目录

[前言 1](#_Toc27068939)

[项目简介 2](#_Toc27068940)

[项目背景(解决什么问题) 2](#_Toc27068941)

[任务分析 3](#_Toc27068942)

[用户群体 3](#_Toc27068943)

[用户需求分析 3](#_Toc27068944)

[任务分析 3](#_Toc27068945)

[可用性要求 4](#_Toc27068946)

[概念设计 4](#_Toc27068947)

[设计草图 5](#_Toc27068948)

[项目实施 9](#_Toc27068949)

[前端页面设计 9](#_Toc27068950)

[后台接头搭建 10](#_Toc27068951)

[页面调整 11](#_Toc27068952)

[系统测试 11](#_Toc27068953)

[Bug修复 11](#_Toc27068954)

[项目整理 11](#_Toc27068955)

[评估 12](#_Toc27068956)

[反思 12](#_Toc27068957)

[项目代码地址 13](#_Toc27068958)

# 前言

随着互联网的发展，我们的生活中几乎随处可以看到互联网的影子，互联网对我们的生活改变无疑是非常巨大的。人类是一个群体，我们生活在这个社交群体中，信息交流是不可避免的，那么相应的信息存储也成了一个我们必须要解决的问题。诸如手机里的通讯录、笔记本上的笔记、还是电脑里的EXCEL表等都是我们存储信息的方式。本系统面向于社交小团体的信息存储，采用的是Web以及后台接口的配合实现了用户信息的存储与页面的可视化。

本系统页面设计考虑了人机交互的便利性、易学性、视觉性等等。后台采用json文件的格式存储信息，实现了前后端的分离，而且项目的可塑性比较高，只需要了解一些简单的Web知识即可根据自己的需求更改需要存储的信息类型。

# 项目简介

## 项目背景(解决什么问题)

在以往的信息存储中大致分为两种，一种是文件存储，一种是数据库存储。文件存储简单来说就是利用Excel之类的文件保存信息，优点是上手简单，只需要简单的学习就可以掌握，缺点就是信息的共享很麻烦，通常团队合作时会需要一个文件到处发，而且信息的同步性很难得到保障。数据库存储是目前市面上比较常见的方法，它几乎解决了文件存储的所有缺点，然而它也有它的问题，就是专业性比较强，首先就是数据库的搭建、维护，其次就是数据库一般得配合软件使用。

而本系统采用Web和json后台服务器的方法实现了信息的存储，将信息存储在服务器，用户可以通过访问Web页面实现用户信息的增删改查。采用Web的方法好处就是不用安装软件，随时随地都可以用电脑或者手机访问网页，实现信息的实时共享。

# 任务分析

## 用户群体

需要信息存储而又缺乏计算机相关知识的小群体（例如学生社团、小公司）

## 用户需求分析

因为面向的是对计算机了解比较少的群体，因此程序的易学性应该摆在首要位置，其次就是页面的设计，因为是用户管理系统，所以界面采用简洁风格，配色不要太多。然后就是因为用户群体性质的不同，所以系统的可塑性也很重要，可以通过很少的修改适应不同的群体不一样的信息存储要求。最后的目标就是完整的实现用户对信息的增删改查，并且有较好的交互性。

