#Pro2

Zuolun Xiao zx798

Saitong Zhao sz2641

According to the project requirement, the information of users includes uid, email, username, nickname and password certainly. We are going to store hashed password encrypted with pbkdf2-sha512, so we are storing the password as char with length less than 150.

And for workspace, we tried to include adminID (who is an administrator of a certain workspace) into the “workspace” table, but it will duplicate lots of information because other members also can be administrators of the same workspace and if there are more than 2 people to be the administrators, the name, description and the time-stamp of the workspace will appear repeatedly. So we decided to create another table “workadmin” to put the id of administrators and workspace together. And to check who is in the workspace, it is necessary to create a table “winclude” so that we can check if someone has the right to create a channel.

And also, there is a table called “channels” mentioned by the guidelines. Since there are three types of channels, public, private, and direct, we need to add an attribute called “cType” to label which kind of channel it is. 0 means “public”, 1 means “private” and 2 means direct. “wid” in every tuple shows which workspace the certain channel belongs to. And “creatorID” shows the user who create this channel. To show the relationship between channels and users, that is to show which group members are included in a certain channel, we create another table “cinclude” including “cid” and “uid”. It’s also very convenient to check if someone in the workspace have the right to see the messages.

For messages, there are attributes “msid”, “cid” (to indicate certain messages belong to which channel), “uid” (to indicate which user sends out a certain message), “mcontext”, “stamp”.

And since the invitation which sent by administrators to the users does not mean that the user can be put in a certain workspace or a certain channel directly since he or she has to decide to or not to accept the invitation, we also need two more tables “winvite” and “cinvite”. Before a user can send out an invitation to someone to join a workspace, we have to check if he is an administrator of the workspace. And similarly, we have to check if a user has right to send out an invitaiton to someone to join in a channel before we can put his invitation into the “cinvite” table. And we are going to use a mechanism of message to implement this invitation feature, we need two bool variables, is\_read and is\_accepted to mark the status of specific invitation.

The relational schema is shown below:

users(uid, uname, upassword, unick, email)

workspaces(wid,wname, wdescript, stamp)

workadmin(adminid, wid)

channels(cid, cname,cdescript,wid, ctype,creatorid,stamp)

messages(msid, cid, uid, mcontext, stamp)

winclude(wid, uid)

cinclude(cid, uid)

winvite(winvid, adminid, inviteid, wid, stamp, is\_read, is\_accepted)

cinvite(cinvid, adminid, inviteid, cid, stamp, is\_read, is\_accepted)

Some attributes in one table are the foreign keys reference to some attributes in other tables. The relationship is shown below:

workadmin.wid is a foreign key reference to workspaces.wid

channels.wid is a foreign key reference to workspaces.wid

channels.creatorid is a foreign key reference to users.uid

messages.cid is a foreign key reference to channels.cid

messages.uid is a foreign key reference to users.id

winclude.wid is a foreign key reference to workspaces.wid

winclude.uid is a foreign key reference to users.uid

cinclude.cid is a foreign key reference to channels.cid

cinclude.uid is a foreign key reference to users.uid

winvite.adminid is a foreign key reference to users.uid,

winvite.inviteid is a foreign key reference to users.uid,

winvite.wid is a foreign key reference to workspaces.wid,

cinvite.adminid is a foreign key reference to users.uid,

cinvite.inviteid is a foreign key reference to users.uid,

cinvite.cid is a foreign key reference to channels.cid

So the whole table structure is put below:

drop database project;

create database project;

use project;

create table users (

uid int auto\_increment primary key,

uname varchar(50) NOT NULL,

upassword CHAR(150) NOT NULL,

unick varchar(50) NOT NULL,

email varchar(50) NOT NULL

);

create table workspaces (

wid int auto\_increment primary key,

wname varchar(50) NOT NULL,

wdescript varchar(100) NOT NULL,

stamp datetime NOT NULL

);

create table workadmin (

adminid int NOT NULL,

wid int NOT NULL,

primary key(adminid,wid),

foreign key (adminid) references users(uid)

