**东 北 林 业 大 学 2021 届 本 科 毕 业 设 计**

**开 题 报 告**

设 计 题 目： 面向GUI的测试项目与事务跟踪系统开发

学 生： 于丽蔷 指 导 教 师： 李 莉 副教授

专业（年级、班级）： 软件工程 2017 级 4 班 学 院： 信息与计算机工程学院

2020 年 12 月 15 日

|  |  |
| --- | --- |
| 选题依据 （选题经 过，现状动  态，初步设想及创新点等）及可行性论述 | **1.选题依据**  多年来，软件工程一直致力于解决软件危机问题，如软件开发周期长、成本高、质量差、维护困难等，软件工程采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，以经济地开发出高质量的软件并有效地维护软件。其中，软件测试作为软件生命周期中一个不可或缺的阶段，针对软件质量差的问题提出了解决方案，对应用软件的确认测试工作，测试人员需要从系统级的软件需求规格说明中构造测试用例，并依据测试用例对软件进行确认测试，以发现软件中尽可能多的故障，使软件具有高可靠性，高可信性。  从软件本身的质量保证而言，软件的可靠性主要取决于两方面，一是软件开发的方法与过程，二是软件产品的测试与验证。测试用例是测试工作的实际操作纲要，也是评估测试结果的度量基准，高质量的设计用例能有效的保障软件测试的质量。软件测试用例设计的难点在于：对状态变量繁多，状态变量之间存在复杂的耦合情况的系统，通过人工构造测试用例时，往往出现考虑不周，设计不全的问题，因此无法通过非形式化的人工审查的方式保证设计出来的确认测试用例对需求进行了充分的覆盖。  本选题针对目前软件测试过程中，测试用例的设计和管理进行项目开发，其中测试用例的管理方便测试用例评审和复用，通过人工评审方式尽可能提高覆盖率，同时在后续同类型业务的测试中，通过复用减少工作量，同时也避免因当时考虑不周而遗漏测试点的情况。除此之外，可以根据已有的科学的测试理论，实现基于软件需求的测试用例自动生成功能，将部分人工工作量转化为自动工作量，提高测试工作效率，降低人为因素对测试过程的干扰，降低基本测试用例的冗余性，减少遗漏，排除测试的随机性和盲目性。  **2.现状分析**  在国外一些软件行业发达的国家，软件测试技术已经发展了非常长的时间，也相对国内得到了更多的重视，技术日益成熟。在一些大型的软件系统开发公司，测试人员在其员工中占有相当大的比重。各种测试软件、自动化测试工具应运而生。软件测试在技术方面也在不断的高提，致力于迈向于通用化、标准化、网络化、自动化的方向。目前国外同类软件主要有 Rational 公司的 SQA Manager 产品，它是 SQA Suite测试软件包的一部分，该软件包以测试工具 SQA Robot 和 SQA LoadTest 为主。SQA Manager 一般用做和测试工具的结合使用，为英文系统，因此产生的各类报告格式西化，没有测试案例具体步骤的管理查询，而且是以客户端的形式呈现的。在产品定位上，面向高端客户，价格昂贵，SQA Manager 作为 SQA Suite 软件包的一部分捆绑出售，不能单卖，用户购买该软件包后经常只需用其中一部分功能，造成不必要的开销。  我国的软件测试技术于上世纪八十年代起源，并伴随着软件工程领域的研究发展而发展。近年来，国内软件行业发展也越来越迅速，但是软件测试技术发展却相对缓慢。国内软件测试水平在国际上也还是属于较年轻类型，具体表现在对软件测试还不够重视，测试单一化，质量监督体系不够完备，自动化程度不够高等方面，软件测试人才缺口也是非常大。一些中小型的企业没有形成较为完备的测试管理体系，导致在生产出的产品质量无法达到期望的水准，错误率也比较高。