Project #2 – Group AN168

Muyao Wang mw4086 Student ID: N12430279

Yiming Zhang yz5293 Student ID: N14388761

Introduction

Snickr is a web application that allows users to register and login, build workspaces, create channels inside workspaces, and invite others into workspaces or channels; and then respond to invitations and send messages to communicate with channel members.

Relational Schema

User(uid, uname, nickname, upassword, email)

Workspace(wid, wname, creatorid, description, wtime)

Workspace.creatorid is a foreign key reference to User.uid

WorkspaceMembership(uid, wid, wjointime)

WorkspaceMembership.uid is a foreign key reference to User.uid

WorkspaceMembership.wid is a foreign key reference to Workspace.wid

WorkspaceInvitation(winvitorid, wid, winviteeemail, winvitetime, wstatus)

WorkspaceInvitation.(winvitorid,wid) is a foreign key reference to WorkspaceMembership.(uid,wid)

Administrator(uid, wid)

Administrator.(uid,wid) is a foreign key reference to WorkspaceMembership(uid,wid)

Channel(cid, creatorid, wid, ctype, cname, ctime)

Channel.(creatorid,wid) is a foreign key reference to WorkspaceMembership(uid,wid)

ChannelMembership(uid, cid, cjointime)

ChannelMembership.uid is a foreign key reference to User.uid

ChannelMembership.cid is a foreign key reference to Channel.cid

ChannelInvitation(cinvitorid, cid, cinviteeid, cinvitetime, cstatus)

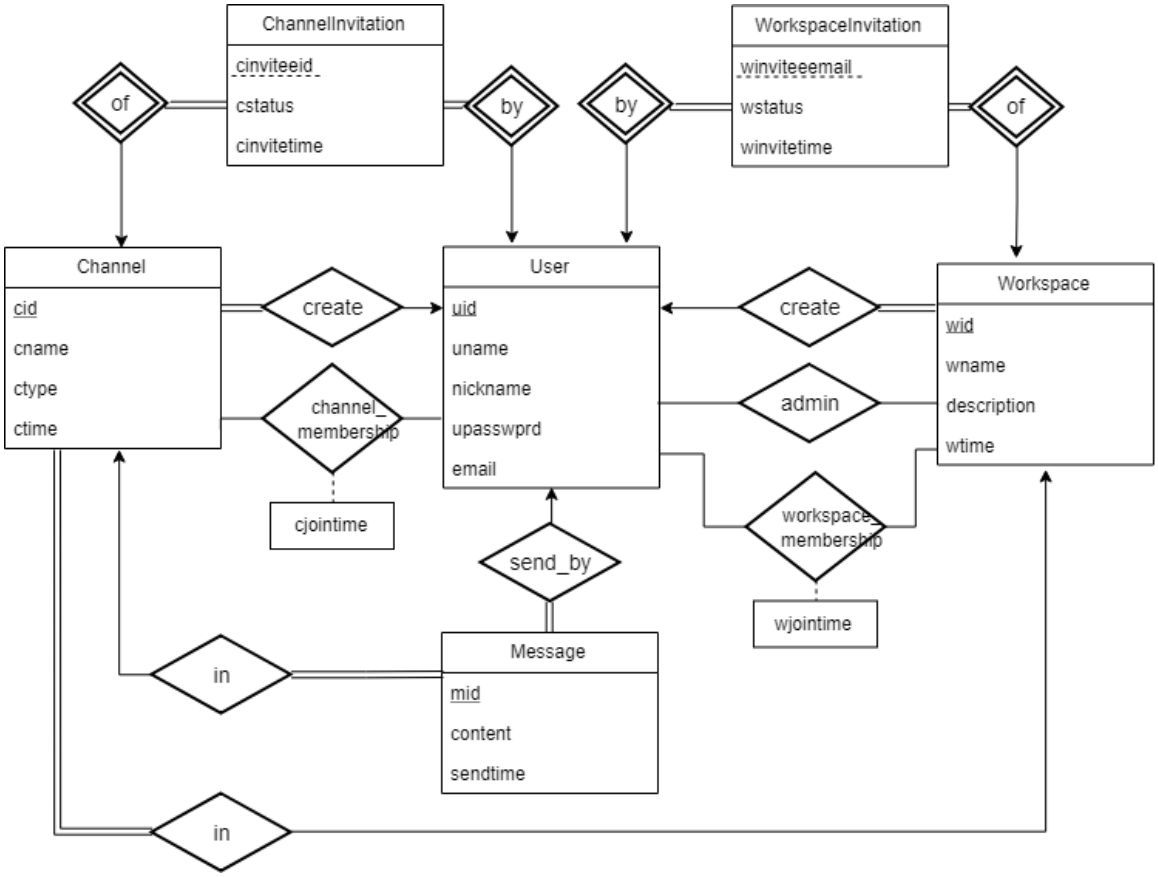
ChannelInvitation.(cinvitor,cid) is a foreign key reference to ChannelMembership.(uid,cid)

ChannelInvitation.cinviteeid is a foreign key reference to User.uid

Message(mid, senderid, cid, content, sendtime)

Message.(senderid,cid) is a foreign key reference to Channelmembership(uid,cid)

ER diagram



Explanations and Assumptions of Schema Design

1. User

   In this table, uid is a unique id of each user, which used to identify each user. The uname is username and nickname is nickname and upassword is the password and email is the registered email.

2. Workspace

   In this table, wid is a unique id of each workspace, which used to identify each workspace. The wname is the name of workspace, and the creatorid is the id of the user who creates this workspace. The description stores information of workspace, and wtime is the time when the workspace is created.

3. WorkspaceMembership

   In this table, uid is the id of the user that belongs to this workspace; wid is the id of this workspace, wjointime is the time when is user joins this workspace.

4. WorkspaceInvitation

In this table, winvitorid is the id of the user that sends the invitation; wid is the id of the workspace, winviteeemail is the email of the invitee, winvitetime is when the invitation is sent and wstatus stores the status of the invitation which indicate whether the invitee accept the invitation or not.

5. Channel

   In this table, cid is a unique id of each channel, which used to identify each channel. And creatorid is the id of the user who creates the channel; the wid is the id of the workspace that this channel belongs to. Ctype field indicates the type of this channel, which can be public, private, and direct, cname is the name of this channel, and ctime is the creation time of this channel.

6. Administrator

   In this table, uid and wid is the admin user id and the workspace id that this user adminidters.

7. ChannelMembership

   In this table, uid is the id of the user that belongs to this channel, cid is the id of this channel, cjointime is the time when is user joins this channel.

8. ChannelInvitation

   In this table, cinvitorid is the id of the user that sends the invitation, cid is the id of the channel, cinviteeid is the id of the invitee, cinvitetime is when the invitation is sent and cstatus field stores status of the invitation, which indicates whether the invitee accept the invitation or not. If the invitee does not react to the invitation, it will be pending at the status field.

9. Message

   In this table, mid is the unique id of each message, which used to identify each message, senderid is the uid of user that sends this message, cid is the id of the channel that this message belongs to, content is the message content, and sendtime is when the message is sent.

Procedures

1. Log in/Sign up by email

   A user can log in or sign up in the front-end by clicking the corresponding tablet and input the information into the input boxes at /index.html. The frontend will then send the user's input messages to the database to check whether the input information is correct. If there exists the account the user tries to log in, the front-end will allow the user to join; if there doesn't exist, the front-end will alter the message. If there doesn't exist the account the user tries to register, the database will add one line into the User table, and the user will register this new account and returns to the main page as logged in.

2. Create workspace

   A user can create a workspace in the front-end at /main.html. The frontend will send related information to the database and add relevant information to the Workspace table.

3. Invite members into workspace

   A user can invite other members to the workspace he belongs to at /workspace.html. The database will add records to the WorkspaceInvitation table.

4. Respond to workspace invitation

   Anyone can receive email notification of invitations into a workspace. If the invitee is not a user yet, he has to register as a user first; then he can choose to accept or reject the invitation at /inviation.html. If he accepts, the related record in the WorkspaceInvitation table will be updated, and one record for his membership in this workspace will be inserted into the WorkspaceMembership table. If he rejects, only the related record in the WorkspaceInvitation table will be updated.

5. Add/remove administrators for workspace

   The workspace creator can assign or remove additional admins for the workspace he manages at /workspace.html. The database will add records to the Administrator table.

6. Delete user from workspace

   The workspace creator and admins can delete a user who is in the workspace that he manages at /workspace.html. The database will delete records in the WorkspaceMembership table and all other related information on delete cascade.

7. Create channel

   Any user can create channels in the workspace that he belongs to at /workspace.html. The frontend will send related information to the server, and database will add one line to the Channel table.

8. Invite members into channels

   A user can invite others into channels that he belongs to at /channel.html. The frontend will check the channel's type to see whether this invitation is legal (for public and private channel, there is no limitation on invitation; for direct channel, the server will check the number of users inside the channel and whether it is illegal to invite other people). If so, the front-end will show the list of users who are in the same workspace with the channel that can be invited. The inviter can choose the users he wants to invites, and the server will send notifications to the invitees and database will add records to ChannelInvitation table.

9. Respond to channel invitation

   A user can receive the notification of invitations into channel at /invitation.html. He can choose to accept or reject the invitation. If he accepts, the related record in the ChannelInvitation table will be updated, and one record for his membership in this channel

will be inserted into the corresponding table. If he rejects, only the related record in the ChannelInvitation/WorkspaceInvitation table will be updated.

10. Send Message in the channel

    A user can send messages in channels that he belongs to at /channel.html. The messages will be sent out and seen on the /channel.html, and the database will add records into Message table.

11. Search messages with keyword

    A user can search for messages with a keyword in /search.html. The messages he has access to and contains the keyword will appear in the search result.

12. Change user nickname

    A user can see his personal information in /profile.html including his name, email address, and nickname. He can change his nickname in the front-end and the server will update the information in the User table.

