**Database project Phase-02 Submission**

**Project Idea: portfolio terrain – A dedicated social media platform for freelance and remote developers**



**Submitted by:**

**Usman Shahid (L1F22BSCS1057) &**

**Ali Hamza Sehole (L1F21BSCS0791)**

**Section: D-8**

**Submitted to:**

**Sir Afham Nazir**

**Database project idea:**

social media platform dedicated for freelance and remote workers – Portfolio Terrain

**project description:**

this project will be a dedicated social media platform for freelance and remote developers that will allow developers to showcase there projects, skills and endorsements to secure freelance work and clients all over the globe. The idea is to provide users with interactive UI that is easy to use for both buyer and seller. The unique thing in this project would be that it will allow buyers to independently select desired freelancers for their projects with right skill set and portfolio projects. There will not be any bidding system although there will be a feature for buyers that they can post their project or job with their requirements and freelancers can apply to those projects. There will be a commission fee of 1% on each project that freelancer would pay to platform in order to use the services of this platform. Freelancers and buyers both will be given the ability to follow accounts of other users in order to increase the number of users of platform. Real time message communication feature shall also be provided to both freelancers and buyers in order to contact with their relevant people. The platform itself will play the role of a middle man and will connect buyer with seller and ensure that the buyer pays the payment to the platform and after successful confirmation of project completion from client and freelancer, the platform will pay the amount to its related freelance user and deduct it’s commission from the total amount

**Overview of Entities required in this project**

1. **User\_Account:**

Stores login credentials for users (freelancers or remote workers or buyers/clients).

1. **Porfolio Projects:**

Users will post these projects to attract buyers.

1. **Likes:**

Clients can like a portfolio project.

1. **Reviews:**

After completion of a certain project, a client can review the work of a freelance/remote worker.

1. **Payments:**

Buyers would pay Users on the completion of tasks.

1. **Projects\_done:**

Users who have completed projects on the site would have the facility to showcase their previous work on their profile.

1. **User\_Profile:**

Stores information about the skills, profile picture, and about info of the user.

1. **Buyer\_Profile:**

Stores information about the payment details, profile picture, and about info of the buyer.

1. **Categories:**

Provides buyers with the list of categories of freelance and remote work.

1. **Messages:**

Users and buyers will have the facility to chat with each other regarding projects.

1. **Notifications:**

For providing users with notifications in regard of their work

1. **Contracts:**

For storing contract related information if any, in between freelancer and buyer

1. **Attachments:**

For storing the filenames that users upload either those files can be of freelancers or buyers

1. **Invoices:**

To keep track of payments and provide user and freelancer with proof of payment transfer

1. **Admin\_Account:**

For administrative functionalities

1. **Management\_Account:**

For management purposes, solely for approving posts of users, either freelancer or buyer

**Attributes and Relations of entities with each other**

**User\_Account:**

1. Attributes: user\_id (Primary Key), username, email, password, role (freelancer/buyer), last\_login
2. Relationships:
3. One-to-One with User\_Profile (user\_id as Foreign Key)
4. One-to-Many with Messages (sender\_user\_id and receiver\_user\_id as Foreign Keys)

**Porfolio Projects:**

1. Attributes: project\_id (Primary Key), user\_id (Foreign Key), title, description, creation\_date, visibility
2. Relationships:
3. Many-to-One with User\_Account (user\_id as Foreign Key)
4. One-to-Many with Likes (project\_id as Foreign Key)
5. One-to-Many with Reviews (project\_id as Foreign Key)

**Likes:**

1. Attributes: like\_id (Primary Key), project\_id (Foreign Key), user\_id (Foreign Key), like\_date
2. Relationships:
3. Many-to-One with Porfolio Projects (project\_id as Foreign Key)
4. Many-to-One with User\_Account (user\_id as Foreign Key)

**Reviews:**

1. Attributes: review\_id (Primary Key), project\_id (Foreign Key), reviewer\_user\_id (Foreign Key), review\_text, rating, review\_date
2. Relationships:
3. Many-to-One with Porfolio Projects (project\_id as Foreign Key)
4. Many-to-One with User\_Account (reviewer\_user\_id as Foreign Key)

**Payments:**

1. Attributes: payment\_id (Primary Key), project\_id (Foreign Key), buyer\_user\_id (Foreign Key), amount, payment\_date
2. Relationships:
3. Many-to-One with Porfolio Projects (project\_id as Foreign Key)
4. Many-to-One with User\_Account (buyer\_user\_id as Foreign Key)
5. Projects\_done:
6. Attributes: project\_done\_id (Primary Key), user\_id (Foreign Key), project\_id (Foreign Key)
7. Relationships:
8. Many-to-One with User\_Account (user\_id as Foreign Key)
9. Many-to-One with Porfolio Projects (project\_id as Foreign Key)

**User\_Profile:**

1. Attributes: user\_id (Primary Key), full\_name, bio, profile\_picture\_url, skills
2. Relationships:
3. One-to-One with User\_Account (user\_id as Foreign Key)

