工程信息精细化管理平台架构设计研究

饶平平[1]　沈　益[1]　朱清鹅[1]　沈程琳[2]

（1. 上海理工大学　环境与建筑学院，上海　200093；2. 上海同筑信息科技有限公司，上海　200093）

【摘　要】本文通过采用前后端分离原则，研究联合多项平台开发底层技术，提出了精细化管理平台开发的架构设计。利用 Vue + Elementui + Webpack 技术搭建平台的前端；利用 SpringCloud + MariaDB + Logback 技术搭建平台后端；分离模式下搭建前后端后，通过后端提供的 API 接口，进行前后端数据连接研究；利用 Flutter、SQLite 数据库、Android 和 IOS 系统进行平台移动端的搭建设计；通过数据库选型、建设、安全、接口设置，为平台数据的管理提供技术支撑。文章通过前端、后端、手机端和数据库管理搭建的研究，设计了一套精细化管理平台，有利于实现工程信息的集成、传递、共享，便于参建各方随时随地查看、修改、存储、传递工程信息，为提高工程项目管理的效率和精细度提供了技术支撑参考依据。

【关键词】前端；后端；移动端；数据库；平台

【中图分类号】TU17　【文献标识码】A

引言

随着社会经济的发展，为了适应高速发展经济需求，计算机、网络等技术被越来越多的行业所应用，并带来了行业颠覆性的影响。例如传统集市 + 互联网→淘宝；传统银行 + 互联网→支付宝；传统交通 + 互联网→滴滴快车等。通过利用云计算、大数据、物联网等实现了资源共享与信息实时传递，极大地促进电子商务和互联网金融健康发展，引导互联网企业拓展国际市场[1]。在建筑行业，工程项目具有建设周期长、涉及专业多、环境复杂等特点，这使得建筑行业数据和信息多而复杂，然而这些数据因粗放式的管理，没有产生数据应有的价值[2]。随着 BIM、云计算、大数据、物联网等技术的出现，将 BIM 模型作为数据信息的载体，通过互联网模式，实现建筑信息的采集和分析，形成数据库，保障信息的及时性、准确性、唯一性[3-4]。

市面上多数工程的环境或安全监管平台解决了监测数据散乱的问题，如现场监控系统、安全指数监测平台等实现了施工现场安全信息的收集和查看，但此类平台往往缺乏日常流程管理功能，同时也在数据分析处理、安全预警、消息传送等功能方面少有涉及；而对于多数工程业务流程管理平台，往往存在日常运行故障多、低移动化、操作冗杂度高等问题，这不利于工程管理平台朝向智能化发展。

因此我们需要建立一个运行流畅、架构安全的精细化平台，实现工程建设的信息化、精细化、规范化管理，从而达到工程信息的实时共享，提高项目管理效率的目的。

1　平台总体架构

精细化管理平台总体架构由五个层次组成，分别为：由网络基础设施和计算机软硬件基础设施两部分组成的基础设施层；包含 BIM 数据库、GIS 数据库、管理数据库、共享数据库的数据层；包含前后端分离技术、Vue 前段框架技术、Springcloud 等后端框架技术、移动端技术的支撑层；包含进度、质量、安全、物料、文档、模型管理六个功能模块的应用层，该层即为精细化管理实际内容的表现；由业主方、施工方、监理方等项目参建单位组成的用户层，为精细化管理组织成员的具体表现。总体架构图如图 1。

