

项目研发报告

1 引言

1.1 编写目的

项目研发报告提供了对整个项目研发过程的总结性描述，具体来说包括以下几部分：

- 详细介绍“南大测试在线业务系统”的项目背景，即该项目是在什么样的环境下开展的，我们为什么要做这个项目，项目实施的社会意义是什么，项目预期达到的目标是什么，项目的发展前景如何，以及我们有哪些优势可以来完成这个项目；
- 说明项目的需求分析过程，即我们从何处获取需求，如何分析和整理需求，并逐步细化出所有的软件功能，如何全面地理解用户的各项要求，就软件功能与客户达成一致，估计软件风险和评估项目代价，并根据需求分析的结果形成一个最终的开发计划；
- 描述整个项目组的任务安排与人员分工情况，包括项目组是由哪些成员组成的，每个成员负责的是项目的哪一部分，各自的完成情况如何，以及汇总每个成员的工作量统计结果；
- 详细说明项目的管理方法，即组内是如何讨论并协调各个成员的开发进度，如何根据需求的最新变化确定下一阶段的开发任务，以及前端组和后端组之间是如何进行对接和同步的；
- 详细介绍项目开发过程中使用到的关键技术，包括项目使用的开发框架和外部基础设施，为什么这样选择，以及代码编写过程中遇到过哪些技术难题，这些问题最终是如何解决的；

1.2 项目背景

南京大学软件测试中心是具有国家资质认定的软件测试机构，可以向社会出具具有证明作用的数据和结果。测试中心的日常业务主要是接取企业客户的软件测试委托，进行相关测试并给出测试结果。长期以来，该机构一直使用基于填写word文档的方式管理整个业务流程，现在他们希望为整个业务流程搭建一个在线管理平台。

开展本项目的目的之一，就是将南大测试中心的业务变为在线的web应用形式，客户可以通过前端页面进行数据填写，并由后端服务将数据保存在专门的数据库中；同时，本项目也是以团队合作的形式进行的，因此开展本项目的另外一个目的在于，让大家能够体验团队合作的过程以及实际的软件开发工作，从需求分析到文档编写。

本项目实施的意义就在于让同学们可以感受一下真实的软件开发究竟是怎样一种情形，软件开发不只是编写代码，或者说写代码只是软件开发过程中的一小部分。除此之外，我们还需要了解到需求分析对于整个代码编写的指导作用，人的协调对于项目进度的深刻影响，外部评价标准对项目开发本身所造成的压力。项目所预期达到的目标，一方面是帮助同学们学习掌握现代web开发技术，将“南大测试在线业务系统”成功的实现出来；另一方面，也是希望能够帮助大家为以后的工作做好准备。

我们相信将传统的word文档改为现代化的web页面形式，能够促进南大测试中心的业务发展，为南大测试中心带来更多的客户资源和合作机遇。我们的优势就在于，作为南大的学生我们都有着扎实的专业基础、优秀的学习能力，可以在规定时间内完成既定的任务目标。

2 研发过程

2.1 需求分析

2.2 项目任务

项目任务总体上可以分为两部分，一部分由前端组负责，即通过HTML、CSS及JavaScript以及衍生出来的各种技术、框架、解决方案，来实现互联网产品的用户界面交互；另一部分由后端组负责，即通过Java语言及其相关生态中被广泛使用的代码框架和第三方依赖，来实现各种数据的增删改查、业务的流程控制和权限管理。更具体的说，后端组的任务又可以分为：

1. 需求分析。
2. 代码编写
 1. 用户管理。用户可以进行注册、登录、登出，修改账号信息，进行身份认证以及权限鉴别；
 2. 委托管理。用户可以填写并提交委托，随时查看委托状态，测试中心人员可以对委托进行审核；
 3. 合同管理。委托双方可以进行合同创建、查询、评审、修改等操作，并封存合同扫描件；
 4. 样品管理。测试工作人员可在平台维护接收到的样品实体的信息、状态。
 5. 文档生成。用户可以根据已有的表单信息请求生成PDF格式的文档并下载留存。
 6. 测试项目。用户可以进行测试期间的文档的管理、测试期间的报告的管理。
3. 文档编写
 1. 需求文档。
 2. 项目研发报告。
 3. 项目测试报告。
 4. 设计说明书。

2.3 人员分工

项目组成员一共有12个人，其中前后端各有6个人：

- 前端组：
- 后端组：杨茂琛（组长），张世茂，戴显灏，姜宁，王岳，孙逸扬

具体而言，各个成员负责和完成的主要任务如下所示：

- 后端组
 - 杨茂琛：
 - 张世茂：
 - 戴显灏：
 - 姜宁：
 - 王岳：
 - 孙逸扬：完成了用户管理、委托管理、合同管理以及应用网关的功能代码和测试代码编写，与其他成员协作完成后端服务和数据库的更新部署，以及API文档、研发文档的部分编写工作。
- 前端组
 -

其中有多项任务是多成员合作完成的，基本上每个成员都负责了多个不同的任务。

2.5 数据统计

2.6 项目管理

3 关键技术

3.1 微服务架构

后端代码采用微服务架构进行开发。微服务是一种软件架构风格，它是以专注于单一责任与功能的小型功能区块为基础，利用模块化的方式组合出复杂的大型应用程序，各功能区块使用与语言无关的API集相互通信。微服务运用了以业务功能为主的设计概念，应用程序在设计时就以业务功能或流程设计先行分割，将各个业务功能独立实现成一个能自主执行的个体服务，然后再利用相同的协议将所有应用程序需要的服务都组合起来，形成一个整体。

在本项目中，每个管理模块都是一个单独的微服务应用，各个服务之间通过HTTP协议进行通信，通过外部数据库来实现数据同步和状态共享。采用微服务架构的优点就在于，当需要针对特定业务功能进行扩展时，只需要对该业务功能的服务进行扩展就好，不需要整个应用程序都扩展；当某个服务模块出现问题不能正常工作时，其它不相关的服务仍然可以正常工作。同时，微服务架构因为使用更小、更独立的划分单元，从而可以方便的进行水平向扩展与集群配置，每个服务可以独立部署并且快速启动，可以由专门的团队负责专门的服务。

3.2 Spring Framework

后端代码主要基于Spring框架进行开发，Spring Framework是Java语言最受欢迎、使用最广泛的代码框架之一。它是一个开源的全栈应用程序框架和控制反转容器实现，通过依赖注入把创建对象的权利交给框架，由框架来管理对象的存储与生命周期；通过面向切面编程将那些与具体业务无关，但被不同模块同时使用的逻辑抽离并封装起来，降低了系统的冗余度与耦合度。Spring为诸多第三方依赖和基础设施提供了整合，也为微服务架构提供了开箱即用的实现。具体来说，我们的项目使用到了以下的Spring模块：

- Spring Boot：使用基于注解和约定的自动配置，使程序员可以摆脱繁琐冗长的XML配置文件，大大简化了Spring应用的搭建和开发过程；通过集成大量的第三方工具使得依赖包的版本冲突得到很好的解决；
- Spring Cloud：极大简化了基于微服务的分布式系统开发，包括整合Eureka以提供服务注册和发现，提供基于响应式编程的Gateway，以及用于负载均衡的LoadBlancer来实现服务间调用等等；
- Spring Data：为数据访问提供熟悉且一致的基于Spring的编程模型，同时仍然保留底层数据存储的特殊特性；提供了对常见数据库如MySQL、Redis、MongoDB的整合，使得访问数据库变得简单易用；
- Spring Session：专注于解决分布式或集群配置场景中的会话管理问题，可以简单快速的集成第三方缓存数据库，从而实现会话数据共享；
- Spring Test：是用于测试Spring应用的程序框架，它为Spring的各个模块提供了测试方法，可以方便的整合JUnit5以实现单元测试，整合Jacoco以实现代码覆盖率统计，同时也可用于集成测试。

除此之外还有其它的一些Spring提供的功能，这里不再一一列举。基于Spring框架进行开发，深刻体现了软件工程中关于代码复用的哲学，即如何利用已有的第三方支持，更快更好的完成项目开发。