**Buyer\_Profile:**

1. Attributes: user\_id (Primary Key), payment\_details, company\_name, company\_logo\_url
2. Relationships:
3. One-to-One with User\_Account (user\_id as Foreign Key)
4. Categories:
5. Attributes: category\_id (Primary Key), category\_name
6. Relationships: None

**Messages:**

1. Attributes: message\_id (Primary Key), sender\_user\_id (Foreign Key), receiver\_user\_id (Foreign Key), message\_text, timestamp, is\_read
2. Relationships:
3. Many-to-One with User\_Account (sender\_user\_id and receiver\_user\_id as Foreign Keys)
4. Notifications:
5. Attributes: notification\_id (Primary Key), user\_id (Foreign Key), notification\_text, notification\_date
6. Relationships:
7. Many-to-One with User\_Account (user\_id as Foreign Key)

**Contracts:**

1. Attributes: contract\_id (Primary Key), project\_id (Foreign Key), freelancer\_user\_id (Foreign Key), buyer\_user\_id (Foreign Key), terms, start\_date, end\_date
2. Relationships:
3. Many-to-One with Porfolio Projects (project\_id as Foreign Key)
4. Many-to-One with User\_Account for freelancer (freelancer\_user\_id as Foreign Key)
5. Many-to-One with User\_Account for buyer (buyer\_user\_id as Foreign Key)
6. Attachments:
7. Attributes: attachment\_id (Primary Key), project\_id (Foreign Key), filename, file\_url
8. Relationships:
9. Many-to-One with Porfolio Projects (project\_id as Foreign Key)

**Invoices:**

1. Attributes: invoice\_id (Primary Key), project\_id (Foreign Key), buyer\_user\_id (Foreign Key), freelancer\_user\_id (Foreign Key), amount, issue\_date, due\_date
2. Relationships:
3. Many-to-One with Porfolio Projects (project\_id as Foreign Key)
4. Many-to-One with User\_Account for buyer (buyer\_user\_id as Foreign Key)
5. Many-to-One with User\_Account for freelancer (freelancer\_user\_id as Foreign Key)