【作者简介】饶平平(1984 -)，男，副教授，主要研究方向：BIM 技术研究与应用。

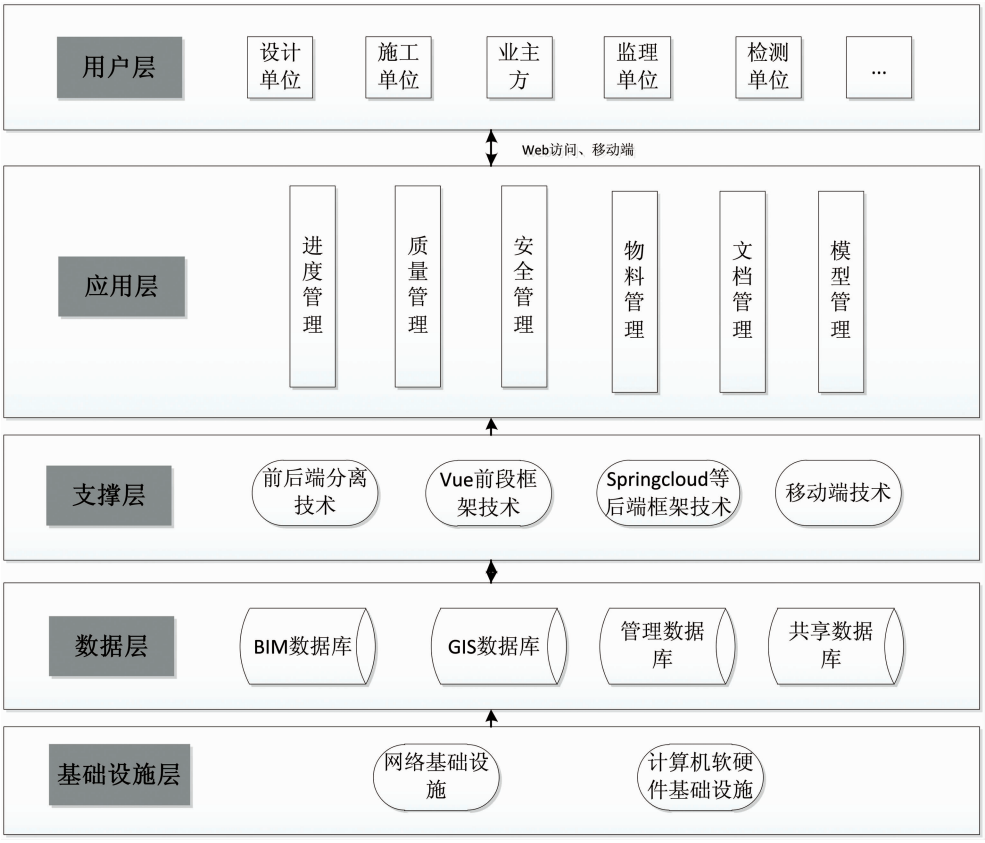

图 1　平台总体架构

2　前后端分离原则

精细化管理平台采用前后端分离进行搭建，前端负责 UI 交互层设计、数据的展现、程序应用逻辑及逻辑渲染、优化用户体验、页面渲染等；后端负责服务层开发、提供数据、存储数据、保障数据传输的稳定性等。前后端最后通过接口实现数据传递和交互。

多数工业管理平台存在前后端某一端业务薄弱的情况，这往往是由于平台开发时，开发人员一起开发前后端，开发存在侧重点。而本研究构想的平台采用开发分离原则，前后端在分离开发模式下，前后端人员各司其职，同时进行开发，可以缩短平台开发时间，提高开发效率。在前后端人员双方了解业务逻辑情况下，前后端人员可以不依赖彼此进行单独开发和调试，从而减少前后端的沟通，有效地降低沟通成本。前端不需要编写任何后端代码，只需要调用后端提供的接口，就可实现前后端的连接，从而避免前后端代码耦合、混杂等现象。

采用前后端分离，后端提供数据接口，手机、平板等不同终端只需要根据数据接口编程，就能实现多个终端与同一个后端的连接，避免开发者针对每一个终端都开发相同的后端服务的重复工作，提高开发者的工作效率，如图 2 终端与后端的连接。

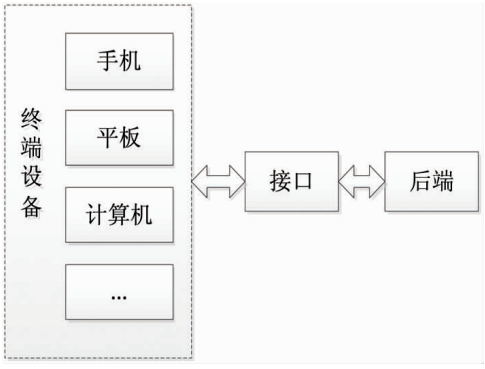

图 2　终端与后端连接

平台前后端分离采用单页面应用 SPA（Single-page application），单页面应用将前后端关注点进行分离，前端负责显示，后端负责数据的计算和存储，明确前后端的逻辑，加载时仅根据需求进行局部更新，提高了加载速度和用户体验感。

3　前端搭建

3.1　相关技术介绍

(1)Vue 技术

Vue 是当前使用广泛的开源的渐进式 JavaScript 框架,可以自由与多种第三方库或第三方插件进行对接,包含多种附带组件,如 Vue-cli、Vue-router 等。

1)Vue. js 前端开发框架

Vue. js[5]是近年来出现的一款轻量级前端框架,相比于 Angular JS 前端框架,Vue. js 更简单、更灵活、更容易优化、性能更好,因此 Vue. js 运用于前端框架被越来越多的人认可。与 Vue. js 相似并被多数工程管理平台应用的前端开发框架 React,其与 Vue. js 区别主要是内部实现本质不同,React 通过对虚拟 DOM 进行渲染,将数据保存在内存中,当状态发生改变时,将会重新对虚拟 DOM 进行渲染,通过补丁的形式将变化部分加载到虚拟 DOM 节点上,并且不是每次改变都渲染整个页面。而 Vue. js 则是以 DOM 为模板,真正操作在 DOM 上,将数据与真实的节点进行绑定[6]。Vue. js 的这种真实 DOM 比 React 的虚拟 DOM 性能更好,能为前端多功能模块提供精细度高的界面显示,更能满足本平台前端开发。

