

NYU Tandon School of Engineering

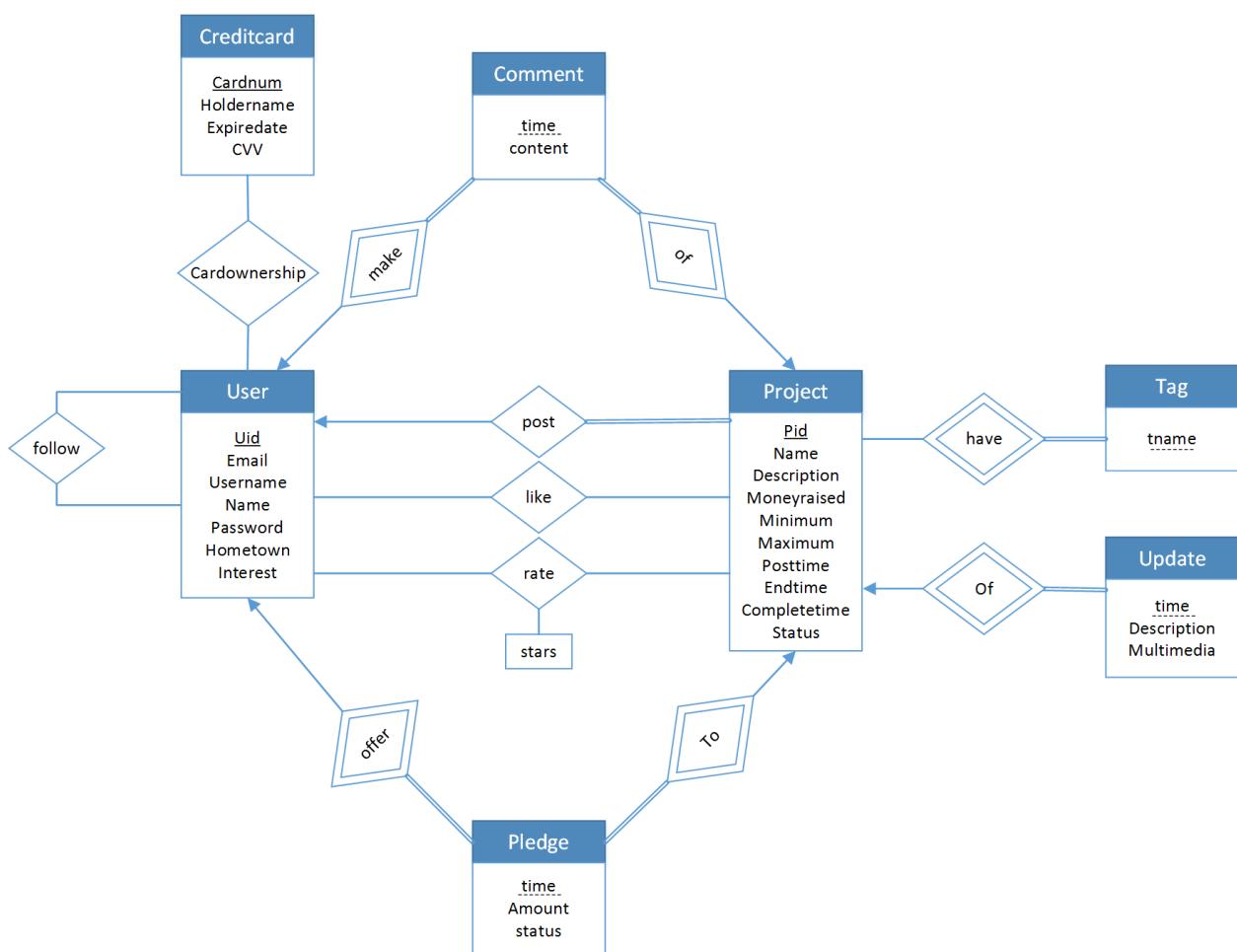
Computer Science and Engineering

CS6083, Spring 2017

zc950 Zexing Cheng

yz3330 Yuqi Zhang

ER diagram



Explanation of our design decisions:

In this design, the table USER is connected to table PROJECT with three relationships, which indicates that users can post their own projects, rate the projects they pledged and like the projects. Since a user can post more than one project while a project can only belong to one owner, the relationship POST is a one-to-many relationship. And it is obvious that the relationships RATE and LIKE should be a many-to-many relationship. The attribute (status) in PROJECT can be ‘ongoing’, ‘succeeded’ and ‘failed’ to indicate the status of projects.

In reality, a user may own several credit cards and when he/she gives money to projects, he/she can choose a credit card from those he/she stored in system. Besides this, one credit card can be used by more than one person. For example, a son may use his father’s credit card. So, it is efficient to use an extra table to store the information about credit cards, like card numbers, holder name, expired date and CVV, which are necessary to complete payment. And we use a relationship CARDOWNERSHIP to store the connection from USER to CREDITCARD. If we chose to store credit cards in USER, it would waste the space because we need to insert all information about the user whenever he/she adds a new card to his/her account.

The table COMMENT and PLEDGE are analogous in our design. They are all weak entities. Table COMMENT stores user’s comments about projects. Table PLEDGE stores the amount of money given from users to projects. In this way, it is simple to manage that a user may give money to a project more than one time. Besides this, the attribute (status) stores the status of pledges. If the project is successfully funded, the status of pledges will become ‘released’, recording a charge is completed. Then the system will complete the charge, which doesn’t need us to model. While if the project

fails to be funded, the status will become ‘refund’, indicating the money will be given back to sponsors. Besides this, the status is ‘pending’ when the project is still raising money before it is successfully funded or the time is expired.

Table UPDATE is a weak entity used to store updates posted by owners of projects. It can store text or multimedia or both. Table TAG is a weak entity as well. At first, we use an attribute (tag) in table PROJECT to store tags or categories of the project. However, one project can have several tags. For example, a project about remixes of Justin Bieber’s songs is posted, then the tags can be ‘remix’, ‘pop’ and ‘Justin Bieber’. If we store tags in table PROJECT, then every new tag will lead to insert all information about the project. So, we use an extra table to store tags.

