系统UML图

我们将系统**分为三层**，针对实际应用情况设计数据库系统--4s\_shop系统，划分出Car、CarBase、CarFactory、 Customer、List、Manager关系模式，对应于相应的Java类，类的数据访问模式设定为私有来保证安全性，对应于数据库的属性集，对象则对应于数据库系统的实例。利用ArrayList结构加在Database与应用程序时间保证数据的**安全性与处理效率**。

再Car、CarBase、CarFactory、Customer添加ID保证了对象的唯一性，建立数据库时需要考虑参照完整性保证数据库的合理性。

我们按照不同的汽车类型，定义了car\_类的**继承类**normal\_car，suv\_car。对于每一个的对象，我们定义set和get方法来保证可用性。数据集对应于UML的最底层，对于数据库的访问我们封装了**Dao接口**方法，利用类的功能函数调用Dao接口实现对数据库的操作，这样既提高了程序运行的效率，又减少了重复书写代码可能出现的bug。Dao接口处于UML图的第二层，功能函数处于第一层。

对于用户友好性，我们着重考虑了UI的制作，根据不同用户相应，对类的功能函数予以**封装**, 根据manager管理的不同对象,对功能函数实现了**多态**。对于系统安全性，我们采用了缓冲的方式，首先从数据库读出信息，置于ArrayList中，接下来基于ArrayList根据用户输入的信息与ArrayList中的信息进行比较，避免了SQL注入攻击。