**一、moodle平台基础架构**

平台采用**阿里云服务器**+阿里云**OSS云存储服务**（基于CDN技术）的形式

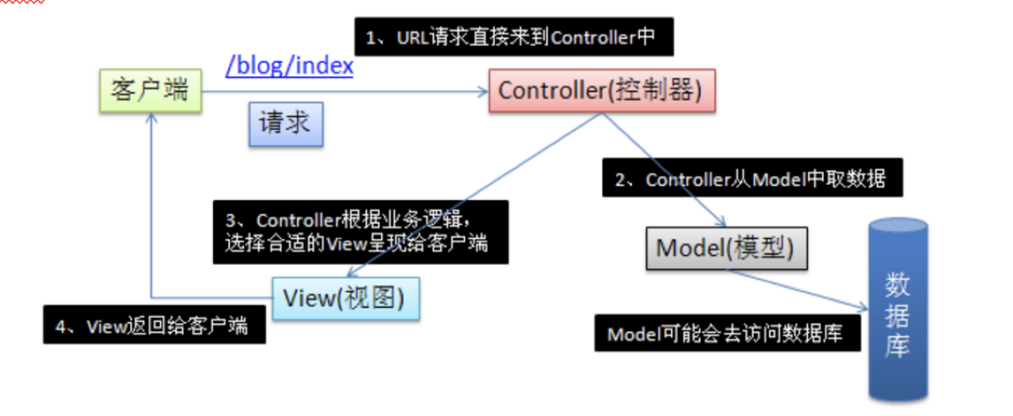
**网站**以及**数据库**主要部署在**ECS阿里云服务器**上，视频等**静态资源**都部署在阿里云的**oss云存储服务**上，具体结构如下图所示：



**1、网站后台采用的技术**

网站后台采用spring体系下的mvc技术架构（spring+springMVC+Mybatis）进行开发，数据库为的mysql数据库（详细代码以及数据表说明见文档第二部分）。

**mvc基本原理**参考网上，大体原理为：controller层负责接收request请求，然后根据请求中的参数，将请求交给相应的service层，service层根据请求参数处理具体业务以及与dal层交互访问数据库数据，dal层负责直接访问mysql数据库，之后controller将service层处理完的数据以response的形式返回给用户浏览器



**2、网站前端采用的技术**

技术采用：html+css+js+jQuery

基本作用为通过请求获得相应的页面服务，另一方面，通过js采集用户的各种行为操作，并通过jquery的ajax实时的将行为信息发送给后台服务器，服务器将这些数据简单处理后存入mysql数据库，供算法模块分析使用

**3、后台OSS服务相关**

项目中主要用OSS服务来存储学生看的学习视频，现在主要存了北大、浙大和国防科大的全套软工视频，包括完整章节视频和切割过的视频（按知识点进行的切割），具体视频你可以登录到oss上看一下，离线的全部视频要是需要的话，你可以问**袁晓芳**要（她现在还住在大兴）。

**4、后台服务器相关环境参数、账号、密码等（已失效-此部分只做参考）**

<1>ECS服务器相关参数、账号、密码

**登录地址、账号及密码：**

**39.107.98.156**

**8080端口**

**账号admin密码ruanWei17**

\***相关环境参数：**

ECS上安装的系统为：ubuntu14.04

编程环境：java\_1.8+Tomcat 7

（其中的端口号是可以通过配置文件进行更改的）

<2>ECS上部署的mysql的相关账号密码

账号root 密码ruanWei17

<3>oss登录以及账号密码

***登录地址：***

*https://oss.console.aliyun.com/index#/bucket/oss-cn-qingdao/software-moodle/object/?path=bW9vZGxlX3ZlZGlvLw%3D%3D&keyword=*