3.3 数据库相关

由于后端应用主要就是为前端提供数据服务，因此数据库的选择和使用非常重要。将数据放在专门的数据库中，一方面有利于数据完整性和安全性；另一方面，由于我们使用的是微服务架构，要想实现数据共享必须将数据放在外部的数据库中。选择正确的数据库不仅有利于代码编写，还能提升应用的整体性能。

3.3.1 MySQL

MySQL是最流行的关系型数据库管理系统之一，由于其性能高、成本低、可靠性好，被广泛应用在各种web应用中。MySQL使用C和C++语言编写，并使用了多种编译器进行测试，具有良好的可移植性。MySQL有可以免费使用的社区版本，并且它为多种编程语言提供了API，其中就包括Java语言的MySQL Connector。

MySQL适合存储大量的关系型数据，能够支持并发访问和事务管理。本项目中使用MySQL保存用户的基本信息，包括用户名、密码和用户身份等等。因为现在容器化技术比较流行，MySQL也提供了Docker容器版本，所以本项目中使用的MySQL其实是Docker容器，跟真正的MySQL Server比它的好处就是不需要改配置文件就能用。

3.3.2 Redis

Redis是最流行的非关系型数据库管理系统之一，它使用C语言编写，是一个高性能的键值对存储数据库。本项目中使用redis主要是为了管理会话，Spring Session提供了与Redis相关的集成，用户登录后的会话数据都保存在外部的Redis数据库中，这样就实现了会话状态在集群间共享。网关服务会向用户服务请求用户的身份信息，以实现后续的权限鉴别；而后者会从Redis数据库中取出保存在会话里的用户数

据，这样就可以避免每次都访问MySQL带来的磁盘I/O开销。另外，如果要用户服务部署成集群模式，多个用户服务实例之间通过Redis共享，将不会丢失会话数据。本项目使用的其实也是Redis的Docker镜像，不过修改了配置文件以加上密码。

3.3.3 MongoDB

MongoDB也是最流行的非关系型数据库管理系统之一，与Redis不同的是，前者主要用于键值对存取；而它主要用于文档对象存取。MongoDB和Redis都提供了Java语言的驱动版本，Spring框架也为这两个数据库提供了整合方案。MongoDB主要将数据存储在内存在中，不过它会利用底层操作系统的内存映射机制，每隔一段时间把数据刷到磁盘上以实现持久化。

本项目使用MongoDB来存储各种表格数据，包括委托表、合同表、测试方案表等等。这些表格都有很多字段，并且字段都是不定长的字符串，使用MySQL这样的关系型数据库存储会很不方便。而MongoDB由于采用基于JSON对象的存储形式，可以很方便的存取这些表格。另外，由于MongoDB本身自带缓存，因此也不需要为它单独准备一个缓存方案，非常的好用。与前面两个类似，本项目使用的MongoDB也是Docker镜像。

3.3.4 MyBatis

MyBatis是一个半ORM框架，它支持自定义SQL，并且免除了几乎所有的JDBC代码以及设置参数和获取结果集的工作，是一个广泛使用的持久层解决方案。MyBatis可以通过简单的XML或注解来配置和映射原始类型、接口和Java POJO（普通老式Java对象）为数据库中的记录。与Spring Data JPA默认使用的Hibernate框架不同的是，前者是一个全自动的ORM框架，从生成SQL到获取查询结果都是自动完成的；而MyBatis保留了由程序员编写SQL的步骤，这使得优化SQL查询更加方便，这也是它被称为半ORM的原因。本项目主要使用MyBatis来访问MySQL数据库，而使用它的好处就在于我们只需要编写SQL语句就行了。

3.3.5 Object storage

对象存储（也称为基于对象的存储）是一种计算机数据存储，它将数据作为对象进行管理，而不是像文件系统那样将数据作为文件层次结构进行管理，以及块存储那样将数据作为扇区和轨道内的块进行管理的其他存储架构。每个对象通常包括数据本身、数量不等的元数据和一个全局唯一的标识符。对象存储试图实现其他存储架构所不具备的能力，如可由应用程序直接编程的接口，可跨越多个物理硬件实例的命名空间，以及数据管理功能，如数据复制和对对象级颗粒度的数据分发。