Build database in MySQL

```
drop database snickr;
create database snickr;
use snickr;

CREATE TABLE User (
    uid INT AUTO_INCREMENT PRIMARY KEY,
    uname VARCHAR(50) NOT NULL,
    nickname VARCHAR(50),
    upassword VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL
);

CREATE TABLE Workspace (
    wid INT AUTO_INCREMENT PRIMARY KEY,
    wname VARCHAR(50) NOT NULL,
    creatorid INT,
    description VARCHAR(200),
    wtime TIMESTAMP,
    FOREIGN KEY (creatorid)
        REFERENCES User (uid)
        ON DELETE CASCADE
);

CREATE TABLE WorkspaceMembership (
    uid INT,
    wid INT,
    wjointime TIMESTAMP,
    PRIMARY KEY (uid , wid),
    FOREIGN KEY (uid)
        REFERENCES User (uid)
        ON DELETE CASCADE,
    FOREIGN KEY (wid)
        REFERENCES Workspace (wid)
        ON DELETE CASCADE
);

CREATE TABLE WorkspaceInvitation (
    winvitorid INT NOT NULL,
    wid INT NOT NULL,
    winviteeemail VARCHAR(50) NOT NULL,
    winvitetime TIMESTAMP,
    wstatus ENUM('accept', 'reject', 'pending'),
    PRIMARY KEY (winvitorid , wid , winviteeemail),
    FOREIGN KEY (winvitorid , wid)
        REFERENCES WorkspaceMembership (uid , wid)
        ON DELETE CASCADE
);

```

```sql
CREATE TABLE Channel (
    cid INT AUTO_INCREMENT PRIMARY KEY,
    creatorid INT,
    wid INT,
    ctype ENUM('public', 'private', 'direct'),
    cname VARCHAR(50) NOT NULL,
    ctime TIMESTAMP,
    FOREIGN KEY (creatorid , wid)
        REFERENCES WorkspaceMembership (uid , wid)
        ON DELETE CASCADE
);

CREATE TABLE Administrator (
    uid INT,
    wid INT,
    PRIMARY KEY (uid , wid),
    FOREIGN KEY (uid , wid)
        REFERENCES WorkspaceMembership (uid , wid)
        ON DELETE CASCADE
);

CREATE TABLE ChannelMembership (
    uid INT,
    cid INT,
    cjointime TIMESTAMP,
    PRIMARY KEY (uid , cid),
    FOREIGN KEY (uid)
        REFERENCES User (uid)
        ON DELETE CASCADE,
    FOREIGN KEY (cid)
        REFERENCES Channel (cid)
        ON DELETE CASCADE
);

CREATE TABLE ChannelInvitation (
    cinvitorid INT NOT NULL,
    cid INT NOT NULL,
    cinviteeid INT NOT NULL,
    cinvitetime TIMESTAMP,
    cstatus ENUM('accept', 'reject', 'pending'),
    PRIMARY KEY (cinvitorid , cid , cinviteeid),
    FOREIGN KEY (cinvitorid , cid)
        REFERENCES ChannelMembership (uid , cid)
        ON DELETE CASCADE,
    FOREIGN KEY (cinviteeid)
        REFERENCES User (uid)
        ON DELETE CASCADE
);

CREATE TABLE Message (
    mid INT AUTO_INCREMENT PRIMARY KEY,
    senderid INT,
    cid INT,
    content VARCHAR(500) NOT NULL,
    sendtime TIMESTAMP,
    FOREIGN KEY (senderid , cid)
        REFERENCES Channelmembership (uid , cid)
        ON DELETE CASCADE
);
```

Insert sample data into the database

1. User Table

```sql
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('1', 'John', 'JJ', 'jj123', 'john@gmail.com');
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('2', 'Rose', 'rosy', 'Rosy876', 'oRose3@gmail.com');
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('3', 'Tompson', 'TommyP', 'Tom123', 'TTP@hotmail.com');
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('4', 'Linda', 'NULL', 'Lin862', 'Linda@nyu.edu');
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('5', 'Kevin', 'KKV', 'kevin111', 'Kevin1@yahoo.com');
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('6', 'Jack', 'Jacky', 'JJlin', 'JaCk082@hotmail.com');
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('7', 'Peter', 'PPT', 'TPP647', '746Peter@hotmail.com');
INSERT INTO `snickr`.`user` (`uid`, `uname`, `nickname`, `upassword`, `email`) VALUES ('8', 'Mary', 'M.', 'mmaryy', 'Mar333@yahoo.com');
```

uid	uname	nickname	upassword	email
1	John	JJ	jj123	john@gmail.com
2	Rose	rosy	Rosy876	oRose3@gmail.com
3	Tompson	TommyP	Tom123	TTP@hotmail.com
4	Linda	NULL	Lin862	Linda@nyu.edu
5	Kevin	KKV	kevin111	Kevin1@yahoo.com
6	Jack	Jacky	JJlin	JaCk082@hotmail.com
7	Peter	PPT	TPP647	746Peter@hotmail.com
8	Mary	M.	mmaryy	Mar333@yahoo.com
NULL	NULL	NULL	NULL	NULL

2. Workspace Table

```
INSERT INTO `snickr`.`workspace` (`wid`, `wname`, `adminid`, `description`, `wtime`) VALUES
('1', 'Maths Group', '3', 'The group for Maths course.', '2019-03-28 11:00:00');
INSERT INTO `snickr`.`workspace` (`wid`, `wname`, `adminid`, `description`, `wtime`) VALUES
('2', 'Computer Science', '4', 'Computer Science Department group.', '2019-03-02 15:30:24');
INSERT INTO `snickr`.`workspace` (`wid`, `wname`, `adminid`, `description`, `wtime`) VALUES
('3', 'Movie Lover', '2', 'Titanic', '2019-02-14 12:00:00');
```

wid	wname	adminid	description	wtime
1	Maths Group	3	The group for Maths course.	2019-03-28 11:00:00
2	Computer Science	4	Computer Science Department group.	2019-03-02 15:30:24
3	Movie Lover	2	Titanic	2019-02-14 12:00:00
NULL	NULL	NULL	NULL	NULL

3. WorkspaceMembership Table

```
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('3', '1', '2019-03-28 11:00:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('5', '1', '2019-03-29 15:25:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('6', '1', '2019-03-29 15:25:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('7', '1', '2019-03-28 11:02:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('2', '2', '2019-03-02 15:40:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('3', '2', '2019-03-02 15:40:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('4', '2', '2019-03-02 15:30:24');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('6', '2', '2019-03-03 12:10:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('7', '2', '2019-03-04 12:10:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('8', '2', '2019-03-04 12:10:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('2', '3', '2019-02-14 12:00:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('6', '3', '2019-02-14 12:10:00');
INSERT INTO `snickr`.`workspacemembership` (`uid`, `wid`, `wjointime`) VALUES ('8', '3', '2019-02-14 12:20:00');
```

uid	wid	wjointime
2	2	2019-03-02 15:40:00
2	3	2019-02-14 12:00:00
3	1	2019-03-28 11:00:00
3	2	2019-03-02 15:40:00
4	2	2019-03-02 15:30:24
5	1	2019-03-29 15:25:00
6	1	2019-03-29 15:25:00
6	2	2019-03-03 12:10:00
6	3	2019-02-14 12:10:00
7	1	2019-03-28 11:02:00
7	2	2019-03-04 12:10:00
8	2	2019-03-04 12:10:00
8	3	2019-02-14 12:20:00
NULL	NULL	NULL

4. WorkspaceInvitation Table

```
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('2', '3', 'JaCk082@hotmail.com', '2019-02-14 12:05:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('2', '3', 'Mar333@yahoo.com', '2019-02-14 12:18:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('3', '1', '746Peter@hotmail.com', '2019-03-28 11:00:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('3', '1', 'JaCk082@hotmail.com', '2019-03-29 14:25:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('3', '1', 'Kevin1@yahoo.com', '2019-03-29 14:25:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('4', '2', '746Peter@hotmail.com', '2019-03-04 12:08:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('4', '2', 'JaCk082@hotmail.com', '2019-03-03 12:00:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('4', '2', 'john@gmail.com', '2019-03-02 15:30:00', 'reject');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('4', '2', 'Kevin1@yahoo.com', '2019-03-03 12:40:00', 'pending');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('4', '2', 'Mar333@yahoo.com', '2019-03-04 12:08:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('4', '2', 'oRose3@gmail.com', '2019-03-02 15:30:00', 'accept');
INSERT INTO `snickr`.`workspaceinvitation` (`winvitorid`, `wid`, `winviteeemail`, `winvitetime`, `wstatus`) VALUES
('4', '2', 'TTP@hotmail.com', '2019-03-02 15:30:00', 'accept');
```

winvitorid	wid	winviteeemail	winvitetime	wstatus
2	3	JaCk082@hotmail.com	2019-02-14 12:05:00	accept
2	3	Mar333@yahoo.com	2019-02-14 12:18:00	accept
3	1	746Peter@hotmail.com	2019-03-28 11:00:00	accept
3	1	JaCk082@hotmail.com	2019-03-29 14:25:00	accept
3	1	Kevin1@yahoo.com	2019-03-29 14:25:00	accept
4	2	746Peter@hotmail.com	2019-03-04 12:08:00	accept
4	2	JaCk082@hotmail.com	2019-03-03 12:00:00	accept
4	2	john@gmail.com	2019-03-02 15:30:00	reject
4	2	Kevin1@yahoo.com	2019-03-03 12:40:00	pending
4	2	Mar333@yahoo.com	2019-03-04 12:08:00	accept
4	2	oRose3@gmail.com	2019-03-02 15:30:00	accept
4	2	TTP@hotmail.com	2019-03-02 15:30:00	accept
NULL	NULL	NULL	NULL	NULL

5. Administrator Table

```
INSERT INTO `snickr`.`administrator` (`uid`, `wid`) VALUES ('3', '1');
INSERT INTO `snickr`.`administrator` (`uid`, `wid`) VALUES ('7', '1');
INSERT INTO `snickr`.`administrator` (`uid`, `wid`) VALUES ('4', '2');
INSERT INTO `snickr`.`administrator` (`uid`, `wid`) VALUES ('2', '3');
INSERT INTO `snickr`.`administrator` (`uid`, `wid`) VALUES ('6', '3');
```

uid	wid
3	1
7	1
4	2
2	3
6	3
NULL	

6. Channel Table

```
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('1', '6', '1', 'public', 'Homework', '2019-03-29 15:30:00');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('2', '3', '1', 'private', 'Exams', '2019-03-28 11:05:10');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('3', '7', '1', 'direct', 'Solution', '2019-03-28 17:20:30');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('4', '7', '2', 'public', 'Database', '2019-03-05 12:13:14');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('5', '8', '2', 'public', 'Algorithm', '2019-03-04 12:30:17');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('6', '6', '2', 'direct', 'Salary Chatting', '2019-03-09 08:30:03');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('7', '3', '2', 'private', 'Lab Hours', '2019-03-02 19:40:37');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('8', '2', '3', 'public', 'Share Movie', '2019-02-14 12:10:00');
INSERT INTO `snickr`.`channel` (`cid`, `creatorid`, `wid`, `ctype`, `cname`, `ctime`) VALUES
('9', '2', '3', 'direct', 'Love', '2019-02-14 12:30:00');
```

cid	creatorid	wid	ctype	cname	ctime
1	6	1	public	Homework	2019-03-29 15:30:00
2	3	1	private	Exams	2019-03-28 11:05:10
3	7	1	direct	Solution	2019-03-28 17:20:30
4	7	2	public	Database	2019-03-05 12:13:14
5	8	2	public	Algorithm	2019-03-04 12:30:17
6	6	2	direct	Salary Chatting	2019-03-09 08:30:03
7	3	2	private	Lab Hours	2019-03-02 19:40:37
8	2	3	public	Share Movie	2019-02-14 12:10:00
9	2	3	direct	Love	2019-02-14 12:30:00
NULL	NULL	NULL	NULL	NULL	NULL

