Project Report

Princeples of Database Systems

Yan Li | N17410712 | yl4752
Shangwen Yan | N17091204 | sy2160

Index

PART 1 Introduction

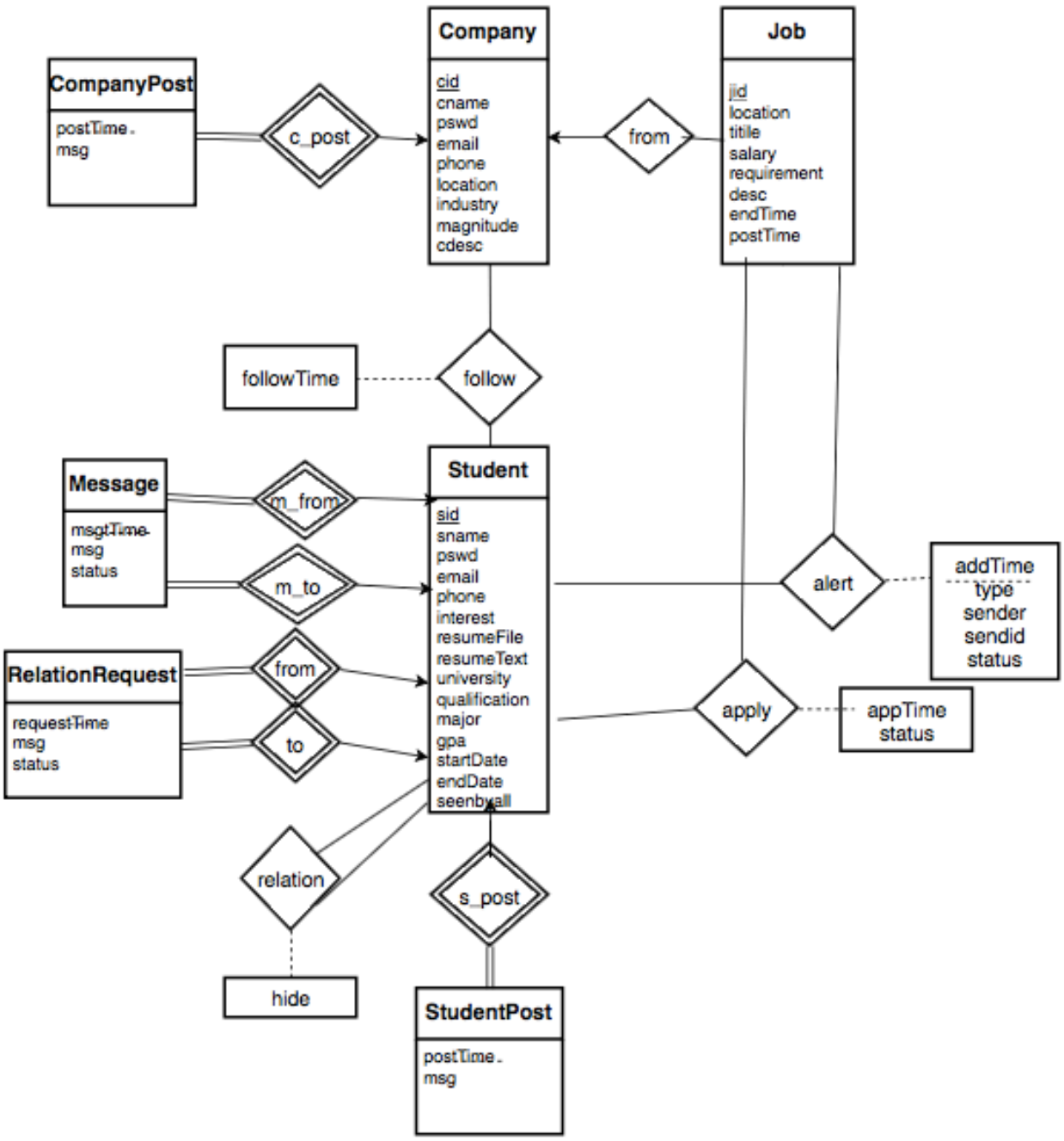
This document is to illustrate the design of our website “Jobster” mainly including two parts: database design and front end web page design.

“Jobster” is a website for both students who are looking for jobs and companies which are planning to hire students. Users as student could sign up, login, search jobs and apply, search companies and follow, add friends and chat with them, post blogs, share a job to a friend and so on. For users who are signed up as company, they can login to post job announcements which would be pushed to followers automatically, or run a filter to advertise a job to any suitable students. Company users can also check applicants’ information and accept or reject.

In this project, we use MySQL as relational database and Apache as web server. The front end and back end are implemented by PHP, Javascript and HTML/CSS, JQuery and AJAX by which could make the Web applications can send and retrieve data from a server asynchronously.

PART 2 Database Design

a. E-R diagram



b. Tables

Student								
sid	sname	pswd	email	phone	interest	resumeFile	resumeText	university

gpa	startDate	endDate	qualification	major	seenbyall

PK: sid
Constraints: email is unique, sid is generated by mysql using auto_increment
Description: email is used for logging in, and phone number is not madatory.
resumeText is used for detailed information about relative experience, and resumeFile , which is of type BLOB, is used for uploading a full resume. Seenbyall is 1 by default meaning the information can be seen by any user. If it is updated as 0, only the company the user is following can see the information.

StudentPost		
sid	**postTime**	msg

PK: sid, postTime
FK: sid references Student.sid
Description: students can post daily info about themselves

Relationrequest				
sid1	**Sid2**	**requestTime**	status	msg

PK: sid1,sid2, requestTime
FK: sid1,sid2 references Student.sid respectively
Description: students can send relationship request to be a friend of another student with a message. If not replied, a student can send request several times later. If this request has been read, status will be 1, otherwise 0.

Relationship		
sid1	**sid2**	hide

PK: sid1,sid2
FK: sid1,sid2 references Student.sid respectively
Description: if Relationship Request is accepted, a relationship is added in this table. For example, if student A with sid 1 send a relationship request to student b with sid 2, and the request is accepted, there will be two entries stored in this table: (1,2) and (2,1) which makes it easier for future queries. And hide is 0 by default. It can be set as 1 by user sid1, then user sid2 could not see sid1's complete information.

Message				
sid1	**sid2**	**msgTime**	msg	status

PK: sid1,sid2,msgTime
FK: sid1,sid2 references Student.sid respectively
Description: msg is sent from student with sid1 to student with sid2 at msgTime. Status will be 1 after the message has been read, otherwise status is 0 by default.

Follow		
sid	**cid**	followTime

PK: sid,cid
FK: sid references Student.sid; cid references Company.cid
Description: Students can decide to follow certain companies (which means they will get all job announcements by those companies)

Company								
cid	cname	pswd	email	phone	location	industry	magnitude	cdesc

PK: cid
Constraints:
email is unique, cid is generated by mysql using auto_increment
Description:
email is used for logging in, and phone number is not madatory.
magnitude means how many employees in that company

CompanyPost		
cid	**postTime**	msg

PK: cid, postTime
FK: cid references Company.cid
Description: companies can post daily info

Job								
jid	cid	location	title	salary	requirement	desc	endTime	postTime

PK: jid
FK: cid references Company.cid
Constraints:
jid is generated by mysql using auto_increment
Description:
a company can post job positions which have unique jid respectively

JobAlert						
sid	**jid**	**addTime**	type	sender	sendid	status

PK: sid,jid,addTime
FK: sid references Student.sid; jid references Job.jid
Description: the table means student with sid will be notified of the job with jid, and this alert is added at addTime; type is one of:0(from followed companies), 1 (from friend forwarding),2 (company notification, which means the company is not followed by this student); one student could be notified of one specific job several times. Sender is the sender's name of this job alert and sendid is the id of the sender. Status is 0 by default meaning unread. It will be updated as 1 after been read.

