Tongji University club Management Center

name	matriculation numbers
谢宇翔	1951708
刘洪博	1952350
蒙俊杰	1952292

Contact Number：13217617695

Email Address: 1487857453@qq.com

1.Functionalities

User

- User sign up.
- User login.
- User logout.
- Check whether user is login.
- Find all Users.
- Find user by user_id.
- Find all users by name.(Include all users who have the same name)
- Change User's password.
- Change User's Information.(such as head_portrait, name).

Club

- Find all clubs.
- Find club by club's name.
- Find club by club's club_id.
- Find a club's all members.
- Find user's level in a specific club.(level=0 means member, level=1 means the administrator, level=2 means the head of the club)
- Delete member from club.(only the head of the club or the administrator can use this function)
- Add member to a club.(without applying)
- Change a member's level of a club.(only the head of the club can use this function)
- User apply to a club.(create an application)
- Find a club's applications.(only the head of the club or the administrator can use this function)
- Delete an application, permit this user to join club or refuse him.(only the head of the club or the administrator can use this function)
- Judge whether a user has already applied a club.
- Judge whether a user is a member of a club.
- Create a club.
- Delete a club.

Activity

- Find all activities.
- Find which club a activity belongs to.
- Find all activities which a specific user participate in.
- Judge whether a user is a member of a club which hosts a specific activity.
- Find all activities by name.
- User participate in an activity.
- Judge whether a user has already participated in a specific activity.
- Add an activity.(Only club can use this function)
- Delete an activity.

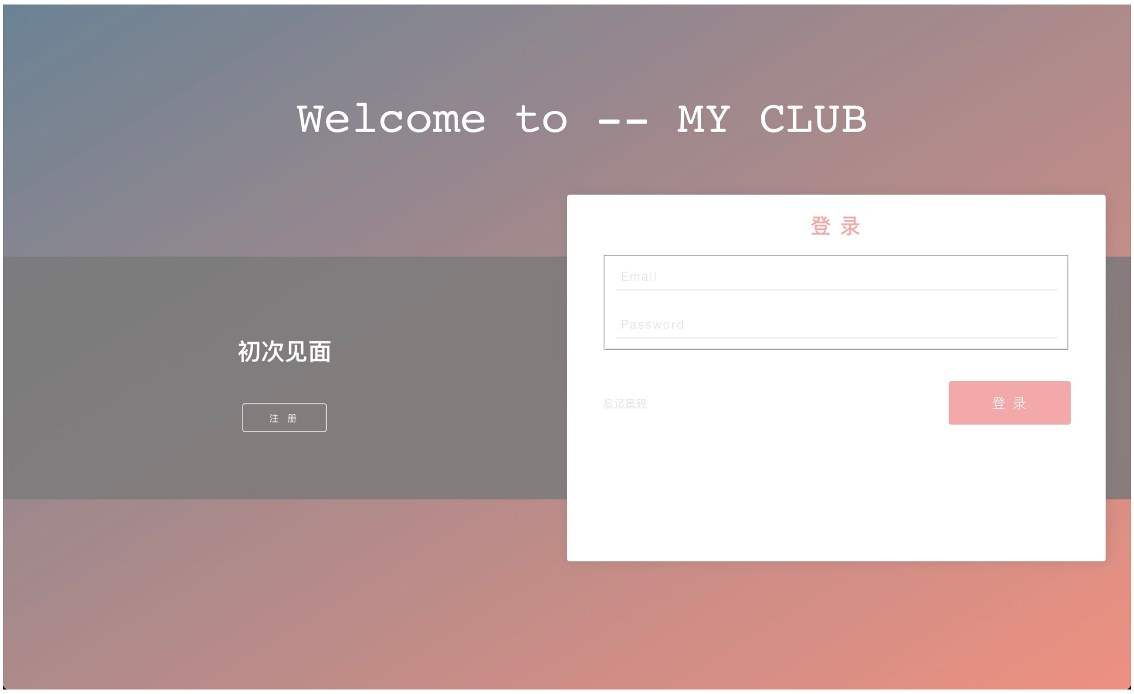
E_mail

- Send verification code to the specified email address.
- Send registration success message to the specified email address.

2.User Manual

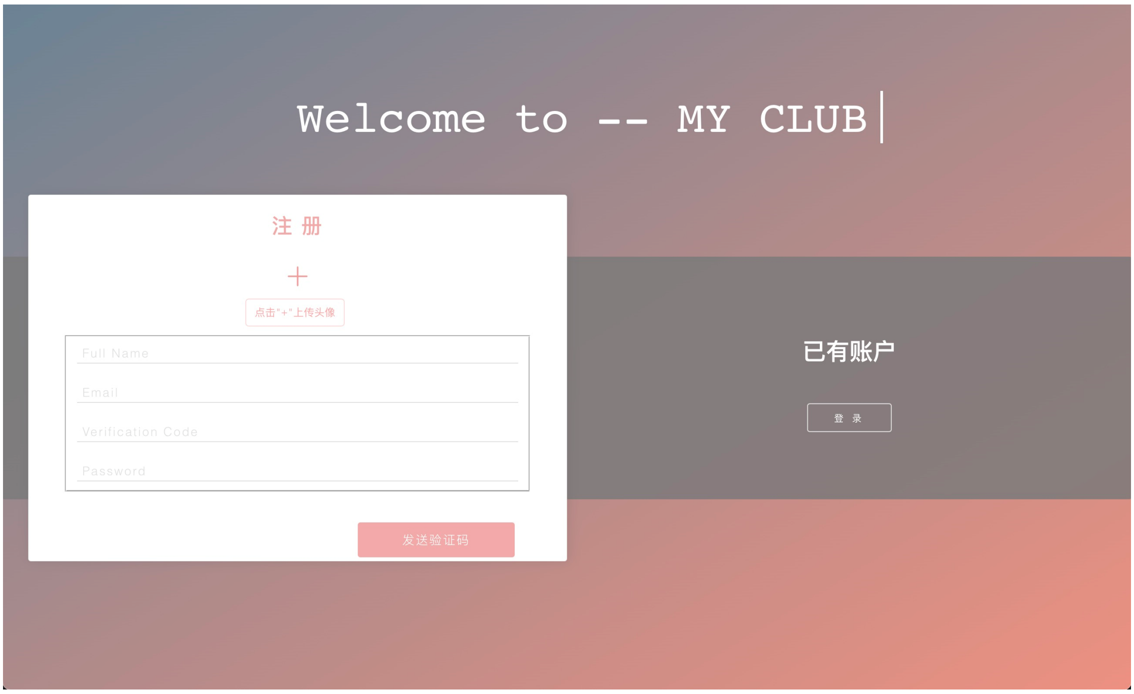
2.1 Page Introduction

Login page

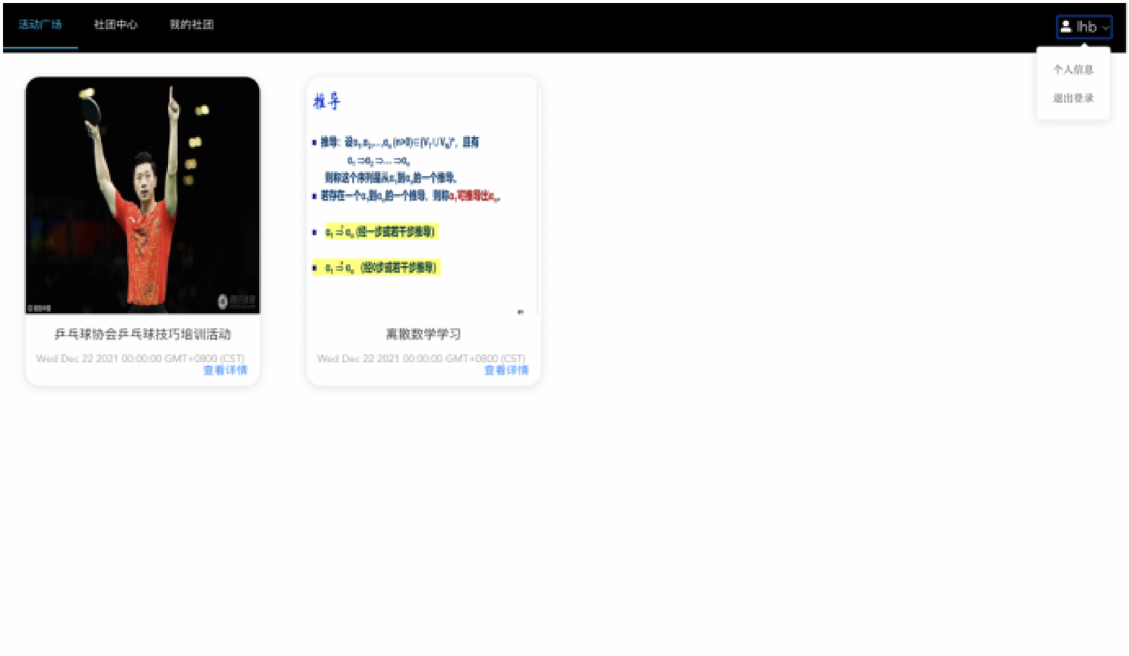
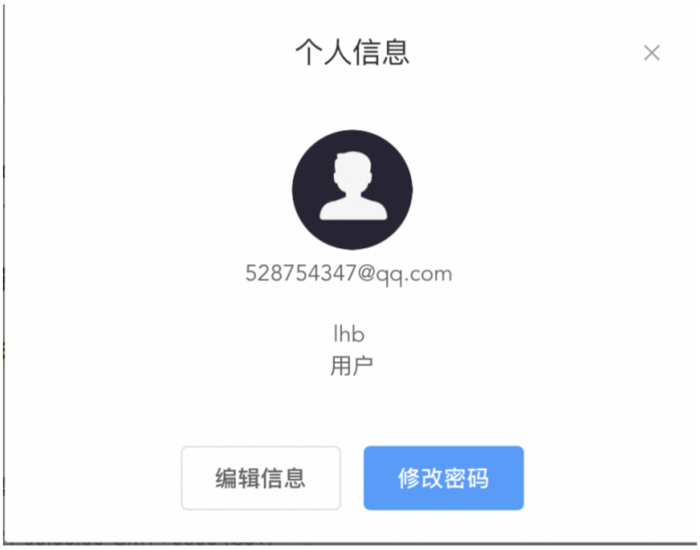
- Click login to enter the system desktop interface
- Click the ‘forgetpassword’ button if you forget your password
- For all screens, you can logout safely by clicking the exit button which is in the upper right corner.
- 

