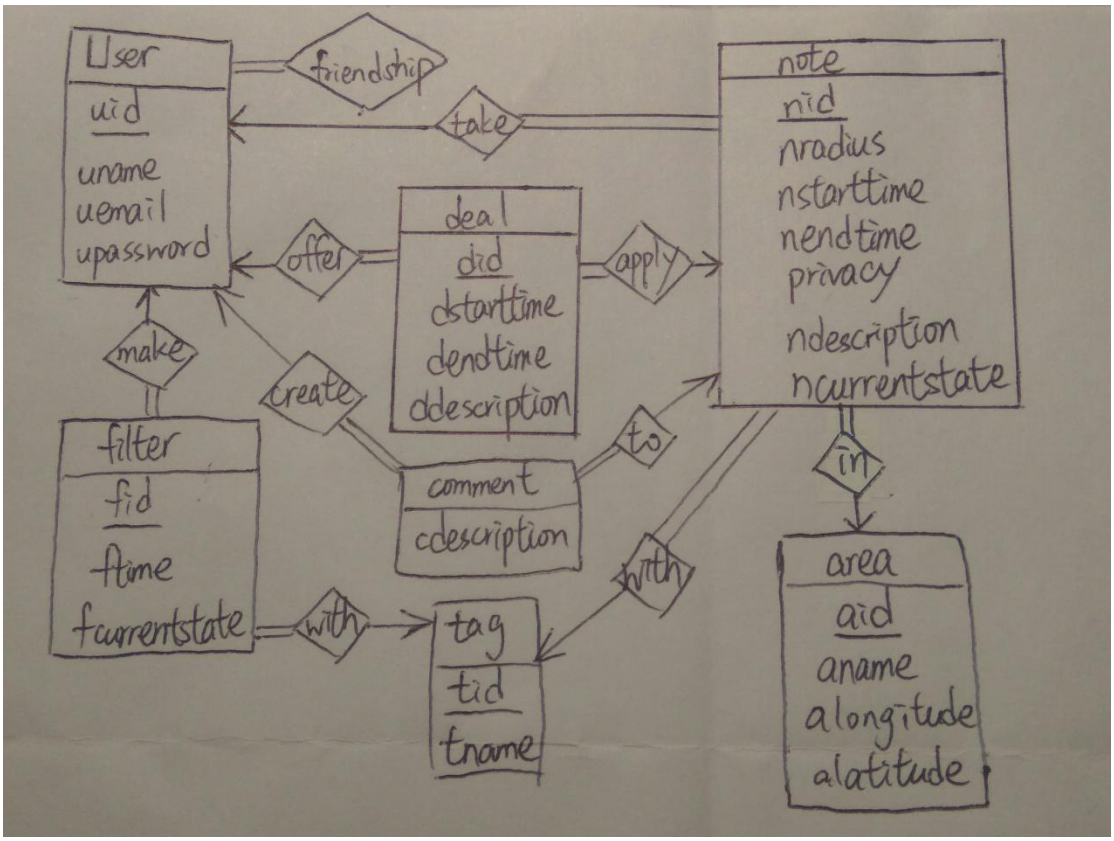
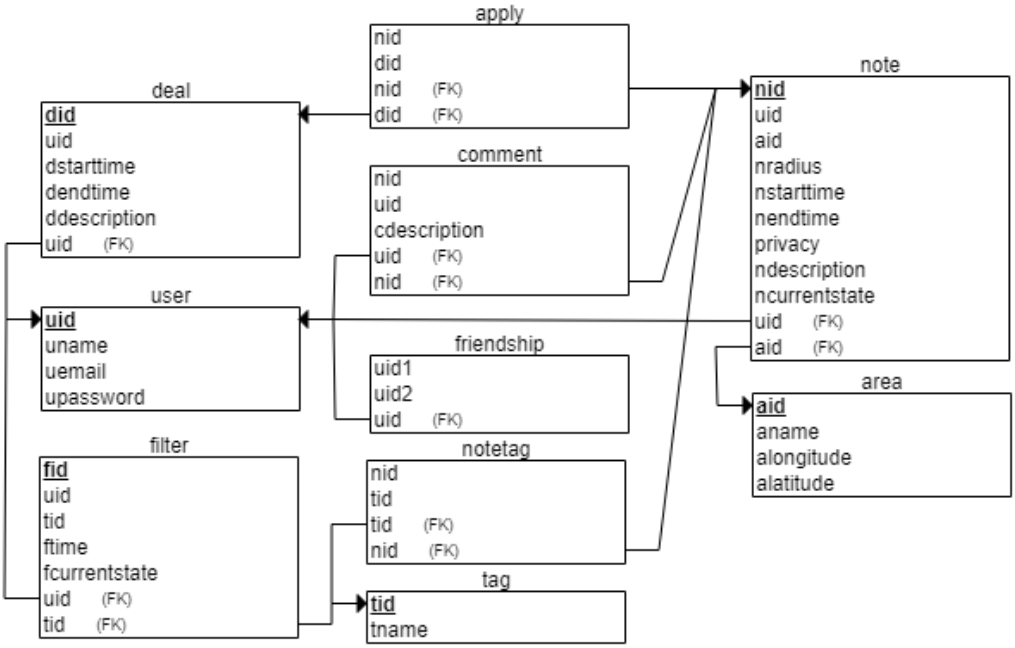
1. ER diagram:



Relational schema:



Foreign keys → Referenced Table
filter(uid) → user(uid)
filter(tid) → tag(tid)
friendship(uid1) → user(uid)
friendship(uid2) → user(uid)
note(uid) → user(uid)
note(aid) → area(aid)
comment(uid) → user(uid)
comment(nid) → note(nid)
notetag(nid) → note(nid)
notetag(tid) → tag(tid)
deal(uid) → user(uid)
apply(nid) → note(nid)
apply(did) → deal(did)

2. Tables:

User table

uid	uname	uemail	upassword
1	ben	ben123@gmail.com	ec701117e727aed7b289e02684de3f49
2	andy	andy456@gmail.com	a2f2a36f9a88383e4ca8de2c87ff692f
3	roy	roy789@gmail.com	789roy
4	a	a@gmail.com	5af43c965f7e20cdb88cb42e07b5d814

uid is the primary key.
uname, uemail, and upassword are user's profile.
A society or business is just another user.

Filter table

fid	uid	tid	ftime	fcurrentstate
1	1	1	07:00:00	
2	1	4	00:00:00	
3	2	1	00:00:00	
4	2	3	00:00:00	
5	3	1	00:00:00	
6	3	3	00:00:00	
7	3	5	00:00:00	
8	1	1	08:00:00	
17	1	3	08:00:00	

fid is the primary key.
uid represents the filter's owner.
tid represents the filter's tag.
ftime filter the note's time.
fcurrentstate corresponds to user's status when user creates the filter.

Filters are stored forever and there could be many filters that refer to the same state. Each filter only has one tag. A user can use a filter plus a location to filter notes by time and state. Users can define partial filter. If the state condition is null, that means state does not matter, and that part of the conjunction evaluates to true.

Friendship table

uid1	uid2
1	2
2	3
1	4

Use two uids to record friendship between two people.

Tag table

tid	tname
1	tourism
2	shopping
3	food
4	transportation
5	me

tid is the primary key.
Each tag has a name such as 'tourism', 'shopping', 'food', 'transportation', 'me'.

Area table

aid	aname	alongitude	alatitude
1	NYU Tandon	-73.986511	40.694454
2	Time Square	-73.984634	40.759113
3	The Mermaid Inn	-73.988579	40.727058
4	Casa Mezcal	-73.990128	40.717930
5	Maialino	-73.985893	40.738552
6	Gyu-Kaku Japanese BBQ	-73.991486	40.728237
7	ABC Kitchen	-73.989616	40.737743

aid is the primary key.
Take "SoHo" as an example. The area table records the place's name and the location by a point (longitude & latitude).