);

create table channels (

cid int auto\_increment primary key NOT NULL,

cname varchar(50) NOT NULL,

cdescript varchar(100) NOT NULL,

wid int NOT NULL,

ctype int NOT NULL,

creatorid int NOT NULL,

stamp datetime NOT NULL,

foreign key (wid) references workspaces(wid),

foreign key (creatorid) references users(uid)

);

create table messages (

msid int auto\_increment primary key NOT NULL,

cid int NOT NULL,

uid int NOT NULL,

mcontext varchar(1000),

stamp datetime,

foreign key(cid) references channels(cid),

foreign key(uid) references users(uid)

);

create table winclude (

wid int NOT NULL,

uid int NOT NULL,

primary key(wid,uid),

foreign key(wid) references workspaces(wid),

foreign key(uid) references users(uid)

);

create table cinclude (

cid int NOT NULL,

uid int NOT NULL,

primary key(cid,uid),

foreign key(cid) references channels(cid),

foreign key(uid) references users(uid)

);

create table winvite (

winvid int auto\_increment primary key NOT NULL,

adminid int NOT NULL,

inviteid int NOT NULL,

wid int NOT NULL,

stamp datetime NOT NULL,

is\_read bool,

is\_accepted bool,

foreign key (adminid) references users(uid),

foreign key(inviteid) references users(uid),

foreign key(wid) references workspaces(wid)

);

create table cinvite (

cinvid int auto\_increment primary key NOT NULL,

adminid int NOT NULL,

inviteid int NOT NULL,

cid int NOT NULL,

stamp datetime NOT NULL,

is\_read bool,

is\_accepted bool,

foreign key (adminid) references users(uid),

foreign key(inviteid) references users(uid),

foreign key(cid) references channels(cid)

);

The ER diagram of your design is shown below:

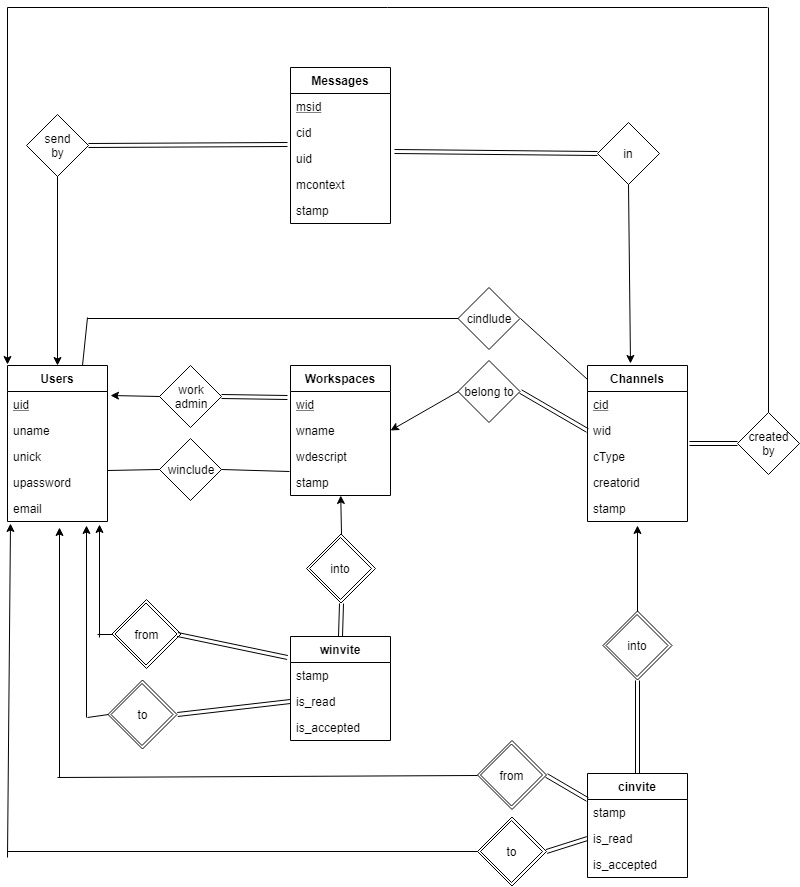


Table “workadmin” :