7. ChannelMembership Table

```
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('2', '4', '2019-03-05 12:13:45');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('2', '5', '2019-03-04 12:40:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('2', '6', '2019-03-09 09:30:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('2', '7', '2019-03-02 20:18:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('2', '8', '2019-02-14 12:10:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('2', '9', '2019-02-14 12:30:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('3', '1', '2019-03-29 15:33:10');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('3', '2', '2019-03-28 11:05:10');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('3', '3', '2019-03-30 16:15:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('3', '4', '2019-03-05 12:15:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('3', '5', '2019-03-05 12:44:08');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('3', '7', '2019-03-02 19:40:37');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('5', '2', '2019-03-28 11:06:08');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('6', '1', '2019-03-29 15:30:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('6', '4', '2019-03-05 12:14:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('6', '5', '2019-03-06 14:25:45');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('6', '6', '2019-03-09 08:30:03');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('6', '7', '2019-03-02 20:10:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('6', '8', '2019-02-14 12:13:00');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('6', '9', '2019-02-14 12:30:30');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('7', '1', '2019-03-29 15:34:08');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('7', '3', '2019-03-28 17:20:30');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('7', '4', '2019-03-05 12:13:14');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('8', '4', '2019-03-05 12:14:06');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('8', '5', '2019-03-04 12:30:17');
INSERT INTO `snickr`.`channelmembership` (`uid`, `cid`, `cjointime`) VALUES ('8', '8', '2019-02-14 12:20:10');
```

uid	cid	cjointime
2	4	2019-03-05 12:13:45
2	5	2019-03-04 12:40:00
2	6	2019-03-09 09:30:00
2	7	2019-03-02 20:18:00
2	8	2019-02-14 12:10:00
2	9	2019-02-14 12:30:00
3	1	2019-03-29 15:33:10
3	2	2019-03-28 11:05:10
3	3	2019-03-30 16:15:00
3	4	2019-03-05 12:15:00
3	5	2019-03-05 12:44:08
3	7	2019-03-02 19:40:37
5	2	2019-03-28 11:06:08
6	1	2019-03-29 15:30:00
6	4	2019-03-05 12:14:00
6	5	2019-03-06 14:25:45
6	6	2019-03-09 08:30:03
6	7	2019-03-02 20:10:00
6	8	2019-02-14 12:13:00
6	9	2019-02-14 12:30:30
7	1	2019-03-29 15:34:08
7	3	2019-03-28 17:20:30
7	4	2019-03-05 12:13:14
8	4	2019-03-05 12:14:06
8	5	2019-03-04 12:30:17
8	8	2019-02-14 12:20:10
NULL	NULL	NULL

8. ChannelInvitation Table

```
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('2', '8', '6', '2019-02-14 12:10:00', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('2', '9', '6', '2019-02-14 12:30:00', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('3', '2', '7', '2019-03-30 12:05:00', 'reject');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('3', '2', '5', '2019-03-28 11:05:30', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('3', '7', '7', '2019-03-02 19:40:37', 'reject');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('3', '7', '6', '2019-03-02 19:40:37', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('3', '7', '2', '2019-03-02 19:40:37', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('6', '1', '7', '2019-03-29 15:30:00', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('6', '1', '5', '2019-03-29 15:30:00', 'pending');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('6', '1', '3', '2019-03-29 15:30:00', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('6', '6', '2', '2019-03-09 08:30:45', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('7', '3', '3', '2019-03-30 16:10:00', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('7', '4', '6', '2019-03-05 12:13:14', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('7', '4', '4', '2019-03-05 12:13:14', 'pending');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('7', '4', '8', '2019-03-05 12:13:14', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('7', '4', '2', '2019-03-05 12:13:14', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('7', '4', '3', '2019-03-05 12:13:14', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('8', '5', '7', '2019-03-06 14:22:00', 'pending');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('8', '5', '6', '2019-03-06 14:22:00', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('8', '5', '4', '2019-03-04 12:30:17', 'pending');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('8', '5', '2', '2019-03-04 12:30:50', 'accept');
INSERT INTO `snickr`.`channelinvitation` (`cinvitorid`, `cid`, `cinviteeid`, `cinvitetime`, `cstatus`) VALUES ('8', '5', '3', '2019-03-04 12:30:50', 'accept');
```

cinvitorid	cid	cinviteeid	cinvitetime	cstatus
2	8	6	2019-02-14 12:10:00	accept
2	9	6	2019-02-14 12:30:00	accept
3	2	5	2019-03-28 11:05:30	accept
3	2	7	2019-03-30 12:05:00	reject
3	7	2	2019-03-02 19:40:37	accept
3	7	6	2019-03-02 19:40:37	accept
3	7	7	2019-03-02 19:40:37	reject
6	1	3	2019-03-29 15:30:00	accept
6	1	5	2019-03-29 15:30:00	pending
6	1	7	2019-03-29 15:30:00	accept
6	6	2	2019-03-09 08:30:45	accept
7	3	3	2019-03-30 16:10:00	accept
7	4	2	2019-03-05 12:13:14	accept
7	4	3	2019-03-05 12:13:14	accept
7	4	4	2019-03-05 12:13:14	pending
7	4	6	2019-03-05 12:13:14	accept
7	4	8	2019-03-05 12:13:14	accept
8	5	2	2019-03-04 12:30:50	accept
8	5	3	2019-03-04 12:30:50	accept
8	5	4	2019-03-04 12:30:17	pending
8	5	6	2019-03-06 14:22:00	accept
8	5	7	2019-03-06 14:22:00	pending
NULL	NULL	NULL	NULL	NULL

9. Message Table

```
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('1', '7', '1', 'How can we solve Q1?', '2019-03-30 17:30:00');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('2', '3', '1', 'Try to use perpendicular rules.', '2019-03-30 17:31:05');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('3', '6', '1', 'Perpendicular Rules. I think.', '2019-03-30 17:35:00');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('4', '5', '2', 'When is the midterm exam?', '2019-03-28 14:05:28');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('5', '3', '2', 'Next Monday.', '2019-03-28 14:06:07');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES
('6', '7', '3', 'I tried perpendicular and it worked.', '2019-03-30 16:30:45');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('7', '3', '3', 'That\'s great.', '2019-03-30 16:31:28');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES
('8', '7', '4', 'Is anyone still looking for a teammate for the project?', '2019-03-05 12:20:18');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('9', '2', '4', 'You can post on piazza.', '2019-03-05 13:28:12');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES
('10', '8', '4', 'I\'m looking for a teammate too and we can build a direct channel?', '2019-03-05 20:17:10');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('11', '8', '5', 'QuickSort is astonishing!', '2019-03-04 21:05:06');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('12', '3', '5', 'I think so.', '2019-03-04 21:30:17');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES
('13', '2', '6', 'Have you received the TA salary for February?', '2019-03-09 10:30:03');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('14', '6', '6', 'Not yet.', '2019-03-09 10:36:08');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES
('15', '3', '7', 'Are you guys coming to the lab today?', '2019-03-03 09:45:37');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('16', '8', '8', 'Avengers is coming out!', '2019-02-14 12:22:06');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('17', '8', '8', 'This weekend!', '2019-02-14 12:22:30');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('18', '2', '8', 'Yeeeeah!', '2019-02-14 12:25:08');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('19', '6', '8', 'Emmmmm....', '2019-02-14 12:28:30');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('20', '6', '9', 'Do you free tomorrow night?', '2019-02-14 12:31:30');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('21', '2', '9', 'Nope, I have a due to work with.', '2019-02-14 12:32:05');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('22', '2', '9', 'But I\'m free this weekend.', '2019-02-14 12:32:30');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('23', '6', '9', 'Okay, how about go hiking?', '2019-02-14 12:35:10');
INSERT INTO `snickr`.`message` (`mid`, `senderid`, `cid`, `content`, `sendtime`) VALUES ('24', '2', '9', 'That sounds great!', '2019-02-14 12:41:30');
```

mid	senderid	cid	content	sendtime
1	7	1	How can we solve Q1?	2019-03-30 17:30:00
2	3	1	Try to use perpendicular rules.	2019-03-30 17:31:05
3	6	1	Perpendicular Rules. I think.	2019-03-30 17:35:00
4	5	2	When is the midterm exam?	2019-03-28 14:05:28
5	3	2	Next Monday.	2019-03-28 14:06:07
6	7	3	I tried perpendicular and it worked.	2019-03-30 16:30:45
7	3	3	That's great.	2019-03-30 16:31:28
8	7	4	Is anyone still looking for a teammate for the project?	2019-03-05 12:20:18
9	2	4	You can post on piazza.	2019-03-05 13:28:12
10	8	4	I'm looking for a teammate too and we can build a direct ...	2019-03-05 20:17:10
11	8	5	QuickSort is astonishing!	2019-03-04 21:05:06
12	3	5	I think so.	2019-03-04 21:30:17
13	2	6	Have you received the TA salary for February?	2019-03-09 10:30:03
14	6	6	Not yet.	2019-03-09 10:36:08
15	3	7	Are you guys coming to the lab today?	2019-03-03 09:45:37
16	8	8	Avengers is coming out!	2019-02-14 12:22:06
17	8	8	This weekend!	2019-02-14 12:22:30
18	2	8	Yeeeeah!	2019-02-14 12:25:08
19	6	8	Emmmmm....	2019-02-14 12:28:30
20	6	9	Do you free tomorrow night?	2019-02-14 12:31:30
21	2	9	Nope, I have a due to work with.	2019-02-14 12:32:05
22	2	9	But I'm free this weekend.	2019-02-14 12:32:30
23	6	9	Okay, how about go hiking?	2019-02-14 12:35:10
24	2	9	That sounds great!	2019-02-14 12:41:30
NULL	NULL	NULL	NULL	NULL

Are you guys coming to the lab today?

Question (c) of Project 1(with the result for question (d))

1. The SQL query is:

```
use snickr;
insert into User (uname, nickname, upassword, email) values
('Elizabeth','Alice','qweraaa','aliceqwer@gmail.com');
```

The User table before creating new account is:

uid	uname	nickname	upassword	email
1	John	JJ	ii123	iohn@amail.com
2	Rose	rosv	Rosv876	oRose3@amail.com
3	Tompson	TommvP	Tom123	TTP@hotmail.com
4	Linda	NULL	Lin862	Linda@nvu.edu
5	Kevin	KKV	kevin111	Kevin1@vahoo.com
6	Jack	Jackv	JJlin	JaCk082@hotmail.com
7	Peter	PPT	TPP647	746Peter@hotmail.com
8	Marv	M.	mmarvv	Mar333@vahoo.com
NULL	NULL	NULL	NULL	NULL

The User table after creating new account is:

uid	uname	nickname	upassword	email
1	John	JJ	ii123	iohn@amail.com
2	Rose	rosv	Rosv876	oRose3@amail.com
3	Tompson	TommvP	Tom123	TTP@hotmail.com
4	Linda	NULL	Lin862	Linda@nvu.edu
5	Kevin	KKV	kevin111	Kevin1@vahoo.com
6	Jack	Jackv	JJlin	JaCk082@hotmail.com
7	Peter	PPT	TPP647	746Peter@hotmail.com
8	Marv	M.	mmarvv	Mar333@vahoo.com
9	Elizabeth	Alice	qweraaa	aliceqwer@amail.com
NULL	NULL	NULL	NULL	NULL

2. If a user not in a certain workspace want to create a public channel inside this workspace, for example user 1 who does not belongs to workspace 1 wants to create a 'test' channel inside workspace 1, he will fail to do so.