除此之外,Vue. js 作为前端开放框架主要是因为其具有模块化、组件化、响应式数据双向绑定、路由功能和状态管理特点。

模块化:在 JavaScript 刚开始运用于前端开发时,不支持程序分解,然后再自由组合。这给开发人员开发大型、复杂的 Web 应用带来很大的阻碍。因此,前端逐渐支持模块化,保证了管理人员基于平台对工程信息的模块式管理。通过 import(导入)模块、export(导出)模块,丰富 Java Script 语言的功能。2015 年所处 JavaScript 最新版本 ECMAScript6. 0(简称 ES6)支持模块化,ES6 模块的静态化设计原则,有效提高模块的加载[7]。Vue. js 支持最新的 ES6 规范,因此利用 Vue. js 作为前端框架,能够很好地体现模块化思想。

组件化:通过组件化,Vue. js 能够将 JS、HTML 和 CSS 等前端开发语言写在一个文件里,且各组件之间不会相互影响,组件之间的关系明确,开发人员清楚划分各组件的功能边界,这有效提高代码可读性和可维护性,有效减少定位并解决开发遇到问题的时间,为工程信息复杂的精细化管理提供了软硬件基础上的流畅性和快捷性。

响应式数据双向绑定:其绑定原理如图 3。与单项数据绑定相比,双向数据绑定减少了增、删、改、查操作,在模型数据与视图显示数据之间,一个数据修改,另一个数据自动随之改变,例如在表单场景中,当填写完表单信息,数据就已经保存到模型数据库,这就省去模型数据增、删、改、查等工作量,节约时间,提高效率。

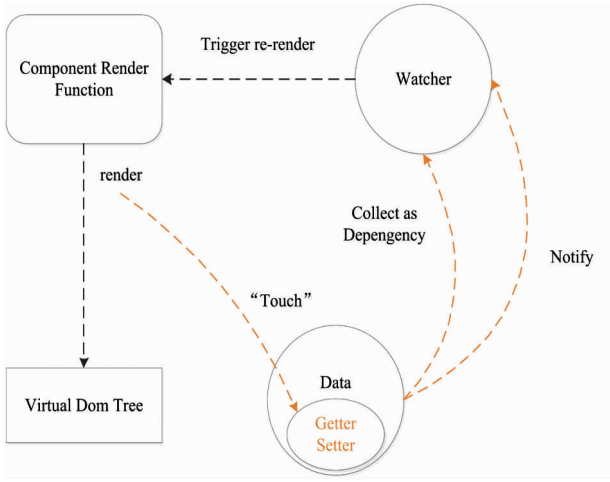

图 3　双向数据绑定原理

路由功能:Vue. js + vue - router 创建单页面应用,通过 Vue. js 的路由功能在路由中配置业务模块,实现单页面跳转。

状态管理:Vuex 专门为解决多个组件共享状态的问题而设计的 Vue. js 框架的状态管理模式,其状态管理原理如图 4,Commit 主动操作 Mutations,Mutations 操作用来存放共享状态的 State。

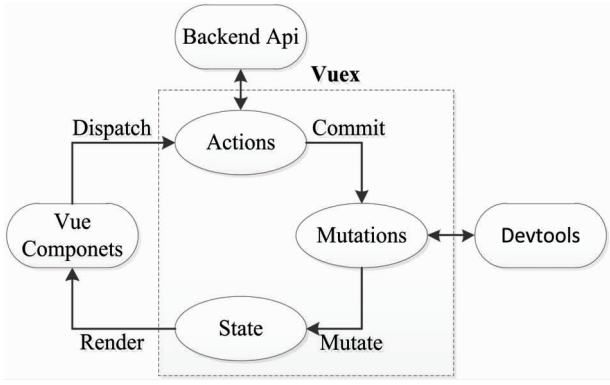

图 4　Vuex 状态管理原理

2)管理平台前端动态页面技术

利用 HTML5 和 CSS3 编写静态页面,然后在静态页面插入 JavaScript 这种能被浏览器解析和运行的脚本语言实现精细化管理平台前端动态页面设计。

HTML5 是 2014 年万维网重新修订的超文本标记语言,它在旧版本的 HTML 基础上添加 < section > 、< header > 和 < nav > 等标签,除此之外还添加音频、视频、画布等,也继承了 SVG 的内容。同样 HTML5也摒弃了旧版本的一些不合理属性和元素,使得 HTML5 更加符合当代网络需求,开发者更容易运用 HTML5 在网页中添加、处理多媒体和图片内容。

CSS3(层叠样式表)是 2001 年 W3C 组织在原版 CSS 基础更新的一门为网页页面添加色彩、样式的计算机语言,它的多背景图、边框、颜色等特性使网页拥有丰富多彩的网页样式。

JavaScript[8]具有解释性、动态性特性,只要将 JavaScript 嵌入到 HTML5 代码中,就能实现 HTML5 页面动态效果。JavaScript 是一种弱类型语言。并且因为 JavaScript 不允许访问本地硬盘文件和对传输文本的修改或删除,防止数据丢失,因此JavaScript 安全性很高。

