项目研发报告

目录

[1. 引言 2](#_Toc107108919)

[1.1 编写目的 2](#_Toc107108920)

[1.2 项目背景 3](#_Toc107108921)

[1.3 名词解释 3](#_Toc107108922)

[2. 研发过程 3](#_Toc107108923)

[2.1 项目任务 3](#_Toc107108924)

[2.2 项目组分工 4](#_Toc107108925)

[2.3 项目进度 4](#_Toc107108926)

[2.4 项目数据统计 5](#_Toc107108927)

[2.4.1 项目代码行数统计 5](#_Toc107108928)

[2.4.2 完成的功能 5](#_Toc107108929)

[2.4.3 文档规模 6](#_Toc107108930)

[2.4.4 GitHub 6](#_Toc107108931)

[2.4.5 测试规模 6](#_Toc107108932)

[2.5 团队管理 6](#_Toc107108933)

[2.5.1 Git/GitHub 6](#_Toc107108934)

[2.5.2 团队交流 7](#_Toc107108935)

[2.5.3 API管理与前后端对接 7](#_Toc107108936)

[2.5.4 制定代码规范 7](#_Toc107108937)

[3. 关键技术 7](#_Toc107108938)

[3.1 Typescript 7](#_Toc107108939)

[3.2 React 8](#_Toc107108940)

[3.3 Apifox 9](#_Toc107108941)

[3.4 UMI 10](#_Toc107108942)

[3.5 Ant Design Pro 11](#_Toc107108943)

[3.6 Robot 11](#_Toc107108944)

[3.7 Selenium 12](#_Toc107108945)

[3.8 TypeDoc 13](#_Toc107108946)

[3.9 Gulp 13](#_Toc107108947)

[3.10 Python Unit Test 13](#_Toc107108948)

[3.11 Beautiful Report 13](#_Toc107108949)

# 引言

## 1.1 编写目的

本项目研发报告的主要目的是：

介绍“南大测试在线业务系统”的项目背景。

阐述本团队的项目研发过程，包括项目的任务与功能概览，项目组的成员分工安排，项目的重要进度节点与项目的数据统计以及项目的管理与协调。

介绍本项目中所使用到的一些关键技术，即技术亮点。

## 1.2 项目背景

南大测试中心是一个具备国家资质认定的测试机构，客户通过“南大测试在线业务平台”向南大测试中心发起软件项目测试委托申请并成功签署合同后，在南大测试中心进行软件项目的测试，“南大测试在线业务平台”帮助客户和测试中心的管理员对整个业务流程进行管理，本软件为WEB应用程序。

## 1.3 名词解释

客户：向南大测试中心发起软件项目测试委托的人员

管理员：南大测试中心的工作人员

# 2. 研发过程

## 2.1 项目任务

该项目的任务是实现南大测试项目在线管理平台，覆盖"南大测试"在线业务功能，主要功能包括：

1. 用户管理：注册与维护用户；用户身份认证；用户授权；
2. 客户管理：客户注册；客户关系管理；客户查询；
3. 委托管理：在线委托；在线受理与审批；项目建立与查询；
4. 合同管理：合同创建、修改、删除、评审、查询等管理；
5. 样品管理：样品接受、流转、归还、归档、销毁、查询与维护；
6. 报告管理：报告编制、复核、批准、查询等管理；
7. 测试项目管理：测试文档管理；测试报告管理；测试报告生成；

## 2.2 项目组分工

本项目组为南大测试项目的**E组**，负责前端开发，团队共6人，分别为何健平（组长）、苏晨阳、刘京龙、刘东旭、张嘉麒、曾庆扬。

团队成员在开发过程中分工有序，不同人员专注于自己的工作任务，具体的分工如下。

|  |  |
| --- | --- |
| 成员 | 负责的工作 |
| 何健平 | 担任组长职务，每周组织开周会统筹进度，编写自动化测试脚本，编写代码说明文档等。 |
| 苏晨阳 | 担任技术总监职务，负责技术选型，与后端对接技术约定，研发用户管理、客户管理模块等。 |
| 张嘉麒 | 担任产品经理职务，负责调研项目需求，编写产品需求文档、项目流程图、项目功能树、设计UI原型图、编写用户手册文档等。 |
| 刘京龙 | 团队成员，负责统筹推动项目代码开发流程，研发委托管理、合同管理、样品管理、报告管理等模块。 |
| 刘东旭 | 团队成员，负责代码开发，研发委托管理、合同管理、样品管理等模块，编写项目研发报告。 |
| 曾庆扬 | 团队成员，负责代码开发，研发委托管理、合同管理、报告管理等模块，编写项目研发报告 |

