

毕业设计(论文)

|  |  |
| --- | --- |
| 课题名称 | 数据可视化分析平台 |
|  | ——Web的设计与实现 |
| 二级学院 | 计算机与软件学院 |
| 专 业 | 软件技术 |
| 班 级 | 软件1851 |
| 学 号 | 1802343124 |
| 姓 名 | 张浩杰 |
| 指导教师 | 许莫淇 |

年 月 日

计算机与软件学院

毕业设计（论文）诚信承诺

我谨在此承诺：本人所写的毕业论文《数据可视化分析平台——Web的设计与实现》，系本人独立完成，没有抄袭行为，凡涉及其他作者的观点和材料，均作了注释与说明，若有不实，后果由本人承担。

承诺人（签名）：

年 月 日

摘 要

现在是一个数据大爆炸的时代，许多人会通过手机和电脑上网浏览信息，并使用很多应用，这给用户的生活和工作带来了许多便利，与此同时应用中也留下了用户的很多信息。随着信息量指数级的暴增，人们已经没有精力浏览全部的信息，为了可以浏览到更加符合自己需求的信息使得应用推送的消息越来越精准，分析在应用中留下的用户信息已经成为开发应用中至关重要的一环。[1]

如何精准的分析数据走向生动展现出数据分析结果并且高效的从数据中挖掘有价值的信息成为了每个数据分析师最迫切的需求。数据可视化平台就可以完美解决数据分析师的需求；海量的数据以可视化的形式展现可以辅助大脑快速处理信息；相较于杂乱的数据图表提供干净易于理解的数据信息，做了以下几方面的工作：

1. 研究数据可视化平台的UI设计，使用当前流行的UI框架，设计出符合当前审美和使用习惯的Web界面。
2. Vue结合v-charts以图表的形式生动展现出数据的发展趋势，借助图形化的手段，清晰有效的传达数据信息。
3. 封装请求接口和Vue组件，提高代码的使用效率增强代码健壮性。

关键词：数据可视化平台；数据分析；v-charts

Abstract

Now is an era of big data explosion. Many people will browse the information through mobile phones and computers and use many applications. This brings a lot of convenience to users' lives and work. At the same time, the applications also leave a lot of users. information. With the exponential increase in the amount of information, people no longer have the energy to browse all the information. In order to be able to browse more information that meets their own needs, making the application push messages more and more accurate, analyzing the user information left in the application has become A vital part of the development and application.

How to accurately analyze the data trend and vividly show the results of data analysis and efficiently mine valuable information from the data has become the most urgent need of every data analyst. The data visualization platform can perfectly solve the needs of data analysts; the visualization of massive data can assist the brain to process information quickly; compared to messy data charts to provide clean and easy-to-understand data information, the following work has been done :

1. Research the UI design of the data visualization platform, and use the current popular UI framework to design a Web interface that conforms to the current aesthetic and usage habits.
2. Vue combined with v-charts to vividly show the development trend of data in the form of charts, and use graphical means to clearly and effectively convey data information.
3. Encapsulate request interfaces and Vue components to improve code usage efficiency and enhance code robustness.

Keywords: data visualization platform; data analysis; v-charts

目 录

摘 要 I

Abstract II

第1章 绪论 1

1.1 研究背景和意义 1

1.1.1 研究背景 1

1.1.2 研究意义 1

1.2 研究目标及内容 1

1.2.1 研究目标 1

1.2.2 研究内容 2

1.3 论文所做工作 2

第2章 开发技术与工具 4

2.1 开发技术 4

2.1.1 v-charts 4

2.1.2 Vuex 4

2.1.3 Vue Router 5

2.1.4 Vuetify 6

2.1.5 Node.js 6

2.1.6 Axios 6

2.2 开发工具 6

2.2.1 Visual Studio Code 6

2.2.2 GitHub 6

2.2.3 GoogleChrome 6

第3章 需求分析 7

3.1 系统总体需求 7

3.1.1 系统稳定性分析概述 7

3.1.2 系统特点分析概述 7

3.2 功能性需求 7

3.2.1 用户模块 7

3.2.2 数据源接入模块 8

3.2.3 数据预处理模块 8

3.2.4 数据分析结果绘图模块 9

第4章 系统概要设计 11

4.1 系统流程图 11

4.2 系统总体功能图设计 11

4.3 系统请求接口封装设计 12

4.3.1 创建一个 Axios 对象 12

4.3.2 配置请求拦截 12

4.3.3 配置响应拦截 13

4.3.4 封装 get 方法 13

4.3.5 封装 post 方法 14

4.3.6 封装 put 方法 14

4.4 Vue组件封装设计 14

4.4.1 消息提示弹窗 15

4.4.2 绘图组件 16

4.5 自定义 Vue 指令 17

4.5.1 新建一个 vue 文件，在里面定义一个变量 17

4.5.1 遍历上文中的 orderArr 变量 17

第5章 系统详细设计 18

5.1 用户模块 18

5.1.1 GitHub第三方登录功能设计 18

5.2 数据源接入模块 18

5.2.1 PostgreSQL、MySQL数据库连接功能设计 18

5.2.2 Excel、CSV多文件上传功能设计 19

5.2.3 新建数据包功能设计 19

5.2.4 数据包中添加数据表功能设计 20

5.3 数据预处理模块 21

5.3.1 数据包中数据源图表展示功能设计 21

5.3.2 自定义数据表功能设计 21

5.3.3 选定数据库自定义字段功能设计 22

5.3.4 自定义X/Y轴数据分析功能设计 22

5.3.5 字段别名功能设计 23

5.4 数据分析结果绘图模块 24

5.4.1 v-charts结合Vuex实现折线图、饼图、柱状图功能设计 24

5.4.2 数据分析结果导出功能设计 25

第6章 系统功能的实现 26

6.1 用户模块 26

6.1.1 GitHub第三方登录功能实现 26

6.2 数据源接入模块 26

6.2.1 PostgreSQL、MySQL数据库连接功能实现 26

6.2.2 Excel、CSV多文件上传功能实现 28

6.2.3 新建数据包功能实现 29

6.2.4 数据包添加数据表功能实现 30

6.3 数据预处理模块 31

6.3.1 数据包中数据源图表展示功能实现 31

6.3.2 自定义数据表功能实现 31

6.3.3 选定数据库自定义字段功能实现 32

6.3.4 自定义X/Y轴数据分析功能实现 33

6.3.5 字段别名功能实现 33

6.4 数据分析结果绘图模块 34

6.4.1 v-charts结合vuex实现折线图、饼图、柱状图功能实现 34

6.4.2 数据分析结果导出功能实现 35

第7章 系统测试 36

7.1 测试环境准备 36

7.2 功能性测试 36

7.2.1 用户模块 36

7.2.2 数据源接入模块 36

7.2.3 数据预处理模块 37

7.2.4 数据分析结果模块 38

7.3 系统部署测试 38

第8章 总结和展望 39

8.1 总结 39

8.2 展望 39

致谢 1

参考文献 2

# 绪论

在大数据时代背景下，大数据技术已经贯穿于各行各业当中，大到行为分析，小到数据统计，正逐渐成为一个优秀企业必不可少的重要组成部分。本课题所属项目是一款基于B/S架构的在线数据可视化分析平台，由Web客户端、基于Python的高效数据处理服务器、后台管理系统所组成，本论文所属内容属于该项目中的Web客户端，包括用户登录、数据上传、数据拆分、自定义计算公式、数据分析、自定义图表绘制、数据导出等功能。设计并实现了其中的主要功能模块，包括用户登录、数据上传、数据拆分、数据分析、自定义图标绘制、数据导出功能。

本章将从课题背景、研究意义、研究目标及内容、与传统数据分析的方式对比以及与国内外当前已有的工具进行对比等三个大方面进行展开介绍。

## 研究背景和意义

### 研究背景

目前，全球有52.2亿人使用手机，相当于世界总人口的66.6%，这意味着互联网已经成为人们日常生活的一部分。随着社交媒体普及率的激增，人们上网时间越来越长，根据最新研究显示：互联网用户平均每天所有设备上使用互联网的时间接近7个小时；[2]数据对产品经理来说至关重要，了解用户平均上网时长，可以对比下自己产品平均使用时长，处于什么样的位置，跟同类产品的对比，可以提高和完善自己的产品质量。因此选择一个优质的数据可视化分析平台来分析数据的走向和变化过程十分重要。

### 研究意义

数据可视化平台可以将单纯的数字通过合适的可视化图表类型展现出来，直观展现出数据的变化趋势、峰值可以使决策者更好的决策。数据可视化的图表类型包含:柱状图、条形图、折线图等多种类型，不同的类型图表对应不同的应用场景。[3]数据分析结果以图表的形式展现搭配不同的颜色使数据结果展现更直观、更容易被人理解。

## 研究目标及内容

### 研究目标

本文的研究目标是以Vue为前端框架搭配Vuetify前端组件实现数据可视化分析平台的Web端。该平台可以分析用户上传的数据源；将枯燥的数据信息以折线图或柱状图等基本的图表形式展现出来。用户通过使用该网站可以使数据的展现形式更直观，通过观察图表的曲折变化有利于用户做出决策。该系统还应包含自定义数据库表、自定义字段等功能可以对数据进行简单的处理，最后的图表结果也可以保存在本地。

### 研究内容

本文的研究内容是采用Vuex和v-charts技术相结合实现数据可视化分析平台的Web端。本系统主要包括用户的登录、数据源的接入、数据分析结果的展示以及数据的预处理其中包括：自定义数据库、自定义字段、自定义构建计算指标；在数据可视化平台的研发过程中主要研究的内容有以下几点：

1. 关键技术选型：对比分析三大主流的前端框架，深入了解其页面渲染速度和性能损失情况；从组建的支持范围性、编写API的整洁性记忆文档教程的详细程度[4]；经过比较最终选择Vue作为系统的前端框架。
2. 数据源接入的连接类型选择：由于数据可视化平台需要处理的数据量极大，因此选择PostgreSQL作为数据库连接的主要类型。相较与线程化的MySQL，PostgreSQL是多线程的数据处理模式，在数据量大并发性高的情况下，更能保证系统的稳定性。
3. 渲染图表组件的选择：为了更鲜明的展现出数据分析的结果，数据可视化平台选择v-charts作为图表展现的组件，v-charts是商业级数据图表，提供直观、生动、可交互的数据可视化图表并且兼容当前绝大部分的浏览器。

## 论文所做工作

为了使系统实现功能更加完备，系统性能更加稳定，在开发之前做了如下的工作：

1. 竞品分析，完善设计功能。目前市面上流行的数据可视化分析平台有：RAWGraphs、ChartBlocks、Tableau等，在开发之前深入使用这些目前比较流行的数据分析工具，熟悉使用流程，通过使用对比自己分析系统的短板，完善设计。
2. 选择高效的开发工具，提升开发效率。开发前端的工具有很多比如HBuilder、WebStorm等，最终选择了Visual Studio Code作为开发工具，Visual Studio Code为轻量级开发工具提供了丰富的插件可以简化开发的流程并且与Chrome浏览器相兼容；两者搭配使用提升了开发的效率。

对比开发技术栈，保持系统性能的稳定性对比分析三大主流的前端框架，深入了解其页面渲染速度和性能损失情况；从组建的支持范围性、编写API的整洁性记忆文档教程的详细程度；经过比较最终选择Vue作为系统的前端框架。V-charts作为国产开发工具，提供了详细的使用教程，与Vue结合使用兼容性强，作为商业级数据图表，兼容当前绝大部分的浏览器。

# 开发技术与工具

工欲善其事，必先利其器。本章内作者将介绍在开发过程中所使用到的所有技术与工具。将技术和工具用到恰当之处，使其发挥成倍的功效

## 开发技术

### v-charts

在使用 ECharts 生成图表时，经常需要做繁琐的数据类型转化、修改复杂的配置项，v-charts 的出现正是为了解决这个痛点。基于 Vue2.0 和 ECharts 封装的 v-charts 图表组件，只需要统一提供一种对前后端都友好的数据格式设置简单的配置项，便可轻松生成常见的图表。

### Vuex

Vuex 是专门为 Vue.js 设计的状态管理库，它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。Vuex 也集成到 Vue 的官方调试工具 devtools extension，提供了诸如零配置的 time-travel 调试、状态快照导入导出等高级调试功能。

* 1. State 单一状态树

状态数据。Vuex所管理的便是状态，其它的如：Actions、Mutations 都是来辅助实现对状态的管理的。Vue组件要有所变化，也是直接受到 State 的驱动而发生变化。

可以通过 this.$store.state 来直接获取状态，也可以利用 Vuex 提供的 mapState 辅助函数将 state 映射到计算属性（computed）中去。

* 1. Getters 可认为是 Store 的计算属性

Getters 本质上是用来对状态进行加工处理。Getters 与 State 的关系，就像 Vue.js 的 computed 与 data 的关系。Getters 的返回值会根据它的依赖被缓存起来，且只有当它的依赖值发生了改变才会被重新计算。

可以通过 this.$store.getters.valueName 对派生出来的状态进行访问。或者直接使用辅助函数mapGetters 将其映射到本地计算属性中去。

* 1. Mutations变化
  2. Mutations的中文意思是“变化”，利用它可以更改状态。事实上，更改 Vuex 的 Store 中的状态的唯一方法就是提交（commit）mutation。不过，mutation 触发状态改变的方式有一点特别，所谓commit 一个 mutation，实际是像触发一个事件，传入一个 mutation 的类型以及携带一些数据（称作payload，载荷）。
  3. Actions