(2)Elementui 技术

Elementui 技术是近年来为设计者准备的基于 Vue2. 0 的桌面端组件库。其组件丰富,页面简洁大方,解决了智能手机和计算机浏览器前端显示效果不兼容现象。利用 Elementui 技术对前端界面元素进行设计,可以有效解决计算机能够显示完整的网页界面而手机端出现网页界面不全的问题。

目前该技术多为生活购物软件平台使用,尚未被工程管理平台广泛应用,但对于手机端与网页端通用的管理平台是有利无害的。在利用 Elementui 技术对精细化管理平台前端界面进行设计时,前端界面上图表展示使用百度 Echarts。Echarts[9]是国内一款能够兼容大部分浏览器和移动设备的 JavaScript图表库,将其运用到前端界面设计中,通过加载、数据转换、定制、渲染四个步骤,可以生成直观、生动的计划统计、产值统计等图表,这使得平台界面的管理内容得到了深度的细化。

(3)Webpack 技术

随着网络技术的发展,前端页面越来越丰富,所包含的 JS 代码、依赖包等各种文件越来越多。为了方便后期前端应用的维护和提高开发效率,在精细化管理平台中运用 Webpack 对前端资源进行模块化管理,并将 Webpack 做为打包工具[10]。Webpack 能够将前端界面中不能识别的的语言打包转换成前端界面能够识别的格式; Webpack 能够把有依赖关系的各种文件打包成符合前端管理的静态资源。Webpack 的工作原理如图 5。

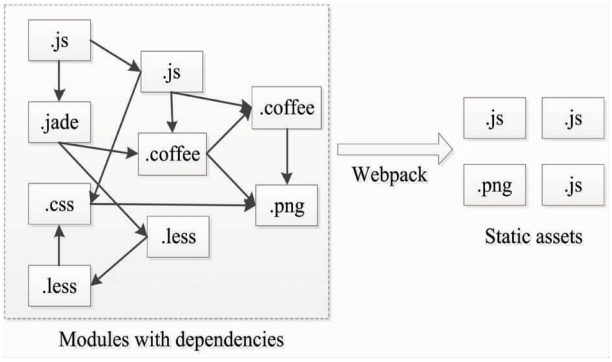

图 5 Webpach 工作原理图

3.2 前端搭建

本文利用 Vue + Elementui + Webpack 技术搭建精细化管理平台的前端,用目前流行、轻量级的 Vue. js 作为该平台前端框架,根据 Vue. js 框架专门设计的 vuex 进行前端跨页面系统的管理,使用 vue-router 进行路由管理,实现前端单页面的跳转,使用 vue-cli 脚手架进行前端页面搭建; 合理运用 HTML5、CSS3、JavaScript 三种网页编程语言,设计出前端动态页面; 运用 Elementui 作为前端页面设计的 UI 框架,设计出符合客服审美的计算机浏览的前端网页界面,同时使用 Echarts 设计前端网页界面图表,使前端网页界面的图表更深度、更形象、更美观; 使用 Webpack 完成项目前端代码的压缩、打包工作,方便后期前端应用的维护和提高开发效率。利用 Vue、Elementui、Webpack 技术,搭建满足功能需求的精细化管理平台前端,如图 6。

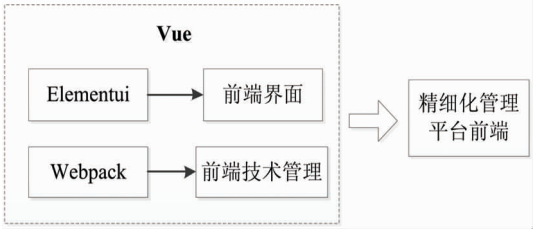

图 6 精细化管理平台前端搭建

4　后端搭建

4.1　Spring Cloud

Spring[11]网站抽象地定义了 Spring Cloud 的概念:Spring 希望集成一组工具,这些工具可以使软件开发人员轻松构建分布式通用组件,协调系统环境并为不同的服务提供模板。Spring Cloud 提供了一组易于使用的 Jaba Spring 库,这些库实现了分布式系统所需的一系列通用模式。Spring Cloud 中的 Cloud 不是云解决方案,只是为开发提供的各种支持组件,为开发人员提供了基于云的分布式软件系统,提高了开发的便捷度。

Spring Cloud[12-13]集成了多个组件,这些组件使得 Spring Cloud 的使用变得更便捷,特别是在分布式系统的开发中。这些组件主要由负责打包现有软件和负责分布式系统研发的补充实现两部分组成。这些组件包括 Spring Cloud Netflix、Spring Cloud Config、Spring Cloud Eureka、Spring Cloud Bus、Spring Cloud Zookeeper、Spring Cloud CLI 等重要内容[14]。众多的组件丰富了 Spring Cloud 的功能,使其能够满足实际研发中的分布式配置、负载平衡、安全性、路由、分布式跟踪、云支持等各种要求。因此利用 Spring Cloud 作为微服务开发套件,能够为整个系统的研发集成便捷、清晰的开发思路。

