

# Lab5 - Last Dance

---

2020年 软件工程课程系列实验

注：本次Lab为小组实验，实验的截止日期是 2020年6月14日24:00。

## 1. 实验目标

现在你的系统可以开设会议，并允许用户针对会议进行投稿了。本次Lab中，我们的主要目标是对 **rebuttal** 相关功能进行完善。同时，这是本学期最后一次实验了，所以请尽可能从各个方面，将项目做得更完善。

## 2. 实验准备

### 2.1 完善Lab4

在正式开始 Lab5 之前，首先你需要对 Lab4 中未完成的功能进行完善，对还没有来得及修复的bug进行修复，否则放任不管的话，它们迟早有一天会再次找上回来。最后一次**Lab**了，再不修复可就来不及了 :)

### 2.2 打标签

同上一次实验一样，在正式开始 Lab5 的开发之前，请寻找到前后端仓库中 **Lab4**截止日期前的最后一次commit，将其打上 `lab4-finish` 的标签，成功打上标签后，需要将其push至华为云仓库中。（如果在上一次实验已经打上标签则请忽略。但在完成Lab5，进行最后一次commit时，将其打上 `lab5-finish` 的标签，并上传至远程仓库中）

## 3. 实验内容

### 3.1 需要实现的功能

功能列表	详细功能描述
chair与PC对于评审结果进行初次讨	以建立帖子和回帖的形式进行讨论，PC可以根据讨论结果决定是否 需要修改自己的当前评审结果。如果要修改，PC Member只有一次 修改评审结果的机会。如果不修改，PC Member需要确认不修改。 确认完评审结果后，Chair发布初次论文录用结果。

论

author提交 rebuttal 信息 author根据评审结果和评语进行rebuttal，rebuttal的目的是针对评审的质疑进行解释。只有没有初次没有录用的论文需要提交rebuttal。一篇论文的3个分数中只要有一个为-1或-2即表示该论文没有录用。author以输入框的形式提交rebuttal信息，rebuttal信息只有一次提交机会。

chair与PC对于rebuttal信息再次进行讨论 以建立帖子和回帖的形式进行讨论，PC可以根据讨论结果决定是否需要修改自己的初次评审结果。如果要修改，PC Member只有一次修改评审结果的机会。如果不修改，PC Member需要确认不修改。确认完评审结果后，Chair再次发布论文录用结果。

发布最终论文录用结果 PC Member们根据rebuttal的情况可以修改自己的初次评分，也可以不修改，所有的PC Member确认自己的再次评分后，chair发布最终论文录用结果。

完善稿件分配策略 PC member不能审自己提交的稿件和自己作为作者的稿件（根据姓名和邮箱来验证PC是否在作者中），chair开启稿件分配的时候，如果没有可行解或者不满足稿件分配条件，系统提示chair稿件分配失败。

修改评审结果 PC对自己已经审过稿件的评审结果进行修改（有两个阶段可以进行修改，第一阶段是开始评审和发布评审结果之间，第二阶段是在用户提交rebuttal信息之后和最终录用结果发布之前）

### 3.2 需求分析报告

需求分析是软件开发过程中的一个重要环节。该阶段主要是分析系统在功能上需要“实现什么”，而不是考虑如何去“实现”。需求分析的目标是把用户对待开发软件提出的“要求”或“需要”进行分析与整理，确认后形成描述完整、清晰与规范的文档，确定软件需要实现哪些功能，完成哪些工作。此外，软件的一些非功能性需求(如软件性能、可靠性、响应时间、可扩展性等)，软件设计的约束条件，运行时与其他软件的关系等也是软件需求分析的目标。

之前的实验，我们都是要求大家使用 DevCloud 中提供的需求规划功能或者看板功能进行需求规划。这次大家需要针对整个项目，额外写一份需求分析报告，并以PDF形式提交，文件命名为：组长姓名-组长学号-需求分析报告。

需求分析报告的内容要求如下：

- 对整个系统开发背景的分析介绍
- 对整个系统业务流程的分析介绍
- 对整个系统功能需求的分析介绍，如注册登录，会议申请管理，投稿审稿等功能需求分析
- 对整个系统非功能性需求的分析介绍，如性能，安全性，运行环境等需求分析
- 其他有关内容可自行添加