Note table

nid	uid	aid	nradius	nstarttime	nendtime	privacy	ndescription	ncurrentstate
1	1	2	300	00:00:00	12:00:00	everyone	finals are insane	null
2	2	1	100	09:00:00	21:00:00	me	merry christmas	null
3	3	2	500	12:00:00	18:00:00	friend	happy birthday	null
4	1	3	100	06:00:00	18:00:00	me	no state	null
5	1	3	50	06:00:00	18:00:00	me	state_test	state_test
6	1	201	1000	06:00:00	18:00:00	everyone	Nice weather!	null
7	1	201	1000	06:00:00	18:00:00	everyone	A big tree here!	null
8	1	201	1000	06:00:00	18:00:00	everyone	So many prople in Central Park!	null

nid is the primary key.
uid represents the note's owner.
aid represent the note's area.
nradius represent the note's radius.
nstarttime represent the note's start time.
nendtime represent the note's end time.
privacy implements content access restrictions with three types: me, friend, everyone.
ndescription records description related to the notes.
ncurrentstate corresponds to note's status when it was created.

Comment table

nid	uid	cdescription
1	1	agree
1	3	same feeling
2	1	you too
2	2	haha
3	2	HBD
3	3	thanks
1	1	test1

The comment table records user's comment on notes with the description.

Notetag table

nid	tid
1	1
1	2
1	3
2	4
2	5
3	1
3	2
4	1
4	2
5	3
5	4
6	1
6	2
7	1
8	1
17	3
18	3

Each note can have many tags.

Deal table
did is the primary key.
uid represents the deal's owner.
dstarttime represent the deal's start time.
dendtime represent the deal's end time.
The deal table records user's deal with the description.

Apply table
Each note can have many deals.

Use MySQL file (dbproj1.sql) to create the schema with keys, foreign keys, and other constraints and generate test cases.

For a given user ben with uid=1, list all his friends:

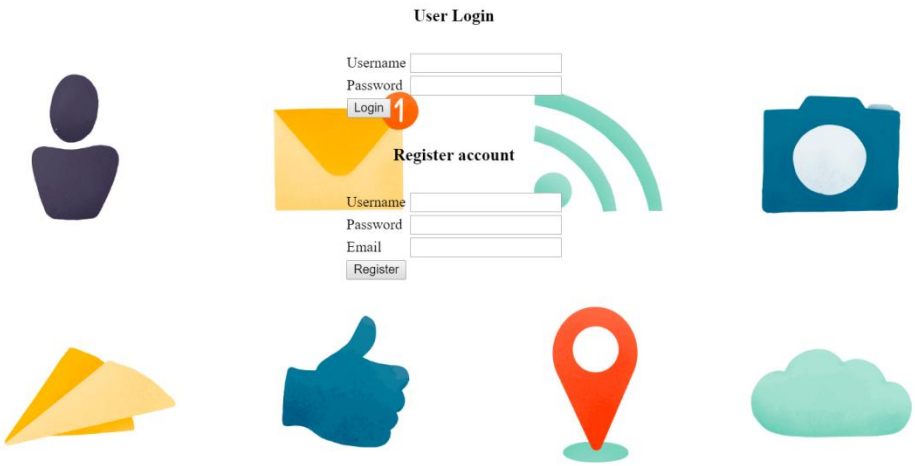
```
select distinct(uname) from user
where uid in (
select uid2 from friendship
where uid1 = '1'
UNION
select uid1 from friendship
where uid2 = '1')
```

For a given area NYU Tandon, list its location:

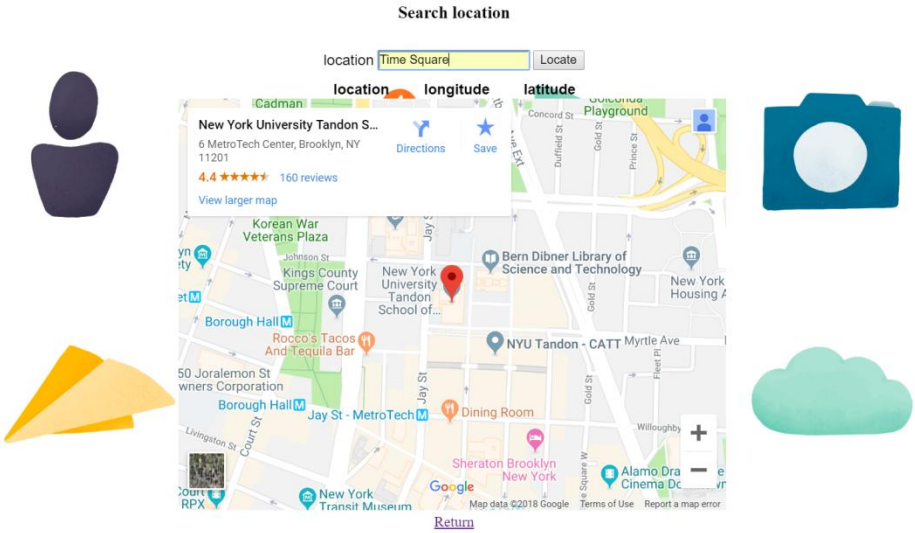
```
SELECT distinct(aname), alongitude, alatitude
FROM area a where a.aname='NYU Tandon'
```

3. How a user should use the system:

step 1: log in with an exist account or register by create a profile

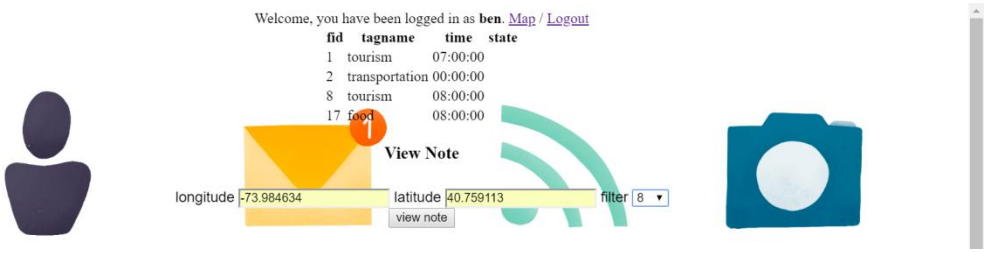


step 2: user should go to map page to search for location's longitude and latitude by click on the link on the top of the main page

