软件测试设计服务平台

软件工程 杨修加 指导教师 刘琴

【摘要】如今，随着软件开发技术的发展，开发一款软件所需的周期、技术门槛也相对下降，但与此同时，软件的用户对软件使用的功能及体验要求却越来越高。在此矛盾下，软件测试变得相对越来越重要了。本文研究开发一个软件测试设计服务平台，通过这个平台，整合测试资源及需求资源。要点有：1）用户功能模块，为需求及服务提供双方用户提供一个具有信息发布及获取功能的交易平台。2）支付模块，通过在线支付方式进行交易，确保资金流动的安全。3）后端接口同时适配移动端，为今后移植到移动端做好兼容性工作。4）后台，能够查看到交易双方信息，方便处理用户申诉等需求。

【关键词】 测试服务设计 服务平台 .Net core 2.0 MVC

【Abstract】Nowadays, with the development of software developing technology, the lifecycle and technical threshold required to develop a software is also relatively lower, but at the same time, the software users have higher and higher requirements for the functions and experience of software use. Under this contradiction, software testing has become more and more important. This article researches and develops a software test design service platform through which to integrate test resources and demand resources. The main points are: 1) The user function module, which provides a trading platform with information release and acquisition functions for both users of requirements and service providers. 2) Payment module to conduct transactions through online payment methods, while ensuring the security of fund flow. 3) The back-end interface adapts to the mobile terminal at the same time to make compatibility work for the future migration to the mobile terminal. 4) In the background, it is able to view the information of both parties of the transaction to facilitate the handling of user complaints and other needs.

【Keywords】 Test service design Service Platform .Net core 2.0 MVC

目录

[第一章 引言 4](#_Toc388250090)

[1.1 课题研究的背景 4](#_Toc388250091)

[1.2 课题研究的内容 5](#_Toc388250093)

[1.3 论文结构 6](#_Toc388250094)

[第二章 系统开发相关技术 7](#_Toc388250095)

[2.1 HTTP 1.1协议 7](#_Toc388250096)

[2.1.1 HTTP 1.1协议概述 7](#_Toc388250097)

[2.2 JAVA EE 7技术 7](#_Toc388250098)

[2.2.1 JAVA EE 技术概述 7](#_Toc388250099)

[2.2.2 Java Servlet 3.0 7](#_Toc388250100)

[2.2.3 JDBC 8](#_Toc388250101)

[2.3 Javascript&JQuery 8](#_Toc388250102)

[2.3.1 Javascript&AJAX 8](#_Toc388250103)

[2.3.2 JQuery 8](#_Toc388250104)

[2.4 MVC 框架 8](#_Toc388250105)

[2.5开发环境 9](#_Toc388250106)

[第三章 新PHOTO-SHOW系统需求分析 10](#_Toc388250107)

[3.1总体需求 10](#_Toc388250108)

[3.2 功能性需求分析 10](#_Toc388250109)

[3.2.1 个人资源管理模块 10](#_Toc388250110)

[3.2.2 个人资源展示模块 11](#_Toc388250111)

[3.2.3 移动端接口模块 12](#_Toc388250112)

[3.2.3 公共功能模块 13](#_Toc388250113)

[3.3 非功能性需求分析 14](#_Toc388250114)

[3.3.1 可行性 15](#_Toc388250115)

[3.3.2 安全性 15](#_Toc388250116)

[3.3.3 可扩展性 15](#_Toc388250117)

[3.3.4 可维护性 15](#_Toc388250118)

[3.3.5 性能需求 15](#_Toc388250119)

[3.4 本章小结 15](#_Toc388250120)

[第四章 新PHOTO-SHOW系统设计 17](#_Toc388250121)

[4.1 系统重构和整体架构 17](#_Toc388250122)

[4.2 功能模块设计 17](#_Toc388250123)

[4.2.1 个人资源管理模块设计 17](#_Toc388250124)

[4.2.2 个人资源展示模块设计 21](#_Toc388250125)

[4.2.3 移动端接口模块设计 21](#_Toc388250126)