## 任务分析

将此系统的开发过程分为**页面布局设计、前端页面开发、后台接口搭建、页面调色以及美观、系统测试、Bug修复、项目的发布**7个部分。下面将分析这几个任务：

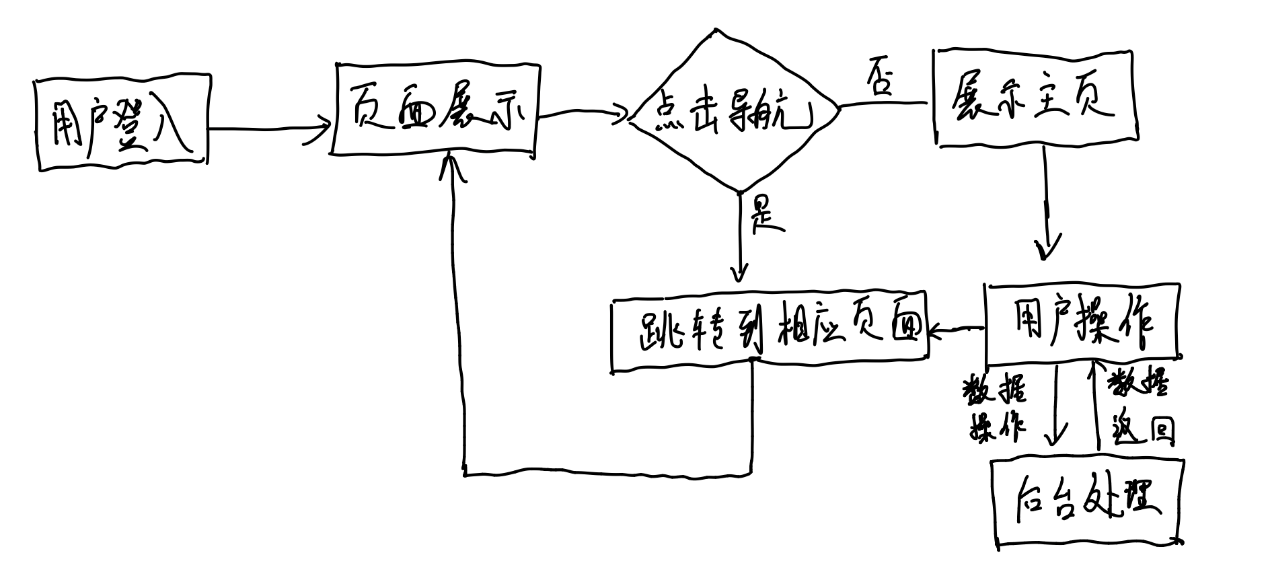
* **页面布局设计：**初步设计页面的布局，可以在纸上或者PS上设计出前端页面的几个页面的大致布局。
* **前端页面开发：**根据初稿设计出相应的Web界面，并实现页面之间的跳转等。先自行模拟数据完成功能开发。等后台接口搭建完成以后与后台对接，实现完整的功能。
* **后台接口搭建：**考虑数据的存储方式，以及给前端开发提供数据接口，根据前端传过来的数据进行增删改查，并返回数据。
* **页面调色以及美观：**待功能实现以后对页面整体的调色等美观问题进行调整。
* **系统测试：**待前后端完成对接以后对系统的功能进行测试，如果有觉得不好的地方也可以提出修改意见。
* **Bug修复：**根据系统测试提供的bug或者意见进行系统的调整与修复。
* **项目的发布：**将最后的项目运行结果录屏，编写readme文件并上传到github上进行项目的存储与开源分享。

