爱德华在线简历制作系统的设计与实现

学　　院：商学院

专　　业：信息管理与信息系统

姓　　名：苏敬雄　　学　　号：150802103440

指导老师：王淞春　　职　　称：高级工程师

中国·珠海

二〇一九五月

诚信承诺书

本人郑重承诺：本人承诺呈交的毕业设计《爱德华在线简历制作系统的设计与实现》是在指导教师的指导下，独立开展研究取得的成果，文中引用他人的观点和材料，均在文后按顺序列出其参考文献，设计使用的数据真实可靠。

本人签名：_____

日期：_____年____月_____日

爱德华在线简历制作系统的设计与实现

摘　要

求职简历对于每一位应聘者来说是十分重要的一块敲门砖，是个人给企业的直观的第一印象，简历能否打动 HR 的心就直接决定了求职应聘者能否得到企业的下一轮笔试面试应聘机会，因此一个好的简历制作平台是至关重要的。

该在线简历制作系统主要功能就是为了提供给用户一个在线能够编辑简历的功能，系统还包含 4 个子系统的功能，分别是用户管理子系统、组件管理子系统、简历管理子系统和统计报表子系统；系统的功能都是紧围绕着这四个子系统进行展开。系统分开前后平台使用，前台供用户使用创建简历，后台主要供管理员对该在线简历制作系统管理维护使用。该简历制作系统操作简单、方便易用、用户体验良好、界面简洁、功能实用、是个人制作简历的一个神兵利器。

关键词：web 在线；B/S 模式；简历制作；求职

Design and Implementation of Edward Online Resume Production System

Abstract

The resume is a very important step for every candidate. It is an intuitive first impression of the individual. The ability of the resume to impress HR directly determines whether the job applicant can get the next round of the company. A written interview is an opportunity, so a good resume production platform is crucial.

The main function of the online resume production system is to provide users with the ability to edit resumes online. The system also includes four subsystem functions, namely user management subsystem, component management subsystem, resume management subsystem and statistical report subsystem. The functions of the system are all around the four subsystems. The system is used before and after the platform is separated. The front desk is used by the user to create a resume, and the background is mainly used by the administrator to manage and maintain the online resume production system. The resume production system is simple in operation, quick to get started, good in human-computer interaction, convenient and easy to use, simple in interface and practical in function, and is a weapon for personal production of resumes.

Key words: Web inline; B/S; Resume Writers and Editors; Job hunting

目　录

第 1 章 绪论

1.1 系统开发的背景和目标

1.1.1 系统开发的背景

无论是初入职场的应届毕业生还是已经在职场上滚爬摸打多年的职场老手，简历都是面试招聘当中不可缺少的重要信息文件，如何简单快速方便的制作一个精美的能让 HR 一眼就心动的简历呢，甚至有没有这样一个专门提供给求职人员制作专属于自己的个人简历的平台呢？目前个人了解到的主要制作简历的途径主要还是以 word 软件为载体，模板为基础而向上加工修改制作的。

随着科技的发展，网络可以说已经是遍布全世界的每一个角落了，因此，我们可以利用当今如此发达的网路技术来构建这样一个在线的制作简历的系统，这样即使是换了别的电脑设备或者在外只要有浏览器能连接上网路，这样就能够随时随地的查看、编辑自己的简历，甚至能够提供各种格式的简历文件下载，方便各位求职人员对自己的简历的管理。当然不仅仅是简历，待业务扩大以后我们还可以对该系统升级，提供不仅仅是制作简历的功能，甚至能够提供在线制作名片，海报等功能。因此该系统的潜在开发价值是值得肯定的。

1.1.2 系统开发的目标

本系统主要功能是为需要制作简历的都可以算是该系统的主要服务人员。提供一个 web 端在线创建、修改、导出简历的功能。用户在该系统的前台当中可以使用系统预设定的一些简历模板快速的创建一个关于自己的个人专属简历出来，或者使用系统预设定以及该网络其他用户提供的编辑小组件以拼积木搭建房子那样的方式构造出自己个性化的简历出来，甚至可以在该系统中构建并且发布自己的个性化组件，在网上发布自己的组件后就能够被该系统的其他用户所使用在他人的简历当中。

平台的管理员则主要使用管理配套的该系统的后台，对用户账号进行管理、平台管理、查看统计报表数据等，对该在线简历制作系统进行一个运维的操作。

1.2 系统的主要功能和特点

本系统的根本功能是提供一个 Web 在线平台给用户构建属于自己的一份简历，这是本在线简历制作系统的最重要也是整个系统的核心功能，因此做好该功能也就是完成了该系统的最重要的部分。

系统主要包含 4 个子系统，分别是用户管理子系统、组件管理子系统、简历管理子系统和统计报表子系统，这四个子系统涵盖了该在线简历制作系统的基本功能，在系统分析、设计和编码实现当中要紧紧围绕着这四个子系统进行展开。

系统主要有 4 个特点第一个特点是操作简单，上手快，人机交互好，用户使用起来方便，快捷；接着是系统的 UI 界面简洁、风格统一，用户使用时候能够体验一种一致的视觉感受；第三个特点是系统维护成本低，管理方便，系统安全可靠以及系统开发周期短，需求能够快速的进行更新迭代。

1.3 设计开发的方法和技术栈的选择

1.3.1 设计开发的方法

目前系统分析设计的方法主要有两大类型，分别是面向对象的软件设计方法和面向过程的软件设计方法。我在该在线简历制作系统的分析设计当中使用的方法为面向过程的软件设计方法，这是一种以过程为中心的编程思想，考虑的是实际的实现，一般的面向过程是从上往下步步求精，所有面向过程最重要的是模块化思想方法，即是自顶向下，逐层分解，高内聚低耦合。

目前系统分析设计的方法主要有两大类型，分别是面向对象的软件设计方法和面向过程的软件设计方法。我在该在线简历制作系统的分析设计当中使用的方法为面向过程的软件设计方法，这是一种以过程为中心的编程思想，考虑的是实际的实现，一般的面向过程是从上往下步步求精，所有面向过程最重要的是模块化思想方法，即是自顶向下，逐层分解，高内聚低耦合。

1.3.2 开发技术栈的选择

在开发该在线简历制作系统当中，我打算使用的主要开发技术语言是 JavaScript，主要是能够统一开发技术语言，使得学习的成本降低。前端页面编写选用 Vue.js 作为

开发框架提高开发的速度和开发的质量；选用 iView 作为界面设计的 UI 框架，统一样式风格，同时使用 less 和 sass 作为样式预处理方便前端页面的样式。服务端则使用的是 Node.js 技术加以 koa2 作为服务端框架，加快开发的速度[6]；数据库则选用了 MongoDB 该非关系型文档型 NoSQL 数据库。

在开发当中为了能够方便开发流程，因此使用了 WebStorm 作为开发的 IDE，postman 作为服务端接口的调试工具，cmder 作为命令行工具，Chrome 浏览器来查看网页效果以及作为调试器调试前端页面代码。画图方面则主要是使用 Visio、Grafio、XMind-Zen 这三个工具。

1.4 论文的内容和结构安排

论文共由 6 个章节组成，主要内容及结构安排如下六个章节。

第 1 章节是绪论，该部分主要是介绍该在线简历制作系统的背景来源、研究目标和主要研究内容，并对系统相关的现状、工具、技术栈选择等做简要分析，明确系统方向。

第 2 章节是系统的规划，对整个系统的设计方案做总体的介绍以及该系统的主要功能子系统的简要规划，把握系统的核心功能和基本需求。

第 3 章节是系统分析，对系统进行整体和局部的规划之后就是系统的设计，主要包括业务逻辑和数据流两方面。

第 4 章节是系统设计，对业务流程和数据流进行设计之后便是设计系统的数据存储，数据结构，数据库表、系统的界面设计以及用户输入和输出、操作等做一个系统的详细的分析和设计规划，为后面的系统实现打下一个坚实的基础。

第 5 章节是系统实现，该部分主要是在之前的系统规划、设计的基础上面真正的去实现这样的一个在线简历制作的系统，主要包括编码和测试两个部分。

第 6 章节是总结部分内容包括对该系统的总结和展望，以及该系统从 0 开始到最后实现的过程的总结设计收获和体会。同时指出整个系统从规划到实现的历程上面的不足之处和需要改进的地方。

第 2 章 系统规划

2.1 初步需求分析

该在线简历制作系统最重要的功能应该就是提供一个web在线的平台给用户制作自己的个人简历，这也是该系统的核心功能。以及围绕着提供制作简历的核心功能需求将该系统主要划分为了 4 个子系统，分别是用户管理子系统、组件管理子系统、简历管理子系统和统计报表子系统，这四个子系统涵盖了该在线简历制作系统的基本功能，在后面的系统详细分析、设计和编码实现当中会紧紧围绕着这四个子系统进行展开。

系统的模块划分主要分为了前台和后台两个主要模块，前台模块主要是提供给用户制作个人简历使用，后台模块只开放给管理员管理系统使用。模块细分之下可以分成账户信息相关模块、简历相关模块、元素组件模块、管理员相关模块、报表模块五个大模块，如图 2-1 的系统总体模块划分图。