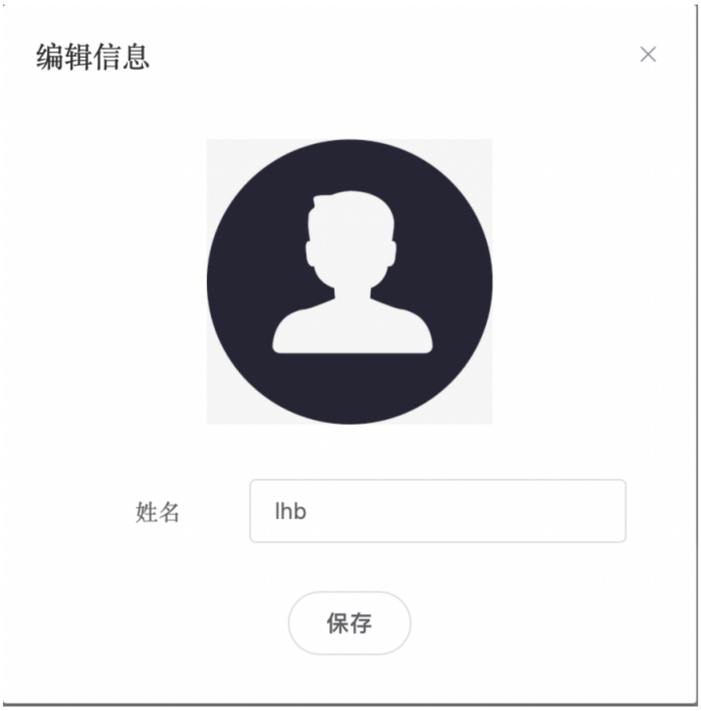
Signup page

- Click the ‘signup’ button if you don’t have an account
- Click the ‘+’ button to upload your head portrait if you want
- Click the ‘send verification code’ button after you input all the information
- Input the verification code and click ‘sign up’ button,then you will have a brand new account

- 

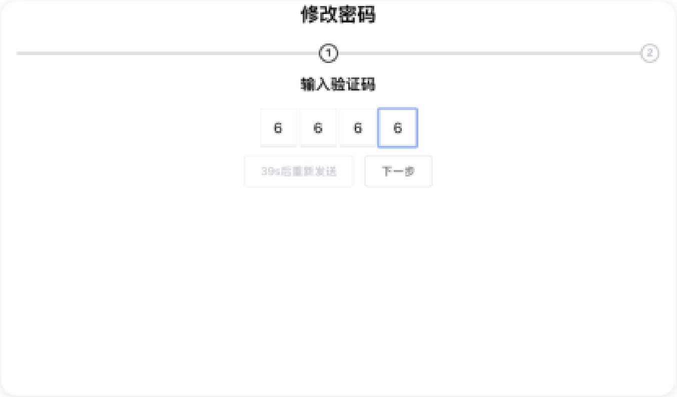
Desktop page

- Click the ‘personal’ information button to see and change your information
- Click the ‘logout’ button to logout safely
- 
活动广场
社团中心
我的社团
lhb
个人信息
退出登录
推导
乒乓球协会乒乓球技巧培训活动
Wed Dec 22 2021 00:00:00 GMT+0800 (CST)
查看详情
离散数学学习
Wed Dec 22 2021 00:00:00 GMT+0800 (CST)
查看详情
- 

- 

  编辑信息

  姓名 lhb

- 

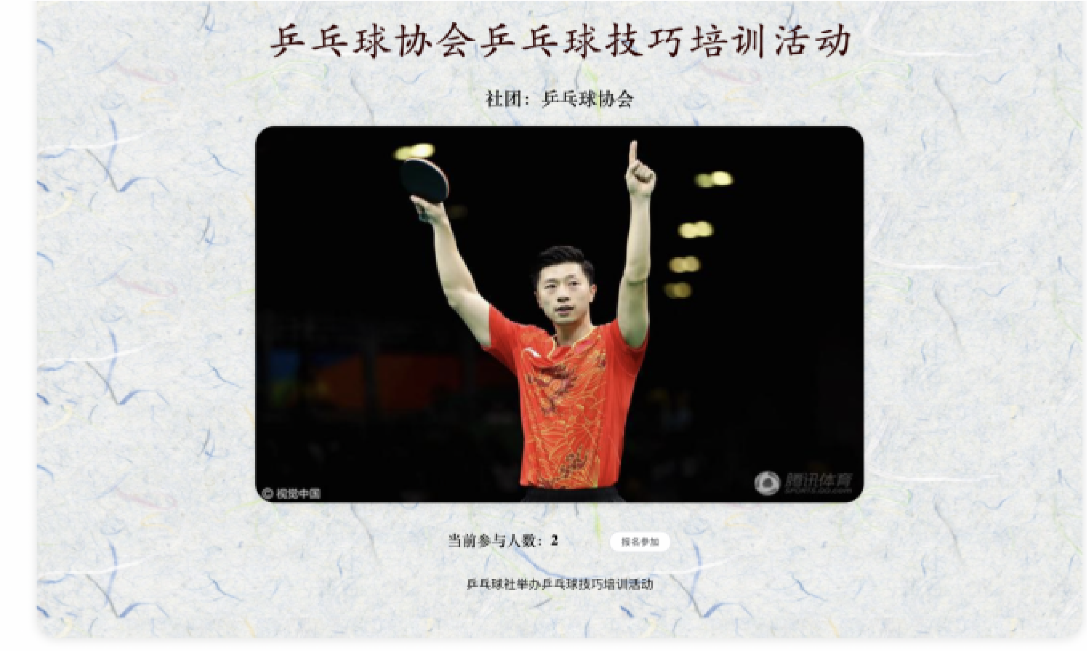
  修改密码

  输入验证码

  6 6 6 6

Activities Center Page

- Click ‘see detailed information’ button to see more information about an activity
- You can join every activity which is showed on this page(well, if you already join the activity, there will not be a ‘join’ button)

