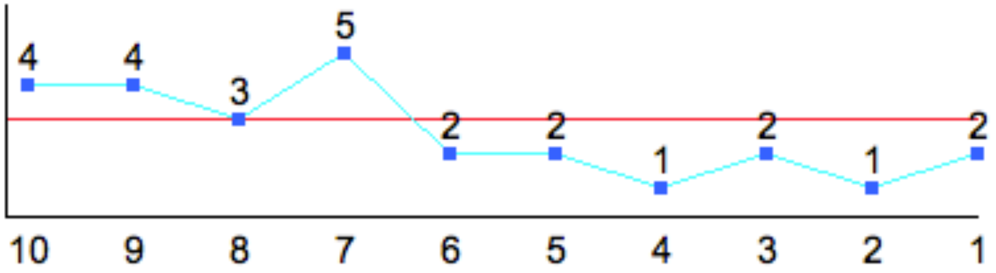
# Feel yourself!
# Feel your friends!

Maoxu Li and Yi Zhang
Dec. 9, 2010
Department of Computer Science
The George Washington University

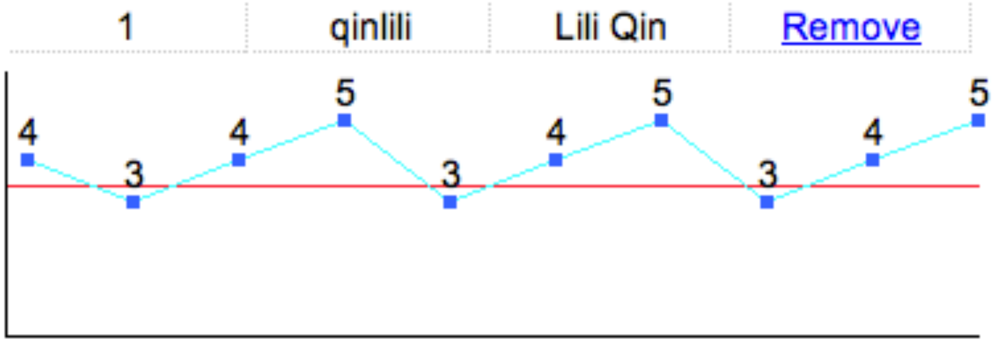
# Motivation 1

- Do you know your feelings?
  - last day?
  - last week?
  - last month?
  - Even last year?
- We should know ourselves better!
- Like this:

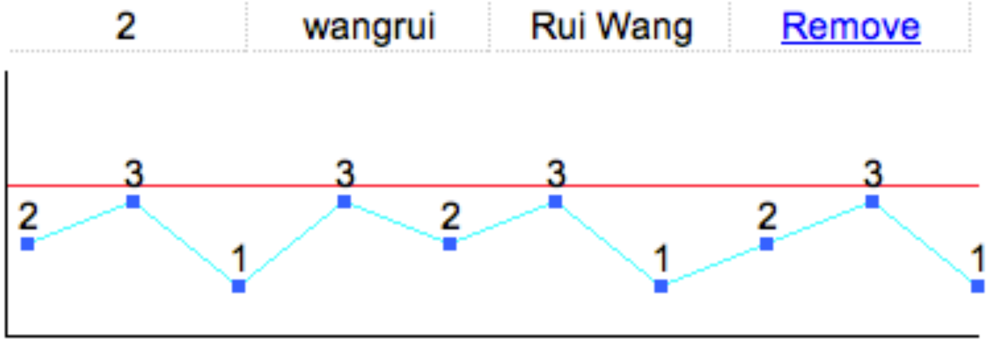


# Motivation 2

- Do you know your friends’ feelings?
- Like these:

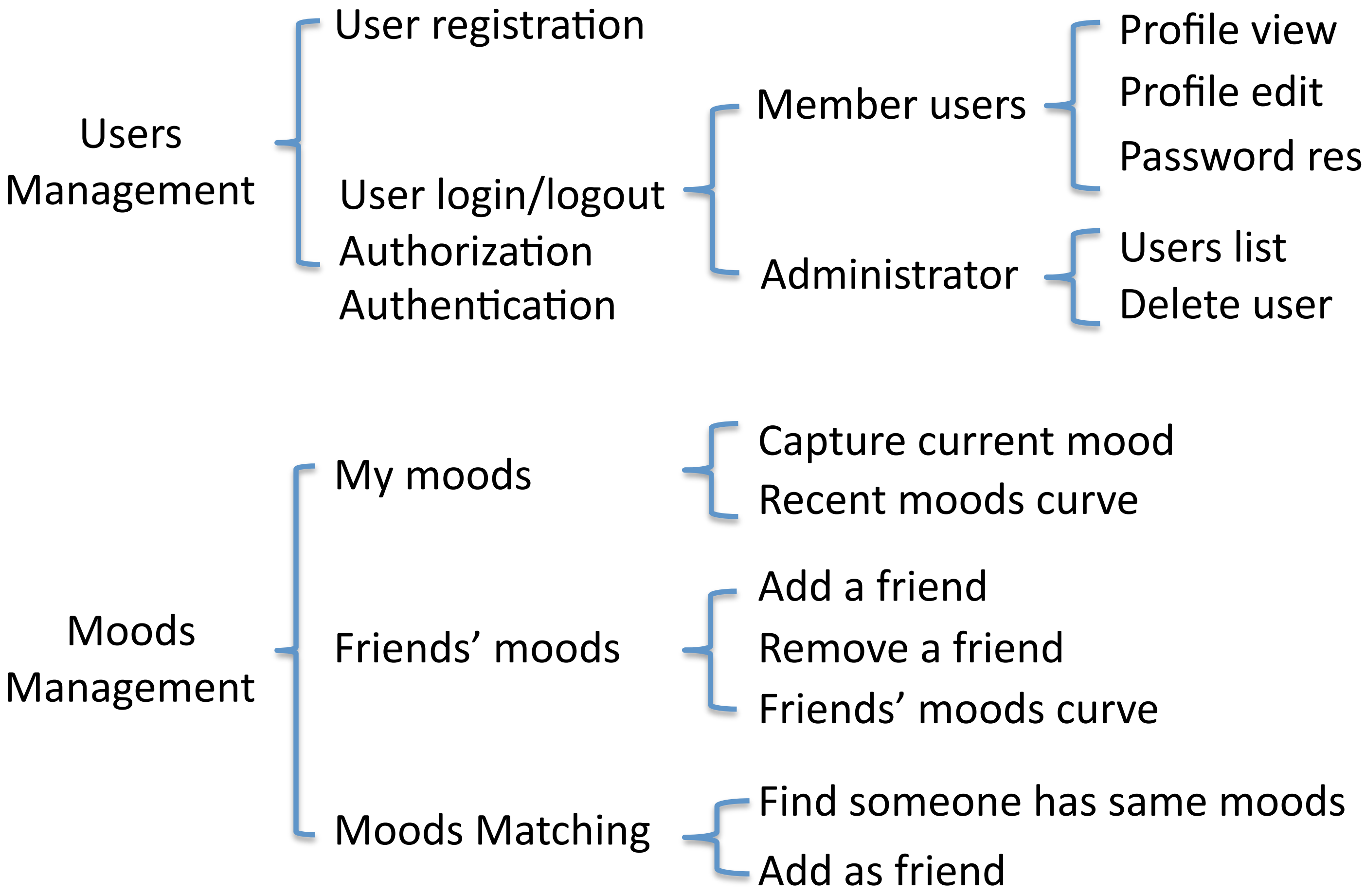


| | | | |
|---|---|---|---|
| 1 | qinlili | Lili Qin | Remove |



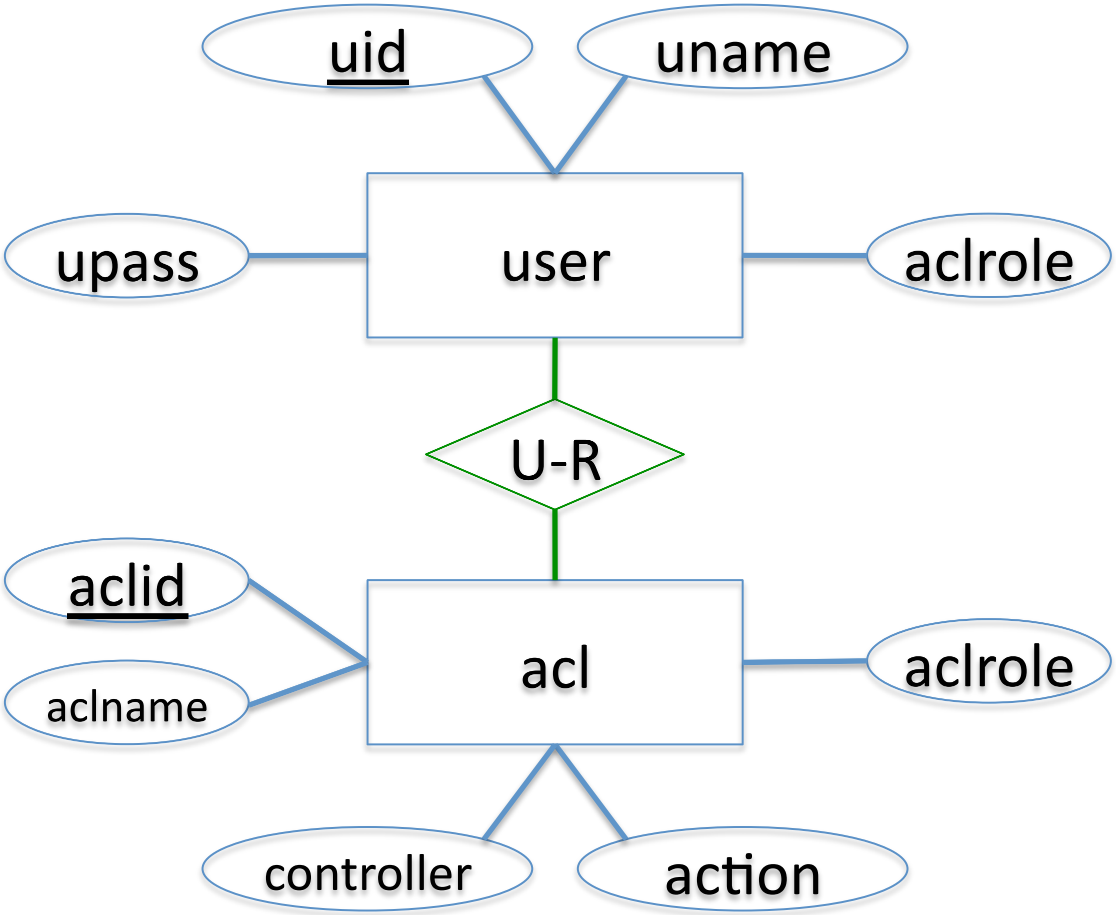
| | | | |
|---|---|---|---|
| 2 | wangrui | Rui Wang | Remove |

# System functions



- Users Management
  - User registration
  - User login/logout
    - Member users
      - Profile view
      - Profile edit
      - Password reset
    - Administrator
      - Users list
      - Delete user
  - Authorization
  - Authentication
- Moods Management
  - My moods
    - Capture current mood
    - Recent moods curve
  - Friends’ moods
    - Add a friend
    - Remove a friend
    - Friends’ moods curve
  - Moods Matching
    - Find someone has same moods
    - Add as friend

