**天津工业大学**

**计算机科学与技术学院**

**2016级六周实训报告**

项 目 名 称:基于SSM的“球探”足球论坛

专　业　班　级: 中软国际—2016级卓越班

指导教师签名： 李伟

实　训　日期：2019.1.2－2019.3.8

合作培养企业：中软国际

姓 名：吴思远

一、引言

## 1.1项目概述

项目目的：

为广大足球爱好者提供一个平台：比分查看、新闻先知、球员近况、趣闻盘点、球迷交流… 足球迷想要的一切，我们在努力打造，期待营造国内领先的专业便捷和谐健康的球迷交流环境

项目背景：

开发项目名称：球探论坛

国内足球粉一直是一个庞大的集体，尤其近几年群体基数大幅增长，“足球热”成为一种风向，“国足”“中超”“C罗梅西”等等成为人们饭后谈资。经过研究发现，现国内足球论坛繁杂众多，质量参差不齐，很难快速找到一个查看比赛近况、球迷发帖交流、比对球员信息、查询积分情况等功能于一体的地方。球探论坛就是为了满足广大球迷的要求而设计的，不仅可以查看近期比赛信息、球队信息，而且可以以帖子方式进行交流，致力于成为国内足球迷的不二家园。

参考资料：企业提供的素材（录屏、课堂笔记、源代码）

FootballFanCast.com

Soccer Blogger

Rayleague

## 1.2 需求分析

## 论坛(Forums) 是在线商业服务中的团体组织。论坛可能运作着一个图书馆、一个聊天室，让人们进行实时的信息交流，甚至它还有一个

## 分类的公告目录。公司可以决策参加或者资助因特网上的论坛，它将会吸引一些带有特殊兴趣的群体。美国在线(America Online) 以其

## 拥有14000 个聊天室而引以为豪。最近，它还引入了一个“朋友表”，当有朋友在线时，它会提醒其成员，以使他们能够交换信息。 [1]

## 2017年10月1日起，国家互联网论坛社区服务提供者应当按照身份实名制、电话认证后注册账号，并对版块发起者和管理者严格实施真实身份

## 信息备案、定期核验等

## 1.3 运行环境

软件环境：

编译器：Eclipse   
项目构建工具：Maven   
数据库：MySQL、Redis  
JDK版本：jdk11.0.2  
Tomcat版本：springboot2.1.2内置

版本管理：github

硬件环境：PC机，CPU 1G Hz以上，内存2G字节，硬盘空间10G处理器：100M Hz以上 ROM+RAM：32M以上

# 二、项目设计

## 2.1 设计思路

概要设计：前端 HBuilder(layui)

后台 MySQL + navicat + eclipse(ssm)

手机端（短信验证码，以及二维码的扫描）

详细设计：

前端界面：

1.注册登陆

2.首页

·做一个轮播图

·最近的比赛情况

·热帖

·球队分类

·积分榜

3.\*\*\*帖子详情页

·上传图片，文字

·实现对贴子的回复和引用

·点赞，赞赏，举报，

·查看他人信息

|

|

4.个人信息页（包括自己和他人）

等级，在线时长，浏览量，好友，最近访问，发消息，最近动态

5.充值界面

包括自己金币和赞赏人的金币的变化

后台数据库：

1.用户表Users（id,name,pwd--varchar,money--int,avatar(头像)--role--varchar，bio(自我介绍)--text,

sightml(个性签名)--varchar,gender(性别)--varchar,pv(访问量)）

2.联赛Leagues（id,name）

3.球队Teams(id,name,leagueId)

\*\*4.球队人员分布Members（id,name,role--varchar(前锋中场后卫门将),num(编号),teamId）

\*\*5.帖子表Posts（id,title,context(text),leagueId,visitId,userId，images(图片)）

6.浏览量表Visits（id,pv(访问量),comment(评论量),like(点赞数)）

-->需要根据访问量进行排序，选出前几个进行展示

7.收藏表Collectors（id,name,userId）

8.关注表Focususers(id,name,userId)

\*\*9.积分表Integrals(id,name,session(场次),win,equal,lose,goal(进球),fumble(失球)，GD(净胜球),score积分,leagueId)

