<有课>

测试报告

版本 <1.0>

[注：其中包括用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File >Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit> Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2019/7/22 | <0.4> | 迭代二完成测试 | 朱朝阳 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 4

4. 测试结果及分析 4

4.1 需求覆盖率及缺陷分布 4

4.2 缺陷严重程度 5

5. 缺陷清单 5

5.1 功能性缺陷 5

5.2 非功能性缺陷 6

6. 测试结论与建议 6

测试报告

# 简介

## 目的

*本文档是有课项目开发的第一次迭代的测试报告，是对从2019/7/12-2019/7/22之间，对于有课项目的测试形成的工作报告。*

## 范围

《测试用例》 C18有课项目

## 定义、首字母缩写词和缩略语

*无*

## 参考资料

《立项建议书》 C18有课项目

《用例规约》 C18有课项目

《迭代计划》 C18有课项目

《测试用例》 C18有课项目

## 概述

测试报告包括测试的目的，范围，参考资料，测试概要，测试环境，测试的结果与分析，需求覆盖率及缺陷分布，缺陷清单，测试结论。其中测试结果与分析将从前后端两个方面来分别阐述。

# 测试概要

测试贯穿与开发全过程，以测试驱动开发，测试参与人员即是测试开发人员。测试方法主要为单元测试，测试内容包括前端react-native测试与后端spring测试。

# 测试环境

迭代二尚未部署服务器，所有测试为程序员本机测试

# 测试结果及分析

后端主要使用Junit4完成测试



D:\qq\1853935753\Image\C2C\9)6@]HD0X{F_PEKLUT~H)DY.png









对于数个新的Controller 和 ServiceImplement，我们都做了详细的测试

## 需求覆盖率及缺陷分布

*[需求覆盖率是指经过测试的需求/功能和软件需求规约中所有需求/功能的比值，通常情况下要达到100%的目标。需给出功能测试及非功能测试的需求覆盖情况。对于功能测试，需给出各个主要功能模块的需求覆盖率。如果需求覆盖率未达到100%可在备注项中说明原因及未测试内容。缺陷率是指本项缺陷占总缺陷数的百分比。请尽量使用图、表进行描述。]*

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **课程评论** | 10 | 100% | 0 | 0 |  |
| **课程评测** | 6 | 100% | 1 | 13% | 受前端控制不会产生。仅受制于连接请求 |
| **课程提问** | 6 | 100% | 2 | 33% | 受前端控制不会产生。仅受制于连接请求 |
| **课程回答** | 6 | 100% | 1 | 13% | 受前端控制不会产生。仅受制于连接请求 |
| **非功**  **能项** | **性能** |  |  |  |  |  |
| **可靠性** |  |  |  |  |  |
| **…** |  |  |  |  |  |
|  | **非功能项小计** |  |  |  |  |  |
|  | **总计** | *28* | *100%* | *4* | *14.3%* |  |

## 缺陷严重程度

*[描述缺陷按严重程度分布情况，包括不同严重程度缺陷个数及占缺陷总数百分比等。尽量使用图、表表示。]*

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** |  |  | 3 |  | 3 |
| **占缺陷百分比** |  |  | 100% |  |  |

# 缺陷清单

## 功能性缺陷

*[简要阐明功能缺陷汇总情况，并给出具体功能缺陷列表。]*

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | Bug1 | 一般 | 课程评测 | 用户检测 | 课程评测并不检测所得的用户名，即非法用户可以为课程评测；  课程评测也不检测课程名，即无需有的课程也可以被评测 | Testcase-16 |
| 2 | Bug2 | 一般 | 课程问题点赞 | 状态 | 课程问题点赞不会检测是否已经点过赞，如果有信号输入可能会重复点赞 | Testcase-021 |
| 3 | Bug3 | 一般 | 课程问题 | 用户检测 | 课程提问并不检测所得的用户名，即非法用户可以为课程添加问题 | Testcase-020 |
| 4 | Bug4 | 一般 | 课程回答点赞 | 用户检测 | 课程回答点赞并不检测是否已经点过赞，如果有信号输入可能会重复点赞； | Testcase-23 |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

# 测试结论与建议

本次测试主要是针对于迭代二实现功能的功能测试，尚未实现性能测试，可靠性测试。通过不断地测试与开发，修改了很多潜在的bug，对于项目的主逻辑部分测试覆盖率较高。这次测试主要针对的模块是迭代二中课程评测、课程提问、提问回答等方面。之后的功能需要在此基础上拓展，测试为之后的开发提供了保障。下一次迭代我们应该更多使用自动化测试工具，让前后端的测试更轻松