The SQL query is:

```
use snickr;
insert into Channel (creatorid,wid,ctype,cname,ctime) values
(1,1,'public','test',now());
```

```
204  22:34:19  insert into Channel (creatorid,wid,ctype,cname,ctime) values (1,1,'public','test',n...  Error Code: 1452. Cannot add or update a child row: a foreign key constraint fa...  0.000 sec
```

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`snickr`.`channel`, CONSTRAINT `channel_ibfk_1` FOREIGN KEY (`creatorid`, `wid`) REFERENCES `workspacemembership` (`uid`, `wid`) ON DELETE CASCADE)

A success example is showed as below (user 5 belongs to worksapce 1):

```
insert into Channel (creatorid,wid,ctype,cname,ctime) values
(5,1,'public','test',now());
```

The Channel Table before creating new channel is:

cid	creatorid	wid	ctype	cname	ctime
1	6	1	public	Homework	2019-03-29 15:30:00
2	3	1	private	Exams	2019-03-28 11:05:10
3	7	1	direct	Solution	2019-03-28 17:20:30
4	7	2	public	Database	2019-03-05 12:13:14
5	8	2	public	Algorithm	2019-03-04 12:30:17
6	6	2	direct	Salary Chatting	2019-03-09 08:30:03
7	3	2	private	Lab Hours	2019-03-02 19:40:37
8	2	3	public	Share Movie	2019-02-14 12:10:00
9	2	3	direct	Love	2019-02-14 12:30:00
NULL	NULL	NULL	NULL	NULL	NULL

The Channel Table after creating new channel is:

cid	creatorid	wid	ctype	cname	ctime
1	6	1	public	Homework	2019-03-29 15:30:00
2	3	1	private	Exams	2019-03-28 11:05:10
3	7	1	direct	Solution	2019-03-28 17:20:30
4	7	2	public	Database	2019-03-05 12:13:14
5	8	2	public	Algorithm	2019-03-04 12:30:17
6	6	2	direct	Salary Chatting	2019-03-09 08:30:03
7	3	2	private	Lab Hours	2019-03-02 19:40:37
8	2	3	public	Share Movie	2019-02-14 12:10:00
9	2	3	direct	Love	2019-02-14 12:30:00
11	5	1	public	test	2019-04-05 22:44:12
NULL	NULL	NULL	NULL	NULL	NULL

3. The SQL query is:

```
use snickr;
SELECT
    Workspace.wname, User.uname
FROM
    Workspace
        JOIN
    Administrator
        JOIN
    User ON Workspace.wid = Administrator.wid
        AND User.uid = Administrator.uid;
```

The result is:

wname	uname
Maths Group	Tompson
Maths Group	Peter
Computer Science	Linda
Movie Lover	Rose
Movie Lover	Jack

4. The SQL query is:

```
use snickr;
SELECT
    t1.cname, IFNULL(temp, 0) AS NotjoinNum
FROM
    (SELECT
        cname
    FROM
        Channel
    JOIN Workspace ON Channel.wid = Workspace.wid
    WHERE
        wname = 'Computer Science') AS t1
        LEFT JOIN
    (SELECT
        cname, COUNT(cinviteeemail) AS temp
    FROM
        Channel
    JOIN ChannelInvitation
    JOIN Workspace ON Channel.cid = ChannelInvitation.cid
        AND Channel.wid = Workspace.wid
    WHERE
        wname = 'Computer Science'
            AND cstatus = 'pending'
            AND DATEDIFF(NOW(), cinvitetime) > 5
    GROUP BY Channel.cid) AS t2 ON t1.cname = t2.cname;
```

The result is:

cname	NotjoinNum
Database	1
Algorithm	2
Salary Chatting	0
Lab Hours	0

5. We choose ‘Homework’ channel as the particular one and list all message inside this channel in chronological order.

The SQL query is:

```
use snickr;
SELECT
    content
FROM
    Message
        NATURAL JOIN
    Channel
WHERE
    cname = 'Homework'
ORDER BY sendtime;
```

The result is:

content
How can we solve O1?
Try to use perpendicular rules.
Perpendicular Rules. I think.

6. We choose ‘Rose’ as the particular user and list all message she posted in any channel.

   The SQL query is:

```
use snickr;
SELECT
    content
FROM
    Message
        JOIN
    User ON Message.senderid = User.uid
WHERE
    uname = 'Rose';
```

   The result is:

content
You can post on piazza.
Have you received the TA salary for February?
Yeeeeah!
Nope. I have a due to work with.
But I'm free this weekend.
That sounds great!

7. We choose ‘Jack’ as the particular user and list all message he is accessible to and contain the keyword ‘perpendicular’in any channel.

   The SQL query is:

```
use snickr;
SELECT
    content
FROM
    Message
        JOIN
    Channelmembership
        JOIN
    User ON Message.cid = Channelmembership.cid
        AND Channelmembership.uid = User.uid
WHERE
    content LIKE '%perpendicular%'
        AND uname = 'Jack';
```

   The result is:

content
Try to use perpendicular rules.
Perpendicular Rules. I think.

Revise our design

We do not change our design of schema in project 1. We adjust the length of upassword in the User table to satisfy the length of an encrypted password using the SHA algorithm. Since the design of the front-end implementation is different from what we expected, we just modified some of the procedures. What's more, we add some new procedures, such as remove admins, delete users from workspace, and change user nickname because we add some feature function to our system.

Implementation

Introduction

We follow the development principles of separation between front-end and back-end. We use Flask for the web application, which includes a web server that can be used for testing and development. Also, we use SQLAlchemy (a Python SQL toolkit) as Object Relational Mapper(ORM) for connecting database with server. For the front-end design, we use HTML and CSS. To get a better model for the webpage, we use the Bootstrap component library. We use jQuery, Ajax, and JSON to communicate between back-end and front-end.

Program dependencies

Flask 1.0.2; SQLAlchemy 1.3.3; PyMySQL 0.9.3; Flask-SQLAlchemy 2.4.0; Flask-Cors 3.0.7

SQL Injection Prevention

Since we use SQLAlchemy as the SQL toolkit and ORM, we can guard against SQL Injection attack.

XSS attack

We use escape() and unescape() function in JavaScript to convert HTML special chars into safe strings and therefore prevent XSS attack.

Password Encryption

We implement the SHA-1 algorithm to encrypt user password and store encrypted passwords in the database, so there is no risk that the user's passwords are leaked when the database is hacked.

Session

We import session from Flask and set session for each login users when they log in. Also, we delete sessions when they log out. Therefore, if the user does not log out, he can still visit the websites.

Assign URLs

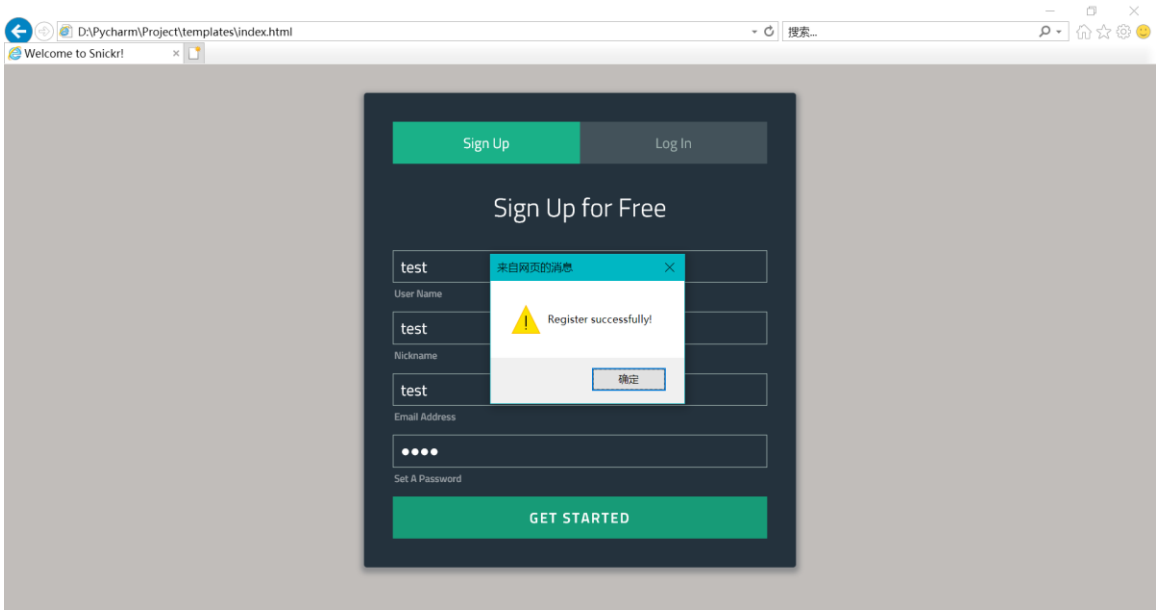
We use the URLs to pass the user's email address across different webpages in the following format: xxx.html?uemail = 'user@mail.com'. When web server visits websites, it only matches the part before '?', so we can access the website and show information related to logged in user

as its email is passed in the URL. Also, when the user enters workspace or channel, we pass the wid or cid in this way.

Interactive process

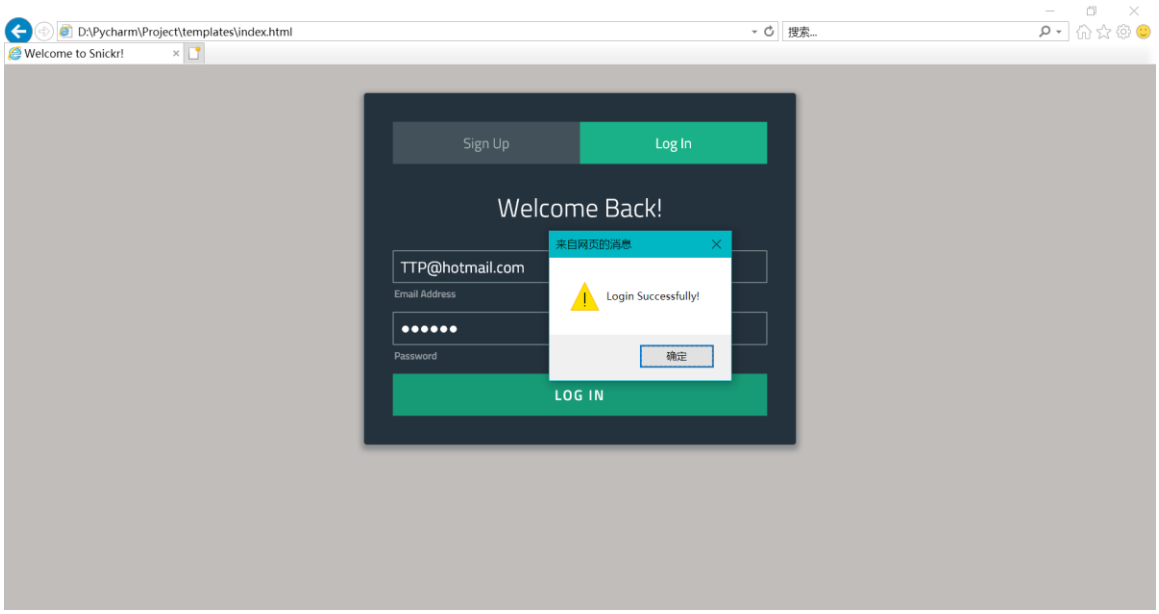
1. Register

Input necessary information in /index.html and click the ‘GET STARTED’ button. The new user is registered successfully, and the web will jump to /main.html related to this user.



