
FEEDADEADF15h

Schedule Tour
第一次测试计划

版本 1.0

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

修订历史记录

日期	版本	说明	作者
17/7/2013	1.0	第一次测试计划原稿	孟繁宇

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

目录

1. 简介.....	4
1.1 目的.....	4
1.2 背景 针对软件学院 2011 级暑假大作业 Schedule Tour 项目的第一、第二次迭代进行的测试	4
1.3 范围.....	4
2. 测试环境.....	6
3. 测试需求.....	6
3.1 用例测试.....	6
3.2 功能性需求测试.....	6
3.3 非功能性需求测试.....	6
4. 测试策略.....	7
4.1 用例测试.....	7
4.1.1 注册新用户.....	7
4.1.2 用户登录.....	7
4.1.3 用户浏览事件安排.....	7
4.1.4 用户通过地图添加事件.....	7
4.1.5 用户通过时间表添加事件.....	8
4.1.6 用户删除事件.....	8
4.1.7 展示不同地点间的路线安排.....	8
4.1.8 优化日程安排.....	8
4.2 功能性需求测试.....	8
4.2.1 数据和数据库完整性测试.....	8
4.2.2 信息推送测试.....	9
4.3 非功能性需求测试.....	9
4.3.1 性能测试.....	9
4.3.2 界面测试.....	9
4.3.3 易用性测试.....	10
5. 资源.....	11
5.1 角色.....	11
6. 项目里程碑.....	13
7. 可交付工件.....	14
7.1 测试用例.....	14
7.2 测试日志.....	14
7.3 缺陷报告.....	14
8. 附录 A: 项目任务.....	15

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

测试计划

1. 简介

1.1 目的

Schedule Tour 的这一“测试计划”文档有助于实现以下目标：

检验并核收前两次迭代中所实现的成果

检验相关模块是否已经实现了非功能性需求

列举测试结果供编码人员参考，修改代码中存在的错误

1.2 背景

针对软件学院 2011 级暑假大作业 Schedule Tour 项目的第一、第二次迭代进行的测试

1.3 范围

测试的主要类型为功能测试、性能测试

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

项目标识

下表列出了制定 *测试计划* 所用的文档，并标明了文档的可用性：

文档 (版本/日期)	已创建或可用	已被接受或已 经过复审	作者或来源	备注
需求规约	√ 是 <input type="checkbox"/> 否	√ 是 <input type="checkbox"/> 否	ST	
功能性规约	√ 是 <input type="checkbox"/> 否	√ 是 <input type="checkbox"/> 否	ST	
项目计划	√ 是 <input type="checkbox"/> 否	√ 是 <input type="checkbox"/> 否	ST	
原型	√ 是 <input type="checkbox"/> 否	√ 是 <input type="checkbox"/> 否	ST	
用户手册	<input type="checkbox"/> 是 √ 否	<input type="checkbox"/> 是 √ 否	--	
业务模型或业务流程	√ 是 <input type="checkbox"/> 否	√ 是 <input type="checkbox"/> 否	ST	
数据模型或数据流	<input type="checkbox"/> 是 √ 否	<input type="checkbox"/> 是 √ 否	--	

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

2. 测试环境

软件平台	操作系统	客户端: Android 4.0 Windows 8 64bits
		服务器: Windows 7 32 位旗舰版
硬件平台	客户端	Windows 平板电脑 摩托罗拉 MT788 智能手机
	客户端浏览器	Chrome 28
	服务器	联想笔记本
	CPU	服务器: 2.30 GHz
	RAM	服务器: 4G

3. 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

3.1 用例测试

1. 注册新用户
2. 用户登陆
3. 用户浏览事件安排
4. 用户通过地图添加事件
5. 用户通过时间表添加事件
6. 用户删除事件
7. 展示不同事件地之间路线安排
8. 优化用户日程安排

3.2 功能性需求测试

1. 数据和数据库完整性测试
2. 消息推送测试

3.3 非功能性需求测试

1. 性能测试
2. 界面测试
3. 易用性测试

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

4. 测试策略

4.1 用例测试

4.1.1 注册新用户

测试目标:	能保证该软件新用户能正确地注册账户
方法:	用户进入注册新用户界面, 输入用户名和密码, 完成注册操作
完成标准:	对于合法的注册用户请求, 系统能够正确地进行注册操作并在数据库中添加相应的用户信息, 否则返回相应的错误信息
需考虑的特殊事项:	注册时的用户名已经被注册 用户的用户名或密码中含有特殊字符 用户的用户名或密码过长或过短