- 

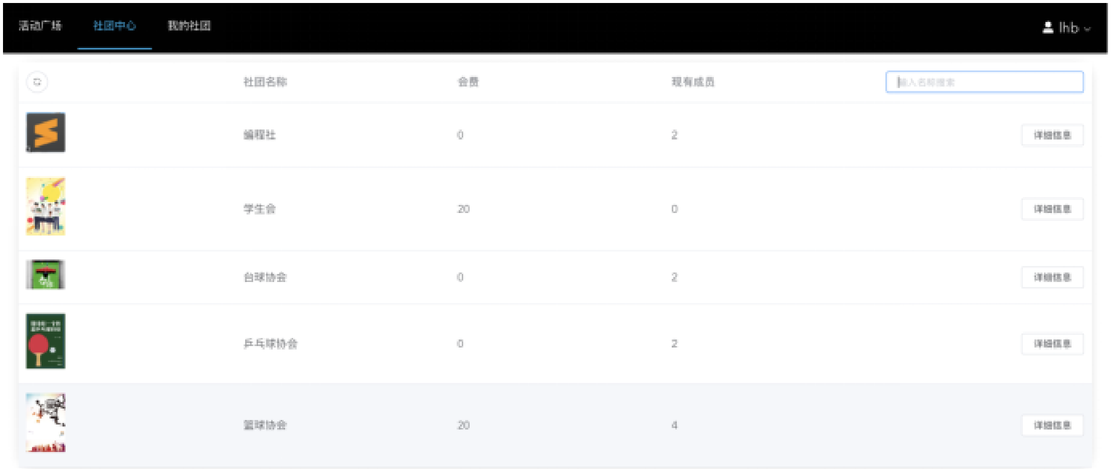
乒乓球协会乒乓球技巧培训活动

社团：乒乓球协会

当前参与人数：2

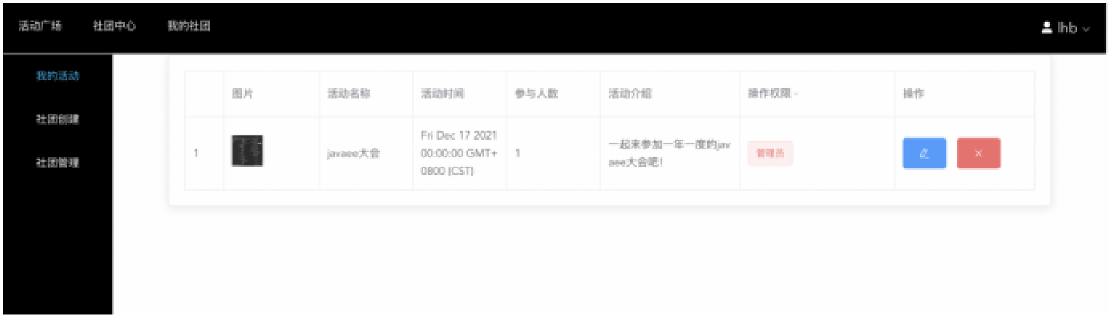
乒乓球社举办乒乓球技巧培训活动

Club Center Page

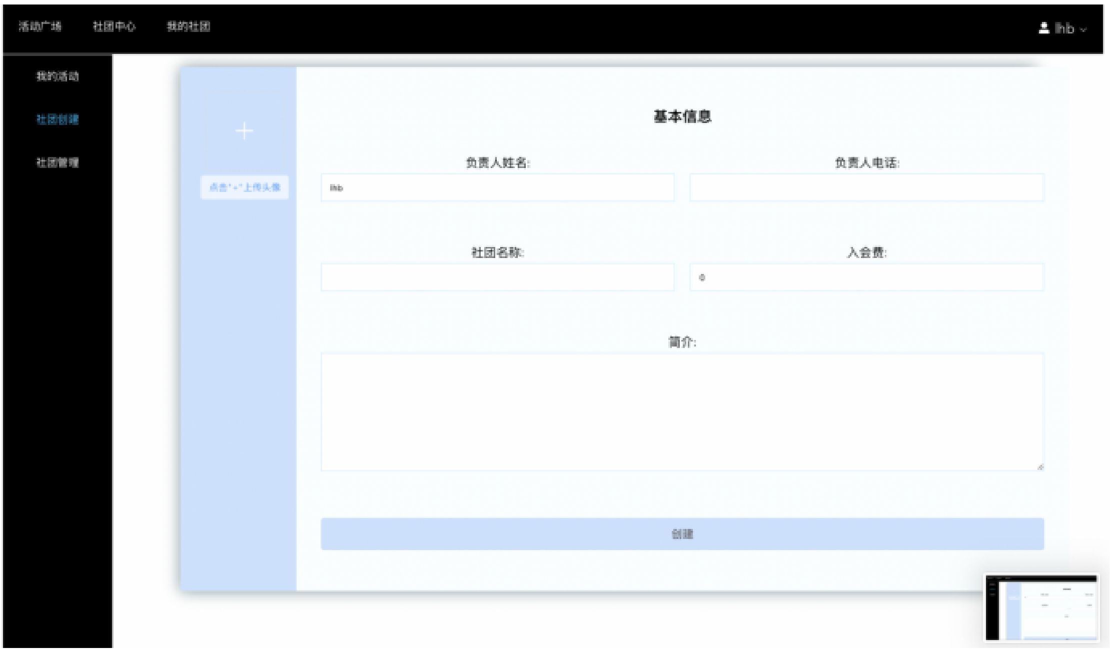
- Click the ‘reorder’ button in the upper left corner to reorder the clubs
- Click the ‘detailed’ button to see the detailed information of the club you choose
- To send a join application you can click the ‘join’ button(well,if you are already in this club,there will be no ‘join’ button)
- 

	社团名称	会费	现有成员	
	编程社	0	2	详细信息
	学生会	20	0	详细信息
	台球协会	0	2	详细信息
	乒乓球协会	0	2	详细信息
	篮球协会	20	4	详细信息

My Clubs Center

- In my activities page,you can see all the activities you joined,and edit the information of the activities if you have sufficient authority
- In club creating page, you can apply for creating your club
- In club managing page,you can manage all your clubs which you joined or created
- 

	图片	活动名称	活动时间	参与人数	活动介绍	操作权限	操作
1		javaee大会	Fri Dec 17 2021 00:00:00 GMT+0800 (CST)	1	一起来参加一年一度的javaee大会吧!	管理员	✎ ✕

- 

  基本信息

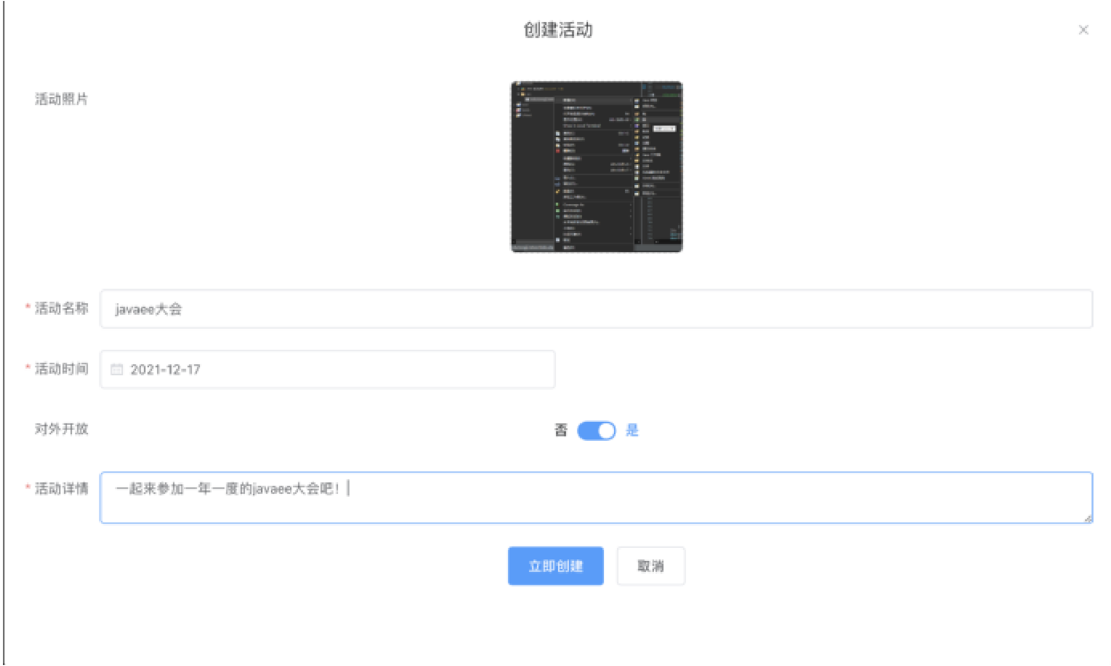
  负责人姓名: lhb

  负责人电话:

  社团名称:

  入会费: 0

  简介:

- 
  创建活动

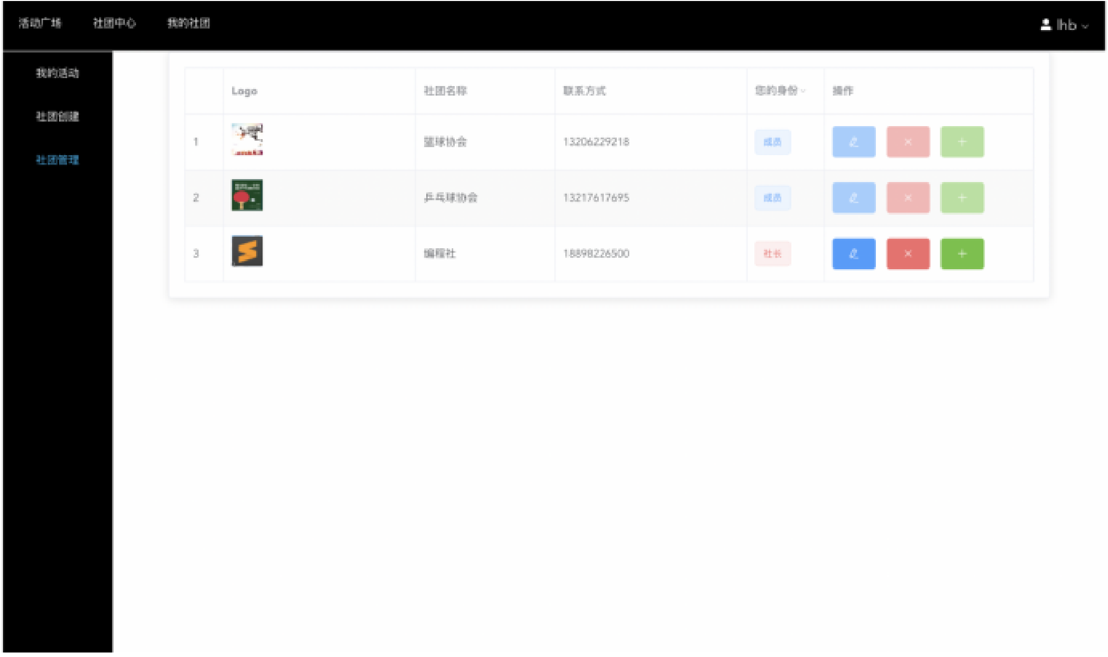
  活动照片

  * 活动名称 javaee大会

  * 活动时间 2021-12-17

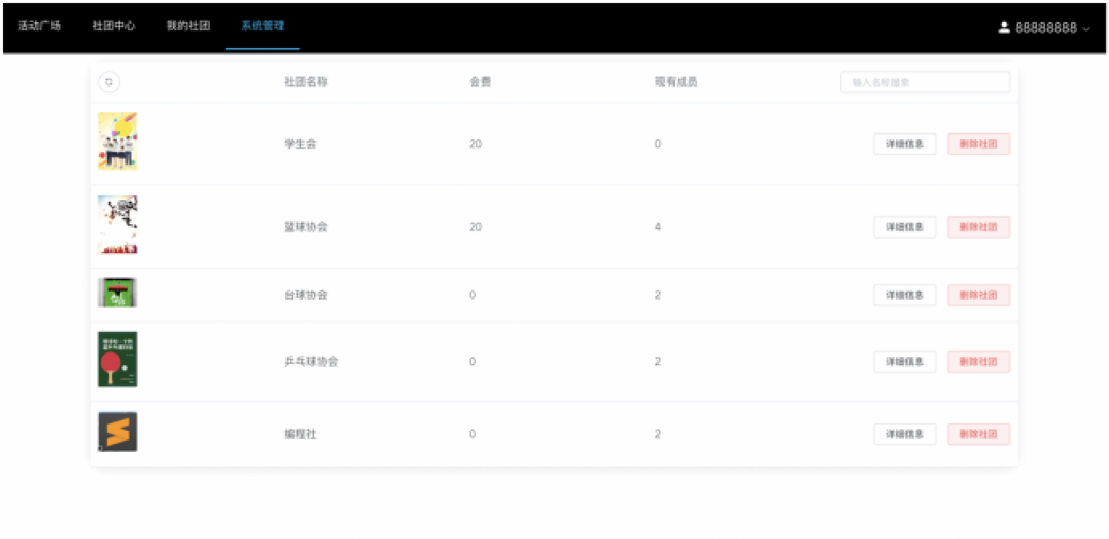
  对外开放 否 ● 是

  * 活动详情 一起来参加一年一度的javaee大会吧！

- 

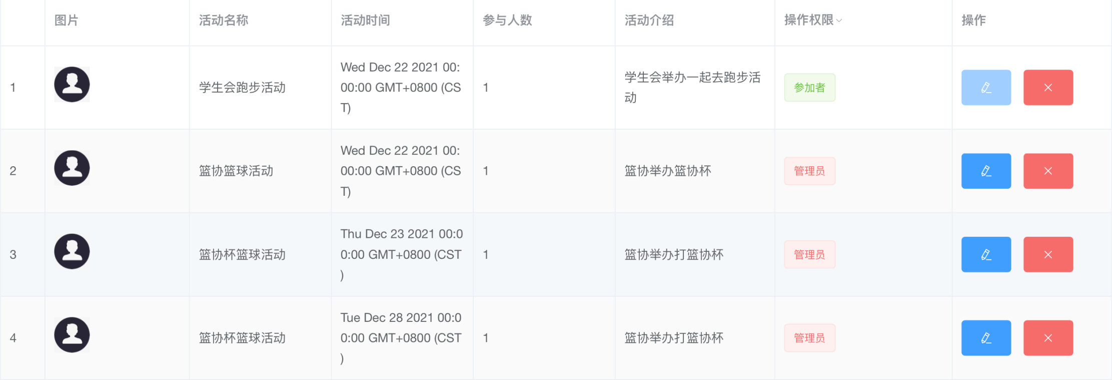
	Logo	社团名称	联系方式	您的身份	操作
1		篮球协会	13206229218	成员	✎ × +
2		乒乓球协会	13217617695	成员	✎ × +
3		编程社	18898226500	社长	✎ × +

System Management Center

- If you are the admin user of the System,you can see the System Management Center.
- In this page, you can manage all the clubs
- 

	社团名称	会费	现有成员	
	学生会	20	0	详细信息 删除社团
	篮球协会	20	4	详细信息 删除社团
	台球协会	0	2	详细信息 删除社团
	乒乓球协会	0	2	详细信息 删除社团
	编程社	0	2	详细信息 删除社团

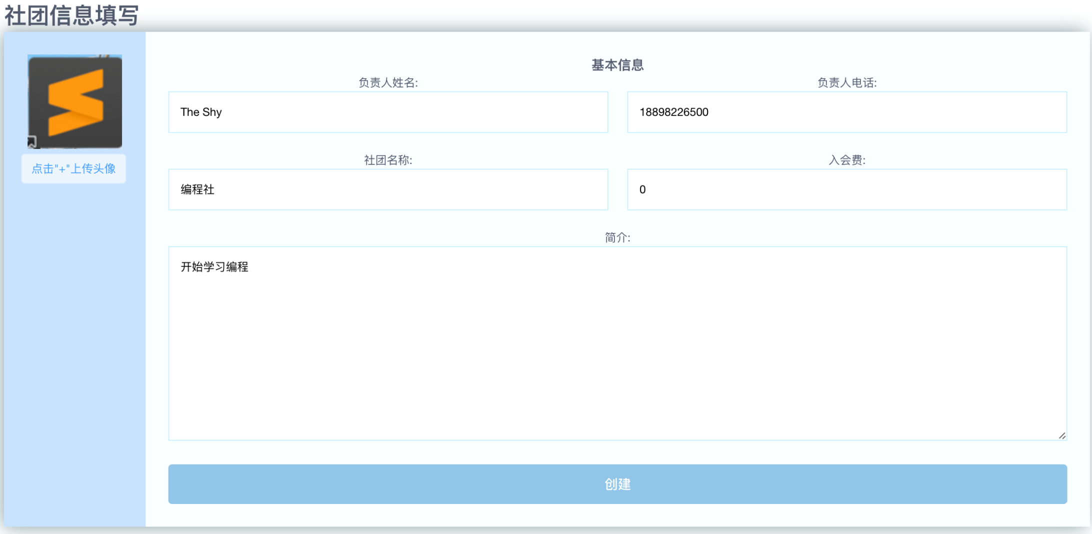
View my activity page

- This page can view all the activities I participate in and the operation rights of these activities. If you are a participant, you can only choose to quit the activity. If you are an administrator, you can choose to delete the activity or modify the activity information in addition to quitting the activity.
- Click the Exit activity in the action to remove the activity from its participation table. The administrator can view the details of the activity and modify the activity information by clicking Modify. Participants cannot modify the activity information.
- You can filter out activities that correspond to permissions.
- 

	图片	活动名称	活动时间	参与人数	活动介绍	操作权限	操作
1		学生会跑步活动	Wed Dec 22 2021 00:00:00 GMT+0800 (CST)	1	学生会举办一起去跑步活动	参加者	✎ ×
2		篮协篮球活动	Wed Dec 22 2021 00:00:00 GMT+0800 (CST)	1	篮协举办篮协杯	管理员	✎ ×
3		篮协杯篮球活动	Thu Dec 23 2021 00:00:00 GMT+0800 (CST)	1	篮协举办打篮协杯	管理员	✎ ×
4		篮协杯篮球活动	Tue Dec 28 2021 00:00:00 GMT+0800 (CST)	1	篮协举办打篮协杯	管理员	✎ ×

Creating club page

- This page can create a new club, need to upload a club profile picture, need to fill in the person in charge of the phone, the person in charge of the name, the name of the club, membership fee, introduction.
- The name of the person in charge is himself and cannot be modified. The telephone number of the person in charge and the name of the association are mandatory.

- 
  社团信息填写

  基本信息

  负责人姓名: The Shy

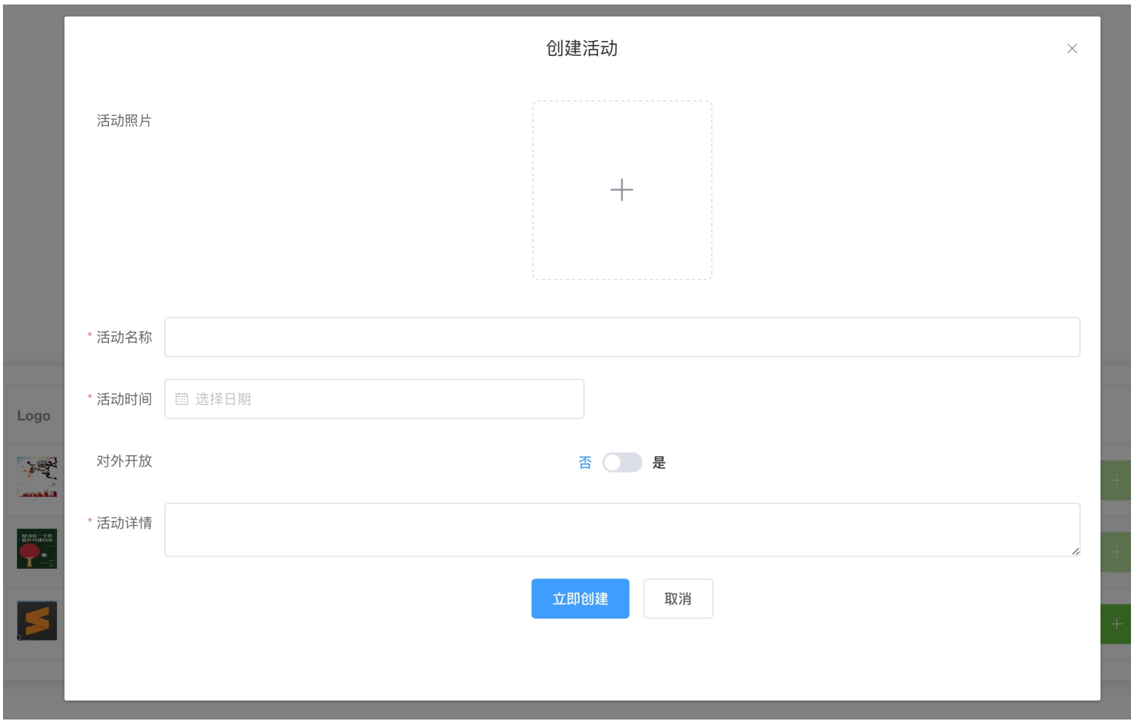
  负责人电话: 18898226500

  社团名称: 编程社

  入会费: 0

  简介: 开始学习编程

My club page

- On this page, you can view all the associations you have joined and display your identity in the associations. You can complete different operations according to your identity permissions.
- Members cannot manage club members, delete club members or add activities (members who quit club need to contact the president/administrator to remove themselves); Administrators can not delete the club, can manage club members, add activities;
- The president has all permissions;
- Those who have the right to click Add Activity can fill in the details of the activity and add an activity of the club; Click Delete club to delete the club; Click Management to view the association management interface.
- You can screen for different identities
- 
  创建活动

