



基于React的超级账号教学支持系统 前端的设计与实现

学号 MF1632080

姓名 邬文怀

导师 任桐炜



答辩提纲



- 研究概述
- 需求分析
- 概要设计与课程活动模块的设计
- 课程活动模块的实现
- 工作总结与展望



项目背景



- 网络教育
- 教学支持系统的出现和发展
- 日渐复杂的教学需求
- 超级账号教学支持系统



教学支持系统的发展现状



■ Sakai

- 插件结构，定制化功能
- 强调自主、协作的学习模式，更适用于国外

■ Moodle

- 功能丰富：课程、权限、作业
- 页面结构不合理，缺乏导航信息

■ NJU CMS

- Moodle的定制版
- 与软件学院现有教学模式无法完全契合



日渐复杂的教学需求



- 教学资源的管理与共享
- 课程活动的记录
- 小组活动
- 师生互动



答辩提纲



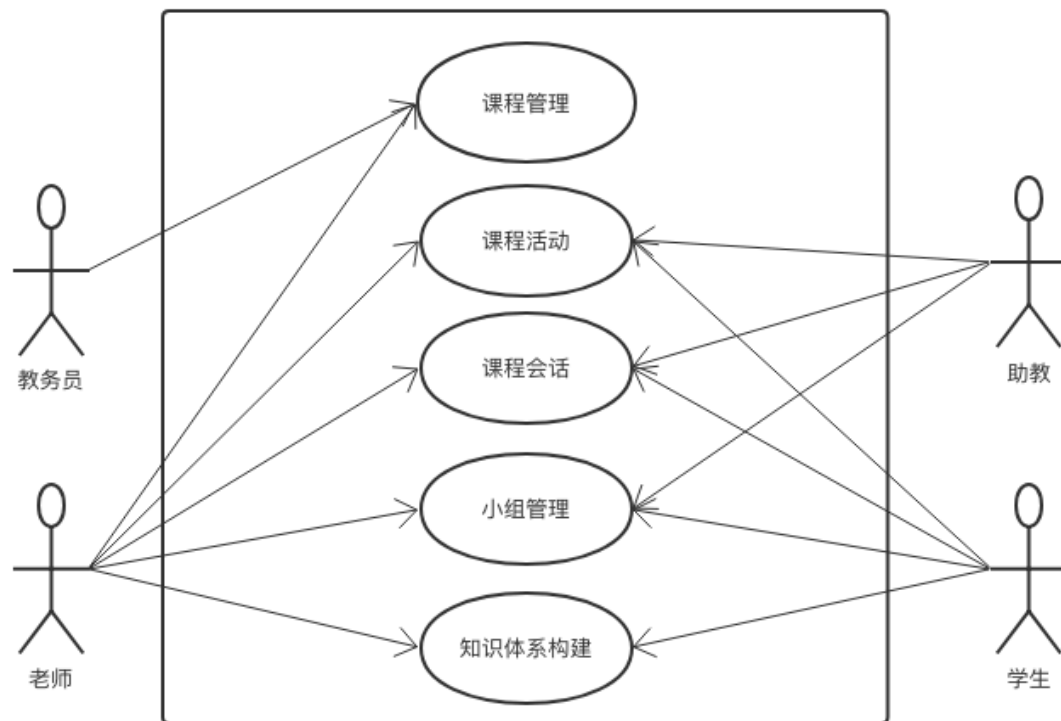
- 研究概述
- 需求分析
- 概要设计与课程活动模块的设计
- 课程活动模块的实现
- 工作总结与展望



