

毕业设计(论文)

|  |  |
| --- | --- |
| 课题名称 | 数据可视化分析平台 |
|  | ——Web端的设计与实现 |
| 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 | 1802343139 |
| 姓 名 | 谢晓茜 |
| 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《基于B/S架构的在线数据可视化分析平台——Web端的设计与实现》，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。

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

[第1章 绪论 1](#_Toc72102446)

[1.1 课题背景与研究意义 1](#_Toc72102447)

[1.1.1 课题背景 1](#_Toc72102448)

[1.1.2 研究意义 1](#_Toc72102449)

[1.2 研究目标及内容 1](#_Toc72102450)

[1.2.1 研究目标 1](#_Toc72102451)

[1.2.2 研究内容 1](#_Toc72102452)

[1.3 所做工作 1](#_Toc72102453)

[第2章 开发技术与工具 2](#_Toc72102454)

[2.1 开发技术 2](#_Toc72102455)

[2.1.1 V-charts 2](#_Toc72102456)

[2.1.2 Vuex 2](#_Toc72102457)

[2.1.3 Vue-router 2](#_Toc72102458)

[2.1.4 Vuetify 2](#_Toc72102459)

[2.1.5 Node.js 2](#_Toc72102460)

[2.1.6 Axios 2](#_Toc72102461)

[2.2 开发工具 2](#_Toc72102462)

[2.2.1 Visual Studio Code 2](#_Toc72102463)

[2.2.2 GitHub 2](#_Toc72102464)

[2.2.3 Google浏览器 2](#_Toc72102465)

[第3章 系统需求分析 3](#_Toc72102466)

[3.1 系统总体需求概述 3](#_Toc72102467)

[3.1.1 系统稳定性分析概述 3](#_Toc72102468)

[3.1.2 产品特点分析概述 3](#_Toc72102469)

[3.2 功能性需求分析 3](#_Toc72102470)

[3.2.1 用户模块 3](#_Toc72102471)

[3.2.2 数据源接入模块 3](#_Toc72102472)

[3.2.3 数据预处理模块 3](#_Toc72102473)

[3.2.4 数据分析结果绘图模块 3](#_Toc72102474)

[第4章 系统概要设计 4](#_Toc72102475)

[4.1 系统流程图 4](#_Toc72102476)

[4.2 系统总体功能图设计 4](#_Toc72102477)

[4.3 系统请求接口封装设计 4](#_Toc72102478)

[4.4 Vue组件封装设计 4](#_Toc72102479)

[4.4.1 提示弹窗 4](#_Toc72102480)

[4.4.2 主题按钮 4](#_Toc72102481)

[4.5 通用样式组件封装 4](#_Toc72102482)

[4.5.1 鼠标指令 4](#_Toc72102483)

[4.5.2 边框样式 4](#_Toc72102484)

[第5章 系统详细设计 5](#_Toc72102485)

[5.1 用户模块 5](#_Toc72102486)

[5.1.1 GitHub第三方登录功能设计 5](#_Toc72102487)

[5.2 数据源接入模块 5](#_Toc72102488)

[5.2.1 PostgreSQL、MySQL数据库连接功能设计 5](#_Toc72102489)

[5.2.2 Excel、Csv多文件上传功能设计 5](#_Toc72102490)

[5.2.3 新建数据包功能设计 5](#_Toc72102491)

[5.2.4 数据包添加数据表功能设计 5](#_Toc72102492)

[5.3 数据预处理模块 5](#_Toc72102493)

[5.3.1 数据包中数据源图标展示功能设计 5](#_Toc72102494)

[5.3.2 自定义数据表功能设计 5](#_Toc72102495)

[5.3.3 选定数据库自定义字段功能设计 5](#_Toc72102496)

[5.3.4 自定义构建计算指标功能设计 5](#_Toc72102497)

[5.3.5 自定义X/Y轴数据分析功能设计 5](#_Toc72102498)

[5.3.6 字段别名功能设计 5](#_Toc72102499)

[5.3.7 分析数据过滤功能设计 5](#_Toc72102500)

[5.4 数据分析结果绘图模块 5](#_Toc72102501)

[5.4.1 V-charts结合vuex实现折线图、饼图、柱状图功能设计 5](#_Toc72102502)

[5.4.2 数据分析结果导出功能设计 5](#_Toc72102503)

[第6章 系统功能的实现 6](#_Toc72102504)