At last, we use a relationship FOLLOW to store the information that users may follow others.

Database Schema

USER (uid, email, username, name, password, hometown, interest)

CREDITCARD (cardnum, holdername, expiredate, CVV)

CARDOWNERSHIP (uid, cardnum)

PROJECT (pid, uid, name, description, moneyraised, minimum, maximum, posttime, endtime, completeme, status)

UPDATE (pid, time, description, multimedia)

FOLLOW (uid, followerID)

COMMENT (uid, pid, time, content)

RATE (uid, pid, stars)

LIKE (uid, pid)

TAG (pid, tname)

PLEDGE (uid, pid, time, amount, status)

Foreign key constraints:

(uid) in CARDOWNERSHIP references to (uid) in USER;

(cardnum) in CARDOWNERSHIP references to (cardnum) in CREDITCARD;

(uid) in PROJECT references to (uid) in USER;

(pid) in UPDATE references to (pid) in PROJECT;

(uid) in FOLLOW references to (uid) in USER;

(followerID) in FOLLOW references to (uid) in USER;

(uid) in COMMENT references to (uid) in USER;

(pid) in COMMENT references to (pid) in PROJECT;

(uid) in RATE references to (uid) in USER;

(pid) in RATE references to (pid) in PROJECT;

(uid) in LIKE references to (uid) in USER;

(pid) in LIKE references to (pid) in PROJECT;

(pid) in TAG reference to (pid) in PROJECT;

(uid) in PLEDGE reference to (uid) in USER;

(pid) in PLEDGE reference to (pid) in PROJECT;

Other constraints:

(status) in PROJECT: `status` ENUM('ongoing', 'succeeded', 'failed') DEFAULT

“ongoing”;

(status) in PLEDGE: `status` ENUM('pending', 'released', 'refund') DEFAULT ‘pending’.

In our design, (uid) is auto-set by increment. The other attributes are set NULL as default.

- List all projects that contain the keyword “jazz” and that are currently looking for funds, sorted in descending order by posting time.

```
SELECT pid, name, posttime
FROM project
WHERE description LIKE '%jazz%'
AND status = 'ongoing'
ORDER BY posttime DESC;
```

This is the PROJECT table.

The screenshot shows the MySQL Workbench interface with the query `select * from project;` in the SQL editor. The results grid displays 8 rows of data from the PROJECT table, ordered by posttime in descending order. The columns are: pid, uid, name, description, moneyraised, minimum, maximum, posttime, endtime, completion, and status. The data includes various jazz-related project names like NuclearJazz, BombJazz, KimJazz, etc., with their respective details.

pid	uid	name	description	moneyraised	minimum	maximum	posttime	endtime	completion	status
1	1	NuclearJazz	This is a new style of jazz. I want to use nuclear...	500	100	1000	2017-01-01	2017-04-19	2017-05-01	ongoing
2	1	BombJazz	This is a new style of jazz. I want to use regular...	100	100	200	2017-01-02	2017-04-19	2017-05-01	ongoing
3	1	KimJazz	This is a new style of jazz. I want to play jazz.	50	100	300	2017-01-01	2017-04-19	2017-05-01	ongoing
4	4	BearFight	Give me money now!	0	1	10	2017-02-01	2017-04-20	2017-05-01	ongoing
5	5	BowtoJazz	Show my respect on jazz.	10	5	10	2017-02-01	2017-04-20	2017-05-01	succeeded
6	3	Smile	Let's smile.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
7	3	Love	Let's love.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
8	3	Dance	Let's dance.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

This is the query result.

The screenshot shows the MySQL Workbench interface with the query `SELECT pid, name, posttime FROM project WHERE description like '%jazz%' and status = 'ongoing' order by posttime desc;` in the SQL editor. The results grid displays 3 rows of data from the PROJECT table, ordered by posttime in descending order. The columns are: pid, name, and posttime. The data includes three jazz-related project names: BombJazz, NuclearJazz, and KimJazz, with their respective posting times.

pid	name	posttime
2	BombJazz	2017-01-02
1	NuclearJazz	2017-01-01
3	KimJazz	2017-01-01
NULL	NULL	NULL

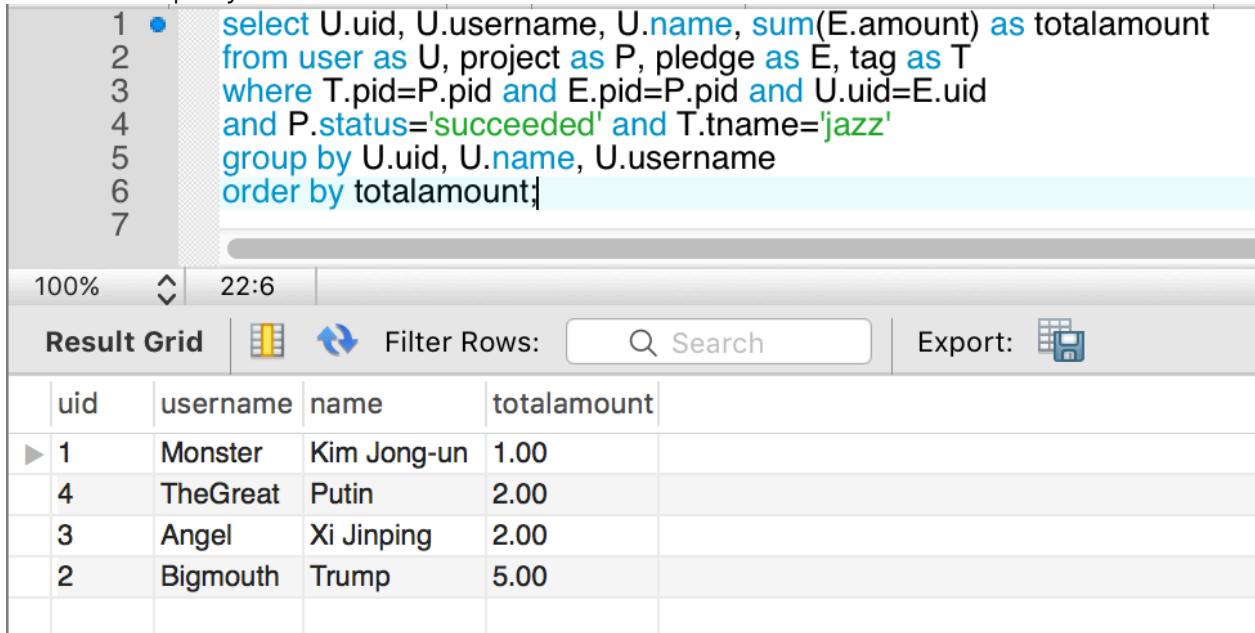
In PROJECT table, four rows contains keyword “jazz” and three of them are currently looking for funds, which are BombJazz, NuclearJazz and KimJazz. The query shows the correct result.