## 可用性要求

实现用户对信息的增删改查，并且有良好的交互性。

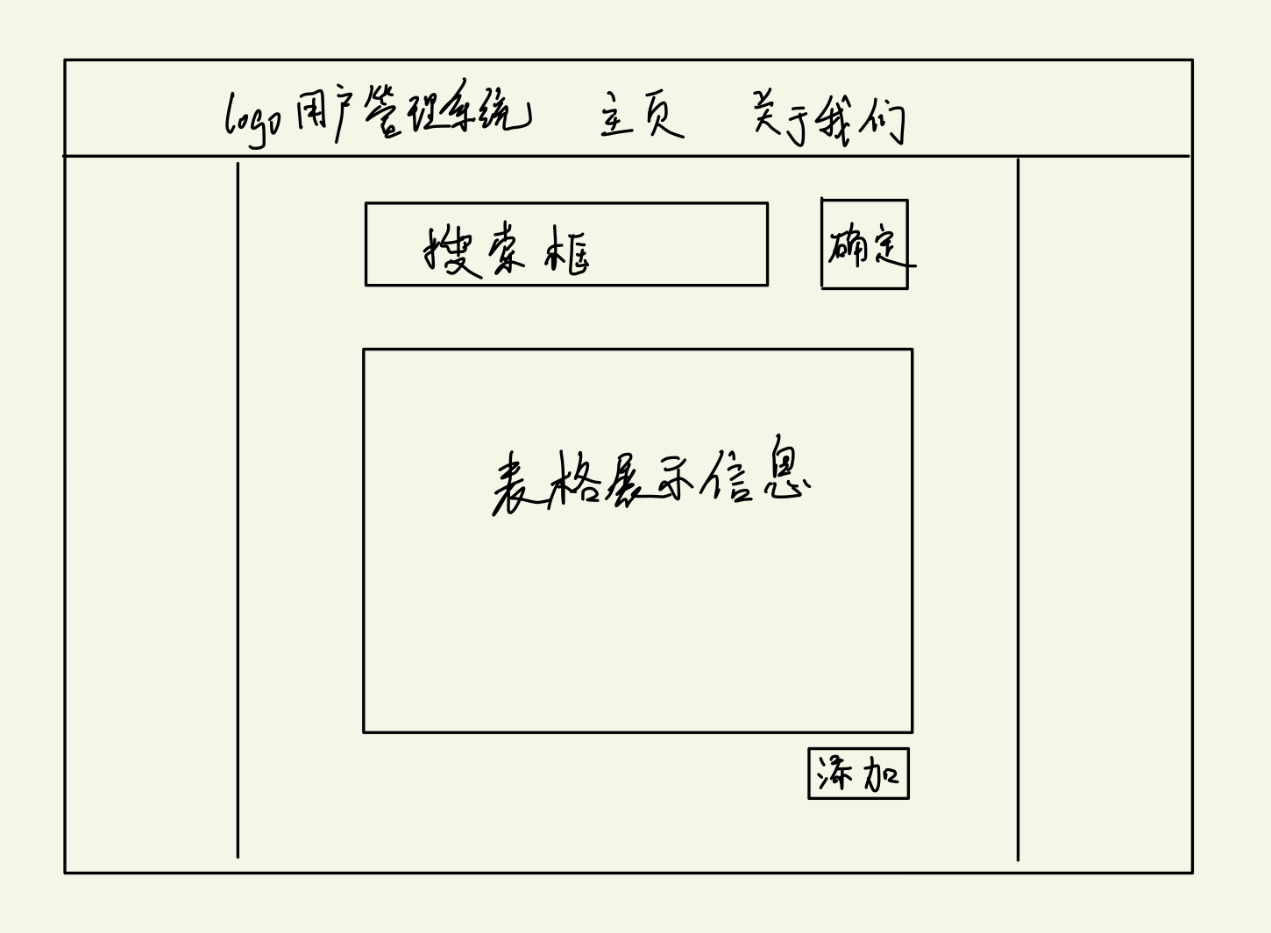
# 概念设计

用户输入网址登入，服务器返回系统主页，此时用户可以点击导航来切换不同的页面，也可以在主页中进行相关的增删改查等操作来实现页面的跳转以及数据的刷新。需要注意的是用户如果更改数据需要和后台数据同步，即同时修改后台的数据。