[6.1 用户模块 6](#_Toc72102505)

[6.1.1 GitHub第三方登录功能实现 6](#_Toc72102506)

[6.2 数据源接入模块 6](#_Toc72102507)

[6.2.1 PostgreSQL、MySQL数据库连接功能实现 6](#_Toc72102508)

[6.2.2 Excel、Csv多文件上传功能实现 6](#_Toc72102509)

[6.2.3 新建数据包功能实现 6](#_Toc72102510)

[6.2.4 数据包添加数据表功能实现 6](#_Toc72102511)

[6.3 数据预处理模块 6](#_Toc72102512)

[6.3.1 数据包中数据源图标展示功能实现 6](#_Toc72102513)

[6.3.2 自定义数据表功能实现 6](#_Toc72102514)

[6.3.3 选定数据库自定义字段功能实现 6](#_Toc72102515)

[6.3.4 自定义构建计算指标功能实现 6](#_Toc72102516)

[6.3.5 自定义X/Y轴数据分析功能实现 6](#_Toc72102517)

[6.3.6 字段别名功能实现 6](#_Toc72102518)

[6.3.7 分析数据过滤功能实现 6](#_Toc72102519)

[6.4 数据分析结果绘图模块 6](#_Toc72102520)

[6.4.1 V-charts结合vuex实现折线图、饼图、柱状图功能实现 6](#_Toc72102521)

[6.4.2 数据分析结果导出功能实现 6](#_Toc72102522)

[第7章 系统测试 7](#_Toc72102523)

[7.1 测试环境准备 7](#_Toc72102524)

[7.2 功能性测试 7](#_Toc72102525)

[7.3 系统部署测试 7](#_Toc72102526)

[第8章 总结和展望 8](#_Toc72102527)

[8.1 总结 8](#_Toc72102528)

[8.2 展望 8](#_Toc72102529)

[致谢 9](#_Toc72102530)

[参考文献 10](#_Toc72102531)

# 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。

## 课题背景与研究意义

### 课题背景

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；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。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[1]；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。

