

测试文档

项目组成员

学号	姓名	Github
1352892	谭靖儒	tztztztzt
1352875	黄安娜	aaana
1352965	晁佳欢	wlmxjm1
1352923	马致远	maerye
1352965	林昌盛	apelyn

文档说明

该文档是本软件复用项目（公共聊天室）的测试文档

测试过程

- 测试计划：充分准备的测试计划工作可以有效地克服测试的盲目性，缩短测试周期，提高测试效率，并且起到测试文档与开发文档互查的作用。测试计划包括对以下方面的计划
 - 测试过程
 - 需求跟踪
 - 测试项目
 - 测试时间安排
 - 测试记录
 - 软件和硬件需求
- 测试设计：制定测试方案、选择测试工具、设计测试用例
- 测试准备：搭建测试环境、掌握相关技术
- 测试执行：按测试计划及测试测试设计执行测试，记录测试日志及测试缺陷，进行回归测试
- 测试评估：对整个测试过程进行评估，总结测试结果

测试内容

- [单元测试](#)
- [功能测试](#)
- [集成测试](#)
- [系统测试](#)

主要负责人

- 黄安娜

单元测试
- 晁佳欢

单元测试
- 马致远

功能测试
- 谭靖儒

功能测试

测试目的

- 程序测试过程，目的是尽可能发现并改正被测试软件中的错误，提高软件的可靠性。它是软件生命周期中一项非常重要且非常复杂的工作，对软件的可靠性保证具有极其重要的意义。
- 对项目进行

单元测试

，测试各单元代码的正确性，尽可能早地发现并改正bug，保证软件产品的质量。
- 对项目进行

功能测试

，验证软件产品满足现阶段需求：
 - 客户端能够申请登陆服务器, 并且登陆成功之后能够发送信息。（登陆、发送消息功能）
 - 服务端能够客户端发送的信息做一定的处理, 并且能够对登陆成功的客户端的信息转发给所有已登陆的客户端（转发功能）
 - 客户端和服务端都能记录活动信息到文件中（性能指标功能）
 - 压缩功能
 - 保存消息功能
 - 文件每天归档功能
 - 同组广播功能
 - 文件每周归档功能

- 加密功能
 - Client按次序接收所有遗漏的消息
- 对项目进行 **集成测试**，在进行单元测试之后，将各单元集成为各个子系统，然后对子系统进行相关的测试
 - 聊天子系统
 - PM子系统
 - LICENSE子系统
 - CM子系统
 - 数据库子系统
 - 压缩系统
- 对项目进行 **系统测试**，在集成测试通过后，将子系统集成起来进行测试

测试时间安排

- 在完成一个单元后进行相关的单元测试
- 在完成一个功能时进行相关的功能测试

测试时间表

日期	说明	测试人员	评审人员
2016.03.27-2016.03.30	单元测试设计及执行	黄安娜、晁佳欢	全体成员
2016.03.30-2016.4.10	基本功能测试	黄安娜、晁佳欢、马致远	全体成员
2016.04.10-2016.04.19	选择可复用构件后重新执行相关的测试用例	全体成员	全体成员
2016.04.25-2016.05.13	扩展功能测试	全体成员	全体成员

测试工具

Junit 4.0

测试源文件夹

[src/test/java](#)

测试详细信息

- [单元测试](#)
 - [配置模块](#)
 - [server/Json解析模块](#)
 - [server/登陆验证模块](#)
 - [server/管道管理模块](#)
 - [server/rateLimiter模块](#)
 - [server/日志模块](#)
 - [server/数据库访问模块](#)
 - [client/Json解析模块](#)
 - [client/日志模块](#)
- [功能测试](#)
 - [登录功能](#)
 - [发送消息功能](#)
 - [性能指标功能](#)
 - [Server/Client保存所有消息到文件](#)
 - [文件压缩功能](#)
 - [每天所有的输出文件归档成一个压缩包](#)
 - [同组广播功能](#)
 - [每周归档功能](#)
 - [加密功能](#)
 - [Client按次序接收所有遗漏的消息](#)
 - [用户上下线提示功能](#)
- [集成测试](#)