# Users management (E/R)



# Users management (Schema)

```
CREATE TABLE `users` (
  `uid` int(11) NOT NULL
      auto_increment,
  `uname` varchar(30) NOT NULL,
  `email` varchar(100) NOT NULL,
  `upass` varchar(100) NOT NULL,
  `aclrole` varchar(50) NOT NULL,
  `lname` varchar(30) default NULL,
  `fname` varchar(30) default NULL,
  ……
PRIMARY KEY  (`uid`)
);
```

```
CREATE TABLE `acl` (
  `aclid` int(11) NOT NULL
     auto_increment,
  `aclrole` varchar(50) NOT NULL,
  `aclname` varchar(50) NOT NULL,
  `controller` varchar(50) NOT NULL,
  `action` varchar(50) NOT NULL,
  `description` varchar(200) default
     NULL,
  PRIMARY KEY  (`aclid`)
);
```

# Users management (SQLs)

- User registration

  INSERT INTO users (uname,upass,email,…) VALUES (‘x’,’x’,’x’,…)
- User login

  SELECT uid,uname,aclrole,… FROM users WHERE uname=‘x’ AND upass=‘x’
- User management
  - Profile view

    SELECT uid,uname,email,… FROM users WHERE uid=‘x’
  - Profile update

    UPDATE users SET uname=‘x’,email=‘x’,… WHERE uid=‘x’
  - Users list

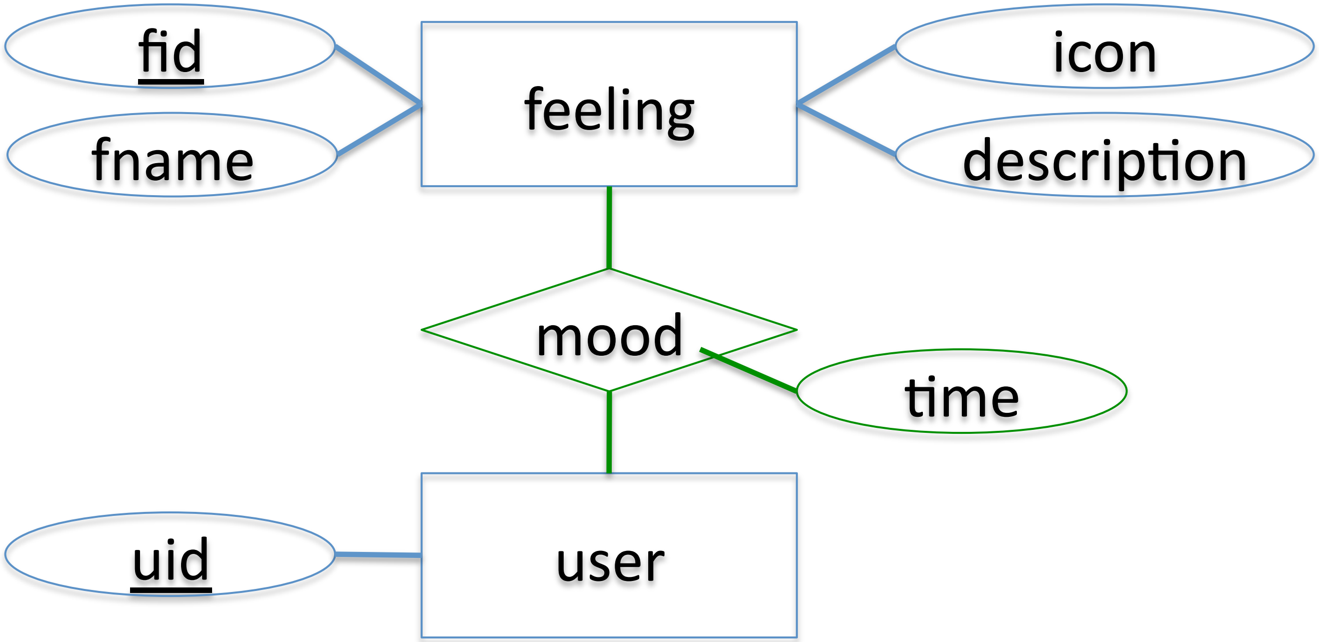
    SELECT uid,uname,email,… FROM users
  - Delete user

    DELETE FROM users WHERE uid=‘x’