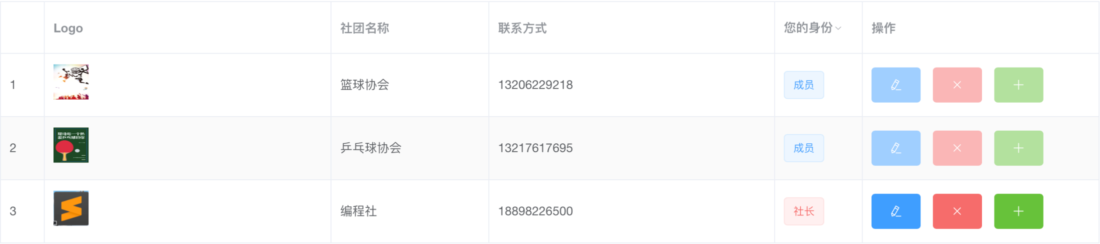
  活动照片

  * 活动名称

  * 活动时间 选择日期

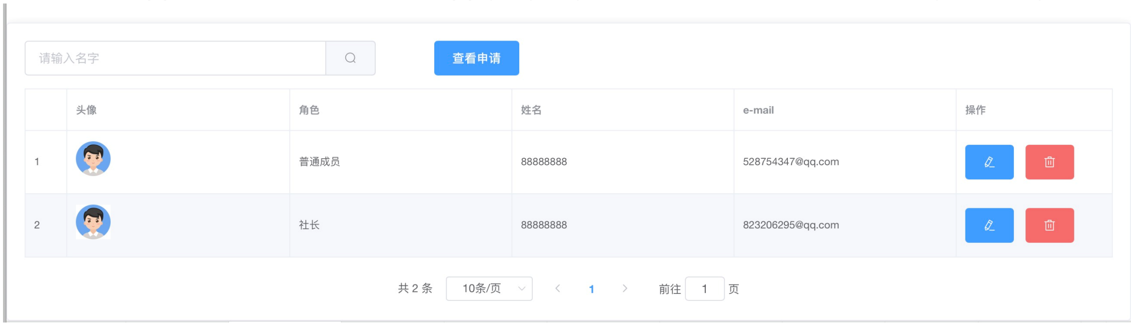
  对外开放 否 ○ 是

  * 活动详情

- 

	Logo	社团名称	联系方式	您的身份	操作
1		篮球协会	13206229218	成员	✎ × +
2		乒乓球协会	13217617695	成员	✎ × +
3		编程社	18898226500	社长	✎ × +

Managing club Interface

- This page provides an interface for the club president and club administrator to manage club members.
- Click Edit position, you can change the position of the member;
- Click kick member can remove the person out of the club;
- Click View Application to see who is applying to join the club and choose to reject or agree.
- 

	头像	角色	姓名	e-mail	操作
1		普通成员	88888888	528754347@qq.com	✎ 🗑
2		社长	88888888	823206295@qq.com	✎ 🗑

2.2 System Deployment and Configuration Instructions

Backend

The back-end springboot has been deployed to **Tomcat** of the personal tencentcloud server 150.158.47.16, whose port number is 521, visit http://150.158.47.16:521/API_NAME to access to the backend interface, to obtain the corresponding services.

mysql

The database has been deployed to the remote server 150.158.47.16:3306. The database can be accessed through the following user information.

- username: javaee
- password: 813201
- url : jdbc:mysql://150.158.47.16:3306/javaee

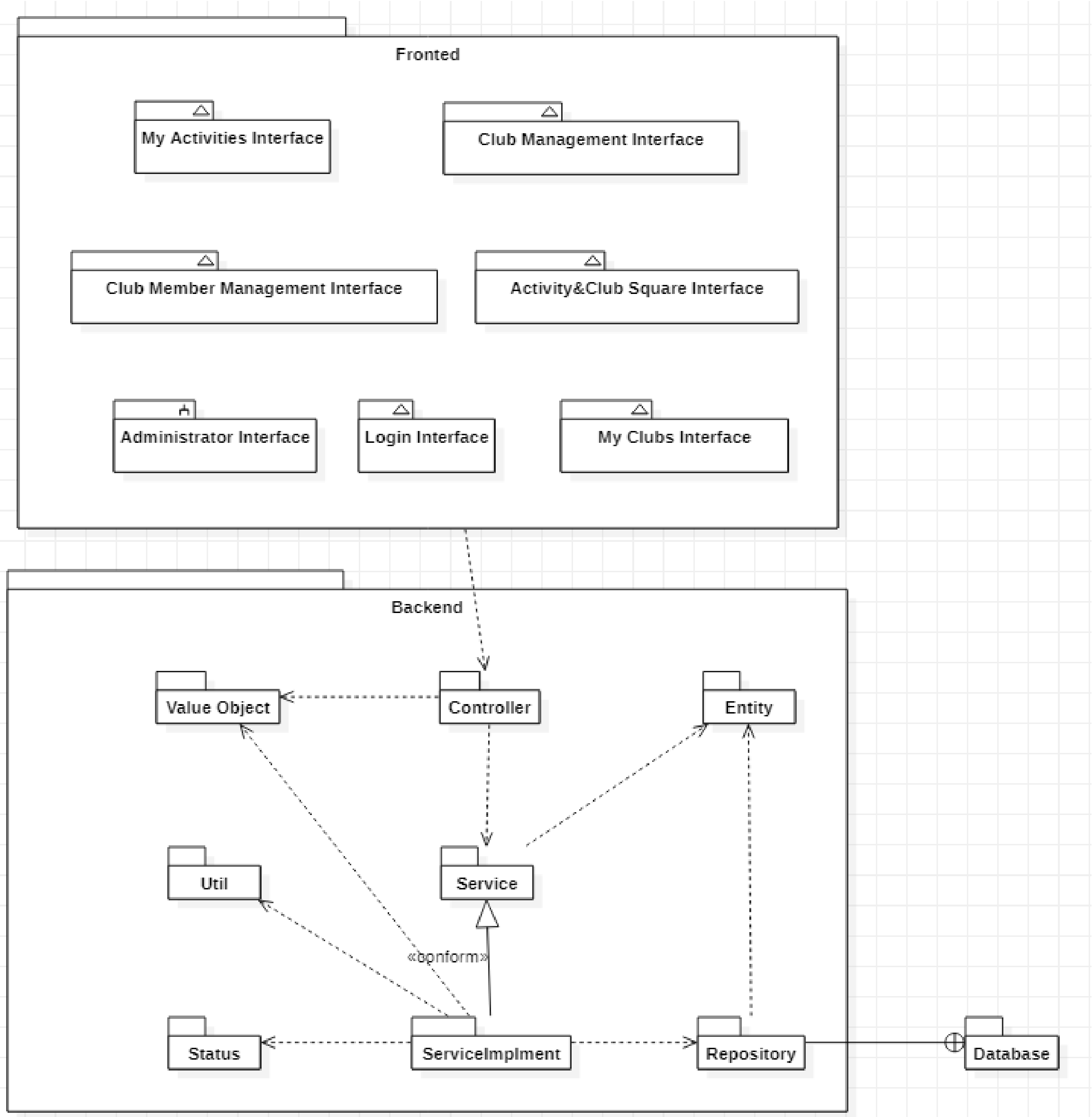
mail

- default-encoding: utf-8
- host: smtp.163.com
- username: tang813201@163.com
- password: MFXTSGCTXSHWUSOC
- protocol: smtp

3. System Architecture and Component Design

3.1Architecture Design

This project adopts the architecture of strict separation of frontend and backend, with the development of the frontend using **Vue** and the development of the back-end using **Springboot**.



- **Frontend**

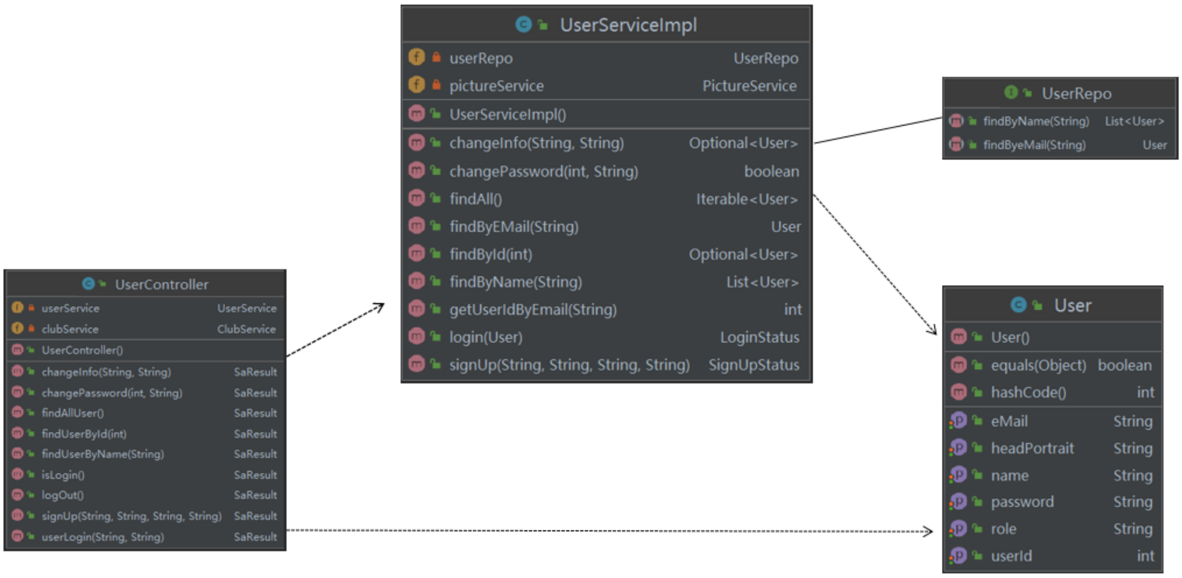
  The front end is a Web application with seven pages, including login interface, my clubs interface, my activities
  interface, club management interface, club member management interface, activity&club square interface, administrator interface.