## 2.3 项目进度

1. 2022.02.14成立南大在线测试E组
2. 2022.02.24团队参观南京大学软件测试中心，了解业务流程，确认开发前端功能
3. 2022.03.20创建飞书群组用于团队集中讨论
4. 2022.03.27团队第一次飞书线上组会
5. 2022.04.12确定与后端B组进行项目对接
6. 2022.05.09项目需求分析与流程图审核通过
7. 2022.05.20完成项目最小原型
8. 2022.06.23开发完成，编写文档
9. 2022.06.25项目结项

## 2.4 项目数据统计

### 2.4.1 项目代码行数统计

文件数：269

总行数：24680

代码行：21426

注释行：2111

空白行：1143

### 2.4.2 完成的功能

1. 用户管理
2. 客户管理
3. 委托管理
4. 合同管理
5. 样品管理
6. 报告管理
7. 测试项目管理

### 2.4.3 文档规模

需求说明书：41页

设计说明书：18页

用户手册：74页

项目研发报告：13页

工作日志：23页

测试报告：1页

### GitHub

branch：6

commit：307

### 测试规模

测试用例：24

## 团队管理

### 2.5.1 Git/GitHub

使用Git进行分布式协作开发，并使用GitHub作为代码托管中心。

项目GitHub网址：<https://github.com/KyokaiNoKanata/SELabE>。

测试脚本：<https://github.com/KyokaiNoKanata/SELabETest>。

### 2.5.2 团队交流

团队建立飞书群组，每周日晚上七点举行线上会议，会议时长半小时至一小时，成员汇报进度，集中讨论，并统筹安排下周任务. 会中和会后及时记录会议文档.

### 2.5.3 API管理与前后端对接

前端组与后端组共同使用API fox平台进行API文档、API调试、API Mock、API自动化测试一体化协作。

### 2.5.4 制定代码规范

前端在ADP框架下完成功能研发、API接口使用，使用统一约定的常量表，编写规范注释等。

# 3. 关键技术

## 3.1 Typescript

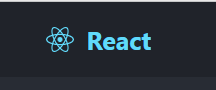
TypeScript是一种由[Microsoft](https://en.wikipedia.org/wiki/Microsoft)开发和维护的[编程语言](https://en.wikipedia.org/wiki/Programming_language)。它是[JavaScript的严格语法](https://en.wikipedia.org/wiki/JavaScript)[超集](https://en.wikipedia.org/wiki/Superset)，并为语言添加了可选的[静态类型](https://en.wikipedia.org/wiki/Static_typing)。它专为开发大型应用程序和[转译](https://en.wikipedia.org/wiki/Source-to-source_compiler)为 JavaScript 而设计。由于它是 JavaScript 的超集，现有的 JavaScript 程序也是有效的 TypeScript 程序。<https://en.wikipedia.org/wiki/Microsoft><https://en.wikipedia.org/wiki/Superset><https://en.wikipedia.org/wiki/JavaScript><https://en.wikipedia.org/wiki/Static_typing><https://en.wikipedia.org/wiki/Source-to-source_compiler>[https://en.wikipedia.org/wiki/TypeScript - cite\_note-4](https://en.wikipedia.org/wiki/TypeScript#cite_note-4)

TypeScript 可用于开发用于[客户端](https://en.wikipedia.org/wiki/Client-side)和[服务器端](https://en.wikipedia.org/wiki/Server-side)执行的 JavaScript 应用程序（与[Node.js](https://en.wikipedia.org/wiki/Node.js)或[Deno](https://en.wikipedia.org/wiki/Deno_(software))一样）。有多种选择可用于转译。可以使用默认的 TypeScript 编译器，或者可以调用 [Babel编译器将 TypeScript 转换为 JavaScript。](https://en.wikipedia.org/wiki/Babel_(transcompiler)" \o "Babel（转编译器）)

TypeScript 支持可以包含现有 JavaScript 库的类型信息的定义文件，就像[C++](https://en.wikipedia.org/wiki/C++) [头文件](https://en.wikipedia.org/wiki/Header_files)可以描述现有[对象文件](https://en.wikipedia.org/wiki/Object_file)的结构一样。这使得其他程序可以使用文件中定义的值，就好像它们是静态类型的 TypeScript 实体一样。[jQuery](https://en.wikipedia.org/wiki/JQuery)、[MongoDB](https://en.wikipedia.org/wiki/MongoDB)和[D3.js](https://en.wikipedia.org/wiki/D3.js)等流行库都有第三方头文件。[Node.js](https://en.wikipedia.org/wiki/Node.js)基本模块的TypeScript 标头也可用，允许在 TypeScript 中开发 Node.js 程序。

TypeScript 编译器本身是用 TypeScript 编写的，并[编译](https://en.wikipedia.org/wiki/Source-to-source_compiler)为 JavaScript。它是根据[Apache License 2.0](https://en.wikipedia.org/wiki/Apache_License_2.0)获得许可的。[TypeScript 与C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))和其他 Microsoft 语言一起作为一流的编程语言包含在[Microsoft Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) 2013 Update 2 及更高版本中。官方扩展还允许 Visual Studio 2012 支持 TypeScript。[C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))的首席架构师、[Delphi](https://en.wikipedia.org/wiki/Delphi_(programming_language))和[Turbo Pascal](https://en.wikipedia.org/wiki/Turbo_Pascal)的创建者[Anders Hejlsberg](https://en.wikipedia.org/wiki/Anders_Hejlsberg)一直致力于 TypeScript 的开发。

## 3.2 React



React（也称为 React.js 或 ReactJS）是一个免费的开放源代码前端 JavaScript工具库，用于基于 UI 组件构建用户界面。它由 Meta（前身为 [Facebook](https://zh.wikipedia.org/wiki/Facebook" \o "Facebook)）和一个由个人开发者和公司组成的社群维护。 React 可用作开发具有 [Next.js](https://zh.wikipedia.org/wiki/Next.js) 等框架的单页、手机或服务器渲染应用程序的基础。然而，React 只专注状态管理和将状态渲染到 DOM，因此创建 React 应用程序通常需要使用额外的工具库来进行路由实现，以及某些客户端功能。

声明式

React 使创建交互式 UI 变得轻而易举。为你应用的每一个状态设计简洁的视图，当数据变动时 React 能高效更新并渲染合适的组件。以声明式编写 UI，可以让你的代码更加可靠，且方便调试。

组件化

构建管理自身状态的封装组件，然后对其组合以构成复杂的 UI。由于组件逻辑使用 JavaScript 编写而非模板，因此你可以轻松地在应用中传递数据，并保持状态与 DOM 分离。

一次学习，随处编写

无论你现在使用什么技术栈，在无需重写现有代码的前提下，通过引入 React 来开发新功能。React 还可以使用 Node 进行服务器渲染，或使用 [React Native](https://reactnative.dev/) 开发原生移动应用。

## 3.3 Apifox

[Apifox](https://www.apifox.cn/) 是 API 文档、API 调试、API Mock、API 自动化测试一体化协作平台，定位 Postman + Swagger + Mock + JMeter。通过一套系统、一份数据，解决多个系统之间的数据同步问题。只要定义好 API 文档，API 调试、API 数据 Mock、API 自动化测试就可以直接使用，无需再次定义；API 文档和 API 开发调试使用同一个工具，API 调试完成后即可保证和 API 文档定义完全一致。高效、及时、准确！



Apifox 功能

1. 接口设计：Apifox 接口文档遵循 [OpenApi](https://www.openapis.org/" \t "https://www.apifox.cn/help/app/introduce/_blank) 3.0 (原 Swagger)、[JSON Schema](https://json-schema.org/" \t "https://www.apifox.cn/help/app/introduce/_blank) 规范的同时，提供了非常好用的可视化文档管理功能，零学习成本，非常高效。并且支持在线分享接口文档。
2. 数据模型：可复用的数据结构，定义接口返回数据结构及请求参数数据结构（仅 JSON 和 XML 模式）时可直接引用。支持模型直接嵌套引用，直接 JSON/XML 智能导入，支持 oneOf、allOf 等高级组合模式。
3. 接口调试：Postman 有的功能，比如环境变量、前置/后置脚本、Cookie/Session 全局共享 等功能，Apifox 都有，并且比 Postman 更高效好用。接口运行完之后点击保存为用例按钮，即可生成接口用例，后续可直接运行接口用例，无需再输入参数，非常方便。自定义脚本 100% 兼容 Postman 语法，并且支持运行 javascript、java、python、php、js、BeanShell、go、shell、ruby、lua 等各种语言代码。
4. 接口用例：通常一个接口会有多种情况用例，比如参数正确用例、参数错误用例、数据为空用例、不同数据状态用例等等。运行接口用例时会自动校验数据正确性，用接口用例来调试接口非常高效。
5. 接口数据 Mock：内置 [Mock.js](http://mockjs.com/) 规则引擎，非常方便 mock 出各种数据，并且可以在定义数据结构的同时写好 mock 规则。支持添加“期望”，根据请求参数返回不同 mock 数据。最重要的是 Apifox 零配置 即可 Mock 出非常人性化的数据，具体在本文后面介绍。
6. 数据库操作：支持读取数据库数据，作为接口请求参数使用。支持读取数据库数据，用来校验(断言)接口请求是否成功。
7. 接口自动化测试：提供接口集合测试，可以通过选择接口（或接口用例）快速创建测试集。目前接口自动化测试更多功能还在开发中，敬请期待！目标是： JMeter 有的功能基本都会有，并且要更好用。
8. 快捷请求：类似 Postman 的接口调试方式，主要用途为临时调试一些无需文档化的接口，无需提前定义接口即可快速调试。
9. 代码生成：根据接口及数据数据模型定义，系统自动生成接口请求代码、前端业务代码及后端业务代码。
10. 团队协作：Apifox 天生就是为团队协作而生的，接口云端实时同步更新，成熟的团队/项目/成员权限管理，满足各类企业的需求。

## 3.4 UMI



Umi，中文可发音为乌米，是可扩展的企业级前端应用框架。Umi 以路由为基础的，同时支持配置式路由和约定式路由，保证路由的功能完备，并以此进行功能扩展。然后配以生命周期完善的插件体系，覆盖从源码到构建产物的每个生命周期，支持各种功能扩展和业务需求。Umi 是蚂蚁金服的底层前端框架，已直接或间接地服务了 3000+ 应用，包括 java、node、H5 无线、离线（Hybrid）应用、纯前端 assets 应用、CMS 应用等。

它主要具备以下功能：

1. 可扩展，Umi 实现了完整的生命周期，并使其插件化，Umi 内部功能也全由插件完成。此外还支持插件和插件集，以满足功能和垂直域的分层需求。
2. 开箱即用，Umi 内置了路由、构建、部署、测试等，仅需一个依赖即可上手开发。并且还提供针对 React 的集成插件集，内涵丰富的功能，可满足日常 80% 的开发需求。
3. 企业级，经蚂蚁内部 3000+ 项目以及阿里、优酷、网易、飞猪、口碑等公司项目的验证，值得信赖。
4. 大量自研，包含微前端、组件打包、文档工具、请求库、hooks 库、数据流等，满足日常项目的周边需求。
5. 完备路由，同时支持配置式路由和约定式路由，同时保持功能的完备性，比如动态路由、嵌套路由、权限路由等等。
6. 面向未来，在满足需求的同时，继续对新技术进行探索。比如 dll 提速、modern mode、webpack@5、自动化 external、bundler less 等等。

## 3.5 Ant Design Pro



Ant Design Pro 是基于 Ant Design 和 umi 的封装的一整套企业级中后台前端/设计解决方案，致力于在设计规范和基础组件的基础上，继续向上构建，提炼出典型模板/业务组件/配套设计资源，进一步提升企业级中后台产品设计研发过程中的『用户』和『设计者』的体验。