绝大多数对象存储的实现都是云存储。基于云存储，本业务平台的二进制文件的分发可以独立于后端服务器的运行；提升在线率和可用性的同时，优化了二进制文件分发的用户体验。

3.4 构建工具

构建工具是用来管理项目的各个模块，以及它们所使用的第三方依赖的。它为整个项目提供了从源码编译到运行测试，再到打包部署的完整流水线支持。任何一个适合大规模项目开发的语言，都一定会有一个与之对应的构建工具，它管理着项目中用到的第三方依赖的版本，以便在不同成员的不同开发环境下都能顺利编译该项目。

3.4.1 Maven

Maven是Java语言使用最广泛，也是最可靠的构建工具之一，它由Apache基金会编写和维护。基于项目对象模型（POM）概念，Maven可以从一条中心信息管理项目的构建、报告和文档。Maven有一个中央仓库，许多第三方应用都会在这个仓库中发布自己的各个版本，包括Spring框架、MySQL驱动以及MyBatis等等。

Maven使用基于XML的配置文件，并且支持各种各样的第三方插件。比如项目中使用Spring官方提供的Maven插件spring-boot-maven-plugin来打包部署Spring应用，这个插件会把所有用到的第三方依赖都打包进去，这样打出来的JAR包是可以直接运行的；同时还使用了maven-failsafe-plugin，这个插件提供了在Maven项目中运行集成测试的方法。

3.5 PDF书写工具

便携式文档格式（英语：Portable Document Format）是一种用独立于应用程序、硬件、操作系统的方式呈现文档的文件格式。每个PDF文件包含固定布局的平面文档的完整描述，包括文本、字形、图形及其他需要显示的信息。本业务平台需要根据运行时的信息生成稳定排版、格式良好的PDF文档以用于测试中心内部归档、客户审阅。

3.5.1 iText

iText是一个用于在Java和.NET中创建和操作PDF文件的开放源代码库。iText提供对高级PDF功能的支持，如基于PKI的签名、40位和128位加密、颜色校正、标签PDF、PDF表格（AcroForms）、PDF/X、通过ICC配置文件和条形码的颜色管理，并被一些产品和服务使用，包括Eclipse BIRT、Jasper Reports、JBoss Seam、Windward Reports和pdftk。

iText有着不俗的运行效率，后端可以稳定地保证在可接受的较短时间内生成PDF完毕并准备分发。本业务平台通过iText提供的API接口从空白页面开始，从头按着指定格式和内容生成PDF文档。通过对字体、排版等样式的指定，本业务平台可以生成格式文档、可用于正式商业环境的PDF文档。

3.6 持续集成

持续集成是一种 DevOps 软件开发实践。采用持续集成时，开发人员会定期将代码变更合并到一个中央存储库中，之后系统会自动运行构建和测试操作。持续集成通常是指软件发布流程的构建或集成阶段，需要用到自动化组件（例如 CI 或构建服务）和文化组件（例如学习频繁地集成）。

3.6.1 Github Actions

GitHub Actions 让团队轻松实现软件工作流程的自动化。从GitHub上直接构建、测试和部署团队的代码。让代码审查、分支管理和问题处理以简单、明晰、高效的方式进行。通过GitHub事件（如push、pull request或release）来启动工作流程。使用由社区建立和维护的服务进行组合并配置行动。

Github Actions 无需自有长期开放、性能足够的服务器。Github Actions 在 Microsoft 提供的虚拟机上全新地装载指定的运行环境，开箱即用、用后即销毁无需担心残留和隐私问题。得益于全新的虚拟机环境，Github Actions 可以并行多个自动化任务。本团队利用 Github Actions 进行：1. 自动化构建；2. 自动化测试；3. 自动化覆盖度测试（提供可下载的覆盖度报告）；4. 代码扫描警报。