Application			
sid	**jid**	appTime	status

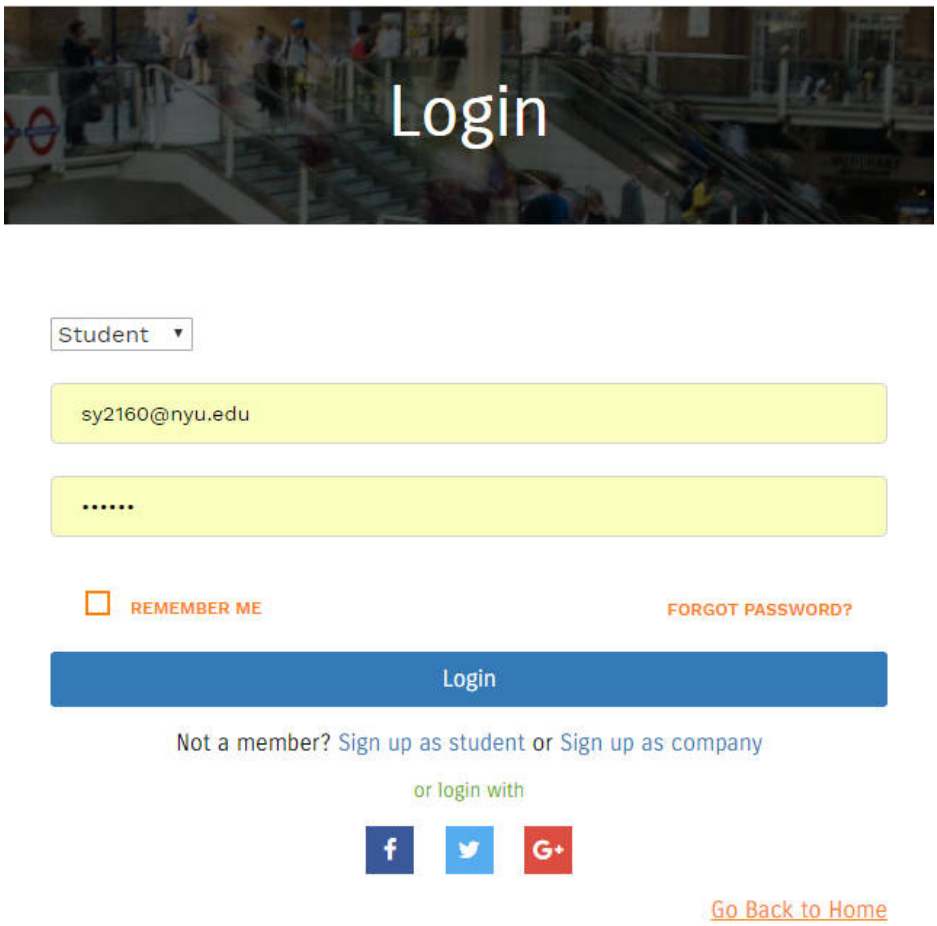
PK: sid,jid
FK: sid references Student.sid; jid references Job.jid
Description: one student could only apply for a job once. Status is one of:0 (unreviewed),1 (accept, which means the student might be notified of later interview),2 (reject),3 (unreplied, which means the application is reviewed by hr but not replied yet)

PART 3 Web Design

a. Page design

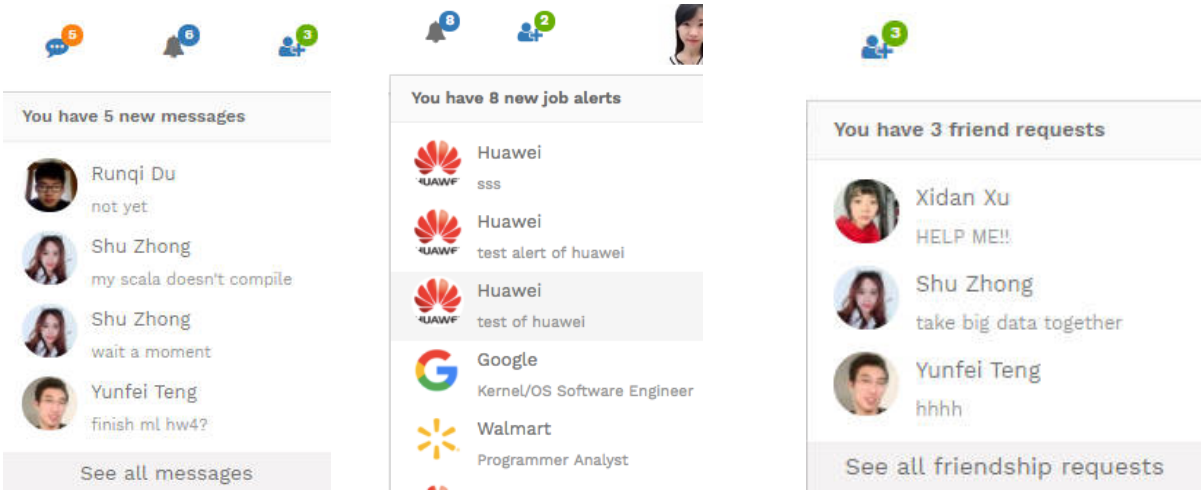
I. Login page:

You can login as student or company at this page by your email and password. If you have not signed up yet, click ‘sign up as student’ or ‘sign up as company’ to the sign up page. Students can upload resume files in sign up page.

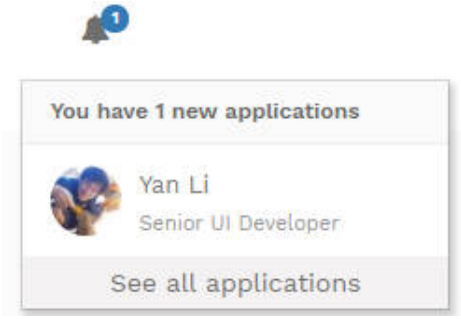


II. After login successfully, below are headers for student and company separately.

Above is header for student users. Users could search jobs in the search field. The header also shows unread messages, unread job alerts and unread friend requests with numbers. By clicking the three icons, they will show detailed unread messages/alerts/requests as below.

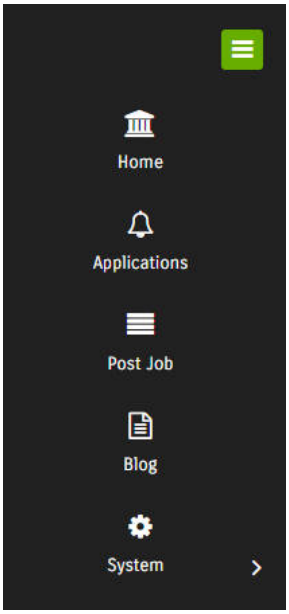


Above is header for company users. Users could search students in the search field. And the header also shows unread applications. Clicking this icon, it will show the unread application as below.

