## Web的整体认知架构:

硬件服务器

Web 服务器

Web 应用程序框架

Web应用程序架构MVC

## Web服务器端整体实现架构可行性方案

**可选方案一：J2EE UNIX/windows+Tomcat+Orecle+Java**

**web应用程序开发语言: Java**

机会:

1. Java的应用范围更加的宽广。可以开发桌面程序，手机游戏、DVD播放器及应用服务器等。Java与某些软件有着天然的内在联系，如Database，ERP等。
2. 开发人员不可能只有做网页而已。一旦开始做大量分析或与其它系统整合，需要涉及到底层的数据分析时，光用PHP之类的东西就很难了，则需 要采用Java代码来完成。
3. 在大型、 复杂的企业级应用领域，Java仍是主流的开发工具，此外，Java在手机平台上越来越受到欢迎。据悉，目前全球有600多款手机运行Java虚拟机，全 球10大手机游戏中有7款是采用Java开发。同时，Java已经成为多数科学研究和计算使用的首选语言。在大学中有越来越多的人在使用Java语言，越 来越多的尖端学术研究完全使用Java语言；各个行业的公司都有很多的Java程序员在开发他们的项目。

优势

1. Java有很多的开源组件与软件可以利用
2. Java有大量的开源社区支持，Apache的Portfolio或SourceForge
3. 现在正在学习java知识，与学校教学接轨

劣势:

1. 框架笨重
2. 知识点多
3. 开发工具过重
4. 不适合快速快发
5. 微调后需要重新部署

威胁:

其他解释型语言如PHP等能进行快速开发且开发工具轻量

市场少很少用java做web page

PHP天生用来做web，不用找外部库

**Web框架: SSM: Spring+SpringMVC+MyBaitis**

机会:

1、市场上不少公司使用该框架

优势:

1. Spring-MVC的配置文件相对来说较为少，容易上手，可以加快软件开发的速度
2. spring mvc的执行效率高
3. spring mvc是spring中的一个模块，所以spring对于spring mvc的控制器管理更加简单方便，而且提供了全 注解方式进行管理，各种功能的注解都比较全面，使用简单
4. springmvc 比较简单，很较少的时间都能上手
5. MyBatis可以进行更为细致的SQL优化，可以减少查询字段。
6. mybatis是一个jdbc框架，配合spring方便
7. mybatis提供了对应各种用途、功能的插件

缺点:

1. MyBatis的数据库移植性不好，不同的数据库需要写不同SQL
2. SQL语句的编写工作量较大，对开发人员编写SQL语句的功底有一定要求
3. spring的文档是英文的

威胁:

1. hibernate成熟，市场推广率高
2. Hibernate 封装性好，屏蔽了数据库差异，自动生成SQL语句
3. Hibernate数据库移植性很好
4. Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便

**可选方案二：LAMP: Linux+Apache+MySQL+Python**

**组员因素、谁来掌握、python不熟悉，学习成本**

**web应用程序开发语言: Python**

优势:

1. 由于Python是一种解释型的脚本语言，开发效率高，所以非常适合用来做Web开发
2. Python有上百种Web开发框架，有很多成熟的模板技术，选择Python开发Web应用，不但开发效率高，而且运行速度快
3. Python部署都很快，变更起来很容易，代码量小
4. Python的标准库和第三方库强大

劣势:

1. Python并不是一门快语言
2. 向后兼容性太差

机遇:

1. Python在运维、大数据、云计算、web、科学计算上优势明显
2. 目前业内几乎所有大中型互联网企业都在使用Python，如：Youtube、Dropbox、BT、Quora(中国知乎)、豆瓣、知乎、Google、Yahoo!、Facebook、NASA、百度、腾讯、汽车之家、美团等。

威胁:

1. 任何可以运行Java虚拟机的电脑或者移动设备都可以运行Java的应用，而不管在哪里运行Python的程序都需要一个编译器来将Python代码转化为特定的操作系统可理解的代码

**Web框架: Django**

**优势**

1. django提供了很多很方便的工具，对很多东西也做了封装
2. django提供的是一套设计模式，按照这个思路去做可以很省力
3. django的官方文档相当详尽
4. Django的社区是最活跃的，在StackOverflow上有80000个相关问题和大量的博客和强大的用户
5. Django是MVC模式
6. Django自带ORM和模板引擎很好用

机会:

1、大部分的岗位需求是web后台和运维系统开发的，web后台又有大部分是要求django的

劣势:

1. Django没有引导用户去打包他们的应用
2. Django是一款重量级框架
3. Django都是面向大型应用的
4. Django自带ORM和模板引擎，灵活和自由度不够高

威胁:

1. Flask自由、灵活，可扩展性强，第三方库的选择面广，开发时可以结合自己最喜欢用的轮子，也能结合最流行最强大的Python库
2. Flask入门简单，即便没有多少web开发经验，也能很快做出网站
3. Flask非常适用于小型网站
4. Flask与关系型数据库的配合使用不弱于Django，而其与NoSQL数据库的配合远远优于Django

## Web前端框架可行性方案

## 可选方案一: Vue

优势

1. 易用，模板式开发，标准HTML元素内只是多了一些属性
2. 采用virtual DOM，每当应用的状态被改变时，不需要重新渲染整个组件树
3. 轻量级的框架，开发周期短，适合小型应用程序
4. 官方文档很清晰
5. 组件化应用，应用分拆成一个个功能明确的模块
6. Vue的核心团队维护配套框架

劣势

1. 没有稳定的社区组件
2. 运行时错误描述不够直观
3. Vue主要是由一位开发者进行维护的，而不像React一样由如Facebook这类大公司维护

机会:

1、Vue.js是2017年发展最快的JS框架之一

威胁

1. 除饿了么，暂时没有大的应用使用

## 可选方案一: React

优势:

1. Virtual DOM
2. 组件化应用，应用分拆成一个个功能明确的模块
3. React由Facebook维护
4. 状态管理适合大型应用

劣势:

1. JSX
2. 每当应用的状态被改变时，全部子组件都会重新渲染
3. React的配套框架是由社区成员维护

机会:

1、React 框架都是最受欢迎也是最成熟的

威胁:

1. 受到Vue.JS的挑战

Oracle

Oracle 在兼容性、可移植性、可联结性、高生产率上、开放性 Oracle产品采用标准SQL