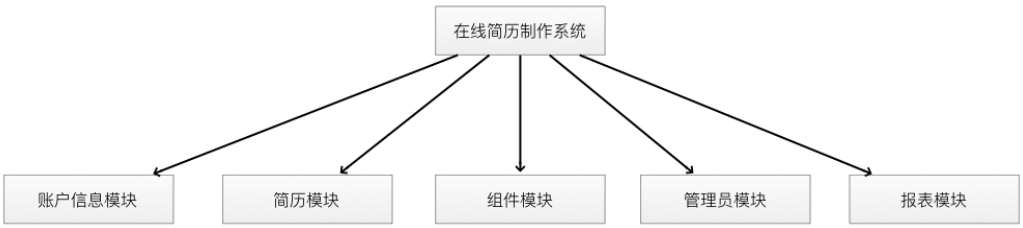

图 2-1　系统主要模块划分

系统风格相关的设计，这里主要选用的是黑色、灰色、白色、蓝色作为三种主题色，如图 2-2 所示的系统主题色。风格简洁，主要目的为了给用户一种简约、舒适，整个系统的风格一致，没有明显冲突的感觉。

图 2-2　系统主要主题色

操作方面，只需要用户简单操作键盘输入和鼠标拖拽、点击等方式，尽可能做到不需要用户不需要学习成本就能够熟练运用该系统的操作，尽可能做到更好的人机交互的体验，也是给用户一个良好的使用系统体验[9]。

2.2 总体结构

经过进一步的系统分析和功能细分，在线简历制作系统的主要功能分为用户前台和管理员后台两个主要的系统，其下分别有各自不同的功能和子系统。

首先是前台用户当中主要包含三个模块子系统，第一个模块是用户账户信息相关模块，这里包括了用户注册登录退出、修改信息等功能；第二个模块为简历相关模块，这个模块包含了创建、编辑、删除、导出简历功能；第三个模块为元素组件模块里面主要是创建、上传个人组件，以及收藏他人上传网络的组件这些功能。如图 2-3 为前台用户主要的操作指示图。

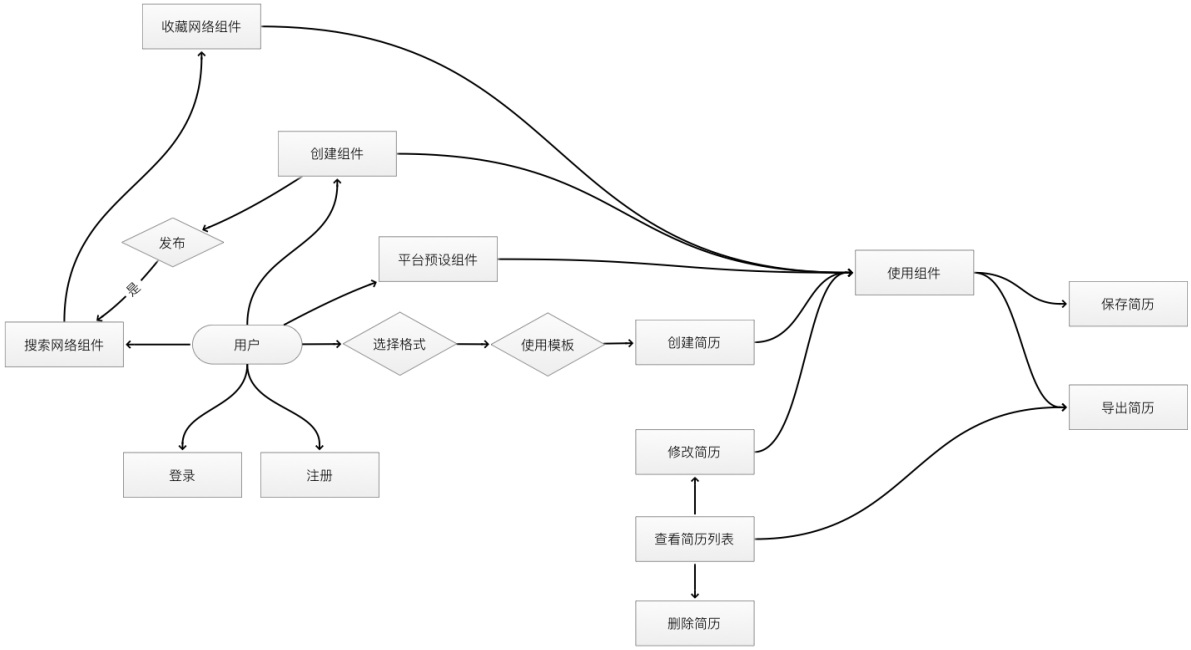

图 2-3　系统前台用户原型图

接着是后台管理员的部分，后台管理员主要被划分成五个主要的功能模块。第一个模块为账户管理模块，该模块主要是管理平台的账户，包括管理员和用户账户的增加修改等。第二个模块为权限管理模块，主要是针对管理员账户的后台权限进行控制的模块，系统采用的是 RBAC 控制权限的方法，因此能够对角色，权限进行增加修改删除等操作；第三个模块是简历模块是管理各种简历格式和对应格式的模板的功能，关系到前台用户在创建建立时候提供选择的简历规格大小和简历是用模板等操作。第四个是比较重要的组件模块，这里是组件配置，设置系统组件，对组件进行管理的模块。最后一个是报表模块能够查看用户在使用该系统的数据报表功能模块。如图 2-4 为管理员在管理平台的主要功能模块原型图。

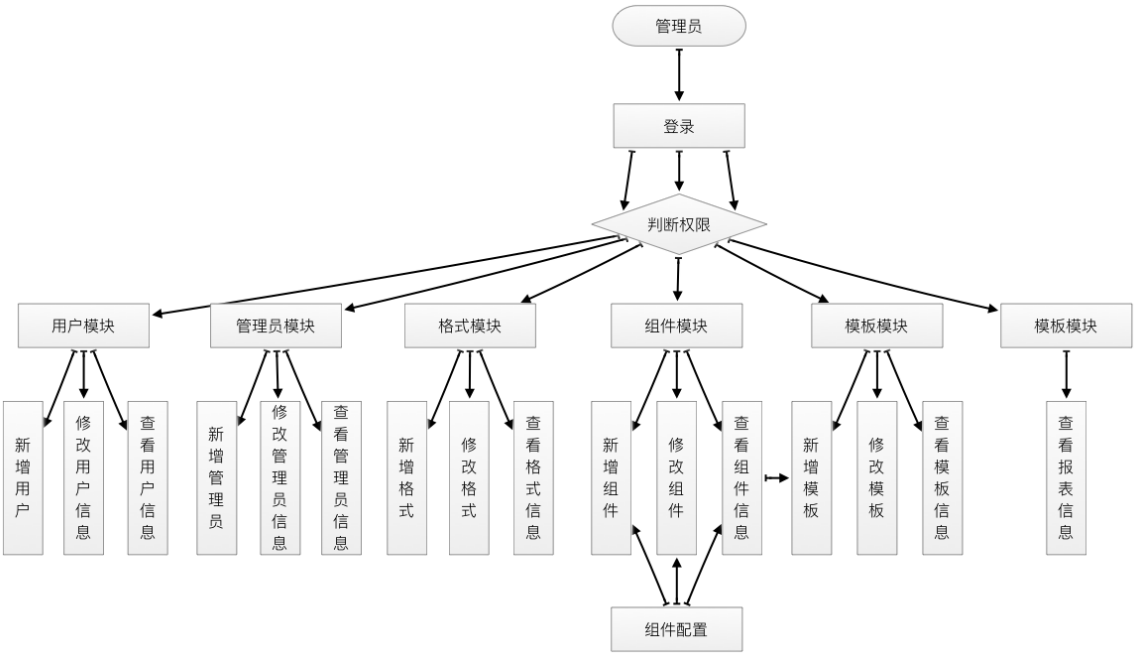

图 2-4　系统后台管理员原型图

系统的功能模块结构规划完成后，则来考虑系统的 UI 界面原型，系统的 UI 界面使用的是常规的网页管理系统的布局，页面布局主要可以分成三个模块，分别是顶部的导航栏，以及左侧的操作导航栏，最后中间延伸到右下角的区域为内容操作区或者内容显示区域。

2.3 可行性研究

从技术可行性研究方面上看，因为使用的为 B/S 架构，因此只需要用户在安装浏览器，并且设备能够连接上互联网就能够使用该 web 在线简历制作系统，但是因为交互与页面的展示可能在移动端上面的适配就会稍微比桌面端差一点。针对该系统，我决定采用前端使用当前极度流行的前端框架 Vue.js 来开发前端页面的代码，服务端使用的技术语言是 TypeScript 和 Node.js,存储的数据库选用的是 MongoDB 这非关系型数据库，同时使用 Webpack 来对所编写的代码进行管理，打包，优化等一系列操作。这样选择是技术语言上尽可能保持一致，同样为 JavaScript，减少额外的学习负担，能够将更多的精力集中到业务逻辑上面的开发。同时，系统的 UI 界面简洁朴素，风格相近，操作简单易懂，将重点放在了用户创建、编辑简历的使用体验上。数据结构设计合理，优化了查询功能，提供图表可视化报表查询功能，极大的方便了管理员的日常统计使用。该简

历制作系统能够符合现代化的简历制作系统的所需功能，能满足对于需要制作简历的用户的需求，系统是确实可行的。

从经济可行性角度来看本平台是能够开放给全世界人员使用，但是这样就需要有一个个人专属的服务器，能够连接上外网有独立的 IP 和域名，阿里或者腾讯都有学生专属的优惠套餐，每个月只需要一顿饭的资金就能够使用一个个人专属服务器，因此经济上负担可以省略不计，是可行的，且系统目前规模程度并不算是大型的项目，因此普通的个人电脑完全能够胜任运行该系统，因此开发该在线简历制作系统在经济上的支出耗费是很小的。