10.评论表Comments(id,name,title,avatar,content)

## 2.2 模块功能介绍

功能：

注册|登陆 （可能手机号验证，或者是扫码登陆）

--> 选择自己喜欢的球队（比如说意甲，西甲，目前只涉及足球板块）

--> 论坛首页（包含我关注的和全部的，热帖，搜索功能，各种分类的条目）

--> 查看帖子（包含楼主和评论，发帖需要上传图片，GIF，文字，网址等等）

--> 浏览量（点赞数，回复数，浏览量）

--> 回帖和评论（赞赏，转发，举报，收藏）

--> 转到充值和朋友圈

个人信息的管理：包括注册时间；发帖条数；关注的好友；

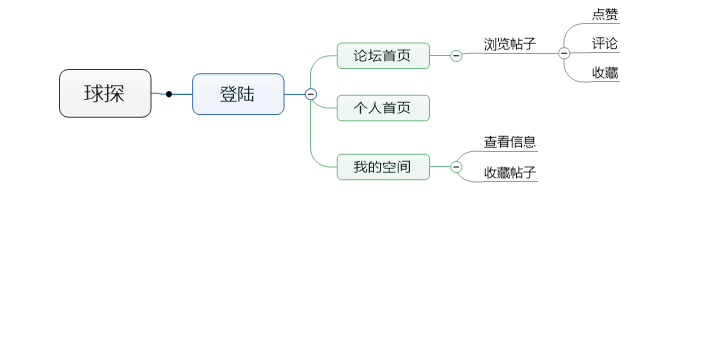
-->查看别人的信息，关注，（互关就类似于好友了）

异常的处理（提高安全等级，密码修改）

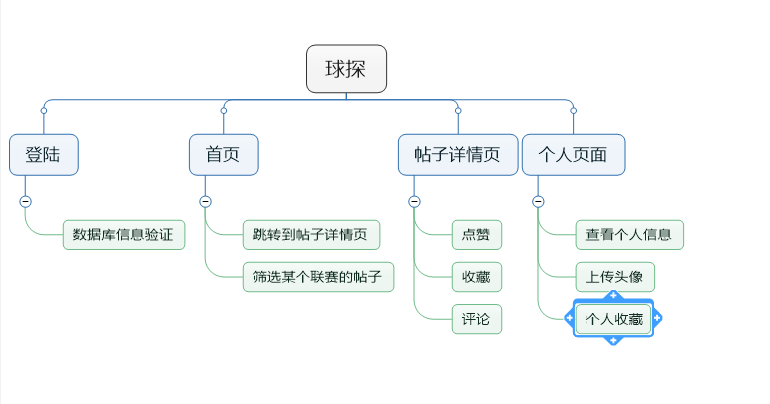
特点：

最主要的是实现信息的实时性，让用户可以及时的浏览最近的足球信息。同时构建球迷交流平台，对于时事热点信息网友可以交流评论。

## 2.3 模块结构图



## 2.4 程序流程图



## 2.5功能设计分工

第一阶段分工：

宗新淇：在github上完成项目的创建，可以让组员随时提交项目;

熟悉Spring MVC;

尽量在前端界面完成的情况下，把数据库的数据同步到View层

扈朋华：熟悉Rose软件，完成用例图，时序图，类图

申一凡：将球队信息实时的插入到数据库中，通过部分数据的插入测试数据库的完备性

王英才：社区首页(主要是轮播图以及下面的比赛时间，最上方的个人信息索引栏)

-->点击更多-->

吴思远|屈林珊 社区详情页(左侧的导航栏，中间的热帖以及右侧的分类卡)

