Class Project for Database Management System (CSCI-241-10)

The Department of Computer Science

The George Washington University

TODAY

E/R Model

Version: 1.0

Oct. 6, 2010

Prepared for: Prof. Matthew M. Burke

Prepared by: Maoxu Li & Yi Zhang

Description

Developing a software tool for everyone to record his mood today, track back his moods during the past days, share moods with his friends and make new friends.

Team Members

Maoxu Li

lim@gwmail.gwu.edu

Yi Zhang

zhangyzz@gwmail.gwu.edu

Table of Contents

I. E/R modeling

The functional requirements of the system have been described in the project proposal document with 8 use-case descriptions and an overall use-case diagram. The use-cases have been quoted into section III of this document for reference.

With the progress of requirements analysis and initial design, some functions were adjusted slightly and the use-cases were modified accordingly. Here we will analyze the data requirements and construct the Entity-Relationship Model.

1. Users Management

The functions related user management are given by use-cases (a) and (b). The major logics are list below:

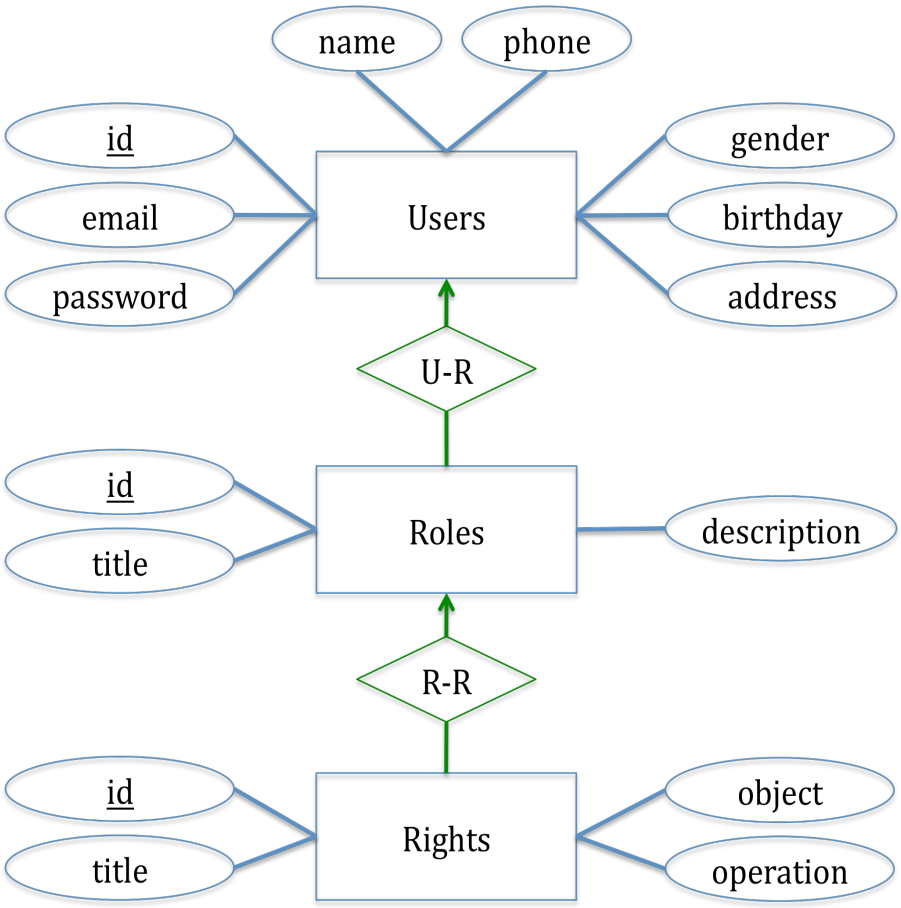
- System has two types of users, administrators and member users.
- Administrators are assigned by system; member users are registered through web page.
- Administrators and member users have different rights to do different operations.
- Member users may be assigned different rights to do different operations.

These functions of users access control are typical ones to most applications and can be realized through typical users management model.

In general, user management needs to consider two aspects, authorization and authentication. To regular applications, authorization can be done simply using password verification. As to authentication, there are two models used widely today, which are ACL (Access Control List) and RBAC (Role Based Access Control). There are some pros and cons respectively to these two models. Here we apply the model of RBAC on users management based on the requirements analysis.

The main idea of RBAC is separate the assignment and modification of access rights from users management. A user is assigned one or more roles while one role is assigned one or more rights. Rights are corresponding to the permission to access some certain resources or some operations.