4.2　MariaDB 技术

MariaDB[15]是 2009 年由 Michael Widenius 主导开发、开源社区维护、采用 GPL 授权的一种小型、快速的关系型数据库管理系统。开发 MariaDB 的目的是完全兼容 MySQL,让 MariaDB 成为 MySQL 的代替品。因此 MariaDB 与 MySQL 一样具有性能高、成本低、可靠性好的特性,除此之外,MariaDB 在扩展功能、存储引擎以及一些新的功能改进方面都比 MySQL 强。自 2009 年以来,随着 MariaDB 的不断成熟,它逐渐从应用在 Internet 上的中小型网站向应用在大规模网站发展。MariaDB 结构如图 7。

4.3　Logback 技术

Logback 是一个高性能的开源日志框架,通过对 Logback 的使用,我们可以控制日志信息传递的目的地是控制台、GUI 组件等,还可以控制每个日志的输出格式,通过定义每个日志信息级别,我们可以更详细地控制日志生成过程。这些功能可以通过一个配置文件灵活地配置,而不需要修改应用程序代码。对于精细化管理涉及的进度、安全、质量、成本等多要素,平台运行时需要读写大量的数据日志,而 Logback 使用内存和本地磁盘缓存可以最大限度地传输具有高强度和大量数据的日志,为精细化管理提供了有力的后台技术支撑。Logback 本地日志缓存无法执行内容搜索,最终需要将日志数据存储在数据库中。

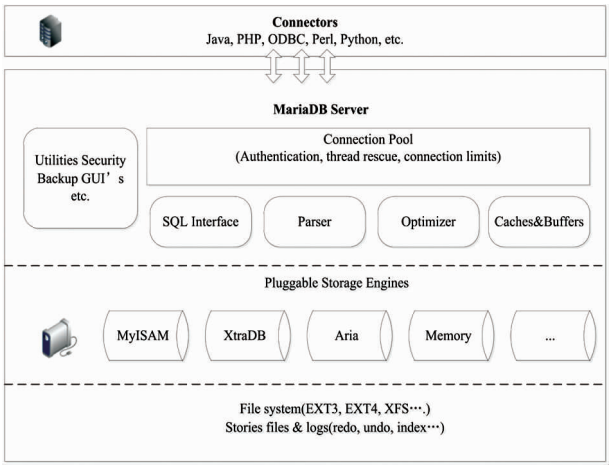

图 7　MariaDB 结构

4.4　后端搭建

本文利用 SpringCloud + MariaDB + Logback 技术搭建精细化管理平台后端。SpringCloud 是当前使用最广泛的微服务开发套件,内部组件满足了大部分的开发需求,并且也能很好地和第三方框架结合,实现自定义功能。因此采用该技术,发挥了开发套件通用性,保证平台组件更新迭代的适用。

在后端设计中,SpringCloud 作为基础框架;SpringCloud 的子项目 Netflix Eureka 作为服务注册机,保存各类 Web 应用的相关信息;SpringCloud 提供 feign 框架作为客服端请求具体业务;SpringCloud 利用 Gatway 分配前端发送过来的请求信息。MariaDB 是数据库 MySQL 的一个分支,完全兼容 MySQL。MariaDB 利用 XtraDB、Aria 等存储引擎为基础,通过 SQL Interface、Parser、Optimizer、Caches & Buffers 对前端传来的信息进行接收、解析、优化、缓存,将最后得到的信息存储到 MariaDB 数据库中。Logback 是开源日志框架,完美兼容 SpringCloud,作为 SpringCloud 的默认日志系统,可以完全零配置使用 SLF4J(Simple Logging Facade For Java)的 logback 来输出日志。将 Logback 作为后端日志框架,日志通过 Logback 保存在 MariaDB 数据库中,有效避免

日志服务直接将日志写入 MariaDB 数据库出现的应用服务器线程阻塞、服务宕机现象。以 SpringCloud 作为基础框架,MariaDB 做为数据库管理,Logback 作为记录日志,搭建精细化管理平台后端,如图 8。

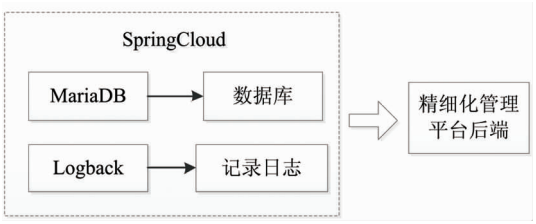

图 8　精细化管理平台后端搭建

5　移动端搭建

5.1　Flutter 移动 UI 框架

Flutter 是 Google 在 2018 年推出的移动 UI 框架。借助 Flutter,可以在 Android 和 IOS 上快速构建高质量的本机用户界面,并以每秒 120 帧的速度连续渲染。它允许用户自定义设计,具备许多优点,使得 Flutter 这一款移动端 UI 框架迅速进入开发者的视野,得要越来越多开发者的使用。