- List all users who have given money for projects containing the tag or category “jazz” in the past, sorted by the total amount they have successfully pledged (meaning, money that was actually charged).

```
SELECT U.uid, U.username, U.name, SUM(E.amount) AS totalamount
FROM user AS U, project AS P, pledge AS E, tag AS T
WHERE T.pid=P.pid AND E.pid=P.pid AND U.uid=E.uid
AND P.status='succeeded' AND T.tname='jazz'
GROUP BY U.uid
ORDER BY totalamount;
```

In this database, there is only one project whose tag is “jazz” and is funded successfully. This project is “BowtoJazz”, belonging to user “Abe”. Four users pledged this project, whose userIDs are 1, 2, 3 and 4.

This is the query result.



The screenshot shows a MySQL command-line interface. The query is:

```
1 •  select U.uid, U.username, U.name, sum(E.amount) as totalamount
2   from user as U, project as P, pledge as E, tag as T
3   where T.pid=P.pid and E.pid=P.pid and U.uid=E.uid
4   and P.status='succeeded' and T.tname='jazz'
5   group by U.uid, U.name, U.username
6   order by totalamount;
```

The results are displayed in a grid:

	uid	username	name	totalamount
▶	1	Monster	Kim Jong-un	1.00
	4	TheGreat	Putin	2.00
	3	Angel	Xi Jinping	2.00
	2	Bigmouth	Trump	5.00

- List all users who have completed at least 3 projects, and where each of their projects received an average rating of 4 stars or higher from its sponsors.

```
CREATE TABLE temp AS
SELECT P.pid
FROM project AS P, pledge AS E, rate AS R
WHERE P.pid = E.pid AND R.pid = P.pid
GROUP BY P.pid
HAVING AVG(stars) >=4;
SELECT U.uid, U.username, U.name
FROM project AS P, temp AS T, user AS U
WHERE P.pid=T.pid AND U.uid=P.uid
```

```

GROUP BY U.uid
HAVING COUNT(DISTINCT T.pid)>=3;
DROP TABLE temp;

```

In this database, only one user has completed three projects, whose username is “Angel”. His projects all get an average rating of 4 stars or higher.

This is the query result.

```

1 • create table temp as
2   select P.pid
3     from project as P, pledge as E, rate as R
4       where P.pid = E.pid and R.pid = P.pid
5         group by P.pid
6           having avg(stars) >=4;
7
8 • select U.uid, U.username, U.name
9   from project as P, temp as T, user as U
10  where P.pid=T.pid and U.uid=P.uid
11    group by U.uid
12      having count(distinct T.pid)>=3;
13
14 • drop table temp;
15
16

```

100% 17:14

Result Grid | Filter Rows: Search | E

uid	username	name
3	Angel	Xi Jinping

- List all comments by users that are followed by user “BobInBrooklyn”.

```

SELECT S.uid, S.username, S.name, C.content
FROM comment AS C, follow AS F, user AS U, user AS S
WHEREF.uid=C.uid AND F.followerID=U.uid AND S.uid=C.uid
AND U.username='BobInBrooklyn';

```

This is the query result.

```

1 • select S.uid, S.username, S.name, C.content
2   from comment as C, follow as F, user as U, user as S
3   where F.uid=C.uid and F.followerID=U.uid and S.uid=C.uid
4   and U.username='BobInBrooklyn';
5

```

100% 35:1

Result Grid Filter Rows: Search Export:

uid	username	name	content
2	Bigmouth	Trump	Putin is a very very very strong leader!
3	Angel	Xi Jinping	Hehe...
4	TheGreat	Putin	I need more!
5	Akita	Abe	OMG, where is my papa?

- Insert a new project for a particular user, with a name, description, and other needed info.

```

INSERT INTO `project` (`uid`, `name`, `description`, `minimum`, `maximum`, `posttime`,
`endtime`, `completetime`)
VALUES ('004', 'BearSeek', 'Help me to seek my bear!', '1', '10', '2017-2-1', '2017-4-20',
'2017-5-1');

```

This is the query result.

```

1 • insert into `project`(`uid`, `name`, `description`, `minimum`, `maximum`, `posttime`, `endtime`, `completetime`)
2   values ('004', 'BearSeek', 'Help me to seek my bear!', '1', '10', '2017-2-1', '2017-4-20', '2017-5-1');
3 • select * from project;
4

```

100% 23:3

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

pid	uid	name	description	moneyraised	minimum	maximum	posttime	endtime	completeti...	status
1	1	NuclearJazz	This is a new style of jazz. I want to use nuclear...	500	100	1000	2017-01-01	2017-04-19	2017-05-01	ongoing
2	1	BombJazz	This is a new style of jazz. I want to use regular...	100	100	200	2017-01-02	2017-04-19	2017-05-01	ongoing
3	1	KimJazz	This is a new style of jazz. I want to play jazz.	50	100	300	2017-01-01	2017-04-19	2017-05-01	ongoing
4	4	BearFight	Give me money now!	0	1	10	2017-02-01	2017-04-20	2017-05-01	ongoing
5	5	BowtoJazz	Show my respect on jazz.	10	5	10	2017-02-01	2017-04-20	2017-05-01	succeeded
6	3	Smile	Let's smile.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
7	3	Love	Let's love.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
8	3	Dance	Let's dance.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
9	4	BearSeek	Help me to seek my bear!	0	1	10	2017-02-01	2017-04-20	2017-05-01	ongoing
				HULL	HULL	HULL	HULL	HULL	HULL	HULL

It is clear that a new project is added into the table. (pid) is auto-set by increment and (moneyraised) is zero by default. (status) is "ongoing" by default.