社会可行性方面研究的话，随着时代的进步，经济和科技的发展，每一天，甚至每一小时，每一秒都可以说都有在求职的人员，这些人员可以说是该在线简历制作系统的潜在使用用户，因此从用户群众基数来看的话是可行的。并且因为该系统采用的基本是自给自足的做法，并不需要太多的管理人员管理该系统，只需要少量的系统维护运维人员和极少数的系统管理人员和开发人员就可以将该系统持续运营下去了。因此从社会可行性角度来讲是足够可行的。

综上所述，在技术、经济、社会这三个可行性方面来分析该在线简历制作系统都是可行的。

第 3 章 系统分析

3.1 业务流程分析

因为该在线简历制作系统涉及的业务逻辑比较多，这里就以几个比较重要的功能做一个分析。主要围绕着用户登录注册，管理员登录平台，用户创建组件，用户创建简历以及管理员查询数据报表这 5 个业务流程来做分析。集中描述各个环节的业务处理内容、处理顺序等要求。

首先是顺序流程的图例如图 3-1 所示，分为 5 类图案，每个图案代表不同的含义。

图 3-1　顺序流程图图例

首先是用户登录/注册的业务流程图，主要流程用文字简单说明就是用户进入系统页面后触发了需要用户登录的操作后会出一个登录注册弹窗，填写用户信息后会验证信息的准确性，如果正确填写用户信息后会允许用户登录，并让用户继续原本的操作。

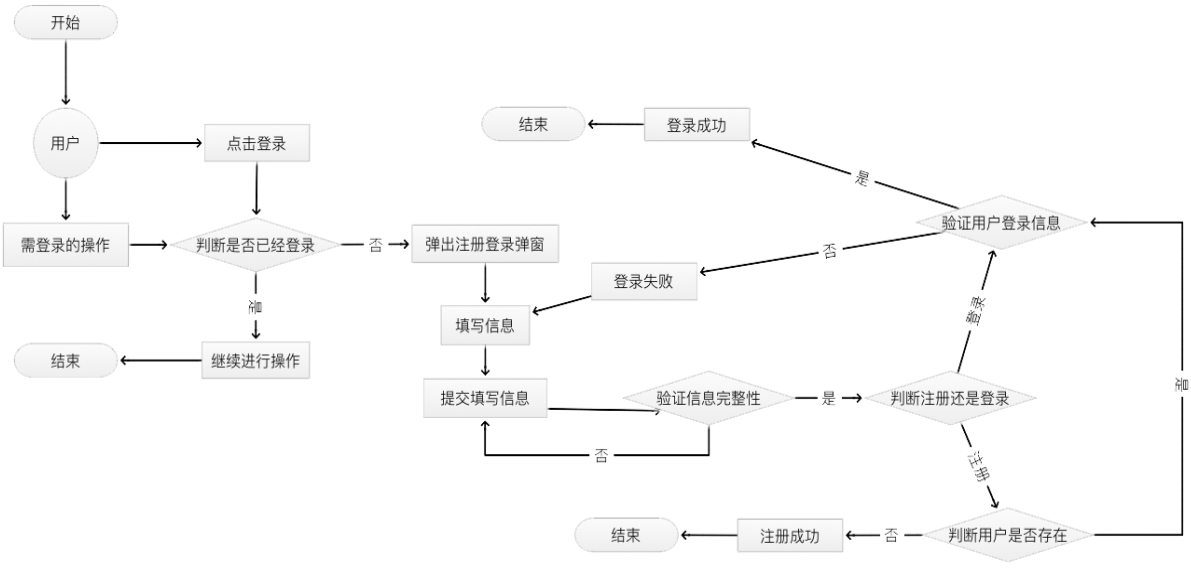

图 3-2　用户登录业务流程图

用户可以在前台创建一个属于自己的组件，并且可以发布到网络上面去让其他用户也能通过搜索来找到该组件。流程主要是通过自己个人喜欢后配置好组件的属性，然后可以预览组件的效果，如果满意的话可以点击保存创建一个属于自己的组件，这样子在简历的编辑页面里面就可以看到自己创建的组件并且可以使用了。

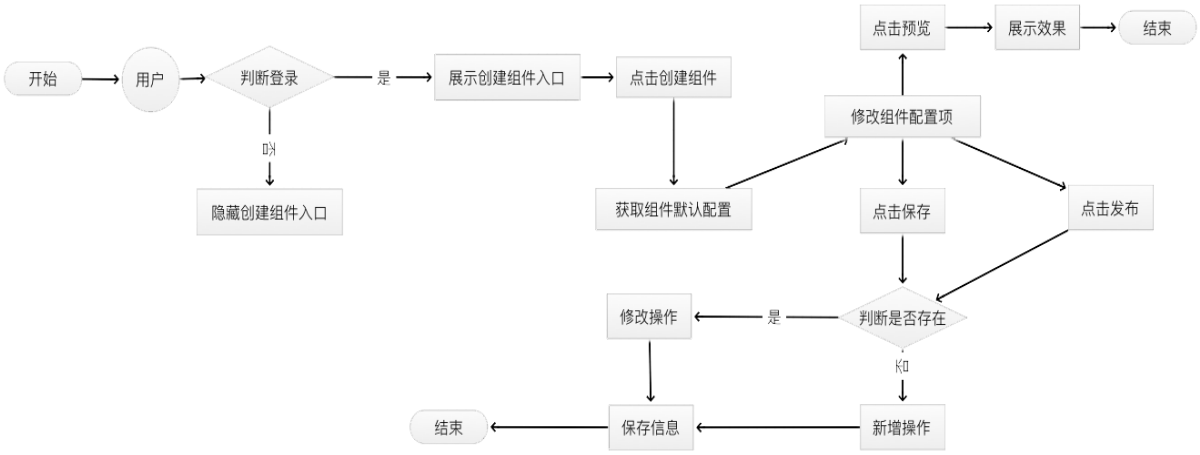

图 3-3　用户创建组件业务流程图

用户创建简历的流程主要是先选择想要创建的简历规格大小，然后是否使用简历模板来简单创建简历，初始化简历后，可以在系统界面左侧拖拽组件来使用组件，并且通过修改组件的各种属性来修改组件的风格样子。一个个组件叠加建造，最后形成一个个人简历，这时候可以保存到服务端数据库当中或者选择导出本地文件储存起来。

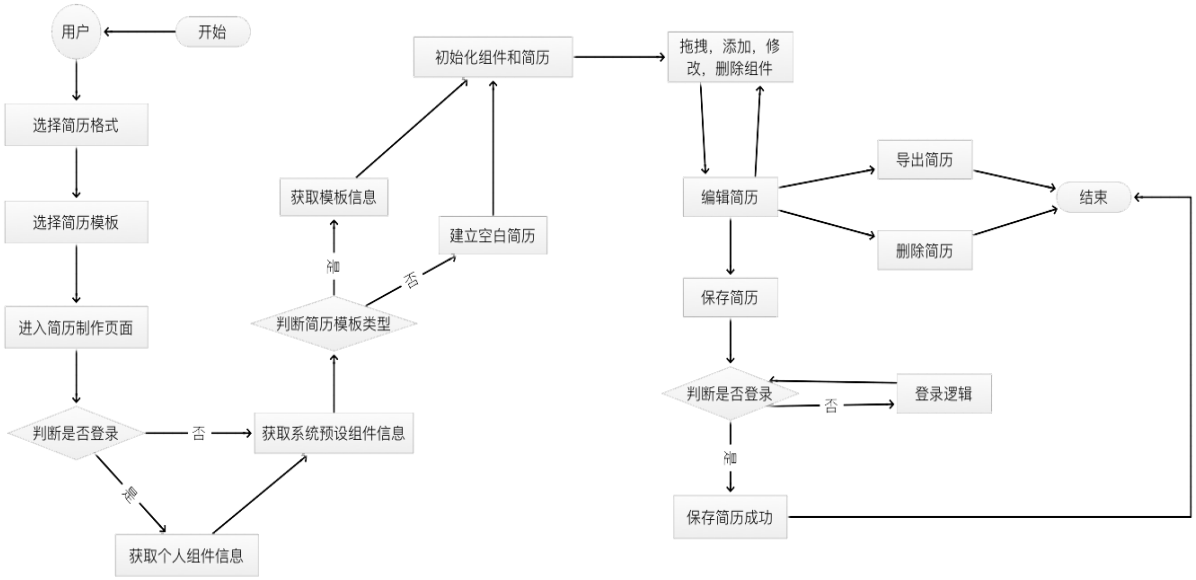

图 3-4　用户创建简历业务流程图

管理员登录后台则与用户前台登录比较类似，但是会多了一个流程，就是判断管理员的权限，若无权限的话虽然管理员能够登录成功，但是系统会拒绝管理员进行操作，这是为了防止管理员误操作的一个安全性操作。