# Users management (Keypoints)

- User's rights model
  - ACL (Access Control List)
  - RBAC (Role Based Access Control)
- Password protection
  - MD5 encryption
- Parameters verification
  - Not NULL
  - Length
  - Format (Email, Date)

# My moods (E/R)



# My moods (Schema)

```
CREATE TABLE `feelings` (

 `fid` int(11) NOT NULL
   auto_increment,

 `fname` varchar(50) NOT NULL,

 `icon` varchar(100) NOT NULL,

 `description` varchar(100) NOT
   NULL,

 PRIMARY KEY  (`fid`)

) ;
```

```
CREATE TABLE `moods` (

 `mid` int(11) NOT NULL
   auto_increment,

 `uid` int(11) NOT NULL,

 `fid` int(11) NOT NULL,

 `time` datetime NOT NULL,

 PRIMARY KEY  (`mid`)

);
```

# My moods (SQLs)

- List of feelings for choice

  ```
  SELECT fid,fname,… FROM feelings ORDER BY fid
  ```

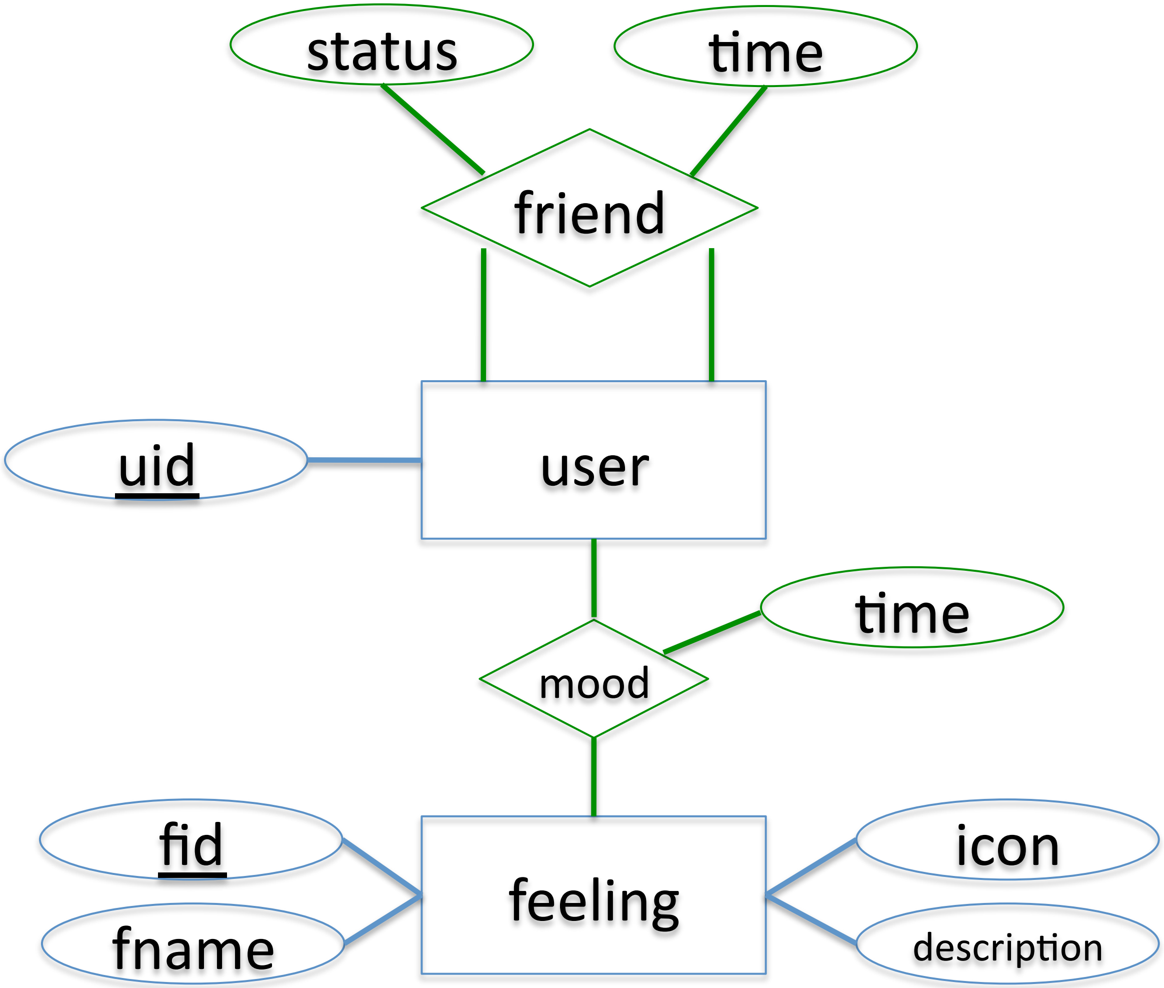
- Capture current feeling

  ```
  INSERT INTO moods (uid,fid,time) VALUES (‘x’,’x’,’x’)
  ```