需求分析



总体用例图



非功能性需求：

■ 可用性：

会话断线重连

■ 安全性：

防范CSRF

■ 易用性

■ 性能

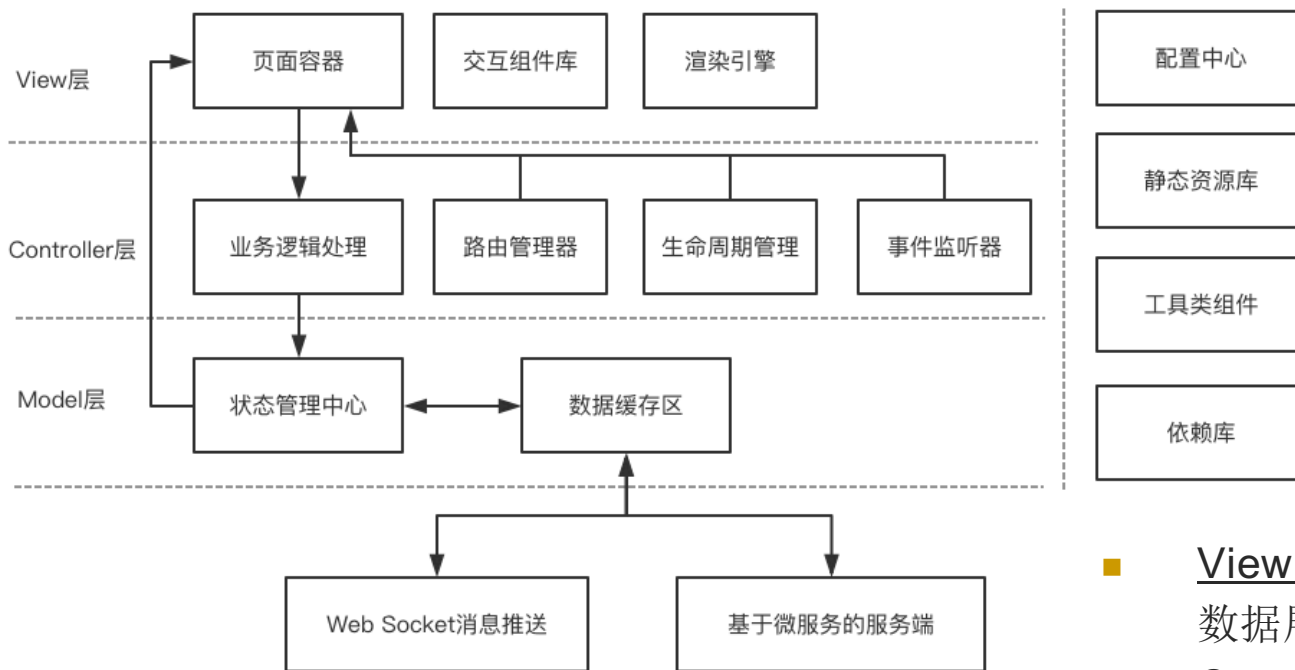
- 教员：课程管理
- 老师：课程管理、课程活动、课程会话、小组管理、知识体系构建
- 学生：课程活动、课程会话、小组管理、知识体系构建
- 助教：课程活动、课程会话、小组管理



答辩提纲



- 研究概述
- 需求分析
- 概要设计与课程活动模块的设计
- 课程活动模块的实现
- 工作总结与展望

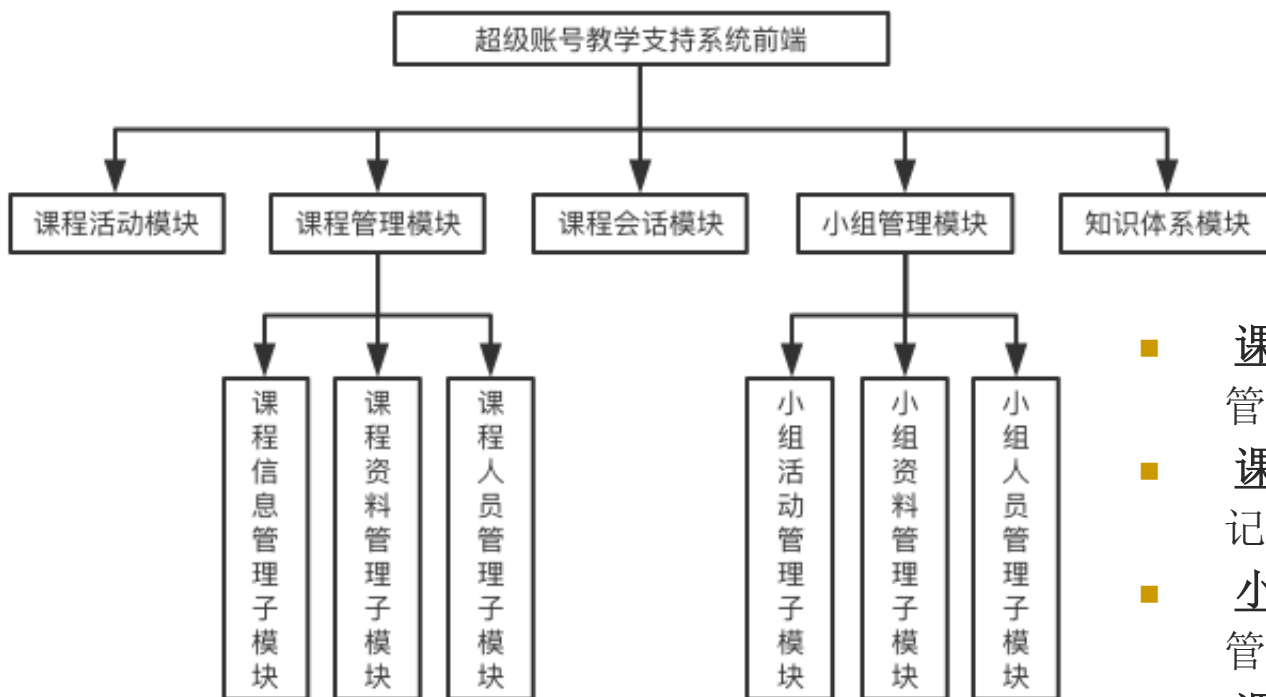


总体架构图

- View层：
数据展示，页面交互
- Controller层：
业务逻辑处理，页面容器管理
- Model层：
数据缓存，数据异步拉取



概要设计



模块划分图

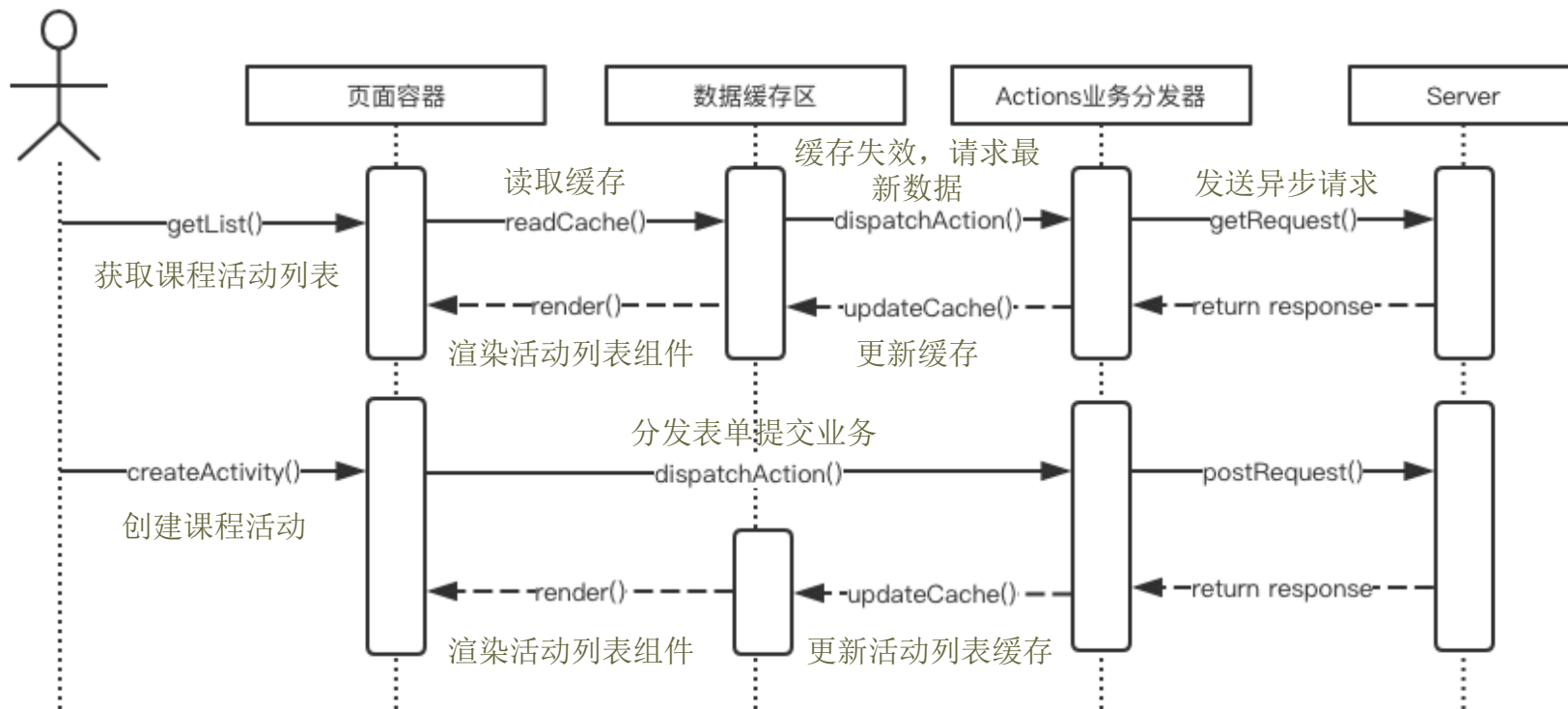
- 课程管理模块:
管理课程信息、人员、课程资料
- 课程活动模块:
记录与分析课程内的各种教学活动
- 小组管理模块:
管理课程内的小组
- 课程会话模块:
即时通讯服务
- 知识体系模块:
构建个人知识体系



课程活动模块的设计



课程活动模块的时序图

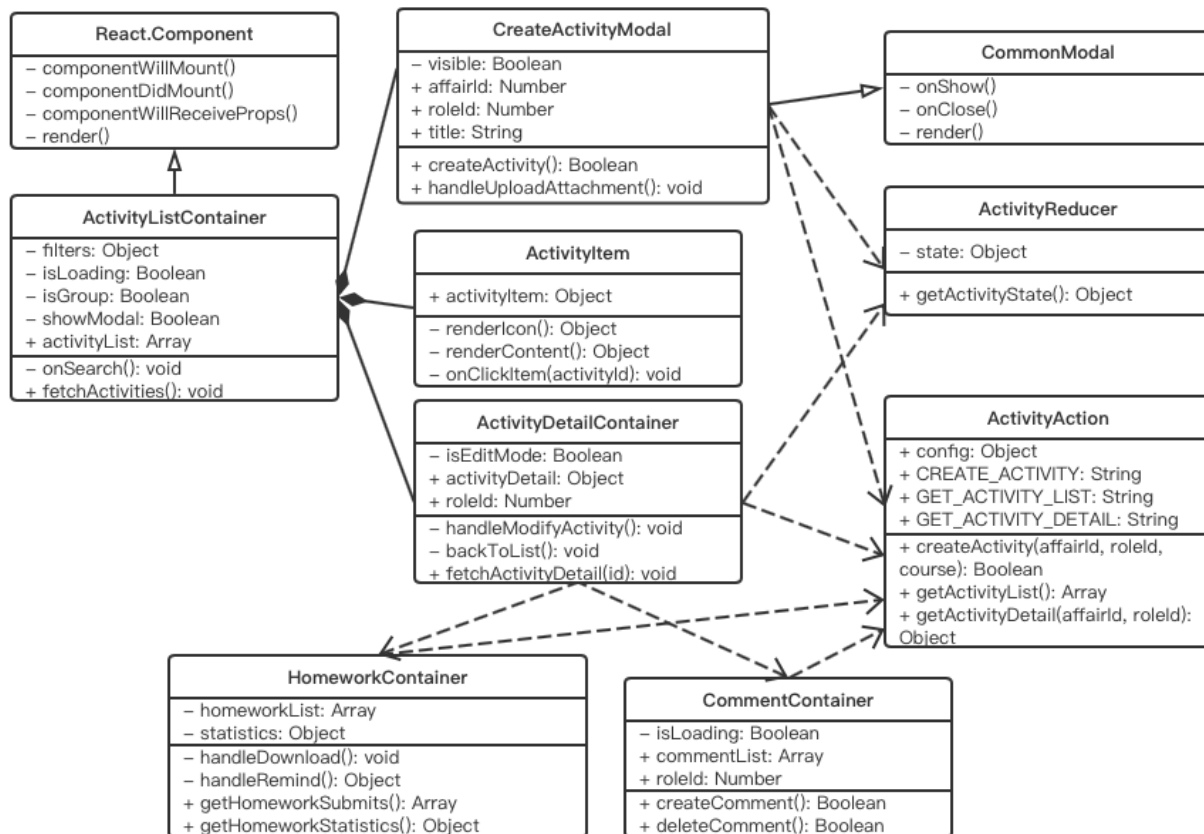




课程活动模块的设计



课程活动模块的类图



ActivityListContainer: 活动列表组件

ActivityDetailConainer: 活动详情组件

CreateActivityModal: 创建课程活动对话框

HomeworkContainer: 活动作业列表组件

CommentContainer: 活动评论区组件

ActivityReducer: 数据缓存中心

ActivityAction: 业务分发



答辩提纲



- 研究概述
- 需求分析
- 概要设计与课程活动模块的设计
- 课程活动模块的实现
- 工作总结与展望



发布课程活动的实现



发布课程活动的代码示例

```
handleOk = () => {  
  // 取出编辑器文本  
  const content = this.editor.getWrappedInstance().getContent()  
  this.props.form.validateFields(  
    (err, values) => {  
      // 表单字段合法性判断  
      if (!err) {  
        // 富文本编辑器内容不为空判断  
        const contentText = JSON.parse(content).blocks.reduce((r, v) => r + v.text, "")  
        if (contentText.length === 0 || Object.keys(JSON.parse(content).entityMap) ===  
0) {  
          message.error(this.props.title + '内容不能为空')  
          return  
        }  
        const activityType = ACTIVITY_TYPE.indexOf(this.props.title)  
        // 活动附件  
        const attachments = values.appendix.map(f => ({fileName: f.name, size: f.size,  
url: f.response.path}))  
        const formObj = {  
          content: content,  
          type: activityType,  
          attachments: attachments,  
        }  
        if (activityType === 1) {  
          formObj['homeworkType'] = values.homeworkType  
          formObj['deadline'] = values.deadline.format('x')  
        }  
        this.setState({ loading: true })  
        const { affairId, roleId } = this.props  
        this.props.createActivity(affairId, roleId, JSON.stringify(formObj)).then(res => {  
          if (res) { this.props.onSuccess() }  
          this.setState({ loading: false })  
        })  
      })  
    })  
  })  
}
```