- Insert a pledge to sponsor a project, for a particular user, project, and amount.

```
INSERT INTO `pledge` (`uid`, `pid`, `time`, `amount`)
VALUES ('4', '009', '2017-01-03', '10');
```

This is the query result.

The screenshot shows the MySQL Workbench interface. The SQL tab contains the following code:

```
1 • insert into `pledge`(`uid`, `pid`, `time`, `amount`)
2 values ('4', '009', '2017-01-03', '10');
3 • select * from pledge;
```

The Result Grid shows the contents of the PLEDGE table after the insertion:

uid	pid	time	amount	status
1	5	2017-02-03	1.00	released
2	5	2017-02-03	5.00	released
3	5	2017-02-03	2.00	released
4	1	2017-01-03	500.00	pending
4	2	2017-01-03	100.00	pending
4	3	2017-01-03	50.00	pending
4	5	2017-02-03	2.00	released
4	6	2017-02-03	10.00	released
4	7	2017-02-03	10.00	released
4	8	2017-02-03	10.00	released
4	9	2017-01-03	10.00	released
NULL	NULL	NULL	NULL	NULL

It is clear that a new record of pledge is added at the bottom of the table. Since someone pledged the project “BearSeek”, some attributes might change.

The screenshot shows the MySQL Workbench interface. The SQL tab contains the following code:

```
4 • select * from project where pid=9;
5
```

The Result Grid shows the contents of the PROJECT table after the update:

pid	uid	name	description	moneyraised	minimum	maximum	posttime	endtime	completeti...	status
9	4	BearSeek	Help me to seek my bear!	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

From this result, we can see that this project has been funded successfully, since one user just gave the maximum money of this project. So, the status tends to be “succeeded”. And this is why the status in table PLEDGE is “released”, indicating this charge is completed.

- Write queries for the end of a funding campaign. E.g., you could use triggers to detect when a campaign is fully funded or time is up; if successfully funded, generate charges to sponsors’ credit cards.

Trigger 1: this trigger is used to update (moneyraised) in table PROJECT when someone gives money to this project.

DELIMITER !

```
CREATE TRIGGER update_moneyraised AFTER INSERT ON Pledge
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Project
```

```
    SET moneyraised = moneyraised + NEW.amount
```

```

    WHERE pid = NEW.pid;
END I
DELIMITER ;

```

Trigger 2: this trigger is used to update (status) in table PLEDGE to reflect the changes of the status of project.

```

DELIMITER I
CREATE TRIGGER update_charge
AFTER update
    ON Project FOR EACH ROW
BEGIN
DECLARE sta VARCHAR(45);
SELECT status INTO sta FROM project WHERE pid=New.pid;
IF sta='succeeded' THEN
UPDATE pledge SET status='released' WHERE pid=NEW.pid;
END IF;
IF sta='failed' THEN
UPDATE pledge SET status='refunded' WHERE pid=NEW.PID;
END IF;
END I
DELIMITER ;

```

Procedure & Event: this event checks the money projects have already raised. Once the money meet the maximum amount or the end time has been expired, the status of project will be updated.

```

DELIMITER //
CREATE PROCEDURE update_status()
BEGIN
    update Project set project.status = 'succeeded' where moneyraised >= maximum;
    update Project set project.status = 'failed' where endtime < curdate() and mon-
eyraised < minimum;
    update Project set project.status = 'succeeded' where endtime < curdate() and
moneyraised >= minimum;
END //
DELIMITER ;

```

```

DROP EVENT IF EXISTS update_project;
CREATE EVENT update_project
on schedule EVERY 1 second
on completion preserve
do CALL update_status();

```

In our design, when user gives money to the project, money will be stored in a temporary place, like the account of the company of this webpage. When the project is still looking for money, the status of the project is “ongoing” and the status of the pledge is “tending”. These indicate that the money is temporarily stored now. The charge is actually generated. If the project get funded successfully, all the money given to this project will be transferred to the account of the owner of this project. The status of pledge becomes “released”, meaning the money is given to the project finally. But, if the project doesn’t get enough money before the time is up, then the money given to this project

will be given back to sponsors. The status of the pledge will become “refunded”, meaning this amount of money is given back to sponsors.

So, in our design, the status of table PLEDGE reflects the circumstance of the pledge. The result shown in prior query has proved that these triggers and events can realize that when a campaign is fully funded or time is up, the charges will be generated or declined.

Sample data:

USER:

uid	email	username	name	password	hometown	interest
1	001@nyu.edu	Monster	Kim Jong-un	123123	North Korea	Nuclear
2	002@nyu.edu	Bigmouth	Trump	121212	The United States	Twitter
3	003@nyu.edu	Angel	Xi Jinping	123456	China	Anti-corruption
4	004@nyu.edu	TheGreat	Putin	111111	Russia	Reappointment
5	005@nyu.edu	Akita	Abe	123321	Japan	Bow
6	006@nyu.edu	BobInBrooklyn	Fake	123321	Brooklyn	Girls
7	22@nyu.edu	NULL	temp	111111	NULL	NULL

PROJECT:

pid	uid	name	description	moneyraised	minimum	maximum	posttime	endtime	completeti...	status
1	1	NuclearJazz	This is a new style of jazz. I want to use nuclear...	500	100	1000	2017-01-01	2017-04-19	2017-05-01	ongoing
2	1	BombJazz	This is a new style of jazz. I want to use regular...	100	100	200	2017-01-02	2017-04-19	2017-05-01	ongoing
3	1	KimJazz	This is a new style of jazz. I want to play jazz.	50	100	300	2017-01-01	2017-04-19	2017-05-01	ongoing
4	4	BearFight	Give me money now!	0	1	10	2017-02-01	2017-04-20	2017-05-01	ongoing
5	5	BowtoJazz	Show my respect on jazz.	10	5	10	2017-02-01	2017-04-20	2017-05-01	succeeded
6	3	Smile	Let's smile.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
7	3	Love	Let's love.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
8	3	Dance	Let's dance.	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded
9	4	BearSeek	Help me to seek my bear!	10	1	10	2017-02-01	2017-04-20	2017-05-01	succeeded