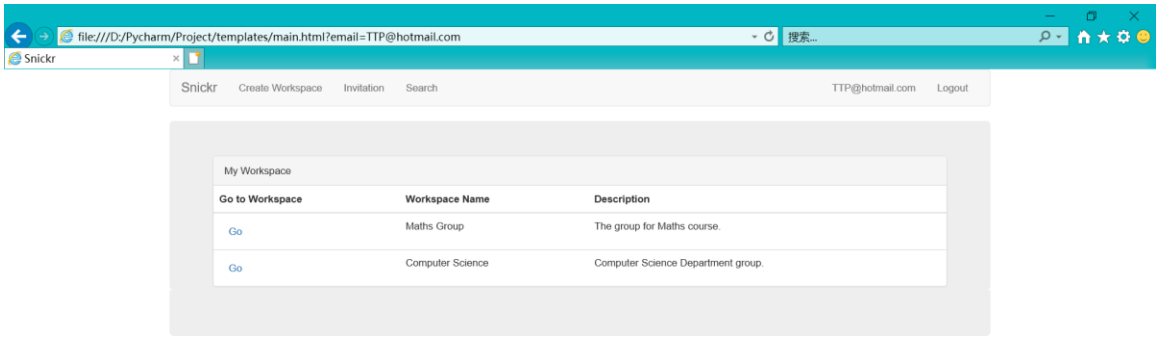
2. Login

Login with an existed user by inputting email address and password and clicking ‘LOG IN’ button.



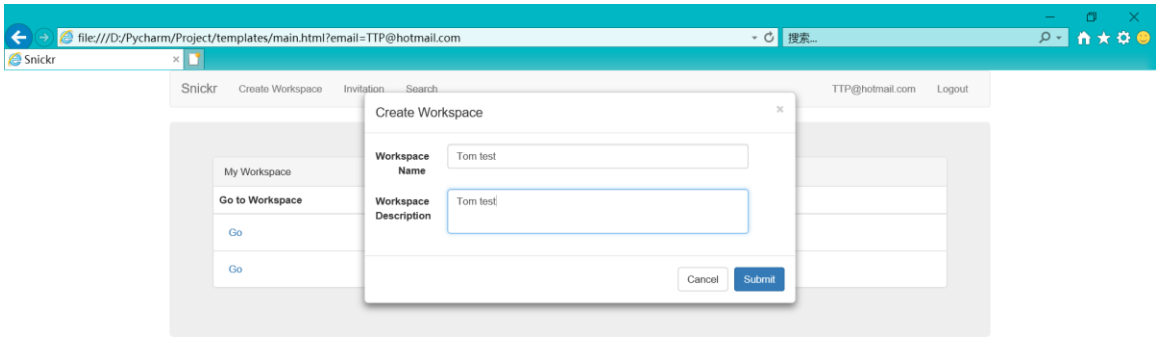
3. Workspace list

In the /main.html of each user, it shows the workspaces he belongs to.

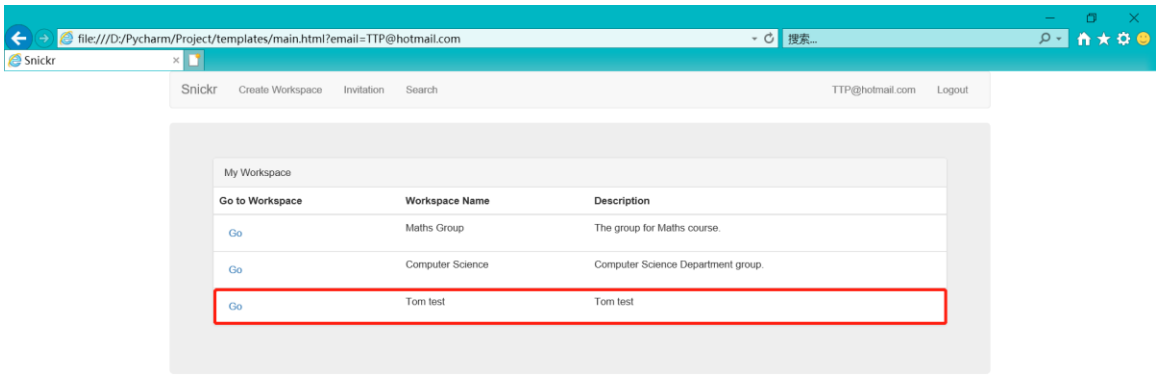


4. Create workspace

Click ‘Create Workspace’ button in /main.html to create a new workspace.

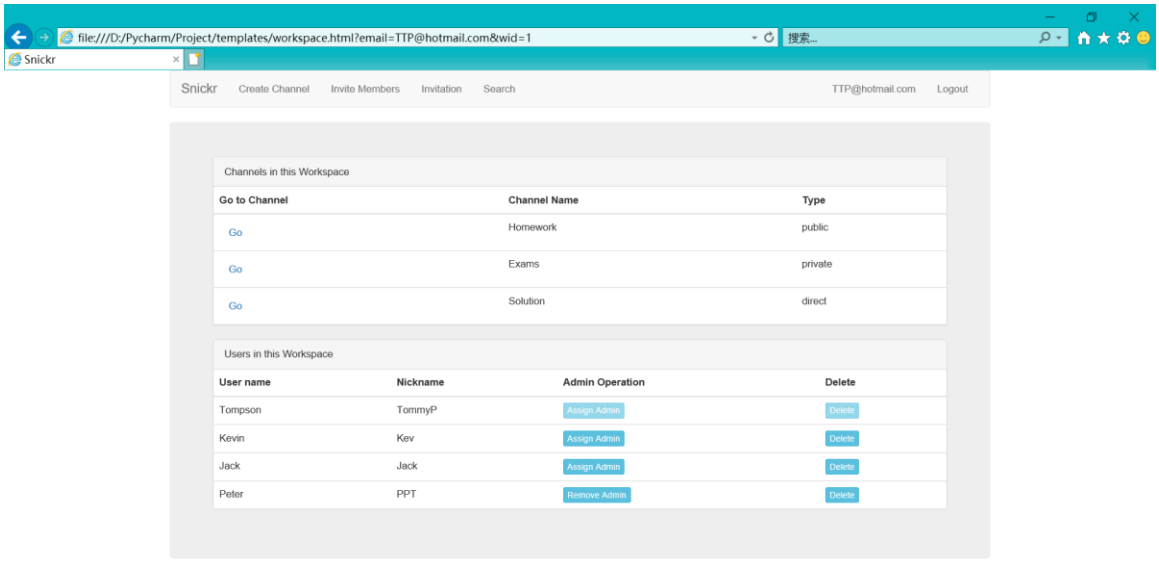


Input name and description of new workspace and click the ‘Submit’ button. The web will be refreshed and show the new workspace list with the one just created.



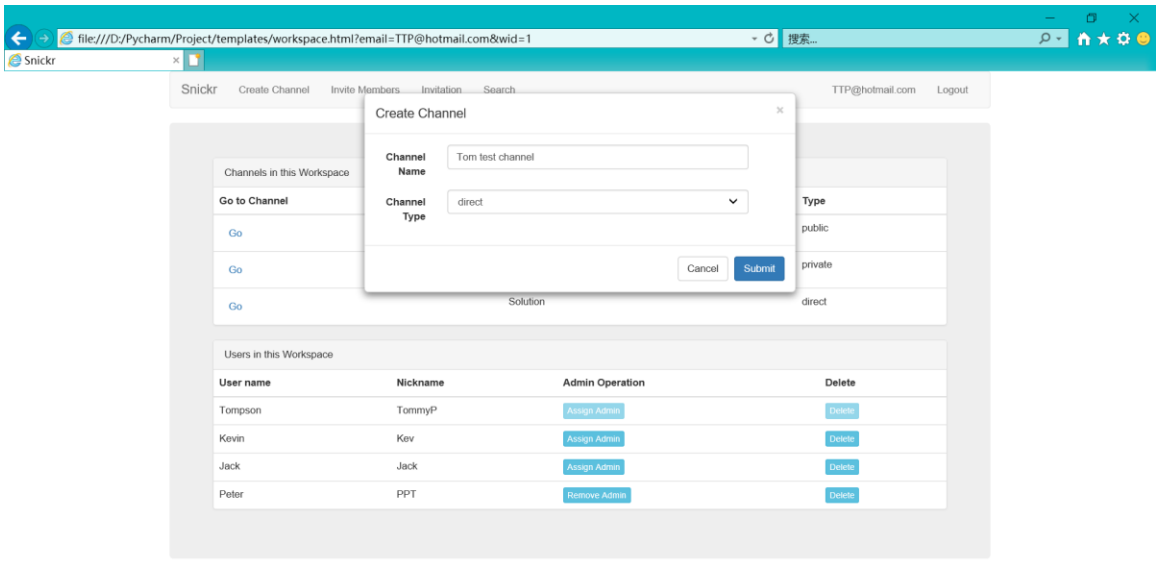
5. Channel and user list in workspace

Click the 'Go' button of each workspace, the web will jump to /workspace.html and show the list of channels and users in the workspace.

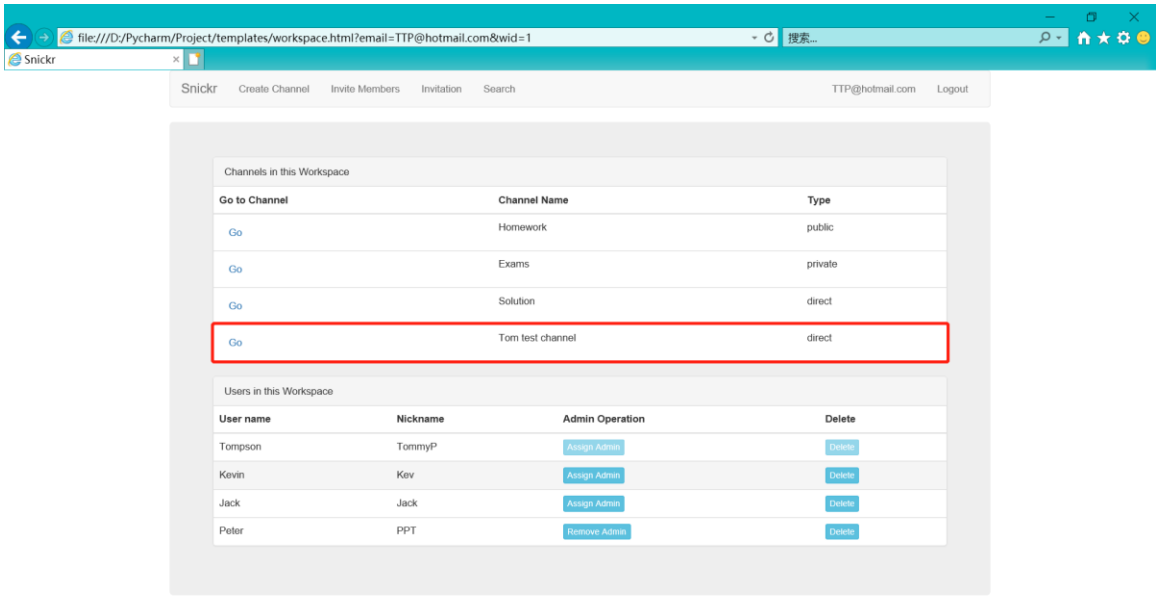


6. Create channel

Click the 'Create Channel button in /workspace.html to create a new channel.