4.1.2 用户登录

测试目标:	能保证用户正确登陆系统
方法:	用户在登陆时输入用户名密码, 完成登陆操作
完成标准:	对于合法的登陆请求, 系统能够正确地进行注册操作并在数据库中添加相应的用户信息, 否则返回相应的错误信息
需考虑的特殊事项:	用户名密码错误 该用户尚未注册 尝试采用数据库注入的方式登陆系统 登陆 Session 的保留时间

4.1.3 用户浏览事件安排

测试目标:	能保证用户顺利地浏览实践安排
方法:	用户能通过传统日程表, 浏览最近事务的安排
完成标准:	能够正确地显示想要浏览的时间范围内的事务
需考虑的特殊事项:	用户名尚未登陆便可以浏览事件安排 用户查询一个历史的日程安排并尝试修改它

4.1.4 用户通过地图添加事件

测试目标:	能保证用户在地图上添加事件
方法:	用户能在地图上, 通过点击的方式, 添加事件
完成标准:	能够在正确的时间与地点, 进行事件的添加
需考虑的特殊事项:	用户尚未登陆便可以添加事件 用户尝试在一个不合法的地点(时间)添加事件 用户在同一个地点添加多个事件 用户尝试在同一个时刻安排多个事件

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

4.1.5 用户通过时间表添加事件

测试目标:	能保证用户在时间表上添加事件
方法:	用户能在时间表上, 通过点击的方式, 添加事件
完成标准:	能够在正确的时间与地点, 进行事件的添加
需考虑的特殊事项:	用户尚未登陆便可以添加事件 用户尝试在一个不合法的地点(时间)添加事件 用户在同一个地点添加多个事件 用户尝试在同一个时刻安排多个事件

4.1.6 用户删除事件

测试目标:	能保证用户删除相应的已安排事件
方法:	用户能在相关的坐标图上, 通过点击的方式, 删除事件
完成标准:	能够正确地删除指定的事件
需考虑的特殊事项:	用户尚未登陆便可以删除事件 用户尝试删除一个不存在的事件 用户删除事件后, 在其它视图中能否同步更新 用户尝试在同一个时刻安排多个事件

4.1.7 展示不同地点间的路线安排

测试目标:	能保证用户能实时浏览一天的路线行程安排
方法:	用户能在地图上, 浏览不同事件地之间的路线规划
完成标准:	能够实时地更新从一个事件点到另一个事件点间的最优路径
需考虑的特殊事项:	用户实时移动时路线的自动更新 用户添加或删除事件后的路线最优性

4.1.8 优化日程安排

测试目标:	能保证用户的日程安排能够最有效地利用时间
方法:	系统能根据用户一天的日程计划, 适当地优化一天的日程安排
完成标准:	用户的空闲时间能得到最大化的利用
需考虑的特殊事项:	有多个可选事件时的日程筛选 优化后日程安排的合理性

4.2 功能性需求测试

4.2.1 数据和数据库完整性测试

测试目标:	确保数据库的访问, 修改等操作能够正常执行, 数据不会被认为损坏
-------	----------------------------------

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

方法:	调用 Mongo DB 中各种能访问到数据的方法, 检查数据与应用程序是否保持一致; 尝试恶意篡改相应的数据, 查看是否存在相应的保护措施
完成标准:	数据库访问逻辑能够按照预期进行, 相应的数据并没有产生不应有的损坏等现象
需考虑的特殊事项:	能否正确处理数据注入的问题 大规模并行调度时的数据正确性 在数据正确性有保障的情况下, 数据库的吞吐量 (非功能需求)

4.2.2 信息推送测试

测试目标:	确保相应的推送消息能够发送到客户端
方法:	在服务器端向客户端发送推送消息, 查看客户端是否能够对其进行响应
完成标准:	客户端能够收到服务器端预期推送的消息
需考虑的特殊事项:	客户端没有收到推送消息时服务器端的处理方法 客户端能否将推送消息按照预期进行排序

