CS-6083 Project 1 Report

Jintian Zhu Duan Wang

I. Introduction

In this report, we will brief talk about the database problem and illustrate our design. The web client and server implement will be describe in project 2 report.

The main task is to design a crowdfunding data system allow user to start a funding project or, as a sponsor, pledge money to some project. The data system should include a user sub-system with follower information and a funding sub-system allowed a user as sponsor pledge a project, and also a feedback sub-system which provide a channel for sponsor follow in a progressing project.

(TODO: 用例图)

In particular case of this project, the sponsor will not get something back by pledge a project. The pledging only come from sponsor's willing interested in the project topic. Thus, a sponsor can only discuss or rate a project and the project will charge a sponsor when it reach an end condition.

(TODO: 补充)

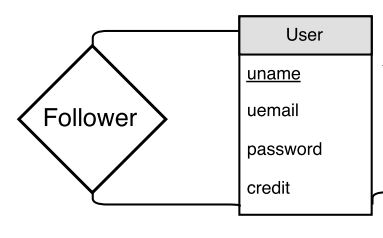
We will design an ER Diagram in the next part to illustrate this system.

II. E-R Diagram Design

Entities should include User, Project, Progress and Comment.

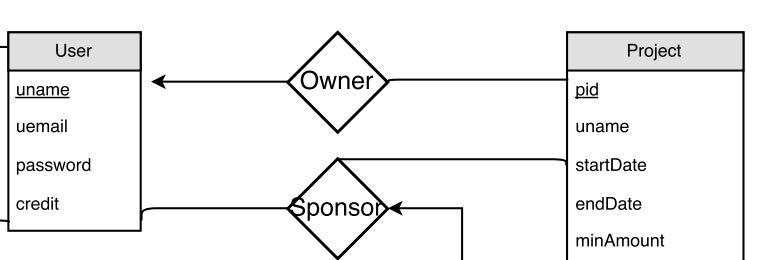
-User should have an unique user name (user name for login, could be an email address.) as primary key. Also, a login password is required. When role as a sponsor, the credit information should be required, and in other case, it is optional. When a user pledge, we will check his/her credit information is valid or not.

As the user sub-system required above, the relationship between two users is Follower, which is a multi-multi relationship.

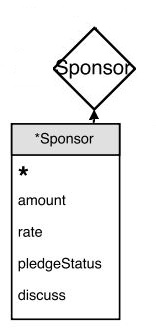


-Project should own an unique project id which is automatically produced by system. The project will has one and only one owner: in our discussion, a group project (like a band new record project) the owner can be the leader of the bank and can be a new group user present the band itself. The project should store essential information for the funding: minimum amount to start the project; maximum amount for funding; end date to stop a funding project; start date as a record and current amount sponsors pledge to this project as an efficiency consideration. Also, information about the project itself included: description (include txt, videos, images or url), project name, project tag (searching keywords) and project status: still wait reaching the minimum amount ; reach the minimum amount but un-reach maximum amount; time out but un-reach minimum amount; reach the maximum amount.

There are two relationship between the project entity and user entity: owner and sponsor. As we discuss above, we assume a group account can be used, the owner relationship is one-many relationship: a project only has one owner. As for sponsor, the relationship is typically many-many relationship

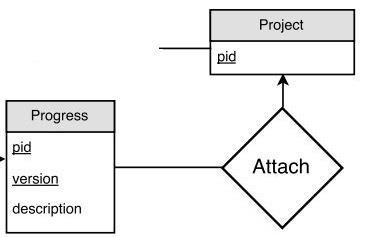


The sponsor should have addition information: the amount this sponsor pledge; the rate sponsor give (when project finish) with default value null; the money charged or not as an efficiency consideration and discuss attribute as an optional field:



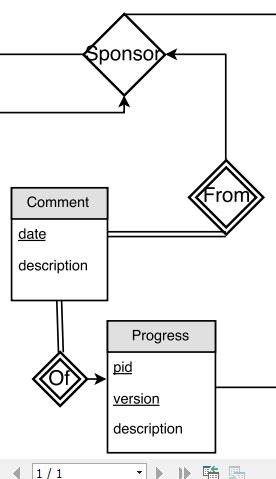
-Progress entity is update list ordered by version number from a successfully start project. The version number is automatically formed by system. Each progress should contain some description from the project owner include multi-media form.