当然现今也有很多企业由起初的“重研发，轻测试”逐渐转变，软件测试的地位也逐渐提高，会有逐步完备的测试管理体系，不断改进的自动化测试工具，日益成熟的测试技术，软件测试将得到更多重视。目前国内使用比较普遍的是 i-Test 管理系统，是由中科软科技股份有限公司开发的，在笔者实习过程中，公司测试相关工作采用的是 i-Case 管理系统。上述系统主要针对测试用例的增删改查、导入导出等功能提供了解决方案，具有一定通用性，但功能单一，且手工录入测试用例工作量大，可能导致覆盖率降低；执行状况记录功能没有得到重视，不能及时更新执行结果，会导致测试的随机性和盲目性。  **3.初步设想**  根据现状分析来看，基于测试人员的需求开发一个具有可视化界面的测试项目与事务跟踪系统显得颇为重要。  一方面，该系统具备已有的测试用例增删改查和导入导出功能，完善执行结果的记录功能；另一方面，系统也会考虑减少测试人员工作量，提供基础测试用例集，根据需求文档分词自动生成相应测试用例的功能，涉及自动化测试，会设计类似GitHub功能的自动化框架代码管理模块，除此之外，考虑到项目协作，系统还会包含用户管理、项目管理、需求管理、版本管理、缺陷管理、进度管理等模块。后续在技术允许的情况下，会增加测试用例覆盖率的自动化分析以及通过深度学习进行自动化生成测试用例的迭代优化。  基于上述需求，计划采用B/S架构完成该系统设计。    **系统功能结构图**  **3.1预期系统用户**  软件开发自测人员、软件测试人员。  **3.2预期业务功能**   1. **用户管理模块：**   本管理系统主要包括三类用户，管理员、开发人员与测试人员。  管理员使用的主要功能有：维护用户信息，添加人员，分配权限，统计和分析相关数据。  开发人员使用的主要功能有：维护个人信息，处理缺陷报告，跟进项目进度。  测试人员使用的主要功能有：维护个人信息，创建项目计划，跟进项目进度，导入开发需求，维护测试用例库，执行测试录入结果，提交缺陷报告，处理相关问题。   1. **项目管理模块：**   项目管理是该系统的基础模块之一，其中新增、修改、删除功能的权限为系统管理员分配给普通用户，无权限只能对项目进行查看和查找功能。该模块包括以下几个子功能：  新建一个项目，新建项目的基本信息包括项目名称、项目简介、项目状态、启动时间、交付时间、项目负责人、项目备注等信息，用于后期新建测试需求、测试计划等时，选择添加该项目；  项目状态管理，包括启动项目、关闭项目、暂停项目等，可以自定义添加并管理，并且将已完结项目做归档处理，查询页面默认显示未归档项目；  修改项目，用于对已存项目的基本信息及状态进行修改，并可添加附件、删除附件等；  删除项目，对无效项目可做删除操作，并校验非空项目不可删除，做安全删除操作，需清空项目下内容方可删除；  在不同项目下，可查看项目的详情，并且有对其版本的控制管理，及对版本的增加、删除、修改、查看功能。   1. **需求管理模块：**   需求管理模块主要包括需求点的整理，方便后续开发活动以及自动生成测试用例。该模块进本操作如下：  1) 用户点击测试需求模块时，菜单中显示的是所有项目列表，默认展示第一个项目中的所有需求；  2) 新建需求：测试人员点击新建需求，或者批量导入需求；弹出面板，选择项目，然后根据提示，输入测试需求的基本信息，字段包括需求的概述（编写目的、项目背景、术语定义、文档约定、测试范围等）、测试任务概述、系统特性 、版本、负责人、状态、数据性质、功能测试要求、安全测试要求、文件传输及其他要求等；完成后点击确认需求创建成功。或者直接导入需求文本，文本中添加项目名称，自动归属到该项目下，导入成功后则需求创建成功；  3) 修改需求：对不同需求，点击修改需求，则能对需求所包含的字段进行修改操作；  4) 查找需求：查看支持 ID 查询和关键字查询，并支持模糊匹配，输入框搜索字段能查找出对应需求，并可查看到详细信息；  5) 导出需求报告：当前页面的所有需求，可一键导出，也可指定导出个别需求，导出的需求以 excel 的形式存储；  6) 删除操作：点击删除按钮，删除当前需求，支持批量删除；  **D.用例管理模块：**  软件工程中的测试用例是一组条件或变量，测试者根据它来确定应用软件或软件系统是否正确工作。测试用例一般包括测试用例编号、标题、测试用例᧿述、测试步骤编号或测试顺序编号、相关依赖（脚本、数据等）、测试分类、负责人等信息，测试完成后还应当追加测试状态，包括是否通过和必要的备注信息以及关联缺陷报告等。该模块包括测试用例的新建（包括手动创建和自动创建）、批量导入测试用例、测试用例的修改、导出测试用例、删除测试用例、测试用例状态查询、测试用例关联缺陷报告等子功能。  1) 测试人员打开测试用例模块，目录中显示为所有项目列表，项目下为测试需求，测试需求中包含子需求或者测试用例；根目录可以显示所有的测试用例。  2) 测试人员点击项目名称，右侧罗列出所有该项目下的测试需求及子需求，需求下包含了测试用例，根据选择，测试用例在与右侧列表中可按规则排序或按规则筛选出所需的测试用例，可查看所有，也可通过 ID 或者名称关键字进行查询，并点击查看详细信息；  3) 当新建测试用例时，会按规则自动生成一个唯一的 ID，便于查询和唯一标识，然后依次填入标题、测试用例描述、测试步骤编号或测试顺序 编号、相关依赖（及上传包括脚本、数据等的附件）、测试分类、负责人、是否需要数据等信息，并确认新建，信息完整性校验之后，则新建成功；  4) 点击修改时，可以修改除 ID 以外的其他信息，当测试用例状态改变时，可以修改状态，测试用例的状态包括 pass、fail、block，以及提交相关缺陷报告的链接，实现直接跳转；  5) 当测试用例过时、或者失效的情况下，则可直接删除，包括批量操作，保证测试用例不冗余。  **E.版本管理模块：**  本模块针对项目的迭代需要进行回归测试这一现实任务，可通过版本进行分类，针对每一版本的回归测试，存储当前版本下的测试用例及其执行情况，以及自动化框架代码进行的相应修改管理。   1. **缺陷管理模块：**   本模块针对软件测试结果失败的情况，需要提交缺陷报告给开发人员，再由开发人员进行修复这一现实任务，可进行测试用例和缺陷报告的关联，测试人员发现问题后新建相应缺陷报告，通过链接与测试用例互相关联，开发人员解决完具体问题后，可以查看失败测试用例进行自测，避免了过程中因复现问题浪费时间，影响工作效率；除此之外，可以随时跟进缺陷问题进度，方便人员协调推进项目进度。   1. **进度管理模块：**   本模块针对软件生命周期中的各项任务，录入项目进度的整体计划安排，主要包括时间安排、人员安排等，录入相关信息后，可根据排期进行进度推进，并且测试和开发人员可以在这部分信息中互相找到对应负责人，提高合作效率。  **3.3 预期架构实现**  网络技术的日益成熟，使得 C/S 架构似乎已经达不到现今信息化的水准了。B/S 架构作为 C/S 架构的改进与升级，已经呈现出要取而代之之势。B/S 应用越来越广，目前大部分的网站，特别是电商网站，都采用了 B/S 架构而构建。B/S 架构使我们不再需要开发出一个客户端软件，这便利了系统的维护和版本的升级；可跨平台操作，不再考虑不同操作系统开发的不同程序，只要安装浏览器软件，就可以作为客户端来访问系统；安全性好，防火墙保证了系统的安全性。在 B/S 体系结构系统中，用户通过浏览器向分布在网络上的许多服务器发出请求，服务器对浏览器的请求进行处理，将用户所需的信息返回到浏览器，B/S 结构简化了客户机的工作，客户机上只需配置少量的客户端软件。结构模型如下图所示：    **B/S 两层结构模型图**  鉴于以上优势，本设计选择了 B/S 架构。  **4.创新点**  该系统基于GUI可视化界面，打造一款面向测试人员和开发自测人员的测试用例管理平台。创新点如下：   1. 业务功能上，从当前实际情况出发，为减少测试工作量，可自动生成基础测试用例，用户在节省录入测试用例的时间的同时，不会遗漏基础测试点；开发人员可以通过基础测试点进行自测，通过后进行提测。提供对测试用例执行结果的统计分析。测试结果以图表形式生动、直观地反映测试用例的执行结果，为下次回归测试筛选测试用例做好充分的准备。   2、模块上，系统包含完整、规范的系统业务流程，制定了规范的软件测试流程，引导项目参与人员进行有效地软件测试，达到提高软件质量的最终目标。针对不同用户开放不同的功能权限，满足管理员开发人员、测试人员的需求，使系统易于管理和维护，项目进度可以精准跟进。  3、系统设计上，设计详细准确的测试用例模板，各模块之间通过测试用例关键词建立关联，使模块之间的耦合度满足实际需要又不影响各模块内的高聚合。管理员可以随时了解测试集的测试状态，对未通过的测试用例以问题报告的形式反馈给开发人员，开发人员修改问题后，重新分配给测试人员重新测试，提高了不同用户之间的并行工作效率。  4、核心算法：  **（1）分词算法HanLP：**用于文档分析，根据分词结果生成测试点集。  构建词网，将词网转化为词图，词图含有各条边以及边上的权值，接下来采用维特比分词器—基于动态规划的Viterbi算法，算法可以概括为下面三点：  ①如果概率最大的路径经过网络的某点，则从开始点到该点的子路径也一定是从开始到该点路径中概率最大的。  ②假定第i时刻有k个状态，从开始到i时刻的k个状态有k条最短路径，而最终的最短路径必然经过其中的一条。  ③根据上述性质，在计算第i+1状态的最短路径时，只需要考虑从开始到当前的k个状态值的最短路径和当前状态值到第i+1状态值的最短路径即可，如求t=3时的最短路径，等于求t=2时的所有状态结点x2i的最短路径加上t=2到t=3的各节点的最短路径。  **（2）字符串匹配算法KMP：**用于关键字查询、匹配测试点集自动生成测试用例。  KMP算法一种改进的模式匹配算法，它的改进在于：每当从某个起始位置开始一趟比较后，在匹配过程中出现失配，不回溯i，而是利用已经得到的部分匹配结果，将一种假想的位置定位“指针”在模式上向右滑动尽可能远的一段距离到某个位置后，继续按规则进行下一次的比较。  算法流程：  ①规定i是主串S的下标，j是模式T的下标。现在假设现在主串S匹配到 i 位置，模式串T匹配到 j 位置。  ②如果j = -1，则i++，j++，继续匹配下一个字符；  ③如果S[i] = T[j]，则i++，j++，继续匹配下一个字符；  ④如果j != -1，且S[i] != P[j]，则 i 不变，j = next[j]。此举意味着失配时，接下来模式串T要相对于主串S向右移动j - next [j] 位。  **5.可行性分析**  **5.1.技术可行性分析**  1、本系统的主要作用是进行测试用例管理和根据文本生成用例，对于测试点的挖掘可以使用哈工大提供的 HanLP 库的 API 对大量需求文本进行分词处理，再从中提取出常见的测试点，对测试用例的存储可以使用MySQL。  2、系统本身基于 Java 实现，平台基于 Web 实现，不存在技术盲区。  **5.2.用户可行性分析**  基于针对身边开发人员和测试人员的调查，众多用户期待能够有一款易于使用，使用直观，界面友好，且具备一定自动化便捷功能的测试用例管理平台。  系统的用户群体为软件测试开发工程师，因此界面只需要清晰便捷即可，使用 Vue.js 和 Bootstrap 等前端主流框架可以完成。  **5.3.经济可行性分析**  （1）硬件成本  正常的具有开发环境的笔记本电脑的性能足够完成该系统。  （2）人力成本  本项目的实现阶段由设计者个人独立完成，并且有足够条件在预期的时间内完成。  （3）经济成本  系统开发阶段无经济成本，在试运行阶段则需要租借服务器保证项目正常使用，不过该经济成本在可接受的范围之内，不影响开发。  **5.4.结论意见**  该研究在技术、经济、用户都具备足够的可行性，系统可进行开发。 |
| 设计撰写过程中拟采取的设计方案、方法、手段或实验方案、实验路线等 | **6.技术路线**  **在开发此平台中使用微服务架构，并基于 B/S 架构进行展现**  系统本身基于微服务架构进行实现，同时具体成果借助微信小程序客户端和 PC 管理端  进行体现。  **开发语言：**Java、JavaScript、Python  **开发框架：**Spring Boot、Vue.js、Bootstrap、Nginx、Mabatis  **数据库：**MySQL  **开发工具：**IntelliJ IDEA、Pycharm、MySQLWorkbench  **软件周期实验路线**：根据进度安排，优先开发核心模块，选用增量模型  IMG_256  **软件周期实验路线图**  可行性研究：基于用户需求进行初步的可行性分析与调研，例如调查是否存在相应  数据源、技术上是否可行、是否符合上线运营规范等。  需求分析：可行性研究完成后，若可行，则针对具体的用户群体进行用户调研，确定系统的最终需求。  概要设计：基于用户的需求对平台的架构与功能模块等进行设计。  功能模块算法实现：针对较为独立的模块，基于相关技术对相关算法与评价体系进  行独立研究。  详细设计：基于核心功能模块算法，对主要功能进行详细设计。  系统开发：对所有功能进行集成、编码和开发。  测试：对系统的关键算法与模块进行测试，重点测试相关算法的精确性，改进算法  的改进程度、应用模块的逻辑严谨性等。  试运行与部署：所有开发步骤完成后，基于阿里云服务器对项目进行部署并进行试  运行。 |
| 计  划  进  度  及  其  内  容 | 2020 年 12 月 01 日～2020 年 12 月 31 日 撰写开题报告。  2021 年 01 月 01 日～2021 年 01 月 10 日 需求分析。  2021 年 01 月 11 日～2020 年 01 月 20 日 概要设计与详细设计。  2021 年 01 月 21 日～2021 年 02 月 28 日 测试用例管理系统基本框架搭建。  2021 年 03 月 01 日～2021 年 03 月 20 日 核心业务的技术分析与实现。  2021 年 03 月 21 日～2021 年 03 月 25 日 软件测试。  2021 年 03 月 26 日～2021 年 03 月 31 日 软件上线试运行与调试。  2021 年 04 月 01 日～2021 年 05 月 15 日 完成毕业设计（论文）撰写。  2021 年 05 月 16 日～2021 年 06 月 准备进行毕业答辩。 |
| 写  作  提  纲 | 1.绪论：论述本文研究问题提出的背景、目的和意义，突出本文所要研究和解决的问题  2.技术分析：自动生成测试用例实现难点分析及研究  3.系统需求分析：进行系统功能需求分析、可采用的相关技术分析等。  4.系统设计：本部分主要包括功能模块概要设计、详细设计、数据库设计以及算法设计。  5.系统功能实现：包括系统采用的开发工具、典型程序代码和典型系统运行界面。  6.测试：包括单元测试、集成测试、系统测试以及压力测试。  7.结论：对系统开发工作效果的评价与展望。 |
| 指导  教师  意 见 | 签名：  年 月 日 |
| 专业  意见 | 签名：  年 月 日 |

注：纸张填写不够可另加附页。