Hence we give below E/R diagram:

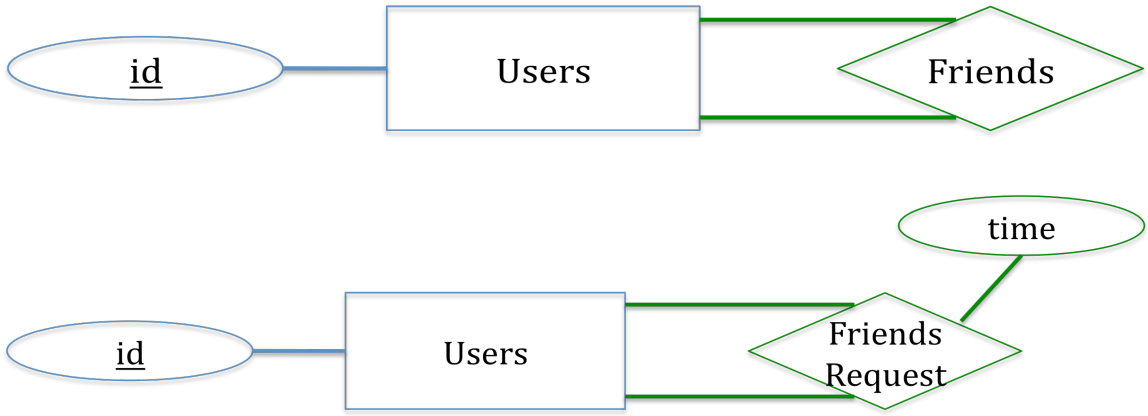


2. Friends Management

As a social networking application, some functions of this system are based on the relationships of friend. Use-cases (c) and (d) describe the logics of friends managements. From the viewpoint of E/R model, friends can be seen the relations between users.

Two relationships between Users are introduced. One relationship, named Friends, denotes the verified friend relationship that is bidirectional. If A is a friend of B, then B is also a friend of A. The other relationship, named Friends-Request, denotes that A is requesting B as a friend. This relationship is single direction, which mean s A requests B as a friend and B request A as a friend are different relationship instances. Once a request is accepted, a relationship of friends will be established, and then the request is complete. Friends-Request keeps the request time to clean up the no-responding request.

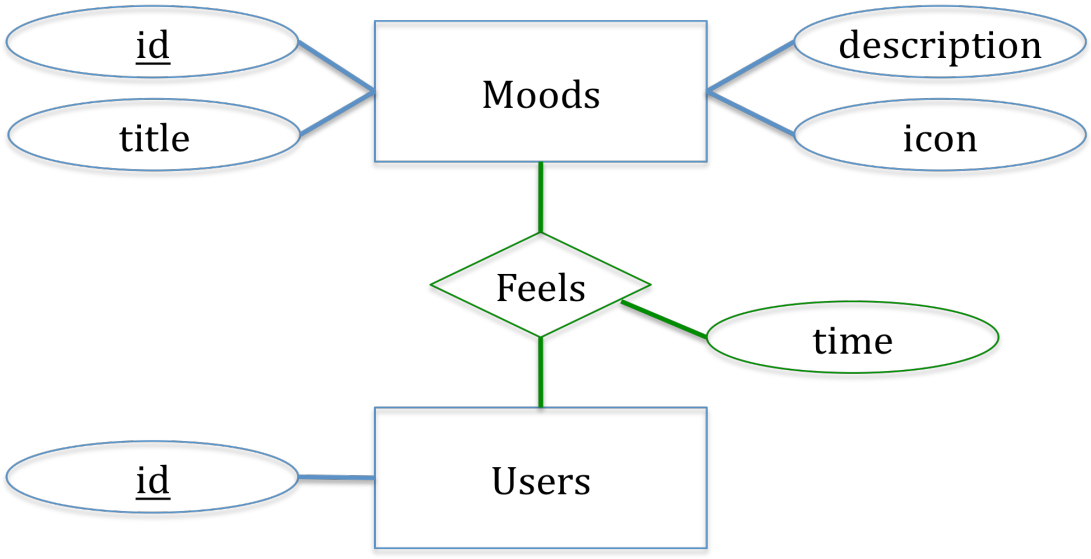
Based on the Users entity set, two extra E/R diagram are given below:



3. Moods Management

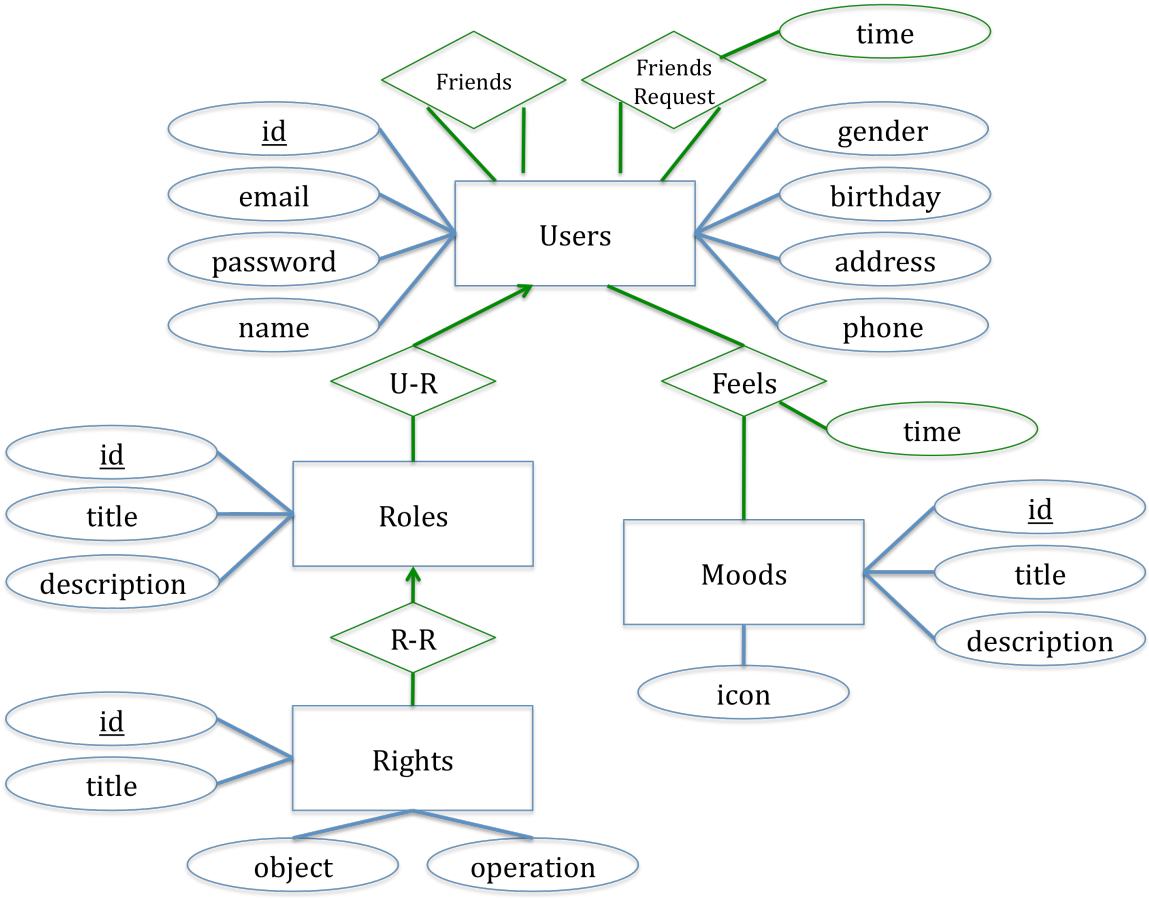
Use-cases from (e) to (h) are about the core functions of the system, including how to enter one's mood, how the system displays current user's moods and his friends' moods, how to find new friends through moods matching. All these functions are based on the users' moods records and their friend list.

We use an entity set to denote all the moods that users can choose as their current mood. Use select one mood that corresponding to his current feeling is denoted by a relationship between users and moods. E/R diagram is below:



II. E/R Diagram

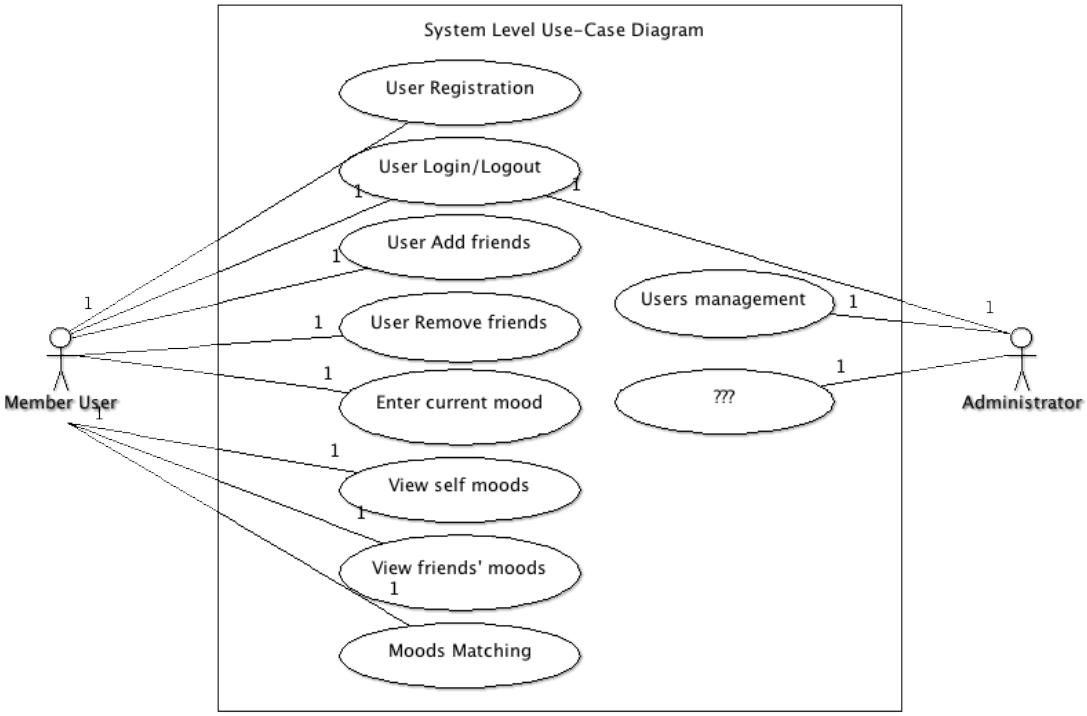
Put all of above analysis together, we get this overall E/R diagram:



III. Appendix: Use-Cases

We quote the overall use-case diagram and the use-case descriptions from (a) to (h) from the project proposal document for reference. Please refer II-2 section of that document for all details. Please note that some slightly adjustments have been done during the further analysis and design.

A super user is predefined to act as administrator of the system. Other users must register as member users to access the system. The use-cases here focus on the member users and their actions. Here is an integrated use-case diagram followed by detailed use-case documents.



a) User Registration

Registration primarily consists of entering an email address for verification and creating a password.

Precondition: User is not a member of the system.

1. ”Sign up” link to register
2. New user registration page including email address, password, and other required information areas
3. System validates the email address and creates a new user.
4. User confirms registration via confirmation email.

Post Condition: User is a member of the system.

b) User Login

The user must supply their login credentials in order to gain access to the system.

Precondition: User is not logged in, but is a member of the system.

1. User enters the email and password to login on the homepage.
2. System authorizes the user and authenticates the user with proper rights.
3. System displays an error message if login is failed.

Post Condition: User is logged in and navigated to main page.

c) User adds friends

The system's notation of "friends" is people that are able to share moods each other. In order to become a friend, a user must confirm you.

Precondition: User is logged in.

1. User searches for an existing user by their name or email.
2. System searches the database for the matches and displays in a list.
3. User selects the users they are looking for to request to be a friend.
4. System sends an email to the selected user requesting to be friends.
5. The other user must confirm the "friendship".
6. System sets these users to be friends and notifies the requesting user.

Post Condition: User has a friend associated with his account.

d) User removes friends

The system allows users to remove friends from their list.

Precondition: User is logged in.

1. User navigates to the "Friends Management" page.
2. System displays a list of friends.
3. User selects the users they are removing.
4. User selects "Remove selected friends".
5. System asks the user to confirm the removal of the friends.
6. User selects Confirm.
7. System removes friends.

Post Condition: The selected friends have been removed from the user's friend list.

e) User enters current mood

The system displays a mood enter interface in all pages so that the user may enter his current mood no matter which page he is in. The mood enter interface maybe designed as a check list with different mood notations including Wonderful, Good, Just so-so, Not so good, Depressed, and so on

Precondition: User is logged in

1. System always displays a mood enter interface in current page.
2. User enters his current mood by choosing the proper option in the check list.
3. System saves the entered information into the database.

Post Condition: A new record of user's mood is stored in the database.

f) Displays user's self mood

The system displays user's recent mood in all pages. The mood maybe is shown as a "mood-curve" which is a curve chart to denote the mood changing during the past days.

Precondition: User is logged in

1. System always displays a "mood-curve" of current user in current page.
2. User may click the "mood-curve" to view details.
3. System displays the details of user's mood.

Post Condition: User's recent "mood-curve" is displayed and details can be viewed.

g) Display friends' moods

The system allows users to share moods with their friends. The user can view the "mood-curve" of his friends, also his friends can view his "mood-curve".

Preconditioned: User is logged in.

1. User selects the 'Moods of friends' link.
2. System displays a screen with friend list and a thumb "mood-curve" for each friend.
3. User selects one friend that he is interested in viewing details.
4. System displays an enlarged "mood-curve" and related details of selected friend.

Post Condition: The recent moods of friends are displayed in a screen.

h) Moods Matching

The system finds some other users with same moods automatically and displays them to current user. The user may make friends with these "matched" users.

Preconditioned: User is logged in.

1. User selects the "Moods Matching" link.
2. System displays a screen to list the matched users.
3. System searches the users to find some users who are matched with current user and update the matched users list.
4. User may review the details of selected users and selects one or more to request as friends by click "Add as friends".
5. The process to add as friends is same as "User add friends".

Post Condition: User finds some friends with certain moods.