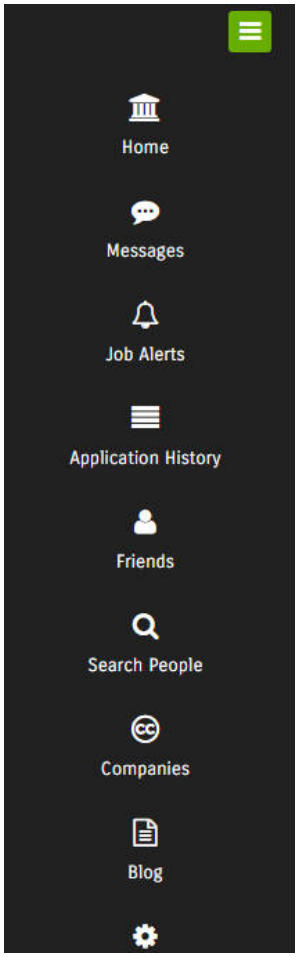


III. Side bar:

Side bar for company users is shown below:

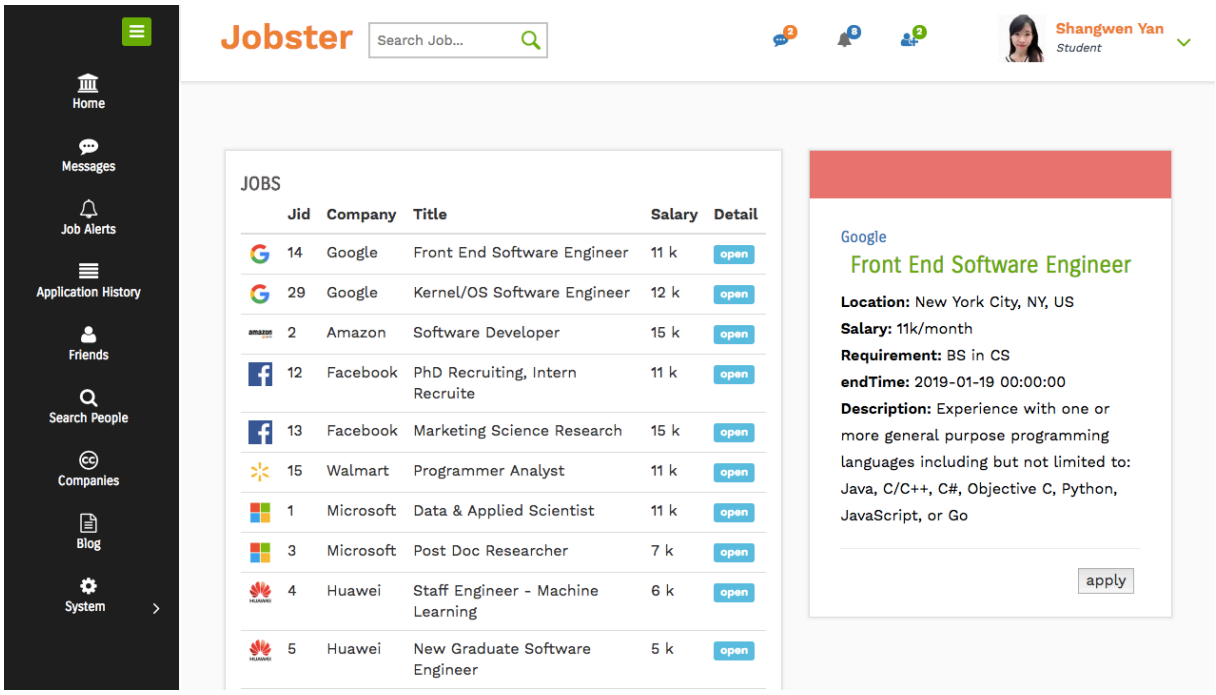


Side bar for student users is shown below:

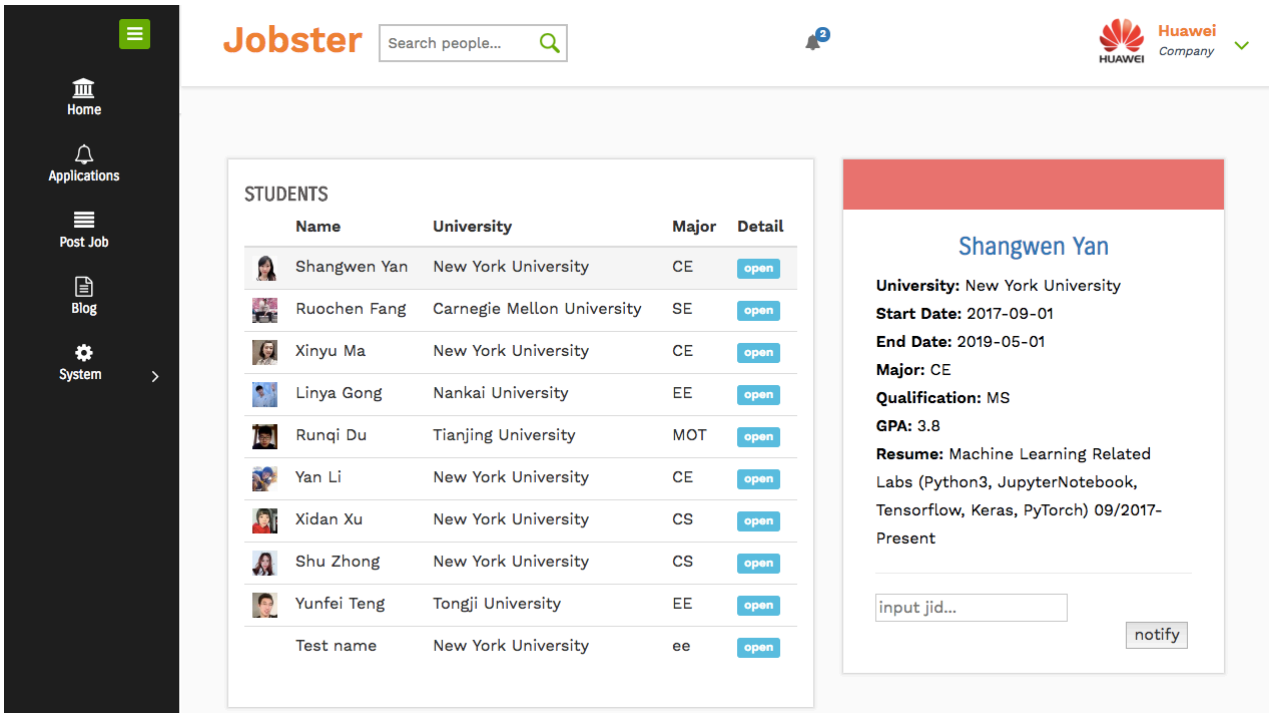


IV. Home page

After student users login, it will jump to home page which lists all the jobs. Users could click the 'open' button of any job to see the job announcement and also apply it by pressing the 'apply' button.