[4.2.4 公共功能模块设计 24](#_Toc388250127)

[4.3 主要界面设计 25](#_Toc388250128)

[4.3.1 个人资源管理界面 25](#_Toc388250129)

[4.3.2 个人主页界面 27](#_Toc388250130)

[4.4 后台系统设计 28](#_Toc388250131)

[4.4.1 文件空间管理系统 29](#_Toc388250132)

[4.4.2 事务管理系统 30](#_Toc388250133)

[4.4.3 权限系统 31](#_Toc388250134)

[4.5 数据库改进与设计 32](#_Toc388250135)

[4.5.1 改进的表 32](#_Toc388250136)

[4.5.2 新增的表 34](#_Toc388250137)

[4.6 本章小结 35](#_Toc388250138)

[第五章 新PHOTO-SHOW系统实现 36](#_Toc388250139)

[5.1 功能模块的实现 36](#_Toc388250140)

[5.1.1 个人资源管理模块的实现 36](#_Toc388250141)

[5.1.2 个人资源展示模块的实现 43](#_Toc388250142)

[5.1.3 移动端接口模块的实现 44](#_Toc388250143)

[5.2 后台辅助系统实现 45](#_Toc388250144)

[5.2.1 文件空间管理系统的实现 45](#_Toc388250145)

[5.2.2 事务管理系统的实现 46](#_Toc388250146)

[5.2.3 权限系统的实现 47](#_Toc388250147)

[5.3 本章小结 48](#_Toc388250148)

[第六章 总结与展望 49](#_Toc388250149)

[参考文献 50](#_Toc388250150)

[谢辞 51](#_Toc388250151)

1. **引言**

**1.1 课题研究的背景**

在软件开发周期中，软件测试是不可替代的一个部分，几乎所有的大型软件开发公司或团队，都有着自己的测试团队，他们与产品、设计及开发人员密切配合，保证软件产品的质量。传统的软件测试，需要专门的测试人员花费大量时间去进行拟定测试计划、撰写测试用例、执行测试、各个版本的回归验证等工作，随着敏捷开发的理念获得越来越多的企业的认可，开发和测试势必会朝着对文档更低依赖的方向发展，且开发和测试的迭代周期会越来越短，效率的要求越来越高。如今的测试的趋势是需求更快的迭代周期、更高的测试效率、更少的人力资本，对于部分更注重功能的企业，特别是非纯软件开发企业或规模较小的企业来说，他们更希望不必去投入资源组建专业的测试团队，但又期望自己的产品发布之前经过专业的测试。本设计的目的，是希望通过一套测试及评估系统，进行对该方向的研究，为以后进一步的在云平台上实现理想化的测试研究奠定基础。通过这个系统，对软件测试有需求的用户，能够非常方便的获得专业测试团队的测试服务。而具有专业测试能力的团队或个人，通过这个平台，可以在任何时间、地点，向需求用户提供服务，同时获取一定报酬。通过这种方式，用户相当于仅在有测试需求的时候，才去维护一支测试团队，且不用提供额外的场地和设备。对于不需要长期进行测试工作的企业来说，这是一种非常高效且节省的手段。

如今市面上提供类似服务的网站主要有：

1. Testin、阿里的MQC云测试、腾讯质量开放平台、百度MTC云测试等，其中Testin、MQC、MTC三个测试平台都是主要针对移动端的应用进行，而腾讯的测试平台针对的是游戏；除此之外，以上几者的方式，都是通过自有的测试团队，或是提供云主机及相应自动化测试方法来进行测试，而没有开放让自由测试者或团队参与的功能。
2. 猪八戒等服务众包平台，它能够让有需求的用户和服务提供者自由选择，使得双方各取所需，但是需求的范围更多是在平面设计、广告、工程设计等方面，而忽略了测试服务这一个非常有潜力的方面。

上述网站都有着不少量的用户群体，但是都没有体现本设计的理念“测试即服务”，将测试作为一种服务，同时也是一种商品，进行供需交易，它不是满足单方面的需求，而是整合了社会中的测试供求资源，使闲散资源的利用率更加充分。

