

**文 献 综 述**

**题 目 企业智能数据管理系统的设计和实现--前端模块**

**英文题目Design and realization of enterprise intelligentplatform management system -- front-end module**

**学生姓名**： **毛慧君 申请学位门类**： **工学**

**学 号**： **201420181014**

**专 业： 软件工程**

**学 院**： **软件学院**

**指导教师： 邓薇 职称： 讲师**

**二 0 一 七 年 五 月 三 十 日**

**摘 要**

本文主要讲述企业智能数据管理系统设计前期所阅读的文献资料，以及所使用的相关技术文献。文中进对企业智能数据管理系统设计所使用的技术进行了描述，详细讲述vue.js、vuex、vue-router的相关知识。

关键词：vue.js；数据管理

**ABSTRACT**

This paper mainly deals with the literature and literature used in the early stage of intelligent data management system design.This paper describes the technology used in the design of enterprise intelligent data management system, and details the relevant knowledge of vue. Js, vuex and vue-router.

**Key words:** vue.js；data management

# 绪论

（1）阅读目的

本次阅读目的主要是在于，在毕业设计前积累一定的专业知识和技能，使得在之后的毕业设计中能够更加顺利和轻松。同时也是是对自学能力的一种很好的锻炼，学以致用具有十分重要的意义。

（2）开发背景

大数据产业在我国已经有了几年的发展，而从整体看，我国信息化程度的确扔不是特别完善，目前还处于探索阶段，若要想进一步发展需要经历几年时间。

随着信息技术飞速发展, 各行各业积累的数据都呈现出爆炸式的快速增长趋势, 我们已不知不觉进入大数据时代. 显然大数据在多个领域都有广阔的应用前景, 已经成为非常重要的战略资源, 存储、管理和分析大数据也已经成为工业界和学术界高度关注的热点。传输、处理、收集、存储、大数据的目的是利用大数据, 而做到有效地利用大数据。

在各种类型的不同数据中，大数据技术即快速获得有价值信息的能力。明白这一点十分重要，使得技术具备走向众多企业的潜力也很大程度上是有这一点决定的。大数据的4个特点有四个层面：第一点，数据体量巨大。从TB级别，一直跃升到PB级别；第二，数据类型众多。例如网络、图片、地理位置、日志、视频信息等等。第三，价值密度低。比如说视频，连续不间断监控过程中，也许有用的数据仅仅有一两秒。第四，处理速度极快。现今，信息时代正渐渐向数据时代转变，数据开始变得愈发重要，它能够反映出公司的经营情况，为公司的经营和决策提供准确的参考依据。而数据平台是连接数据和企业的媒介，可以将这些数据通过一定的规格，清晰地展现给领导。

当前的大数据发展非常复杂。尤其是当用户需要挑选适合自身的大数据技术时，往往是踌躇忐忑的，既要考虑这个技术是不是真正解决了自身的问题，又要考虑该技术是否得到了积极快速的发展，还要忧心是不是还有新的技术加入很多能够提升的新功能。事实上，当不同的技术之间差别不是很大的时候，用户是可以随便选某一种技术以所用。但是纵观当今大数据产业的发展，在未来，很大机会是可能出现更加集中化、一体化的技术。

（3）开发工具介绍

Visual Studio Code是一个轻量级但功能强大的源代码编辑器，它可以在您的桌面上运行，可以用于Windows、macOS和Linux。它附带了对JavaScript、打字稿和节点的内置支持。js和其他语言（如C++、cjava、Python、PHP、Go）和运行时（例如。NET和Unity）的扩展生态系统非常丰富。

# 2. 引言

## 2.1 需求分析

所谓"需求分析"，是指对要解决的问题进行详细的分析，弄清楚问题的要求，包括需要输入什么数据，要得到什么结果，最后应输出什么。可以说，“需求分析”就是确定要计算机“做什么”。 需求分析是一项重要的工作，也是最困难的工作。

作为一个企业智能数据管理系统需要切实帮助企业分析管理大数据，通过对数据的管理对过往不足进行反省也对未来形势进行预测，从而使得企业朝着更好地方向蓬勃发展。

## 2.2 目标

实现一个企业智能数据管理系统的目标，一方面在于掌握庞大的数据信息。另一方面则在于大数据与产业的结合，对海量数据进行专业化处理，实现数据的“增值”和应用的“价值”。

（1）系统界面友好，操作简单，由用户控制界面而不是界面控制用户。

（2）业务管理智能化。

（3）采用流行的浏览器/服务器方式，响应速度快，安全性高，扩充方便。

（4）系统设计尽可能考虑到在实际管理中会遇到的情况。

（5）系统维护方便可靠，具有灵活性、通用性及安全性的特点。

为了能达到以上目标，需要系统地了解IT行业的相关知识，以及开发时所使用的技术。

# 3. 企业智能数据管理系统

数据化管理是科学管理的基础。科学管理的目标是目标明确、决策准确、措施有效、执行有力。数据化管理是将业务工作中的基本状况，通过翔实的数据直观地展现，并通过适当地分析，明确经营基本状况，发现业务工作中的不足之处，为管理者提供准确的决策依据，促进管理层进行有针对性地改进和有效地决策，是科学管理的基础。数据化管理是科学领导的参考。领导学认为领导的艺术与方法是达成领导效能与发展的关键因素。数据化管理是优秀的管理方法之一。