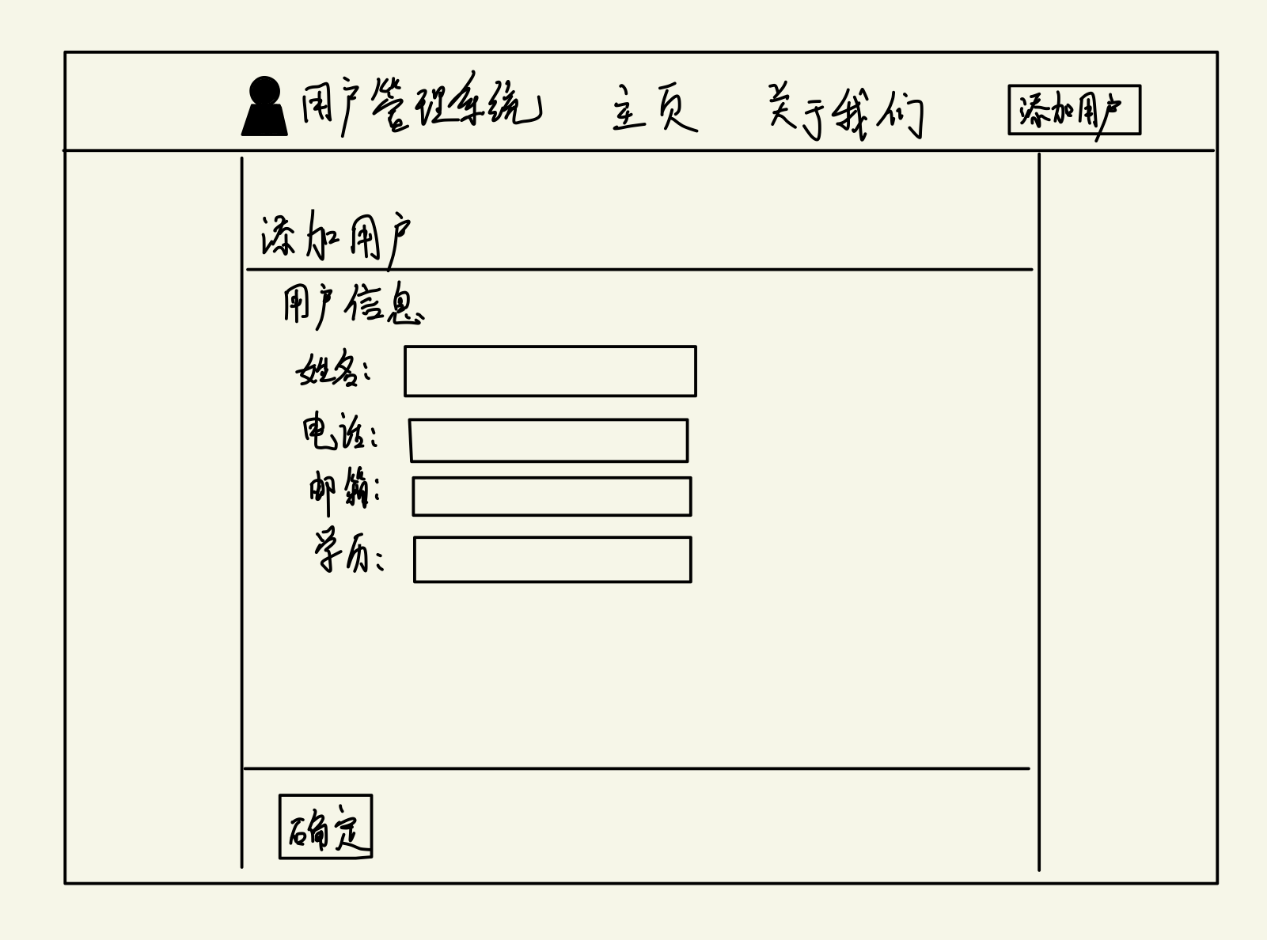
# 设计草图

**主页初版草图：**



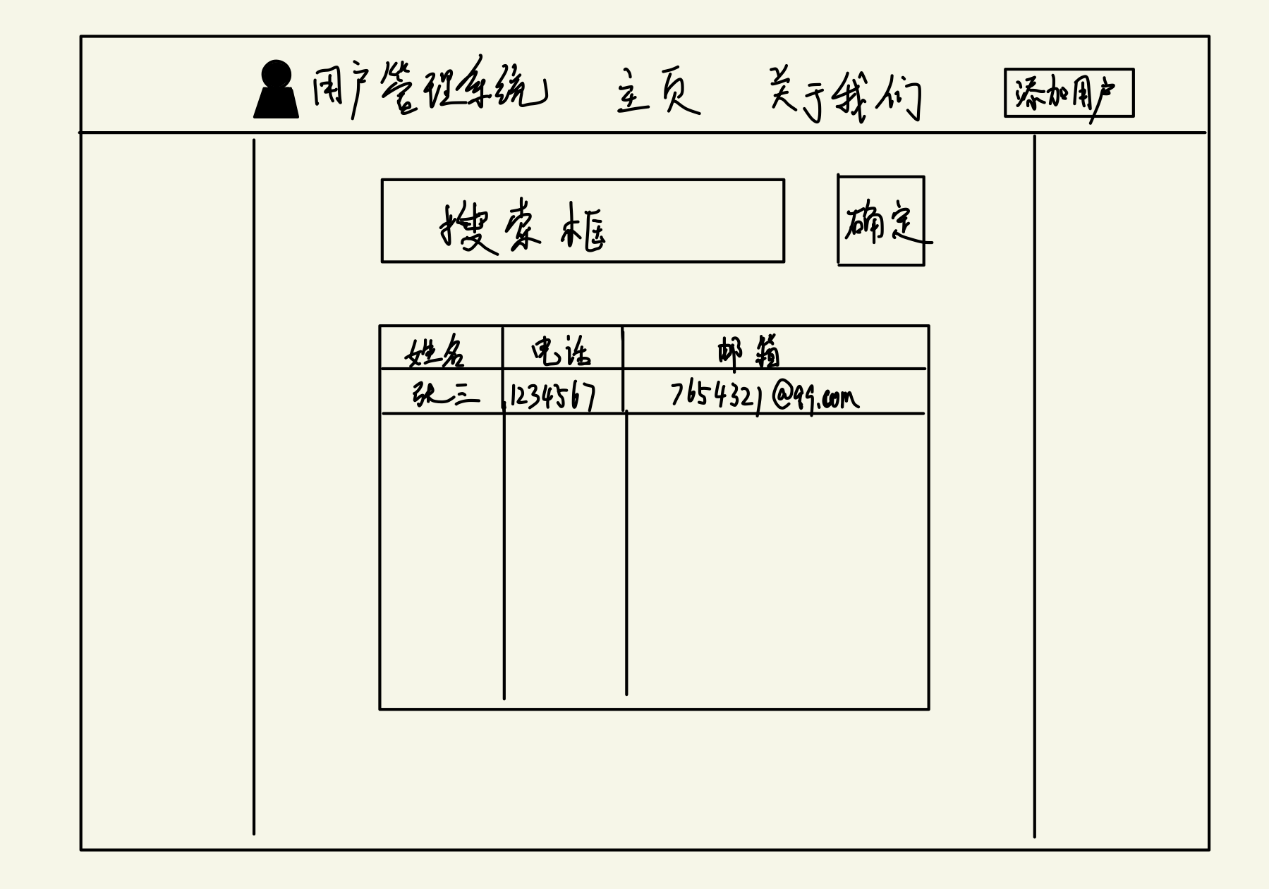
整个页面最上面是导航栏，用于页面的跳转，导航栏左边是logo(暂定)和系统名称，主页面左右等宽留白，搜索框放在最上面，便于用户输入信息查找信息，搜索框右边一个确定按钮，用户输入以后直接点击确定即可去查找信息并展示在表格这一栏。将添加用户的按钮放在了主页的表格最下面。

**添加用户页面草图：**



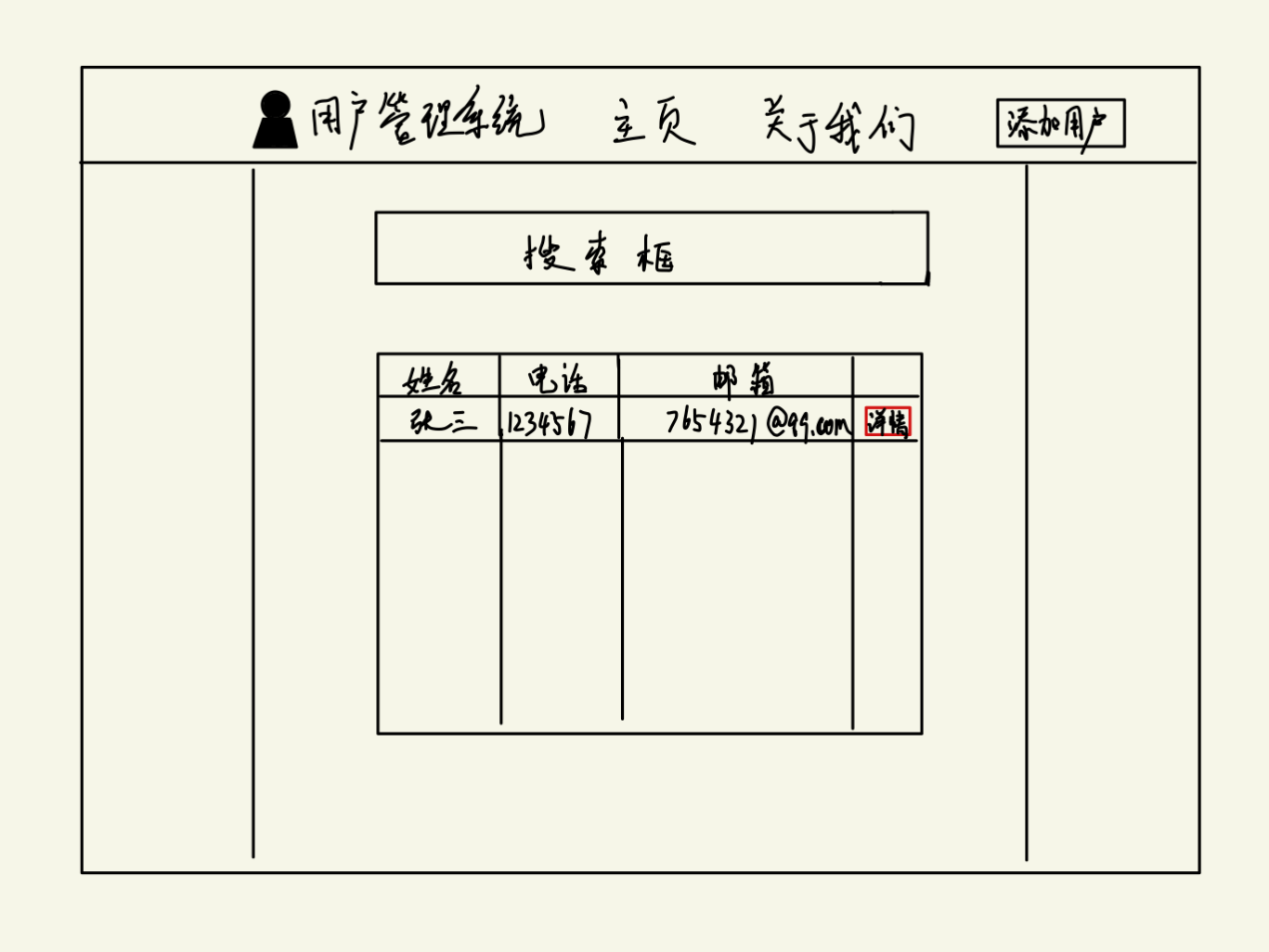
添加用户时用户在“用户信息”栏填写相应的信息，然后点击左下角的确定将信息更新到后台数据。并跳转到主页。

**主页修改版草图：**



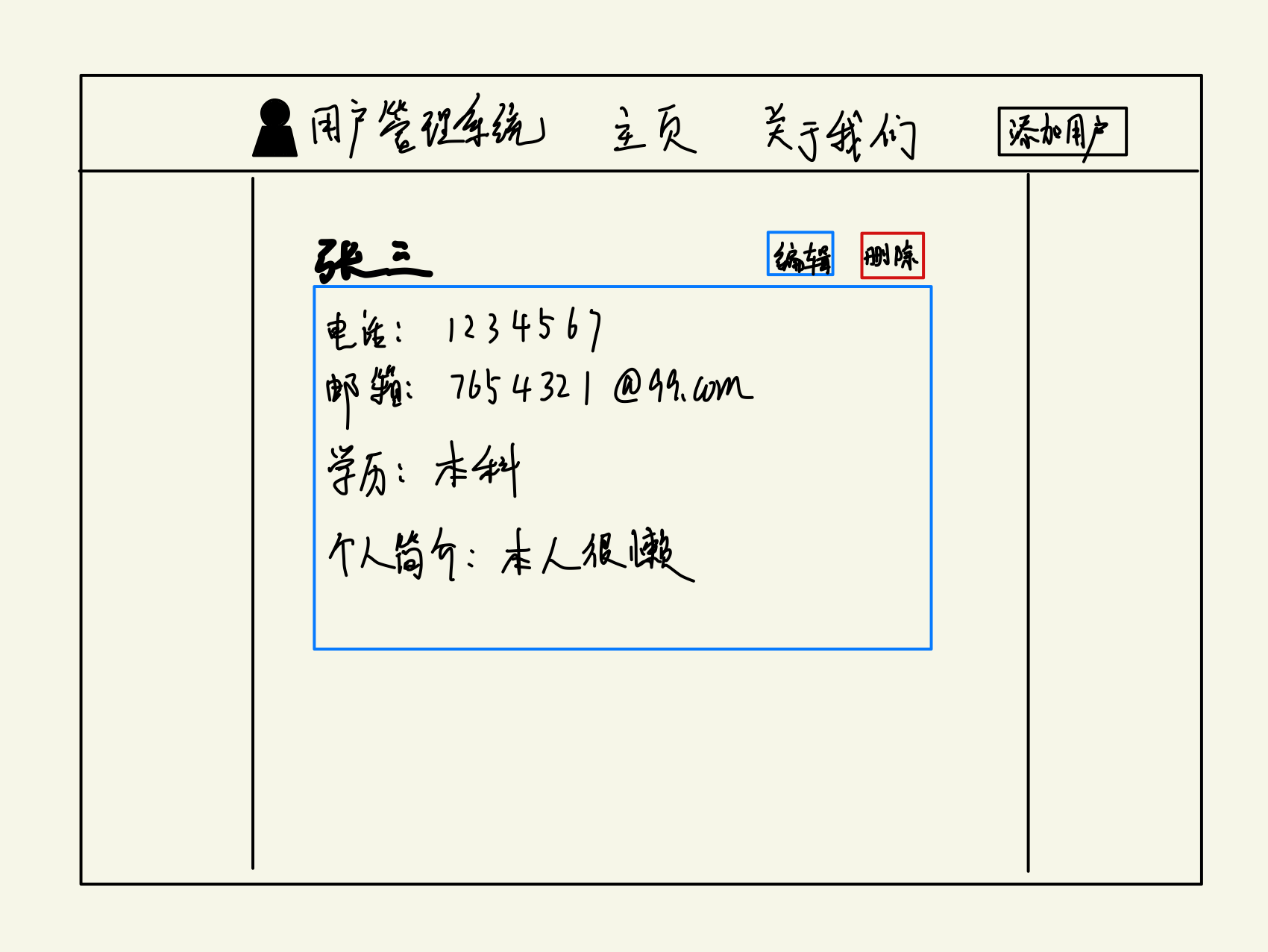
确定了logo的实际样式，因为是用于后台管理，所以采用了简单的一个小人作为logo，将添加用户按钮移到了导航栏，方便于用户在其他页面时也能够随时添加用户，而不用先返回主页再点击添加用户，减少了用户的点击次数。确定了表格的展示选项。