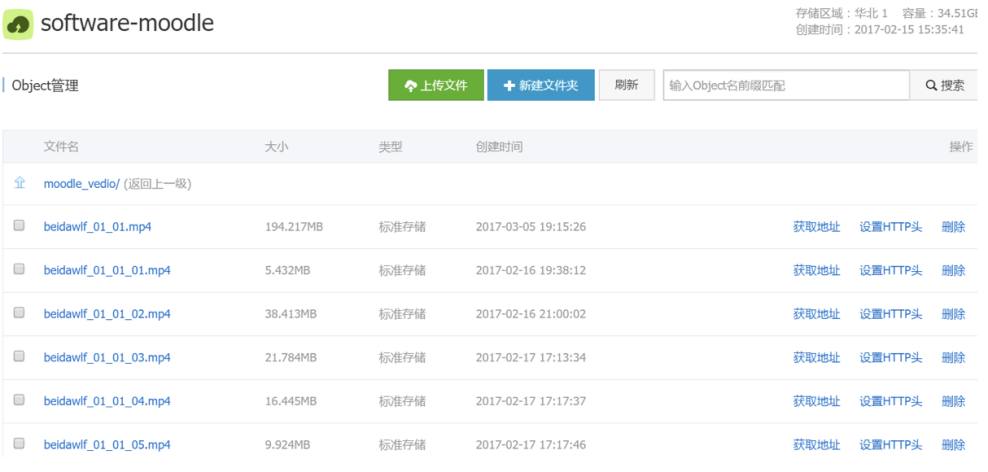
***阿里云oss服务账号：****fishplusflag*

***密码：****pkuss\_moodle*

***安全邮箱：****[pkuss\_moodle@163.com](mailto:pkuss_moodle@163.com)*

***安全邮箱账号****：pkuss\_moodle@163.com*

***密码：****pkussmoodle*



**二、代码结构以及数据表说明**

**1、代码结构说明**

解压moodle.zip，并将工程导入到javaEE版本的Eclipse里面，并配置部署好调试用的Tomcat和工程所需的Maven（详细配置方法见网上资料）。

基本的代码结构为：

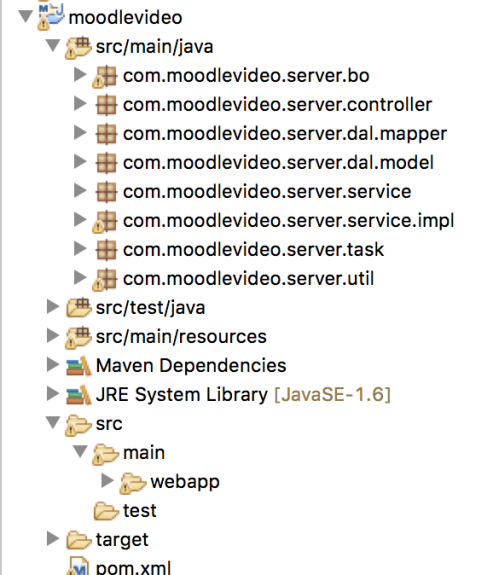


图2-1

其中最下面的pom.xml为工程的maven配置文件，有什么需要的依赖包或者插件从这里进行管理部署。

1. 图2-2中的webapp文件夹下是主要的前端代码文件以及工程核心文件，其中的web.xml和rest-servlet.xml是主要的spring（以及springMVC）配置文件（详细配置方法见网上资料）