## 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。
3. 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

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）。

* 1. 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。

# 系统需求分析

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。

## 系统总体需求概述

### 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：

交互性：用户可以方便地以交互的方式管理数据。

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。

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。

## 功能性需求分析

### 用户模块

1. 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。

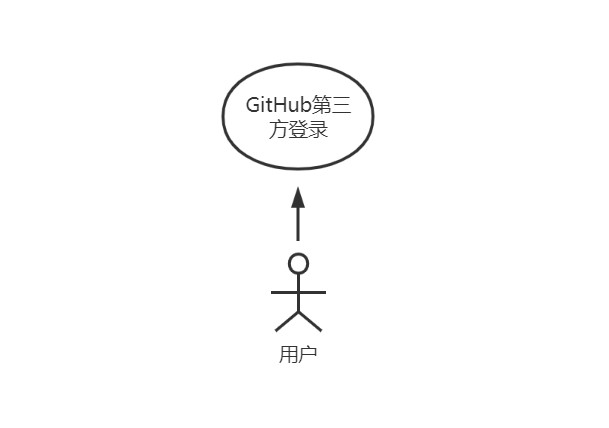


图 3.1 用户模块用例图

### 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。

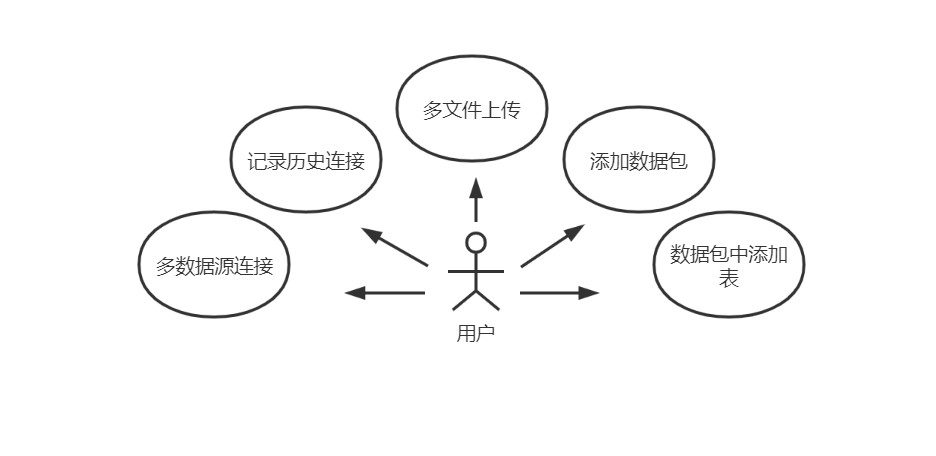


图 3.2数据源接入模块用例图

### 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。

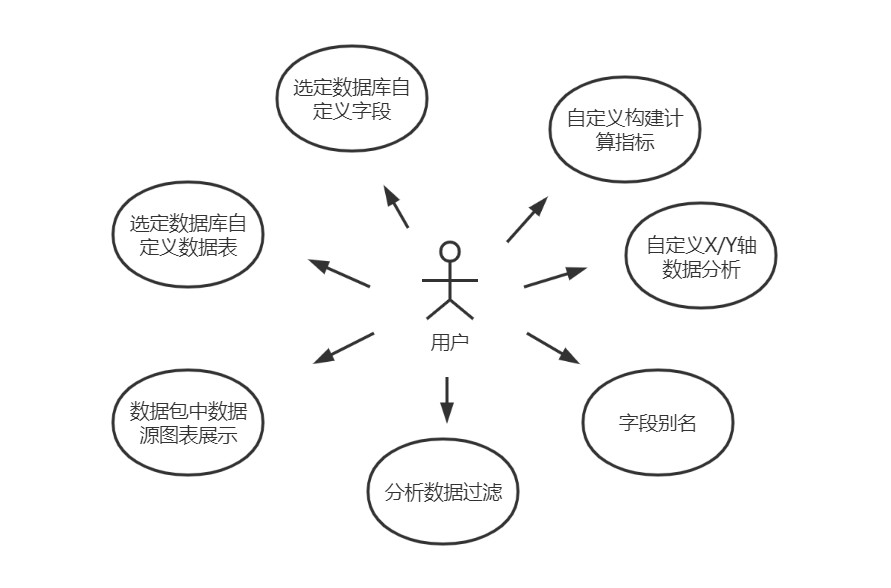


图 3.3 数据预处理模块用例图

### 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。

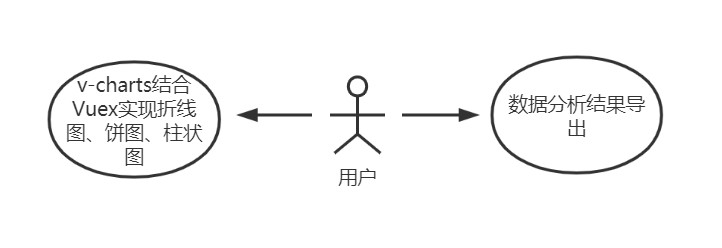


图 3.4 数据分析结果绘图模块用例图

# 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，流程顺序依次为GitHub第三方登录、新建数据包、

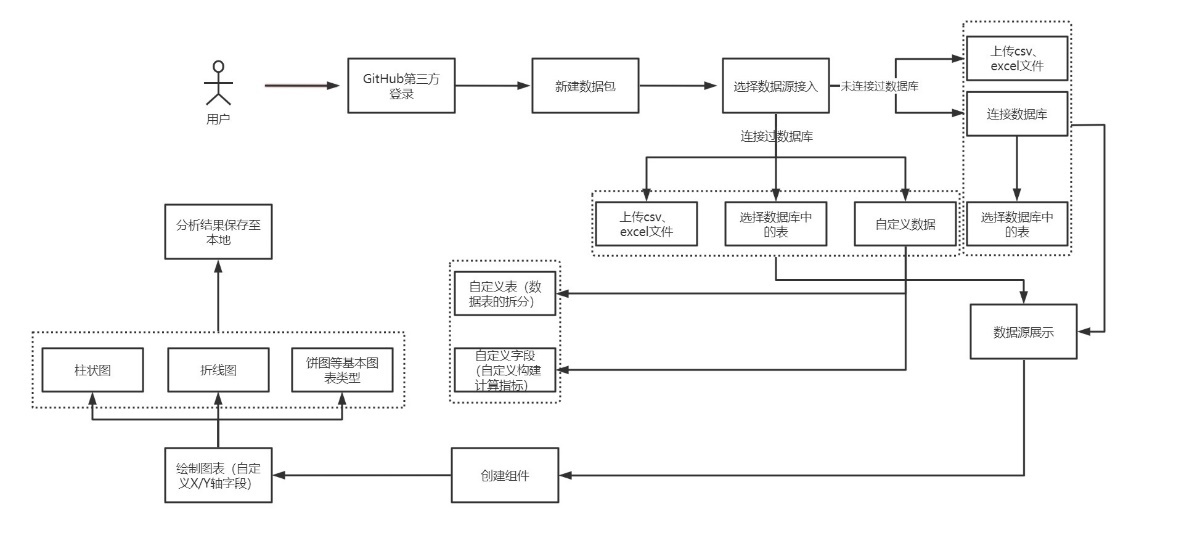


图 4.1系统流程图

## 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4.2所示：

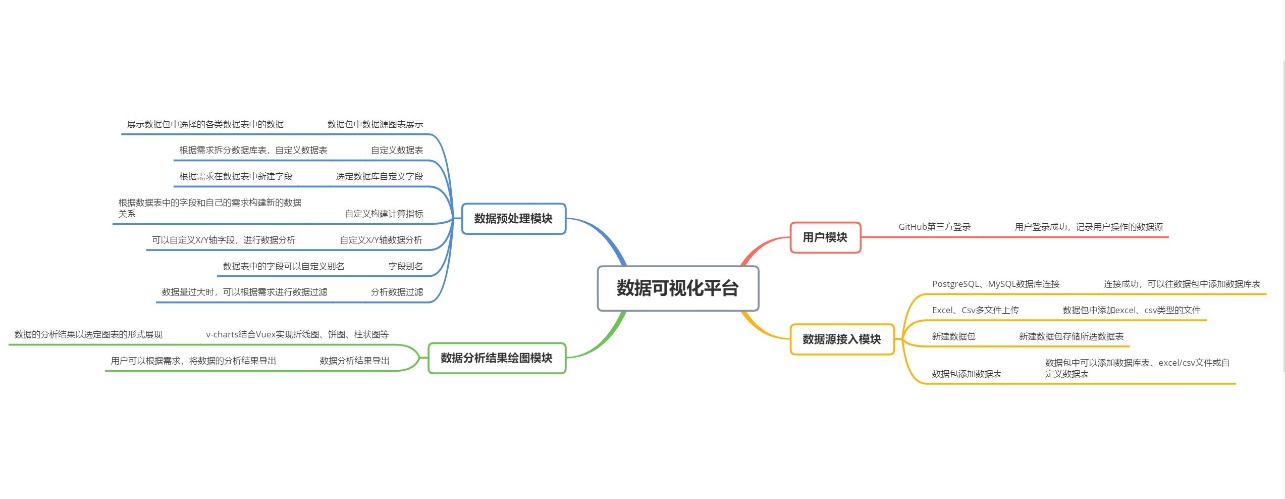


