**商城项目：**

**用户，订单，商品，购物车，分类**

**基础知识点：**

（笔试）编码规范

软件工程，UML建模，类图和时序图

JavaBean

JSP和HTML:四个作用域 九个对象

#默认会被缓存(jsp->java代码->.class)

Servlet

filter过滤器，编码过滤器

JSTL标签库和el表达式的基本用法${context}

web应用的基本目录结构 jsp / 静态资源

web的MVC思想和JavaWeb四层架构Model模型层/DAO层/Service层/Controller

mysql数据库的基本配置和基本操作，熟练掌握SQL语句编写包括DDL建库建表语句和CRUD语法，多表连接 join和 left join 和 right join

JDBC的基本操作 DBUtils做ORM映射，properties文件配置的读写

数据库事务，存储过程，索引，批处理，数据库安全(SQL注入)

网络编程HTTP协议

Tomcat7.0 web.xml , server.xml 热更新（类加载机制和JVM）

**HTML5**

**Jquery**

**BootStrap**

**AJAX：前台验证**

**中文乱码，数据批量插入，多表连接，**

**Junit单元测试 : ApacheBench压力测试**

JSON解析和XML解析

**使用Spring框架**

1.Spring的自动扫描功能，极大降低xml文件冗余，对大型应用效果显著。

2.使用Spring来帮我们管理事务，声明式可以不侵入的跟我们的原工程整合，而原工程不需要修改一句JAVA代码。

3.IOC

4.AOP

**使用SpringMVC框架，**

1.可以很容易跟手机端整合，通过导入jackson处理包，springmvc提供了与JSON对象之间的序列化和反序列化、

2.SpringMVC框架能很轻易的实现RESTFUL风格，利用@PathVariable实现通过url进行参数，而不是原始的参数传参方式，对用户更加友好

3.整合了上传和下载功能模块

4.天然的与Spring框架共存，而不需要担心系统整合问题

**Struts2**

1.注意你的action类需要设置为prototype原型，因为Struts2的ActionSupport本身是线程不安全。这也会导致Struts2的效率较低。

2.不要使用Struts2的标签库，有比较严重的性能问题。

**使用参数验证框架Hibernate-Validation，双重验证机制**

参数验证是WEB开发必不可少的部分，我们使用前后端双重验证来提升WEB网站的性能。后端我们使用Hibernate-Validation框架，因为他实现了JSR303验证规范，并且它提供了一直i18n国际化的关系来实现错误信息提醒。

前端我们采用了Jquery-Validation框架，因为前端我们整合了Jquery和它的一系列的UI框架，使用Jquery-Validation能很轻易的与jUqery还有Bootstrap整合，而不引入新的JS框架包，功能足够满足我们的需求。

**数据库相关**

**ORM框架**我们选择了Mybaits作为我们的ORM框架，因为Mybaits使用原生SQL的方式来处理数据库连接，相比Hibernate他在易用性上稍微欠缺，但是在我们自己实现方言类Dialect来实现分页效果等不同数据库差异较大的功能时，Mybatis更直观更容易优化的优点要比Hibernate重度封装更适合我们的系统，因为我们更关注于数据库的效率。

Mybaits既解决了JAVA对象和输入参数与结果集的映射，又能够让用户方便的手写SQL语句。

我们使用Spring来管理我的ORM框架，能很轻易的听过配置文件的方式实现数据库的切换(暂时支持Mysql Oracle SqlServer)

**数据库连接池**我们选择了阿里的druid的，相比其他经典的数据库连接池DBCP/C3P0，druid的数据库连接池了提供强大的连接池运行状态的监控和实时查询功能，并且具有高容错，高性能，开源，并且在淘宝商城经过了大量项目的实战，有阿里的背书更加可靠。

**缓存**系统我们选用Ehcache，这是当前最流行的系统级别缓存，纯JAVA实现，快速，精干，也是Hibernate的默认缓存提供类，支持多种缓存策略：内存和磁盘；

并且缓存数据会在虚拟机重启的过程中写入磁盘。

**分布式缓存** 我们选用Redis/【MemCache】，在日记统计等安全不敏感的数据，而且对性能要求较高，需要大量查询分析的数据，不再写入mysql数据库，而是采用nosql的方式。这也是业界主流的方法。

**权限管理/安全框架ApacheShiro**

CMS后台需要细粒度的权限管理机制，具体到按钮那一级别，这种粒度的权限管理用传统的AOP过滤器拦截url路径的方式显得不是那么方便。我们选用ApacheShiro安全框架来实现细粒度的权限管理机制，Shiro本身较为复杂功能强大，但相比SpringSecurity起来，他本身的逻辑清晰，并且后期还能进一步整合CAS单点登录模块。

我们利用Shiro实现安全编码，用户提交的所有数据都进行加密，避免用户信息泄露。登录用户密码进行SHA1散列加密，此加密方法是不可逆的。保证密文泄露后的安全问题。

权限认证：系统对所有管理端链接都进行用户身份权限验证，防止用户直接通过URL进行未授权页面。并且我么所有的JSP页面统一放在/WEB-INF/views路径下，避免用户直接用过全路径访问。shrio还提供了前台权限显示机制，对于不满足的按钮直接进行屏蔽，按用户操作不会迷惑 混乱。

**静态网页模板Velocity**

我们选用Velocity作为我们的静态网页模板，跟Freemarker相比他的语法简单更加接近于Java的语法，跟thymyleaf相比他的渲染效率要高于许多，淘宝也选用了Velocity作为他的模板引擎，因为他的页面复杂，并且Velocity自身没有封装很多标签，使用上更加灵活。

通过SpringMVC整合Velocity可以很轻易与原来的JSP模板共存。

**任务调度Quartaz**

**单元测试Junit**

使用单元测试和spring框架整合，用于在测试数据库增删改查操作而不会影响原有的代码。

**布局框架SiteMesh**

**全文搜索：Apache Lucene /SOLR**

**搭配中文分词IKAnalyse / PAODING(庖丁解牛)**

**日记模块：Slf4j/log4j**

记录日记 以便日后排查错误

**工作流引擎：Activiti**

**第三方工具类:**

通用工具包Apache-Commons ： commons-io / commons-lang / commons-fileupload

通用工具包Google Guava

日期处理类JodaTime

文档处理Apache POI

**WEB服务器:**

Tomcat7.0

Tomcat是一个WEB服务器和Servlet/jsp容器，了解web.xml的基本配置以及用server.xml配置外部应用

Ningx

前台框架：

\* JS框架：JQuery 1.9<br>

\* CSS框架：Twitter Bootstrap 2.3.1。

\* 客户端验证：JQuery Validation Plugin 1.11。

\* 树结构控件：jQuery zTree / Jquery EasyUI

\* 数据表格：jqGrid

\* 日期控件： My97DatePicker

\*

富文本：CKEditor

UWebEditor政府和国企项目更喜欢这个富文本，因为它更接近于DOC的语法，功能更加强大也更笨重，但政府的中年大妈更习惯这种方式。

而CKeditor是全球最流行的富文本插件，语法更接近于更高效通用的HTML语法。

\* 文件管理：CKFinder

\* 动态页签：Jerichotab

\* 手机端框架：Jingle

\* 对话框：jQuery jBox

\* 下拉选择框：jQuery Select2<br>

**Maven**

管理和发布我们的工程

**GIT/CVS/svn**

多人协作工具，防止代码互相覆盖