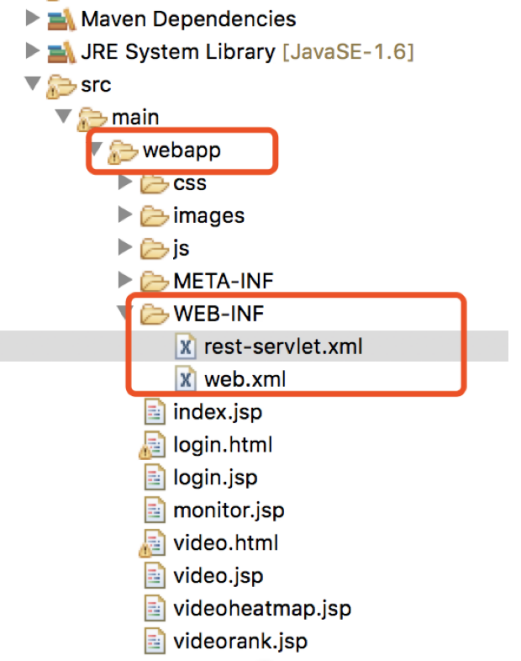


图2-2

1. 这个resouces文件夹下主要为spring的一些java\_bean的注入配置文件，spring的一些配置子文件，以及主要的mybatis配置文件。

（具体配置文件的阅读需要结合网上相关参考资料进行）

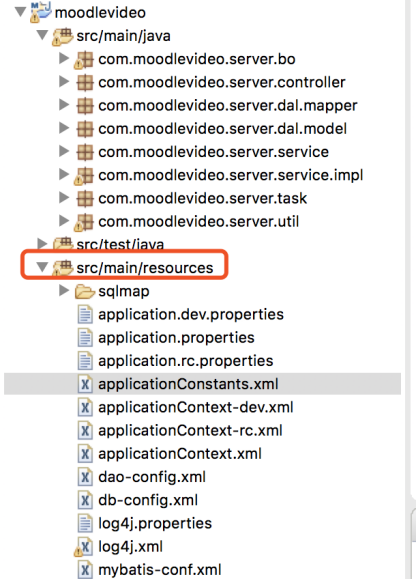


图2-3

1. 主要工程业务代码为图2-4

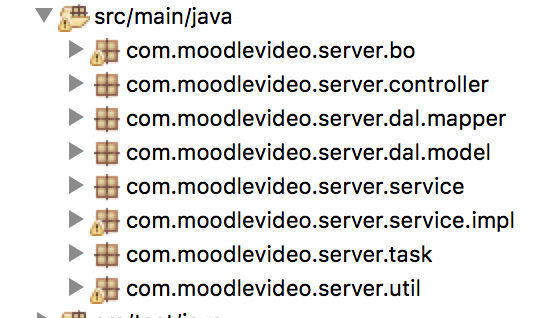


图2-4

其中

<1> controller文件夹下为控制层代码（接request，返回response）；

<2> service以及service.impl文件夹下为具体的业务层代码，主要的业务处理以及与数据层打交道的代码都在业务层（controller将请求参数交给service层，service层处理完之后再将controller需要的response返回码以及数据返回给controller层）；

<3> mapper、modle以及前面的resources文件夹下的sqlmap共三个文件夹下的代码共同构成数据层代码（都是与数据库打交道的mybaits相关代码），这部分很多公共代码可以借助mybatis-generator来自动生成（具体方法可参考网上资料），并在生成的代码基础上结合业务需求进行修改；

<4> task文件夹下主要为一些定时任务代码（一定周期自动执行的代码），例如算学生每周观看视频时长的代码就在这里（每天中午定时算一次）；

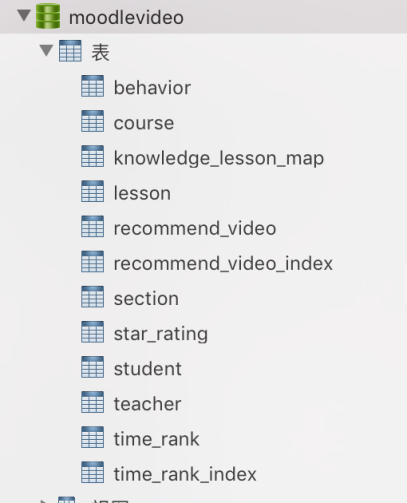
<5> bo文件夹主要是一些辅助用的业务对象的代码（bo即business object，具体概念参考网上资料）