- **Backend**

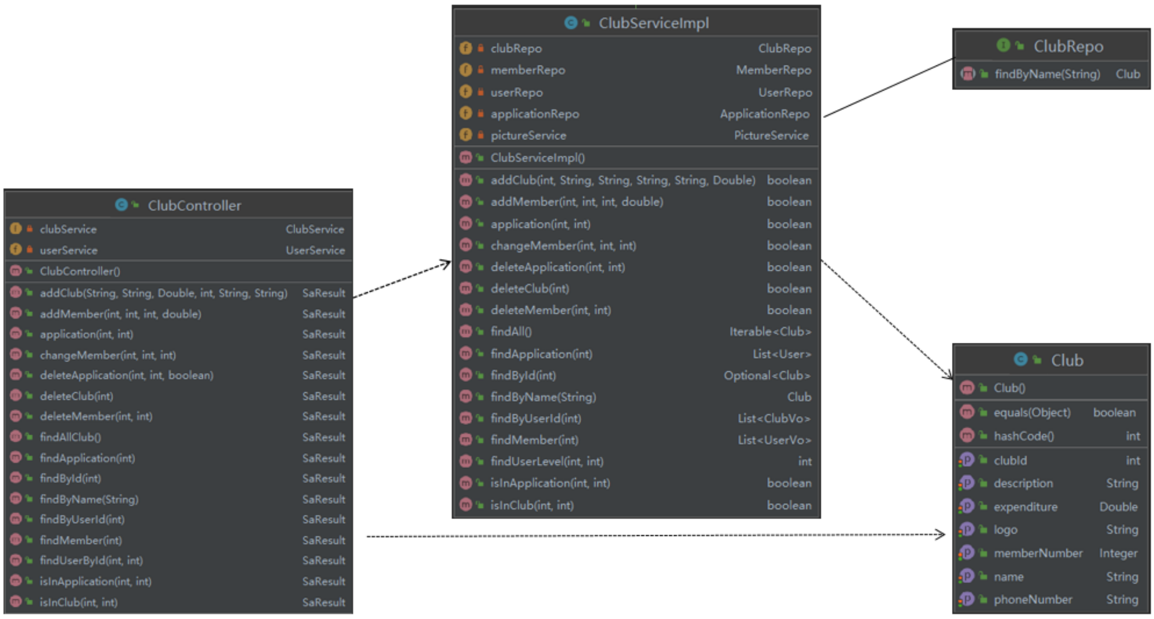
  The back end adopts a hierarchical architecture, divided into Entity layer, Service layer, Repository layer and Controller layer. The **Controller layer** interacts directly with the front end to respond to the needs of the front end users, here we use @RequestMapping and the value in @PostMapping and @GetMapping in the control layer as the URL path for access. Controllers will invoke the **Service** to implement the business logic, each service is an interface, and there is an implementation class for the services that integrate the Service with the **Repository**. We use **Hibernate** to manage persistence layer and map the database into entities and create **Entity layer**. Then, we create repositories for every entities, **Repository layer** can help us easily operate database by using Encapsulated SQL.

3.2 Component Design

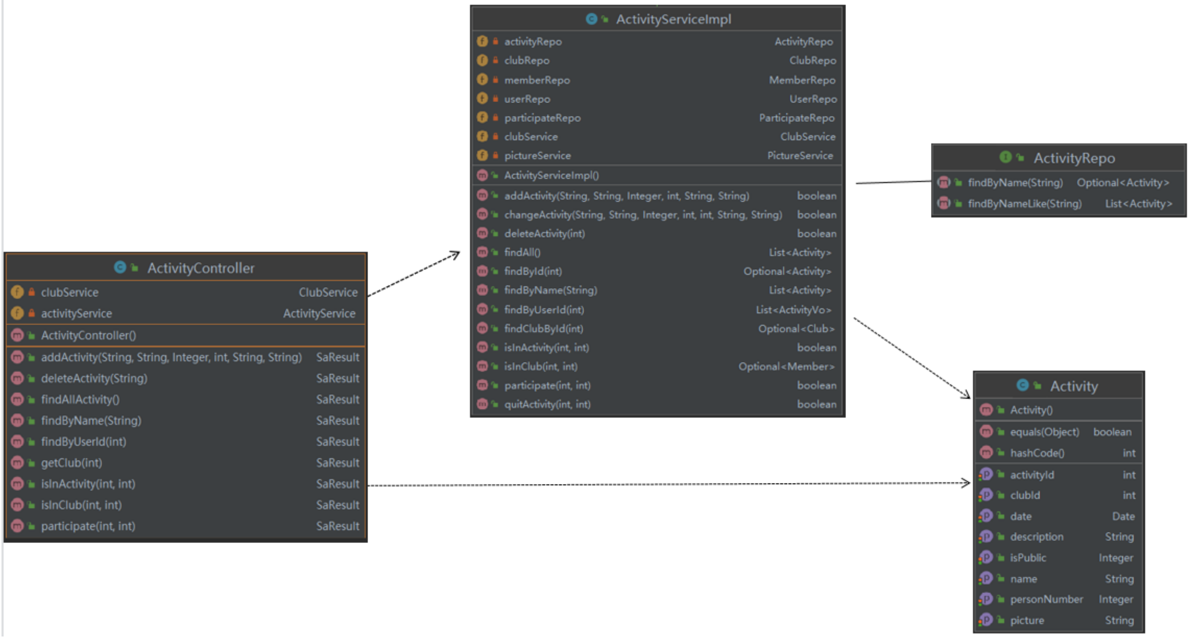
User


UserServiceImpl
userRepo UserRepo
pictureService PictureService
UserServiceImpl()
changeInfo(String, String) Optional<User>
changePassword(int, String) boolean
findAll() Iterable<User>
findByEMail(String) User
findById(int) Optional<User>
findByName(String) List<User>
getUserIdByEmail(String) int
login(User) LoginStatus
signUp(String, String, String, String) SignUpStatus
UserRepo
findByName(String) List<User>
findByeMail(String) User
UserController
userService UserService
clubService ClubService
UserController()
changeInfo(String, String) SaResult
changePassword(int, String) SaResult
findAllUser() SaResult
findUserById(int) SaResult
findUserByName(String) SaResult
isLogin() SaResult
logOut() SaResult
signUp(String, String, String, String) SaResult
userLogin(String, String) SaResult
User
User()
equals(Object) boolean
hashCode() int
eMail String
headPortrait String
name String
password String
role String
userId int

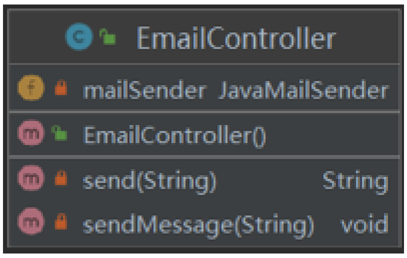
Club


ClubServiceImpl
clubRepo ClubRepo
memberRepo MemberRepo
userRepo UserRepo
applicationRepo ApplicationRepo
pictureService PictureService
ClubServiceImpl()
addClub(int, String, String, String, String, Double) boolean
addMember(int, int, int, double) boolean
application(int, int) boolean
changeMember(int, int, int) boolean
deleteApplication(int, int) boolean
deleteClub(int) boolean
deleteMember(int, int) boolean
findAll() Iterable<Club>
findApplication(int) List<User>
findById(int) Optional<Club>
findByName(String) Club
findByUserId(int) List<ClubVo>
findMember(int) List<UserVo>
findUserLevel(int, int) int
isInApplication(int, int) boolean
isInClub(int, int) boolean
ClubRepo
findByName(String) Club
ClubController
clubService ClubService
userService UserService
ClubController()
addClub(String, String, Double, int, String, String) SaResult
addMember(int, int, int, double) SaResult
application(int, int) SaResult
changeMember(int, int, int) SaResult
deleteApplication(int, int, boolean) SaResult
deleteClub(int) SaResult
deleteMember(int, int) SaResult
findAllClub() SaResult
findApplication(int) SaResult
findById(int) SaResult
findByName(String) SaResult
findByUserId(int) SaResult
findMember(int) SaResult
findUserById(int, int) SaResult
isInApplication(int, int) SaResult
isInClub(int, int) SaResult
Club
Club()
equals(Object) boolean
hashCode() int
clubId int
description String
expenditure Double
logo String
memberNumber Integer
name String
phoneNumber String