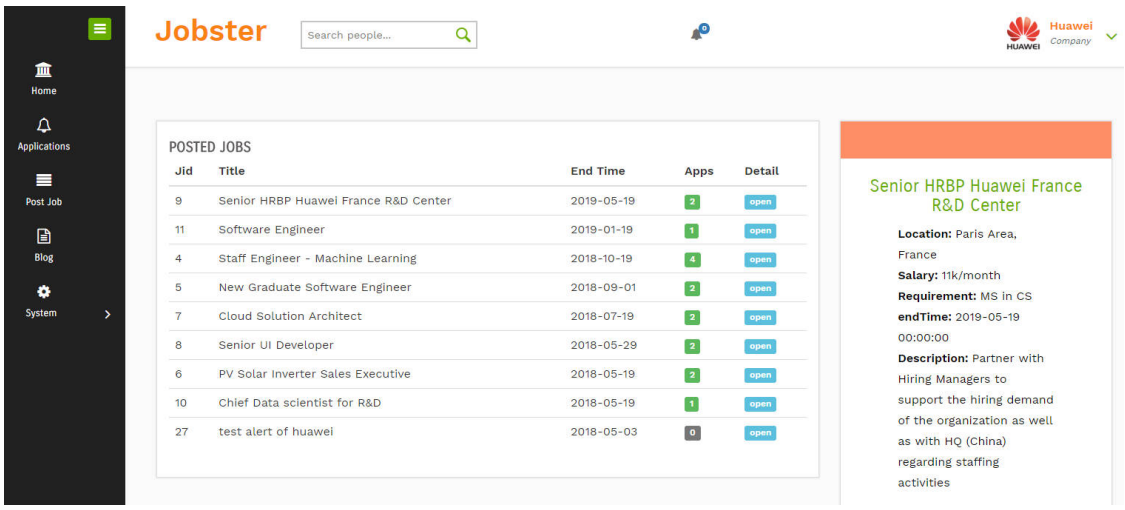
After company users login, it will jump to home page which lists all the students. Users could click the ‘open’ button of any student to see the information of this student and push a job by jid to this student.



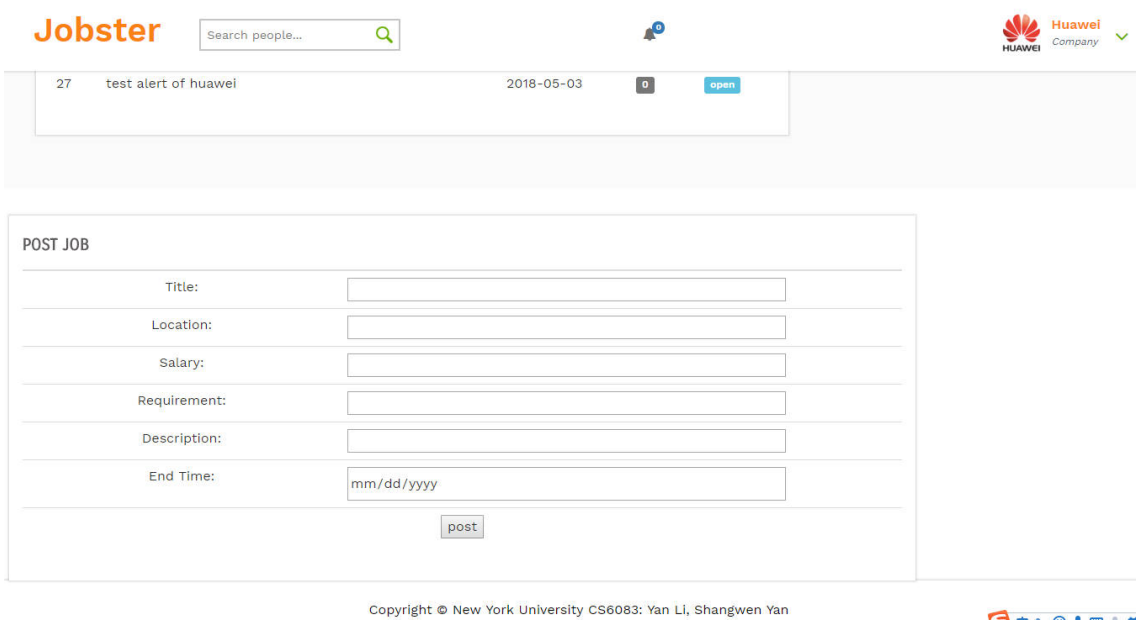
b. Basic function

I. Post jobs:

After company users clicking ‘post jobs’ in the side bar, posted jobs will be listed as below. ‘Apps’ means the total number of applicants for he job.

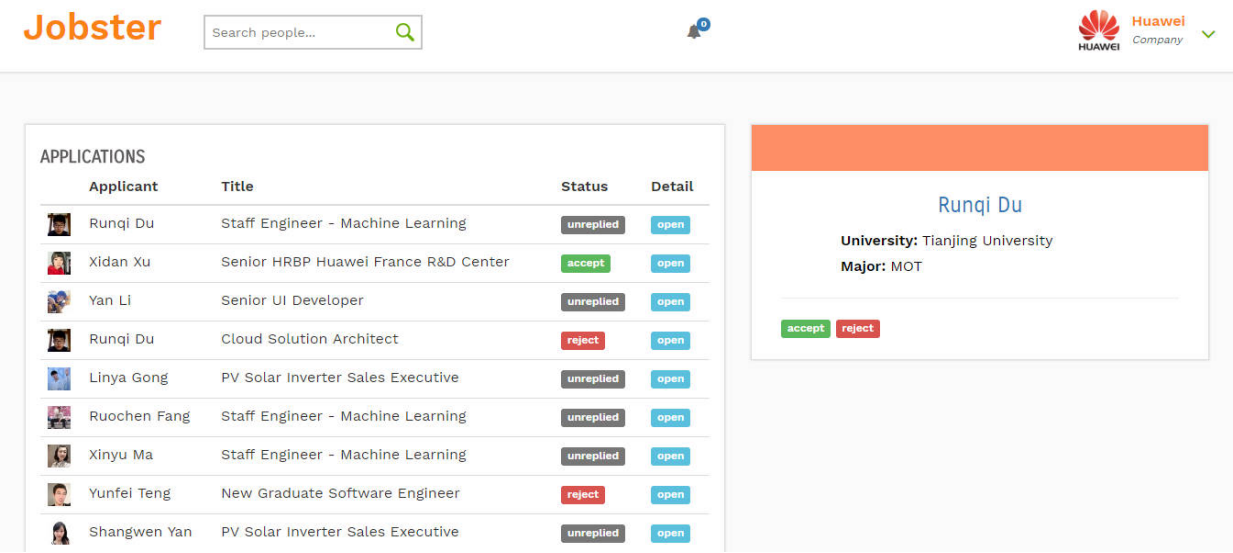


Scroll down, then the POST JOB area will be found. Users can post jobs here and the job announcements will be pushed to all followers automatically.