<6> util文件夹下为一些静态方法的代码，例如json转化代码、返回码相关代码等，理解mvc主要结构原理后，这一部分代码简单阅读以下就能明白

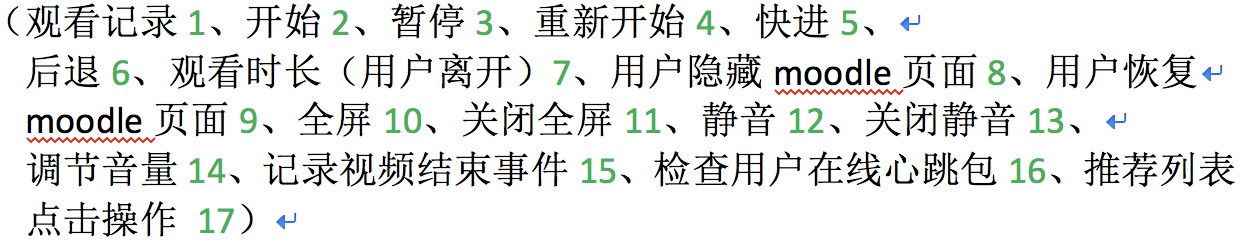
**2、数据表相关说明**

可借助三方可视化数据库工具进行查看管理（这里用的是**Navicat**）

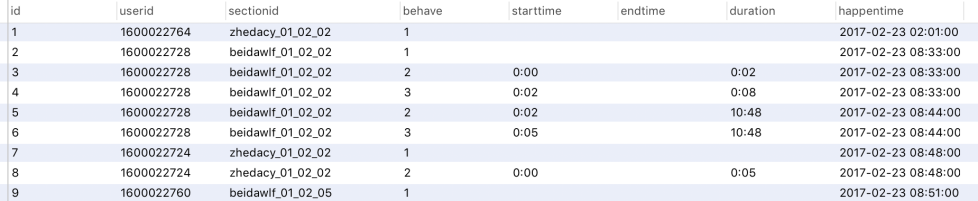
（1）表说明，所有表如下图所示：



<1> behavior表为用户行为记录表,一共有17中行为：



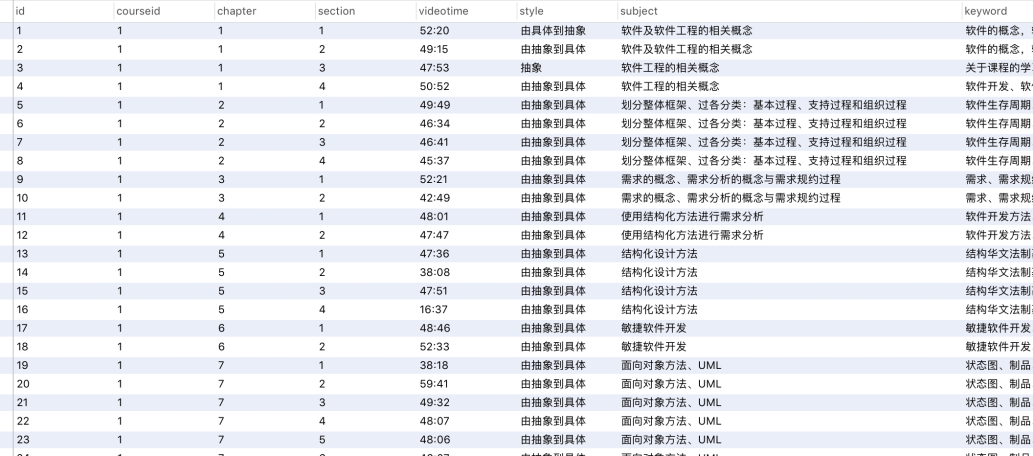
（其中，sectionid为对应的视频编号，behave为行为类型，duration为从打开页面到行为发生时持续的时间，happentime为行为发生时的标准时间，starttime为行为发生时视频播放的具体时刻，endtime为部分行为结束时对应的视频播放时刻）