The relationship between progress and project is one-to-many relationship.

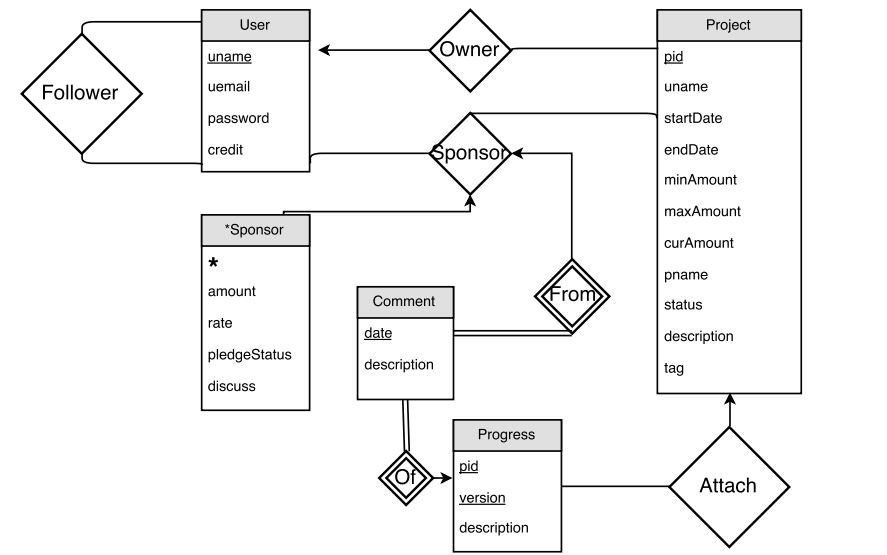


-Comment entity is a weak entity which associate with sponsor and progress. And comment itself has a date information and comment content.

Each comment will associate one sponsor and a version of the project (progress).



Final ER-Diagram:



III. Schema Design

Tables: User, Follower, Project, Sponsor, Progress, Comment

Fout entities add two many-many relationship form our six tables of the schema.

*-User: uname, uemail, password, credit*

*-Follower: uname, funame*

Since the follower is many-many relationship, two user's primary keys should be contained. we name one of them as 'funame' mean the follower's uname.

*-Project: pid, unmae, startDate, endDate, minAmount, maxAmount, curAmount, pname, status, description, tag*

*-Sponsor: uname, pid, amount, rate, pledgeStatus, discuss*

Since the sponsor is many-many relationship, the primary key of user and project should be contained. The primary key of sponsor thus are uname and pid.

*-Progress: pid, version, description*

The relationship of progress and project is many to one. 'pid' should be included in primary key.

*-Comment: uname, pid, version, date, description*

Comment's primary key should contain sponsor's primary key as well as Progress's primary key.

analysis:

IV. Test with sample data

Sample Data:

Test query:

- Create a record for a new user account, with a name, a login name, and a password.

- List all projects that contain the keyword ‘‘jazz’’ and that are currently looking for funds, sorted in descending order by posting time.

- List all users who have given money for projects containing the tag or category ‘‘jazz’’ in the past,  
sorted by the total amount they have successfully pledged (meaning, money that was actually charged).

- List all users who have completed at least 3 projects, and where each of their projects received an average rating of 4 stars or higher from its sponsors.

- List all comments by users that are followed by user ‘‘BobInBrooklyn’’.

- Insert a new project for a particular user, with a name, description, and other needed info.

- Insert a pledge to sponsor a project, for a particular user, project, and amount.

- Write queries for the end of a funding campaign. E.g., you could use triggers to detect when a campaign is fully funded or time is up; if successfully funded, generate charges to sponsors’ credit cards.

V.