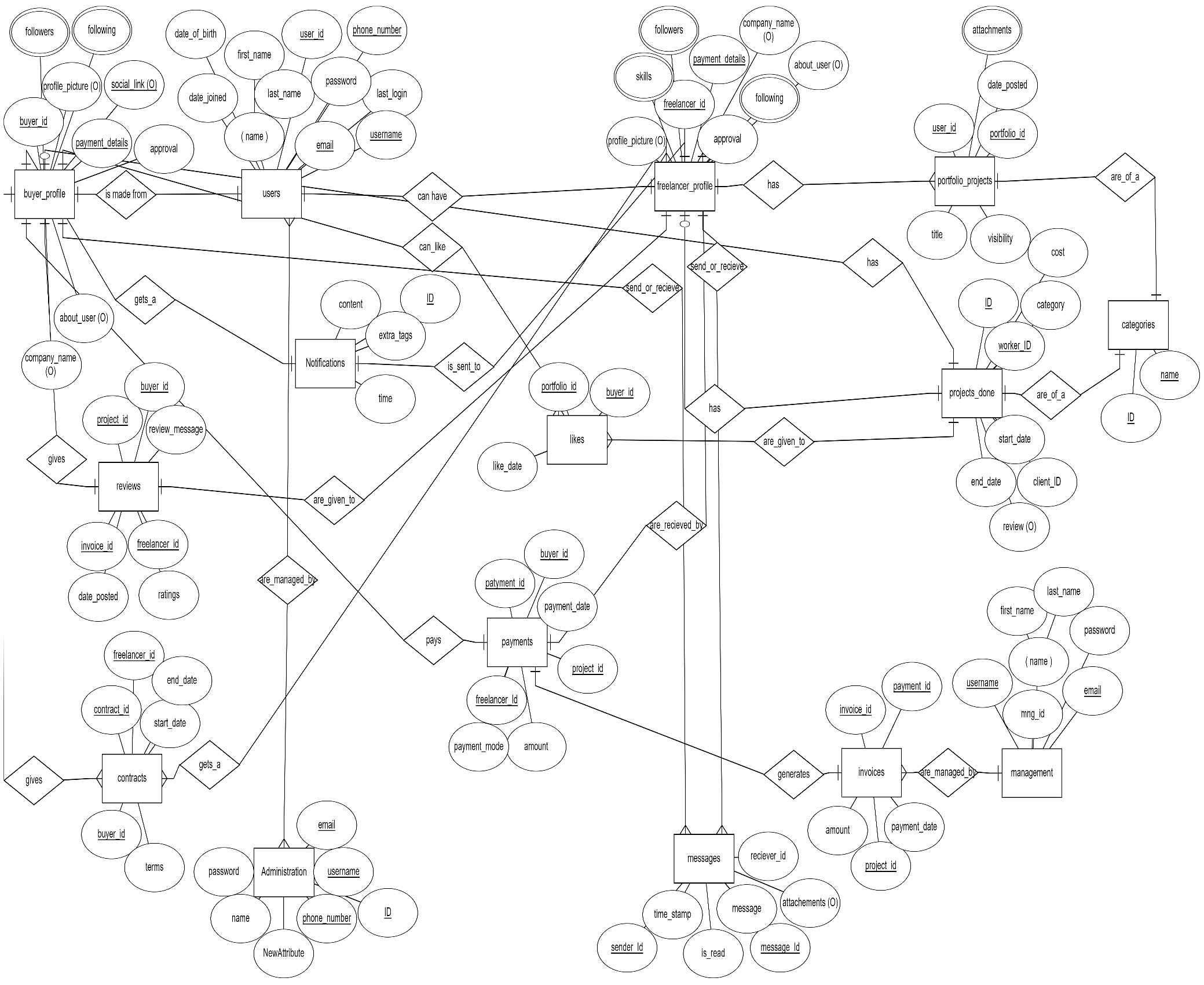
**Admin\_Account:**

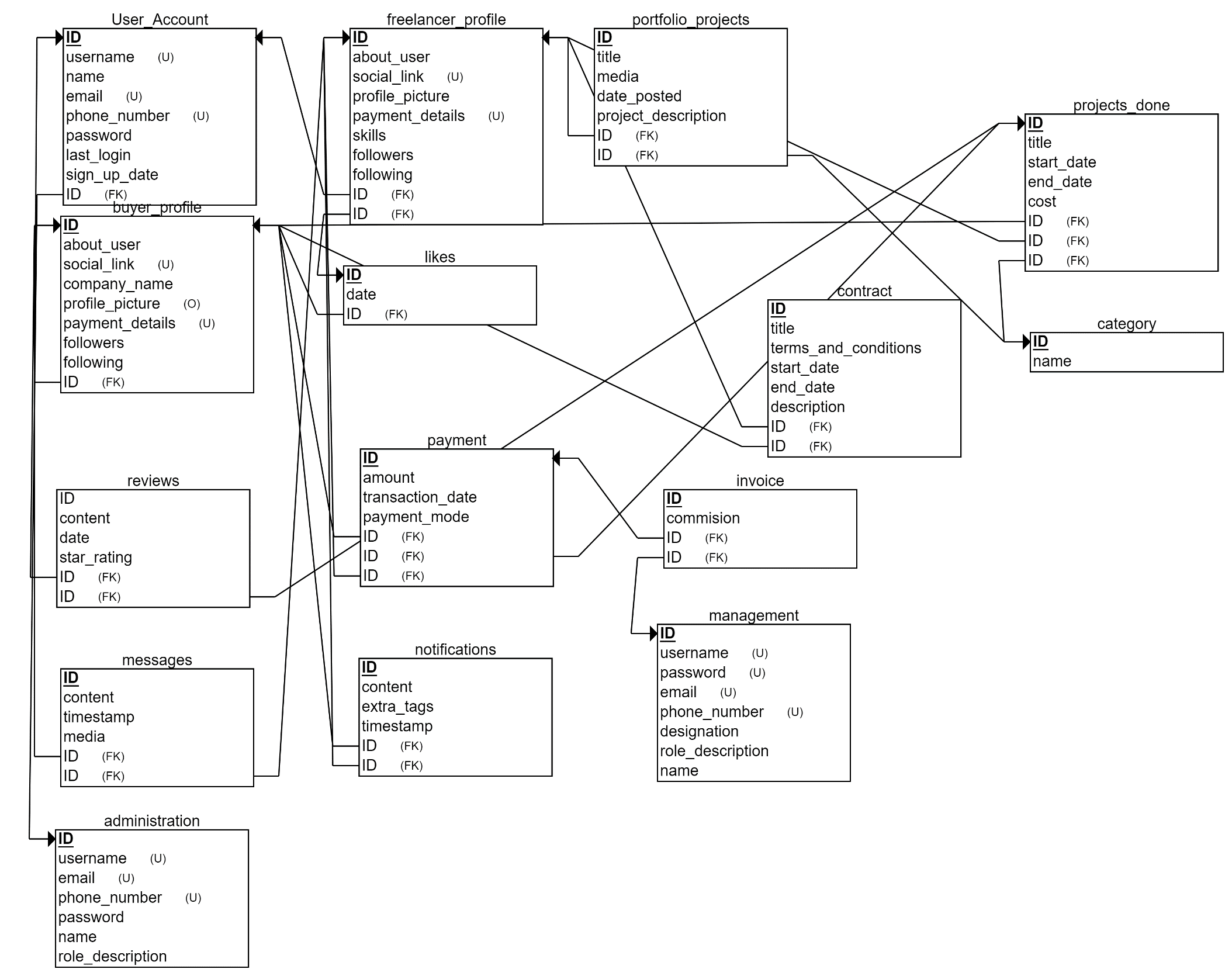
1. Attributes: admin\_id (Primary Key), username, email, password
2. Relationships: None

**Management\_Account:**

1. Attributes: management\_id (Primary Key), username, email, password
2. Relationships: None

**ERD Diagram And Relational Schema:**

****

****