<2> course表记录共有哪些老师的课程



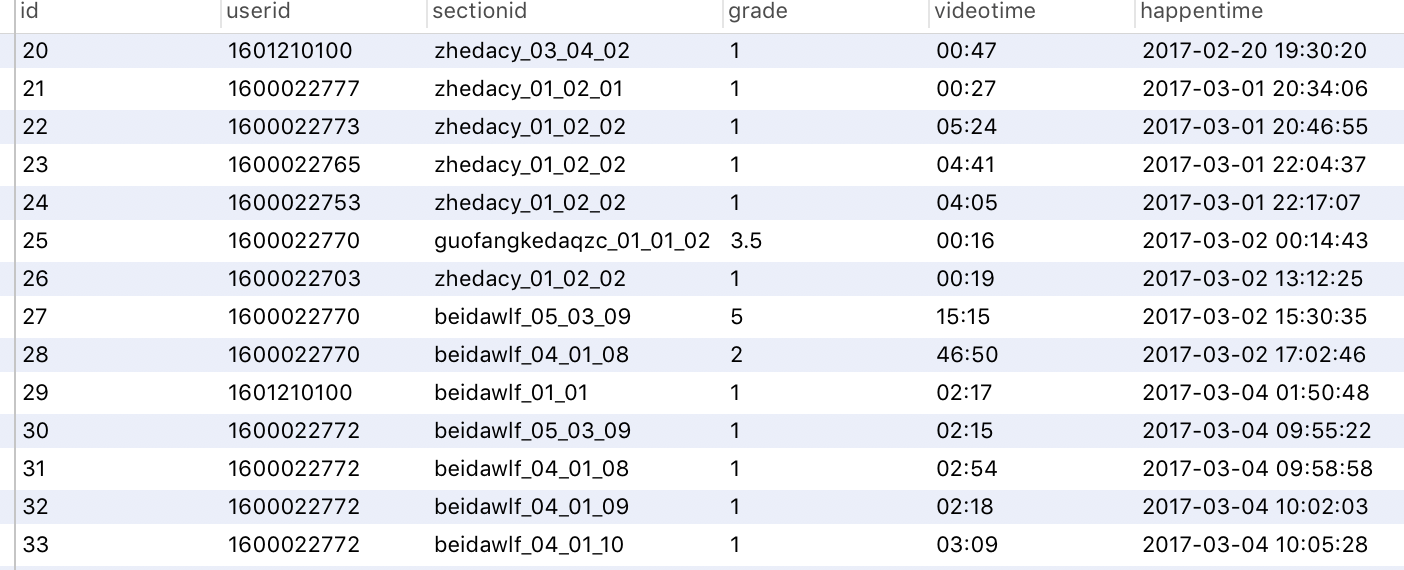
<3> lesson表记录具体的课程信息（按章节）



<4> section表记录按知识点切割后的视频信息，lessonid为章节信息



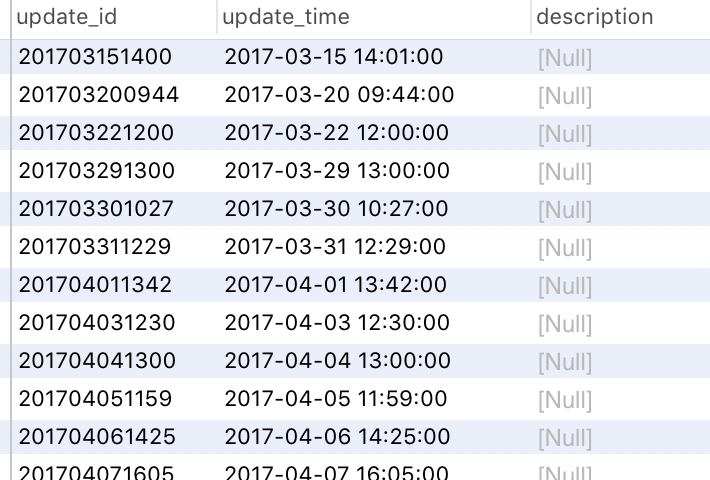
<5> star\_rating表记录对视频的打分记录信息



<6> student表和teacher表分别为学生表和老师表的详细信息

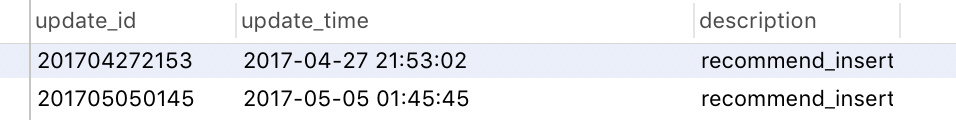
<7> time\_rank表和time\_rank\_index表分别为学生学习观看时间排名表（每一条是每天算出的一周观看时间以及排名）和排名计算程序的执行记录表（每周期执行一次程序就记录一条）





（8）recommend\_video和recommend\_video\_index表分别为所有用户的推荐结果记录表和每次执行推荐程序的程序执行记录表



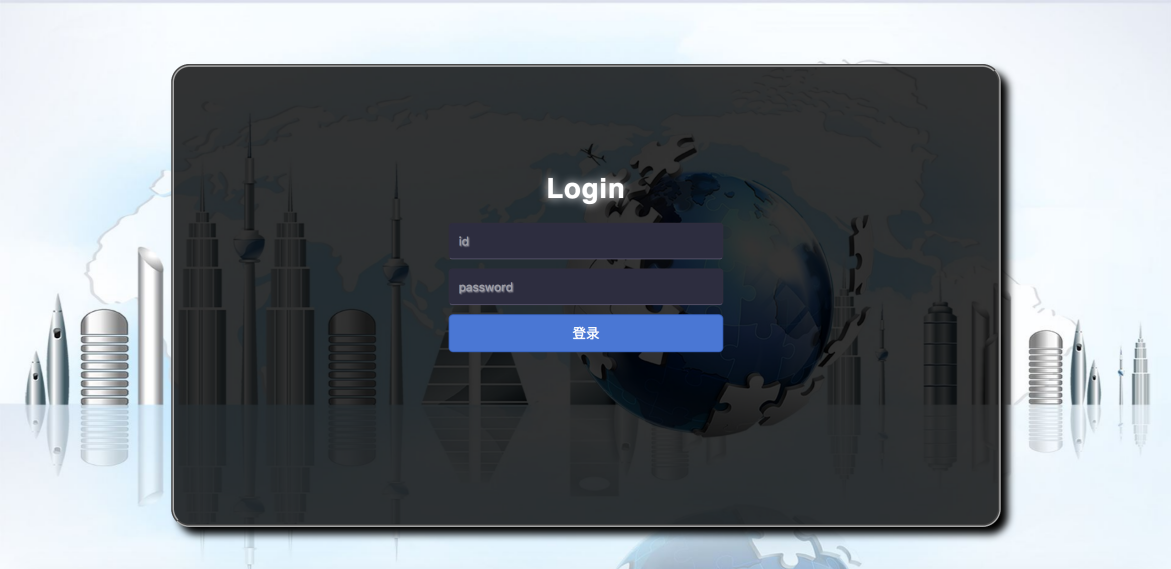


**三、系统界面说明**

**测试页面地址:**

http://139.224.228.113:8057/moodlevideo/videojob/section?sectionid=beidawlf\_01\_01