CREDITCARD:

cardnum	holdername	expiredate	CVV
1232123212321232	Putin	2020-01-01	444
1233123312331233	Xi Jinping	2020-01-01	333
1234123412341234	Kim Jong-un	2020-01-01	111
1235123512351235	Trump	2020-01-01	222

CARDOWNERSHIP:

uid	cardnum
4	1232123212321232
3	1233123312331233
1	1234123412341234
2	1235123512351235

COMMENT:

uid	pid	time	content
2	4	2017-04-01	Putin is a very very very strong leader!
3	5	2017-04-01	Hehe...
4	4	2017-04-01	I need more!
5	1	2017-04-01	OMG, where is my papa?

FOLLOW:

uid	followerID
1	6
2	6
3	6
4	6
5	6

LIKE:

uid	pid
2	5

PLEDGE:

uid	pid	time	amount	status
1	5	2017-02-03	1.00	released
2	5	2017-02-03	5.00	released
3	5	2017-02-03	2.00	released
4	1	2017-01-03	500.00	pending
4	2	2017-01-03	100.00	pending
4	3	2017-01-03	50.00	pending
4	5	2017-02-03	2.00	released
4	6	2017-02-03	10.00	released
4	7	2017-02-03	10.00	released
4	8	2017-02-03	10.00	released
4	9	2017-01-03	10.00	released

RATE:

uid	pid	stars
4	6	5
4	7	5
4	8	5

UPDATE:

pid	time	description	multimedia
1	2017-04-17	I may fall, guys. I cannot control the bomb from blowing on my country.	HULL

TAG:

pid	tname
1	jazz
2	jazz
3	jazz
5	jazz

Page Logic:

Users can browse index page and view the detail of projects we represent on the homepage, but they cannot make comments, donate to the projects or rates the projects unless they have logged in.

Users should be able to sign up or log in on the index page and the project page. So we provided the user interface on the navigation bar.

Users should be able to start a new project or visit his or her personal page after they have logged in. So we provided the user interface on another navigation bar.

Users should have their personal home pages, where they can modify their profiles and display all kind of information about themselves.

Every project should be displayed on a project pages, users can modify their own project pages as well as visit the project pages created by the others.

Specific page design and implement:

Index page: Index page is prepared for both the guests and the users, we use a carousel design to display the most important information about our website, below which we provide three projects randomly recommended by us for users to explore.

Navigation bar:

Every page should have the navigation bar with UI consistency, but the user interface provided by the navigation bar could be different depending on the state of user: being login, being log out. So we extracted the code of navigation bar out as two independent file.

```
<?php
if(isset($_SESSION['login']) && $_SESSION['login']==true)
{
    include "nav-login.php";
}
else
{
    include "nav-sign.php";
}
?>
```

Code above is used to decided with navigation bar file should be included.

Project page:

Project page is one of the most important pages, which is prepared for both the guests and the users. However, the content could be different, for example, our users can make comments to that project, like the project, donate to the project and so on. But the guests can only browse the information about that project.

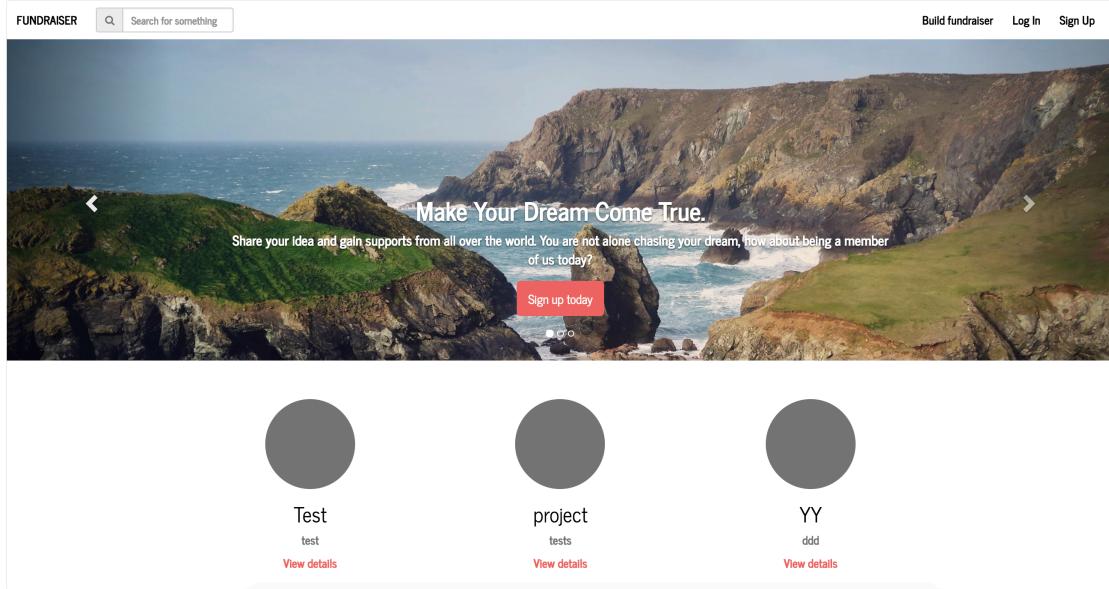
We achieved that by constructing the page dynamically, below is an example.

```
<?php
if(isset($_SESSION['login'])){
    echo "<form action='postcom.php' method='POST'>";
    echo "<textarea class='form-control' rows='1' name='title' placeholder='Title' ></textarea>";
    echo "<textarea class='form-control' rows='6' name='content' placeholder='Comment'></textarea>";
    echo "<br>";
    echo "<input class='btn btn-info' type='submit'>";
    echo "</form>";
}
?>
</div>
```

When the user have logged in, he or she will be able to make comments to the project, but if the guest have not logged in, the comments interface won't be shown on the html page sent back to he or her.