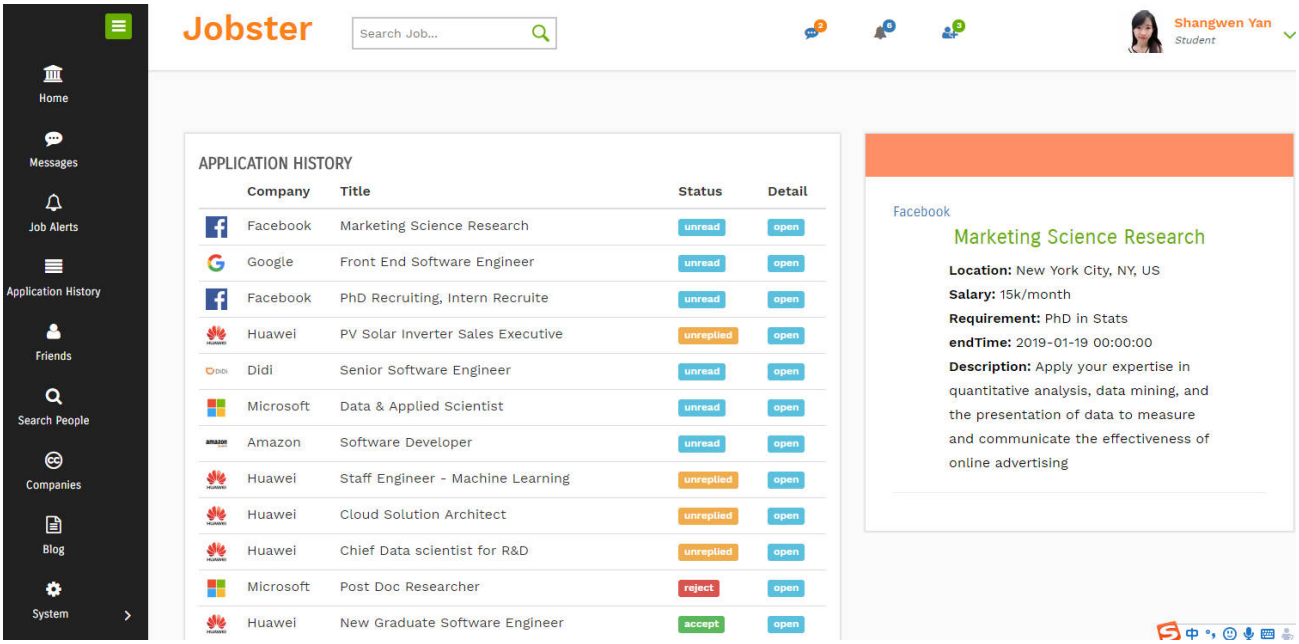


II. Deal with applications:

After company users clicking 'applications' in the side bar, all applications will be listed as below. 'Status' indicates the application is accepted, rejected, unreviewed or unreplied. As long as a new application is opened, the status of this application will change from unreviewed to unreplied. Even if a company user has rejected an application but regrets later, he could still press accept and then certain application's status will become accepted.



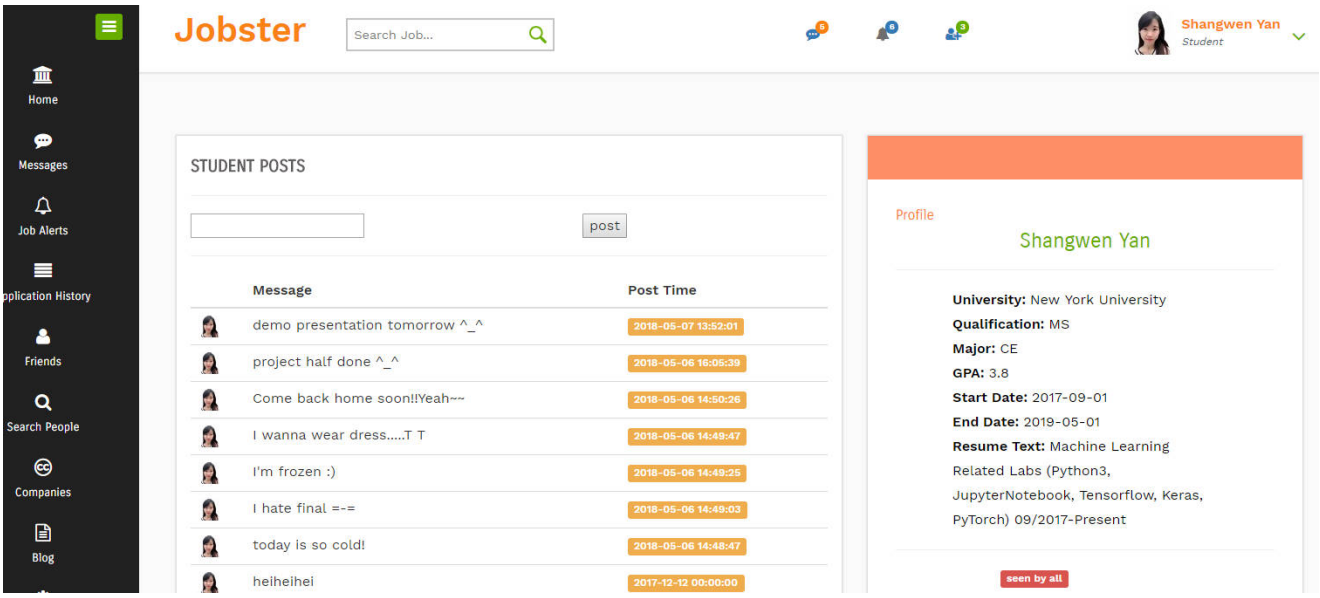
For student users, click application history in the side bar to check the status of users' previous applications.



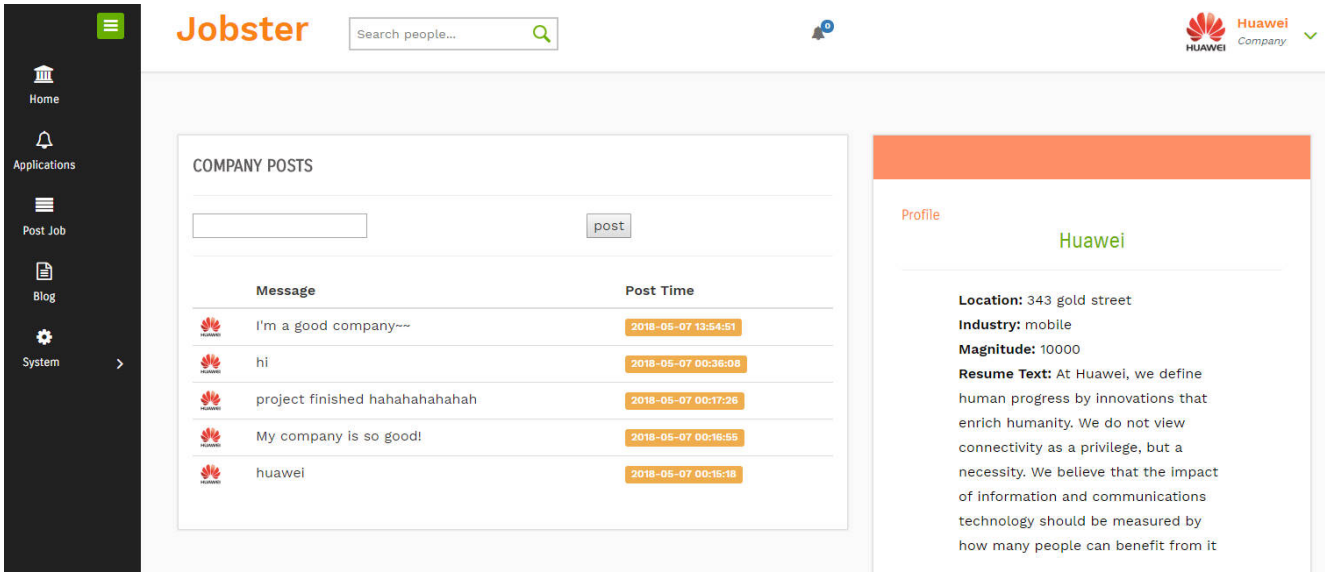
III. Post blogs:

Both students and companies can click blog in the side bar and post blogs which can be seen by all.