### 3.3 CI/CD流水线(提高要求-卓越班)

注：此条为卓越班小组内容要求，非卓班小组自愿选做。该部分不会影响到非卓班小组的评分。

在Lab4中，我们已经构建了一条相对完善的流水线，相信你已对其有了一定的了解。其实构建流水线本身并不复杂，而我们的目标其实是通过流水线进行符合规范的敏捷开发实践。简单来说，你要有合适的分支管理策略，并且对合适的分支进行CI/CD（例如master），所以这种情况下，每次当你想要往master分支上push代码的时候，你要尽可能地保证这个版本的代码是可用的，所以你的流水线才能够顺利跑完。所以本次Lab中，我们对流水线的健康度提出要求，请保证流水线中的每个任务的健康度都在四星及以上。

## 4. 实验要求

### 4.1 实验过程要求

#### 4.1.1 需求规划

在实验过程中，使用 DevCloud 中提供的需求规划功能或者看板功能进行需求规划。

#### 4.1.2 Git协同开发

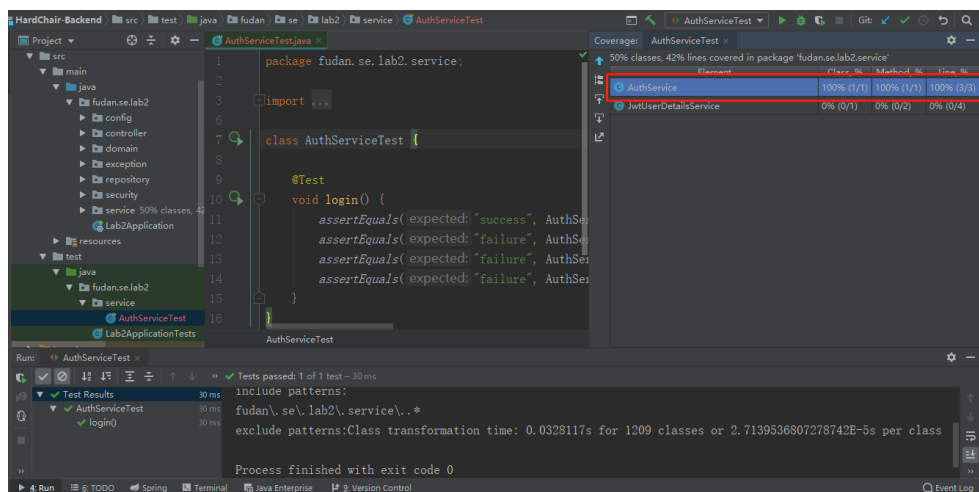
在实验过程中，使用 Git 进行小组协同开发。良好的 Git 使用习惯包括但不限于：

- 及时commit，以及规范的commit信息。
- 合适的分支管理策略。

### 4.2 测试要求

#### 4.2.1 单元测试

同上次实验一样，要求大家对每个类或方法进行“单元测试”，并保证测试覆盖率不低于80%，并在实验报告中附图说明。（注：主要是对后端进行单元测试，前端可自行功能测试）



## 4.2.2 集成测试

之前的几次实验都是要求大家对每个类或方法进行单元测试。在做单元测试的时候，我们希望只实例化测试目标类的实例，并且屏蔽目标类的依赖。然而当涉及到，目标类对外部资源存在依赖时，我们可以采取模拟的方法，如构造模拟数据以完成对目标类的单元测试。

但是我们并没有对类与类之间的依赖进行测试验证。实践证明，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常的工作。一些局部反映不出来的问题，在全局上很可能暴露出来。所以这次要求大家增加一个“集成测试”。集成测试是在单元测试的基础上，将所有模块按照设计要求组装成为子系统或系统，进行集成测试。这个时候的测试就会超出一个类的范围，所以我们需要保证自测试目标类及以下的依赖类，都能够在测试中被实例化。

我们要求大家针对自己项目中的不同功能模块，如注册登录模块，会议申请管理模块，论文投稿模块，稿件分配模块，审稿发布模块，管理员模块等进行集成测试，并将各模块的测试过程和测试结果详细记录到实验报告中。