完善的数据化管理能够明确指出下属业务工作中存在的各类问题，以实事求是的方法并辅之于其他的管理手段，能够有效地指导若干下属开展工作，能够根据问题的严重性与重要性进行有针对性地改善，促进团队的整体进步，从而实现领导效能，是科学领导的有效参考。数据化管理是企业管理改进的关键。优秀的企业管理应该具备完善的运营数据分析体系。一切企业活动，最终都以数据最为参考， 达成一定的数据指标，循环改进，持续发展。数据化管理存在于企业的每个环节，通过参考经营数据管理的企业体制是确保企业良性发展的关键。数据化管理是一种全新的管理方法，其推广和运用可以促进民族企业的发展，增强国际竞争力。

各类数据在企业生产经营中起着至关重要的作用，数据是企业，生产，经营，战略，等等，几乎所有的经营活动所依赖的，不可或缺的信息。数据就犹如企业经营者的眼睛一样，通过数据可以反映出经营的问题，就犹如舵手依赖导航一样。数据类型可以分为：财务数据，生产数据，销售数据，市场数据，人力资源数据，等等，各种各样的数据，起到的作用也是不一样的。例如可以反映销售状况，通过不同的时间，市场环境，好坏，趋势，等等变量，反映经营状况，生产状况，企业经营者要根据数据做判断，来指导销售，生产，以及库存，制定生产计划等等。例如：去年焦炭企业连续亏损，企业就要根据市场数据做生产调整，来压缩产能，换取市场价格回升。等等。财务数据，生产数据，等等的作用都是必须的，并且都是至关重要的，数据的缺失，或者统计不出来。

# 4. vue.js

Vue.js是一套渐进式框架，用以构建用户界面。主要采用自底向上增量开发的设计。Vue的核心库只关注视图层，它不仅容易学习，还便于与第三方库或已有项目。它最大的特点就是双向数据绑定。

主要是先实现一个数据监听器Observer，监听所有数据的对象，如果数据有变化即可拿到新值并通知订阅者，再者实现一个指令解析器Compile，来扫描和解析每个元素的指令，根据指令模板替换数据，以及绑定相应的更新函数。而实现一个Watcher，是连接Observer和Compile的桥梁，能够收到并订阅每个属性变化的通知，执行指令绑定的相应回调函数，从而更新视图。最后mvvm入口函数，整合以上三者。

# 5. vuex

Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式。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](https://github.com/vuejs/vue-devtools" \t "_blank)，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。

虽然 Vuex 可以帮助我们管理共享状态，但也附带了更多的概念和框架。这需要对短期和长期效益进行权衡。如果您不打算开发大型单页应用，使用 Vuex 可能是繁琐冗余的。确实是如此——如果您的应用够简单，您最好不要使用 Vuex。一个简单的 global event bus 就足够您所需了。但是，如果您需要构建一个中大型单页应用，您很可能会考虑如何更好地在组件外部管理状态，Vuex 将会成为自然而然的选择。

每一个 Vuex 应用的核心就是 store（仓库）。“store”基本上就是一个容器，它包含着你的应用中大部分的状态 (state)。Vuex 和单纯的全局对象有以下两点不同：

Vuex 的状态存储是响应式的。当 Vue 组件从 store 中读取状态的时候，若 store 中的状态发生变化，那么相应的组件也会相应地得到高效更新。

你不能直接改变 store 中的状态。改变 store 中的状态的唯一途径就是显式地提交 (commit) mutation。这样使得我们可以方便地跟踪每一个状态的变化，从而让我们能够实现一些工具帮助我们更好地了解我们的应用。

我们通过提交 mutation 的方式，而非直接改变 store.state.count，是因为我们想要更明确地追踪到状态的变化。这个简单的约定能够让你的意图更加明显，这样你在阅读代码的时候能更容易地解读应用内部的状态改变。此外，这样也让我们有机会去实现一些能记录每次状态改变，保存状态快照的调试工具。有了它，我们甚至可以实现如时间穿梭般的调试体验。

由于 store 中的状态是响应式的，在组件中调用 store 中的状态简单到仅需要在计算属性中返回即可。触发变化也仅仅是在组件的 methods 中提交 mutation。

**参考文献**

[1] 陆凌牛.HTML5与CSS3权威指南[M].北京.机械工业出版社,2015.

[2] Eric T Freeman，Elisabeth Robson.Head First HTML5 Programming[M].北京.中国电力出版社,2012.

[3] 施密特,斯普森,李强.HTML5经典实例[M].北京.中国电力出版社,2013.

[4] 弗兰纳根.JavaScript权威指南[M].北京.机械工业出版社,2012.

[5] 张容铭.JavaScript设计模式[M].北京.人民邮电出版社,2015.

[6] 罗彻麦尔.深入理解JavaScript[M].北京.人民邮电出版社,2016.

[7] 埃克尔.java编程思想[M].北京.机械工业出版社,2013.

[8] 辛运帏.java程序设计[M].北京.清华大学出版社,2013.

[9] 付森,石亮.MySQL开发与实践[M].北京.人民邮电出版社,2012.

[10] Lyle.Head First Ajax[M].北京.中国电力出版社,2010.

[11] 李浪,朱雅莉,熊江.软件工程[M].武汉.华中科技大学出版社,2013.

[12] 耿祥义,张跃平.Java面向对象程序设计[M].北京.清华大学出版社,2010.

[13] 刘新.Java开发技术大全[M].北京.清华大学出版社,2009.

[14] 张银鹤,梁文新.JavaScript完全学习手册[M].北京.清华大学出版社,2008.