MeetHere 单元测试计划书

文档作者：刘畅 日期：2020/6/23

审核 ：刘畅 日期：2020/6/23

批准 ：刘畅 日期：2020/6/23

# 引言：

## 目的

本文档是为MeetHere项目的单元测试提供范围、方法、资源和进度方面的指导

本文档的主要读者是开发（测试）经理、测试人员和开发人员。

## 1.2、测试策略

以类为单元，采用独立的单元测试策略，通过设计相应的Driver和使用mock来进行次要测试对象的替代。

## 1.3、范围

单元测试主要包含计划阶段、设计阶段和执行阶段3个阶段。具体的进度安排表在之后会说明

# 二、任务概述

## 2.1、人员安排

|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 具体职责 | 人员技能要求 | 人员安排 |
| 测试设计人员 | 1、设计测试方案  2、编写测试用例 | 熟悉被测试对象如函数或模块的功能，以及该函数或模块与其他函数或模块的关系。熟练掌握编程语言，能够根据测试需求编写测试代码。 | 刘畅 |
| 测试执行人员 | 1、执行测试用例  2、输出测试报告 | 熟练掌握编程语言，能够执行单元测试用例，能够领会测试人员的意图，准确记录测试结果。 | 刘畅 |
| 测试经理 | 1、评估单元测试有效性  2、对单元测试具有否决权 | 掌握测试理论及软件工程的基本知识。能够评估单元测试，指出单元测试的不足之处。 | 刘畅 |

## 2.2、测试环境

|  |  |
| --- | --- |
| 硬件平台 | Intel® Core(TM) i7-7700HQ |
| 网络环境 | 单机测试 |
| 操作系统 | Windows 10 |
| 代码语言 | Java |
| 测试平台 | IntelliJ IDEA |
| 测试工具 | Junit、Mockito、SpringBootTest |

## 2.3、测试项目及内容

2.3.1、UserSerciveImpl类测试

|  |  |  |
| --- | --- | --- |
| 方法标识符 | 方法名 | 测试关注点 |
| Unit\_001\_Function\_001 | User findByUserID(String userID) |  |
| Unit\_001\_Fuction\_002 | User findById(int id) |  |
| Unit\_001\_Fuction\_003 | Page<User> findByUserID(Pageable pageable) |  |
| Unit\_001\_Function\_004 | User checkLogin(String userID, String password) |  |
| Unit\_001\_Fuction\_005 | int create(User user) |  |
| Unit\_001\_Fuction\_006 | void delByID(int id) |  |
| Unit\_001\_Fuction\_007 | void updateUser(User user) |  |
| Unit\_001\_Fuction\_008 | int countUserID(String userID) |  |

2.3.2、UserController类测试

|  |  |  |
| --- | --- | --- |
| 方法标识符 | 方法名 | 测试关注点 |
| Unit\_002\_Function\_001 | String signUp() |  |
| Unit\_002\_Fuction\_002 | String login() |  |
| Unit\_002\_Fuction\_003 | String login(String userID, String password, HttpServletRequest request) |  |
| Unit\_002\_Function\_004 | void register(String userID, String userName, String password, String email, String phone, HttpServletResponse response) |  |
| Unit\_002\_Fuction\_005 | void logout (HttpServletRequest resquest, HttpServletResponse response) |  |
| Unit\_002\_Fuction\_006 | void quit (HttpServletRequest resquest, HttpServletResponse response) |  |
| Unit\_002\_Fuction\_007 | void updateUser(String userName, String UserId, String passwordNew,String email, String phone, MultipartFile picture,HttpServletRequest request, HttpServletResponse response) |  |
| Unit\_002\_Fuction\_008 | boolean checkPassword(String userID, String password) |  |
| Unit\_002\_Function\_009 | String user\_info(Model model) |  |

2.3.3、VenueServiceImpl类测试

|  |  |  |
| --- | --- | --- |
| 方法标识符 | 方法名 | 测试关注点 |
| Unit\_003\_Function\_001 | String venue\_magage(Model model) |  |
| Unit\_003\_Fuction\_002 | String editVenue(Model model, int venueID) |  |
| Unit\_003\_Fuction\_003 | String venue\_add() |  |
| Unit\_003\_Function\_004 | List<Venue> getVenueList() |  |
| Unit\_003\_Fuction\_005 | int create(Venue venue) |  |
| Unit\_003\_Fuction\_006 | void update(Venue venue) |  |
| Unit\_003\_Fuction\_007 | void delById(int id) |  |
| Unit\_003\_Fuction\_008 | int countVenueName(String venueName) |  |

2.3.4、AdminVenueController类测试

|  |  |  |
| --- | --- | --- |
| 方法标识符 | 方法名 | 测试关注点 |
| Unit\_004\_Function\_001 | Venue findByVenueID(int id) |  |
| Unit\_004\_Fuction\_002 | Vunue findByVenueName(String venueName) |  |
| Unit\_004\_Fuction\_003 | Page<Venue> findAll(pageable pageable) |  |
| Unit\_004\_Function\_004 | List<Venue> findAll(@RequestParam(value = "page",defaultValue = "1")int page) |  |
| Unit\_004\_Fuction\_005 | void addVenue(String venueName, String address, String description, int price, MultipartFile picture, String open\_time, String close\_time, HttpServletRequest request, HttpServletResponse response) |  |
| Unit\_004\_Fuction\_006 | void modifyVenue(int venueID,String venueName, String address, String description, int price, MultipartFile picture, String open\_time,String close\_time, HttpServletRequest request, HttpServletResponse response) |  |
| Unit\_004\_Fuction\_007 | boolean delVenue(int venueID) |  |
| Unit\_004\_Fuction\_008 | boolean checkVenueName(String venueName) |  |

# 三、测试方法

保证所有语句、分支被覆盖

参考等价类划分方法

参考边界值分析方法

参考使用错误猜测方法

测试脚本使用java语言实现，使用Junit，mockito作为单元测试。

# 四、测试通过/失败标准

通过标准：

所有的单元测试的用例都被执行并通过

所有发现的缺陷都被修正并通过回归测试

所有被测对象的前置条件和后置条件组合覆盖率达到100%，或能明确给出不需要达到的理由

失败标准

严重缺陷密度大于15个/KLOC

发现软件结构有重大设计问题，其修改会导致20%以上的接口、功能、数量的变化

发现关键功能未被设计

# 五、测试进度安排表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 任务 | 开始时间 | 结束时间 | 可交付物 |
| 1 | 编写单元测试计划 | 2020/6/20 | 2020/6/21 | 单元测试计划书 |
| 2 | 编写单元测试用例 | 2020/6/21 | 2020/6/23 | 单元测试代码 |
| 3 | 执行单元测试并撰写结果报告 | 2020/6/23 | 2020/6/25 | 单元测试结果报告 |