**1、登录界面**



可用测试账号：1601210101

**2、主界面**



标号<1>处为用户的姓名、学号，点击进去可查看学生的学习进度以及相关热力图；

标号<2>点击后可查看用户本周的视频观看排名；

标号<3>点击后返回moodle平台

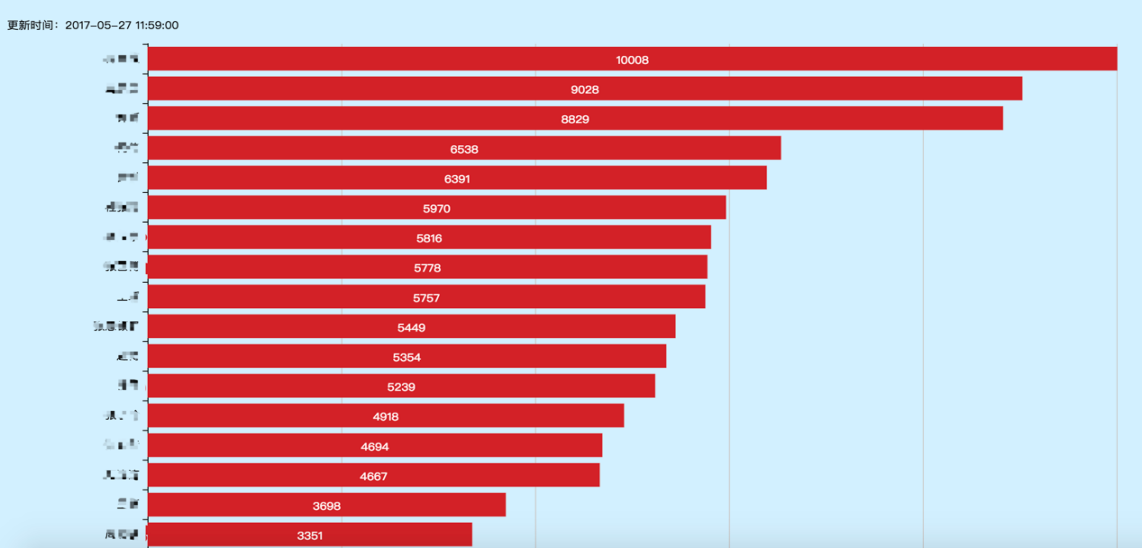
标号<4>为视频推荐列表

标号<5>为视频评分

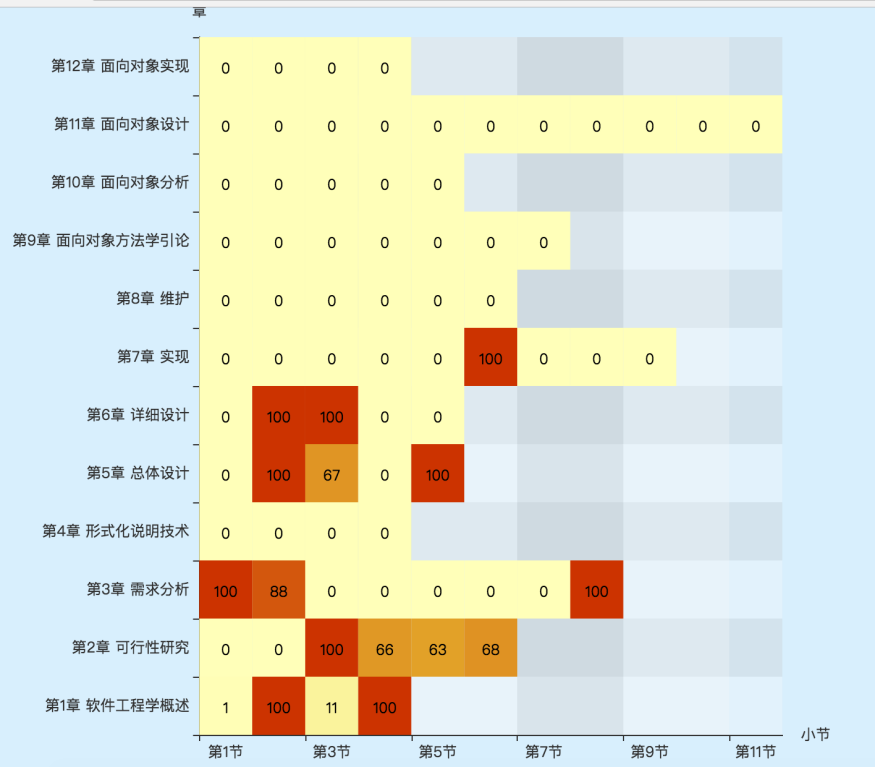
标号<6>为课程简介

**3、其他相关数据展示界面**

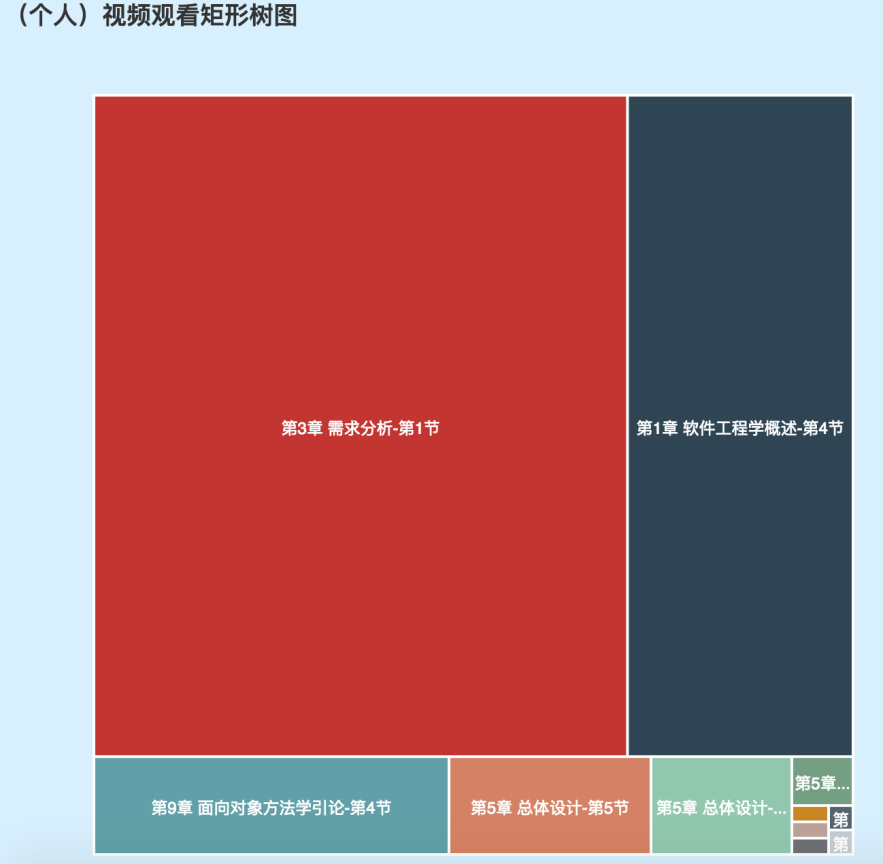
（1）学生每周的视频观看时间排行榜（根据用户行为数据计算得到）



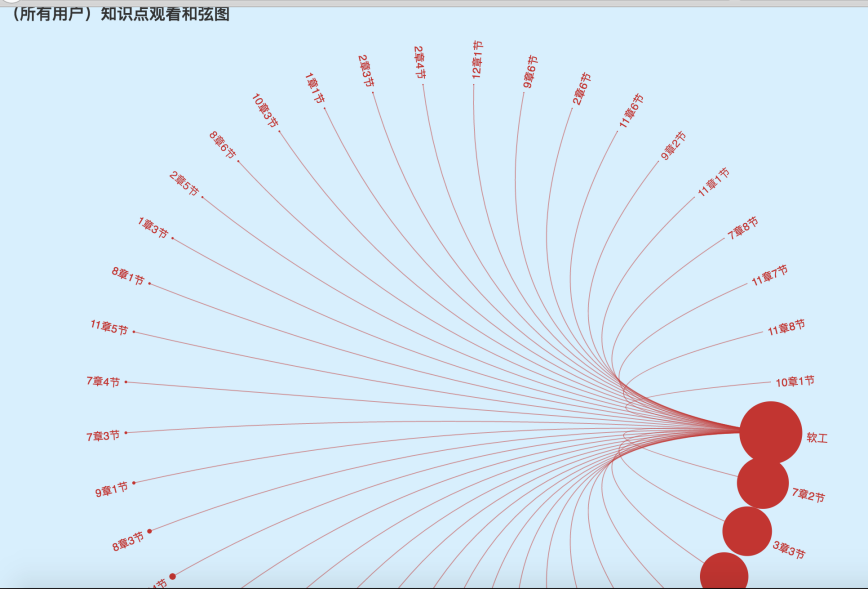
（2）按每个章节知识点的完成度来统计的，学生观看的视频热力图，0为没看过，100为学习时间已达到相应知识点的学习视频总长度（根据用户行为数据计算得到）



（3）按（个人）章节知识点的观看时间长短来统计的，观看的时间越长，相应的矩形就越大（根据用户行为数据计算得到）



（4）按（所有用户）章节知识点的观看时间长短来统计的，所有用户看的最多的视频，相应的圆圈就越大（根据用户行为数据计算得到）



Application.properties设置jdbc相关属性

参考<https://www.cnblogs.com/yokoboy/archive/2013/03/01/2939315.html>

Mapper（创建函数？）引用model（定义） 问题：他们之间是怎样的关系？以及sqlmap

<https://www.cnblogs.com/selene/p/4607004.html>

timerank以及recommendation需要用python写