同样,相较于许多平台采用的 React 移动端框架,Flutter 不仅跨平台,而且性能好、使用流畅。相比于 React Native 移动端框架,不仅跨平台,而且代码量不大,易于初学者学习,在动画性能方面体验好。此外,Flutter 可以通过其内置的精美质感设计,Cupertino(ios-flavor)小工具和丰富的动画 API 来构建令人满意的移动终端界面;使用 Flutter 的热重载可以快速、轻松地测试、重构 UI、添加功能和修复 bug。Flutter 利用 API,有效解决 2D、动画、手势、效果等难题,唯一不足的是该移动端框架在处理多类型数据时数据吞吐量有一定限制。

Flutter 使用 Dart 编程语言进行编译,并且在开发阶段采用 JIT 模式[16]。如果发生更改,则无需编译,这有助于减少 Flutter 的开发时间。发布时,AOT 用于生成有效的 ARM 代码以保证 Flutter 应用程序的性能。Dart 虚拟机可以快速为短暂的对象分配内存,从而提高渲染速度。

5.2　SQLite 轻型数据库

SQLite 是 D. Richard Hipp 创建的、运用 C 语言编写的一款公共开源数据库引擎。因其速度快、可移植性好、占用内存少等优点,得到大量使用的同时受到广泛关注。SQLite 是独立的、零配置的嵌入式数据库引擎,与其它数据库引擎区别在于它没有自己独立的服务进程,而是直接读写磁盘上的数据库文件。SQLite 数据库是包含表、索引等的独立文件,其具有跨平台的特性,可以将 32 位设备上的 SQLite 数据库直接应用到 64 位设备上。

SQLite 是紧凑型数据库,在应用时,可以将其占用的存储空间容量大小一直控制在 500KB 以内,也可以根据需求,将该容量值进一步缩小到 300KB 以内。在如此小的空间内,SQLite 运行速度、性能仍然能够正常进行,使得手机、平板等内存有限的设备将其作为数据库。

5.3　Android 和 IOS

(1)Andriod 操作系统

Andriod 操作系统最初由 Andy Rubin 以 Linux 为基础开发的一款开源框架。Andriod 又名“安卓”,自 2005 年被 Google 收购后,对其进行改善和开发,现已经完全适用于 Google 的任何一款软件产品。对于用户而言,Andriod 操作系统为生活带来了便利,如在 Andriod 操作系统上安装谷歌地图,便于用户出行;对于开发者而言,可以根据自己的喜好和风格设计产品,可以重复使用或替换应用程序框架组件,除此之外,Andriod 操作系统利用 SQLite 数据库进行数据存储、利用 Java 进行应用程序语言编写、基于 Web 引擎开发 Andriod 浏览器,这些都有助于开发者很快入门;对于手机生产商而言,可以体现产品个性。因此 Andriod 操作系统受到广大用户、开发者、手机生产商的喜爱和支持。

(2)IOS

IOS(iPhone Operation System 的简写)即 iPhone 操作系统,专为 iPhone、iPad 等苹果公司推出的移动设备而开发的操作系统。IOS 是以 Darwin 为基础,利用编程语言 Object-C 和 Swift 在 Xcode 环境下开发的 iPhone 操作系统,并且必须在 Mac OS X 上运行。Mac OS X 是以 BSD(Berkeley Software Distribution)Unix 内核开发为基础,利用 SDK(Soft Ware Development Kit)软件开发工具包进行开发的一款封闭式的操作系统。

5.4　移动端搭建

由于 SQLite 数据库资源占用低、运行速度快、可移植性好、能够跟很多程序语言相结合,且提供零配置的运行模式,将 SQLite 数据库嵌入到 Andri-

od、IOS 系统中,实现对数据的存储、管理、维护。运用 Flutter 在 Android 和 IOS 系统上,建立满足客服需求、满足程序应用功能需求的美观手机移动端界面。本文以 Andriod、IOS 系统做为基础,将 SQLite 数据库嵌入 Andriod、IOS 系统作为移动管数据库管理系统,Flutter 作为移动端界面设计,搭建精细化管理平台的移动端,如图 9。

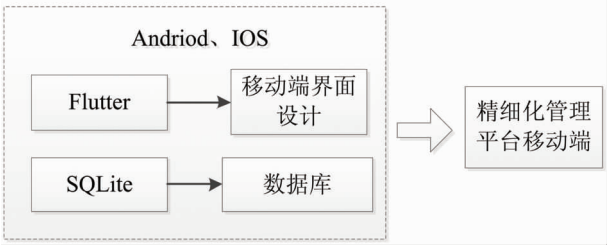

图 9　精细化管理平台移动端搭建

6　平台数据架构

6.1　数据库选型

为了保证平台信息的收集、存储、管理和维护,在平台搭建中,数据库管理系统是必不可少的部分。同时,为了保证数据安全性、完整性、分布、异构等性能,应根据实际需求,选择恰当的数据库类型。根据平台功能的实际需求,数据库的建立需要满足支持各种流行的软硬件平台、支持 Web 技术并具有多媒体处理能力、支持联机分析处理、支持异种数据库的互连、具有比较高的安全性和可靠性、具有数据仓库的性能、支持商务智能和数据挖掘的基本要求。MariaDB 数据库系统具有支持多处理器、支持 SQL 的 GROUP BY 和 ORDER BY 子句、支持聚合函数、支持大型的数据库、支持多种存储引擎、没有内存漏洞、提供 ODBC 和 JDBC 等多种数据库连接途径、提供用于优化数据库操作的管理工具、可以修改源代码来开发自己的 MariaDB 系统等优点,尤其支持复杂信息的高速细分和处理指令的大容量发送,完全契合精细化管理平台开发对数据库的选型要求,所以本文数据库建设采用 MariaDB 为核心的数据库。