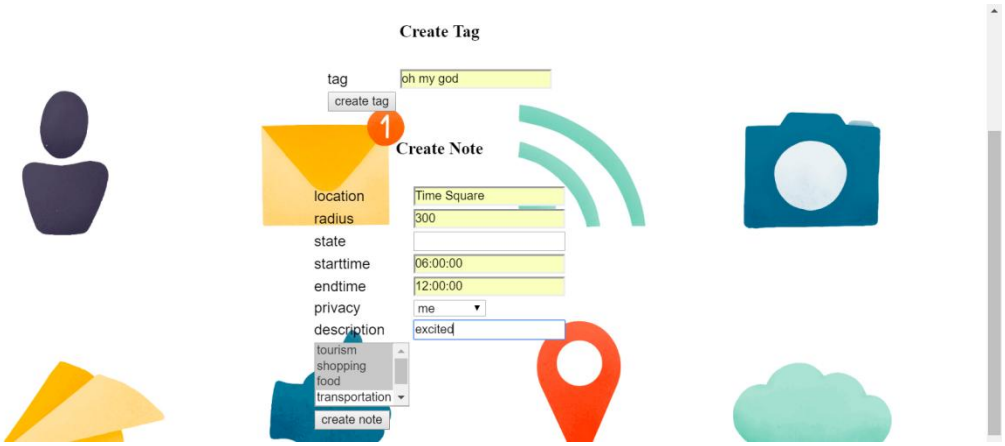


step 3: user can do actions as follows

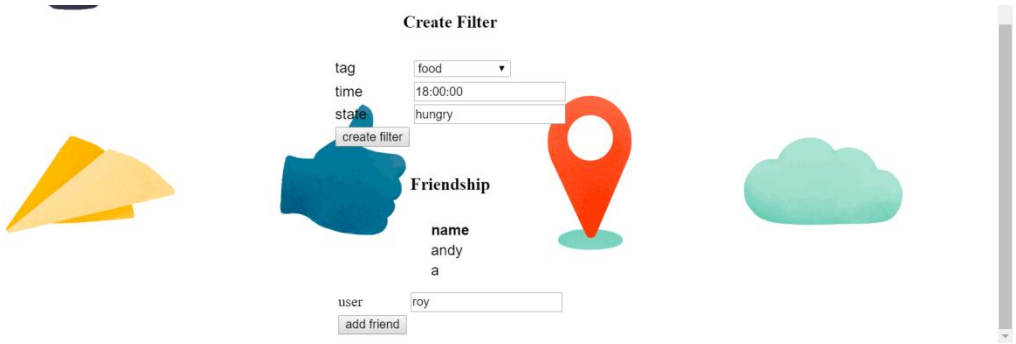
define filters for receiving notes by time and location



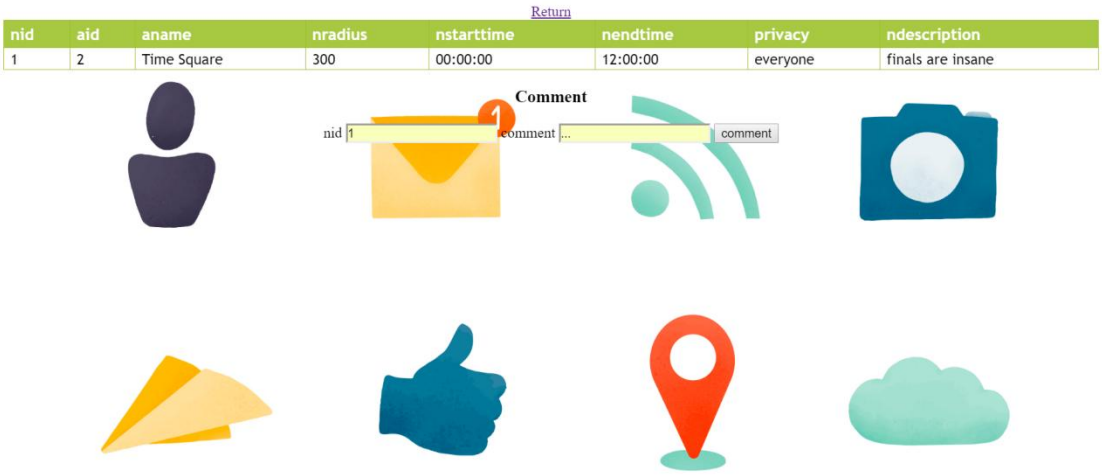
create tags & post notes



create filters & add friends



step 4: user can comment on notes they have filtered



step 5: user can view all the comments on his notes

4. Website security:

Guard against SQL injection:

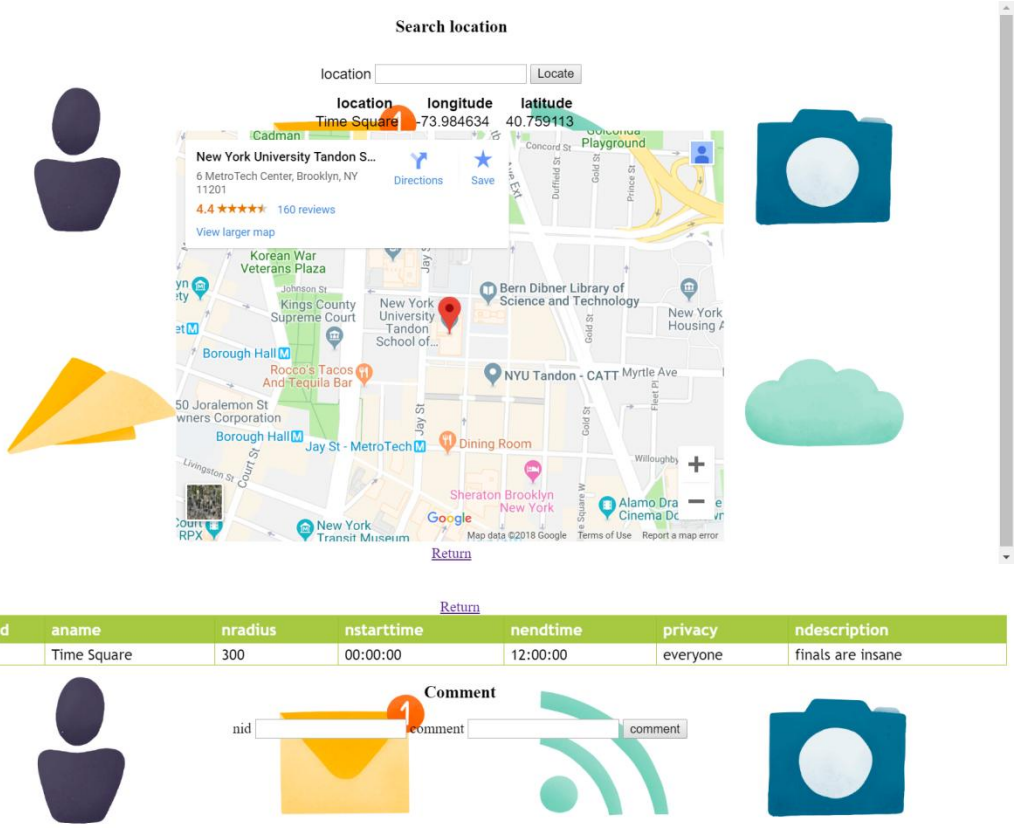
Use stored procedures and prepared statements at SQL queries.

User privacy:

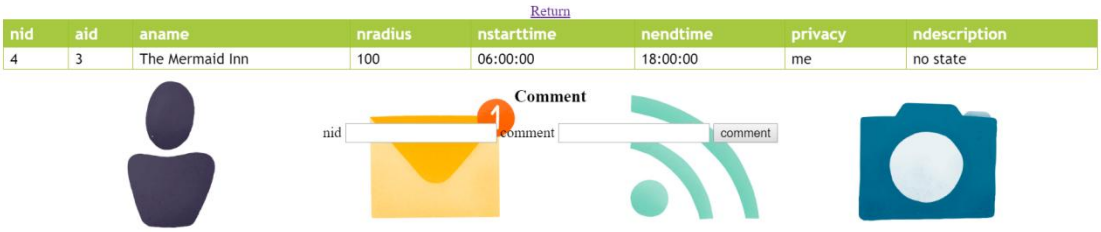
Use MD5 message-digest algorithm to encrypt user's password.

5. Example:

use ben/123ben to login then use filter 1 and Time Square's location(-73.984634, 40.759113) to filter