首先是顺序流程的图例如图 3-1 所示，分为 5 类图案，每个图案代表不同的含义。

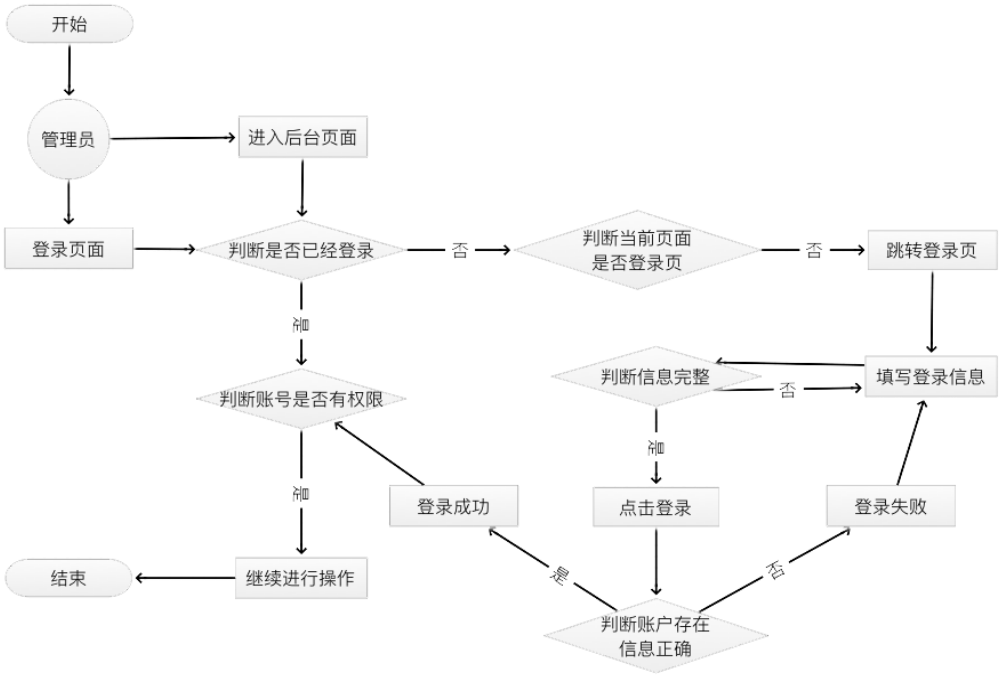

图 3-5　管理员登录后台业务流程图

管理员在查看报表的时候主要是用户在使用该系统后的统计数据，管理员通过限制搜索条件来查询不同的报表数据，通过图形化的报表数据直观的展示了用户对该系统的使用情况，从而更有效的对系统进行针对性管理和需求的迭代。

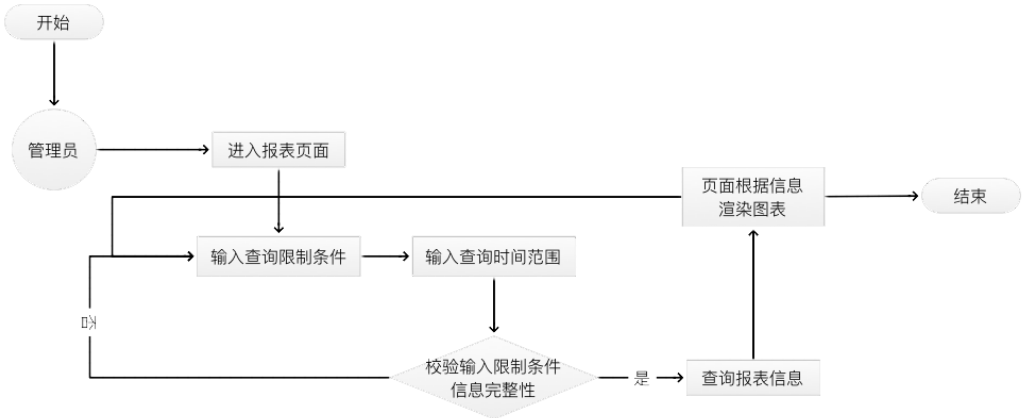

图 3-6　管理员查询报表业务流程图

3.2 数据流程分析

3.2.1 数据流图

首先是数据流图的图例如图 3-6 所示，分为 5 类图案，每个图案代表不同的含义。

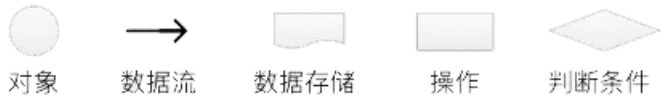

图 3-6　数据流图图例

管理员在后台管理系统查询数据流图显示了管理员在进行查询报表时候的数据流向，首先是要进入报表查询的页面，然后管理员需要填写和选择相关查询报表的表单信息数据，例如查询范围，名词搜索等然后系统会检验管理员输入的信息是否完整、符合规范；如果查询的限制条件数据没问题则前端页面会向服务端发送一个 ajax 请求，服务端接收到该 ajax 请求后会根据该查询条件信息数据向数据库查询对应条件的统计数据，该统计数据来源为用户在使用该在线简历制作系统时候所产生的各种操作数据，查询完成后将该查询数据返回给前端页面，前端页面最终再根据限制条件信息数据和查询出来的统计报表数据来显示不同的页面报表图表。

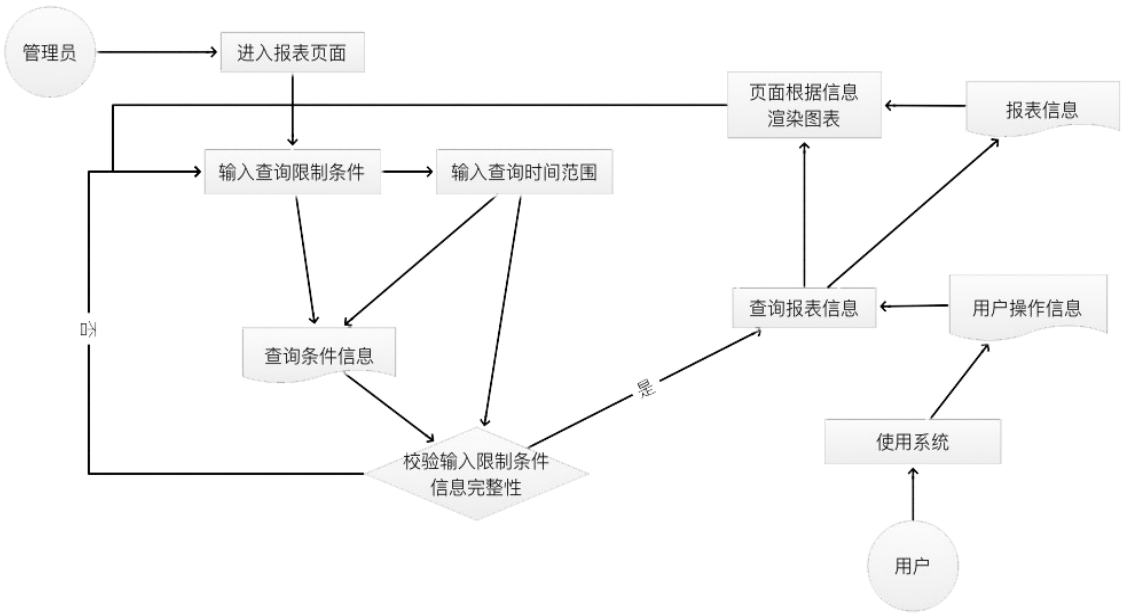

图 3-7　管理员查询报表数据流图

创建一个简历虽然说起来好像好简单，但是其实整个流程下来并不简单。首先是管理员吧需要在后台创建不同的简历格式，初始化了简历格式的数据，然后再供用户选择自己喜欢的格式进行创作，选择了格式后可能还会选择模板来方便自己制作简历，该模板数据也是管理员需要在后台进行模板的创建编辑修改等操作，用户才能在前台获取到各种各样的模板数据。简历数据初始化完成了，用户也应该使用组件来编辑、搭建自己的简历了，这时候使用平台的组件这样就涉及到各方面组件数据来源了，主要来源包括了管理员管理平台创建的组件以及用户在前台创建并发布的个人组件。最后编辑修改完成形成个人简历，这时候可以保存到服务端数据库当中或者选择导出本地文件储存起来。

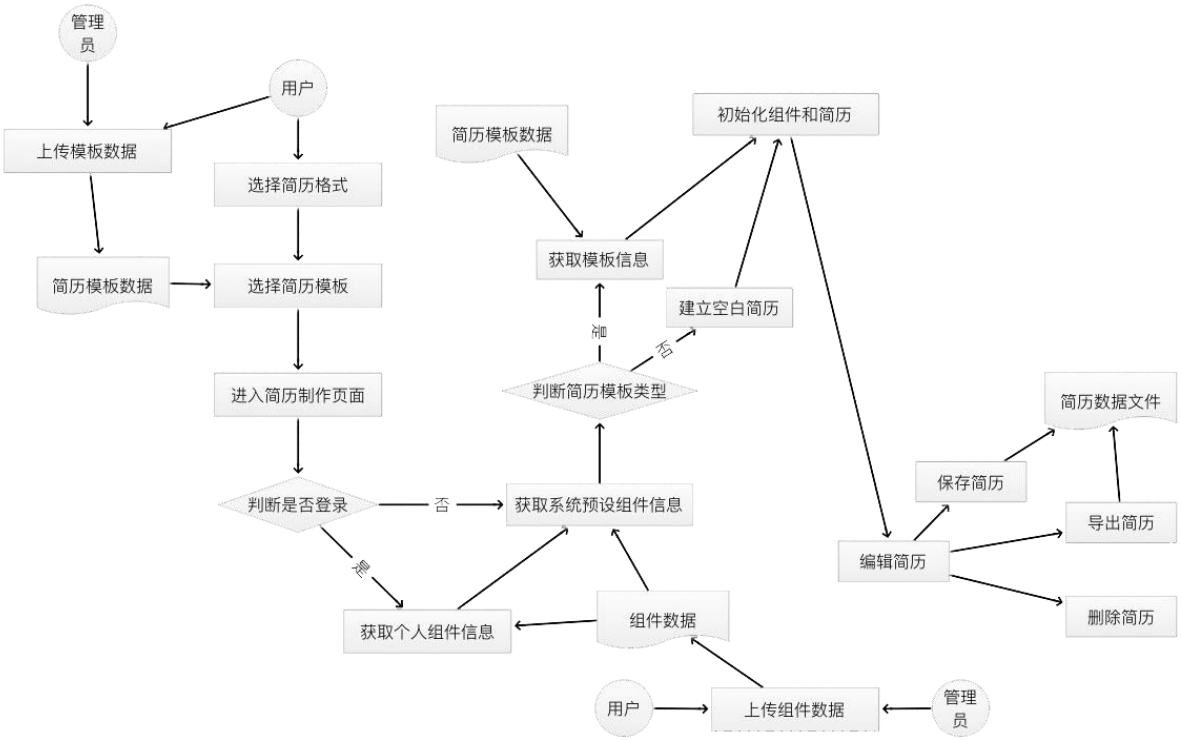

图 3-8　用户创建简历数据流图

第 4 章 系统设计

4.1 总体设计

个人的系统设计流程主要是功能设计，界面原型设计到数据库设计。首当其冲的是系统的总体功能设计，该在线简历制作系统最重要的功能当然是制作简历这个功能，其次是功能是登录注册、制作简历的组件相关功能，以及管理系统的一个数据统计功能。