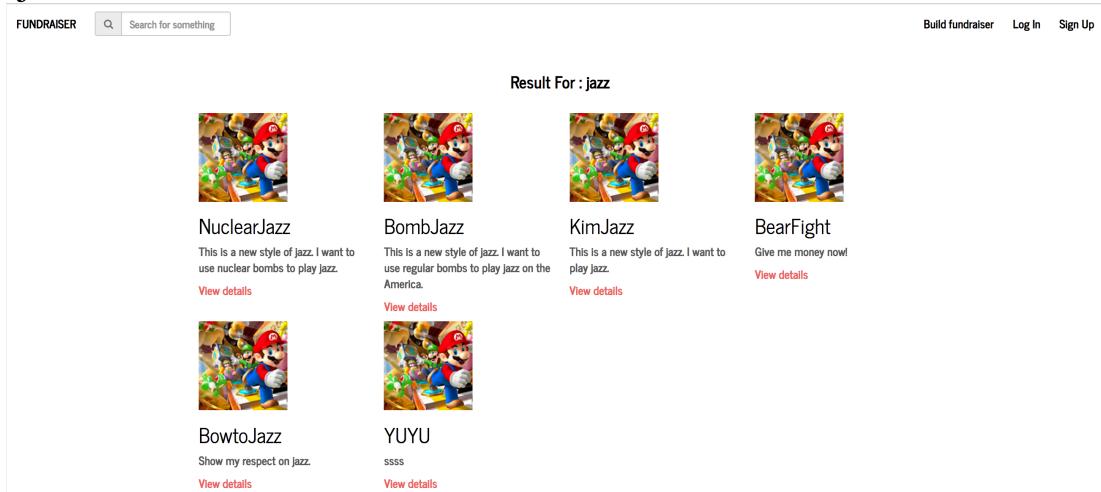
Usage:

When a user visits my systor the first timeem f, he/she will see the main page as picture following:

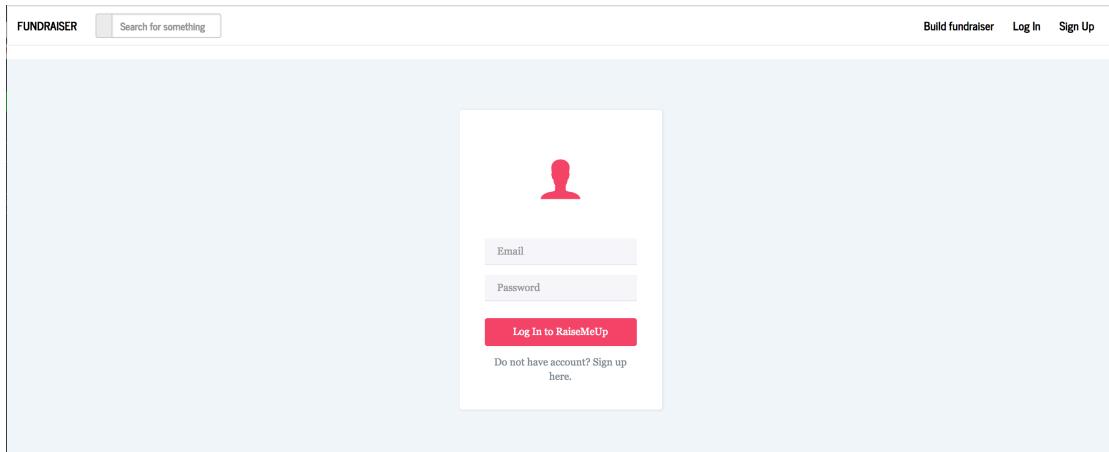


Basically, the main page has three parts from top to bottom.
First part is the navigation bar.

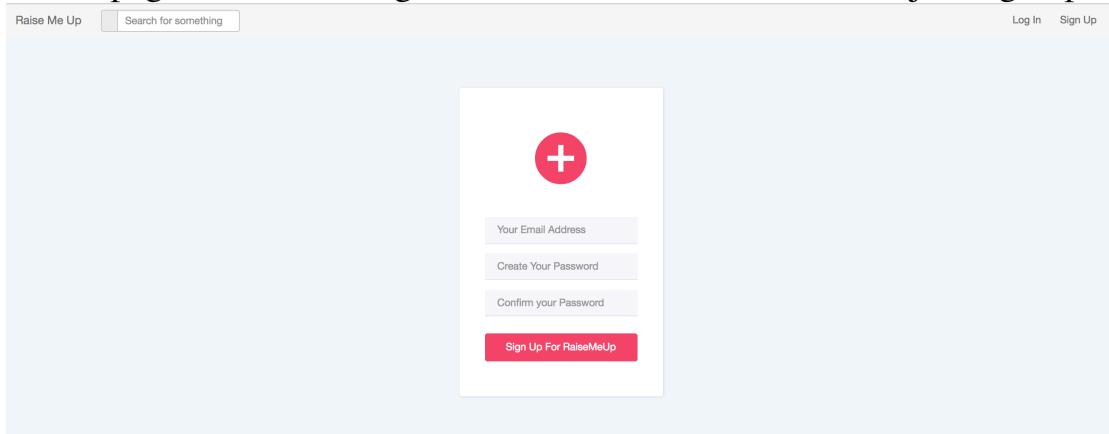
User can click 'FUNDRAISER' to return to this page at whichever page he/she is. He/she can search for projects by typing some key words he/she is interested in. For example, if he/she want to find out any projects concerns with 'jazz', he/she can type 'jazz' in search bar and the result will be like this:



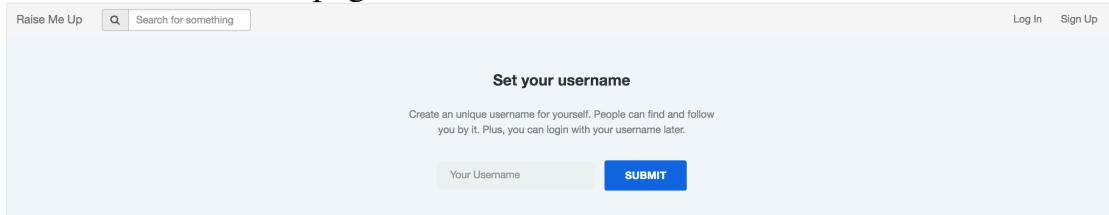
Any projects containing the string 'jazz' at their name or description or tags will show up because of search. By the way, if the user inputs nothing and search, all projects will show up. Since the user is a guest for now, he/she need to login or sign-up before he/she posts a project or donate. If he/she is guest and clicks 'BUILD FUNDRAISER', the page will be redirected to a login page.