## 4.3 代码质量

### 4.3.1 华为云代码检测

同上次实验一样，大家继续使用代码检查工具，并根据质量评估结果，对代码问题，圈复杂度，代码重复率都需要进行修改。每个代码问题华为云都会有相应的修改意见，大家可以直接进行修改，最后要保证代码问题数为‘0’全部解决，质量门禁显示‘passed’，而圈复杂度和代码重复率尽可能低，并在实验报告中附图说明。（注：如果前端开源组件等有质量问题无法处理，可以在质量检查的时候，设置忽略该组件的相关代码文件。其他可能出现的质量问题如果合理，可在实验报告中说明情况）

任务	质量门禁	问题	最近一次检查	操作
<div>Lab2_frontend_official</div>	Passed	<div>0 未解决</div> <div>10 已解决</div>	手动触发 检查于 2020-03-23 14:10:54	...
<div>Lab2_backend</div>	Passed	<div>0 未解决</div> <div>4 已解决</div>	手动触发 检查于 2020-03-23 01:20:01	...

### 4.3.2 SonarQube代码检测

华为云提供的代码检测能力有限，为了进一步解决代码中Bad Smell的问题，本次实验中，我们要求使用 SonarQube 分别对前后端代码进行检测，并解决扫描出来的Bug, Vulnerability, Code Smell等。请按照课堂上助教的演示指导，在自己电脑或者服务器上安装SonarQube，并对项目代码进行检测。最后要保证**Bug**和**Vulnerability**数目为0，**Code Smell**的数目在合理范围之内（无需解决的**Code Smell**需要在实验报告中解释说明）。

我们已经在课堂上演示过Java项目如何接入SonarQube。关于前端项目如何接入SonarQube，你可以查看[官方文档](#)进行解决。

## 4.4 构建部署要求

同上次实验一样，要求对 `lab5-finish` 标签所对应的 `commit` 版本在华为云上进行前后端的编译构建和部署。

### 4.4.1 编译构建

在上次的实验中发现很多组，没有按照要求选择特定的Tag版本进行编译构建。在华为云中，支持使用特定的Tag版本进行编译构建。具体来说，在新建编译配置步骤里面，你需要：

1. 点击高级设置。
2. 选择使用 指定Tag构建，填写合适的克隆深度。

新建编译构建任务

基本信息

选择代码源

选择构建模板

源码源:

DevCloud

GitHub

通用Git

码云

来自流水线

源码仓库:

HardChair-Frontend

分支:

master

收起高级设置

使用指定代码仓库Tag或CommitID构建:

不指定

指定Tag构建

指定CommitID构建

克隆深度:

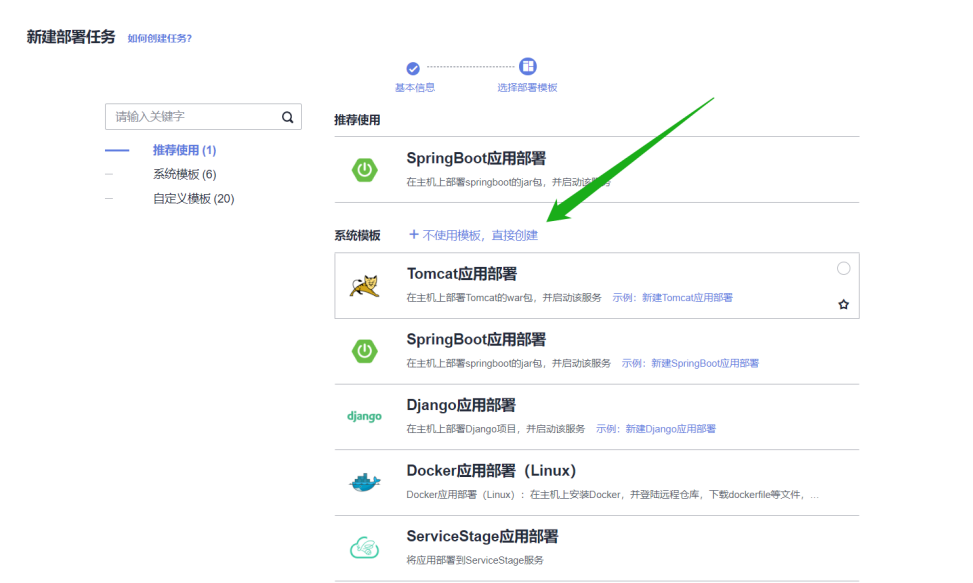
10

这样，在执行编译构建的时候，你需要输入相应Tag的名称，它就会选择相应的版本，进行编译构建了。

注意：在实验截止后，你仍可以任意地往仓库中提交代码，但是请保证在80端口部署`lab5-finish`所对应的项目版本，以供助教检查。

### 4.4.2 部署

上次实验中发现，有的小组在前端上部署出现了问题。在华为云的部署中，并没有提供一个开箱即用的 Nginx部署模板，因此你需要选择 **不使用模板，直接创建**，然后参考手动部署过程，自行添加所需要的步骤，形成一个自定义的前端部署模板。



我们建议使用华为云对前后端进行编译构建和部署。但是由于系统环境以及前端部分第三方包可能出现的不兼容性，同样的代码，可能会出现在本机编译通过，而在服务器上编译失败的情况。因此，如果碰到类似情况，我们允许在本机编译构建出目标文件夹后，直接上传至Github仓库中。不过，你仍然需要使用华为云的部署步骤，将其部署至服务器上。

## 4.5 其他要求

### 1. 明确会议时间逻辑

要求投稿截止日期必须要在结果发布日期之前，结果发布日期必须要在会议举办日期之前，而且3个日期都必须为现在及未来的日期，不能是过去的日期，即 截稿日期 < 成绩发布日期 < 会议举办日期。

### 2. 注明管理员账号和密码

请每个小组在自己的实验报告里注明自己项目的管理员账号和密码。



### 3. 修改\*\*项目名称\*\*

请组长在 DevCloud 中将自己组的项目名称修改为 2020软件工程\_<项目名称>\_<小组名称>，在项目描述中 介绍组内成员。所有同学在 DevCloud 中个人头像处点击 设置昵称，将昵称设置为自己的真实姓名。

## 5. 实验提交

### 5.1 打标签

在完成本次Lab的所有内容后，首先你们需要使用Git为项目打上相应的标签。具体来说，你们需要分别选择前后端截止日期前的一次commit，将其打上 lab5-finish 的标签，并上传至远程仓库中。

注意，我们将使用此标签所对应的commit版本进行评分。

### 5.2 实验提交

实验报告部分的要点如下：

- 每个小组提交一份pdf版本的实验报告
- pdf文件的命名为：组长姓名-组长学号-Lab5实验报告
- 注明自己项目的管理员账号和密码
- DevCloud中对Lab5进行项目需求规划（Scrum项目）或者使用看板分配任务（看板项目）的截图
- 对Lab5的项目代码进行单元测试及代码覆盖率的结果截图
- 对Lab5的项目的功能模块进行集成测试的过程和结果截图
- DevCloud中对Lab5的项目代码使用代码检查的检查结果截图
- SonarQube中对Lab5项目代码使用代码检测的实验过程以及相关截图
- 利用Git对Lab4截止日期前的最后一次commit打tag的截图
- 利用Git给Lab5截止日期前的最后一次commit打tag的截图
- 前后端项目仓库构建 CI/CD 流水线的截图（卓越班要求）
- 项目各个页面的截图以及使用说明
- 每个组员的任务分配情况
- 小组的实验过程记录，遇到的问题以及解决方案
- 每个小组成员单独的实验总结

需求分析报告部分的要点如下：

- 每个小组提交一份pdf版本的实验报告
- pdf文件的命名为：组长姓名-组长学号-需求分析报告
- 对整个系统开发背景的分析介绍
- 对整个系统业务流程的分析介绍
- 对整个系统功能需求的分析介绍，如注册登录，会议申请管理，投稿审稿等功能需求分析
- 对整个系统非功能性需求的分析介绍，如性能，安全性，运行环境等

## 需求分析

最后，请组长将 实验报告 和 需求分析报告 ，打包成一个压缩文件，并在 Classroom上按时提交。