接着是界面的原型设计，需要注意的地方主要是系统的总体色调要保持一致、风格要简洁，以及操作要简单不能复杂，最重要的是要给用户一种清爽、看着舒服的感受。

最后是数据库的设计，这里主要是简单的总结了各个表之间的关系方便后续设计详细数据库表的字段的设计，如图 4-1 是的数据表之间的关系，详细的数据库表的设计在后面的章节会有所讲解。

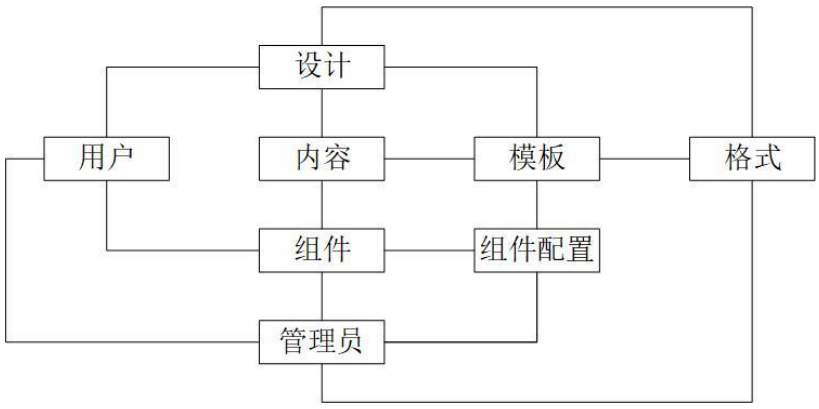

图 4-1　数据表之间的关系

4.2 数据库设计

4.2.1 对象关系模型

User (_id, name, avatar, mail, salt, password, cTime, lTime, status);
Admin (_id, nickName, userId, trueName, roleId, salt, password, cTime, lTime, status);
Role (_id, name, power, cTime, status);
Format (_id, name, logo, size, cTime, status);
Component (_id, name, logo, category, graphics, special, tags, conf, status);
Design (_id, name, logo, author, tags, bg, size, cell, cTime, mTime, status);
Template (_id, name, logo, formatId, tags, bg, size, cell, cTime, mTime, status);
ComponentCollection (_id, componentId, userId, time);

4.2.2 概念模型

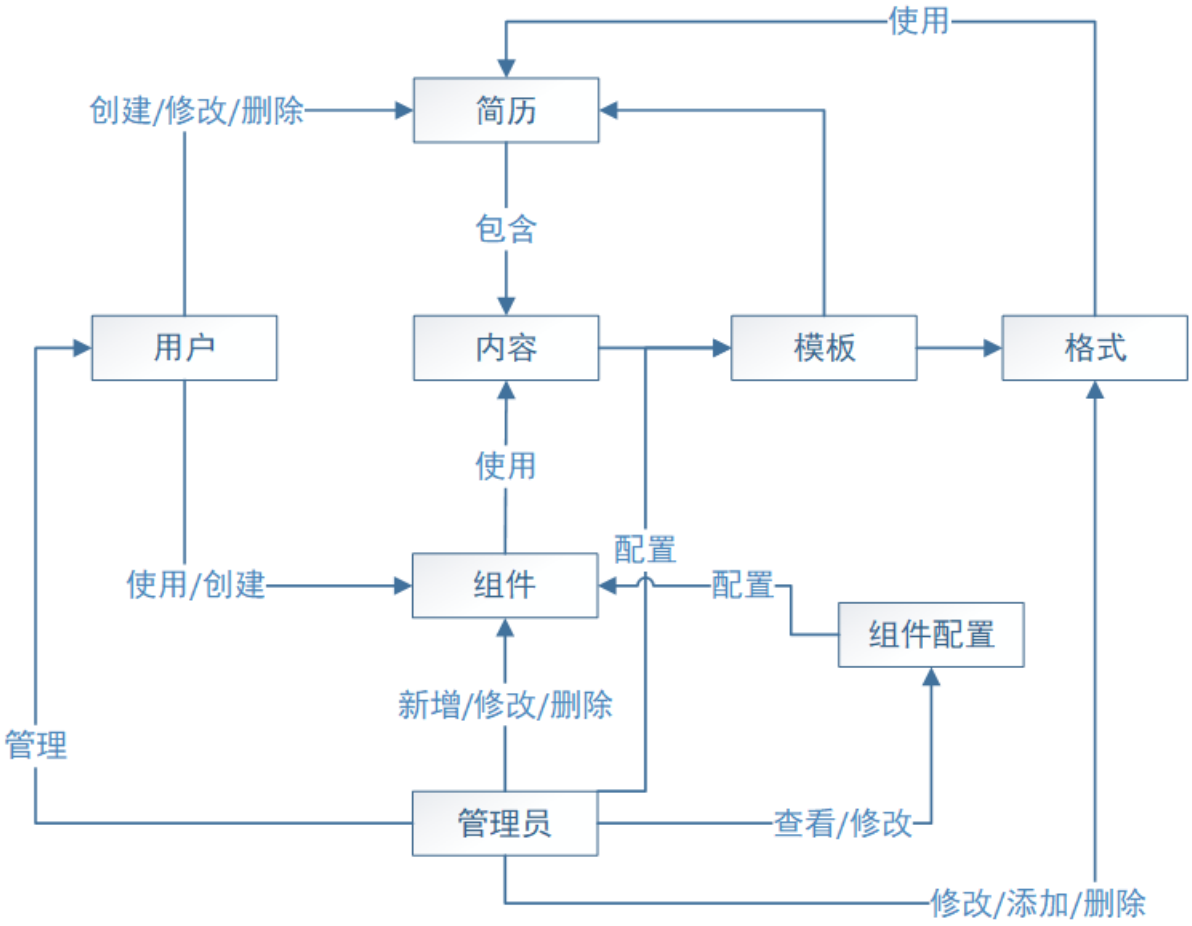

图 4-2　概念模型

4.2.3 逻辑模型

本系统采用的是 MongoDB 该数据库，因此为非关系型数据库模式。使用了如下数据表：

表 4-1　用户信息表 User

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
name	用户名		String	
avatar	头像		String	
mail	邮箱地址		String	
salt	密码盐		String	密码加密时候的盐

password	密码		String	加盐后 MD5 加密
cTime	创建时间		Date	
lTime	最后登录时间		Date	
status	账户状态		Boolean	true 正常，false 不使用

首先是基本的用户信息表，该表主要是存储用户的基本信息，包括用户名、头像、邮箱、密码等基本字段。

表 4-2　管理员信息表 Admin

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
nickName	昵称		String	管理员登录昵称
userId	用户 ID		String	对应用户 User 表_id
trueName	真实名称		String	管理员的真实名称
roleId	权限 ID		String	对应角色 Role 表_id
salt	密码盐		String	密码加密时候的盐
password	密码		String	加盐后 MD5 加密
lTime	最后登录时间		Date	
status	账户状态		Boolean	true 正常，false 不使用

后台管理员的信息表该表主要是存储管理员的基本信息，和用户信息表有所关联，一个管理员账户需要绑定一个对应的前台用户账号，方便管理员使用前台进行监控使用系统，并且该表也与下表 4-3 的角色表有所关联，一个管理员对应一个角色，用于后台的 RBAC 权限管理对管理员进行权限上的管理。

表 4-3　管理员角色信息表 Role

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
name	角色名		String	
power	权限		String	该角色所拥有的权限
cTime	创建时间		Date	
status	角色分类状态		Boolean	true 正常，false 不使用

该角色表存储的是后台的不同角色，不同角色有着不同的后台权限，用来区分不同用户类的不同权限。

表 4-4　简历格式信息表 Format

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
name	格式名		String	
logo	封面		String	图片的路径 & 文件名
size	长宽大小		Array	格式的长和宽大小
cTime	创建时间		Date	
status	状态		Boolean	true 正常，false 不使用

表 4-4 是存储用户创建新简历时候默认提供选择的简历规格大小的数据库表。

表 4-5　简历组件信息表 Component

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
name	组件名		String	
logo	封面		String	图片的路径 & 文件名
category	组件分类		String	extend/base/prevent/advance
graphics	图形类组件		Boolean	是否为图形类
special	特殊组件		Boolean	特别定制组件
tags	搜索标签		Array	
conf	组件基本配置		Object	组件的详细配置属性
status	状态		Boolean	true 已发布，false 未发布