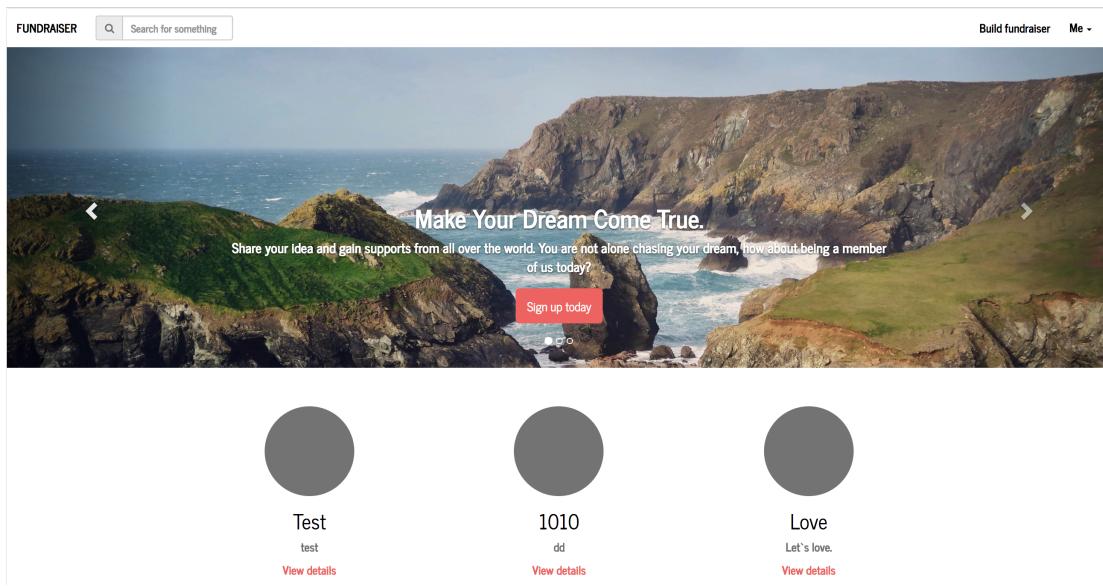
In this page, he/she can login in if an account is available or just sign up.



When he/she signs up, he/she needs to input the email and password to create an account. If the email is valid and never used at my system before and two passwords are same, the page will be redirected to another one to set username.



The username is set to be unique, so if the username isn't occupied, the user will find out he/she is at main page again except for the change at the navigation bar.



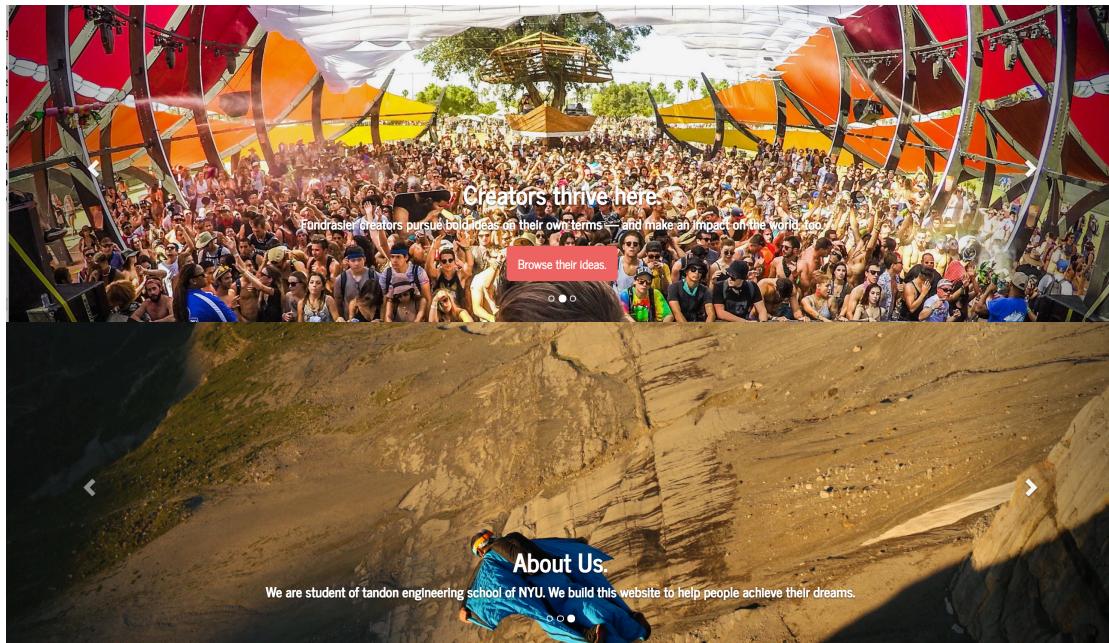
It is clear that no ‘login’ and ‘signup’ remain at nav-bar, the user can click ‘me’ to enter personal page or log out.

Everything is good for now. But if when the user inputs an email occupied already or mis-input password or just uses a username used by others, alert will turn out to tell the user what is going wrong.

Now, assuming the user has signed up or logged in successfully, he/she will be returned to the main page as said before.

The second part of the main page is combined with three big picture.





First one contains a link to sign-up page. The second one contains a link to show all projects in my system. And the third one contains info about my partner and me. The third part of the main page is three projects randomly fetched from database.



By clicking the 'VIEW DETAILS', the user can enter in a project page.