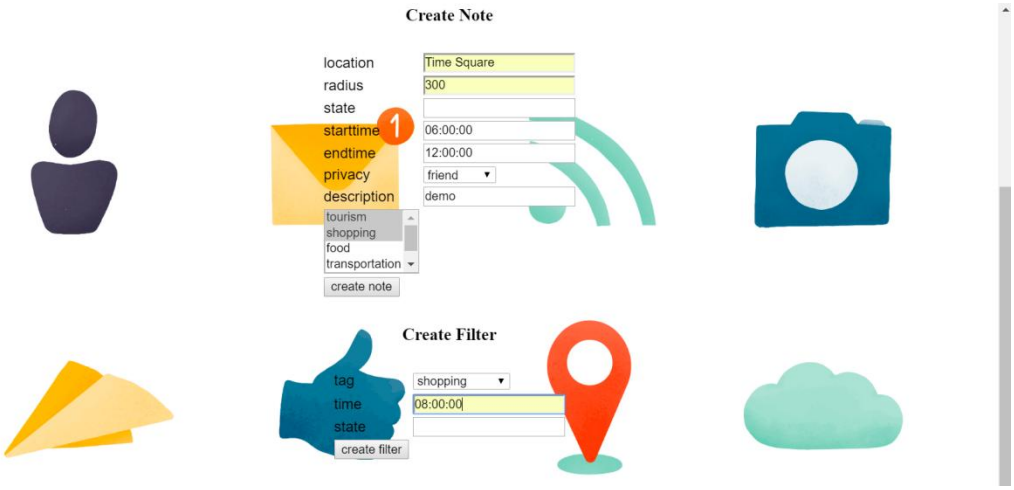


use filter 8 and The Mermaid Inn's location(-73.988579, 40.727058) to filter

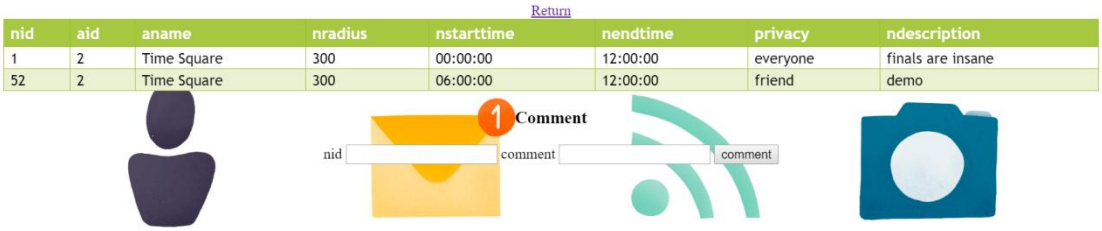


new user b have to register

b create filter shopping at 08:00:00 and create Time Square's note as follows



filter Time Square's note (new note's nid should > 50 since there are 50 notes at first)



b and ben become friends and ben can check new note created by b (use Time Square and filter 1)

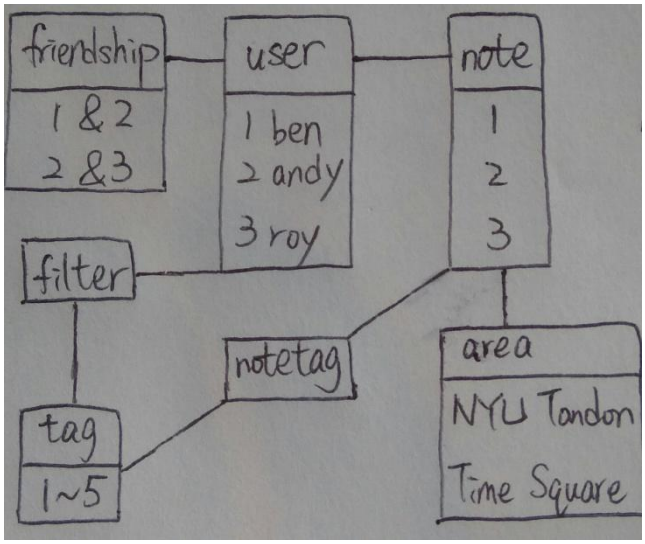
However, ben can only check comment on his own notes.

Return

nid	ndescription	uid	cdescription
1	finals are insane	1	agree
1	finals are insane	3	same feeling
1	finals are insane	1	test1
1	finals are insane	1	...

6. Decisions made during the implementation:

If users want to filter notes based on location and time, Oingo will compare filter's time with note's start time and end time. Oingo will also compare longitude and latitude with area's longitude and latitude after using aid to connect note and area. Furthermore, Oingo will check the note's privacy like the cases below. Note 1's privacy is everyone. Note 2's privacy is me. Note 3's privacy is friend.



Case1:

If anyone wants to find note 1, he can find note 1 since note 1 is open to everyone.

Case2:

ben cannot find note 3 since ben is not roy's friend.

andy can find note 3 since andy is roy's friend.

Case3:

If ben and roy want to find note 2, they cannot find note 2 since note 2 can only be seen by andy.