1.2 课题研究的内容

课题的主要研究内容包括：构建简洁清晰的交互平台，使用户能够快速了解如何使用；具有异步响应的后端接口，实现设计的各种功能，同时能够保证多用户同时访问；数据库相关；网上支付接口，使用第三方的在线支付接口，用户能够方便的进行扫码支付，安全快捷；网站项目部署，考虑租借云服务器及域名报备工作。

具体内容及逻辑如下：

1. **用户交互平台**

此平台是用户交互的主要工具，因此在保证功能的同时，要确保界面足够的简洁明了。平台用户分两类，需求测试服务的用户，和提供测试服务的用户，两者使用同一账号登录，即一个账号有双重身份，类似淘宝账号既可以开店销售，也可以购买商品，所以界面需要能够在两种不同的身份下显示不同的内容。用户可以进行账号基本操作，登录、注册、修改个人信息等；需求测试的用户，可以发布需求、删除需求，在接到服务提供方的订单后，可以选择确认接收订单、拒绝订单等操作，确认接收过程包括有付款等流程，订单结束后，如对交易有异议则可以向平台进行申述；提供测试服务的用户，可以查看他人当前发布的需求，挑选适合自己的，与对方联系交流细化具体责任与收益，在得到统一意见后，选择接受需求单，并填写相应信息构成双方订单，完成后，再待对方确定，至此双方协议达成，在工作期间，提供测试服务的用户，可以按要求在界面上更新任务的进度，此进度可以被对方查看，作为双方的补充交流。

1. **后端开发**

后端接口为前端界面展现的各种功能服务，除了进行相应的逻辑处理，也对数据库进行相应的操作。后端接口接收Http协议的Get和Post请求，对URL进行相应的路由解析后，分配到对于的控制器，再对相应的参数进行处理，将结果返回给前端展现给用户。

1. **数据库**

数据库负责储存相应的信息，包括用户的个人信息、需求及订单的各种数据等。

1. **网上支付**

平台的支付方式设计为在线支付，在线支付是现在的主流支付方式之一，比起传统的网上银行等支付方法，移动端的在线支付无疑更加的方便快捷，同时安全性也有很好的保障。目前开放网上支付接口的主要有微信支付和支付宝，接入对应的接口需要做相关的认证工作。

**（5）网站项目部署**

此项目面向的是多用户在线使用，由于官方部门的限制未经备案的地址不会开放80等主要端口，而无论是作为调试还是使用，部署于本地显然都是不可靠以及不方便的。目前市面上常见的商用云服务器有阿里云、腾讯云、AWS等，除了部署到云服务器，还需要申请对相应域名的备案。

1.3 论文结构

本文共分为6章。

第一章为引言，介绍该课题的研究背景，包括软件测试领域目前的发展及前景，分析了软件开发领域对测试技术的依赖以及与需求相应的资源付出的矛盾，以及对未来可能的测试方式的预测，此预测是本项目立项的基础。除此之外，对比了目前市面上已有的，和本项目可能存在竞争的一些网站或工具，他们包括阿里、腾讯、百度等技术雄厚且涉及领域广阔的互联网巨头，同时也分析了此项目为何仍能有足够的生存空间。

第二章介绍了课题中，各个模块使用到的技术，以及使用的这些技术与其它热门技术的优劣势对于，由此分析选择相应技术的原因。

第三章介绍了本项目的需求分析，以及实现需求的相应功能逻辑和操作逻辑。

第四章展现了系统的详细设计，包括各个模块的展示，数据的流动，数据库表的设计，以及模块之间连接以及通信的方式。

第五章为系统实现，详细描述了该系统主要功能的实现细节以及在实现过程中遇到的问题和相应的对策。

第六章是对本文的总结并对该课题提出展望，包括未来的前景和下一步发展计划。

1. **开发相关技术及比较**

2.1 HTTP 协议

2.1.1 HTTP 1.1协议概述

[超文本传输协议](https://baike.baidu.com/item/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE)（HTTP，HyperText Transfer Protocol)是[互联网](https://baike.baidu.com/item/%E4%BA%92%E8%81%94%E7%BD%91" \t "_blank)上应用最为广泛的一种[网络协议](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%8D%8F%E8%AE%AE)。所有的[WWW](https://baike.baidu.com/item/WWW)文件都必须遵守这个标准。在本项目中，用户可见的网页前端所有的提交，都是通过这一协议传输到后端进行处理，同样，后端处理后的返回，也是通过这样一个传输协议传输到前端展现在用户的眼中。在OSI七层网络模型中，对HTTP协议的定位是应用层协议，他由请求和响应构成，是一个标准的客户端服务器模型。HTTP是一个无状态的协议。

2.2 .NET core 2 Framework

2.2.1 .NET Framework 概述

.NET Framework 是微软研发的一款开发框架，常见使用C#语言做后端开发,html\css\js做前端开发。.NET框架是一个多语言组件开发和执行环境，它提供了一个跨语言的统一[编程环境](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B%E7%8E%AF%E5%A2%83)。.NET框架的目的是便于开发人员更容易地建立Web应用程序和Web服务，使得Internet上的各应用程序之间，可以使用Web服务进行沟通。从[层次结构](https://baike.baidu.com/item/%E5%B1%82%E6%AC%A1%E7%BB%93%E6%9E%84)来看，.NET框架又包括三个主要组成部分：[公共语言运行时](https://baike.baidu.com/item/%E5%85%AC%E5%85%B1%E8%AF%AD%E8%A8%80%E8%BF%90%E8%A1%8C%E6%97%B6)（CLR：Common Language Runtime）、[服务框架](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E6%A1%86%E6%9E%B6" \t "_blank)（Services Framework）和上层的两类应用模板——传统的Windows应用程序模板（Win Forms）和基于ASP.NET的面向Web的网络应用程序模板（Web Forms和Web Services）。本项目使用的，是在此基础上有所更新的Web网络应用程序模版。

2.2.2 .NET Core 2

.NET Core是微软最新推出的跨平台框架，具有跨平台 ([Windows](https://baike.baidu.com/item/Windows)、Mac OSX、[Linux](https://baike.baidu.com/item/Linux" \t "_blank)) 能力的应用程序开发框架 (Application Framework)，未来也将会支持 [FreeBSD](https://baike.baidu.com/item/FreeBSD) 与 Alpine 平台，也是微软在一开始发展时就开源的软件平台。这解决了.NET框架一直以来为人所诟病的问题，一是不开源，这对于最求低成本的开发来说是很致命的一点，同时没有强大的社区作为支持，使得所有工作都必须依靠微软的原生工具；二是只支持Window平台，这使得无论是开发还是部署工作，都不具备高兼容性和扩展性，相对封闭的整套环境，使得目前国内使用这套框架做开发的少之又少。而这些问题的解决也让.NET框架未来和JAVA、PHP等的目前绝对主流有了竞争之力。

 .NET Core 包含 .NET Framework 的类库，但与 .NET Framework 不同的是 .NET Core 采用包化 (Packages) 的管理方式，应用程序只需要获取需要的组件即可,这是.NET相应于Gradle、Maven等框架的对等方式，使开发获取所需的各种依赖变得更加方便。

清晰的项目结构，依靠如今最为优秀的开发平台Visual Studio，同时可以使用Entity Framework等技术，方便的LINQ语句，同时有着微软作为稳定性保障，也是本项目选择这个框架的原因。

2.2.3 Entity Framework

[ADO.NET](https://baike.baidu.com/item/ADO.NET) Entity Framework 是微软以 ADO.NET 为基础所发展出来的对象关系对应 ([O/R Mapping](https://baike.baidu.com/item/O%2FR%20Mapping)) 解决方案。以 Entity Data Model (EDM) 为主，将数据逻辑层切分为三块，分别为 Conceptual Schema, Mapping Schema 与 Storage Schema 三层，其上还有 Entity Client，Object Context 以及 LINQ 可以使用。微软在 Entity Client 的上层加上了一个供编程语言直接访问的界面，它可以把 EDM 当成对象般的访问，此界面即为 Object Context (Object Service)。在本项目中，Objecy Context有非常广泛的使用。