4.3 非功能性需求测试

4.3.1 性能测试

测试目标:	确保程序事务的响应时间控制在预期范围内
方法:	测试单独用户在操作软件的过程中不同操作之间的响应时间 测试多个用户并行执行多个操作时, 每个操作之间的响应时间
完成标准:	在所有操作保证正确性的前提下, 每个操作的响应时间不超过 3s
需考虑的特殊事项:	用户在一个操作的相应等待时间内, 同时进行了另一个操作 用户在两个不同的操作之间, 操作执行先后的正确性

4.3.2 界面测试

测试目标:	确保软件 UI 界面对用户简洁、友好
方法:	查看不同按钮、手势之间的操作是否能够正确执行, 文字描述是否明确简洁, 相应工作区域的大小是否合理
完成标准:	所有按钮、工作区域都能够正确地显示
需考虑的特殊事项:	在不同客户端的界面是否能保持一致性

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

4.3.3 易用性测试

测试目标:	确保一名新用户在接受简单的培训后能快速学会该软件的操作与使用
方法:	寻找一名从前没有使用过该软件的用户，对其进行简单培训，观察其用多长时间会学会该软件的基本操作功能
完成标准:	用户能在三分钟左右学会基本操作
需考虑的特殊事项:	该用户最好对日程管理有一定的了解，并且曾经使用过类似于记事本之类的日程管理软件

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

5. 资源

5.1 角色

下表列出了在此项目的人员配备方面所作的各种假定。

人力资源		
角色	推荐的最少资源 (所分配的专职角色数量)	具体职责或注释
测试经理, 测试项目经理	孟繁宇	进行管理监督。 职责: <ul style="list-style-type: none"> 提供技术指导 获取适当的资源 提供管理报告
测试设计员	孟繁宇、董明凯	确定测试用例、确定测试用例的优先级并实施测试用例。 职责: <ul style="list-style-type: none"> 生成测试计划 生成测试模型 评估测试工作的有效性
测试员	孟繁宇	执行测试。 职责: <ul style="list-style-type: none"> 执行测试 记录结果 从错误中恢复 记录变更请求
数据库管理员	邱喆	确保测试数据(数据库)环境和资产得到管理和维护。 职责: <ul style="list-style-type: none"> 管理测试数据(数据库)
设计员	张凯源	确定并定义测试类的操作、属性和关联。 职责: <ul style="list-style-type: none"> 确定并定义测试类 确定并定义测试包

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

实施员	张凯源	<p>实施测试类和测试包，并对它们进行单元测试。</p> <p>职责：</p> <ul style="list-style-type: none"> • 创建在测试模型中实施的测试类和测试包
-----	-----	--

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

6. 项目里程碑

里程碑任务	工作量	开始日期	结束日期
制定测试计划	1	2013. 7. 17	2013. 7. 18
设计测试	1	2013. 7. 18	2013. 7. 18
实施测试	1	2013. 7. 19	2013. 7. 19
执行测试	2	2013. 7. 19	2013. 7. 19
评估测试	2	2013. 7. 19	2013. 7. 20

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

7. 可交付工件

7.1 测试用例

测试结束后，将把测试用例提交给编码人员

7.2 测试日志

测试结束后，将发布测试结果以及软件在整个测试过程中的整个系统状态

7.3 缺陷报告

测试结束后，将没有达到预期目标的测试结果告知编码组，以供其参考

Schedule Tour	Version: 1.0
第一次测试计划	Date: 7/17/2013

8. 附录 A: 项目任务

以下是一些与测试有关的任务:

- 制定测试计划
 - 确定测试需求
 - 评估风险
 - 制定测试策略
 - 确定测试资源
 - 创建时间表
 - 生成测试计划
- 设计测试
 - 准备工作量分析文档
 - 确定并说明测试用例
 - 确定并结构化测试过程
 - 复审和评估测试覆盖
- 实施测试
 - 记录或通过编程创建测试脚本
 - 确定设计与实施模型中的测试专用功能
 - 建立外部数据集
- 执行测试
 - 执行测试过程
 - 评估测试的执行情况
 - 恢复暂停的测试
 - 核实结果
 - 调查意外结果
 - 记录缺陷
- 评估测试
 - 评估测试用例覆盖
 - 评估代码覆盖
 - 分析缺陷
 - 确定是否达到了测试完成标准与成功标准