这组件信息表是该系统的核心表之一，存储着各种编辑简历时候使用的组件的信息。通过配置这些信息可以简单创建一个属于自己的特殊的组件，并且可以使用到自己的简历当中。

表 4-6　简历信息表 Design

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
name	组件名		String	
logo	封面		String	图片的路径 & 文件名
author	作者		String	对应用户 User 表_id
tags	搜索标签		Array	
bg	简历背景		String	存储简历的背景色
size	长宽大小		Array	格式的长和宽大小
cell	简历内容		Array	储存简历内容（使用的组件）

cTime	创建时间		Date	
mTime	修改时间		Date	
status	状态		Boolean	true 正常，false 不使用

简历信息表是存储用户创建保存的个人简历的信息，里面存储了该简历的各种信息，如背景色、简历规格大小、组件使用的情况、创建时间、最后修改时间，创建人等信息。

表 4-7　简历模板信息表 Template

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
name	组件名		String	
logo	封面		String	图片的路径 & 文件名
formatId	格式 ID		String	对应格式 Format 表_id
tags	搜索标签		Array	
bg	简历背景		String	存储简历的背景色
size	长宽大小		Array	格式的长和宽大小
cell	简历内容		Array	储存简历内容（使用的组件）
cTime	创建时间		Date	模板的组件信息
mTime	修改时间		Date	
status	状态		Boolean	true 正常，false 不使用

简历模板表主要是记录一个不同简历格式的初始化模板，提供一个能够给用户方便快捷创建漂亮的简历的功能，里面记录了该模板的信息，主要是该模板使用的组件的配置信息。

表 4-8　简历组件收藏信息表 ComponentCollection

字段名	意义	主键	类型	备注
_id	ID	*	String	区分每一条记录
userId	用户 ID		String	对应用户 User 表_id
componentId	组件 ID		String	对应组件 Component 表_id
time	时间		Date	

该表会存储用户收藏组件表，一条数据对应用户 id 和组件 id 和收藏组件的时间，可以用于将来统计需求迭代使用。

4.3 I/O 设计

这里主要以用户前台编辑简历和后台管理员查询统计报表这两个子功能为例子来描述系统的 I/O 设计思想。

首先管理员在后台查询组件使用情况报表的例子，如图 4-2 所示，获取报表数据的输出，需要填写开始统计的日期时间还要截至统计的日期时间，以及需要选择一个组件的不同类型的数据；填写完查询条件的输入数据后，点击确定查询的按钮后向后台发送查询报表数据的请求并且根据该查询条件查询输出对应的时间范围内不同组件类型下的所有组件的使用情况，并且前端根据这些输出数据描绘成一个饼状统计图，计算使用比率所占比例，图形化展现输出的报表数据，方便管理员查看和分析。

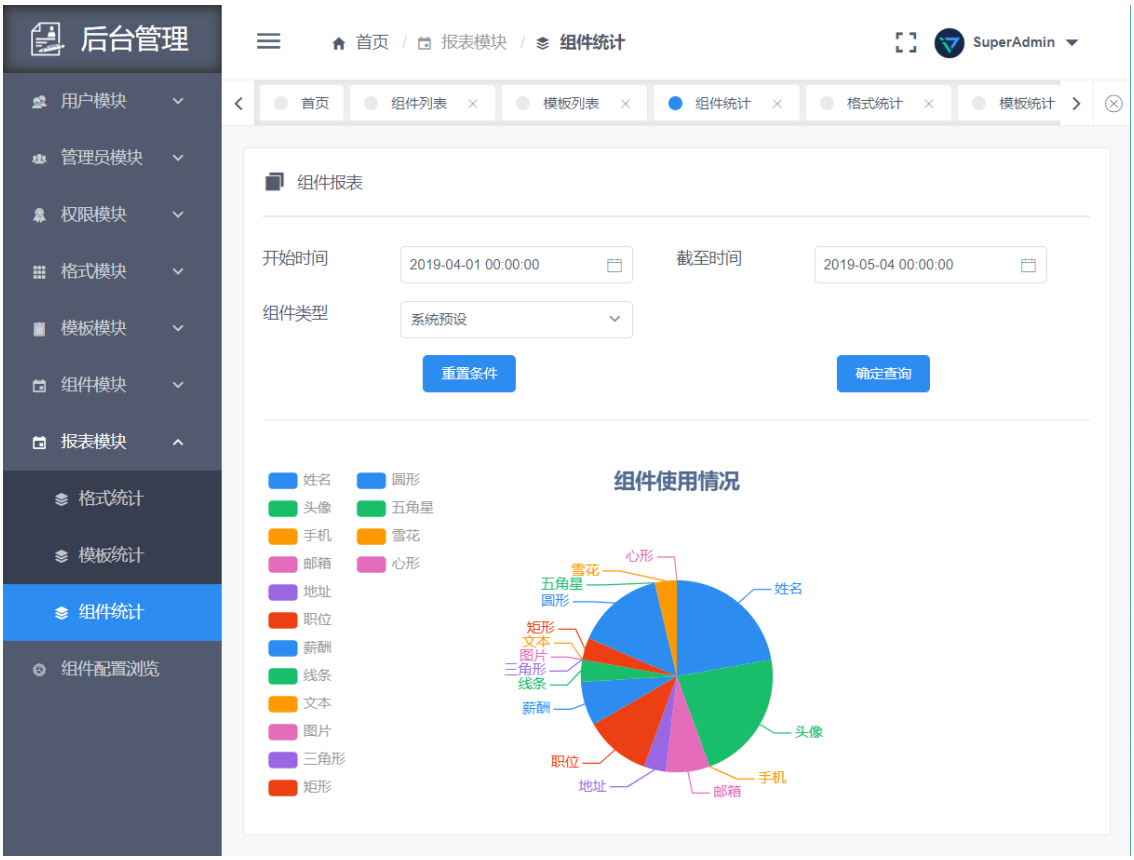

图 4-3　管理平台查询组件使用情况报表界面

首先用户前台编辑简历的例子，如图 4-3 所示，该在线简历制作系统编辑简历的输入主要为拖拉拽菜单组件，以及组件进行配置。如下图系统的页面部分，左侧菜单选择组件，右侧为当前简历上所含有的组件，上面导航栏是当前组件的配置项，可以通过这

几个地方对简历进行输入，形成一个个性化搭配构建，进而搭配出自己的一个简历。输出即为根据自己配置的一个个组件搭配起来，最后形成自己的个性化的专属简历。如下图中的中间部分即为简历输出的显示区域。

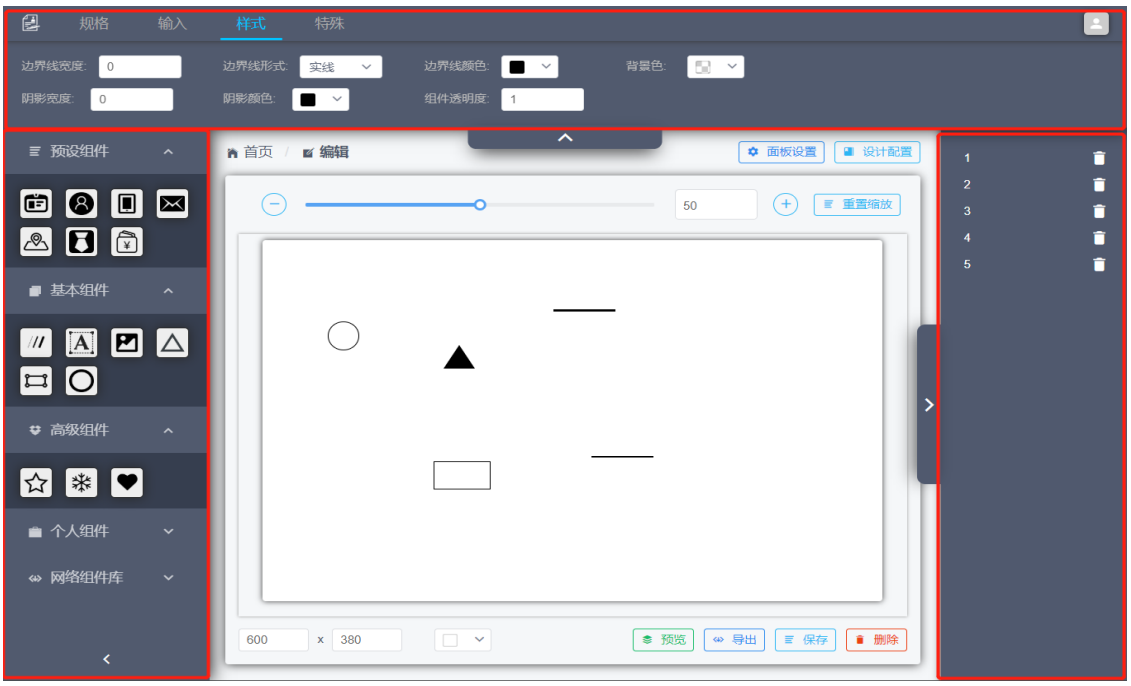

图 4-4　用户前台编辑简历界面

第 5 章 系统实现

5.1 系统实现

一个系统的实现，抛去需求分析以后首要的事情也是最重要的一件事情就是技术栈的选择。如果选择了一个不恰当的技术栈，不仅在开发的流程上会遇到各种各样的险阻，并且也会在以后的需求迭代上面造成阻碍。因此技术选型是系统实现中十分关键的一步。所以这次技术上的选型就主要是 JavaScript 为开发使用的语言基础，前端使用 Vue.js,后端服务端使用的是 Node.js，主要是为了降低学习的成本，将有限的精力尽可能投入到业务逻辑上面去[4]。并且使用常见的并且热门的技术，网络上就会有成熟完善的方案、一些好用的组件能够提供给我们在实际开发当中借鉴使用，能够减少 BUG 的出现、闭门造车的几率。