2.2.4 LINQ

LINQ（Language Integrated Query）语言集成查询是一组用于[c#](https://baike.baidu.com/item/c%23)和[Visual Basic](https://baike.baidu.com/item/Visual%20Basic" \t "_blank)语言的扩展。它允许编写C#或者Visual Basic代码以操作内存数据的方式，查询数据库。使用LINQ语句，可以非常方便的进行对数据库的查询等操作，相比于传统的新建SQL string然后执行，LINQ语句使用起来更加自然且和C#可以无缝的连接使用，对数据库的操作就像普通代码编写一样，直接嵌入到代码主体中，理解起来更为方便直观，效率也得到保障。

2.3 CSHTML & Razor/JavaScript & AJAX

2.3.1 Cshtml & Razor

cshtml 是 MVC3 之后新增的视图文件，语法上与HTML基本相同，解释引擎是 Razor Rendering Engine。Razor是.NET特有的标记语法，用于在网页中嵌入服务器端代码。Razor 语法基于 ASP.NET 框架，该框架是微软的 .NET 框架特别为 web 应用程序开发而设计的组成部分。当网页被写入浏览器时，基于服务器的代码能够创建动态内容。在网页加载时，服务器在向浏览器返回页面之前，会执行页面内的基于服务器代码。由于是在服务器上运行，这种代码能执行复杂的任务，比如访问数据库。在前端页面中嵌入的Razor，使得编程更加简洁，这也是相比之前版本的aspx 视图引擎更加轻量高效的视图引擎。

2.3.2 JavaScript & AJAX

JavaScript是一种运行在浏览器中的解释型的编程语言，也是号称世界上最流行的编程语言，它可插入 HTML 页面，可由所有的现代浏览器执行。JS是基于原型的多范式动态语言，支持面向对象，命令式和声明式（例如函数式编程）风格。JS的作用非常多，最常用的包括：对浏览器中的事件作出响应，比如点击、悬浮于指定区域等；操作HTML中的元素，动态的更改其表现或内容；在前端将请求发送前或收到来自后端的信息前，对数据进行预处理，比如验证格式、长度等；控制Cookies。本项目中，部分效果通过JS实现，同时处理来自后端的返回信息。

AJAX 即Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。异步传输是面向字符的传输，它的单位是字符；而同步传输是面向比特的传输，它的单位是帧，它传输的时候要求接受方和发送方的时钟是保持一致的。本项目中，HTTP请求通过JS和Ajax来构建及发送。

2.4 SQL Server

2.4.1 Visual Studio 集成的MSSQL SERVER LocalDB

SQL Server 是Microsoft 公司推出的关系型[数据库管理系统](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F)，而SQL Server LocalDb是SQL Server Express的轻量级版本，在Visual Studio中有集成。在本项目中，使用静态mdf文件来作为数据库储存信息。选择MSSQL Server数据库，原因之一它相比其他数据库来讲非常轻量化且易于移植和部署，在项目初期需要储存的数据量不大、对并发查询响应速度或效率要求不是特别高的时候，使用它是非常方便的；原因之二是它微软自行研发的数据库，和Visual Studio的关联性较好，可以在VS中直接对数据进行修改，测试起来非常方便。

2.5 .NET MVC Framework

2.5.1 .NET MVC Framework简介

[MVC](https://baike.baidu.com/item/MVC)全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。.NET MVC是微软官方提供的以MVC模式为基础的.NET Web应用程序(Web Application)框架。

在.NET MVC框架中：

Model即是一个类（class），它定义了相同一系列的数据都拥有着什么样的属性，在这一点上它和传统意义上的类没有区别，除此之外，由于Entity Framework的引入，Model不仅作为类，还代表着与数据库中的表的映射，一个Model对应着数据库中的一张表，Model中的每一个属性，与之对应的就是数据库中对应表的一列，使用Code First的数据迁移方式，我们可以通过构建Model来生成对应的数据库表；