**为网站添加数据库功能**：首先在dao-config.xml里配置bean，然后在dal中建立相应的数据库文件，具体格式参考文件夹内的其他文件，然后在sqlmap内编写sql文件，也就是命令。之后在controller内编写调用功能，编写方法可参考相应的功能函数，然后跟踪函数查看函数的编写，该部分还会涉及到service部分的编写，格式同样参考相应文件的格式。

相关数据流：1、在jsp文件中通过ajax的post向后台提交2、在controller中request post的内容，然后在此编写需求3、mapper以及model设计数据库，以及数据库所要进行的操作4、service中声明controller所涉及到的函数5、Impl内定义service中的函数6、在sqlmap中编写数据库操作

注意：每个Mapper，也就是dao文件必须在dao的配置文件中进行配置，否则无法生成bean

**新增数据库表格**

Dailytime为学生每天观看时长，每半小时更新一次，时间格式000000代表00小时00分00秒

Chapter\_index为章节关系表，chapter为当前章节，chapter之间没有先后关系，follow\_chapter为后续章节。Follow\_chapter内的章节通过“+”隔开，“+”之间没有先后关系。

Subchapter\_index为章节内的子目录之间的关系表，chapter为所在章节，subchapter为当前子章节，subchapter之间没有先后关系，follow\_subchapter为后续子章节，后续子章节之间没有先后关系，通过“+”连接，parallel\_subchapter为与当前子章节并行的章节，即有关系，但是可以不考虑先后。

推荐课程时的调用说明：1、首先根据学生观看的视频片段在section表中查找对应的lessonid号（这个号码对应了课本中的相应章节）2、根据相应算法推荐出学生应该看的课程章节。在这个过程内，先通过chapter\_index查找章节之间的关系，然后根据这个关系去对应的章节内查找子章节，在subchapter\_index表内查找对应章节的子章节之间的关系3、得到应该推荐的章节后，在section表内根据lessonid号推荐对应的视频id