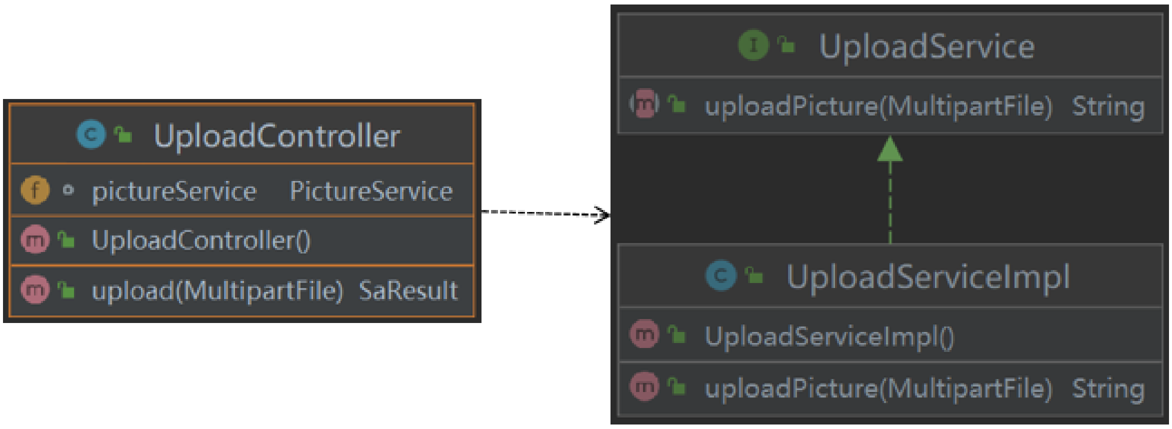
Activity


ActivityServiceImpl
activityRepo ActivityRepo
clubRepo ClubRepo
memberRepo MemberRepo
userRepo UserRepo
participateRepo ParticipateRepo
clubService ClubService
pictureService PictureService
ActivityServiceImpl()
addActivity(String, String, Integer, int, String, String) boolean
changeActivity(String, String, Integer, int, int, String, String) boolean
deleteActivity(int) boolean
findAll() List<Activity>
findById(int) Optional<Activity>
findByName(String) List<Activity>
findByUserId(int) List<ActivityVo>
findClubById(int) Optional<Club>
isInActivity(int, int) boolean
isInClub(int, int) Optional<Member>
participate(int, int) boolean
quitActivity(int, int) boolean
ActivityRepo
findByName(String) Optional<Activity>
findByNameLike(String) List<Activity>
ActivityController
clubService ClubService
activityService ActivityService
ActivityController()
addActivity(String, String, Integer, int, String, String) SaResult
deleteActivity(String) SaResult
findAllActivity() SaResult
findByName(String) SaResult
findByUserId(int) SaResult
getClub(int) SaResult
isInActivity(int, int) SaResult
isInClub(int, int) SaResult
participate(int, int) SaResult
Activity
Activity()
equals(Object) boolean
hashCode() int
activityId int
clubId int
date Date
description String
isPublic Integer
name String
personNumber Integer
picture String

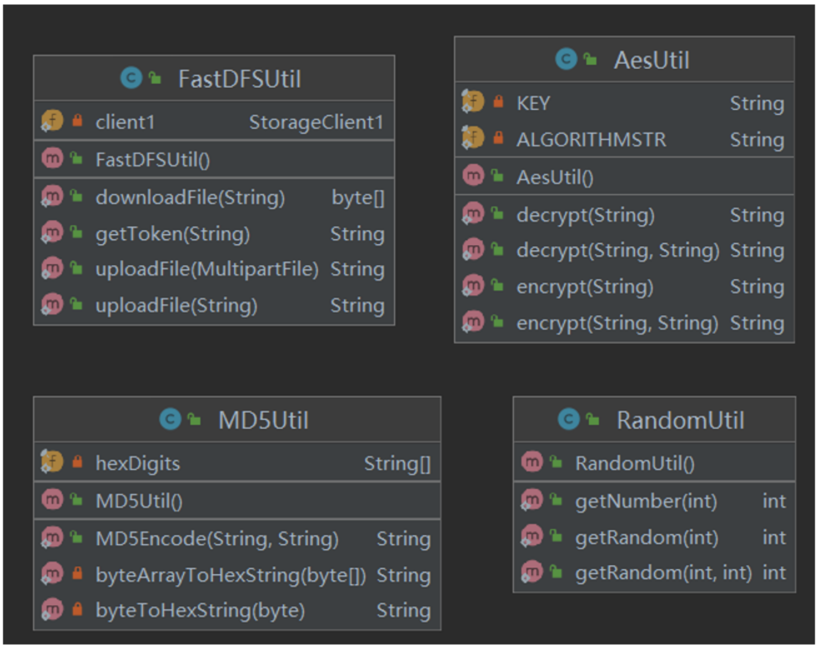
Email



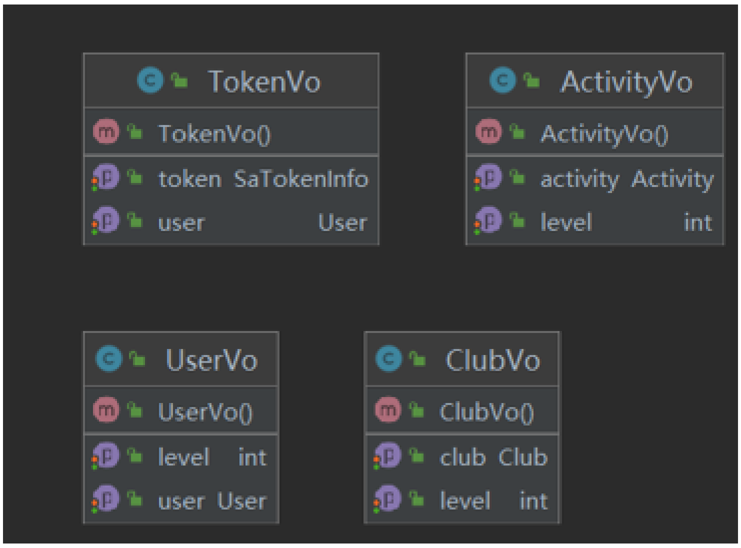
Upload


UploadService
uploadPicture(MultipartFile) String
UploadController
pictureService PictureService
UploadController()
upload(MultipartFile) SaResult
UploadServiceImpl
UploadServiceImpl()
uploadPicture(MultipartFile) String

Util



Value Object



4.Database Design

- Mysql database

4.1 User Table

```
create table user
(
    name          varchar(20)                null,
    password      varchar(200)               not null,
    head_portrait varchar(100)               null,
    e_mail        varchar(30)                not null,
    role          varchar(20) default 'user' null comment 'admin user',
    user_id       int auto_increment
        primary key,
    constraint user_e_mail_uindex
        unique (e_mail)
);
```

4.2 Activity Table

```
create table activity
(
    activity_id   int auto_increment
        primary key,
    description   varchar(200)  null,
    date          date          null,
    name          varchar(20)   null,
    is_public     int default 0 null,
    club_id       int           not null,
    picture       varchar(100)  null,
    person_number int default 0 null,
    constraint activity_club_club_id_fk
        foreign key (club_id) references club (club_id)
            on delete cascade
);
```

4.3 Club Table

```
create table club
(
    club_id       int auto_increment
        primary key,
    logo          varchar(535)     null,
    description   varchar(200)     null,
    name          varchar(20)      not null,
    member_number int    default 1 not null,
    expenditure   double default 0 null,#entrance fee
    phone_number  varchar(20)      null #header's number
);
```

4.4 Application Table

- User apply to join a club.

```
create table application
(
    user_id int default 0 not null,
    club_id int default 0 not null,
    primary key (user_id, club_id),
    constraint application_club_club_id_fk
        foreign key (club_id) references club (club_id)
            on delete cascade,
    constraint application_user_user_id_fk
        foreign key (user_id) references user (user_id)
            on delete cascade
);
```

4.5 Member Table

- Relationship table between User and Club.

```
create table member
(
    user_id   int              not null,
    club_id   int              not null,
    level     int    default 0 null, #0 means member,1 means admin,2 means
header
    club_cost double default 0 null,
    primary key (club_id, user_id),
    constraint member_club_club_ID_fk
        foreign key (club_id) references club (club_id)
            on delete cascade,
    constraint member_student_student_ID_fk
        foreign key (user_id) references user (user_id)
            on delete cascade
);
```

4.6 Participate Table

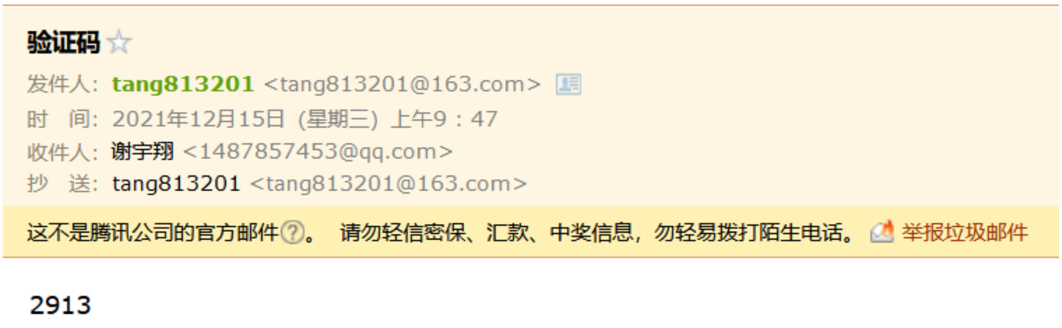
- Relationship table between User and Activity.

```
create table participate
(
    user_id     int default 0 not null,
    activity_id int default 0 not null,
    primary key (activity_id, user_id),
    constraint participate_activity_activity_id_fk
        foreign key (activity_id) references activity (activity_id)
            on delete cascade,
    constraint participate_user_user_id_fk
        foreign key (user_id) references user (user_id)
            on delete cascade
);
```

5.Innovative Design

5.1 Send E-mail system

- In our system, we aggregate send e-mail function:
  - By access **/email/sendcode** url, system will send a verification code(4 numbers) to specific email address. This function improves the security of system data.