- 系统测试

单元测试

配置模块

测试用例

ConfigReaderTest.java

此模块用于读取配置文件，可配置路径名，直接读成Java bean中

用例编号	用例描述	输入数据	预期结果	实际结果
1	读取配置文件	"config/conf.json"	host=="localhost",port==8080,maxMsgNumber==100,maxMsgNumberPerSec==5	host=="localhost",port==8080,maxMsgNumber==100,maxMsgNumberPerSec==5

server/Json解析模块

测试用例

JsonToObjectHandlerTest.java

此模块用于将Json解析为message对象

编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	Json解析为类型为chatting的message对象	类型为chatting的message序列化的Json字符串	类型为chatting的message对象	类型为chatting的message对象	通过	2016.3.27	无
2	Json解析为类型为authority的message对象	类型为authority的message序列化的Json字符串	类型为authority的message对象	类型为authority的message对象	通过	2016.3.27	无

server/登陆验证模块

测试用例

AuthorityHandlerTest.java

此模块用于进行登陆消息的验证。对于登陆消息，账户正确则验证成功，否则验证失败；对于聊天消息则直接忽略

编号	用例描述	输入数据	预期结果
1	正确的登陆消息验证	正确的登陆 message ({LoginContent("100","123456"),MessageStatus.NEEDHANDLED,MessageType.AUTHORITY})	message的Status为NEEDHANDLED ({LoginContent("100","123456"),MessageStatus.NEEDHANDLED,MessageType.AUTHORITY})
2	错误的登陆消息验证	错误的登陆message ({LoginContent("123","123456"),MessageStatus.NEEDHANDLED,MessageType.AUTHORITY})	message的Status为LOGINFAIL LOGINFAIL ({LoginContent("123","123456"),MessageStatus.LOGINFAIL,MessageType.AUTHORITY})
3	聊天消息验证	聊天message ({ChatContent("hello"),MessageStatus.NEEDHANDLED,MessageType.CHATTING})	未发生改变的message ({ChatContent("hello"),MessageStatus.NEEDHANDLED,MessageType.CHATTING})

server/管道管理模块

测试用例

ChannelManagerHandlerTest.java

该模块用于在登陆成功时，将该channel以及所对应groupId维护在Manager.clientChannels中

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果
1	登陆成功则添加channel	验证成功的message ({LoginContent("100","123456"),MessageStatus.NEEDHANDLED,MessageType.AUTHORITY})	Manager.clientChannels的长度+1	Manager.channels的长度+1	通过
2	登陆失败channel个数不变	验证失败的message ({LoginContent("123","123456"),MessageStatus.LOGINFAIL,MessageType.AUTHORITY})	Manager.clientChannels的长度不变	Manager.channels的长度不变	通过
3	聊天消息	聊天 message ({ChatContent("hello"),MessageStatus.NEEDHANDLED,MessageType.CHATTING})	Manager.channels的长度不变	Manager.clientChannels的长度不变	通过

rateLimiter模块

测试用例

LimiterHandlerTest.java

该模块用于进行License检查

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	正常发送	不超速的发送message	信息状态变为NEEDHANDLED（不变）	信息状态变为NEEDHANDLED（不变）	通过	2016.3.28	无
2	发送过频繁	超过指定速率发送message	信息状态变为TOOFREQUENT	信息状态变为TOOFREQUENT	通过	2016.3.28	无
3	发送过多	超过指定额度发送message	信息状态变为OVERRANGE	信息状态变为OVERRANGE	通过	2016.3.28	无

server/日志模块

测试用例

LoggerHandlerTest.java

该模块用于服务端PM日志的记录

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	登录失败	验证失败的message	LoggerHandler的invalidLoginNumber+1，其余不变	LoggerHandler的invalidLoginNumber+1，其余不变	通过	2016.3.28	无
2	登陆成功	验证成功的message	LoggerHandler的validLoginNumber+1，其余不变	LoggerHandler的validLoginNumber+1，其余不变	通过	2016.3.28	无
3	聊天消息（正常）	聊天message	LoggerHandler的receivedMessageNumber+1，forwardMessageNumber+1，其余不变	LoggerHandler的receivedMessageNumber+1，forwardMessageNumber+1，其余不变	通过	2016.3.28	无
4	聊天消息（发送过多）	聊天message	LoggerHandler的receivedMessageNumber+1，ignoredMessageNumber+1，其余不变	LoggerHandler的receivedMessageNumber+1，ignoredMessageNumber+1，其余不变	通过	2016.3.28	无
5	聊天消息（发送过频繁）	聊天message	LoggerHandler的receivedMessageNumber+1，ignoredMessageNumber+1，其余不变	LoggerHandler的receivedMessageNumber+1，ignoredMessageNumber+1，其余不变	通过	2016.3.28	无