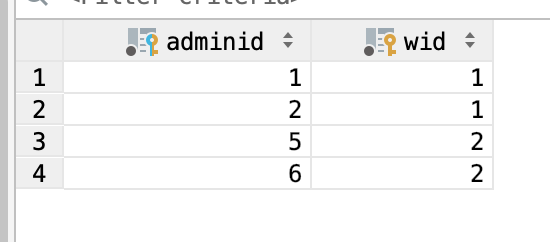


Table “workspaces”:

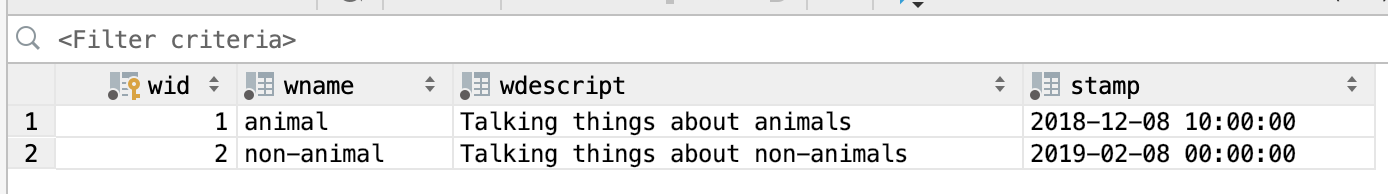


Table “channels”:

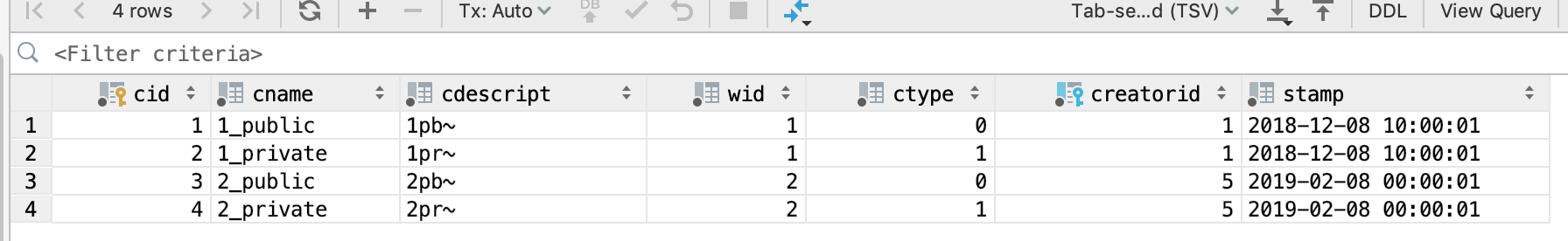


Table “cincline”:

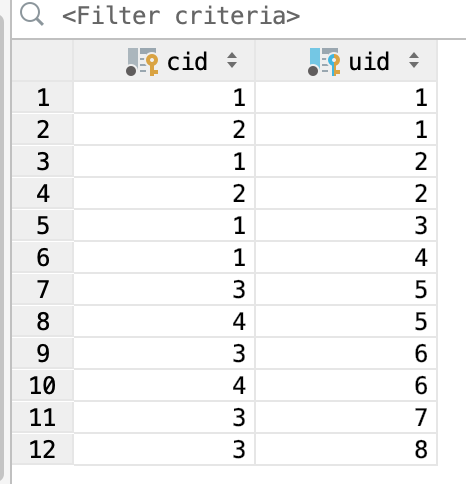


Table “cinvite”:

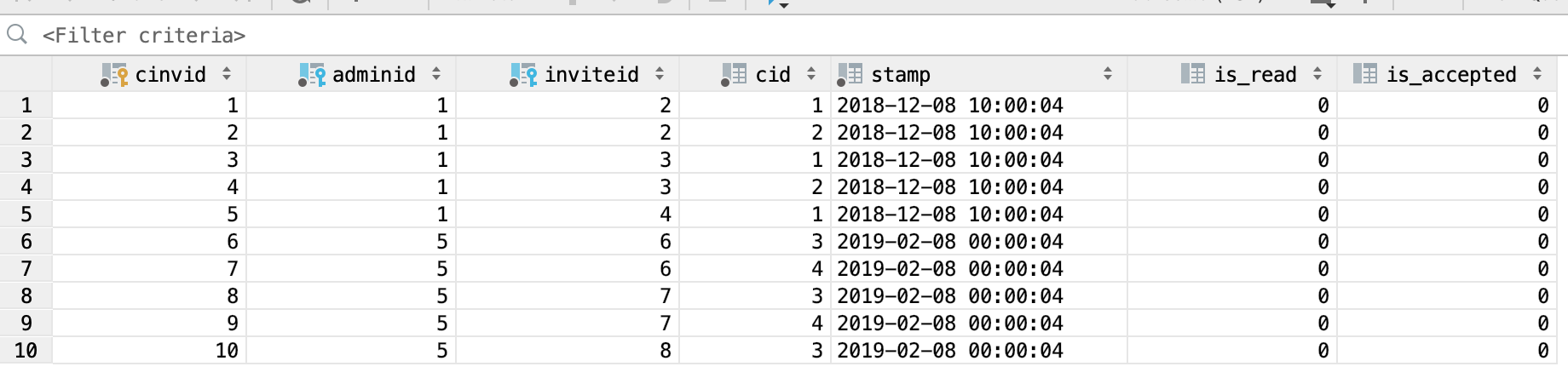


Table “messages”:

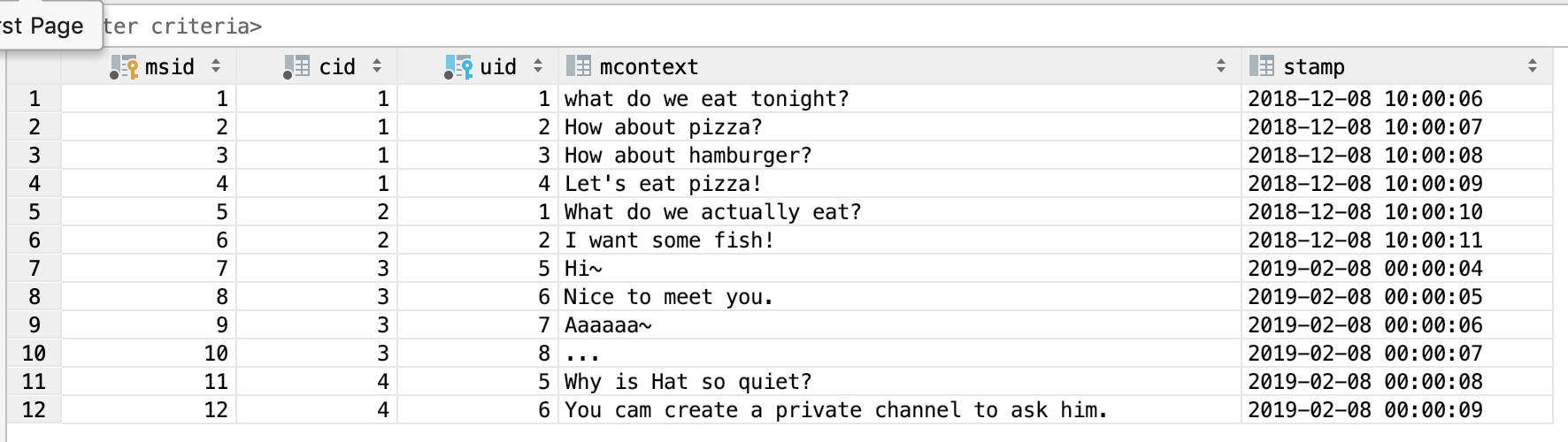


Table “users”

A screenshot of a cell phone

Description automatically generated

Table “winclude”:

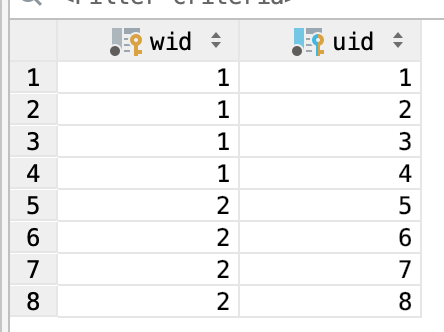
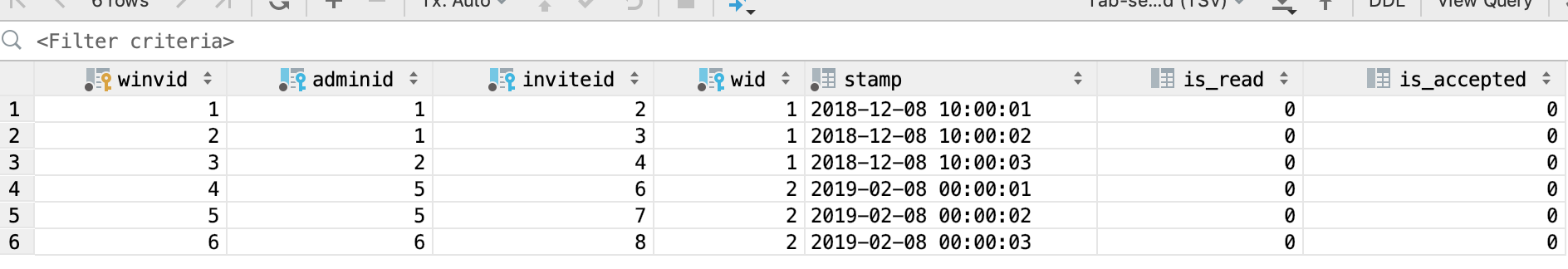


Table “winvite”:



In the “users” table, “uid” is unique to identify every single user. “uname” is to put users’ name including first name and last name. “upassword” is to save users’password. “unick” is to put users’ nickname. “email” is to put every user’s email address.

In the “workspaces” table, “wid” is unique to identify every single workspace. “wname” is the name of the workspace, “wdescript” is to put a short description and “stamp” is used to save timestamp.

In the “workadmin” table, “adminid” is the id of an administrator of the workspace whose wid is saved as “wid” in this table.

In the “channels” table, “cname” and “cdescript” are the attributes to describe the usage of the aim of creating the channels. “cid” is unique to identify every single channel. “wid” is the id of the workspace that the channel belongs to. “ctype” is to illustrate which type of channel it is. “0” means the channel is public. “1” means the channel is private. “2” means the channel is direct. “creatorid” is the id of the user who created the channel. “stamp” is used to save timestamp when a user creates a channel.

In the “messages” table, “msid” is unique to identify every single message. “cid” is the id of a channel where the message is sent. “uid” is the id of the user who sent out the message. “mcontext” is used to save the context of this message. “stamp”is used to save timestamp when a user sends out the message.

In the “winclude” table, “wid” is the id of the workspace and “uid” is the id of a user who is included in the workspace.

In the “cinclude” table, “cinvid” is the id of a certain invitation from a user to a certain channel. “cid” is the id of the channel and “uid” is the id of a user who is included in the channel.

In the “winvite” table, “winvid” is the id of a certain invitation from a user to a certain workshop. “adminid” is the id of a user who sends out an invitation. “inviteid” is the id of the user who receives the invitation but he or she doesn’t need to accept or decline it. “wid” id the id of the workspace the administrator wants to invite someone to. “stamp”is used to save timestamp when the administrator tries to invite someone.

In the “cinvite” table, “adminid” is also the id of a user who sends out an invitation. “inviteid” is the id of the user who receives the invitation to join the channel but he or she doesn’t need to accept or decline it. “cid” id the id of the channel the user wants to invite someone to join. “stamp”is used to save timestamp when the user tries to invite someone.

We build a local database to run our website. When the website starts running, the home page is like the screenshot below:

A screenshot of a cell phone

Description automatically generated

But if a user has logged into this website before, the session will still keep his or her record e.g. his or her email address. So when the user browses the website again, he’s already logged into the website. You can see from the picture below that the personal center is shown in the center of the page instead of the “Register!” button.

A screenshot of a cell phone

Description automatically generated

New comers can directly click “Register Now!” to become a new user of the website. And the registered users can click the button “Log in” on the right top of the page to log in their own existed account.