和 mutation 类似。它接收的第一个参数是一个与 Store 实例具有相同方法和属性的 context 对象，因此你可以调用 context.commit 提交一个 mutation，或者通过 context.state 和 context.getters 来获取 state 和 getters。

不过，mutation处理函数中所做的事情是改变state，而 action 处理函数中所做的事情则是commit mutation。

* 1. Module 模块

它实际上是对于 Store 的一种切割。由于 Vuex 使用的是单一状态树，这样整个应用的所有状态都会集中到一个比较大的对象上面，那么，当应用变得非常复杂时，Store 对象就很可能变得相当臃肿！

为了解决以上问题，Vuex 允许我们将 store 分割成一个个的模块（module）。每个模块拥有自己的 state、mutations、actions、getters，甚至是嵌套子模块（从上至下进行同样方式的分割）。

### Vue Router

Vue Router 是 Vue.js (opens new window)官方的路由管理器。它和 Vue.js 的核心深度集成，让构建单页面应用变得易如反掌。包含的功能有：

* 1. 嵌套的路由/视图表
  2. 模块化的、基于组件的路由配置
  3. 路由参数、查询、通配符
  4. 基于 Vue.js 过渡系统的视图过渡效果
  5. 细粒度的导航控制
  6. 带有自动激活的 CSS class 的链接
  7. HTML5 历史模式或 hash 模式，在 IE9 中自动降级
  8. 自定义的滚动条行为

### Vuetify

Vuetify 是一个纯手工精心打造的 Material 风格样式的 Vue UI 组件库。并且Vuetify 是建立在Vue.js之上的十分完备的界面框架。该项目的目标是向开发人员提供他们积累丰富且吸引用户体验所需的工具。与其它框架不同，Vuetify 从一开始就设计为易于学习、易于上手的组件库，并且拥有来自 Material设计规范的数百个精心设计组件。

### Node.js

Node.js 是一个开源与跨平台的 JavaScript 运行时环境。npm 的简单结构有助于 Node.js 生态系统的激增，现在 npm 仓库托管了超过 1,000,000 个可以自由使用的开源库包。

### Axios

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。简而言之是可以发送 get、post 等请求。在 Jquery 比较受欢迎的时候，大家都在用它。但是由于Vue、React等框架的出现，Jquery 不再那么吃香了。也正是因为 Vue、React 等框架的出现，促使了 Axios 轻量级库的出现，因为 Vue 不需要操作Dom，所以不需要再引入 Jquery.js，Axios相继崛起。

## 开发工具

### Visual Studio Code

Visual Studio Code（简称VS Code）是一款由微软（Microsoft）开发并且为跨平台的免费开源代码编辑器。该软件支持语法高亮、代码自动补全（又称IntelliSense）、代码重构、查看定义功能，并且内置了命令行工具和Git版本控制系统。用户可以更改主题和键盘快捷方式实现个性化设置，也可以通过内置的扩展程序商店安装扩展以实现独具自己特色的编译软件。深受广大前端开发者的喜爱。

### GitHub

GitHub是一个代码托管云服务网站，帮助开发者存储和管理其项目源代码，且能够追踪、记录并控制用户对其代码的修改。对于团队开发十分便捷。

### GoogleChrome

是由Google开发的一款设计简单、高效的Web浏览工具。 Google Chrome的特点是简洁、快速。 GoogleChrome 对于前端开发者也是十分友好、方便。

# 需求分析

科技发展的同时，必定会产生大量的操作数据，单日的数据量在不同行业、不同领域都会显示出不同的差距，高达TB级别，低至GB级别，因此，整个系统的稳定性将会面临更大的挑战。与此同时，用户进行数据分析的形式和展现的方式也有所不同。结合上文所述已有产品的现状分析，本章将从系统的总体需求和功能性需求分析等两方面进行展开介绍。

## 系统总体需求

### 系统稳定性分析概述

数据可视化分析平台需要分析海量的数据，用户登录完可以在系统内进行数据源接入、数据预处理等操作；因为平台为B/S架构的在线系统，所以数据存储在Vue Stroe中，数据在后端做分页操作，每次请求数据只有100条，因此可以保证系统的稳定性。

### 系统特点分析概述

该系统有如下几个特点：

1. 交互性：用户可以方便地以交互的方式管理数据。
2. 多维性：数据表中的属性可以按照每一维度的字段进行排列组合显示。
3. 可视性：数据可以使用折线图、饼图、柱状图等基本图型进行显示。

## 功能性需求

### 用户模块

用户第三方登录。通过GitHub授权，获取用户信息，完成登录。用户模块需求分析用例图如图 3.1所示：

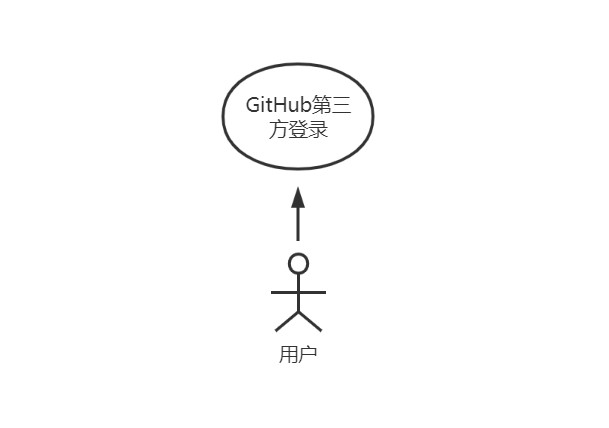


图 3.1 用户模块用例图

### 数据源接入模块

1. 多数据源连接。展示多种数据库类型的图标，对用户选择的某个类型的数据库，进行连接。
2. 记录历史连接。将连接历史记录成列表。
3. 多文件上传。支持上传Excel、Csv两种格式的文件，并且支持多文件上传。
4. 添加数据包。用户自行对上传的表进行分类收纳。
5. 数据包中添加表。共有三种方式添加表：数据库表、上传文件、自定义数据表。

数据源接入模块需求分析用例图如图 3.2所示：

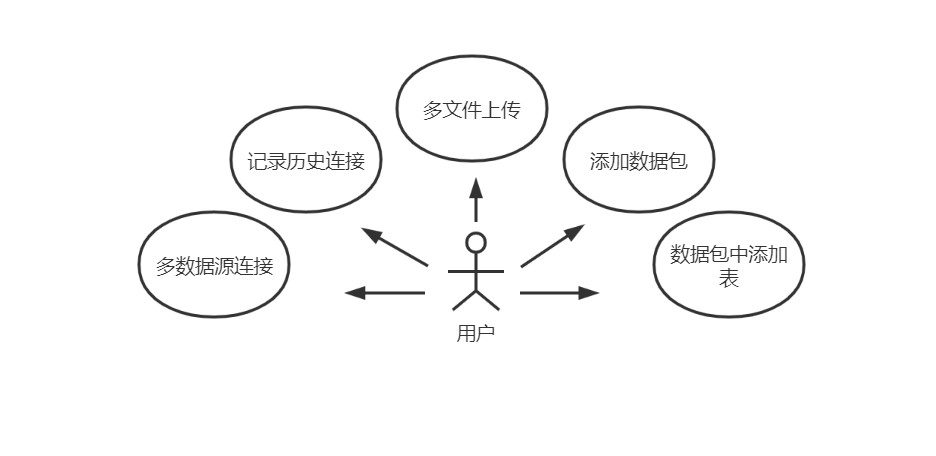


图 3.2数据源接入模块用例图

### 数据预处理模块

1. 数据包中数据源图标展示。数据包选择的数据源以表的形式展现。
2. 选定数据库自定义数据表。用户指定数据表可以进行数据库的拆分，形成新表。
3. 选定数据库自定义字段。用户指定数据表可以在表中定义新的字段，插入表中。
4. 自定义构建计算指标。用户可以根据自己的数据分析需求，根据已有的字段构建新的计算指标，插入在表中。
5. 自定义X/Y轴数据分析。用户分析数据时，可以根据表中的字段，拖选自己想要选择的字段至X、Y轴，分析数据，数据结果以图的形式展现。
6. 字段别名。用户可以对数据源中的数据字段进行字段别名的操作。
7. 分析数据过滤。在数据分析阶段，用户可以对海量数据的分析选择提供的数据过滤功能。

数据预处理模块需求分析用例图如图 3.3所示：

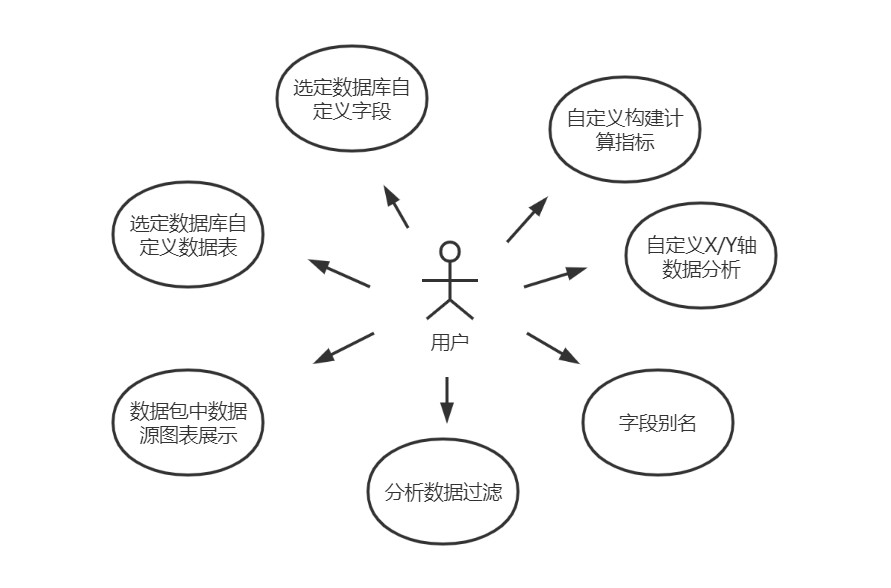


图 3.3 数据预处理模块用例图

### 数据分析结果绘图模块

1. v-charts结合Vuex实现折线图、饼图、柱状图。数据的分析结果以折线图、饼图或柱状图等基本数据分析图的形式展现
2. 数据分析结果导出。对于数据的分析结果图，用户可以选择以图片的形式导出，保存至本地。

数据分析结果绘图模块需求分析用例图如图 3.4所示：

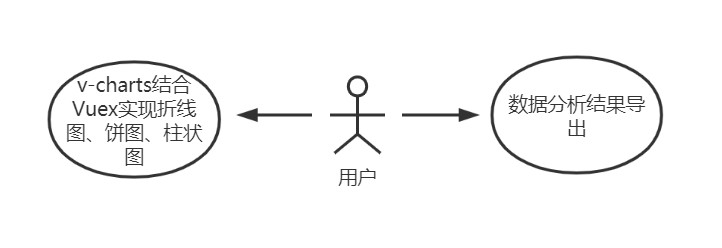


图 3.4 数据分析结果绘图模块用例图

# 系统概要设计

在宏观上把控系统的整体设计叫做系统概要设计，根据上文的系统需求分析，明确系统的整体功能结构之后，本章从宏观角度将整个系统进行全局把控，进行组件封装、自定义Vue指令等操作。

## 系统流程图

数据可视化平台的功能流程图如图 4.1所示：

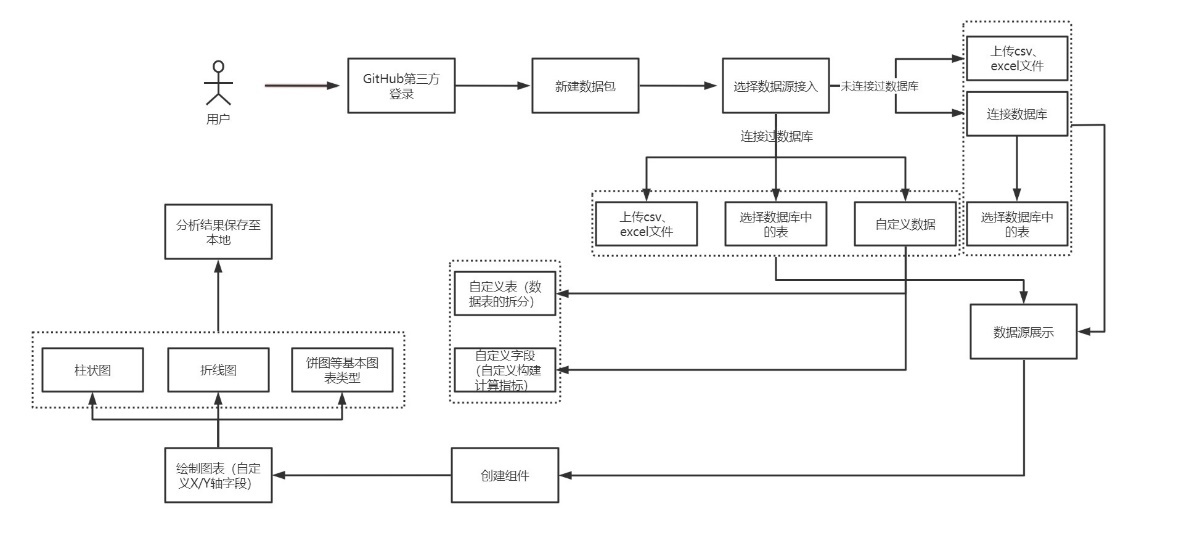


图 4.1系统流程图

## 系统总体功能图设计

完成系统的需求分析，我们将系统总体划分为四个模块：用户模块、数据源接入模块、数据预处理模块和数据分析结果绘图模块；每个模块实现的功能逻辑清晰，系统总体实现的功能如图 4.3所示：

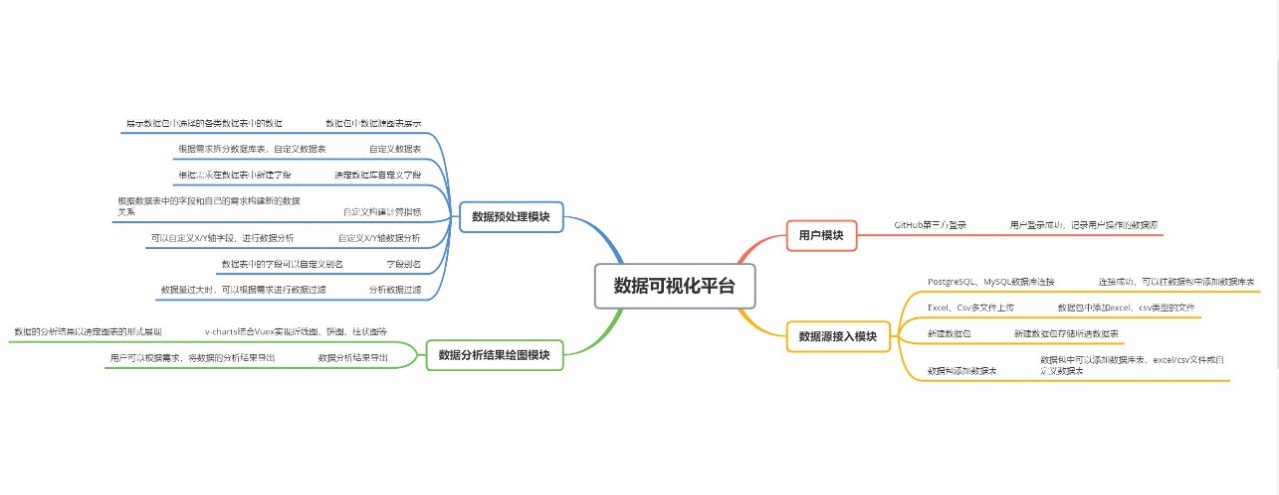


图 4.2 系统总体实现功能图

## 系统请求接口封装设计

与 Vue 配合使用的接口请求工具是 Axios。在项目中，会出现数次前后端数据交互的现象，那么 Axios 的重复使用必定会产生众多重复的代码。因此，将 Axios 的常用方法进行封装是必须要做的。

### 创建一个 Axios 对象

生成全局的 Axios 对象，用来全局配置请求的baseURL、responseType、headers。这样做的优点在于：在将来的任何一次请求中，都会自动拼接baseURL，不仅方便全局统一切换管理URL，而且在请求接口的过程中，也能少些URL中前半部分的重复代码。以此类推，还有请求中的响应类型和请求头的设置，都可实现全局的统一设置、切换和管理，如图 4.3所示：



图 4.3 Axios对象的创建信息图

### 配置请求拦截

所有的网络请求都会先执行这个方法，我们可以在它里面为请求添加一些自定义的内容，也可减少代码的重复使用，方便全局统一管理，如图 4.4所示：



图 4.4 配置强求拦截图

### 配置响应拦截

所有的网络请求返回数据之后都会先执行此方法，此处可以根据服务器返回的状态码做相应的处理，如图 4.5所示：



图 4.5 配置响应拦截图

### 封装 get 方法

利用上文的变量 instance 变量（Axios类型），调用 get 方法，传入两个参数，一个指定接口的URL，一个指定调用的参数，如图 4.6所示：

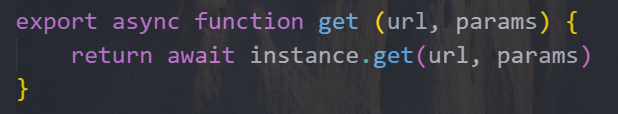


图 4.6 封装 get 方法图

### 封装 post 方法

利用上文的变量 instance 变量（Axios类型），调用 post 方法，传入两个参数，一个指定接口的URL，一个指定调用的参数，如图 4.7所示：

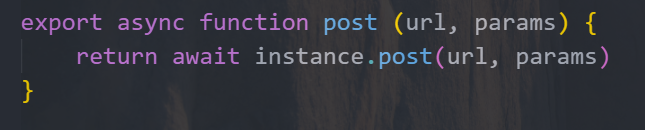


图 4.7 封装 post 方法图

### 封装 put 方法

利用上文的变量 instance 变量（Axios类型），调用 put 方法，传入两个参数，一个指定接口的URL，一个指定调用的参数，如图 4.8所示：

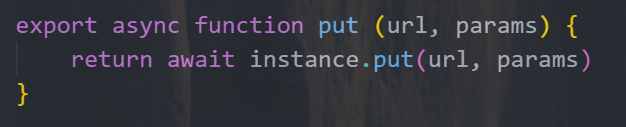


图 4.8 封装 post 方法图

至此，项目中有关前端请求接口的封装就结束了。我们在后续的使用中只需要调用此文件中对应的 get、post、put 方法即可。可以在项目中省略很多重复代码，体现出编程中的封装思想。

## Vue组件封装设计

在前端项目的实现中，会有很多场景使用到相同的组件，例如整个系统中的消息提示弹窗、绘图组件等。如果不进行封装，在指定地方重复使用相同的代码，不仅麻烦，还使得整个 vue 文件代的码结构混乱。因此将常用的、共性的组件进行封装是必不可少的操作。

### 消息提示弹窗

1. 在指定文件夹内，新建一个 vue 文件，文件内自定义好一个消息提示弹窗，如图 4.9所示：



图 4.9 消息提示弹窗图

1. 在 Vuex 中定义一个弹窗数组，用来控制弹窗的显示与否、弹窗的消息内容等数据，如图 4.10所示：



图 4.10 提示框数组定义图

1. 包括使弹窗自动消失的函数（计时器），这个函数需要放在 watch 监听器内，用来监控Vuex 中弹窗数组变量的变化。来控制当前弹窗组件的显示与否，如图 4.11所示：



图 4.11 计时器函数图

### 绘图组件

在此组件中，通过 v-if 来控制显示哪一个类型的图表。作为判断的参数将由调用组件的地方传入，如图 4.12所示：

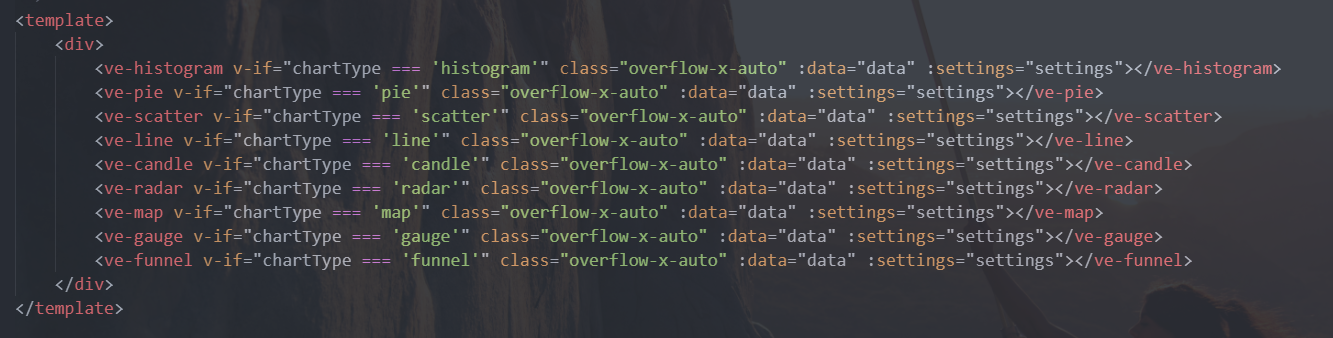


图 4.12 绘图组件图

## 自定义 Vue 指令

### 新建一个 vue 文件，在里面定义一个变量

将来这个变量将会变为一个嵌套对象，即它的某个属性也是一个对象，格式如下图所示，一定要加 inserted 属性。我们可以在这个 orderArr 对象中，追加多个属性，将来这些属性将会变为我们的自定对指定，如图 4.13所示：

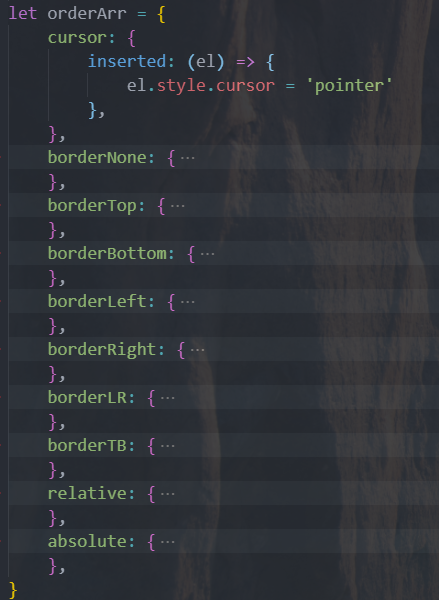


图 4.13 定义 orderArr 变量图

### 遍历上文中的 orderArr 变量

再遍历的过程中，调用 Vue 的 directive 方法，使我们的自定义指令生效即可。在我们后续的开发中，可以使用我们自定义的指定实现出不同的效果，如图 4.14所示：

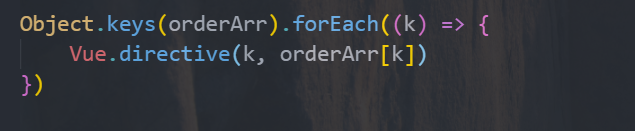


图 4.14 遍历 orderArr 变量图

# 系统详细设计

根据上文中的系统需求分析和系统概要设计，本章将进行详细介绍各功能的实现流程和注意事项。全面细致的进行模块划分、策略选定等。

## 用户模块

### GitHub第三方登录功能设计

在登录界面提供GitHub第三方登录的按钮，点击按钮跳转到GitHub提供的第三方登录界面，同时请求获取用户信息接口，用户登录成功后授权给此平台，接着，跳转回平台首页，接收接口返回的用户id，在页面加载时请求查询用户信息接口，获取用户信息并将头像和昵称显示在系统导航栏上，头像与昵称显示成功即登录功能完成，过程如图 5.1所示：

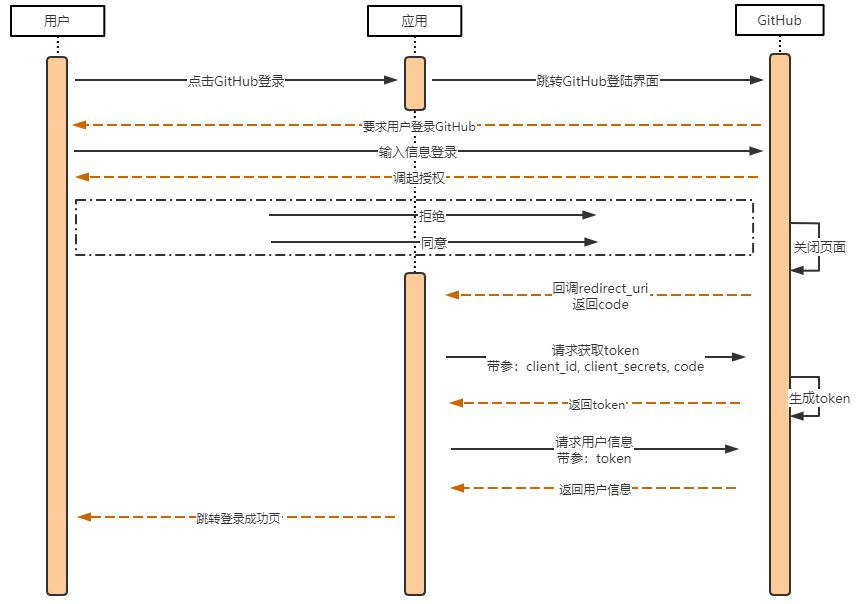


图 .1 GitHub第三方登录时序图

## 数据源接入模块

### PostgreSQL、MySQL数据库连接功能设计

根据用户选择的数据库类型，将类型作为参数，请求后端接口，返回数据状态码为200则连接成功。

如图 5.2所示：

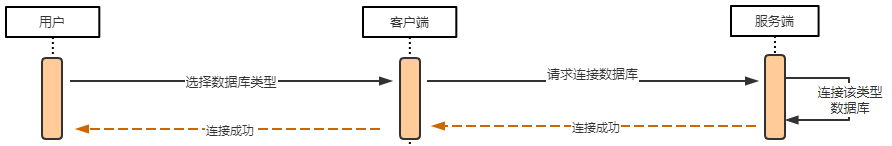


图 5.2 PostgreSQL、MySQL多数据库连接时序图

### Excel、CSV多文件上传功能设计

在数据源接入界面有标识上传文件的按钮，点击按钮用户进入上传文件的页面也可以从数据包添加数据表的入口进入该页面。上传文件的类型有Excel和CSV两种；用户可以为指定的数据包中上传文件，也可以在该页面选择数据包进行上传；当数据包数量为零时，会自动创建一个默认数据包盛放用户上传数据。上传文件前会检查上传数据是否重复，已上传的数据包不允许重复上传。在页面中有数据包中已存在数据表的数量提示，文件上传成功时，提示框显示文件上传成功，没有上传文件就点击上传按钮会提示“请选择上传文件”的字样提示，过程如图 5.3所示：