技术选型完成后是根据之前的需求分析进行整体系统的 UI 界面的原型设计，主要描绘页面的布局，按钮等，以及数据库的设计、建库、建表、模拟测试数据等。开发实现系统最基础的首先是将前面所分析设计的数据库表创建出来，这里选择的数据库类型是 MongoDb 该非关系型数据库主要原因是该数据库存储的数据格式可以很方便的和 JSON 数据格式转换，并且因为之前是原型设计，所以设计的数据库表的数据结构可能并不是最好的，后期还可能因为某些需求的变更迭代导致数据库表数据结构的变更，MongoDb 在修改存储数据的数据结构方面对比其他关系型数据库有着得天独厚的优势，因此在这里我选择了 MongoDb 该数据库作为存储数据的数据库[10]。

创建了表后面则是页面前端和服务端 API 接口的实现了。这里所选用的编程实现语言统一是 JavaScript，这样子就能够简单统一前后端的实现技术，降低学习成本和能够提高前后端的开发速度和效率。虽然选用了同样的编程语言来实现该系统，但是在真正编程实现功能需求上面还是会遇到不少的问题或者思路上还不清晰的阻碍以及免不了会遇到各种技术上的问题，例如组件的位置、放大缩小问题，颜色调节问题，不同浏览器下面的微小差异等。例如在前端组件可视化构建的实现当中则遇到两个比较麻烦的问题，一个是放大缩小的问题在前端有些组件例如图片上传，复杂 CSS 图形的组件在实现拖拽放大缩小上难以实现，另一个问题则是在一个编辑简历面板区域当中，不同组件的层级关系问题，层级优先问题处理不好不仅会影响到用户体验，并且最终简历显示效果也会受到明显的影响。在后台管理平台的实现当中遇到的最大问题则是管理员的权限问题，首先是这个权限如何在前端体现出来菜单上的切换显示以及禁止没有权限的管理员直接访问显示。以及一个权限大小粒度的切分，不同用户组有什么权限的这两个问题。

以及还有使用层面上的如何更好的优化用户和系统之间的人机交互。都是值得去花时间钻研和解决的。当自己真正一个人独立开发一个系统的时候发现开发一个系统的确需要极大的精力、时间和人的耐心，一个现代化的庞大的系统是靠多人协作以及时间、需求的不断迭代出来的，这就不仅仅是个人技术水平上的问题，还更加考验人与人之间的沟通、合作能力[12]。

5.2 系统测试

从系统的开发阶段上面来看，可以将测试分成单元测试、集成测试、系统测试这三种不同类型的测试方法。

首先是单元测试是针对软件设计的最小单位（功能、程序模块）进行校验正确性的一种测试方法，这一个在线简历制作系统的实现难点基本都在前端的页面展示和用户的交互上面，因此我将系统的测试主要都偏重在前端上面。但是因为前端以及某些功能是需要进行一定的复杂操作才能看到效果，因此测试起来也比较麻烦，单元测试就能够对系统进行更细小粒度的测试，更加精确的校验程序各个小功能模块之间时候存在错误，功能的实现。例如一个个组件的功能实现，正好可以使用单元测试的方法对一个个组件进行测试，来检验组件的功能时候正常。

当系统的功能逐渐开发迭代到了完成了一个个子系统、子功能模块之后我们就可以进入下一个阶段，我们就可以在这个阶段进行集成测试，集成测试是一种以单元测试的为基础，在此之上进行的一种测试方法，是把各个小程序、功能模块组合后形成一个子系统来进行一个单元之间的测试，是希望能够检验出软件单元之间的接口关系，以期望通过测试发现各软件单元接口之间存在的问题。例如我们在开发完成用户登录、用户信息修改等子单元功能模块，我们就可以组合成一个用户子系统，对该用户子系统进行集成测试，看一整套流程操作下来，系统内部之间的信息有没有正常的传递，功能是否能够正常的运作。如此循环迭代下去，开发完一个子系统然后进行一个个子系统的集成测试，查看各个软件单元直接的接口是否存在问题。

在整个系统的各个子系统都完成了并且进行了对应的集成测试以后，我们就可以进行最后一步系统整体测试了，在系统整体测试我们首先要需要注意测试一个系统的功能是否正常、完整，然后是系统运行的性能的测试，不能拉下的边界测试和容错性测试，在用户输入错误或者输入了一些意外的数据以及输入一些极端情况下的数据情况下，系统仍然能够正常运行并且提示用户信息错误等。安全性测试和可靠性测试也是不能够轻视的，因为在这个网络社会当中信息数据的安全显得尤其的重要，一个系统不应该很容

易的就被攻击并且泄露了信息的话是一个很危险的存在，应该能够抵挡一定程度上的网络攻击。

我们在开发的不同过程当中采用这三种不同的测试方法，并且因为我们同时是开发者，能够知悉代码实现的一切，因此从是否关心内部结构的角度来看我们同时也是使用了白盒测试的测试方法。一个系统尽管只有自己亲身经历过一次次修改，一次次点击调试，不断踩坑并且总结上一次出现 BUG 的根本原因、难点、注意事项，这样才会有更大的进步。

在程序当中无论是微不足道还是极其重要的一个功能，都是由人一个个字符敲击出来的，就免不了会发生错误，因此编码以后的测试是开发流程当中不可缺少的一部分[11]。不经过测试的程序是极度危险的，宛如一个定时炸弹，不知何时不知何地，会因为某些导火索而引爆。在系统的编码实现过程当中也因为及时的自我测试，从而发现了许多问题，例如因为某些标点符号上面的错误使用而导致系统报错，因为调用函数的时间不对而导致程序发生意想不到的结果等；总结来看，很大程度上，程序会出现 BUG 都是因为自己的粗心大意造成的，个人会牢记这些血淋淋的教训，尽可能在职业道路上少犯错。在编码的时候要时刻怀着你的代码一定有 bug 的心理，要保持着敬畏之心。

总结与展望

在开发该在线简历制作系统的期间遇到了各式各样的预期所没预料到的情况，但是凭借着坚强的意志力和踏实的做出行动，最后还是一一克服了各种意外情况。成功的设计并实现了该在线简历制作系统出来。

该在线简历制作系统主要功能就是为了提供给用户一个在线能够编辑简历的功能，系统还包含 4 个子系统的功能，分别是用户管理子系统、组件管理子系统、简历管理子系统和统计报表子系统；系统的功能都是紧围绕着这四个子系统进行展开，且分开前后平台开发与使用，前台供用户使用创建简历，后台主要供管理员对该在线简历制作系统管理维护使用。实现了如下几个主要功能：

1、 简历的创建、编辑、修改、保存导出功能。这个功能可以说是整个系统的最核心、最重要的一个功能。

2、 用户和管理员的登录注册，修改信息，管理员 RBAC 权限管理等相关功能。

3、 简历组件的创建、修改管理功能。

4、 平台用户使用情况报表展示功能。

系统主要操作简单、上手快、人机交互好、方便快捷；UI 界面简洁、风格统一；系统维护成本低、管理方便、系统安全可靠以及系统开发周期短，需求能够快速的进行更新迭代的特点。

但是任何系统都不可能说是完美的，从需求的实现、业务的扩展和迭代还是系统交互的优化、bug 的修复来看任何系统都是不完美的，该在线简历制作系统也是如此，是不完美的。总结了一下，该在线简历制作系统还存在以下三个不足或者能够改进和业务迭代的方面。

1、 前端编辑简历时候编辑区域组件层级问题。不同组件的不同层级关系，层级优先问题处理不好不仅会影响到用户体验，并且最终简历显示效果也会受到明显的影响。

2、 RBAC 权限分配的问题，首先是这个权限如何在前端体现出来菜单上的切换显示以及禁止没有权限的管理员直接访问显示。以及一个权限大小粒度的切分，不同用户组有什么权限的这两个问题。

3、 最后是系统的运营和需求迭代、系统更新问题，虽然系统的功能更新迭代可以

说很方便、简单、快速，但是目前系统也只是一个原型，如果真正投入到使用当中的话还需要考虑运营，系统的后期维护更新以及一些业务功能需求的迭代。这方面是只有从真正工作实践当中才能够获得更多更有质量的经验，因此对于在线简历制作系统想要真正投入使用还是任重而道远。

毕业系统设计是难得的一次可以从零开始的逐步分析、构建搭建一个系统出来的机会，因为个人比较偏向于代码实现的方面上，在工作以及学习上比较少几乎能够亲身参与到整个系统的详细设计中，虽然说系统可能功能上并不是特别完善，但是这样却能够更加让我个人深刻的了解熟悉到一个项目的完整的开发上面的流程是怎样的，将来对自己的人生职业路线规划相信也是一个很重要的体验。首先是该系统的问题的定义和规划，这阶段主要是确定该系统的开发目标，目的已经该系统的可行性。接着是系统的需求分析，需求阶段是一个很重要的阶段，在这一个阶段上面更加应该多花点时间下去研究，这一个阶段将会为一个软件项目的开发打下一个良好的坚实的地基。许多系统在后期面临着维护困难的问题就是因为在前期并没有做足够的需求分析的功夫与时间，导致了后期经常改需求，进一步拖延了项目的进度和加大了项目开发的困难度。因此，在这里更加看中了系统需求分析这块的内容。系统需求分析后便是软件设计的部分，这部分就需要根据上一部分需求分析所构思好的功能等来进行系统的设计（总体设计和详细设计），如框架设计，页面设计，数据库设计等。一个好的软件设计会为软件编写实现提供良好基础。设计后就需要编码实现，这部分就可能需要自身的技术实力了，日常也很少有机会能够整个人去挑起整个系统的开发任务，因此这次是一个很重要的经历，也是对自身的一次挑战。幸好选用的技术栈还算比较合适，再遇到问题的时候能够在网上快速的找到对应的解决方案，节约了不少时间和精力，能够更加专注于业务逻辑上的问题。最后便是软件的测试了，在软件设计完成甚至在设计中就应该对系统进行不同的测试了，如单元测试、组装测试、系统测试等方法，来对系统进行一个全面的测试。