View及视图，它绘制了一个用户所看到的一切界面，在.NET中，一个View就是一个cshtml文件，一个cshtml文件中，包含由Razor、html、JS共同编写的代码，以及对css样式表和其他JS文件的引用。html/CSS用于绘制页面，JS用于前端逻辑的控制以及HTTP请求的构建发送及返回信息处理，Razor除了渲染页面，也能实现一定的逻辑操作及数据引用。

Controller即传统意义上的后端，负责处理功能逻辑以及对数据库的操作。每一个Controller中，有着若干个Action,每一个Action即一个函数，对应于不同的URL，这根据Route规则来映射，而这些Action接收不同的参数，处理后返回给请求端，这些返回可以是一个Json，可以是一个带有Model数据的View，也可以是指向其他某个Action或View的重定向。因此可以说，每一个Action即一个功能API。

除此之外，在Model中还往往会有着一种View Model,它也是一个类（class），不同于映射向数据库表的Model，View Model更符合一个“类”的定义,它被实例化用于数据传递，将各种属性绑定于这样一个类，使用一个类作为参数，用作形参接收前端参数，或用作实参向前端返回，这更符合面向对象程序设计的思想。为了弥补这样传递数据的不灵活，.NET 还有 ViewData、ViewBag两种方式作为补充。

1. 项目需求分析

3.1 引言

本需求分析用于指导软件测试设计服务平台的开发，通过对设计功能的具体分析以及对非功能性需求等的分析，给出明确的功能逻辑及实现要求，以求设计出与预期更为贴近的服务平台，同时以本需求分析作为参考，降低后期升级进一步开发或维护的成本。

3.2 总体需求分析

项目采用B/S架构，为保证兼容性及可移植性，项目构建所用工具、框架、技术应选用较为通用，且版本较新而稳定的。

项目应围绕主题核心，“测试即服务”为主要设计点，界面及功能上，以两种身份的用户完成交易的整个过程涉及到的方面为主，以用户信息等方面为辅，进行开发。界面要求简洁明了显示准确，功能要求完整且及时响应。

项目结构应该按照前后端分离，模块独立的思想来构建，清晰且可复用，代码风格良好，方便后期的维护或者升级进一步开发。

3.3 功能性需求分析

3.3.1 公共部分功能需求

公共部分功能及是指两种身份——测试服务需求用户和测试服务提供用户——都会使用到的功能。

1. 登录、注册功能：用户可以在登录页面通过填写账号密码，确认密码后注册新账号，新账号具有默认的个人信息；已经注册过的用户，可以直接使用自己的账号密码进行登录。没有登录认证的用户，直接通过URL或自行构造POST请求试图访问权限不足的页面时，会自动重定向到登录页面
2. 个人信息功能：用户可在导航栏中的个人信息栏中，访问到自己的个人信息，包括联系方式，邮箱地址，注册时间等。其中联系方式和邮箱地址可以随时更改，而注册时间在账号注册时就确定且不可更改。
3. 退出登录：用户退出登陆后，无法在继续访问需要登录认证的界面，且当前页面重定向到登录界面，其他已经打开的页面上的任何操作也会直接重定向到登录页面，保证退出后用户的信息安全。

3.3.2 测试服务提供用户（以下简称乙方）功能需求

（1） 查看可接需求：查看当前可以接收的测试需求，这些需求是由测试服务需求用户（以下简称甲方）创建并发布于公共页面的，乙方在首页列表能够查看到测试需求的基本信息，并可以点击选中的某一个测试需求进入到详细页面。在详细页面中，可以查看到更为详细的信息，并可以选择返回到列表，或者选择接单，选择接单后，进入创建订单功能。