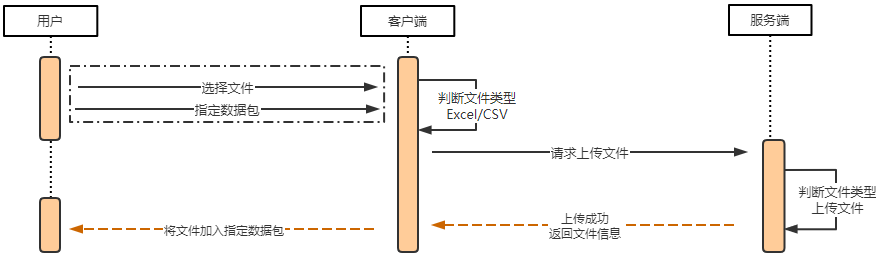


图 5.3 Excel/CSV多文件上传时序图

### 新建数据包功能设计

在数据集界面，呈现一个“添加数据包”按钮，点击按钮弹出填写新建包信息的弹窗，用户填写新建包名后，点击确定，界面上出现一个以用户所填包名命名的数据包图标，过程如图 5.4所示：

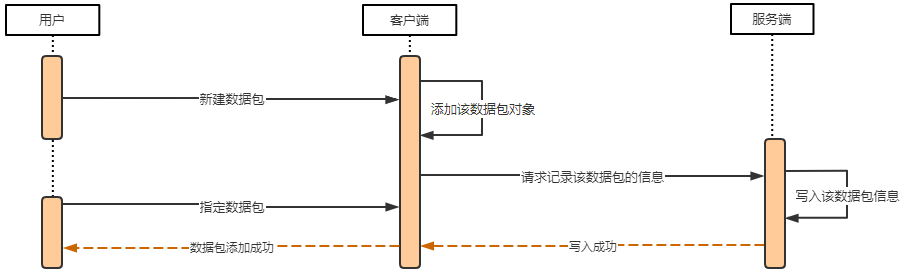


图 5.4新建数据包时序图

### 数据包中添加数据表功能设计

在数据包界面，呈现一个“添加表”按钮，可以从三种不同数据源添加表。当数据包中没有表时，点击“添加表”按钮，出现两个选项卡，分别是“数据库表”、“上传文件”。点击“数据库表”选项卡时，如果用户尚未连接数据库，则跳转至“数据库连接”界面，如果已经连接，则在当前界面内切换界面内容为“选择数据库表”部分，左侧显示历史连接，右侧显示选中连接中所有的表，默认显示第一个连接中所有的表。用户选中表后，点击确定，回到数据包界面，左侧显示用户添加的全部表，右侧预览数据，以表格分页展示表中所有数据。点击“文件上传”选项卡时，则跳转到“文件上传界面”。当数据包中存在表时，点击“添加表”按钮，可出现三个选项卡，“数据库表”、“上传文件”、“自助数据集”。点击“自助数据集”选项卡时，跳转至“自助数据集界面”，过程如图 5.5所示：

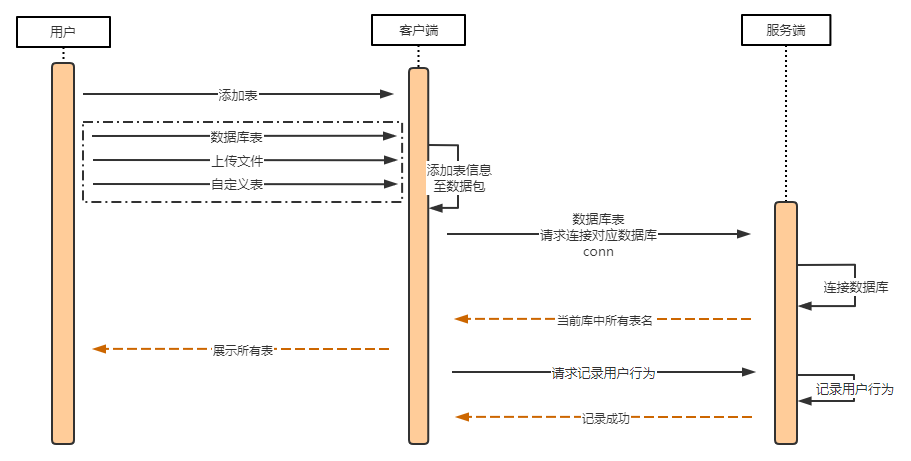


图 5.5添加数据表时序图

## 数据预处理模块

### 数据包中数据源图表展示功能设计

1. 当数据包中存在数据表时，数据展示分为三种情况，数据库表、文件表、自定义表。
2. 数据库表，请求接口获取当前表的所有数据，进行数据展示。
3. 文件表，文件上传界面将文件信息存入vuex中，在当前界面取出对应文件信息进行图表展示。
4. 自定义表，由于是合并/拆分而成，需要存到采用存入Vuex中，在当前界面取出对应信息图表展示。

过程如图 5.6所示：

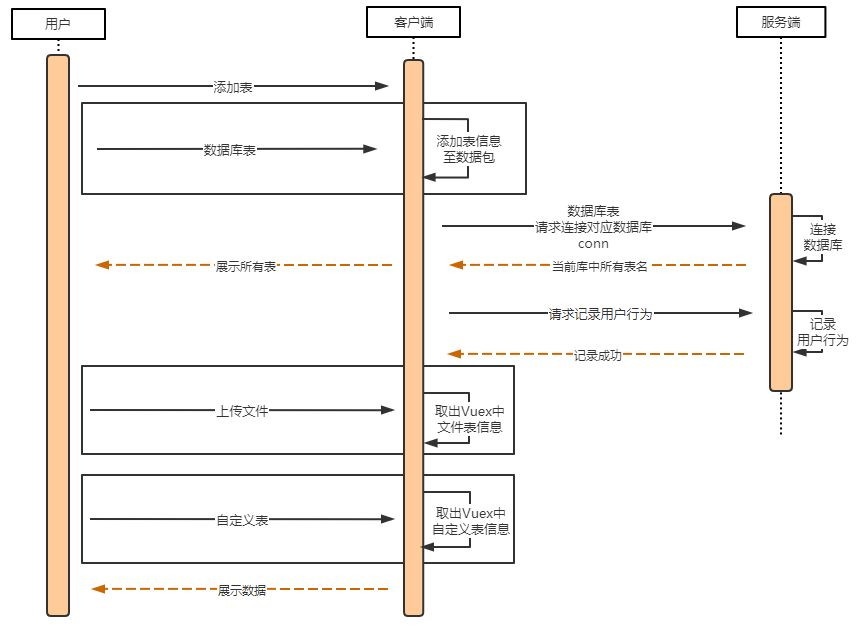


图 5.6数据源图表展示时序图

### 自定义数据表功能设计

用户在数据包中添加完数据表之后就可以对选择的数据库表进行拆分成新的数据库表；用户点击自定义数据，页面跳转到数据自定义页面，页面上方会要求填入自定义数据表的名称，然后点击已经上传的数据库进行字段选择，选择完成可以对选择的数据进行预览，选择完数据表中需要的字段点击页面上方的完成按钮，自定义数据表功能结束，用户可以在数据包中数据源图表展示页面查看到自己新定义的数据表，过程如图 5.7所示。

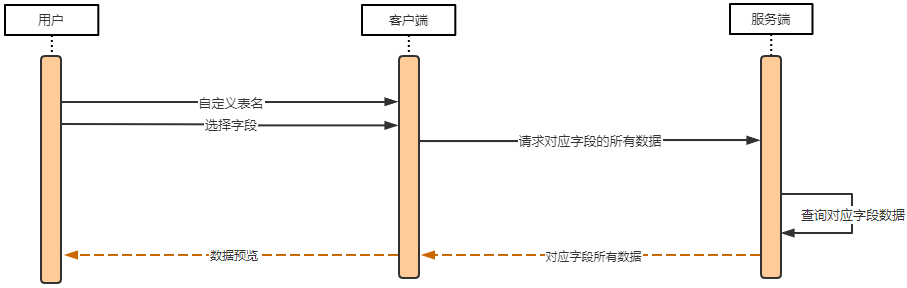


图 5.7自定义数据表时序图

### 选定数据库自定义字段功能设计

在自定义数据页面，对于选定的数据表可以添加新字段，点击页面上的添加字段的按钮，会弹出添加字段操作的弹框。弹框上方的输入框需要输入新定义字段的别名，左侧则是从选定数据库中筛选出来的数值类型的字段，用户可以根据分析需求，选择字段进行基本的算数运算；弹框的右侧是用户选择的字段和运算符号组成的运算式。基本的运算符号显示在算数式的上方。在选择算数式的过程中，会有相对的规则判断如：字段和预算符号必须交叉选择，必须先选择左括号右括号才可以选择等。用户新增字段时也有去重判断，已经添加在数据表中的字段，无法再添加进入数据表中，过程如图 5.8所示：



图 5.8自定义字段时序图

### 自定义X/Y轴数据分析功能设计

在图表绘制页面中，第一步：绘制出两个输入框，用来记录 X 轴和 Y 轴上的数据；第二步：在侧边区域绘制出两块单独的区域，用来存放数值型的数据字段和非数值型的字段，只要是为了控制在横轴上尽量实现对非数值型数据的分析，而在纵轴上则通过数值型的数据来统计分析计算；第三步：实现侧边区域内的各个字段可以进行拖拽效果并且实现 X 轴和 Y 轴的两个输入框中允许存放被拖拽的元素功能；第四步：通过鼠标拖拽侧边区域内各个字段到相应的输入框内，即 X 轴和 Y 轴；第五步：待两个输入框内（X 轴和 Y轴）均存在数据之后，调用后端 API 接口，进行对应字段的数据分析；第六步：将接口的返回值赋值给图表数据变量，进行图表绘制。注意：在此期间，当两个输入框内（X 轴和 Y 轴）的字段发生替换或者减少（删除）时，也可同步调用后端 API 接口，重新进行分析处理操作；在拖拽交互中，元素可以实现在初始位置内消失，在目标区域内显示的功能效果，并且当两个输入框内（X 轴和 Y轴）内的元素减少时，被操作的字段也可恢复到它自己最初的位置显示，过程如图 5.9所示：

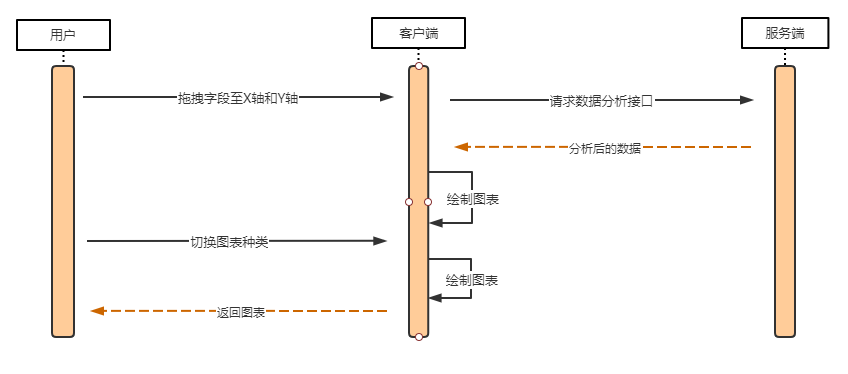


图 5.9自定义X/Y轴数据分析时序图

### 字段别名功能设计

在图表绘制页面中，主要通过对目标字段通过鼠标左键双击事件来实现字段别名功能的实现。第一步：将在侧边区内的各个字段设置为两个状态，其中一个为正常显示的样式效果，另一个即为输入框的样式效果，主要目的在于接收用户对其重命名的参数；第二步：将用户重命名后的字段进行更改，然后在全文内实现统一一致，包括后面请求接口时传递参数的一致性，过程如图 5.10所示：

