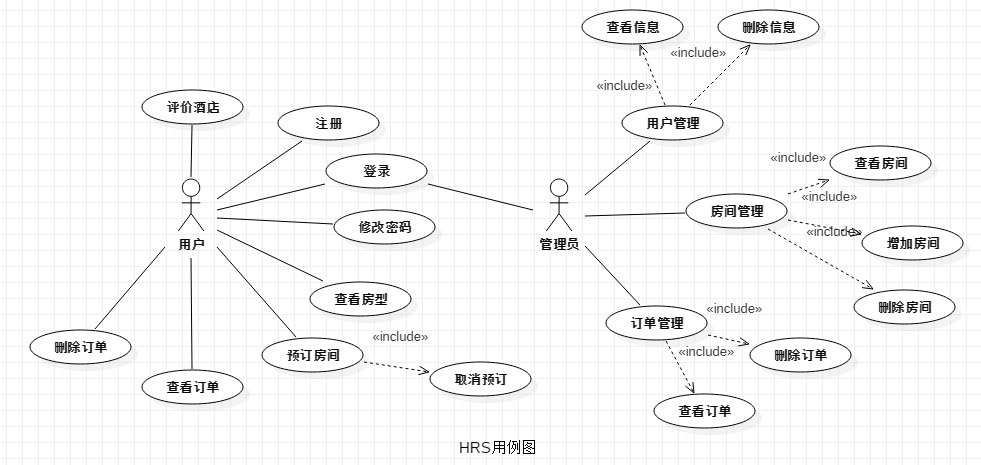
# HRS架构分析

* 酒店预订系统的需求分析：



通过以上用例图可清晰地得出该系统的业务需求（主要功能），如下：

1. 用户预订酒店，可取消。
2. 酒店管理员对用户进行管理——包括对用户信息的修改、用户的删除。
3. 酒店管理员处理预订订单。
4. 酒店管理员对酒店内房间进行管理。

* HRS具体质量场景：

1. 用户进行一个操作，发出一个请求，系统需在几秒内做出响应。
2. 用户浏览的始终是最新的房间信息。
3. 用户的数据不能被其它系统或软件窃取。
4. 用户预订房间，要能够方便快捷地操作。
5. 管理员获取订单要准确、最新。
6. 管理员处理订单时，简单快捷的操作。
7. 管理员添加房间能及时、及时地被用户获取。

* 选择架构驱动因素：
  1. 系统整个设计要简单、精要，方便使用者操作。
  2. 对于一个请求，要能够及时响应，不超过数秒。
  3. 数据不能被盗取，保证系统足够安全。
  4. 管理员进行的某些操作，用户端能够及时获取。
  5. 系统能正确实现目前所有功能，并能够退出新功能。
* 分析涉及的质量属性：

1. 性能。这个系统要求使用者的操作能够及时响应，并限制在几秒内，这就涉及了性能这个质量属性。
2. 易用性。系统完成之后，要保证使用者使用得称心，这就表明该系统要满足易用这个质量属性。
3. 可用性。系统的所有功能，均要能正常使用，不会掉线、死机，或出现其它异常情况，这使得对该系统的涉及，要满足可用性这个质量属性。
4. 安全性。系统中的数据不能被窃取，或者其它非法入侵能够阻止，这需要在涉及系统时，考虑安全性这个质量属性。
5. 互操作性。用户端和管理员端，这两者之间所做的更改，都要及时反映给对方，体现出两个不同模块之间的信息交互，需满足互操作性。
6. 可修改性。系统允许添加新的功能，就得保持其是可修改的，这涉及了系统的可修改性。
7. 可测试性。要能够保证系统正确地实现所有功能，不得不做的一个工作就是测试，所以这个架构需要考虑可测试性这个质量属性。

* 选择实现质量属性的战术：

1. 针对性能。采用提高计算效率、更有效地调度策略和引入并发的战术，来保证及时响应用户操作。
2. 针对易用性。采用分离用户接口和支持用户主动的战术来使得用户操作简单。
3. 针对可用性。采用心跳检测、回滚、异常抛出的战术，来实现系统的可用性。
4. 针对安全性。采用数据加密、验证身份、授权访问、恢复的战术，来保证系统的安全。
5. 针对互操作性。采用管理接口的战术来实现，用户和管理员之间的操作互动。
6. 针对可修改性。采用分离模块，增加接口的战术来实现系统的可修改性。
7. 针对可测试性。采用将接口与实现分离、内置监视器的战术来保证系统是可测试的，满足可测试性。