Ant Design Pro 在力求提供开箱即用的开发体验，为此提供完整的脚手架，涉及[国际化](https://umijs.org/zh-CN/plugins/plugin-locale" \t "https://pro.ant.design/zh-CN/docs/_blank)，[权限](https://umijs.org/zh-CN/plugins/plugin-access)，mock，[数据流](https://umijs.org/zh-CN/plugins/plugin-model" \t "https://pro.ant.design/zh-CN/docs/_blank)，[网络请求](https://umijs.org/zh-CN/plugins/plugin-request)等各个方面。为这些中后台中常见的方案提供了最佳实践来减少学习和开发成本。

## 3.6 Robot



RoboForm是谷歌应用商店的一款扩展应用程序，可以安全地储存密码，遇到一长串的网页表单时，AutoFill功能可帮助省下输入个人和账单资讯的时间。

**什么是自动填表功能？**

表单填充器是即时填写 Web 表单的软件。对必须反复不断地键入相同的联系人和支付信息感到厌倦？借助 RoboForm 的一键式表单填充器，可以毫不费力地准确输入您的地址、银行账号、信号卡和更多信息。您需要在网上例行提供的几乎任何信息，都可以利用 RoboForm 进行存储和填写！

**表单填充器可用在什么地方？**

表单填充功能特别适合网上购物、简历、竞赛、纳税申报表等！对于那些独一无二的表单，使用我们的自定义字段功能。即使最奇特的在线表单对 RoboForm 来说也不在话下。在您的所有设备上使用 RoboForm。适用于 Windows、Mac、iOS、Android、Linux 和 Chrome OS。适用于所有主要浏览器的扩展程序。

**RoboForm 的表单填写功能是否安全？**

RoboForm 使用只有您知道的密钥，在您的设备上本地加密您的所有信息。这意味着您可以安全地存储、访问和更新您的个人数据。在填充信用卡和社会保障号码等敏感字段之前，RoboForm 甚至会要求您进行确认。

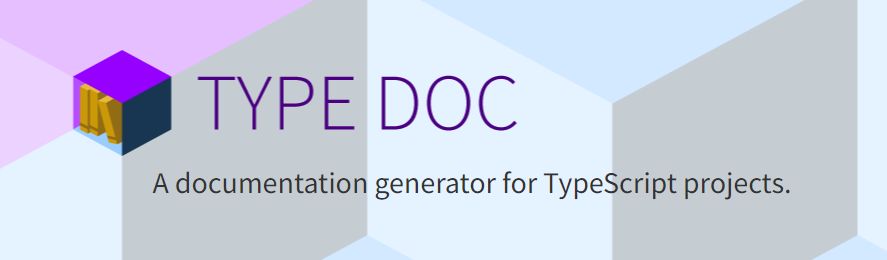
## 3.7 Selenium



Selenium是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。支持的浏览器包括IE（7, 8, 9, 10, 11），Mozilla Firefox，Safari，Google Chrome，Opera，Edge等。这个工具的主要功能包括：测试与浏览器的兼容性——测试应用程序看是否能够很好得工作在不同浏览器和操作系统之上。测试系统功能——创建回归测试检验软件功能和用户需求。支持自动录制动作和自动生成.Net、Java、Perl等不同语言的测试脚本。

本项目中使用的是Python脚本测试，配合Python Unit Test进行测试工作。

## 3.8 TypeDoc



TypeDoc是一个面向TypeScript语言的文档生成工具。在本项目中与Gulp配合使用，利用注释生成代码文档。

## 3.9 Gulp

A picture containing text, tableware, cup, paper cup

Description automatically generated

Gulp （Gulp.js）是基于 node 实现 Web 前端自动化开发的工具，利用它能够极大的提高开发效率。 在 Web 前端开发工作中有很多“重复工作”，比如压缩CSS/JS文件。而这些工作都是有规律的。找到这些规律，并编写 gulp 配置代码,让 gulp 自动执行这些“重复工作”。

## 3.10 Python Unit Test

Python的单元测试框架。用于执行测试。

## 3.11 Beautiful Report

用于生成单元测试报告。