活动内容非空校验

attachments为活动附件
附件首先会上传到OSS服务器，
OSS服务器返回文件URL，提交
表单时只保存附件的URL



活动作业统计的实现



活动作业统计的代码示例

```
componentWillMount(){
  const { affairId, activityId, role, getHomeworkStatistics, isGroup } = this.props
  // 是否需要统计信息
  const needStatistics = (role.get('roleType') == USER_ROLE_TYPE.ASSISTANT ||
    role.get('roleType') == USER_ROLE_TYPE.TEACHER) && !isGroup
  const getList = getHomeworkSubmits(affairId, role.get('roleId'), activityId)
  // 请求列表
  let promiseList = [getList]
  if (needStatistics) {
    const getStatistics = getHomeworkStatistics(affairId, role.get('roleId'), activityId)
    promiseList.push(getStatistics)
  }
  Promise.all(promiseList).then(res => {
    let homeworkList = this.state.homeworkList
    let statistics = this.state.statistics
    if (res[0].code === 0) {
      homeworkList = fromJS(res[0].data)
    } else {
      notification['error']({ message: '获取提交作业列表失败' })
    }
    if (res[1] && res[1].code === 0) {
      statistics = res[1].data
    } else if (res[1]){
      notification['error']({
        message: '获取作业统计数据失败',
        description: res[1].data
      })
    }
  })
}
```

根据用户权限判断是否需要统计数据，如果不需要，不会拉取统计数据

如果需要作业统计数据，页面容器同时发送作业列表请求和统计数据请求，**Promise.all()**方法会等待两个请求都执行完之后，再执行后续操作



活动作业统计的实现



活动作业统计的界面示例

评论	工作	文件	作业统计
参与人数 / 组数: 229			提交: 191
截止时间: 2017年12月14日 星期五 23:59:59			未交: 10
			剩余时间: 已截止
作业列表:			
	这里是作业的标题.doc 补交	张毅	2017/12/18 10:12 
	这里是作业的标题.doc	刘文竟	2017/12/18 10:12 

- 参与人数
- 已提交、未提交人数
- 作业截止时间
- 剩余时间
- 作业提交列表
- 补交



活动详情版本对比的实现



活动详情版本对比的代码示例

```
export default function differenceHighlighter(contentState, history) {
  if (!history.length) return contentState
  history = history[0]
  if (!history) return contentState

  // 判断新增的内容
  let rawState = convertToRaw(contentState)
  history.insert.forEach((ins) => {
    let newBlocks = rawState.blocks
    .slice(0, ins.position)
    .concat(ins.content)
    .concat(rawState.blocks.slice(ins.position + ins.content.length - 1))
    rawState.blocks = newBlocks
    contentState = convertFromRaw(rawState)
    ins.content.forEach((v) => {
      const entityKey = Entity.create("INSERT", "IMMUTABLE", {
        text: v.text,
      })
      contentState = Modifier.applyEntity(contentState, new SelectionState({
        anchorKey: v.key,
        anchorOffset: 0,
        focusKey: v.key,
        focusOffset: v.text.length,
      }), entityKey)
    })
  })

  // 判断删除的内容
  // ...
  // 判断替换的内容
  // ...
}
```

根据“最短编辑距离算法”计算出不同版本之间的差异，保存在`history`变量中

`history.insert`记录了所有新增的段落，根据段落内容生成新的`block`，并填充每个`block`的样式信息和实体信息



活动详情版本对比的实现



活动详情版本对比的界面示例



- 新增的内容
- 替换的内容
- 删除的内容



答辩提纲



- 研究概述
- 需求分析
- 概要设计与课程活动模块的设计
- 课程活动模块的实现
- 工作总结与展望



工作总结



- 本人独立完成超级账号教学支持系统前端的开发及构建部署工作，其中包括前端单页面应用的技术选型、构建工具选型、前端各模块的设计与实现
- 目前，该系统已在南京大学软件学院的部分课程中开始试用。从前期的运营数据来看，系统运行良好、服务稳定。师生在使用过程中给予了较好的体验反馈。



展望



- 富文本编辑器组件的功能完善
 - 表格大小的动态修改
 - 优化用户体验
- 课程会话模块中视频聊天的功能完善
 - 目前的视频聊天需要插件支持
 - 寻找替代方案，降低使用门槛



结束语



感谢各位老师！