经过一系列的开发流程后，我更加深刻的认识到一个好的项目、系统的开发不仅仅依靠过硬的开发技术，还更加需要前期的分析与设计，因为只有好的一个分析与设计，编码实现才会更加顺畅地进行下去。前期的规划就如同地基，只有稳固的地基才能够承受得住一座华丽的高楼大夏。最后引用阮一峰老师的一句话：画作永远没有完工的一天，你只是不再画下去而已[13]。

参考文献

[1] 田艳.管理信息系统（第二版）[M].广州:暨南大学出版社, 2011.
[2] 龚晓庆.面向对象系统分析与设计（第二版）[M].北京:清华大学出版社,2008.
[3] David Gourley，Brian Totty，Marjorie Sayer，Sailu Reddy，Anshu Aggarwal.HTTP 权威指南[M].北京:人民邮电出版社,2012
[4] David Flanagam.JavaScript 权威指南[M].北京:机械工业出版社,2012
[5] 阮一峰.ES6 标准入门（第三版）[M].北京:电子工业出版社,2017
[6] 朴灵.深入浅出 Node.js[M].北京:人民邮电出版社,2013
[7] Gary Nutt.操作系统[M].北京:机械工业出版社,2005
[8] Steve Krug.点石成金:访客至上的网页设计秘笈[M].北京:机械工业出版社,2006
[9] 孟祥旭.人机交互教程[M].北京:清华大学出版社,2016
[10] Kristina Chodorow.MongoDB 权威指南[M].北京人名邮电出版社,2014
[11] 梅尔斯.软件测试的艺术[M]. 机械工业出版社,2004
[12] Kenneth C.Laudon/Jane P.Laudon.管理信息系统[M].机械工业出版社,2015
[13] Paul Graham.黑客与画家[M].人民邮电出版社,2011

谢辞

时光荏苒，大学四年的时光即将流逝完结，随着这一份毕业设计的完成，也正是意味着我的大学时光的尾声。在做毕业设计之前，个人职业方向是主要偏重于编码实现方面，在日常的开发当中对一个系统的整体把握程度上来说是比较薄弱的，但是经过导师王淞春的提点指引下，从零开始，从系统规划、需求分析起步，一步一个脚印的走下去，最终实现了该在线简历制作系统。在这个系统设计实现过程当中让我体会到了开发编码实现只是一个系统实现当中一个台阶，虽然说不上最重要，但是也是实现系统的一个必不可缺的步伐。更重要的是系统整体之间的联系，只有开始的系统规划、可行性研究明确了系统的目标方向，需求分析明确了系统的详细功能，到了编码阶段才能够更加明确的朝着系统目标方向进发，最后还要配合系统测试阶段，对实现的系统功能与需求预期的效果是否相符合，这样子一个系统才能够说完成了。反观我们的人生也是如此，在学校当中要和同学、老师们之间相处融洽，遇到困难时候要互相帮助，只有懂得融入社会团体当中，在团队当中贡献出自己的一份力量，才能够使得这个团队更加有凝聚力，更加强大。

附录

附录 1 程序源代码

1、前端单个组件源码：

```
<template>
  <vue-draggable-resizable :w="cellData.conf.format.size.size[0] + 6"
                           :h="cellData.conf.format.size.size[1] + 6"
                           :x="cellData.conf.format.position.axis[0]"
                           :y="cellData.conf.format.position.axis[1]"
                           :minw="0" :minh="0" :parent="true"
                           @dragging="onDrag" @resizing="onResize" >
    <div class="draggable-cell">
      <component :is="cellData.special ? cellData.special : 'NormalCell'"
                 :data="cellData.conf" />
    </div>
  </vue-draggable-resizable>
</template>

<script>
  import VueDraggableResizable from 'vue-draggable-resizable'
  import NormalCell from '../NormalCell/NormalCell.vue'
  import SpecialCellLine from '../SpecialCell/SpecialCellLine.vue'
  import SpecialCellCircle from '../SpecialCell/SpecialCellCircle.vue'
  import                    SpecialCellTriangle                    from
'../SpecialCell/SpecialCellTriangle.vue'
  import                    SpecialCellFiveStar                    from
'../SpecialCell/SpecialCellFiveStar.vue'
  import SpecialCellSnow from '../SpecialCell/SpecialCellSnow.vue'
  import SpecialCellHeart from '../SpecialCell/SpecialCellHeart.vue'

  export default {
    name: 'DraggableResizableCell',

    components: {
      VueDraggableResizable,        NormalCell,        SpecialCellLine,
SpecialCellHeart,         SpecialCellSnow,         SpecialCellFiveStar,
SpecialCellTriangle, SpecialCellCircle
    },
```

```
    props: {
      cellData: {
        type: Object,
        default: () => {
          return {};
        }
      }
    },

    data: function () {
      return {
        width: 0,
        height: 0,
        x: 0,
        y: 0
      }
    },

    methods: {
      onResize: function (x, y, width, height) {
        this.x = x
        this.y = y
        this.width = width
        this.height = height
      },
      onDrag: function (x, y) {
        this.x = x
        this.y = y
      }
    }
  }
</script>

<style lang="scss" scoped>

</style>
```

2、前端简历编辑区域源码：

```
<template>
  <div ref="editContainer" class="edit-container"
       :style="`width:${$store.state.designConf.size[0]}px;
height:${$store.state.designConf.size[1]}px;                    background:
${$store.state.designConf.bg};`">
    <DraggableResizableCell       v-for="(cellItem,       index)       in
$store.state.designConf.cell"
                            :id="`cell${cellItem._id}`"
                            :key="`${cellItem._id}${index}`"
                            :cell-data="cellItem" />
  </div>
</template>

<script>
  import                    DraggableResizableCell                    from
'./Cell/DraggableResizableCell/DraggableResizableCell.vue'

  export default {
    name: 'EditContainer',

    components: {
      DraggableResizableCell
    },

    mounted () {
      this.$refs.editContainer.addEventListener('drop', (event) => {
        let                          cellData                          =
JSON.parse(JSON.stringify(this.$store.state.dragComponent));
        const size = cellData.conf.format.size.size;
        cellData.conf.format.position.axis = [event.offsetX - size[0] / 2,
event.offsetY - size[1] / 2];
        this.$store.commit('changeDesignConfCell', {
          op: 'add',
          cell: cellData
        });
      });
      this.$refs.editContainer.addEventListener('dragover', (event) => {
        event.preventDefault();
      });
    },

    beforeDestroy () {
```

```
        this.$refs.editContainer.removeEventListener('drop', null);
        this.$refs.editContainer.addEventListener('dragover', null);
      }
    }
</script>

<style lang="scss" scoped>
  .edit-container {
    margin: 0 auto;
    display: inline-block;
    position: relative;
    overflow: hidden;
    box-shadow: 0 0 6px #383838;
    border-radius: 3px;
  }
</style>
```

3、服务端简历数据保存逻辑：

```
const Router = require('koa-router');
const router = new Router();
router.put('/saveDesign', saveDesignCtr);

/**
 * 保存设计信息
 * @param ctx
 * @param next
 */
exports.saveDesignCtr = async (ctx: any, next: any) => {
  const param = ctx.request.body;
  const designId = param.id;
  let result;
  if (designId) {
    result = await updateDesignSer(designId, param);
  } else {
    result = await createDesignSer(param);
  }

  ctx.body = {
    status: 200,
    msg: '保存成功',
    data: result
  }
```

```
};


/**
 * 创建设计
 * @param designData
 */
exports.createDesignSer = async (designData: object = null) => {
  let design = new DesignModel(designData);
  const result = await design.save();
  if (result.errors) {
    return false;
  }
  return design._id;
};

/**
 * 更新设计
 * @param designId
 * @param designData
 */
exports.updateDesignSer = async (designId: string = '', designData: object
= null) => {
  const design = await DesignModel.findOne({_id: designId});
  Object.assign(design, designData);
  await design.save();
  return designId;
};

import { Schema, Model, model } from 'mongoose'
import IDesign from '../interface/IDesign'

const DesignSchema: Schema = new Schema({
  name: String,
  logo: String,
  author: String,
  tags: Array,
  bg: String,
  size: Array,
  cell: Array,
  cTime: {
    type: Date,
    default: Date
```

```
    },
    mTime: {
      type: Date,
      default: Date
    },
    status: {
      type: Boolean,
      default: true
    },
  }, {versionKey: false});

  const Design: Model<IDesign> = model<IDesign>('Design', DesignSchema);
```

4、Node.js(Koa2)连接 MongoDB 数据库：

```
  const mongoose = require('mongoose');
  // import config from './../config'
  // 连接
  mongoose.connect('mongodb://localhost/inline_cv', {
    useNewUrlParser: true
  });
  // 连接成功
  mongoose.connection.on('connected', function() {
    console.log('Mongoose connection success');
  });
  // 连接异常
  mongoose.connection.on('error', (err: string) => {
    console.log('Mongoose connection error: ' + err);
  });
  // 断开连接
  mongoose.connection.on('disconnected', () => {
    console.log('Mongoose connection disconnected');
  });
```