整体项目分工：

宗新淇：主要负责ssm的框架构建，实现后台的数据和前端页面的交互

第一周：先将数据库的数据显示在控制台上，然后在显示在前端页面的相应位置

第二周：将前端输入的数据加载到数据库中，并实现浏览量，点赞等的相应变化

第三周：通过用例测试发现问题，然后解决，在对某些功能实现完善

扈朋华：文档的提交和用例图的设计以及测试

第一周：每日文档的提交和熟悉mindmanager,rose等软件

第二周：完成用例图，类图，完成junit测试

第三周：通过测试发现问题，并在解决后完成最后所有的文档并提交

申一凡：数据库

第一周：搜集近期比赛的帖子和数据，并输入到数据库中

第二周：检验前端传入的数据是否符合要求，并与其他连接表进行比对，如果出现问题，进行分析并改正

第三周：时间允许的话，增加其他的功能，填入数据完善数据库

王英才：

吴思远|屈林珊：前端

第一周：完成帖子详情页和分类页面的设计

第二周：完成个人信息页面和充值页面

第三周：完善其他功能并实现对应页面的设计

# 三、详细设计

详细设计：

前端界面：

1.注册登陆

2.首页

·做一个轮播图

·最近的比赛情况

·热帖

·球队分类

·积分榜

3.\*\*\*帖子详情页

·上传图片，文字

·实现对贴子的回复和引用

·点赞，赞赏，举报，

·查看他人信息

|

|

4.个人信息页（包括自己和他人）

等级，在线时长，浏览量，好友，最近访问，发消息，最近动态

5.充值界面

包括自己金币和赞赏人的金币的变化

后台数据库：

1.用户表Users（id,name,pwd--varchar,money--int,avatar(头像)--role--varchar，bio(自我介绍)--text,

sightml(个性签名)--varchar,gender(性别)--varchar,pv(访问量)）

2.联赛Leagues（id,name）

3.球队Teams(id,name,leagueId)

\*\*4.球队人员分布Members（id,name,role--varchar(前锋中场后卫门将),num(编号),teamId）

\*\*5.帖子表Posts（id,title,context(text),leagueId,visitId,userId，images(图片)）

6.浏览量表Visits（id,pv(访问量),comment(评论量),like(点赞数)）

-->需要根据访问量进行排序，选出前几个进行展示

7.收藏表Collectors（id,name,userId）

8.关注表Focususers(id,name,userId)

\*\*9.积分表Integrals(id,name,session(场次),win,equal,lose,goal(进球),fumble(失球)，GD(净胜球),score积分,leagueId)

10.评论表Comments(id,name,title,avatar,content)

外键：

fk\_Posts\_Visits(visitId)

fk\_Users\_Posts(userId)

fk\_Users\_FocusUsers(userId)

fk\_Users\_Collectors(userId)

fk\_Leagues\_Posts(leagueId)

fk\_Leagues\_Teams(leagueId)

fk\_Leagues\_Integrals(leagueId)

fk\_Teams\_Members(teamId)

1：1{

Posts Visits √

}

1:n{

Users Posts √

Us详细设计：

前端界面：

1.注册登陆

2.首页

·做一个轮播图

·最近的比赛情况

·热帖

·球队分类

·积分榜

3.\*\*\*帖子详情页

·上传图片，文字

·实现对贴子的回复和引用

·点赞，赞赏，举报，

·查看他人信息

|

|

4.个人信息页（包括自己和他人）

等级，在线时长，浏览量，好友，最近访问，发消息，最近动态

5.充值界面

包括自己金币和赞赏人的金币的变化

后台数据库：

1.用户表Users（id,name,pwd--varchar,money--int,avatar(头像)--role--varchar，bio(自我介绍)--text,

sightml(个性签名)--varchar,gender(性别)--varchar,pv(访问量)）

2.联赛Leagues（id,name）

3.球队Teams(id,name,leagueId)

\*\*4.球队人员分布Members（id,name,role--varchar(前锋中场后卫门将),num(编号),teamId）

\*\*5.帖子表Posts（id,title,context(text),leagueId,visitId,userId，images(图片)）

6.浏览量表Visits（id,pv(访问量),comment(评论量),like(点赞数)）

-->需要根据访问量进行排序，选出前几个进行展示

7.收藏表Collectors（id,name,userId）

8.关注表Focususers(id,name,userId)

\*\*9.积分表Integrals(id,name,session(场次),win,equal,lose,goal(进球),fumble(失球)，GD(净胜球),score积分,leagueId)

10.评论表Comments(id,name,title,avatar,content)

