

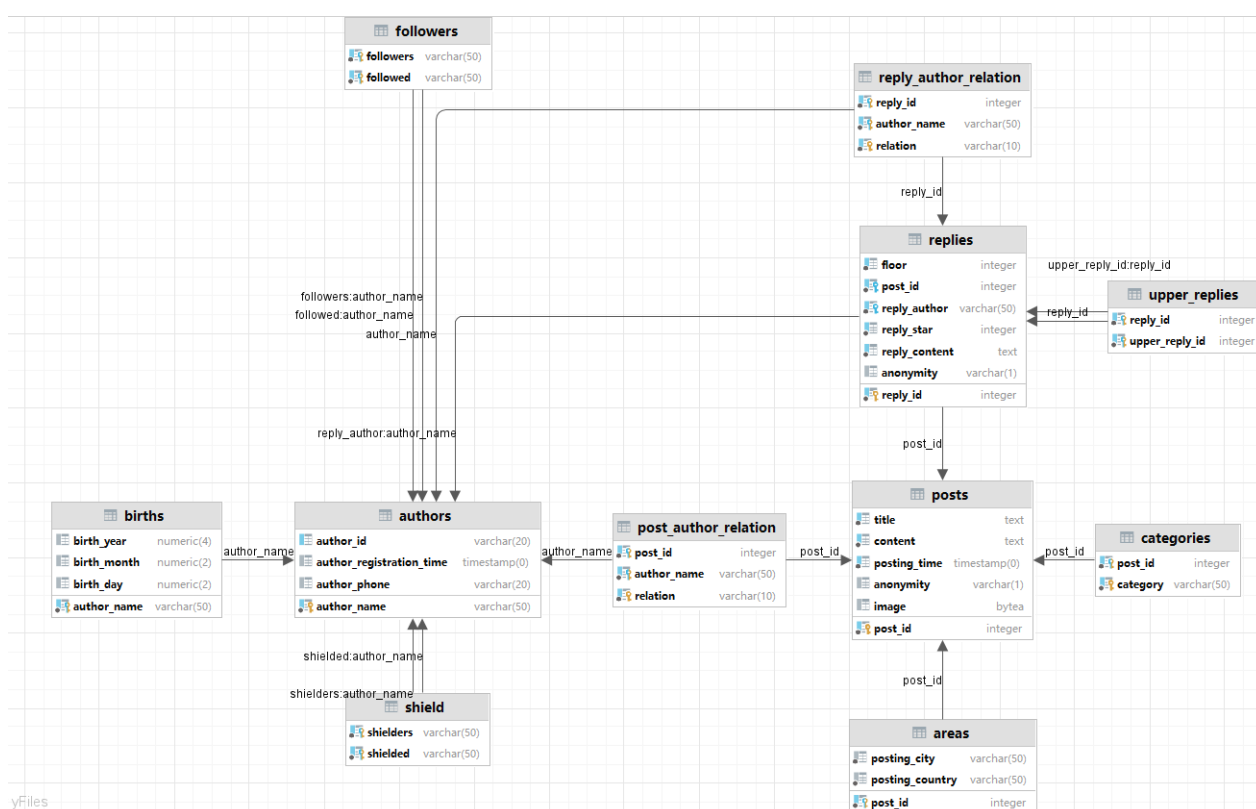
数据库原理 Project2 报告

第一部分：项目概述

数据库原理 project2 是基于 postgresql 数据库，使用 Java 进行前端设计，实现一个小型论坛的功能。该项目是由贺小珊（学号：12110848）和匡晟（学号：12011130）共同完成的（Group Number：217），贺小珊同学和匡晟同学共同完成前端 UI 和后端数据库设计，贡献比分别为 50% 和 50%。

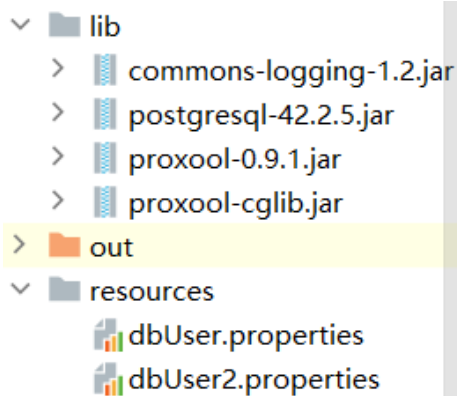
第二部分：项目 API

数据库设计：

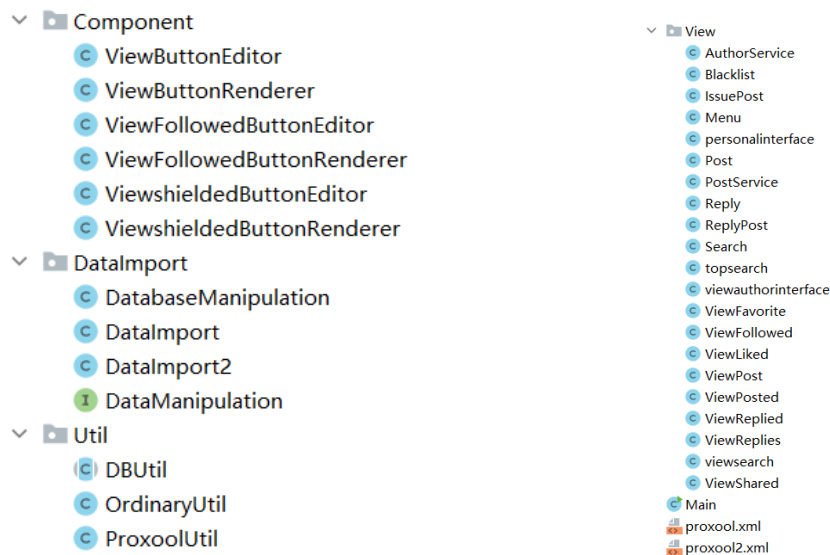


前端设计：

主要使用 Java Swing 设计与用户交互的界面，其中：
lib 文件夹有 postgresql 数据库和 proxool 连接池相关驱动，resources 文件夹存储 user 信息。



在 src 文件夹中，Component 包中主要存储 swing 的 ui 小组件，DataImport 包中提供 Java 与数据库连接的接口，Util 包为 proxool 连接池的相关接口。View 包为主要的 UI 界面代码，AuthService 为用户服务，实现登出和注销用户功能，Blacklist 为用户的屏蔽列表，IssuePost 为用户发帖界面，可以实现发帖功能，Menu 为主界面，可进行下一步操作，personalinterface 为用户中心界面，可以查看自己 like、favorite 等过的帖子，Post 为帖子界面，展示帖子的具体内容，PostService 为帖子服务界面，可进行下一步查看、搜索梯子和查看热搜榜功能，Reply 为回复界面，展示回复的具体内容，ReplyPost 为发表回复界面，可以实现回复功能，Search 为搜索界面，可填写相关关键词进行搜索，topsearch 为热搜榜界面，可以展示热度高的帖子，viewauthorinterface 为用户查看他人主页界面，可以展示其他用户的具体信息，ViewFavorite、ViewFollowed、ViewLiked、ViewPosted、ViewReplied、ViewShared 均为展示用户个人 favorite、follow、like 等过的帖子或用户，而 ViewPost 和 ViewReplies 展示 post 和 reply 的列表，可进行具体查体 post 和 reply 的具体内容的操作，viewsearch 为展示搜索的帖子界面。而 Main 为启动程序。



接口设计（这里仅展示基础功能，高级功能于第三部分详述）：

API	入参	返回值	功能	对应文件
Create Account	String author_name String author_id String author_phone	boolean 是否创建成功	用户输入用户名、ID和电话号码，实现用户注册。若用户名已存在，将弹出提示窗口。	Menu
Login	String author_name	boolean 是否登陆成功	用户输入用户名，实现用户登录。若用户不存在，将弹出提示窗口。	Menu
Like	int post_id String author_name String relation	boolean 是否点赞成功	用户浏览帖子，点击按钮可点赞或取消点赞该贴	Post
Favorite	int post_id String author_name String relation	boolean 是否收藏成功	用户浏览帖子，点击按钮可收藏或取消收藏该贴	Post
Share	int post_id String author_name String relation	boolean 是否分享成功	用户浏览帖子，点击按钮可分享或取消分享该贴	Post
View Like	int post_id String author_name String relation	List<int> post_id List<String> post_title	用户查看用户点赞过的帖子列表，点击即可查看详细帖子	ViewLiked
View Favorite	int post_id String author_name String relation	List<int> post_id List<String> post_title	用户查看用户收藏过的帖子列表，点击即可查看详细帖子	ViewFavorite
View Share	int post_id String author_name String relation	List<int> post_id List<String> post_title	用户查看用户分享过的帖子列表，点击即可查看详细帖子	ViewShared
View Posted	int post_id String author_name String relation	List<int> post_id List<String> post_title	用户查看用户发表过的帖子列表，点击即可查看详细帖子	ViewPosted
View Replied	int post_id String author_name String relation	List<int> post_id List<String> post_title	用户查看用户回复过的帖子列表，点击即可查看详细帖子	ViewReplied
Follow	String author_name String followed_author_name	boolean 是否关注成功	用户关注其他作者，点击按钮可关注或取消关注该作者	Post/Reply
View Follow	String author_name	List<String> author_id List<String> author_name	用户查看用户关注的作者列表，点击即可取消关注	ViewFollowed
Issue Post	String Title String Content String Category1 String Category2 String Category3 String Category4 String City String Country String anonymous byte[] image String author_name Timestamp posting_time	int post_id 发布的帖子ID boolean 是否发布成功	用户发布帖子，可发送图片，也可进行匿名	IssuePost
Issue Reply	String Content String author_name id upper_reply_id id reply_post_id String anonymous	id reply_id 发布的回复ID boolean 是否发布成功	用户发布回复，可回复帖子或回复帖子，可进行匿名	ReplyPost
View Post	id post_id	String Title String Content String Category1 String Category2 String Category3 String Category4 String City String Country String anonymous byte[] image String author_name Timestamp posting_time	用户查看帖子，可查看图片，也可进行相关操作	Post
View Reply	id reply_id	String Content String upper_reply_content String author_name id upper_reply_id id reply_post_id String anonymous	用户查看回复，也可进行相关操作	Reply

第三部分：项目高级功能

·完善 API 设计，在用户可使用的功能上做拓展

1、匿名发言

在 posts 和 replies 表中建 anonymity 列，“Y”为该帖子或回复为匿名，“N”相反。用户在发帖或回复时在提交按钮上方有一个小按钮决定是否匿名。

City:
Country:

☐ Anonymous

Submit

在浏览帖子或回复时，如果为匿名用户发表，author 一栏将提示为“anonymous!”

Author: anonymous!

Posting time: 2023-05-30 16:23:28
Posting area: Shenzhen, China

like(0)

favorite(0)

share(0)

reply

view replies

如果为非匿名用户发表，author 一栏将显示该帖或回复的作者名字。

Author: anonymous!

Posting time: 2023-05-30 16:23:28
Posting area: Shenzhen, China

like(0)

favorite(0)

share(0)

reply

view replies

2、支持图片

在 posts 表中建 image 列，数据类型为 btype[]。用户在发帖、浏览帖子时可以加载图片，在发帖时，前端读取图片路径并转为 byte[]，然后转为十六进制进行传输。

```
private byte[] convertImageToByteArray(String imagePath) throws IOException {
    BufferedImage bufferedImage = ImageIO.read(new File(imagePath));
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    ImageIO.write(bufferedImage, "jpg", baos);
    return baos.toByteArray();
}
```

1 usage

```
private static String byteArrayToHex(byte[] bytes) {
    StringBuilder hex = new StringBuilder();
    for (byte b : bytes) {
        hex.append(String.format("%02X", b));
    }
    return hex.toString();
}
```

在浏览帖子时，postgresql 数据库传输 byte[] 至前端，并展示于 UI 界面。

3、屏蔽或拉黑功能

使用建表语句创建新表“shield”来存储用户的屏蔽信息，更改初始化页面语句，except 用户所屏蔽的作者的帖子，使其不会显示在浏览页面上，对于用户所新屏蔽的作者，会弹窗提示其已屏蔽，并指路用户中心—黑名单，来取消屏蔽，并且自动关闭所屏蔽的帖子。

Post id: 192

Title: My Thoughts on Meditation and Its Benefits

I have recently started meditating, and I must say, I am impressed with its benefits. Meditation has helped me become more mindful, less stressed, and more focused. It has improved my overall well-being and has become an essential part of my daily routine.

消息

Your have shield the author! If want to cancel, please go to the personal interface--Blacklist!

确定

Category:

Author: political_inflation

Posting time: 2021-07-07 16:30:14

Posting area: Valencia, Spain

like(50)

favorite(18)

share(29)

reply

shield(political_i...

view replies

view author interface

4、热搜榜功能

对全站帖子进行热度排行，基于 like、favorite、share 来计算帖子的热度值，展示热度前十的帖子并显示热度值，排名支持动态变化。

Post ID	Title	Hot Value	View
120	My Favorite Music Genre and Why ...	172	View Post120
1	My First Time Skiing Experience	171	View Post1
114	My Trip to Japan	170	View Post114
24	The Benefits of Yoga for Stress R...	161	View Post24
100	My Experience at a Music Festival	158	View Post100
170	My Experience Learning a New La...	155	View Post170
159	A Hiking Adventure in the Mountains	152	View Post159
5	The Importance of a Good Night's ...	151	View Post5
93	My Favorite Hiking Trail	150	View Post93
134	The Best Hiking Experience I Ever ...	148	View Post134

5、多参数搜索功能

用户可以输入关键词、作者、种类、起止时间的任意组合来进行搜索，其中，起止时间必须同时出现或者同时缺失，否则会弹窗提示错误；关键词将用于帖子的 title 和 content 的模糊查询。

Issue Post

key word :

<- Back

author:

category :

start time:

end time:

Submit

在后端使用 SQL 语句“select distinct (posts.post_id),title from”+ilike 来进行忽略大小写的模糊查询。

6、其他

制作了其他用户的主界面的展示功能。通过在帖子展示页面加入新的 button——“view author interface”，点击之后会展示此作者的个人页面，包括用户的创立时间以及此用户所发布的所有帖子。

For example:



·使用数据库连接池

本数据库采用 proxool 连接池，并进行多线程访问数据库

```
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <!-- the proxool configuration can be embedded within your own application's.
4  Anything outside the "proxool" tag is ignored. -->
5
6  <something-else-entirely>
7    <proxool>
8      <alias>postgres</alias>
9      <driver-url>jdbc:postgresql://localhost:5432/Project1</driver-url>
10     <driver-class>org.postgresql.Driver</driver-class>
11     <driver-properties>
12       <property name="user" value="postgres"/>
13       <property name="password" value="KSa1303151717"/>
14     <!--
15     <!--
16       <property name="user" value="kuang"/>-->
17       <property name="password" value="123456"/>-->
18     </driver-properties>
19     <minimum-connection-count>5</minimum-connection-count>
20     <maximum-connection-count>10</maximum-connection-count>
21   </proxool>
22 </something-else-entirely>
```

```
-----Thread 24 visiting DB!-----
--- Active:1    Available:4    Max:10 ---
Insert statement failed
错误: 对表 post_author_relation 权限不够
 在位置: SQL 语句 "insert into post_author_relation(post_id, author_name, relation) values (postid, name, relations)"
在SQL语句的第3行的PL/pgSQL函数inserts(character varying,character varying,integer)
六月 02, 2023 10:38:25 下午 org.logicalcobwebs.proxool.postgres shutdown
信息: Shutting down 'postgres' pool immediately [Shutdown Hook]
```

·页面 UI 设计，展示简洁美观的 GUI 页面

本 project 采用 Java Swing 进行 UI 设计，主界面有三个按钮，分别展示用户中心、帖子服务和用户服务

Menu



个人中心界面可查看用户 like、favorite、share 等过的帖子并以表格形式展示

View Liked

Post ID	Title	View
53	My First Time Travel Experience	View Post53
9	I Just Started Learning a New Language	View Post9
15	My Favorite Sport is Basketball	View Post15
13	My First Time Hiking in the Mountains	View Post13
1	My First Time Skiing Experience	View Post1

帖子界面可查看、搜索帖子并浏览热搜榜，在具体的 post 和 reply 界面，我们展示帖子的具体内容用户可以进行相关操作

Post221

Post id: 221

Title: 孙燕姿是最棒的歌手！

如果你不开心，就听孙燕姿吧，她的歌声足够治愈。
如果你太开心，就听孙燕姿吧，她的歌声足够虐心。



Category: 孙燕姿 冷门歌手

Author: kkk

follow kkk

Posting time: 2023-06-02 23:45:48

Posting area: 深圳, 中国

like(0)

favorite(0)

share(0)

reply

shield(kkk)

view replies

view author interface

Reply4251

Reply id: 4251

Reply Post: 221

UPPER REPLY:
No upper reply!

REPLY CONTENT:
我认为你说得对！《Stefanie》是世界上最好的专辑！

Author: kkk

follow kkk

star(0)

reply

view replies

本 project UI 设计总体而言简洁明了，对用户较为友好。

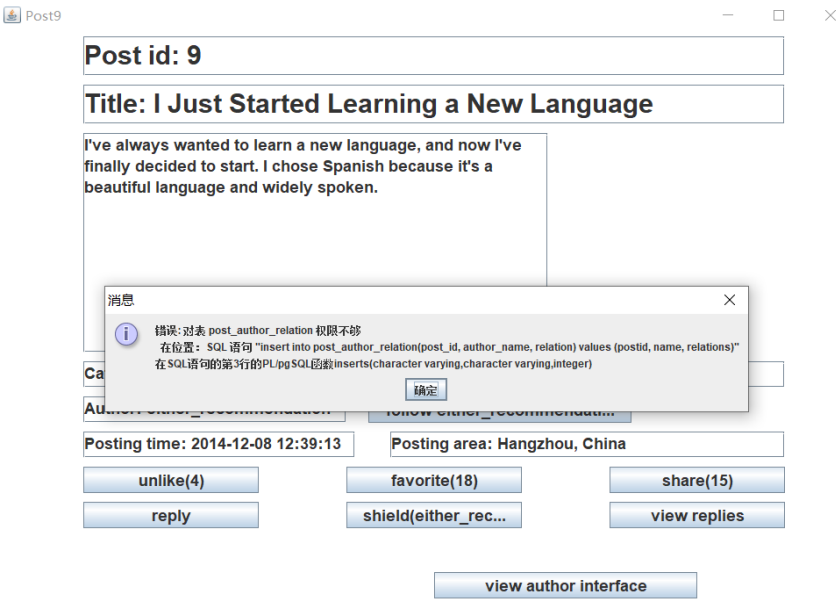
·合理使用用户权限、过程、索引、视图等功能

1、用户权限

本 Project 支持访客系统，对于访客而言我们新建了一个用户 kuang，赋予该用户查看搜索表的权限，但不支持插入、删除等操作。相关 sql 语句如下。

```
create user kuang with password '123456';
grant all privileges on database "Project1" to kuang;
grant select on table areas to kuang;
grant select on table authors to kuang;
grant select on table births to kuang;
grant select on table categories to kuang;
grant select on table followers to kuang;
grant select on table post_author_relation to kuang;
grant select on table posts to kuang;
grant select on table replies to kuang;
grant select on table reply_author_relation to kuang;
grant select on table shield to kuang;
grant select on table upper_replies to kuang;
```