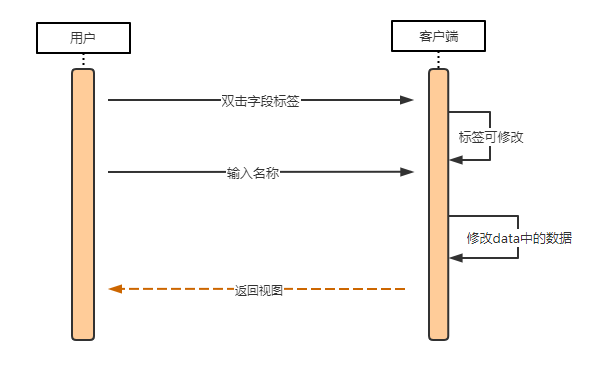


图 5.10字段别名时序图

## 数据分析结果绘图模块

此模块过程如图 5.11所示：

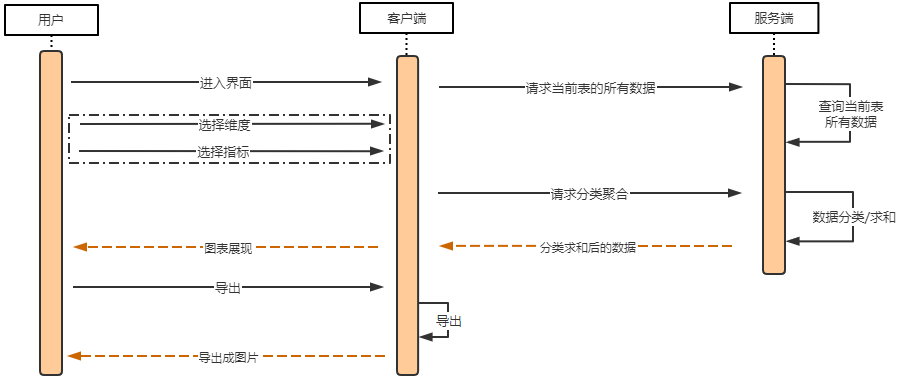


图 5.11数据分析/处理模块时序图

### v-charts结合Vuex实现折线图、饼图、柱状图功能设计

在此绘图功能内，我们主要使用 Vuex 作为一个参数传递的中间件作用。第一步：利用 Vuex 记录前面操作所选择的分析对象基本信息；第二步：在图表绘制页面内，从 Vuex 里取出那个被分析对象的基本信息，那这个对象的基本信息去调用后端的 API，实现数据分析功能；第三步：将接口的返回值拿到赋值给图表数据变量，绘制柱状图（默认）；第四步：在图表绘制页面的中间区域，绘制出一个九宫格的区域，用来存放9大类常用的数学统计图；第五步：在右侧的区域内，选择一个较大区域作为图表展示区域；第六步：点击中间区域内9大类常用的数学统计图按钮，实现图表类型的转换功能。

### 数据分析结果导出功能设计

第一步：新建一个仪表盘界面，用来存放绘制好的各类图表；第二步：在仪表盘界面的顶部区域，实现一个导出按钮，并且在点击之后，弹出 PNG、JPEG等选项卡；第三步：在中间区域内，选定一块较大的区域，用来存放前一个页面传递过来的数据分析图表；第四步： 点击对应的导出类型之后将该页面中的图表导出，默认名为 pic。

# 系统功能的实现

本章作者将详细介绍各功能点的操作流程积极实现效果，按模块进行逐一解释说明。

## 用户模块

### GitHub第三方登录功能实现

登录界面实现了第三方登录功能。首先点击登录按钮时，用Html中的href属性在新窗口打开url，url为GitHub的登录授权uri与回调地址中间以“?”拼接而成，新窗口为GitHub用户登陆界面，同时GitHub根据我们设置好的xxxx，请求后端接口，接口负责获取用户信息。接着，用户在填写GitHub账密并登陆成功后，会询问用户是否授权给此平台，用户点击确认授权后，接口会重定向至此平台首页，同时接口返回该用户的id并拼接在重定向地址尾部，最后，请求用户信息接口，从地址尾部截取id作为参数，请求成功后接口返回用户信息，将用户信息存储至本地缓存，登录功能结束。

## 数据源接入模块

### PostgreSQL、MySQL数据库连接功能实现

目前，项目只支持对 PostgreSQL、MySQL 两类数据的连接操作。用户点击左侧的数据准备功能按钮后，会跳转到数据连接的界面，此时，页面中有两种数据上传的方式，一个是选择连接数据库，上传数据库中的数据；另一个是选择本地的数据文件进行上传。在中间区域还有一个历史连接记录区域，将来会记录下用户的历史连接记录数据信息。若此时，存在一条历史连接记录数据，用户点击之后，会弹出一个对话框，里面将会展示出这个连接的详细信息，包括连接名、主机、端口、初始化数据库、封面图、用户名和密码。接下来着重介绍上传数据库的数据方式。当用户点击“新建连接”按钮后，页面就会发生一个切换，将会在右侧展现出众多数据库的基本信息，用户可以选择适合自己的数据库进行连接（目前只支持 PostgreSQL 和 MySQL 数据库）。以 PostgreSQL 数据库为例，当用户点击 PostgreSQL 数据库之后，当前页面会再次发生一个切换，会出现一个数据库连接的表单，表单内要求填写数据库连接所需的基本参数。当用户将所有的参数填写完成之后，在右上角有三个功能按钮，第一个为“测试连接”按钮，第二个为“保存”按钮，第三个为“关闭”按钮。用户需要先点击“测试连接”按钮，进行连接测试操作，确保数据填写的准确性。若测试连接失败，系统将会弹出一个错误信息提示框，提示“数据库测试连接失败”，用户需要重新核对信息的准确性；若测试连接成功，系统也会弹出一个提示信息框，告知用户测试连接成功，此时用户可以点击“保存”按钮，将此次的连接对象进行保存操作。若用户在没有通过测试连接之前点击了“保存”按钮，系统也会弹出一个错误信息提示框，告知用户在保存之前必须先通过测试连接。如果用户点击了第三个按钮，则页面会发生跳转，切回上一个状态。此时当用户保存成功之后，在页面的最下面会出现两个按钮，一个为“新建连接”，一个为“去选择表（/包）”按钮，此按钮会根据用户的行为操作动态变化。并且此时在左侧的历史连接区域会增加一条信息，即刚刚保存的连接对象。当用户点击“新建连接”按钮后，会重新切换到选择数据库连接的状态，可以继续进行连接操作。若用户点击“去选择表（/包）”按钮后，页面将发生跳转，跳转到数据集页面，进行接下来的操作，效果如图 6.1所示：

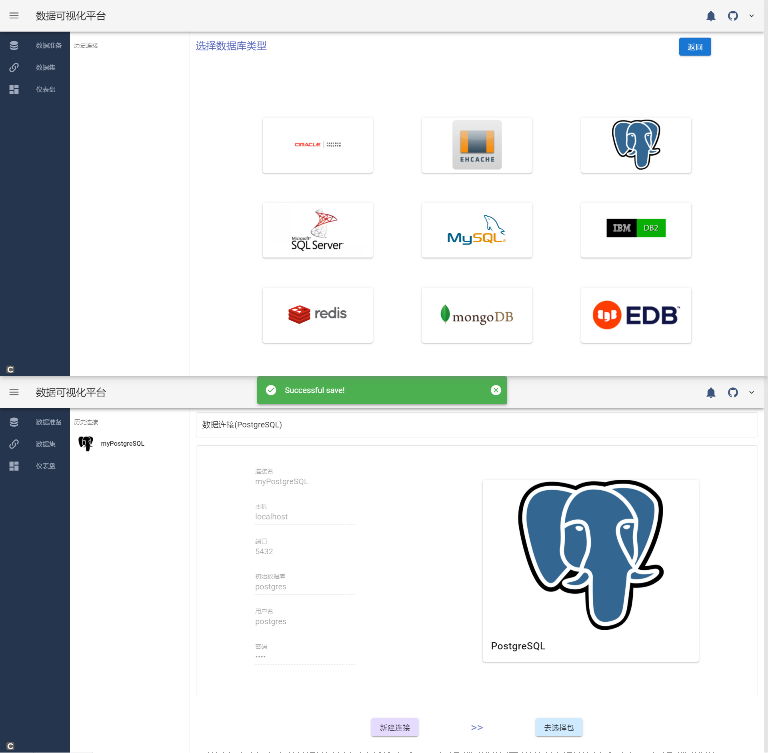


图 6.1数据源接入效果图

### Excel、CSV多文件上传功能实现

进入文件上传页面的入口有两个，一个是从数据连接的界面进入；另一个是从数据包添加数据源页面进入。在文本框内用户可以选择本地的文件进行上传，在文件上传的文本框上预先设置允许上传的文件类型csv和xlsx，其他类型的文件均不允许上传。用户可以选择单个文件上传，也可以选多文件上传，用户上传的文件以formdata类型的数据格式传给后端，由后端进行文件内容的读取。文件上传成功，使用store对formdata进行存储，方便上传的文件信息在其他页面的显示。页面上也会有“上传成功”的提示框显示。在用户点击上传按钮时，前端会对用户已经上传到数据包的数据表进行遍历，防止文件的重复上传，如果选择上传的文件中有重复的数据表，页面上会有“请勿重复上传文件”的显示；在文件上传文本框为空时，点击上传文件的按钮，页面有“请选择需要上文件”的提示。页面上方的下拉框允许用户在该页面选择其他的数据包进行文件的上传，效果如图 6.2所示：

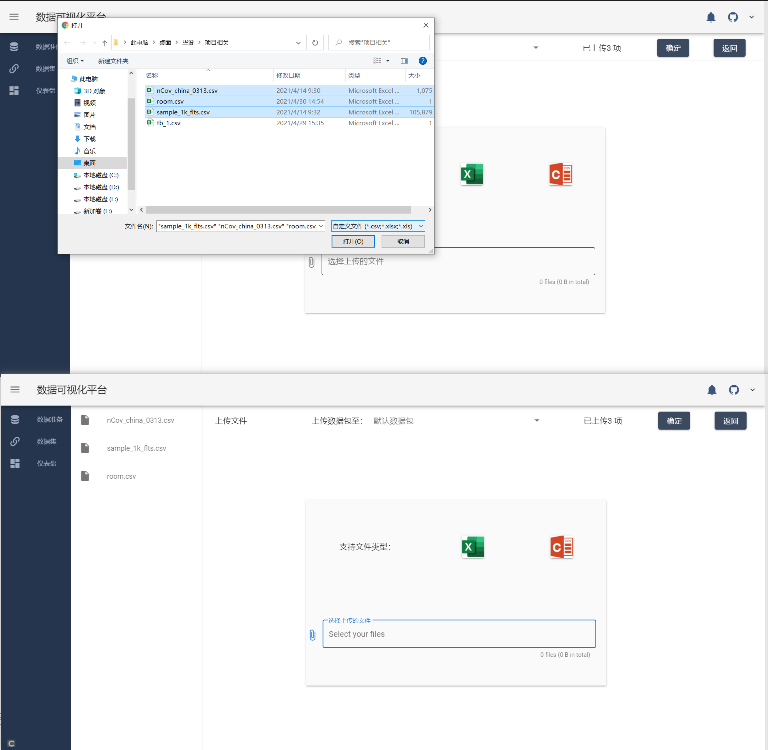


图 .2多文件上传效果图

### 新建数据包功能实现

用户填写数据包名称后，用v-model绑定该变量，作为将要显示在界面上的图标的名字，并将信息记入数据包数组中，效果如图 6.3所示：

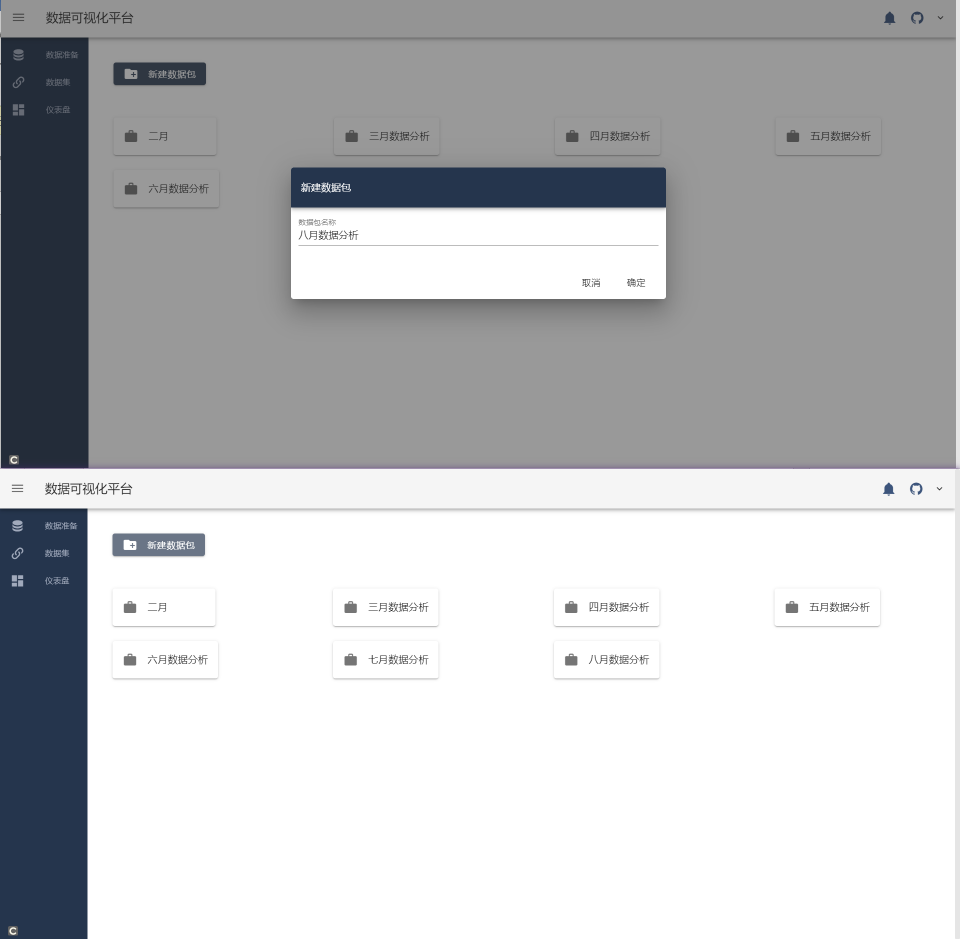


图 6.3新建数据包效果图

### 数据包添加数据表功能实现

使用Vuetify中的v-menu组件，设置三个option分别是：数据库表、上传文件和自定义表。每个option对象设置对应的跳转路由和show参数，当此包的数据表数组为空时，设置show变量为false，使自定义表option不显示，效果如图 6.4所示：