FUNDRAISER Search for something

Build fundraiser Me -

Project / Updates



BearFight
Post by [TheGreat](#)
Give me money now
[Like](#)

Project Progress

Project Popularity: Hot	30%
Raised: \$3	Goal: \$10
3 Donations	Ends 2017-04-20

Donation

Donor	Amount	Time
BobInBrooklyn	1.00	2017-05-09
Test	1.00	2017-05-09
Bigmouth	1.00	2017-05-08

Rate

Rate 1 to 5.
Submit

Reviews

Cool
Putin is a very very very strong leader!
[Bigmouth](#) 2017-04-01

More
I need more!
[TheGreat](#) 2017-04-01

Since this project has successfully gotten minimum money and the time has expired, no donation option appears in this page. The user can figure out how much money the project has gotten and who donated to it. A 'RATE' option appears because the user I logged in is the one who donated before with username 'BIGMOUTH'.

Rate**Rate**

5

[Submit](#)

Bigmouth

These two pictures show that I just rated a project with 5 stars.

At the bottom of the page, the user can see comments posted before and he/she can post a comment.

Reviews

Cool

Putin is a very very very strong leader!

Bigmouth 2017-04-01

More

I need more!

TheGreat 2017-04-01

Title	
Comment	

提交

The user also can ‘like’ the project so he/she can get update information about this project at his/her personal page.

The screenshot shows a user profile page for 'Xi Jinping'. The profile section includes basic information: Name: Xi Jinping, Hometown: China, Interest: Anti-corruption, and an 'Edit' button. Below this is a 'Likes' section listing 'KimJazz', 'BowtoJazz', 'Smile', 'Love', 'BearSeek', and 'Test'. To the right, there's a 'Projects / Messages' tab. Under 'Likes', there are three circular icons labeled 'Smile', 'Love', and 'Dance' with their respective descriptions: 'Let's smile.', 'Let's love.', and 'Let's dance.' Each like has a 'View details' link. Below the likes is a 'Rates' section showing a rating of '★★★★★' for 'BowtoJazz'. The final section is 'Comments' with entries 'hehe' and 'Hehe...'. A small Mario character icon is visible at the top right of the main content area.

Here is the personal page of another user. From this page, the user can get to any projects he/she liked or posted or rated or commented. For example, he/she want to get to the project with the name ‘LOVE’.

Love
Post by Angel
Let's love.
[Like](#)

Project Progress

Project Popularity: Hot

Raised: \$10 Goal: \$10
100%
1 Donations Ends 2017-04-20

Now get back to personal page,

Profile
Name: Xi Jinping
Hometown: China
Interest: Anti-corruption
[Edit](#)

Update
Well done, guys
[Love](#) 2017-05-09
Update
Project is sooooo hard
[Test](#) 2017-05-09
New Projects
[Test](#)
[Test](#) 2017-05-09

Likes
KimJazz
BowtoJazz
Smile
Love
BearSeek
Test

This page can also show messages about any one he/she followed and projects he/she liked. By clicking the name of another user, he/she can enter other's personal page.

Profile
Name: Test
Hometown: NY
Interest: HAHA

Likes
NuclearJazz
BombJazz
BearFight
Love
Test

Test
test
[View details](#)

Like this one with name ‘TEST’, the user can see information about this user and follow this user.

When he/she decides to post a project, he/she can click “BUILD FUNDRAISER” at the nav-bar.

The screenshot shows a web application interface for creating a project. At the top, there are navigation links for 'FUNDRAISER', 'Search for something', 'Build fundraiser', and 'Me'. The main area is titled 'Publish your Project' and contains several input fields:

- Name: [Input field]
- Minimum amount of money in need: [Input field]
- Maximum amount of money in need: [Input field]
- End time For Fundraising: [Input field] (with placeholder '年 /月/日')
- Completion time For Project: [Input field] (with placeholder '年 /月/日')
- Description: [Input field]

A blue 'CREATE' button is located at the bottom right of the form.

He/she needs to input some information about the project as the picture shows.

This screenshot shows a project creation form with sample input values:

- Name: HAHA
- Minimum amount of money in need: 10
- Maximum amount of money in need: 20
- End time For Fundraising: 2017/06/02
- Completion time For Project: 2017/06/04
- Description: game

A blue 'CREATE' button is located at the bottom right of the form.

As the picture show, I just created a project with name ‘HAHA’ and tag ‘GAME’. After this, the user is redirected to the project page he/she just builds.

FUNDRAISER Search for something Build fundraiser Me ▾

Project / Updates



Haha
Post by Angel
dfsfsfsfsfsfs
[Donate](#) [Like](#)

Project Progress

Project Popularity: Hot

Raised: \$0 Goal: \$20
0 Donations Ends 2017-06-02

Donation

Donor	Amount	Time

He/she can post updates like this:

FUNDRAISER Search for something Build fundraiser Me ▾

Project / Updates

Updates

提交

Fund Raiser @ 2016 [f](#) [t](#) [g](#)

FUNDRAISER Search for something Build fundraiser Me ▾

Project / Updates

2017-05-11 Just for test

Updates

提交

Or donate, like this:

Project Progress

Project Popularity: Hot

Raised: \$0 Goal: \$20
0 Donations Ends 2017-06-02

Support This Project

Once the project successes, money will be released to it, otherwise we will refund your money.

Choose card: ****1233 / [Add New Card](#)

10 [DONATE](#)

Or add a new card, like this:

Payment Details

CARD NUMBER
1111222222222222

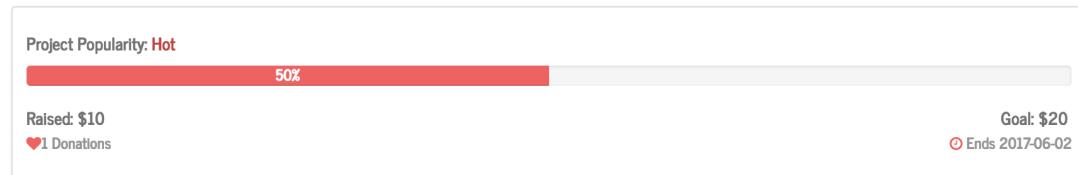
MONTH 12 **YEAR** 2020 **CV CODE** 112

CARD HOLDER
HAHA

提交

Finally, the donation is completed, like this:

Project Progress



Donation

Donor	Amount	Time
Angel	10.00	2017-05-11