验证码 ☆

发件人：tang813201 <tang813201@163.com>

时　间：2021年12月15日（星期三）上午9：47

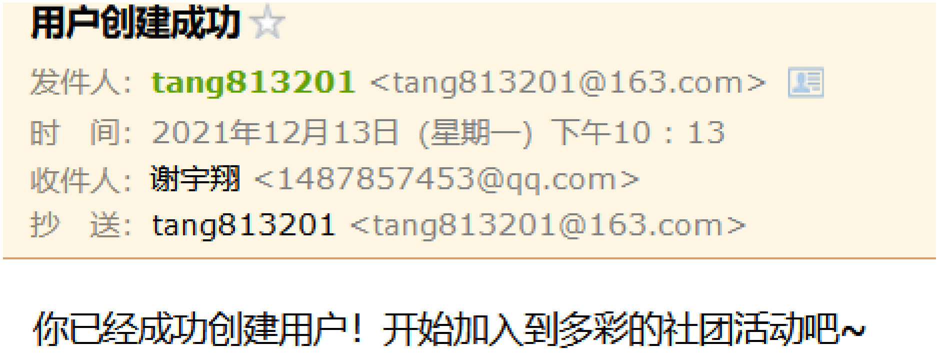
收件人：谢宇翔 <1487857453@qq.com>

抄　送：tang813201 <tang813201@163.com>

这不是腾讯公司的官方邮件？。　请勿轻信密保、汇款、中奖信息，勿轻易拨打陌生电话。　举报垃圾邮件

2913

  - By access **/email/sendmessage** url, system will send a message to user's email address when a user sign up a new user.



用户创建成功 ☆

发件人：tang813201 <tang813201@163.com>

时　间：2021年12月13日（星期一）下午10：13

收件人：谢宇翔 <1487857453@qq.com>

抄　送：tang813201 <tang813201@163.com>

你已经成功创建用户！开始加入到多彩的社团活动吧~

5.2 Upload File

- Our system use **FastDFS** util to implement upload file function.
- Our system allow user to upload a file when a user signs up or creates a activity or create a activity. Our system will upload this file into our server and return an url by which we can obtain it again.
- By using this function, we allow user upload his head portrait, club owner upload club picture, activity picture and show these pictures to front end later.

5.3 Double encryption

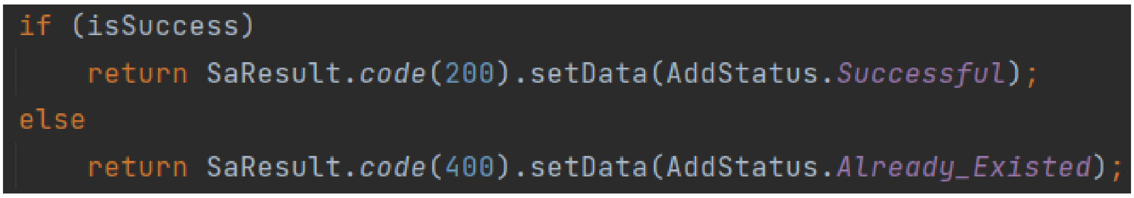
- In data security assurance, we use **AES** and **MD5** encryption to encrypt data. Details are as follow:
  - **AES** is one of the most popular symmetric encryptions. Our system set a same **KEY** which will be use in encrypt and decrypt together with content. This KEY is invisiable to any other users. So, even our encrypted code is intercepted, they can not translate our code whithout KEY.
  - **MD5** message-digest Algorithm is a widely used password hash function. We use it to encrypt user's password and save it into database, we will never decrypt it, so even the administrator could not get user's password.
  - We encrypt verification code by using **AES algorithm** in the backend and transmit it to the front end, and front end will decrypt it and check whether the code is the same as

the code entered by the user.

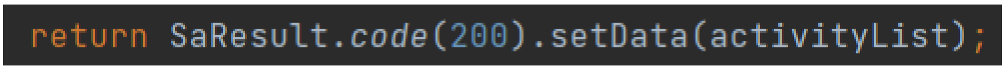
- User's password will be encrypted by using **AES** in the front end, and transmit to back end, the back end will use **AES** to decrypt password code to true password. Then back end will use **MD5** to encrypt the true password and save it into our database, it fully ensures data and system security.

5.4 Usage of SaResult

- Encapsulate the return data of all controller interfaces in SaResult, it's easy for frontend to get data and false reason.



```
if (isSuccess)
    return SaResult.code(200).setData(AddStatus.Successful);
else
    return SaResult.code(400).setData(AddStatus.Already_Existed);
```



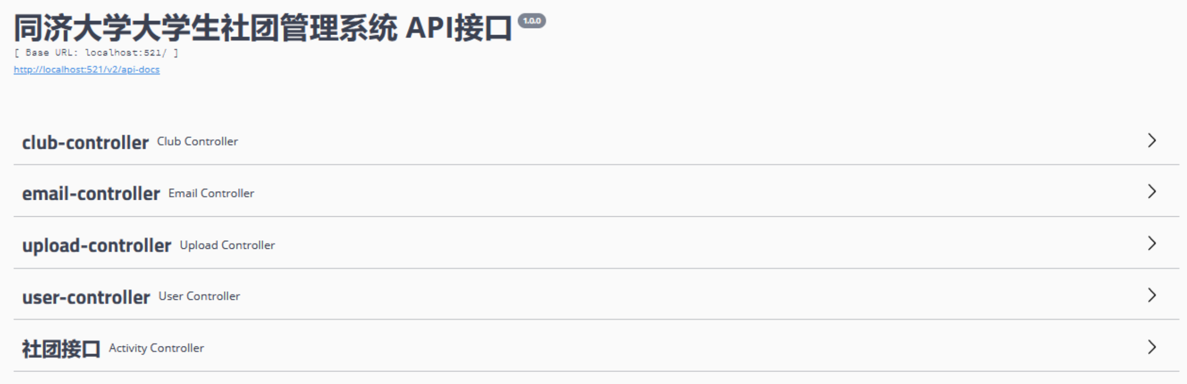
```
return SaResult.code(200).setData(activityList);
```

5.5 Usage of Satoken

- Sa-token is a lightweight Java permission authentication framework, which can be used to solve a series of permission related problems such as login authentication, permission authentication.

5.6 Usage of Swagger

- Our system uses the Swagger component to present the back-end interface.



同济大学大学生社团管理系统 API接口 1.0.0

[Base URL: localhost:521/]

http://localhost:521/v2/api-docs

club-controller Club Controller

email-controller Email Controller

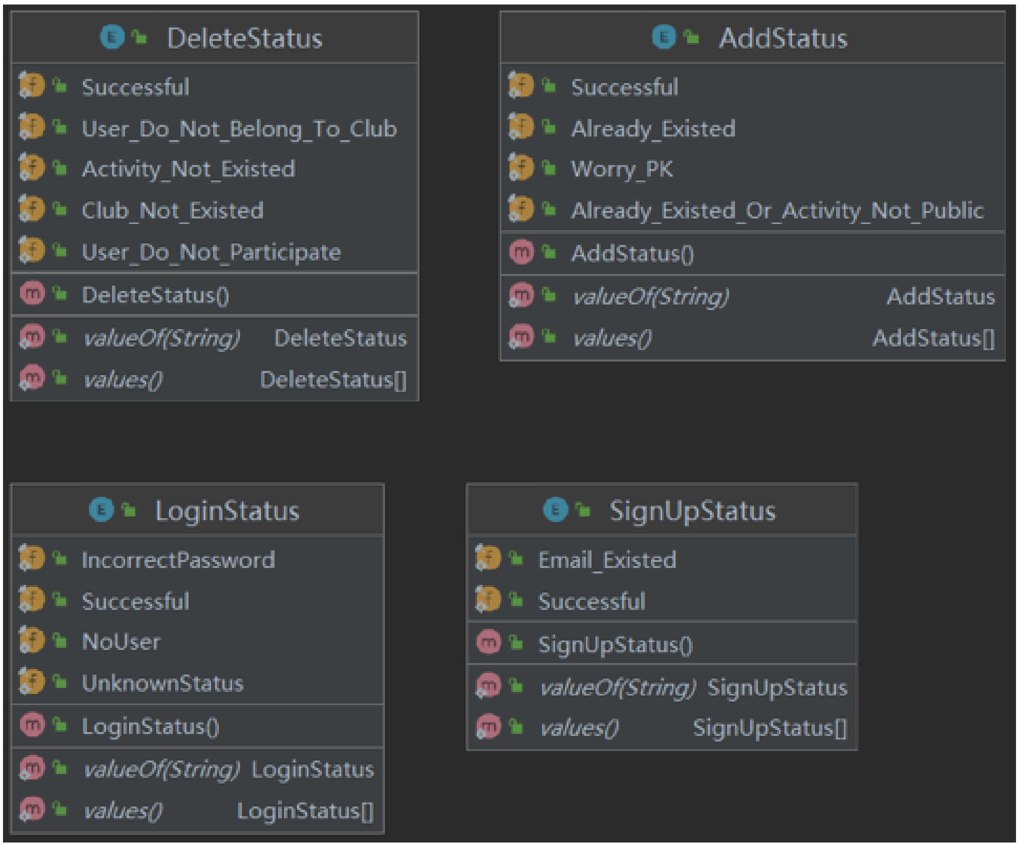
upload-controller Upload Controller

user-controller User Controller

社团接口 Activity Controller

5.7 Usage of Status

- Considering Boolean type is lack of message. We create many Status to transmit specific message.



DeleteStatus

Successful
User_Do_Not_Belong_To_Club
Activity_Not_Existed
Club_Not_Existed
User_Do_Not_Participate

DeleteStatus()

valueOf(String) DeleteStatus
values() DeleteStatus[]

AddStatus

Successful
Already_Existed
Worry_PK
Already_Existed_Or_Activity_Not_Public

AddStatus()

valueOf(String) AddStatus
values() AddStatus[]

LoginStatus

IncorrectPassword
Successful
NoUser
UnknownStatus

LoginStatus()

valueOf(String) LoginStatus
values() LoginStatus[]

SignUpStatus

Email_Existed
Successful

SignUpStatus()

valueOf(String) SignUpStatus
values() SignUpStatus[]