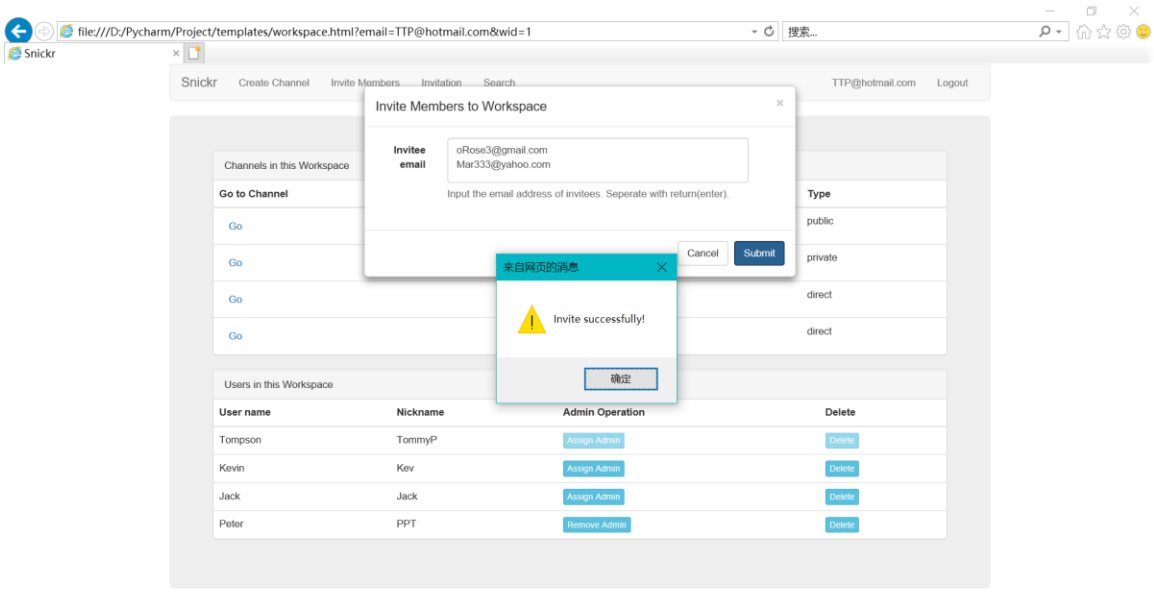


Input name and choose the type of new channel and click the ‘Submit’ button. The web will be refreshed and show the new channel list with the one just created.



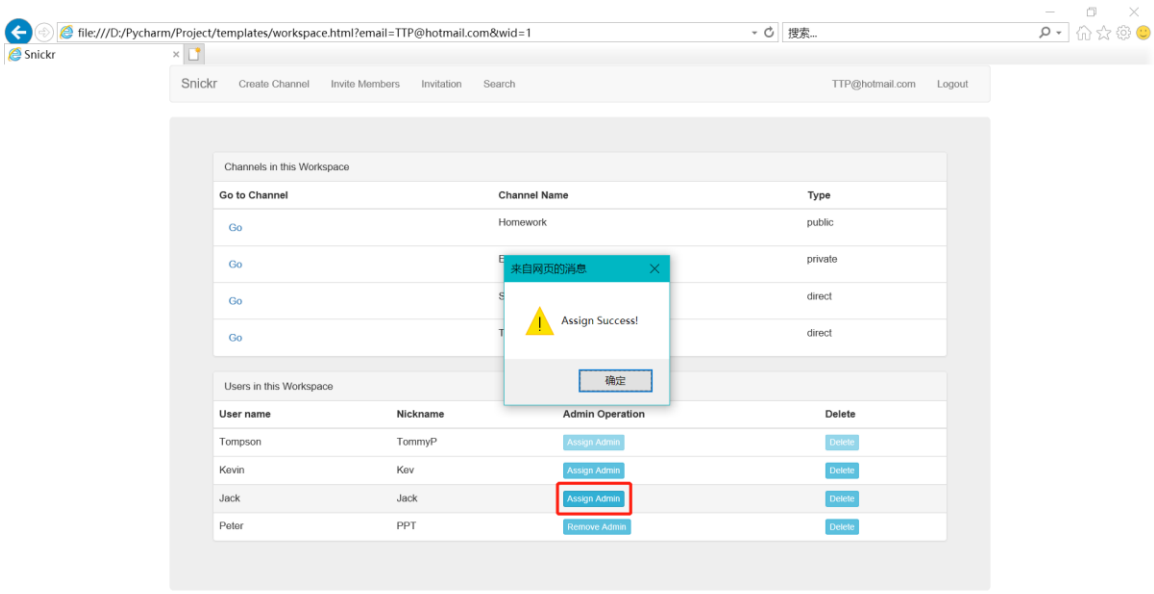
7. Invite members to workspace

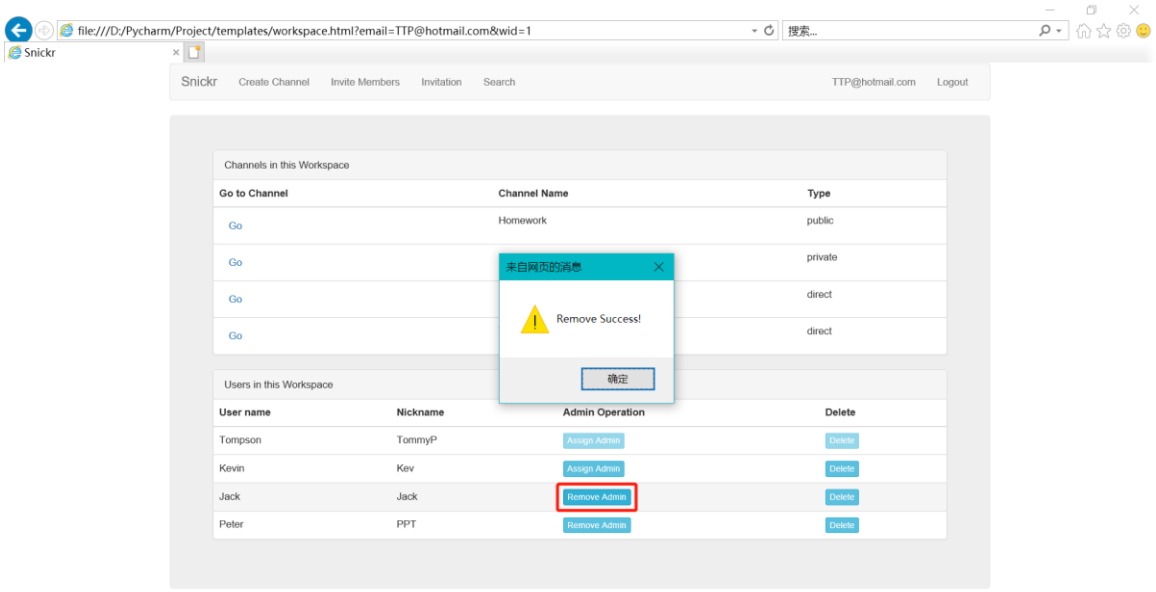
Click ‘Invite Members’ button in /workspace.html to invite members into the workspace. Input invitees’ email addresses and click the ‘Submit’ button.



8. Assign/remove admin of workspace

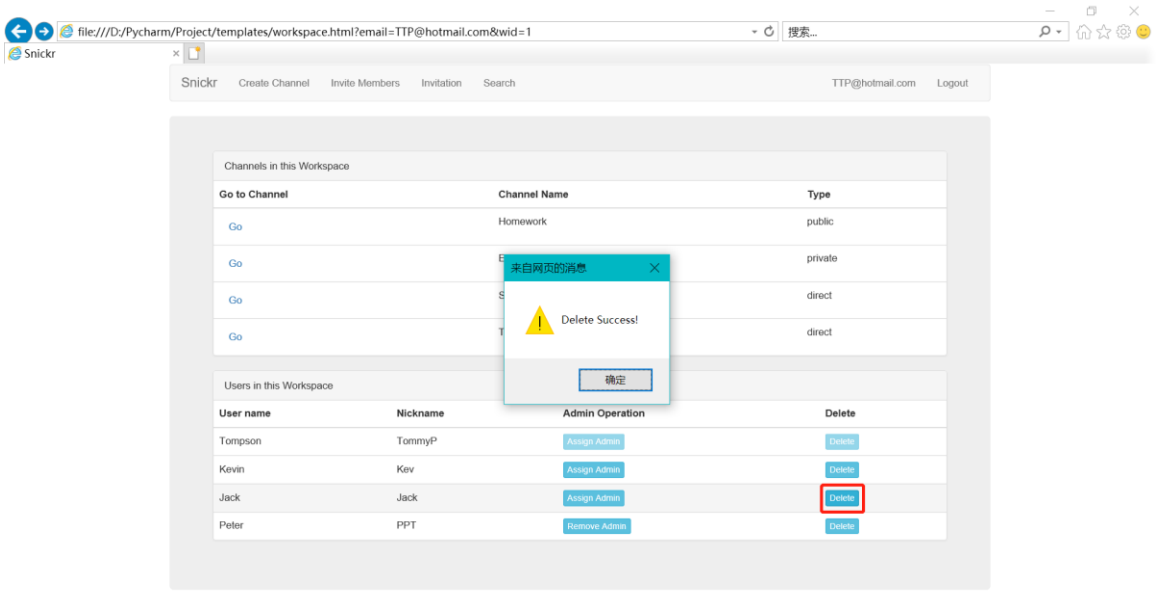
Click the ‘Assign Admin’ or ‘Remove Admin’ button in the user list of /workspace.html to manage admins. This can only be done by the creator of the workspace; for other users, the buttons are disabled.





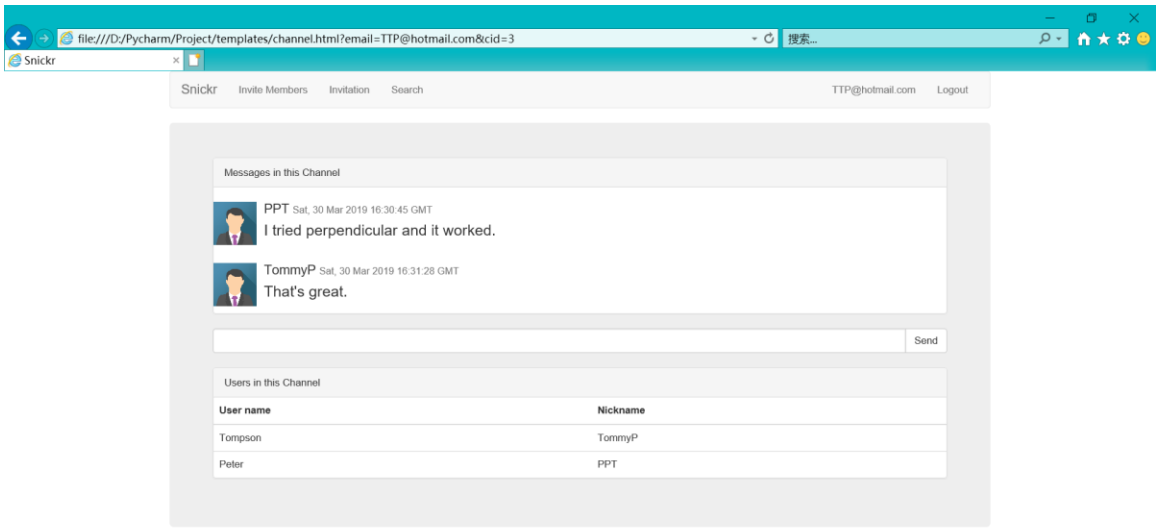
9. Delete user from workspace

Click the ‘Delete’ button in the user list of /workspace.html to delete users from workspace. This can only be done by admins of the workspace; for other users, the button is disabled.



10. Message and user list in channel

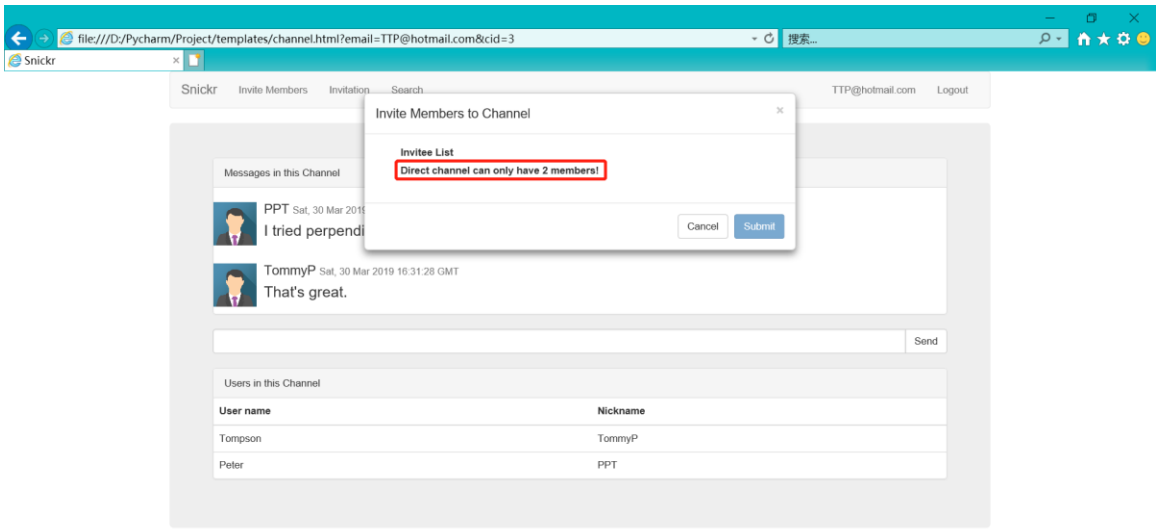
Click the ‘Go’ button of each channel, the web will jump to /channel.html and show the list of messages and users in the channel.



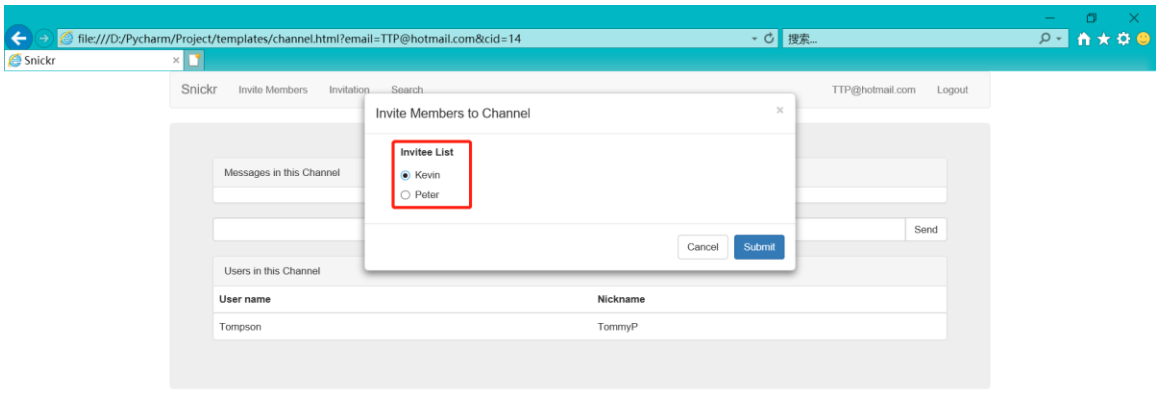
11. Invite members to channel

Click ‘Invite Members’ in /channel.html to invite members into channels. The pop-up window will show the list of users that can be invited (who is in the same workspace of this channel but not in the channel yet).

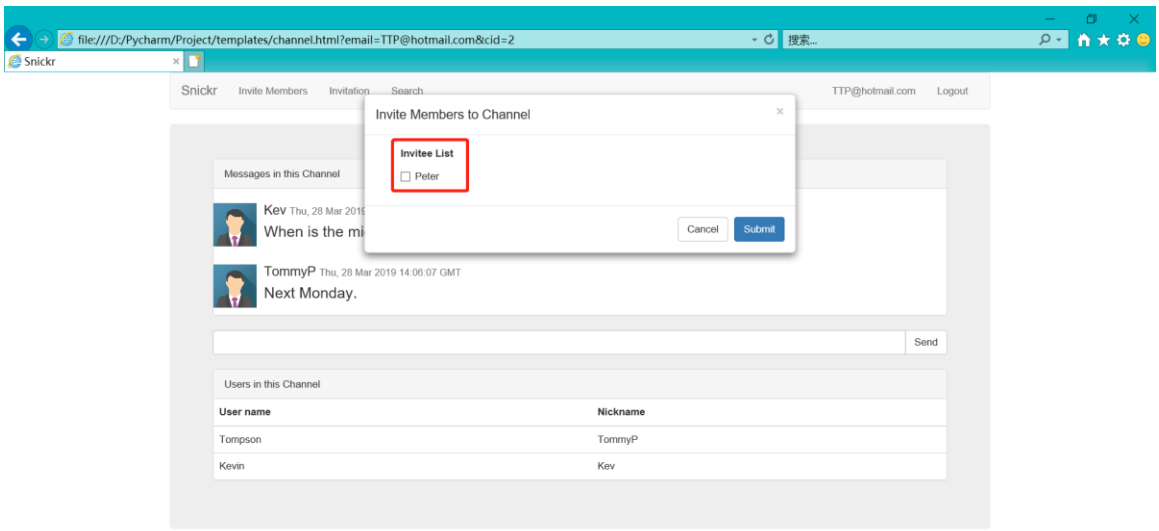
For a direct channel with two users, the window will alert that there can only have two users in direct channel and the ‘Submit’ button is disabled.



For direct channel with only one user, the window will show the radio button list which the user can only choose one of the users as the invitee and click the ‘Submit’ button.

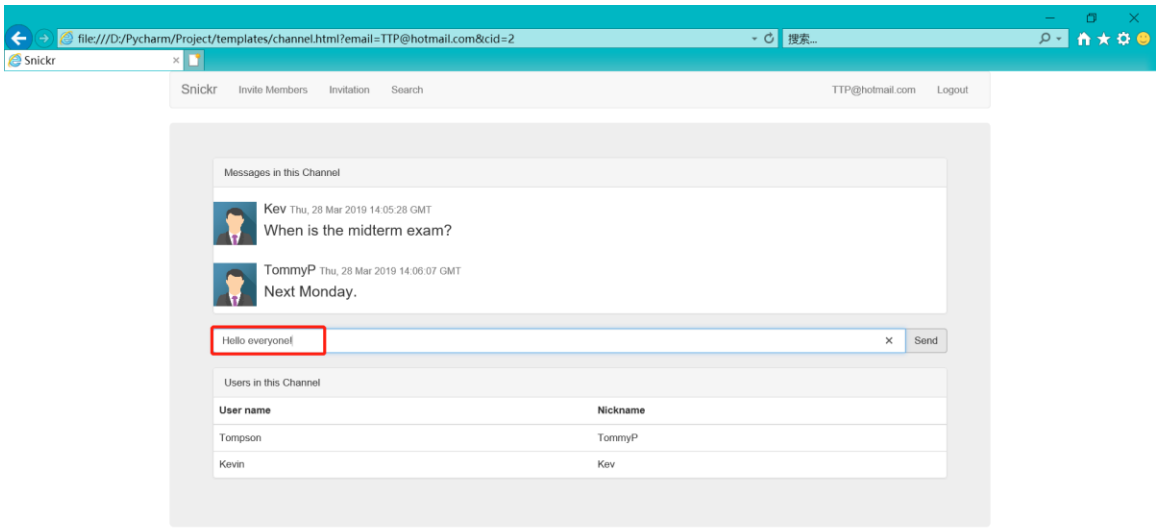


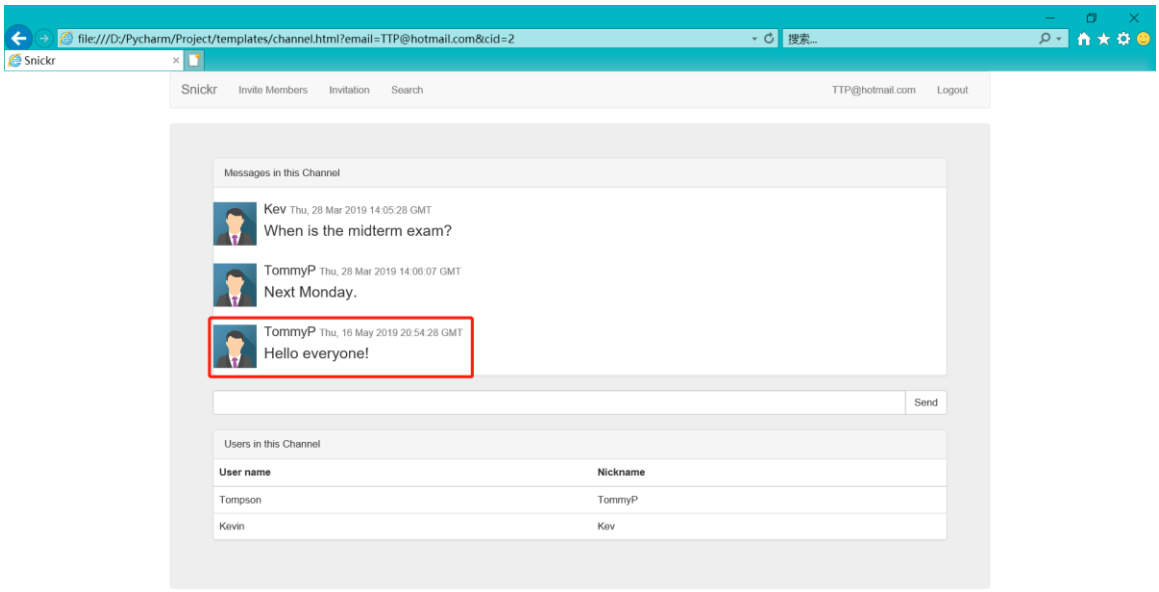
For public and private channel, the window will show the checkbox list which the user can choose users as the invitees and click the ‘Submit’ button.