6.2　数据库建设

数据库主要包含 BIM 数据、GIS 数据、管理数据、共享数据四部分,分别对标工程精细化 BIM 模型、GIS 信息、管理处理指令、共享文档这四个管理数据,整个数据架构图如图 10。

(1)BIM 数据库

此处所指信息数据包括建筑设计平面图、剖面图、材料表等等。模型基于 3D 技术,是项目信息的详细表达。它解决了用软件描述建筑信息的问题,为项目的协同工作和精细管理奠定了坚实的基础。

(2)GIS 数据库

该数据展示 BIM 项目工程周边建筑、道路等环境,对建筑周边有直观的了解。

(3)数据集成

在数据集成中,主要采用以下几种方法进行数据集成:

1)对于结构化数据,采用数据整合服务进行数据集成。

2)对于无法采用数据整合工具进行集成的数

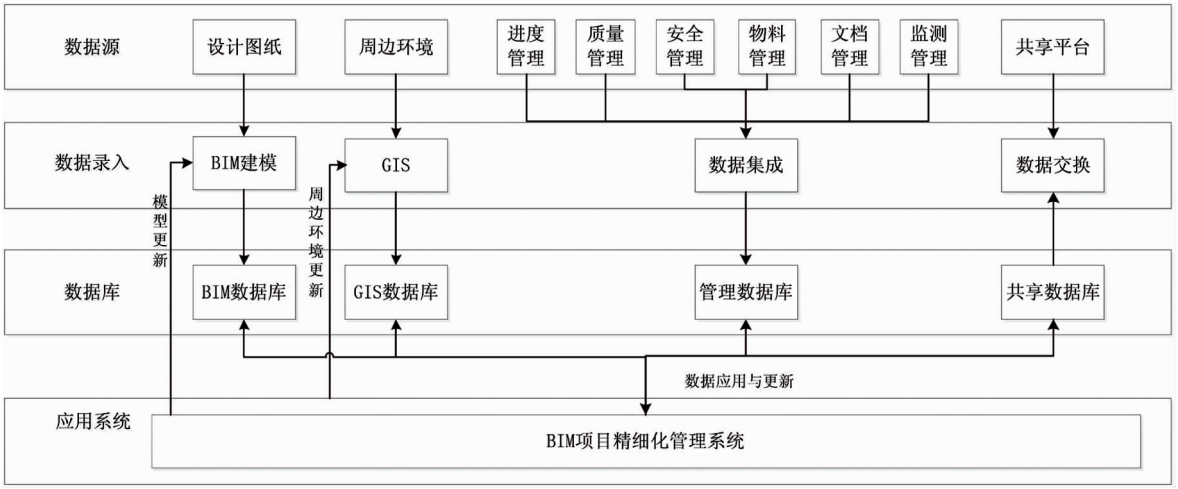

图 10　数据系统架构图

据,如 GIS 数据、BIM 模型数据,必须采用人工方式进行整合。通过流程控制工具,对人工数据整合过程进行流程化管理,保证人工数据处理质量。

(4)共享数据库

在精细化管理平台中,各参建方能够对设计图纸、工程日志等数据进行下载,对一些共享数据精细数据交换机更新。

6.3 数据库安全

由于工程上平台用户往往同用一个或几个账号,而多数管理平台对于账号管理过于宽容,这使得数据库系统主要面临以下威胁:合法特权的过度滥用、特权提升、数据库漏洞和数据泄露、数据库通信协议漏洞、不健全的认证和审计等。因此,为确保数据库系统安全运行,保证数据安全,数据库系统进行以下安全设计:

(1)对所有数据库用户建立数据库账号,并将数据库账号与用户身份进行关联;

(2)限制用户登录不成功次数,进行身份验证失败处理;

(3)操作系统用户与应用进程绑定,实现用户级追溯;

(4)对数据库访问日志进行记录;

(5)对数据库访问进行授权控制;根据用户名和用户组对用户权限进行控制,控制粒度可选择为目录、表、记录、字段;

(6)采用专用审计设备,对数据库的安全日志和使用记录进行审计。

6.4 数据库接口

本平台数据接口类型分为三种,分别为:用户接口、外部接口、内部接口。

用户接口:本平台用户接口分为 PC 浏览器,手机端 APP(Android,iOS)。数据录入使用 jsp 表单,以上传和下载的方式进行文件导出和导入,通过客户端控件嵌入进行图像显示和更新。