外键：

fk\_Posts\_Visits(visitId)

fk\_Users\_Posts(userId)

fk\_Users\_FocusUsers(userId)

fk\_Users\_Collectors(userId)

fk\_Leagues\_Posts(leagueId)

fk\_Leagues\_Teams(leagueId)

fk\_Leagues\_Integrals(leagueId)

fk\_Teams\_Members(teamId)

1：1{

Posts Visits √

}

1:n{

Users Posts √

Users FocusUsers √

Users Collectors √

Leagues Posts √

Leagues Teams √

Leagues Integrals √

Teams Members √

}

Po实体类

User.java(id,name,pwd,money,avatar,bio,sightml,gender,pv)

League.java(id,name)

Team.java(id,name,leagueId)

Member.java(id,name,role,num,teamId)

Post.java(id,title,context,leagueId,visitId,userId)

Visit.java(id,pv,comment,like)

Collector.java(id,name,userId)

Focususer.java(id,name,userId)

Integral.java(id,name,session,win,equal,lose,goal,fumble,score,leagueId)

与实体类对应的Dao、xml文件

dao:

UserDao

LeagueDao

TeamDao

MemberDao

PostDao

VisitDao

CollectorDao

FocususerDao

IntegralDao

spring/mapper:

UserDao.xml

LeagueDao.xml

TeamDao.xml

MemberDao.xml

PostDao.xml

VisitDao.xml

CollectorDao.xml

FocususerDao.xml

IntegralDao.xml

Service:

MemberService

UserService

Controller:

web.controller

TestController

4.代码的实现

可能遇到的问题和难点：

1.首先在github上创建一个项目，实现各个成员分工的提交

2.数据库数据的插入和定期的更新，数据量可能较大，比较耗费时间

3.一般的登陆注册应该可以一两天完成，需要完善的就是界面的美观和与数据库的连接<!-- 能不能实现在密码错误三次后，完成

找回密码的操作-->

4.首页的难点是轮播图和导航栏，以及右侧类似于卡片的分区

5.Spring MVC的构建，难点除了环境的配置以及映射之外，就是前后台数据的交互

--根据(浏览量表Visits)的排序实现帖子按热度呈现

--根据(联赛Leagues,球队Teams)实现对帖子的筛选|查询操作

--通过首页中呈现的题目，从数据库中找到对应的

帖子表Posts（id,title,context(text),leagueId，visitId，userId）将context中包含的文件上传到网页，

实现网页的跳转（帖子详情页）

6.帖子详情页要将回复动态的加到对应的{帖子表Posts}中，点赞（小手按钮）同步到数据库

赞赏-->转移到赞赏界面-->显示余额-->金币不够-->充值界面

查看他人信息-->通过 Inner Join(用户表Users,帖子表Posts)查看最近动态

收藏帖子会自动添加到个人界面(Inner Join(收藏表Collectors和帖子表Posts))

（+关注）Inner Join(关注表FocusUsers,用户表Users)显示好友name

5.Junit测试（其中要涉及到测试用例，基本要求是通过登陆不同的用户测试以上的各个功能）

6.ers FocusUsers √

Users Collectors √

Leagues Posts √

Leagues Teams √

Leagues Integrals √

Teams Members √

}

Po实体类

User.java(id,name,pwd,money,avatar,bio,sightml,gender,pv)

League.java(id,name)

Team.java(id,name,leagueId)

Member.java(id,name,role,num,teamId)

Post.java(id,title,context,leagueId,visitId,userId)

Visit.java(id,pv,comment,like)

Collector.java(id,name,userId)

Focususer.java(id,name,userId)

Integral.java(id,name,session,win,equal,lose,goal,fumble,score,leagueId)

与实体类对应的Dao、xml文件

dao:

UserDao

LeagueDao

TeamDao

MemberDao

PostDao

VisitDao

CollectorDao

FocususerDao

IntegralDao

spring/mapper:

UserDao.xml

LeagueDao.xml

TeamDao.xml

MemberDao.xml

PostDao.xml

VisitDao.xml

CollectorDao.xml

FocususerDao.xml

IntegralDao.xml

Service:

MemberService

UserService

Controller:

web.controller

TestController

4.代码的实现

可能遇到的问题和难点：

1.首先在github上创建一个项目，实现各个成员分工的提交

2.数据库数据的插入和定期的更新，数据量可能较大，比较耗费时间

3.一般的登陆注册应该可以一两天完成，需要完善的就是界面的美观和与数据库的连接<!-- 能不能实现在密码错误三次后，完成

找回密码的操作-->

4.首页的难点是轮播图和导航栏，以及右侧类似于卡片的分区

5.Spring MVC的构建，难点除了环境的配置以及映射之外，就是前后台数据的交互

--根据(浏览量表Visits)的排序实现帖子按热度呈现

--根据(联赛Leagues,球队Teams)实现对帖子的筛选|查询操作

--通过首页中呈现的题目，从数据库中找到对应的

帖子表Posts（id,title,context(text),leagueId，visitId，userId）将context中包含的文件上传到网页，

实现网页的跳转（帖子详情页）

6.帖子详情页要将回复动态的加到对应的{帖子表Posts}中，点赞（小手按钮）同步到数据库

赞赏-->转移到赞赏界面-->显示余额-->金币不够-->充值界面

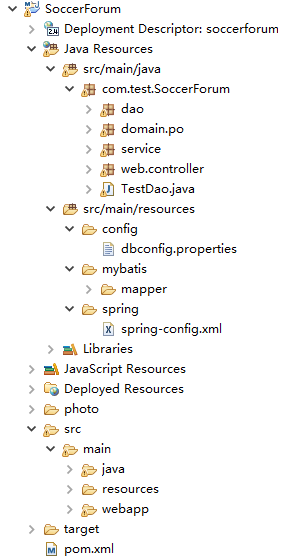
查看他人信息-->通过 Inner Join(用户表Users,帖子表Posts)查看最近动态

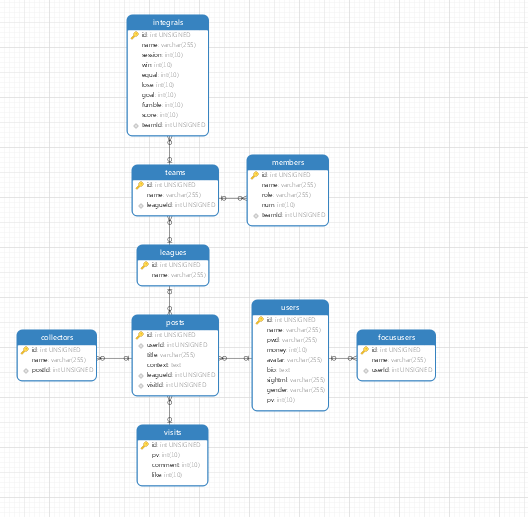
收藏帖子会自动添加到个人界面(Inner Join(收藏表Collectors和帖子表Posts))

（+关注）Inner Join(关注表FocusUsers,用户表Users)显示好友name

5.Junit测试（其中要涉及到测试用例，基本要求是通过登陆不同的用户测试以上的各个功能）

6.