Students' blogs as below:

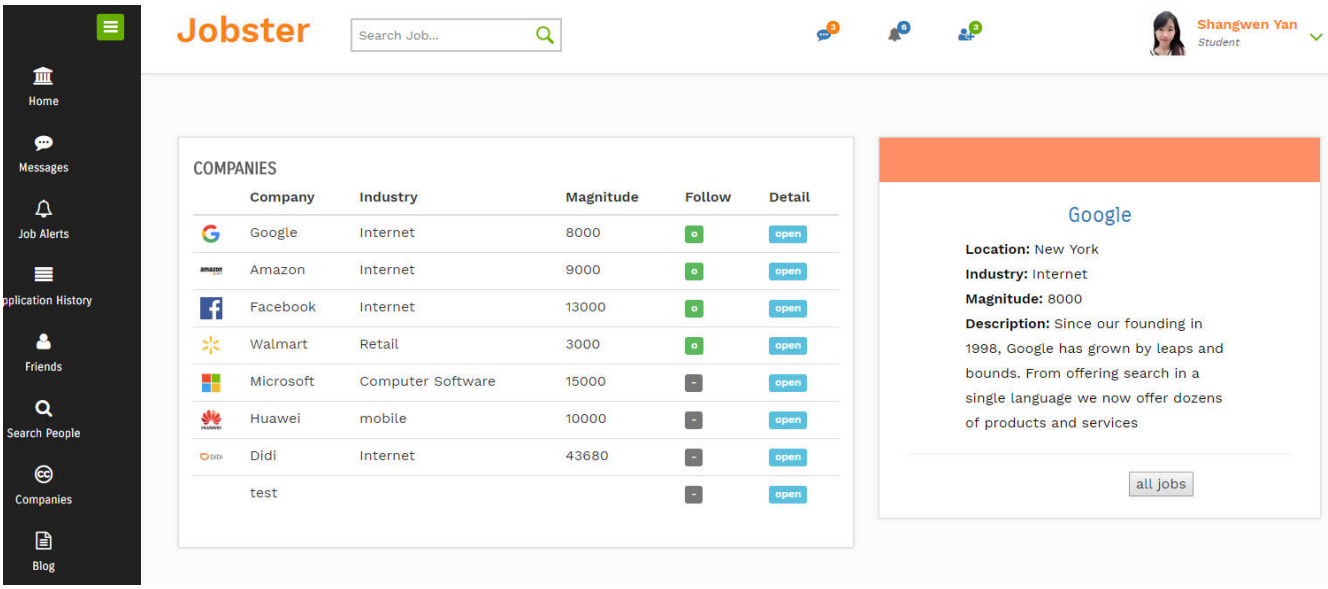


Companies' blogs as below:



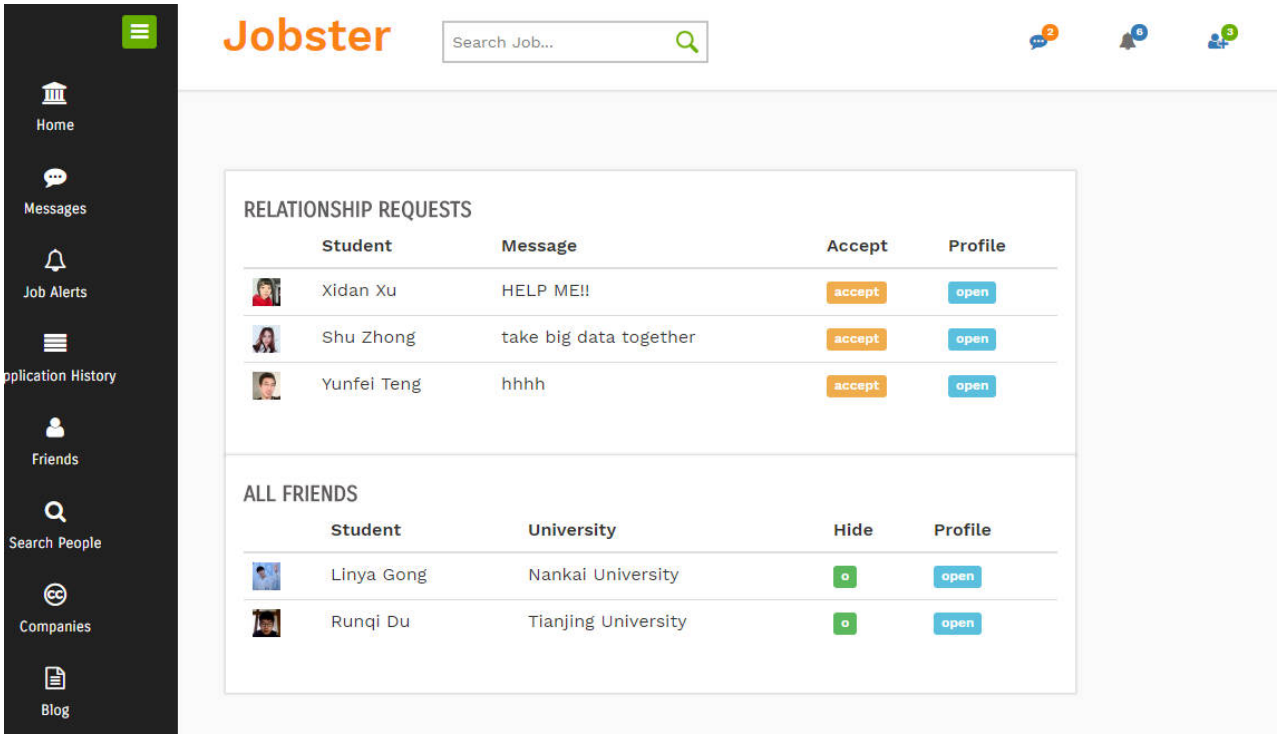
IV. Follow companies:

For student users, click companies in the side bar. Then all companies will be listed. All companies the user has followed will be listed at top. The green circle in Follow field indicates that the company is currently followed by the user. Click the green circle to unfollow or click the gray line to follow.

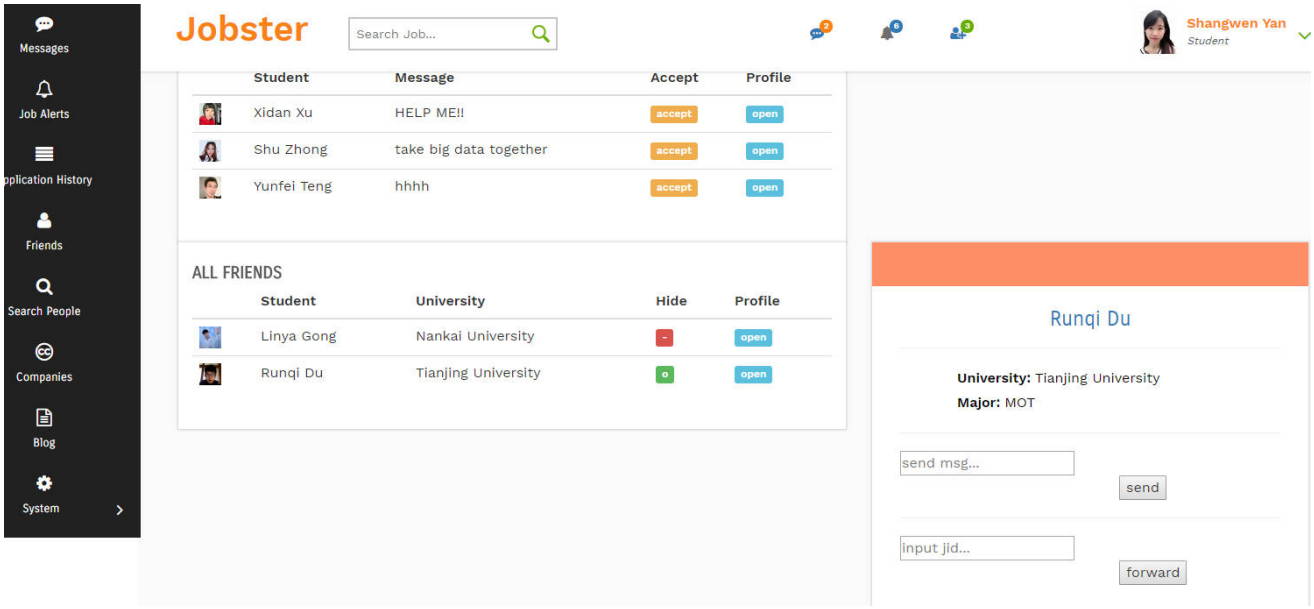


V. Friends:

Unhandled friend requests and all friends will be listed when student users click Friends in the side bar. Student users can click ‘accept’ to accept the friend requests. In the ALL FRIENDS area, green circle in Hide field indicates that friend could see the user’s all information. If click the green circle, it will becomes red line which means friend can only see limited basic information of the user.