外部接口:本平台的外部接口主要是文档接口,使用数据交换格式文件,清晰地理解及掌握各模块功能。

内部接口:Elementui、HTML、CSS 等层次编译为一个应用,托管于 apache,通过浏览器应用运行。

精细化管理平台主要采用 Web Service、DB、文件接口 FILE、FTP(File Transfer Protocol 文件传输协议的简称)四种接口形式。

7 结论

本文通过采用前后端分离原则,联合多项平台开发底层技术,研究平台架构设计,构想了精细化管理平台的开发,得出以下结论:

(1)利用 Vue + Elementui + Webpack 技术,通过发挥真实 DOM 性能来提升开发效率,可搭建轻量级平台的前端。

(2)利用 SpringCloud + MariaDB + Logback 技术搭建平台后端,发挥技术的模块式优势来简化后台数据处理流程,这给平台分布式管理提供便捷性开发依据。

(3)利用 Flutter、SQLite 数据库、Android 和 IOS 系统对平台移动端的搭建进行设计,通过数据库选型、建设、安全、接口设置,给平台数据的稳定传输提供了理论支撑。

(4)通过后端提供的接口,将前端、手机端与后端、数据库连接,来完成精细化管理平台的搭建,有利于实现工程信息的集成、传递、共享,便于参建各方随时随地查看、修改、存储、传递工程信息,从技术角度上为提高工项目管理的效率和精细度提供了参考。

参考文献

[1] 赵媛媛. 基于 BIM 技术与“互联网 +”的项目协同管理平台的研究[J]. 科技研究,2018(8):24-28.

[2] 李志龙, 张智云,黄少惠,等. 基于 BIM 标准的大数据服务平台研究[J]. 建筑建设科技,2018,1-3.

[3] 弓瑞强. 基于 BIM 的地铁工程精细化管理[D]. 华侨大学,2018.

[4] David Bryde. The project benefits of Building information Modelling(BIM)[J]. International Journal of Project Management,2013(3):1-10.

[5] 王金滔. 校园闲置商品交易平台的设计与实现[J]. 电脑知识与技术,2018(36):74-77.

[6] 李广中. 利用计算机网络构建人力资源信息系统的实践探[J]. 计算机光盘软件与应用,2014(7):30-31.

[7] Likness J. Model-view-viewmodel(mvvm) explained[J]. Code Project, 2010,8(21): 2-9.

[8] Ellingwood,Justin. How To Use NPM to Manage Node. js Packages on a Linux Server[M]. Digital Ocean,2016-10-22/2016-12-20.

[9] 陆遥. 数据可视化探索系统的设计和实现[D]. 浙江大学,2016.

[10] 江庆. Vue + Webpack 框架在银行 App 前端开发的应用[J]. 金融科技时代,2016,16(11): 15-19.
[11] Geary D,Horstmann C. Core JavaServer Faces[M]. Addison Wesley,2004.
[12] Jakub Kozik, Dmitry Shabanov. Improved algorithms for colorings of simple hypergraphs and applications[J]. Journal of Combinatorial Theory, Series B, 2016 (116): 413-420.
[13] 郝伟华. 浅析人力资源管理信息系统的应用[J]. 企业导报,2013(15):170-171.
[14] 王磊. 微服务架构与实践[M]. 电子工业出版社,2016.
[15] 刘韬. Free RADIUS 服务与 Maria DB 数据库的集成[J]. 信息技术与信息化,2017(3):52-54.
[16] 翁子欣, 吴明晖. 基于 Flutter 的图片风格转换 App 设计与实现[J]. 计算机时代,2020(2):67-71.

Research on Architecture Design of Engineering Information Refined Management Platform

Rao Pingping[1], Shen Yi[1], Zhu Qing'e[1], Shen Chenglin[2]

(1. *School of Environment and Architecture*, *University of Shanghai for Science and Technology*, *Shanghai* 200093, *China*;
2. *Shanghai Tongzhu Information Technology Co., Ltd.*, *Shanghai* 200093, *China*)

Abstract: By adopting the principle of separation of front and back ends, this paper studies the underlying technology of joint development of multiple platforms, and proposes an architecture design for the development of refined management platforms. Using Vue + Elementui + Webpack technology to build the front-end of the platform; using SpringCloud + MariaDB + Logback technology to build the back-end of the platform; after the front-end and back-end are built in the separated mode, the front-end and back-end data connection research is carried out through the API interface provided by the back-end. Using Flutter, SQLite database, Android and IOS systems to carry out the construction and design of the platform mobile terminal; through database selection, construction, security, and interface settings, which provides technical support for the management of platform data. Through the research on front-end, back-end, mobile phone and database management, the article designs a set of refined management platform which is conducive to the integration, transmission and sharing of engineering information, and convenient for all parties to view, modify, store, and trasfer engineering information anytime and anywhere. The platform also provides a reference and technical support for improving the efficiency and precision of engineering project management.

Key Words: Front End; Back End; Mobile Terminal; Database; Platform