如果该用户进行 like、favorite 等操作，前端会提示访客无相关权限，而且访客无权限查看回复，这在当今流行的社交媒体和 BBS 论坛中也是符合主流的。



而其他的普通用户赋予的权限是超级用户 postgres，具有一切权限。

2、过程

创建过程：

```

create or replace procedure deletes(name varchar(50),relations varchar(10),postid int )
    language plpgsql
as
    $$
begin
    delete from post_author_relation where author_name=name and relation=relations and post_id=postid;
end;
$$;

create or replace procedure inserts(name varchar(50),relations varchar(10),postid int )
    language plpgsql
as
    $$
begin
    insert into post_author_relation(post_id, author_name, relation) values (postid, name, relations);
end;
$$;

```

将删除和插入用户分享、点赞、收藏等功能简化存储下来，使用的时候只需要 call + 过程名 + 传入用户名和帖子 id 信息即可，简化了过程，提高了效率。

3、索引

```

create index search1 on posts(posting_time);
create index search2 on posts(title);
create index search3 on posts(content);
create index search4 on posts(posting_time,title);
create index search5 on posts(posting_time,content);
create index search6 on posts(title,content);
create index search7 on posts(title,content,posting_time);

```

主要使用在 advance 部分的多参数搜索功能中，使在特定范围内的搜索速度加快。

4、视图

本次 Project 支持多级回复，所以回复的上层回复是单独建表的，我们通过视图将 reply 信息和 upper_reply 整合，便于搜索。

```

create view reply_view as
select replies.reply_id, replies.floor, replies.post_id, replies.reply_author, replies.reply_star, replies.reply_content,
       replies.anonymity, upper_reply_id
from replies left join upper_replies on replies.reply_id = upper_replies.reply_id order by reply_id;

```

5、触发器

创建用户时用户需要填写 ID 号码，而我们可以根据 ID 号码推断出用户的生日信息，并插入 births 表中，因此我们做了两个相关的 trigger。（此处仅展示插入的 trigger）

```

create or replace function birth_insert()
    returns trigger
    as
    $$
    declare
        year int := 0;
        month int := 0;
        day int := 0;
    begin
        year := (ascii(substr(new.Author_ID, 7, 1)) - 48) * 1000 + (ascii(substr(new.Author_ID, 8, 1)) - 48) * 100 + (ascii(substr(new.Author_ID,
        month := (ascii(substr(new.Author_ID, 11, 1)) - 48) * 10 + (ascii(substr(new.Author_ID, 12, 1)) - 48);
        day := (ascii(substr(new.Author_ID, 13, 1)) - 48) * 10 + (ascii(substr(new.Author_ID, 14, 1)) - 48);
        insert into births(author_name, birth_year, birth_month, birth_day) values (new.author_name, year, month, day);
        return new;
    end;
    $$
    language plpgsql;

create trigger authors_trigger_insert
    after insert
    on authors
    for each row
    execute procedure birth_insert();

```

第四部分：总结与反思

这次 Project 其实起手很早，但是由于 UI 的搭建过于生疏，唯一一次搭建 UI 还是上次的 Java project，对于过程、index 等的概念和基本用法、用途等不是很清楚，还需要再去翻阅资料才能将其完成，所以拖得很晚。总而言之，编程语言的运用和数据库的基础知识还有待加强。