（2）创建订单：创建订单前，需要乙方通过甲方留下的联系方式与甲方沟通，确认具体的需求、开始截止时间、订单金额等，考虑到沟通内容的实时性、严谨性和细致性，本服务平台暂不集成在线交流工具，但可以预留后期开发的空间。在确定以上信息后，乙方在创建订单页面填写后提交。此时订单应处于waiting状态，等待甲方确认乙方填写的、经沟通后统一意见的方案信息。

（3）订单进程管理：乙方可以修改处于open状态的订单的进程情况，进程情况代表着乙方对自己工作占总需求百分比的一个评估，需要如实反馈给甲方。这个进程情况随着修改，能够在甲方界面刷新后及时更新。

3.3.3 测试服务需求用户（甲方）功能需求

（1）发布测试需求：甲方在有测试需求的时候，可以到相应界面创建新的需求，填写对于信息后发布到需求列表当中，此列表对所有用户可见，因此甲方（同时也可能具有乙方测试服务提供能力）不能够接收自己发出的需求，会有相应提示。

（2）管理发布的需求：甲方可以在个人列表中查看到自己发布的所有需求，这些需求不会随着被乙方接单而自动删除（因为可能一个需求需要由多个乙方来完成），甲方可以在列表中删除掉过期的或者有改动的需求。

（3）订单管理：当有乙方接单并创建好订单后，甲方能在订单管理界面看到这个处于waiting状态的订单，确认信息与约定无误后，选择接受并进行支付，支付采用第三方支付平台接口。完成后，订单状态变为open，意示双方达成协议，此订单会随着阶段改变状态，但不能够被删除，这是用户申述的关键依据。当然，甲方若不满意乙方填写的订单内容决定要临时更改，或订单内容与当初协商有出入，可以选择拒绝此订单，订单状态由此变为closed。当订单进度被乙方改为100%，乙方确认服务达成后，选择确认完成，订单状态变为finish；或者有异议，可以选择申诉。由此整个流程生命周期结束。

3.3.3 后台功能性需求

后台系统与服务平台页面相分离，采用独立的特殊账号登录，可以查看到所有的用户、订单、需求等等信息。同时，当有用户申诉时，后台能够看到具体的争议订单，及有争议的双方，在与双方联系并参考签订订单的具体协商内容后，后台可以将订单涉及的预付金额，在总量不超过的情况下，分比例向双方支付。

3.4 非功能性需求分析

3.4.1 可行性分析

本项目采用的框架或技术都是开源且成熟的，实现上的难点仅在于时间，所以需要注重对项目周期长度的把控。服务器采用阿里云，认证学生身份后也能够以非常优惠的价格租借低配服务器两个月，对于目前阶段的负载来说足够使用。平台在能够提供高质量服务且拥有一定量稳定客户前，采取免费及无广告模式，维护及开销由个人支付，因此需要注意控制成本。

3.4.2 性能分析

本项目不涉及图像或视频等大尺寸文件的传输，所以性能压力主要来源于并发的访问量，考虑到初期用户群里规模较少，并发访问的用户量并不大，且由于平台无特殊时间段的区别，所以小概率会遇到用户集中访问的情况。由于租借的服务器性能亦有限，因此暂计划满足一百内的并发访问。

3.4.3 可扩展性分析

本项目定位是希望通过一套测试及评估系统，进行对该方向的研究，为以后进一步的在云平台上实现理想化的测试研究奠定基础，因此后期可做的扩展内容极多，需要保证项目的可扩展性。首先在数据库表设计方面，满足基础范式要求，同时使得表属性尽量完备，甚至可以预留空置属性，表属性的新建或更新比起表的新建更为复杂，因此在设计之初就尽量保证不做或少做更改，当存储的数据达到一定量而又需要对表进行重建时，数据迁移的代价及复杂度将会直线上升。在控制器及接口设计上，尽量保证功能相互独立而不冗余，对公用的方法进行适度封装以方便后期开发复用，提高项目的可扩展性。

3.5 本章小结

第三章项目需求分析，主要对项目的功能性需求和非功能性需求进行了分析，此章作为指导开发的重要依据。