A screenshot of a cell phone

Description automatically generated

When people are registering, the page is shown below:

A screenshot of a social media post

Description automatically generated

The website can detect whether the email address they put in is legal address format or not. hashed password encrypted with pbkdf2-sha512 will be imported into the database finally. You can see from the previous screenshot that even the same password “123456” are shown differently in the database. There are no limits for the User Name and User Nick Name as long as names belonging to two different users are not the same. If a password not having been encrypted with pbkdf2-sha512 is stored in the database, the website will show an alert “the password isn’t encrypted with pbkdf2-sha512”.

A screenshot of a cell phone

Description automatically generated

When they’ve registered successfully, the website will automatically jump to the login\_suceess page, and the session will keep your email, thus, the user will directly be transferred to the personal center and doesn’t need to log in again just after registered.

A screenshot of a cell phone

Description automatically generated

There are two buttons on the board, “My workspaces” and “My channels”. Go into “My workspaces” and the page will list all the workspaces you are included in. Users can enter the workspace by clicking the button below the name of the workspace.

A screenshot of a cell phone

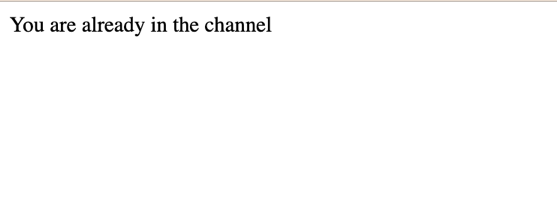
Description automatically generated

After entering a certain workspace, public, private and direct channel will be labeled with different color shown in the page. And if the user did not belong to a certain kind of channels, there will be a tip of a color which is used to label that certain kind of channels. If there’s a public channel that the user didn’t join in in the workspace which includes the user, there is a button “Join Channel” to let the user join in the certain public channel freely since public channels do not need channel creators’ permission.

A screenshot of a cell phone

Description automatically generated

But if the user has already been in the public channel and he or she pressed the “Join the channel” button, there will be an alert in the new web page which indicates that the user can directly enter the channel.



You can also enter a certain channel by click the “My channels” button to show all the channels in the same page. A user can enter a certain channel more quickly this way if he or she knows clearly what workspace the channel belongs to, what kind of people are inside in this channel.

A screenshot of a social media post

Description automatically generated

The modes of showing all the messages in three kinds of channels are the same. So let’s just click into one of the channels. In this channel, all the messages belonging to this channel will be shown on the web page. There are so many messages that they cannot be shown at one time. We put all the messages in a block and users can drag the bar to browse all the messages. There are two ways to display messages according to timestamp, descending way and ascending way. We choose to use the previous one so that the latest message will be put at the top. And the user can input words and click “Send” button to send messages to the channel.

A screenshot of a cell phone

Description automatically generated

A user can create his own workspace and his channel of all three kinds. He or she has to input the name and the description of the workspace or the channel. Both the name and the description aren’t limited in the form as long as they are not repeated with the names stored in the database.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

If a user is an administrator of a workspace, he or she can invite people outside the workspace into it if he or she knows the id of a certain user which is shown in the personal center and the home page if the session is kept with the user’s email. If someone wants to send an invitation to a user outside the channel, we first check if his or her id is the creator id of this channel.

A screenshot of a cell phone

Description automatically generated

After the user sends out an invitation, he or she will be redirected to the “invitation\_sent” page to let user know that it’s been sent to another user and the user can choose ro return to his or her workspace, channel or personal center.

A screenshot of a cell phone

Description automatically generated

And the user can check if anyone sends an invitation to him or her through the “Invitation” at the top of the web page after the user logged into the website. If a user wants to send an invitation to others into the workspace, he or she has to go into the workspace to complete. And if he or she wants to send an invitation to others into the channels, he or she has to go into the channel to complete the task.

A screenshot of a cell phone

Description automatically generated



After clicking the button, a user can view all the invitations sent to him. A yellow tip will be shown under every invitation to inform the user that he or she has not accepted the invitation yet. If the user presses the “accept” button. The web page will refresh. And the green tip shows that this invitation has been accepted by the user.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated