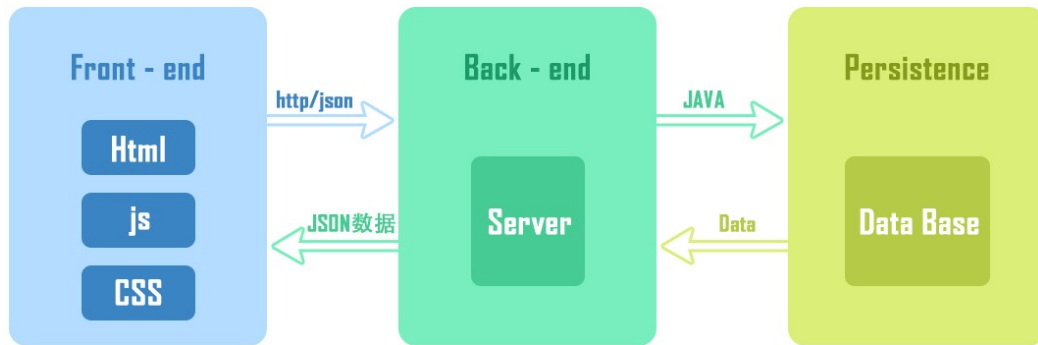


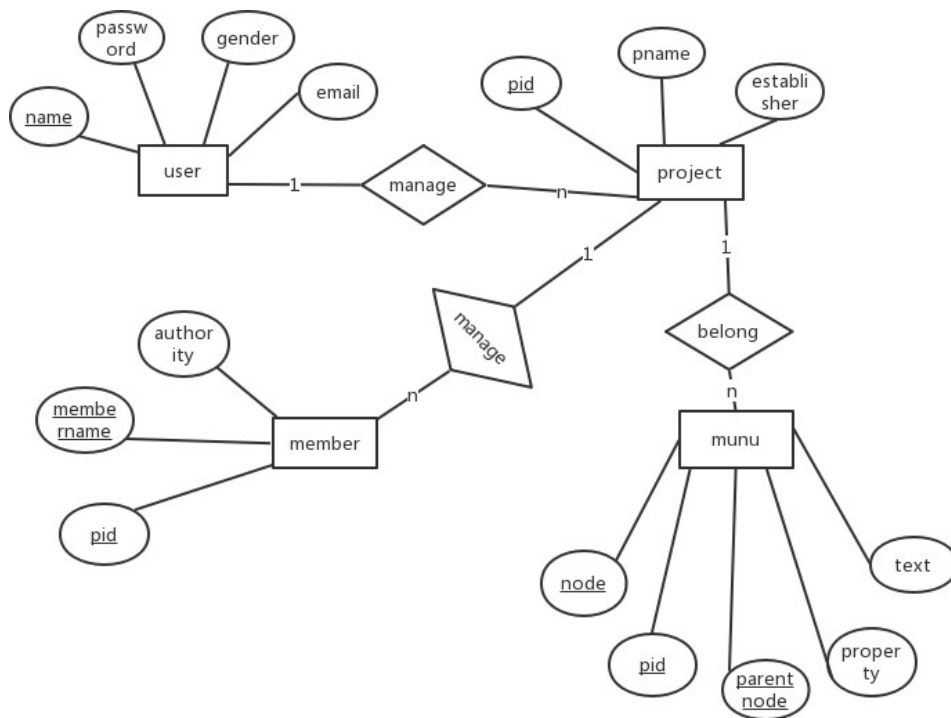
系统架构设计

实现前后端分离。由Html、TypeScript、CSS、Angular框架完成前端的页面的设计，使用http、json调用后台Restful API接口并用json实现与Server端的数据传递。

由后端提供Restful Api，并给出详细的文档说明，以便前端人员进行页面渲染前台的任务发送API请求（GET,PUT,POST,DELETE等）并得到相应的数据与相应的状态。



数据库设计:



用户表:

Name varchar[20]

Password varchar[20]

Gender integer //女0; 男1

Email varchar[30]

Name	password	gender	email
aa	*****	0	1234567788@163.com
bb	*****	1	1122334455@qq.com

项目表：

Pid integer

Pname varchar[20]

Establisher varchar[20]

Pid	pname	establisher
123	aa	ss
456	zz	xx

项目成员及权限表：

Pid integer

Membername varchar[20]

Authority interger

pid	membername	authority
1122	asd	7367
3344	zxc	3452

项目录表：

Pid integer

Node carchar[50]

Parentnode varchar[50]

Property interger//1代表是文件夹 0代表不是文件夹

Text mediumtext

pid	node	parentnode	property	text
156656	Fbgr4	Ver3	1	docx
556776	5tgef3	Sde23	1	txt

接口设计：

用户命令处理接口：

定义status类为返回类型，包含状态码，路由的url和异常说明

1. 登录：

1.1. 请求（GET / accounts / {name}）

Get请求报文：

```
{"user_password":" "}
```

响应报文：

若成功，则返回：

```
{"code":"200"
```

```
"data":{"空  
}"
```

若错误，则返回相应的错误码，

客户端错误状态码：

401：用户没有权限（或用户名、密码错误）

404：数据库中没有该用户想要的信息的记录

406：用户请求格式不可得（要求json格式） 服

务端错误状态码（同适用于所有错误返回）：

500：服务器发生错误，用户将无法判断发出的请求是否成功

data里跟错误附加信息

注：**401** 没登录错误

406用户请求格式错误

500：服务器器发生错误，用户将无法判断发出的请求是否

为通用错误，下述接口口均有可能发生该错误，故不再赘述正确时，为正确状态码加**data**：数据 以下只描述**data**数据

内容

2. 注册：

2.1. 请求（POST / accounts）

Post请求报文：

```
{"user_name":"","  
,"user_password":"","  
,"user_gender":" "  
,"user_email":" "}
```

响应报文：

正确 状态码201data

```
{"user_name":"","  
,"user_password":"","
```

```
,"user_gender":" "  
,"user_email":" "}
```

错误则返回相应的状态码及错误原因：

422：创建用户名时，发生验证错误，用户名已经存

在

3. 个人人信息管理理

3.1. 请求（GET / datamanagers / {name}.

GET请求报文：

无

响应报文：

正确 状态码200data为

```
{"user_name":" "  
,"user_password":" "  
,"user_gender":" "  
,"user_email"=" "}
```

错误

404数据库没有用户想要的信息记录

3.2. PATCH / datamanagers / {name}

请求报文：若不需要更改某一项更改为空即可

```
{"user_name":" "  
,"user_password":" "  
,"user_gender":" "  
,"user_email"=" "}
```

响应报文：

成功 状态码201data为

```
{"user_name":" "  
,"user_password":" "  
,"user_gender":" "  
,"user_email"=" "}
```

失败则返回错误的状态码及错误原因

422：更改信息时，发生验证错误，用户名或者密码数、性别不符合存储要求；

文档存储数据库访问接口口：

4.工程管理理：

1. 新建工程：POST / projectsetting

post请求报文:

```
{
  "project_pname"=" "
  ,"project_establisher"=" "
  ,"project_describe"=" "
}
```

响应报文:

成功 状态码201data为

```
{
  "pid"=" "
  ,"project_pname"=" "
  ,"project_establisher"=" "
  ,"project_describe"=" "
}
```

失败则返回错误的状态码及错误原因

422: 创建项目时, 发生验证错误, 项目名称等不符合要求;

2. 查看某一工程配置: GET/ projectsetting /{pid}

get请求报文: 无

响应报文

成功 状态码200data为

```
{"project_pname"=" "
,"project_establisher"=" "
,"project_describe"=" "
,"content":[{"project_member"=" "
project_authority=" "
,"project_describe"=" "}]....}]}
```

失败则返回错误的状态码及错误原因:

404数据库没有用户想要的信息记录

403: 权限不足

3. 工程配置更改PATCH/projectsettings/{pid}

patch请求报文: 若不需要更改某一项更改为空即可

```
{"project_pname"=" "
,"project_establisher"=" "
,"project_describe"=" "
,"content":[{"project_member"=" "
project_authority=" "
,"project_describe"=" "}]....}]}
```

响应报文

成功 状态码201data为

```
{ "project_pname"=" "
  , "project_establisher"=" "
  , "project_describe"=" "
  , "content":[{ "project_member"=" "
    , project_authority=" "
    , "project_describe"=" " }....]}
```

失败则返回错误的状态码及错误原因

404: 用户发出的请求针对的记录不存在;

422: 更改项目配置时, 发生验证错误, 所修改信息不符合要求;

403: 权限不足

4. 4查看用户参与的所有工程: GET / projects /

get请求报文: 无

响应报文: 状态码200data为

```
[{ "project_member"=" "
  , project_authority=" "
  , "project_describe"=" " }....]
```

失败则返回错误的状态码及错误原因:

5文档管理理:

1. 查看文档: GET / filemanagers / {fname} / file / {path}

get请求报文: 无

GET响应报文:

成功data为

```
{ "file_text":" " }
```

失败则返回错误的状态码及错误原因:

404数据库没有用户想要的信息记录

403: 权限不足

2. 修改文档: PUT / filemanagers / {fname} / file / {path}

PUT请求报文:

```
{ "file_text":" " }
```

响应报文:

成功 状态码201data为空

失败则返回错误的状态码及错误原因:

403用户得到授权, 但是没有权限修改;

404用户请求修改的文档不存在；

3. 新建文档：POST / filemanagers / {fname} / file / {path}

POST请求报文：

```
{ "file_node"=" " "
  , "file_fname"=" " "
  , "file_parentnode"=" " "
  , "file_property"=" " "
  , "file_text"=" " " }
```

响应报文：

成功：状态码201data为

```
{ "file_node"=" " "
  , "file_fname"=" " "
  , "file_parentnode"=" " "
  , "file_property"=" " "
  , "file_text"=" " " }
```

失败则返回错误的状态码及错误原因：

403：用户得到授权，但是无权限新建

422：用户创建对象时，发生验证错误，文档名不符合要求，或者文档不符合；

4. 删除文档：DELETE / filemanagers / {fname} / file / {path}

DELETE请求报文：空

响应报文：

成功状态码204及一个空文档

失败则返回错误的状态码及错误原因：

403用户得到授权，但是无权限删除文档；

404数据库没有用户想要的信息记录

6目录管理理：

1. 目录查看：GET / menus / {filenode} / filepaths / {path}

GET请求报文：无

响应报文：

成功则返回状态码200 data为

```
[{"file_node":" ", "file_property":" " },....]
```

失败则返回错误的状态码及错误原因：

404数据库没有用户想要的信息记录

2.目录修改：PATCH / menus / {filenode} / filepaths / {path}

PATCH请求报文：不需要修改则为空

```
{"menu_text=" "}
```

响应报文：

成功则返回状态码201 data为

```
{"menu_text=" "}
```

失败则返回错误的状态码及错误原因：

404：用户请求的记录不存在；

403：用户得到授权，但是无权限修改目录；

数据库设计 李书瀚

接口口设计 李书瀚 祝晶晶 李文旭

总体修改 李文旭