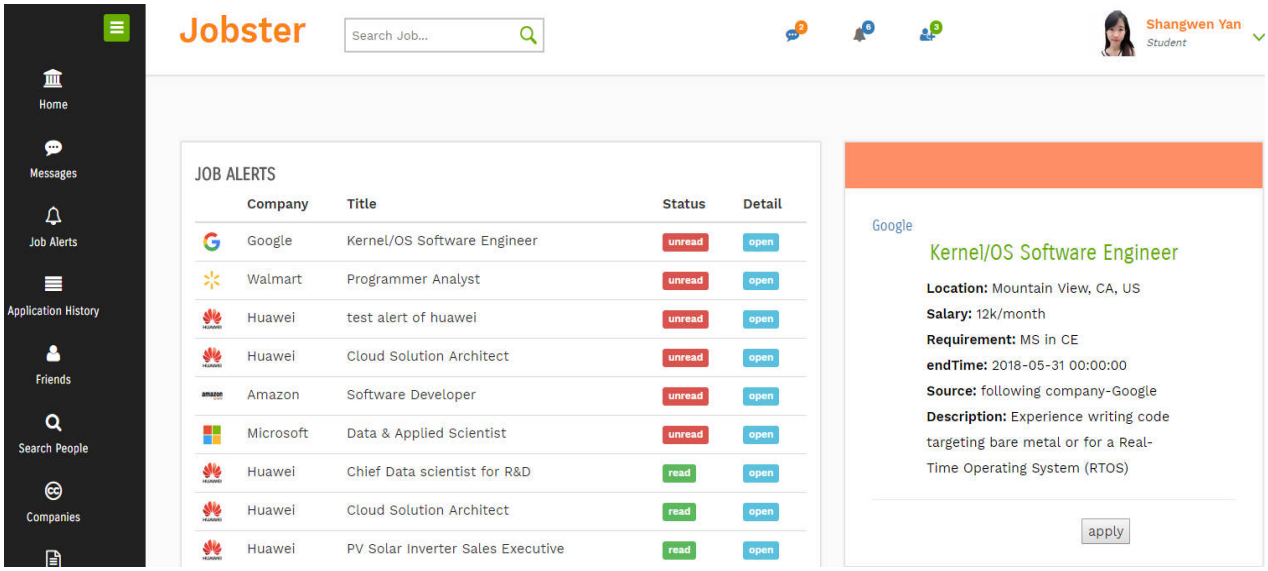


After clicking open in profile field, student users can see the information of friends and send messages or share a job by jid.



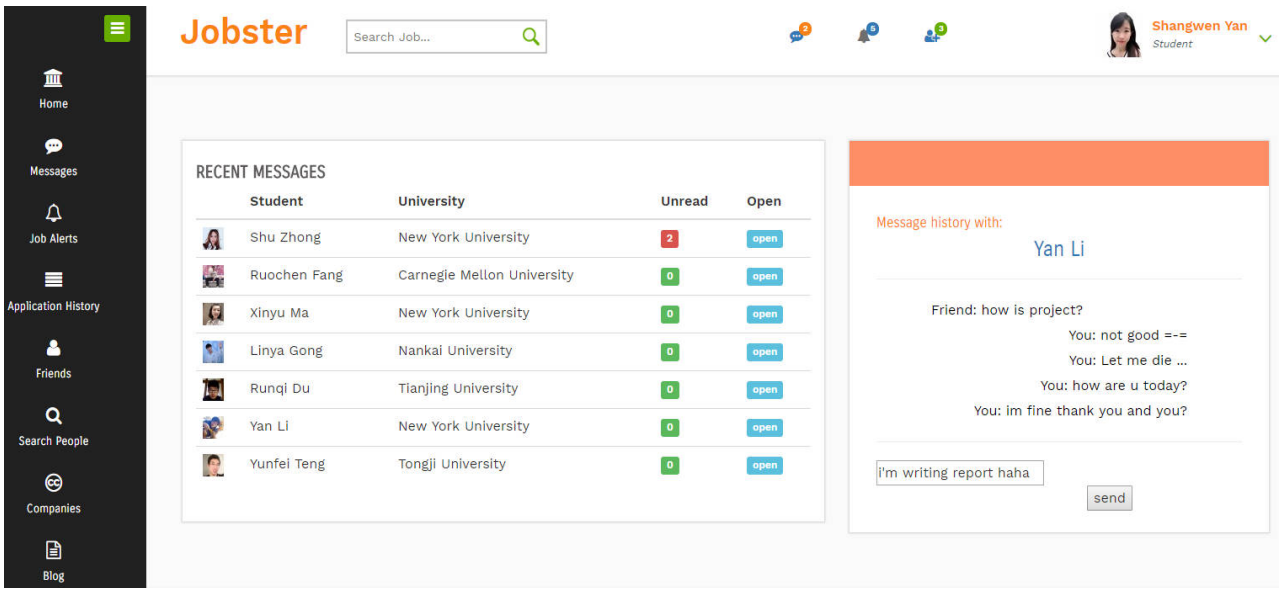
VI. Job alerts:

For student users, click Job Alerts in side bar to check received job alerts. Users can find out who sent this job alert in the source field after clicking ‘open’, as well as apply this job by clicking ‘apply’. As long as users click open of a new job alert, the status of the job alert will change from unread to read. Unread job alerts will be listed at top.



VII. Chat with friends:

For student users, click Messages in side bar to see recent messages. All friends who have conversations with user will be listed and friends whose messages are unread will be listed at top (red number indicates how many unread messages). After clicking open, conversation will be shown on the right. Users can also send messages here.



VIII. Restrict information:

Students' all information is set by default that all people or company could see. Student users could set sensitive information (eg. GPA) only seen by companies the user has followed by clicking the green button 'seen by all' in blog/profile page (The button then turns to red).

Student users can also restrict information from one specific friend by click the green hide button in Friends page.

The difference between restricted and unrestricted information can be seen as below:

Shangwen Yan

University: New York University

Start Date: 2017-09-01

End Date: 2019-05-01

Major: CE

Qualification: MS

GPA: 3.8

Resume: Machine Learning Related Labs (Python3, JupyterNotebook, Tensorflow, Keras, PyTorch) 09/2017-Present

Shangwen Yan

University: New York University

Major: CE

PART 4 Security and Concurrency

I. Guard against SQL injection: use prepared statement

```
$sql="INSERT INTO student (sname,pswd,email,phone,interest,resumefile,resumetext,university,
                qualification,major,gpa,startdate,enddate) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?);";
$stmt=mysqli_stmt_init($con);
if(!mysqli_stmt_prepare($stmt,$sql)){
    echo "SQL error";
}else{
    mysqli_stmt_bind_param($stmt, types: "sssssssssssss" ,
        &var1: $sname, &...: $password,$email,$phone,$interest,$resumefile,$resumetext,$university,
            $qualification,$major,$gpa,$startdate,$enddate);
    mysqli_stmt_execute($stmt);
    $stmt->close();
```

II. Guard against cross-site scripting: use mysqli_real_escape_string

```
$email = mysqli_real_escape_string($con,$_POST['email']);
$sname = mysqli_real_escape_string($con,$_POST['sname']);
$phone = mysqli_real_escape_string($con,$_POST['phone']);
$interest = mysqli_real_escape_string($con,$_POST['interest']);
$password = mysqli_real_escape_string($con,$_POST['pswd']);
$password2 = mysqli_real_escape_string($con,$_POST['pswd2']);
$resumefile=mysqli_real_escape_string($con,$_POST['resumefile']);
$resumetext=mysqli_real_escape_string($con,$_POST['resumetext']);
$university = mysqli_real_escape_string($con,$_POST['university']);
$qualification=mysqli_real_escape_string($con,$_POST['qualification']);
$major = mysqli_real_escape_string($con,$_POST['major']);
$gpa = mysqli_real_escape_string($con,$_POST['gpa']);
$startdate = mysqli_real_escape_string($con,$_POST['start']);
$enddate =mysqli_real_escape_string($con, $_POST['end']);
```

III. Deal with concurrency: use transaction control

```php
//begin transaction, in case of concurrency
mysqli_begin_transaction($con);


$sql="INSERT INTO student (sname,pswd,email,phone,interest,resumefile,resumetext,university,
              qualification,major,gpa,startdate,enddate) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?);";
$stmt=mysqli_stmt_init($con);
if(!mysqli_stmt_prepare($stmt,$sql)){
    echo "SQL error";
}else{
    mysqli_stmt_bind_param($stmt, types: "sssssssssssss" ,
        &var1: $sname, &...: $password,$email,$phone,$interest,$resumefile,$resumetext,$university,
           $qualification,$major,$gpa,$startdate,$enddate);
    mysqli_stmt_execute($stmt);
    $stmt->close();


    //commit
    mysqli_commit($con);
```

Test Data (as required in P1)

(only part of the data, which is used for SQL queries in next part, is shown)

Student

A	B	C	D	E	F	G	H	I	J	K	L	M	N
sid	sname	pswd	email	phone	interest	resumeFile	resumeText	university	qualification	major	gpa	startDate	endDate
1	Shangwen Yan	sy2160	sy2160@nyu.edu	3473207075	NULL	NULL	017 U Processeddatatoana	New York University	MS	CE	3.8	2017/9/1	2019/5/1
2	Ruochen Fang	ruochen	ruochen@gmail.com	3252399898	NULL	NULL	bout music. Establish	Carnegie Mellon University	MS	SE	3.51	2018/1/1	2019/5/1
3	Xinyu Ma	xm546	xm546@nyu.edu	3253773345	NULL	NULL	ling/downloading, discus	New York University	MS	CE	3.7	2017/9/1	2019/5/1
4	Linya Gong	lg2740	lg2740@nyu.edu	3209847899	NULL	NULL	, implemented load balar	Nankai University	BS	EE	3.1	2016/9/1	2018/5/1
5	Runqi Du	rd233	rd233@nyu.edu	2372342390	NULL	NULL	luate Impact Investment	Tianjing University	BS	MOT	2.9	2016/9/1	2018/5/1
6	Yan Li	yl4752	yl4752@nyu.edu	3283783378	NULL	NULL	ppender function in log	New York University	MS	CE	4	2017/9/1	2019/5/1

Company

cid	cname	pswd	email	phone	location	industry	magnitude
1	Google	Google	googlehr@google.com	3473307987	New York	Internet	8000
2	Amazon	Amazon	amazonhr@amazon.com	3242787976	Seattle	Internet	9000
3	Facebook	Facebook	facebookhr@facebook.c...	2897688976	Mello Park, CA	Internet	13000
4	Walmart	Walmart	walmarthr@walmart.com	3478978867	San Bruno, California	Retail	3000
5	Microsoft	Microsoft	microsofthr@microsoft.c...	2787637787	Redmond, WA	Computer Software	15000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Follow

sid	cid	followTime
1	3	2018-02-14 00:00:00
1	5	2018-04-14 00:00:00
2	5	2018-04-11 00:00:00
6	5	2017-04-15 00:00:00

Job

jid	cid	location	title	salary	requirement	desc	endTime	postTime
1	5	New York City, NY, US	Data & Applied Scientist	11	MS in CS	Assisting in statistical debugging, core algorithm...	2018-04-14 00:00:00	2018-02-14 00:00:00
2	2	Cambridge, MA, US	Software Developer	15	MS in CS	Participate in the development and maintenanc...	2018-05-19 00:00:00	2018-04-14 00:00:00

Relationship

sid1	sid2
2	1
3	1
6	1
1	2
1	3
1	6

Queries & Results(as required in P1

(1)

```
# 1. Create a record for a new student account, with a name, a login name, and a password.
insert ignore into Student(sname,email,pswd)
values('Shangwen Yan','sy2160@nyu.edu','sy2160');
```

(2)

```
# 2. List the names of all friends of a particular user (e.g. user with sid = 1)
select sname
from Student inner join RelationShip on Student.sid = RelationShip.sid2
where sid1 = 1;
```

Result Grid

sname
Ruochen Fang
Xinyu Ma
Yan Li

(3)

```
# 3. Delete all friendship requests that are older than one month and that have not been answered.
delete from RelationRequest
where TIMESTAMPDIFF(month,requestTime,now())>=1 ;
```

(4)

```
# 4. List all students from NYU that are following Microsoft.
select distinct sid, sname
from Follow natural join Student inner join Company
using(cid)
where university = 'New York University' and cname = 'Microsoft';
```

Result Grid

sid	sname
1	Shangwen Yan
6	Yan Li

(5)

```
# 5. List all job announcements posted in the last week that are looking for someone with an MS in CS.
select * from Job
where TIMESTAMPDIFF(day,postTime,now())<=7 and requirement like '%MS%CS%';
```

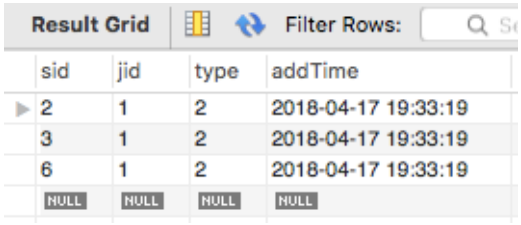
Result Grid | Filter Rows: Search | Edit: | Export/Import:

jid	cid	location	title	salary	requirement	desc	endTime	postTime
2	2	Cambridge, MA, US	Software Developer	15	MS in CS	Participate in the development and maintenanc...	2018-05-19 00:00:00	2018-04-14 00:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

(6)

```
# 6. For each student with GPA > 3.5 whose resume contains the keywords “database systems”, create a
# notification telling the student about a particular new job announcement that a company has posted
# e.g. : jid = 1
insert ignore into JobAlert
    select sid,1,2,now() from Student
    where resumeText like '%database systems%' and gpa > 3.5;
```

result shown in JobAlert:



sid	jid	type	addTime
2	1	2	2018-04-17 19:33:19
3	1	2	2018-04-17 19:33:19
6	1	2	2018-04-17 19:33:19
NULL	NULL	NULL	NULL