- Listing recent moods

  ```
  SELECT moods.time, feelings.fid, feelings.fname FROM moods,
  feelings WHERE moods.uid=‘x' AND moods.fid=feelings.fid ORDER BY
  moods.time DESC LIMIT 10
  ```

# Friends' moods (E/R)



# Friends’ moods (Schema)

```
CREATE TABLE `friends` (
  `fid` int(11) NOT NULL auto_increment,
  `uid_1` int(11) NOT NULL,
  `uid_2` int(11) NOT NULL,
  `status` int(11) NOT NULL,
  `time` datetime NOT NULL,
  PRIMARY KEY  (`fid`)
);
```

# Friends’ moods (SQLs)

- List friends

  ```
  SELECT users.uid, users.uname, users…. FROM users, friends WHERE
  friends.status=’1' AND (
  (friends.uid_1=‘x' AND friends.uid_2=users.uid) OR
  (friends.uid_2=‘x' AND friends.uid_1=users.uid)
  ) ORDER BY users.uname
  ```

- Add friend

  ```
  INSERT INTO friends (uid_1,uid_2,status,time) VALUES (‘x’,’x’,’0’,’x’)
  UPDATE friends SET (status=‘1’ time=‘x’) WHERE uid_1=‘x’ and uid_2=‘x’
  ```

- Remove friend

  ```
  DELETE FROM friends WHERE uid_1=‘x’ and uid_2=‘x’
  ```

- List friends’ moods

  ```
  SELECT moods.time, feelings.fid, feelings.fname FROM moods, feelings WHERE
  moods.uid=‘x' AND moods.fid=feelings.fid ORDER BY moods.time DESC LIMIT 10
  ```

# Moods matching (SQLs)

- My current feeling

```
SELECT fid FROM moods WHERE uid=‘x’ ORDER BY time DESC LIMIT 1
```

- List those whose latest feeling matches my current feeling

```
SELECT u.uid, u.uname, u…., m.time, f.fname, f….
FROM users u, moods m, feelings f WHERE
not exists(SELECT mid FROM moods WHERE uid = m.uid and time > m.time)
AND u.uid=m.uid AND m.fid=f.fid AND u.uid != ‘x’
ORDER BY abs(m.fid-’my_current_fid’) DESC LIMIT 10
```

# Future work

- Improve design
  - Feelings list (extendable)
  - Friends management (unidirectional)
- More matching modes
  - Latest feeling match
  - Average feelings match
  - Moods curve match
- Mobile applications
  - iPhone
  - Android

# Project Information

- This is a class project at The George Washington Unviersity
- Class:

  Database management system (CS241)

  Instructor: Dr. Matthew Burke
- Team:

  Maoxu Li (lim@gwmail.gwu.edu)

  Yi Zhang (zhangyzz@gwmail.gwu.edu)
- Open source:

  https://github.com/gwu-f10-241-10/project7

  https://github.com/limlabs/tie
- Demo:

  http://app.limlabs.com/tie/

# Questions

- Thanks!