**主页最终版草图：**



将确定按钮去除，改为实时的对用户输入的信息进行查询，监听用户的输入事件，当输入变化时自动刷新表格。在表格后加了一个详情按钮，点击时进入另一个页面用于展示用户的详细信息，此处是考虑了不同用户群体可能需要存储的信息的多少，当用户信息较多时在表格展示显然不是一个好的方法，所以分出一个单独的页面进行信息展示。

**用户详情草图：**



将名字大写加粗，下面的框框用来展示用户的全部信息。右上角是两个按钮，编辑和删除。点击时分别对该用户进行编辑或删除。打算最终的信息展示用图标代替（例如“电话”用一个电话的矢量图代替）

# 项目实施

## 前端页面设计

利用确定的最终草图实现前端页面的搭建，前端页面采用目前主流的Vue框架搭建而成，页面之间的跳转利用Vue提供的路由，效率比较高。在数据获取方面通过vue-resource向后台提供的接口获取数据，然后将获取的数据经过处理展示在页面上。最终页面示例图：



（如需查看完整效果，请访问github：<https://github.com/onechunlin/User-manage>）

## 后台接头搭建

采用jsonserver搭建后台数据接口，安装好环境以后进行环境的配置，然后按照前端所需的要求建立初始数据格式，并提供给前端，以便进行前后端的交互。

完成效果图示例：



此时访问相应地址即可得到相应的数据，接口搭建完成。

## 页面调整

因为系统定位的原因，所以采用简洁风格配色，考虑了十几种方案，在和前端的配合下最终确定了以浅黄色为基调，配合橘黄色和墨绿色的方案，最终的效果配色可见前端页面最终示例图。

## 系统测试

在明白了系统功能以后以一个初学者的方向去随意的点击系统，以便系统没有跳转之类的低级bug，在测试完这些以后就是功能方面的测试，分别测试了断网重连、连接时断网、编辑过程未保存关闭等异常情况，得到的结果都还符合预期。最后就是功能优化方面的测试，最终在搜索时测试到当用户输入查找时英文区分了大小写，在最终小组讨论以后决定采用不区分英文大小写的方式进行搜索，提交bug给前端设计。

## Bug修复

将测试提出的英文不区分大小写问题解决，解决方法就是讲用户输入做全部小写处理，然后将其与接口数据中全部小写以后的数据进行对比，如果相等则显示该条数据。

## 项目整理

将项目最终结果进行打包整理，进行效果的录屏，然后整理开发报告，编写Readme文件，提交github做项目的留存。

# 评估

最终的系统效果我们找了十个同学（7男3女）进行测试。在通过观察他们使用系统的过程中，发现他们对于页面设计都比较满意，上手都很快，几乎只要是使用过电脑的人就可以不用我们指导完成他所需要的的操作。基本功能在测试过程中也没有出现问题。对于配色方面也比较满意，但是有个同学提出了一个建议，就是在搜索展示下面的结果时，希望将他输入的关键字高亮。

# 反思

在开发了这个项目以后，我了解了开发一个项目的具体流程。在项目开发的过程中要一步步的规划好，先做好初步的规划，然后在将其细化，刚开始考虑得太细后面其实也大概率没用，不要盲目的去开发一个项目，这样很容易事倍功半。还有就是草图在开发过程中的指导意义真的很大！而且学到了一个很重要的事，你的项目是面向用户的，所以得考虑到用户使用的各种问题，用户得摆在第一位。

当然这个系统在我这个开发者的方向来看还是有些不足之处，例如评估中同学提出的关键字高亮的问题，还有就是数据安全的问题，因为没有进行用户的认证，所以任何知道网址的人都可以查找信息。但是由于面向的用户群体是社团等以及时间的原因所以在这方面进行了取舍。

这个系统让我比较满意的地方就是它的创新方式，采用Web和json服务器的方式进行信息的管理，灵活性非常不错。可以根据用户的需要简单的修改就可以适用于不同的用户群体。

# 项目代码地址

项目代码已上传至本人github，如需查看详细效果或源码请访问：

<https://github.com/onechunlin/User-manage>