图 4.2 系统总体功能图

## 系统请求接口封装设计

## Vue组件封装设计

### 提示弹窗

### 主题按钮

## 通用样式组件封装

### 鼠标指令

### 边框样式

# 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xx所示。

## 数据源接入模块

### PostgreSQL、MySQL数据库连接功能设计

### Excel、Csv多文件上传功能设计

### 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。

### 数据包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”。

## 数据预处理模块

### 数据包中数据源图表展示功能设计

### 自定义数据表功能设计

### 选定数据库自定义字段功能设计

### 自定义构建计算指标功能设计

### 自定义X/Y轴数据分析功能设计

### 字段别名功能设计

### 分析数据过滤功能设计

## 数据分析结果绘图模块

### v-charts结合Vuex实现折线图、饼图、柱状图功能设计

### 数据分析结果导出功能设计

# 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。效果图如图xx所示。

## 数据源接入模块

### PostgreSQL、MySQL数据库连接功能实现

### Excel、Csv多文件上传功能实现

### 新建数据包功能实现

### 数据包添加数据表功能实现

## 数据预处理模块

### 数据包中数据源图表展示功能实现

### 自定义数据表功能实现

### 选定数据库自定义字段功能实现

### 自定义构建计算指标功能实现

### 自定义X/Y轴数据分析功能实现

### 字段别名功能实现

### 分析数据过滤功能实现

## 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

### 数据分析结果导出功能实现

# 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

## 功能性测试

## 系统部署测试

# 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

## 展望

# 致谢

# 参考文献

**参考文献**

[1] 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.