# 实现方案

1.登陆页面



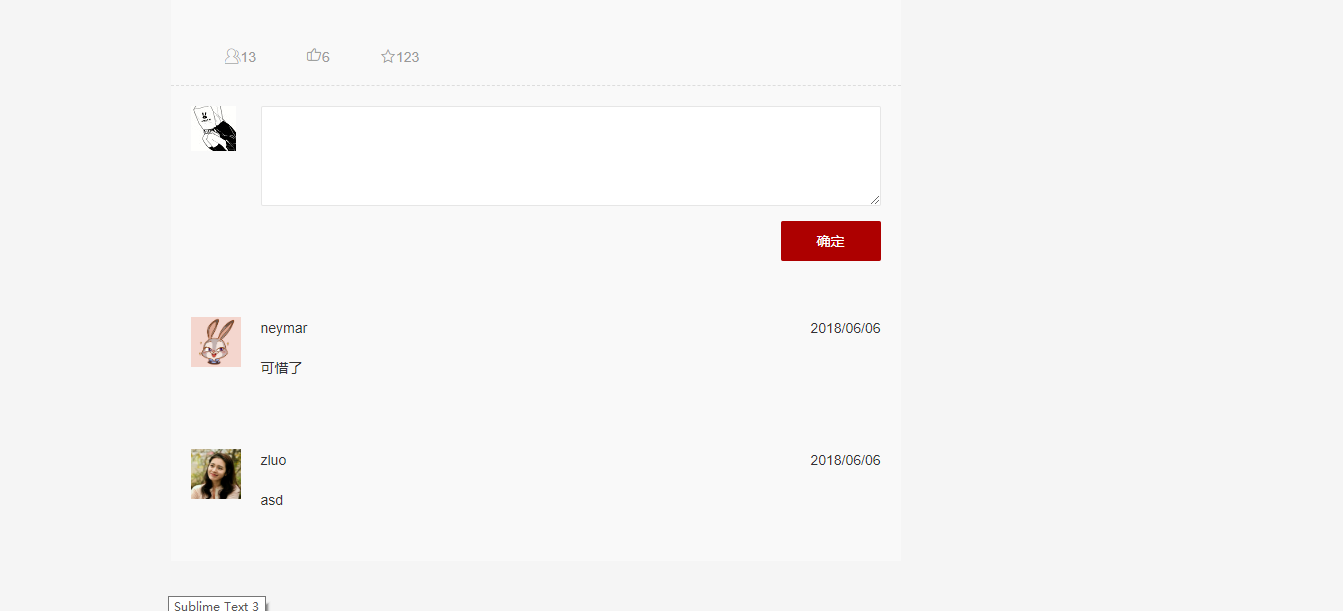
2.首页



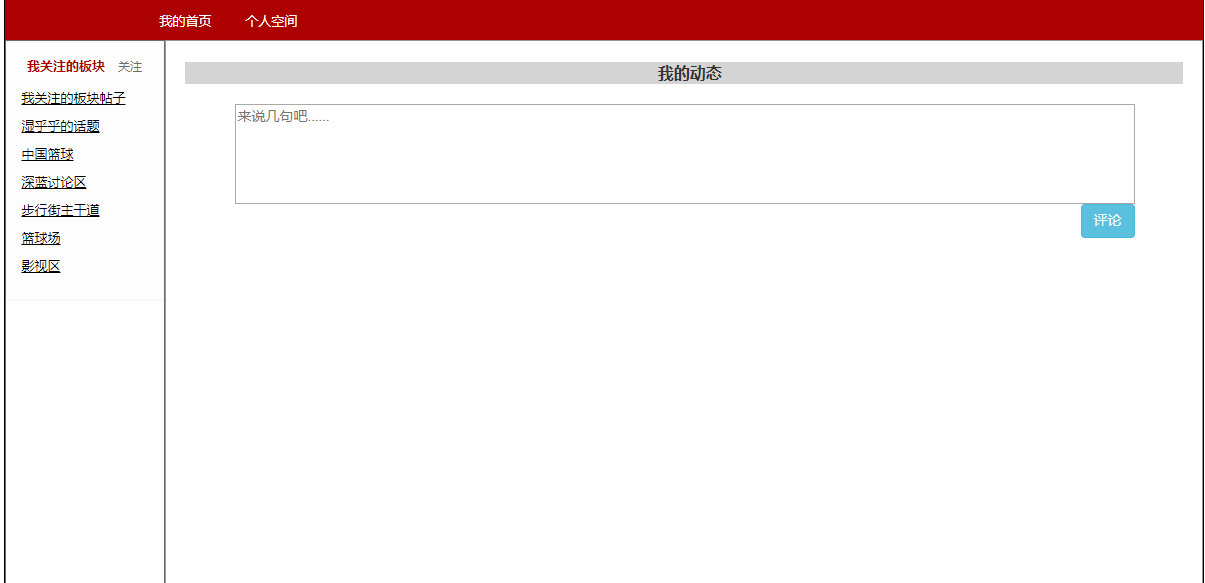
3.帖子详情页



4.评论，收藏，点赞



5.个人页面



6.我的空间



# 五、系统测试

测试方法：Junit测试