12. Send message in channel

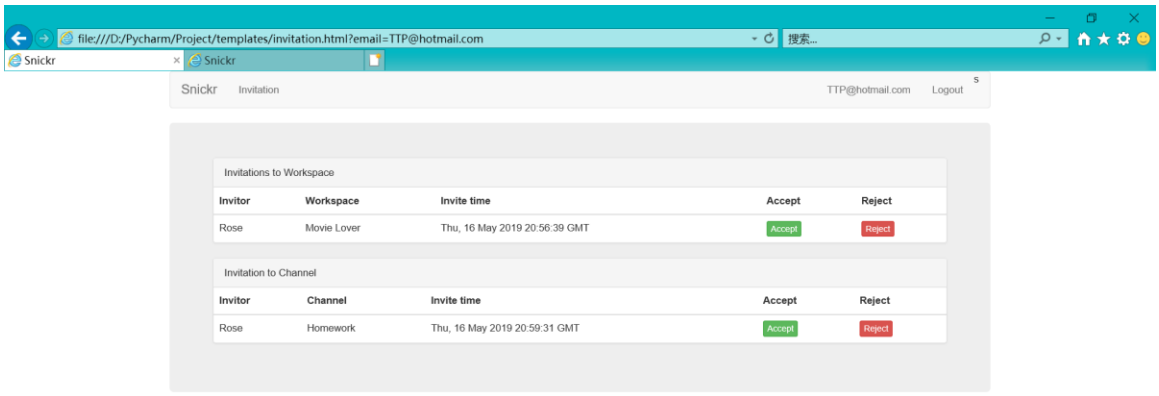
Input message and click the ‘Send’ button, the web will refresh and show the message just sent.



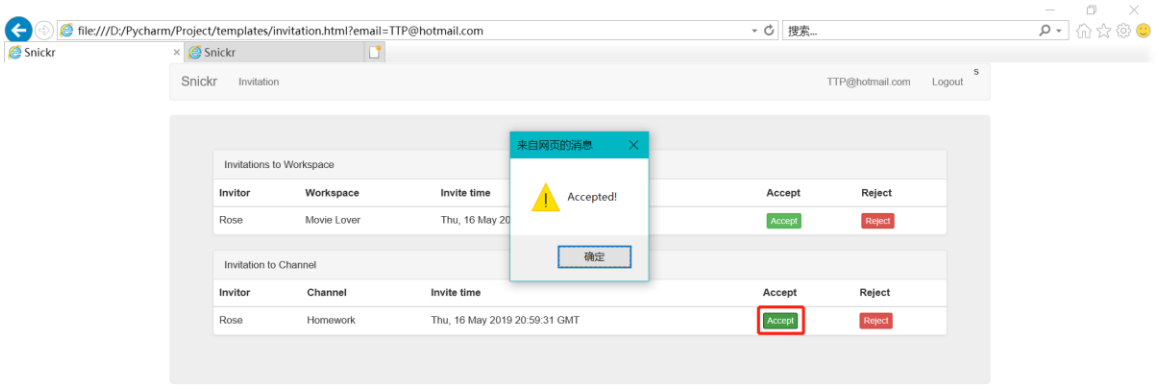


13. Respond to invitations

Click the ‘Invitation’ button, and the web will jump to /invitation.html and show the invitation list of this user.



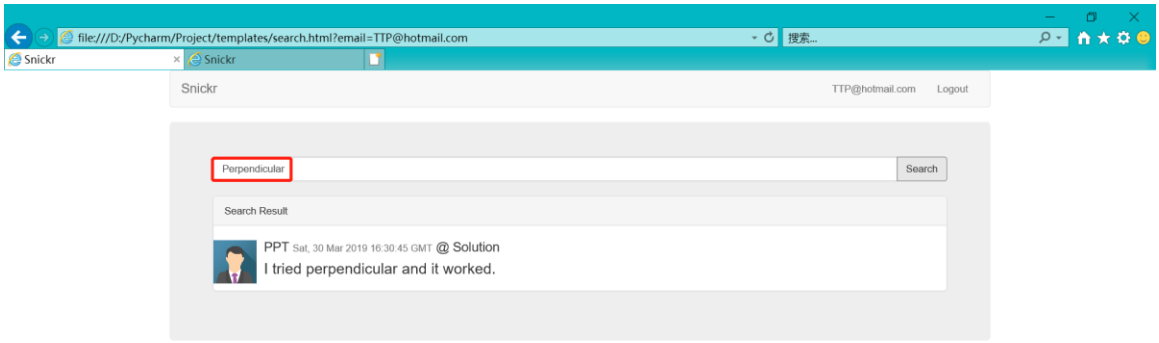
Click the ‘Accept’ or ‘Reject’ button of each invitation to respond.



14. Search messages with keyword

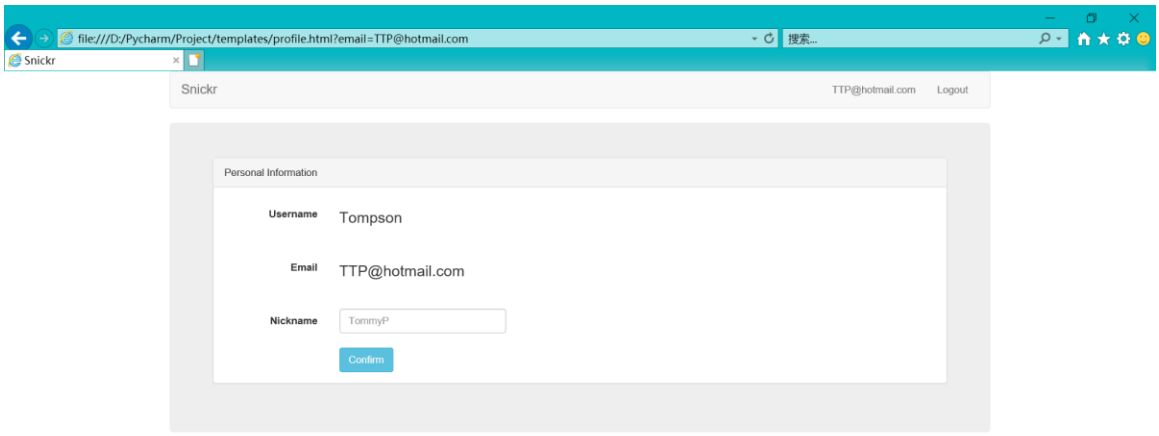
Click the ‘Search’ button, and the web will jump to /search.html.

Input the keyword and click the ‘Search’ button. The web will refresh and show the message list contain the keyword.

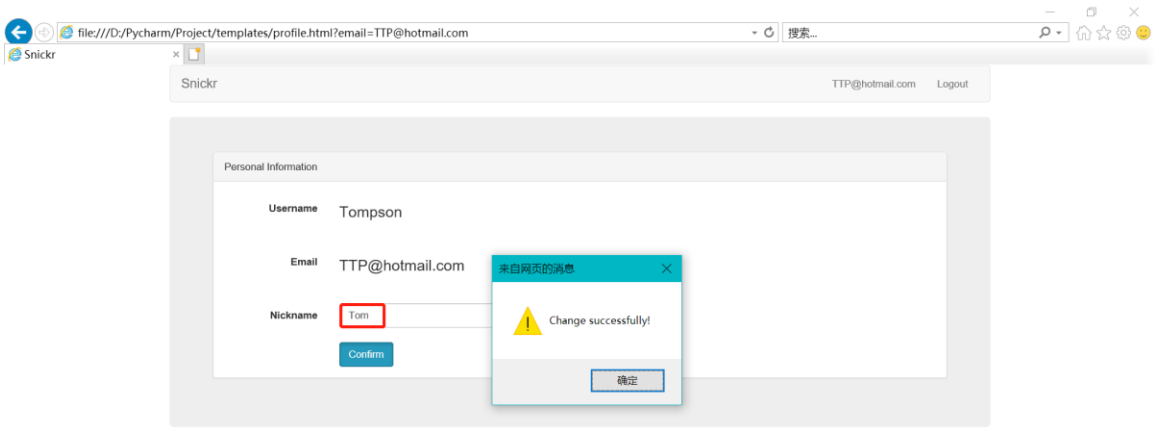


15. Change nickname

Click user email address button on the top right, and the web will jump to /profile.html.

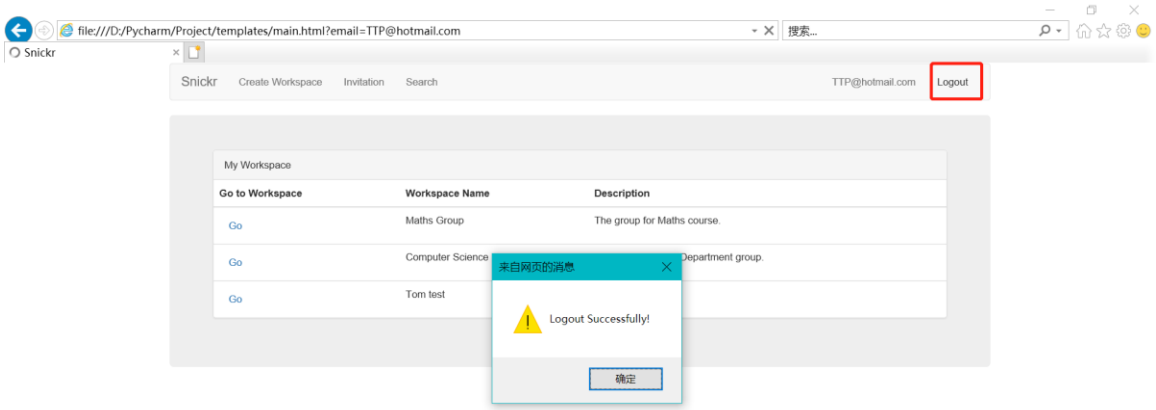


Input new nickname and click the ‘Confirm’ button, the web will refresh and show the new nickname.



16. Logout

Click the ‘Logout’ button, and the user will log out from the web.


file:///D:/Pycharm/Project/templates/main.html?email=TTP@hotmail.com
搜索...
Snickr
Snickr
Create Workspace
Invitation
Search
TTP@hotmail.com
Logout
My Workspace
Go to Workspace
Workspace Name
Description
Go
Maths Group
The group for Maths course.
Go
Computer Science
Department group.
Go
Tom test
来自网页的消息
Logout Successfully!
确定