server/数据库访问模块

测试用例

LoginServerTest.java

该模块用于连接数据库，处理登陆信息

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	成功登录	正确的用户名和密码	返回相应组号	返回相应组号	通过	2016.5.3	无
2	登陆失败	带有字符'的用户名	返回-1	返回-1	通过	2016.5.3	无
3	登陆失败	空用户名	返回-1	返回-1	通过	2016.5.3	无
4	登陆失败	用户名过长，密码为空	返回-1	返回-1	通过	2016.5.3	无
5	成功登录	正确的用户名和密码	返回相应组号	返回相应组号	通过	2016.5.3	无

client/Json解析模块

测试用例

JsonHandlerTest.java

该模块用于客户端的Json解析，将Json解析成ACK对象

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	Json解析为类型为SENDSUCCESS的ACK	类型为SENDSUCCESS的ACK序列化的Json字符串	类型为SENDSUCCESS的ACK	类型为SENDSUCCESS的ACK	通过	2016.3.29	无
2	Json解析为类型为OTHERSMESSAGE的ACK	类型为OTHERSMESSAGE的ACK序列化的Json字符串	类型为OTHERSMESSAGE的ACK	类型为OTHERSMESSAGE的ACK	通过	2016.3.29	无
3	Json解析为类型为TOOFREQUENT的ACK	类型为TOOFREQUENT的ACK序列化的Json字符串	类型为TOOFREQUENT的ACK	类型为TOOFREQUENT的ACK	通过	2016.3.29	无
4	Json解析为类型为REDOLOGIN的ACK	类型为REDOLOGIN的ACK序列化的Json字符串	类型为REDOLOGIN的ACK	类型为REDOLOGIN的ACK	通过	2016.3.29	无
5	Json解析为类型为LOGINFAIL的ACK	类型为LOGINFAIL的ACK序列化的Json字符串	类型为LOGINFAIL的ACK	类型为LOGINFAIL的ACK	通过	2016.3.29	无

client/日志模块

测试用例

ClientLoggerHandlerTest.java

该模块用于客户端PM日志的记录

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	登录失败	类型为LOGINFAIL的ACK	loginFail+1，其余不变	loginFail+1，其余不变	通过	2016.3.29	无
2	登陆成功	类型为LOGINSUCCESS的ACK	loginSuccess+1，其余不变	loginSuccess+1，其余不变	通过	2016.3.29	无
3	接收消息成功	类型为OTHERSMESSAGE的ACK	receiveMsgNumber+1，其余不变	receiveMsgNumber+1，其余不变	通过	2016.3.29	无
4	成功发送消息	类型为SENDSUCCESS的ACK	无改变	无改变	通过	2016.3.29	无
5	发送消息过频繁	类型为TOOFREQUENT的ACK	无改变	无改变	通过	2016.3.29	无
6	发送消息过多	类型为REDOLOGIN的ACK	无改变	无改变	通过	2016.3.29	无

功能测试

登录功能

- 需求分析：
 - 当用户正确地输入用户名和密码时登录成功跳转到聊天界面
 - 当用户没有输入用户名或密码时，弹出错误对话框，提示登陆失败
 - 当用户输入错误的用户名或密码时，弹出错误对话框，提示登陆失败
- 测试用例集：

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	只输入用户名	用户名	登录失败	弹窗提示登录失败	通过	3.29	无
2	只输入密码	密码	登录失败	弹窗提示登录失败	通过	3.29	无
3	不输入任何	无	登录失败	弹窗提示登录失败	通过	3.29	无
4	输入错误的用户名/密码组合	用户名，密码	登录失败	弹窗提示登录失败	通过	3.29	无
5	输入正确的用户名/密码组合	用户名，密码	登录成功	登录成功并跳转到聊天界面	通过	3.29	无

发送消息功能

为方便测试，将客户端发送消息上限改为5条，频率上限改为2条/s

- 需求分析：
 - 当客户端发送消息满足 数目未超上限且频率未超上限 的条件时，该客户能够正常发送消息，消息能够正常转发
 - 当客户端发送消息数目超过数目上限时，弹出对话框，提示发送消息数目超过上限，同时跳转到登陆界面需要重新登录，其他客户不会收到该条消息
 - 当客户端发送消息数目超过频率上限时，弹出对话框，提示发送消息过于频繁，此时无需重新登录，其他客户不会收到该条消息
- 测试用例集：

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	多个客户端之间相互发送消息	每个客户端各自发送一条消息	剩余客户端收到其他客户端发送的消息	同预期结果	通过	3.29	无
2	客户端发送消息数目超过上限	客户端发送五条消息	在发送第五条消息时提示发送超过上限并跳转至登录界面,其他客户端无法收到第五条消息	同预期结果	通过	3.29	无
3	客户端发送消息超过频率上限	以很快的速率发送消息	弹窗提示发送消息过快,其他客户端无法收到提示过快的消息	同预期结果	通过	3.29	无

性能指标功能

为方便测试，将客户端发送消息上限改为5条，频率上限改为2条/s

- 需求分析：
 - 客户端记录以下性能指标到文件
 - loginFailNumber
 - loginSuccessfulNumber
 - sendMessageNumber
 - receiveMessageNumber
 - 服务端记录以下性能指标到文件
 - invalidLoginNumber
 - validLoginNumber
 - receiveMessageNumber
 - forwardMessageNumber
 - ignoreMessageNumber

- 测试用例集：

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	登录失败，查看客户端服务端日志	错误的账号密码	客户端日志的Loginfailnumber+1且服务端日志的invalidLoginNumber +1	与预期一致	通过	3.30	通过
2	登录成功，查看客户端服务端日志	正确的账号密码	客户端日志的Login successful number+1且服务端validLoginNumber+1	与预期一致	通过	3.30	无
3	客户端发送一条消息，查看日志	消息	客户端的sent message number+1且服务端的forwardMessageNumber+1	同预期	通过	3.30	无
4	客户端发送五条消息，查看日志	五条消息	客户端sent message number+5且服务端的receivedMessageNumber+5, forwardMessageNumber +4, ignoredMessageNumber不变。	同预期	通过	3.30	无
5	客户端快速发送消息	消息	服务端ignoredMessageNumber+一定数目	同预期	通过	3.30	无
6	客户端收到消息	其他客户端发送一条消息	客户端receviedMessageNumber+1	同预期结果	通过	3.30	无

Server/Client保存所有消息到文件

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	Client保存所有收到的消息到messageRecords下以account为文件名的log文件中	用户100和101登陆系统互相发送消息，100收到的所有消息保存在100.log中，101收到的所有消息保存在101.log中	与预期一致	通过	2016.05.03	无	
2	Server保存所有收到的消息到messageRecords下的server.log文件中	用户100和101登陆进系统，并发送消息	与预期一致	通过	2016.05.03	无	

文件压缩功能

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	先使用Log.writeFile()将消息写入文件中，再使用Log.compress()对所写文件进行压缩。	文件路径，消息内容	所有被写入的文件都被压缩进目标文件中去。	同预期结果	通过	2016.5.3	无

同组广播功能

- 需求：配置同组成员，并只向同组成员广播消息
- 用户信息

username	password	groupid
100	123456	1
101	123456	1
200	123456	2
201	123456	2
202	123456	2
300	123456	3

- 测试用例集：

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	同组用户间发送消息	同属于组1的用户100和用户101登陆进系统，互相发送消息	能够互相收发消息	与预期一致	通过	2016.05.03	无
2	不同组用户间发送消息	属于组1的用户100和属于组2的200用户登陆进系统，互相发送消息	互相收发消息	与预期一致	通过	2016.05.03	无
3	同组和不同组用户间发消息	用户100、101和200登陆进系统，互相发送消息	100和101能够互相收发消息而与200不能互相收发消息	通过	2016.05.03	无	

对压缩包重新压缩

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	先使用Log.write()功能写入一些文件，先使用Log.setCompressPath()设置压缩包的路径，再使用Log.compress()功能对所写文件进行压缩。然后使用Log.resetCompress()进行重置，再进行一遍上述过程，生成另外一个压缩包。最后使用Log.reCompress()对上述两个压缩包进行重新压缩，生成一个压缩包。	文件路径，消息等	最后生成的压缩包中包含前两个压缩包的内容	同预期结果	通过	2016.5.3	无

加密功能

加密构件 CIPHER | 用例编号 | 用例描述 | 输入数据 | 预期结果 | 实际结果 | 测试结果 | 测试时间 | bug描述 |----:|----:|----:|----:|----:|----:|----:| |1|1|用给定的密匙对文件进行加密解密|文件输入路径, 文件输出路径|生成加密, 解密后的文件|预期结果|通过|2016.5.6|无 |1|1|用密匙变量对文件进行加密解密|文件输入路径, 文件输出路径, 密匙|生成加密, 解密后的文件|预期结果|通过|2016.5.6|无

多任务压缩功能

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	定义压缩任务,查看是否正确把一个目录下的所有文件压缩到指定的路径	定义输出路径为'archive', 资源目录为'logs'	在'archive'文件夹中得到一个zip压缩包, 内容是logs的所有文件	通过	5.10	通过	
2	定义多个压缩任务,并定义一个压缩管理器, 把任务加入管理器, 开启所有任务	调用compressorManager 的 startAll	所有压缩任务正常开启而且表现正常	与预期一致	通过	5.10	无
3	定义多个压缩任务,并定义一个压缩管理器, 把任务加入管理器, 停止任务	调用compressorManager 的 stopAll	所有压缩任务正常关闭	同预期	通过	5.10	无
4	定义多个压缩任务,并定义一个压缩管理器, 把任务加入管理器, 开启指定任务	调用compressorManager 的 startTask	客压缩任务正常开启而且表现正常	同预期	通过	5.10	无
5	定义多个压缩任务,并定义一个压缩管理器, 把任务加入管理器, 关闭指定任务	调用compressorManager 的 stopTask	压缩任务正常关闭	同预期	通过	5.10	无

Client按次序接收所有遗漏的消息

- 需求： Client登录后, 有序接收所有遗漏的消息
- 用户信息

username	password	groupid
100	123456	1
101	123456	1
200	123456	2
201	123456	2
202	123456	2
300	123456	3

- 测试用例集：

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	同组用户能够按次序接收遗漏的所有消息	Client200登陆, 发送消息后, 201登陆并发送消息, 202登陆	201按次序收到200的消息, 202按次序收到200和201的消息	同预期结果	通过	2016.5.13	无
1	用户不会接受收其他组遗漏的消息	Client200登陆, 发送消息后, 100登陆	100不会收到消息	同预期结果	通过	2016.5.13	无

用户上下线提示功能

- 用户信息

username	password	groupid
100	123456	1
101	123456	1
200	123456	2
201	123456	2
202	123456	2
300	123456	3

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	同组用户能够查看同组在线的人	先登录101，再登录100	100上线时，100能够看到101已经在线，101看到100在线	与预期一致	通过	2016.05.18	无
2	同组用户下线时其他用户能够得到提示	100与101同时在线，101退出	100收到提示101下线，且100的界面中同组成员列表中会移除101	100确实收到了提示，但是100的界面中并没有移除101	未通过	2016.05.18	界面中相应部分没有进行更新

在bug修复后，又进行了相关的回归测试，回归测试通过

遗漏消息恢复功能

用例编号	用例描述	输入数据	预期结果	实际结果	测试结果	测试时间	bug描述
1	同组用户能够收到自己不在线时遗漏的消息	先登录101，发送一条消息，再登录100	100上线时100上线能够看到之前的一条消息	与预期一致	通过	2016.05.18	无

集成测试

编号	子系统名称	操作方法	测试结果	测试时间	bug描述
1	聊天子系统测试	功能验证（用户登录、同组间收发消息、保存收到的消息到文件）、边界条件测试、路径测试	测试通过	2016.05.13	无
2	PM子系统测试	接口测试、边界条件测试、路径测试	测试通过	2016.05.13	无
3	LICENSE子系统测试	接口测试、边界条件测试	测试通过	2016.05.13	无
4	CM子系统测试	接口测试、路径测试	测试通过	2016.05.13	无
5	数据库子系统测试	接口测试	测试通过	2016.05.13	无

系统测试

在真实环境下运行该项目进行测试

测试评估

单元测试覆盖率达到95%，功能测试覆盖率为100%。在测试过程中只发现了一个bug，出现在用户上下线提示功能中，经过修复后，重新进行相关的回归测试，回归测试通过。从而保证了软件产品的质量，通过需求跟踪，确保软件产品满足用户的需求。