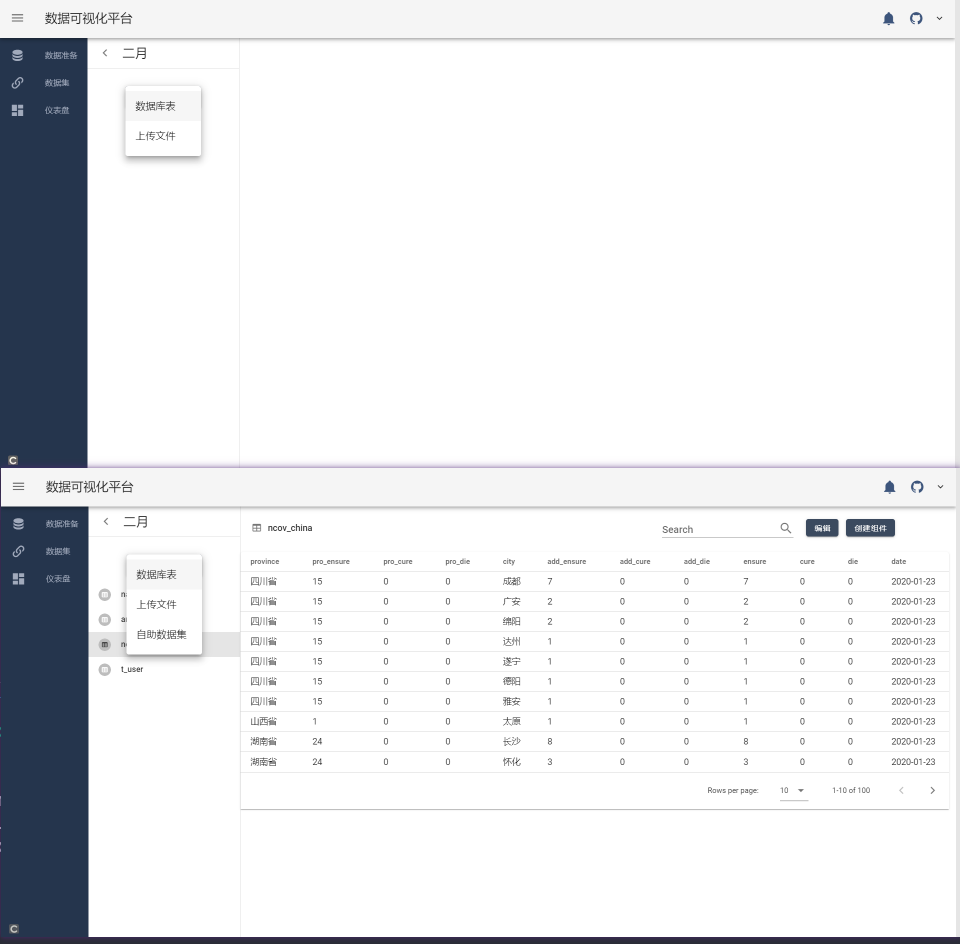


图 .4添加数据表效果实现

## 数据预处理模块

### 数据包中数据源图表展示功能实现

将点击的数据库连接信息作为参数，请求接口获取该库中所有表名，以按钮形式展示，设置动态样式，使点击高亮，再点取消高亮，高亮代表选中，将选中的表信息存入当前数据包数组，包括连接信息和表名，如所示：

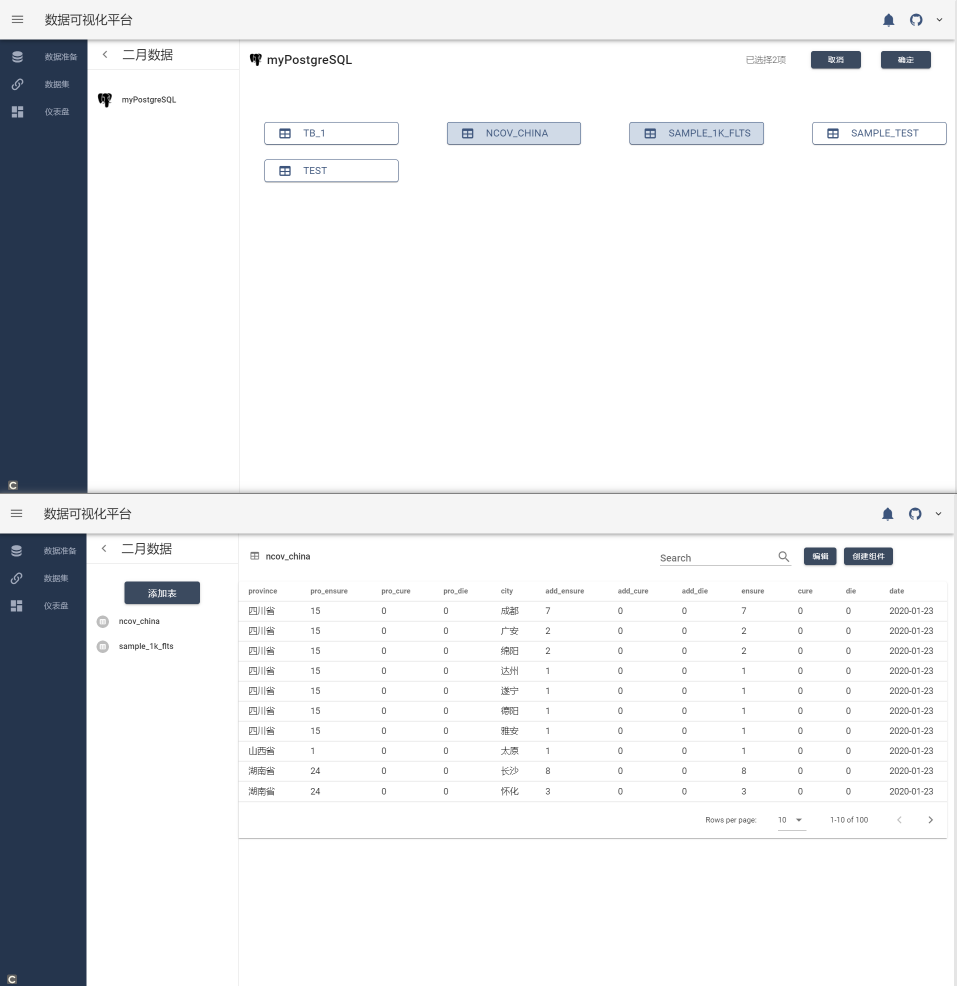


图 6.5数据源展示

### 自定义数据表功能实现

用户选择完成需要的数据表和文件之后可以进行自定义数据表功能，进行数据表的拆分。在数据包中数据源展示界面，点击自定义数据列表，进入自定义数据界面。在页面的上方的文本框中用户必须填入自定义的数据表名。页面左侧会根据store中存储的内容显示该数据包中的所有的数据表。点击左侧的数据表会显示数据表中所有的字段，数据库表的字段是根据所选数据表名到后端请求查到的字段返回给前端显示的，而文件的字段是在读文件时第一行数据内容的展示。用户可以选择数据表中需要数据分析字段组成新的表格。点击所需的字段可以对数据进行预览。字段选择完成即自定义数据表功能完成，自定义的数据库表名称存入store之前也会去重查询，如果有重复的表名会清空自定义数据表的文本框，重新定义表名，效果如图 6.5所示：



图 6.5自定义数据表效果图

### 选定数据库自定义字段功能实现

在分析数据的过程中，用户可以根据自己的分析需求，在已有字段数据的基础上构造新的数据关系，在数据表中定义新的字段。用户点击添加字段按钮，页面会弹出自定义字段的操作界面。界面上方需要输入定义字段的别名，该字段别名不能与数据表中的字段冲突，所以在确认添加字段之前，会对数据表中的字段进行遍历，没有重复字段成功添加，出现重复的字段，文本框内容清空，重新填入字段别名。界面左侧是查询选中数据表中的数值类型的字段方便用户选择，进行数字运算。在选择算式的过程中会有一些操作的判断:第一个选择的必须为字段而不能选择运算符号、字段和运算符必须交叉选择(括号除外)、没有左括号，右括号也不能选择……用户选择失误的操作会有提示框的提示，确保用户选择的算数式是可以在后端运算的。构建完算数式填写完表名即选定数据库自定义字段功能实现完成，如图 6.6所示：

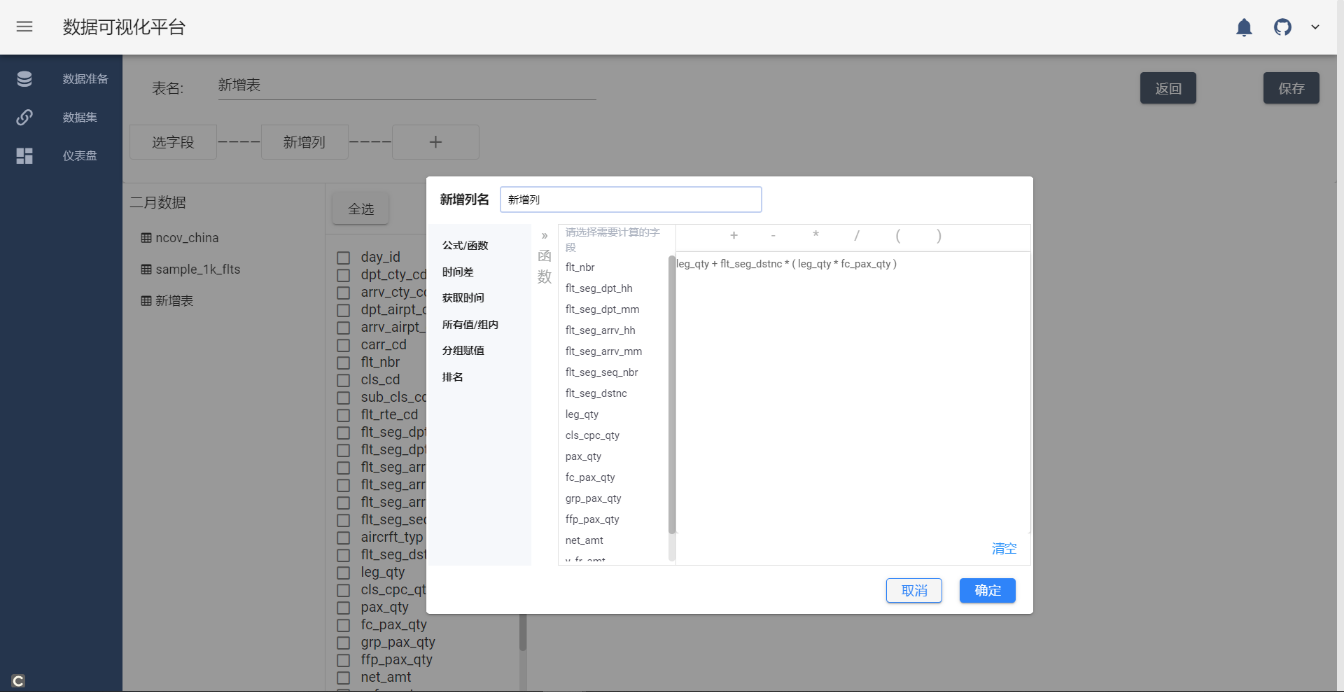


图 6.6选定数据库自定义字段功能效果图

### 自定义X/Y轴数据分析功能实现

在实现好的图表绘制页面中，用户首先需要拖动左侧的各个字段标签到右上侧的两个输入框内（二者选其一即可）。当用户拖动的字段为数值型的数据，用户只能托放置下面的文本框内（Y 轴），此时被拖放的字段标签将从原位置消失，并且出现在下面的文本框内（Y 轴），若用户将此标签拖放至上面的文本框内（X 轴），则系统会发出错误的提示框信息（请将“指标”内的字段放在“Y”轴上），并且该标签将自动复原至原位置。当用户点击两个文本框内的某个标签的删除按钮时，该标签将从所在的文本框内消失，并且恢复到它最初的位置，如图 6.7所示：

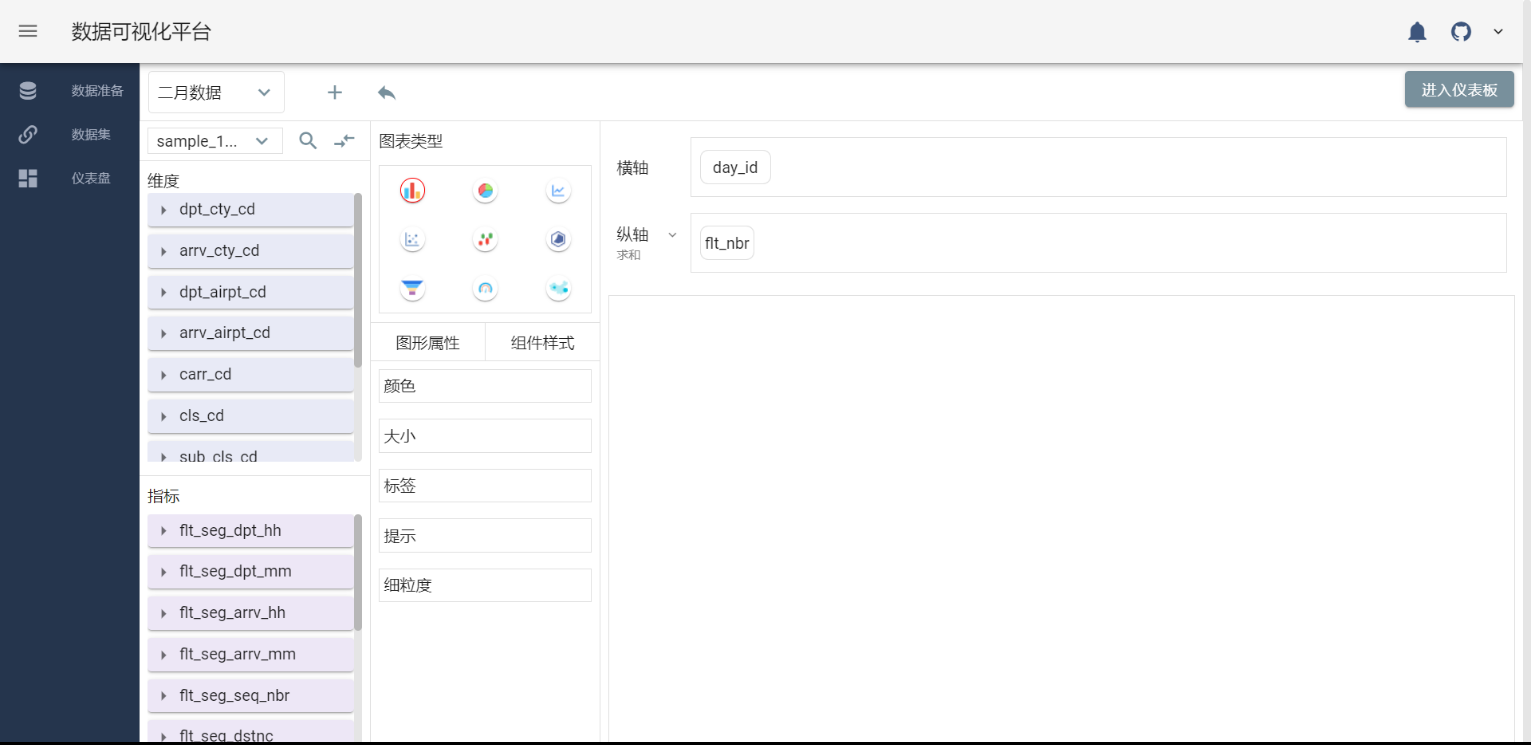


图 6.7自定义X/Y轴数据分析功能效果图

### 字段别名功能实现

用户在未进行拖放字段前，双击（左键）目标字段，可以发现目标字段标签变为文本输入框，此时，用户可以进行字段的重命名操作，当文本框失去焦点或者鼠标点击任意地方时，即可自动保存修改。然后在后面的拖放以及数据分析中都将保持数据的一致性，如图 6.8所示：

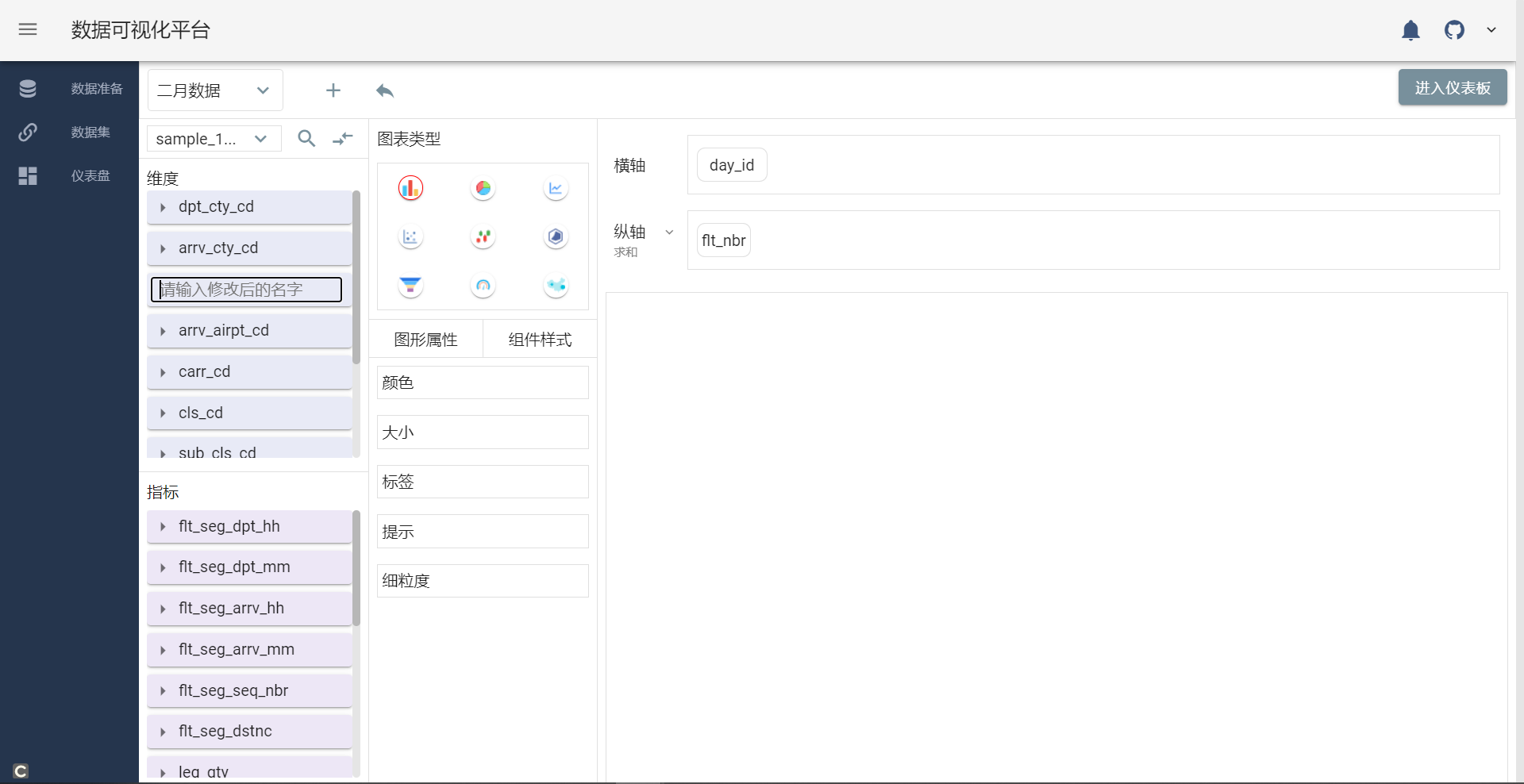


图 6.8字段别名功能效果图

## 数据分析结果绘图模块

### v-charts结合vuex实现折线图、饼图、柱状图功能实现

用户可从两个入口进入到图表绘制界面，一个是直接点击侧边栏中的功能按钮，进入到图表绘制界面，如果此时用户已经创建了文件夹，并且在该文件夹内已经上传好数据文件，那么用户可在中间区域的包名、列名按钮区域选择下拉操作，选择对应包内的指定表进行数据分析绘图操作；另一个是用户从某一个数据集中的某张具体表的详情页面点击“创建组件”按钮的方式跳转过来，此时 Vuex 会记录下用户所选择的具体表的具体信息，进入到图表绘制页面之后，系统会自动从 Vuex 中取出数据，进行数据的查询分析处理。当用户拖拽指定字段到指定文本框后，系统会默认绘制出一张柱状图，此时用户可以点击中间区域的图表类型按钮，选择不同类型的图表切换。用户也可以继续拖拽新的标签到指定文本框内，系统同时会重新处理，当用户删除文本框内的某个字段时，效果也是如此。在纵轴下面，有一个小的下拉框选项，用户点击之后可以选择统计的类型，比如：求和、方差、平均值、最大值、最小值、标准差等数据计算，如图 6.9所示：

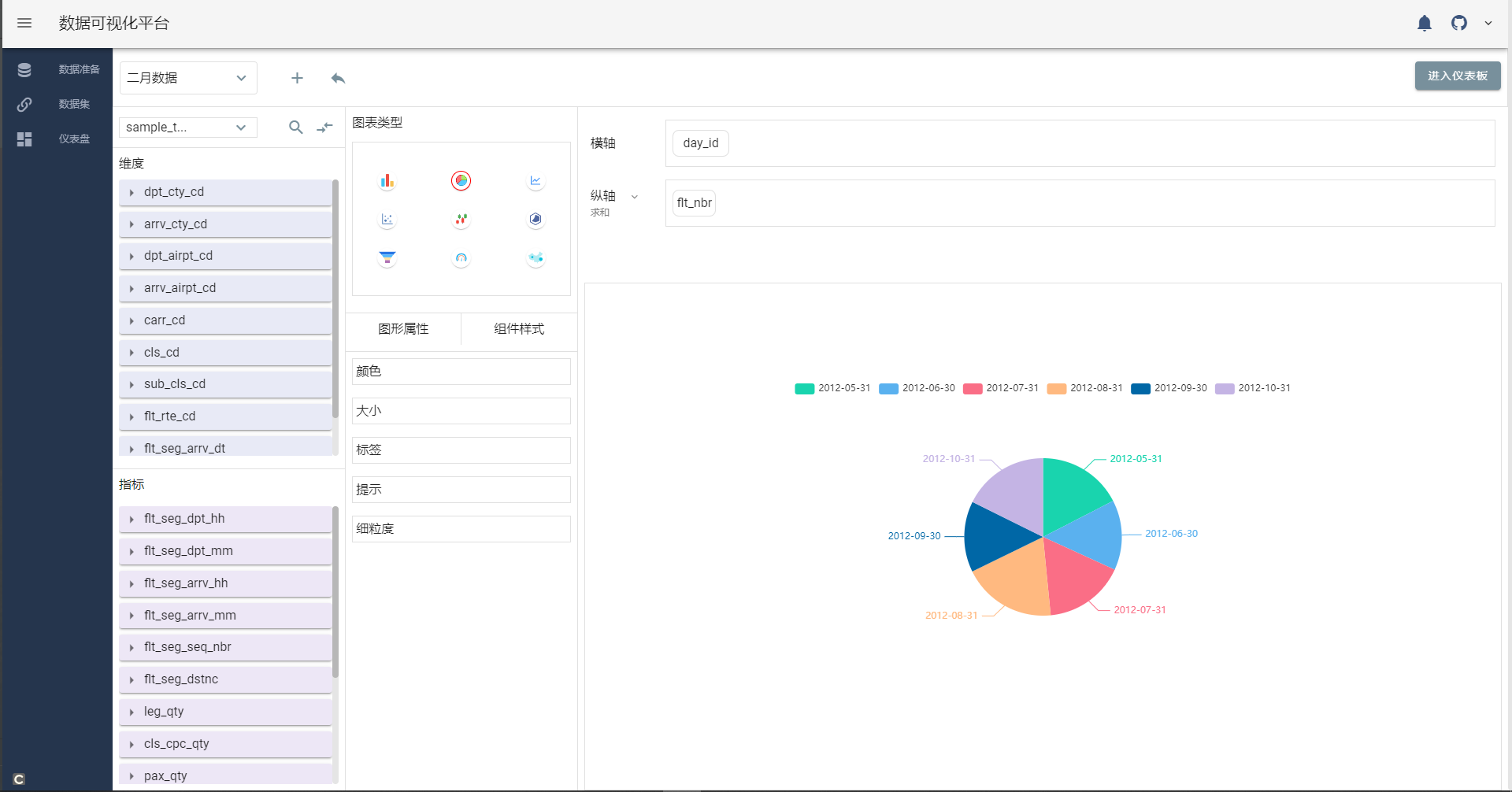


图 6.9 v-charts结合vuex实现折线图、饼图、柱状图功能效果图

### 数据分析结果导出功能实现

用户在图表绘制界面里，经过自己精心的操作并绘制出想要的图表之后，可点击右上角的“进入仪表板”按钮，系统将会通过路由传参，将当前图表的数据对象传递给仪表板界面。当用户进入到仪表板界面之后，系统将取到页面中传递过来的对象参数，解析为一张指定类型的数据图表。用户可以点击顶部的导出按钮，会弹出 PNG、JPEG 等选项，用户选择并点击之后，系统将会弹出一个对话框，用户选择对应的路劲后，便可得到一张名为 pic 的图片，如图 6.10所示：

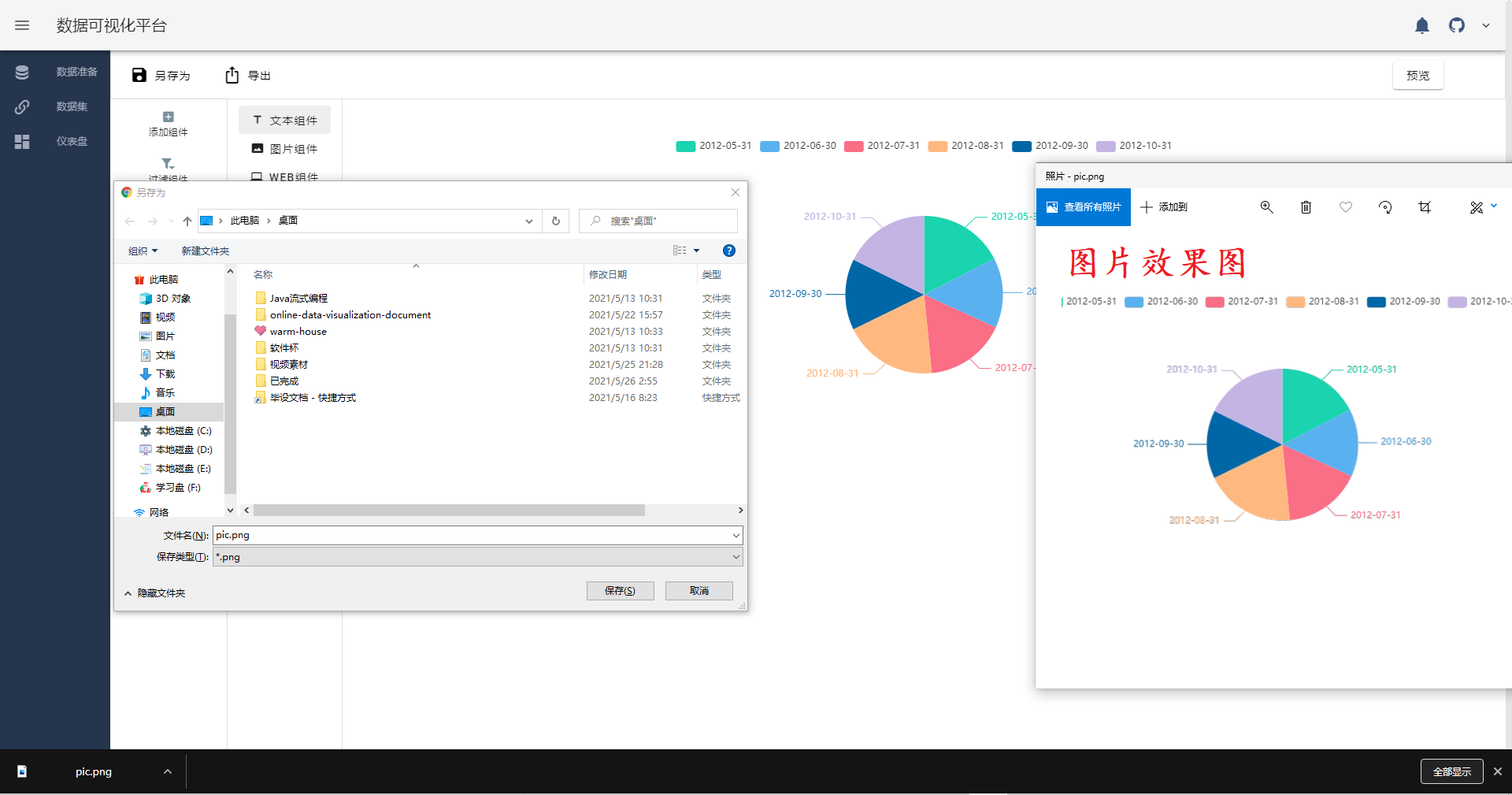


图 6.10数据分析结果导出功能效果图

# 系统测试

在软件开发过程中，软件测试也是必不可少的一环。为给客户极佳的体验，开发完成后进行了各方面的测试操作。本章将介绍测试环境的准备、功能性测试和系统部署测试并将测试结果罗列出来。

## 测试环境准备

为了使测试结果更加准确，检测开发页面的兼容性和页面运行的稳定性，把web界面运行到各个浏览器：其中包括：Goolg Chrome浏览器、IE浏览器、火狐浏览器。在每个浏览器上逐个观察页面效果和功能的完善性，确保开发系统的完备性。

## 功能性测试

### 用户模块

对用户模块进行测试，主要对GitHub第三方用户登录功能进行测试，测试结果如表 7.1所示：

表 7.1 用户模块功能性测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **测试功能** | **测试步骤简述** | **预期结果** | **结果** |
| 1 | GitHub第三方登录 | 点击登录按钮，跳转GitHub登陆界面，输入用户信息，点击授权，跳转进入系统数据集界面。 | 登录成功，跳转进数据集页面 | 功能测试成功 |

### 数据源接入模块

对数据源接入模块进行测试，主要对指定数据库连接功能、文件上传功能、新建数据包等功能进行测试，测试结果如表 7.2所示：

表 7.2数据源接入功能性测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **测试功能** | **测试步骤简述** | **预期结果** | **结果** |
| 1 | PostgreSQL、MySQL数据库连接功能 | 选择连接类型，输入用户名、端口号、密码等信息，点击测试连接，点击保存 | 数据库测试连接成功，并保存到历史连接列表 | 功能测试成功 |

续表 7.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **测试功能** | **测试步骤简述** | **预期结果** | **结果** |
| 2 | Excel、CSV多文件上传功能 | 选择csv、excel类型的文件，上传至所选数据包 | 上传文件保存至所选数据包 | 功能测试成功 |
| 3 | 新建数据包功能 | 输入数据包名称，数据包新建成功 | 数据包新建成功 | 功能测试成功 |
| 4 | 数据包添加数据表功能 | 点击所选数据包，可以查询到连接的所有数据库，选择需要的数据表 | 数据包中可以查询到选择的数据表 | 功能测试成功 |

### 数据预处理模块

对数据预处理模块进行测试，主要对数据源图标展现、自定义数据表、自定义X/Y轴等功能进行测试，测试结果如表 7.3所示：

表 7.3数据预处理模块功能性测试用例图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **测试功能** | **测试步骤简述** | **预期结果** | **结果** |
| 1 | 数据包中数据源图表展现 | 点击数据包中的数据表，可以展现数据表中的所有数据 | 数据展现成功 | 功能测试成功 |
| 2 | 自定义数据表功能 | 在自定义界面定义数据表名，选择数据表中需要的字段，保存成功到数据源页面展现自定义表 | 在数据源展示页面看到自定义的数据表和数据 | 功能测试成功 |
| 3 | 选定数据库自定义字段功能 | 选定数据表，自定义新的字段，根据自定义算数式，构建出新的字段，显示在数据表中 | 数据表中出现此字段 | 功能测试成功 |

续表 7.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | 自定义X/Y轴数据分析功能 | 拖拽所选的字段到指定文本框，可以绘制出所需图表 | 图表绘制成功 | 功能测试成功 |
| 5 | 字段别名功能 | 选择需要修改的字段，修改字段名 | 字段名修改成功 | 功能测试成功 |

### 数据分析结果模块

对数据分析结果模块进行测试，主要对分析结果图表展现、数据分析结果导出等功能进行测试，测试结果如表 7.4所示：

表 7.4数据分析模块功能性测试用例图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **测试功能** | **测试步骤简述** | **预期结果** | **结果** |
| 1 | 分析结果图表展现功能 | 选择数据包中的数据表的字段拖到文本框，根据所选字段绘制图表。 | 根据选择的数据字段绘制图表 | 功能测试成功 |
| 2 | 数据分析结果导出功能 | 点击导出按钮，导出绘制图表 | 绘制图表以图片形式保存在本地 | 功能测试成功 |

## 系统部署测试

整个系统开发完成并且功能测试通过之后，将开发代码打包放到搭建好nginx环境的Centos 8服务器上，部署完成后可以通过指定的域名或IP地址访问页面，并且功能实现完整，则表示系统部署成功，测试完成。

# 总结和展望

善于总结归纳是有始有终里的终。整个Web客户端系统设计并完成之后，作者对整个项目有了自己的总结，以及自己对当前项目的发展和未来有了不一样的认识。本章作者从总结和展望两方面进行展开介绍。

## 总结

“在线数据可视化分析平台”项目共有三个子课题，分别是Web端的设计与实现、API的设计与实现和后台管理系统的设计与实现。本文主要针对Web端展开论述，首先，通过调查课题背景了解到，随着数据量不断攀升，本地的数据分析效率低下，B/S架构的BI工具更加适合大数据的潮流，所以，项目的主要目的是设计实现一个BI平台，采用B/S架构，负责为需要者提供数据深入挖掘的需求，为决策提供数据依据。本文主要完成工作如下：

1. 学习研究了相关技术和数据处理。学习了易于敏捷开发的普遍用于数据处理的Python语言和一个Web框架Flask，学习了Python语言中用于数据处理、分类聚合的Pandas库。
2. 对项目进项了详尽的需求分析。对比竞品，挑选、结合了多家竞品的功能，认真绘制了UI设计图，结合了用例图描述了各个模块的功能需求。基于需求分析，结合了时序图，对每个模块进行了设计与实现。
3. 对系统进行测试，每个功能编写了具体的测试用例，逐个进行测试，并记录了测试结果，验证系统的稳定性。

## 展望

基于B/S架构的在线数据可视化分析平台可扩展和深入的内容还有很多，包括已有功能的优化、用户体验和系统的易维护性设计。

1. 功能虽然初步形成闭环，但每一个步骤中的功能点较少，略显单一。可扩展的部分还有：给X轴的时间维度加入可选分类，如按年聚类，按周聚类等，给Y轴的指标加入分类，如计数、方差等。BI工具在仪表板部分大多可以动态多图表联动，这是一个挑战也是项目未来扩展新功能的方向之一。
2. 用户体验方面，由于平台处理数据量过于庞大，对数据处理的效率达不到预期，需要一定的加载时间，虽然实现了数据可视化的效果，但实际使用中却难以满足对流畅度的需求。后续需进一步研究优化算法和前端渲染效率。
3. 系统管理是每个项目都应该重视的，虽然对项目功能不会产生实质性影响，但是对可维护性来说是至关重要的因素，目前后端具有记录操作日志的功能，还需要更完善的配置，如统一异常管理，自定义异常管理等。

# 致谢

三年的校园时光转眼即逝，大学生活步入了尾声，在这三年中，我学到了许多专业技能，结识了不少志同道合的伙伴，有提升到个人的综合素养，这一切的收获都离不开在这一路上帮助我的人，在这段成长的经历中，他们起着至关重要的作用。

复盘三年我学到了这些知识：编程思想方面有，单例模式、工厂模式、适配器模式等设计模式、面向对象编程和函数式编程的编程泛型，前端方面有JavaScript、CSS、HTML语言、Vue、微信小程序和uni-app，经过不间断的迭代学习，又深入学习到Vuex、Vue-router、Vue-cli，在快速开发时又接触到Vuetify和Element库用来改善UI，后端方面有Java语言、Nginx、Tomcat服务器、MySQL数据库等，开发工具有IDEA、VScode、HBuilder、GitHub等强大且高效的工具。

首先，要感谢我的班主任，也是这次毕业设计的指导老师——许老师。是运气爆棚被许老师授课了一个学期，也是自己积极进取，才成功成为了许老师的学生，让我真正了解到我所学的专业可以写软件，能在电脑浏览器、手机浏览器、微信小程序里等等平台上运行，并自己做自己的程序体验官，让我对自己的专业产生了非常浓厚的兴趣，记得某一天，老师让我们绘制一张名为“我的知识清单的思维导图”，我才恍然大悟，不知不觉，已经接触了这么多主流技术。老师的付出是不可估量的，在每次的项目中，都会先行找到更高效的小技巧，更优的解决问题的方式，通过直播会议和录制详细的讲解视频传授给我们，使我们做起事来事半功倍。

然后，我要感谢小组成员，尚宇驰和张浩杰。和他们在一起做项目的过程中，总是有着充分的良好的沟通，遇到问题耐心解答，当然，做项目总不是一帆风顺的，遇到争议时，我们会各抒己见，经过商讨，总会达成统一，得到大家都满意的结果，遇到难题，卡住解决不了时，永远互相鼓励，无论多晚都会互相帮忙，共同进退。

最后，感谢各位老师评审阅读我的论文，为我提供指导和提出宝贵建议，使我的文章更加完善。

参考文献

1. 陈濛. 基于大数据平台的用户行为分析系统设计与实现\_陈濛[D]. 南京大学, 2020.
2. 王江波. 基于Yaf的火星直播后台管理系统的设计与实现\_王江波[D]. 哈尔滨工业大学, 2018.
3. 李珊. 基于QAR数据分析的航空公司运行品质研究\_李珊[D]. 中国民用航空飞行学院, 2020.
4. 张帅. 智慧城市大数据可视化云平台的设计与实现\_张帅[D]. 沈阳大学, 2021.
5. 彭飞. 面向银行后台管理系统开发与维护的快速开发框架设计与实现\_彭飞[D]. 西南交通大学, 2019.
6. 乔普拉,乔希,摩突罗. 精通Python自然语言处理[M].人民邮电出版社:, 201708.225.
7. 吕晓玲,宋捷. 大数据挖掘与统计机器学习[M].中国人民大学出版社:大数据分析统计应用丛书, 201607.239.
8. 陈新龙. Python核心语法——列表、元组[N]. 电脑报,2021-04-12(029).
9. 你使用的Python对象占用了多少内存？(上)[N]. 电脑报,2019-07-22(034).
10. 宋春颖. 基于Python的数据挖掘与分析[A]. 天津市电子学会、天津市仪器仪表学会.第三十四届中国（天津）2020’IT、网络、信息技术、电子、仪器仪表创新学术会议论文集[C].天津市电子学会、天津市仪器仪表学会:天津市电子学会,2020:3.
11. 叶惠仙,游金水.Python语言在大数据处理中的应用[J].网络安全技术与应用,2021(05):51-54.
12. Isaac Sacolick,Charles. 实现更智能数据可视化的5个步骤[N]. 计算机世界,2019-05-20(003).
13. 刘磊. 数据可视化赋予企业真正的大数据认知能力[N]. 贵阳日报,2016-08-01(005).