CODEFURY

CONTACTS & NETWORKING APPLICATION

DONE BY

TEAM – ODE2CODE

CONTENT

1- INTRODUCTION

Contact-pool is a web-based application that allows users to add and manage contacts. It supports unlimited users and provides several features for socializing. It has two types of users: System administrators and application users. There are following modules which are present along with entities which are provided so that it helps in establishing a successful working of the given project.

2- MOTIVATION

The main aim of doing this is to help socializing with peers through the help of our website. Multiple users can be added into our database and the following accounts will help in establishing a relation with other users. Networking involves connecting with the people that you know, or can get to know, for help and advice. By starting with your own contacts and then asking them about their contacts, and so on, you are likely to find someone who can help you to find out more about a particular area of interest.

3- REQUIREMENTS & TOOLS

Operating system: 64 bit Windows/Ubuntu
Front end: Html, CSS, Bootstrap
Backend: MySQL database
Programming language: Java, JavaScript, HTML, CSS, JSP Browser: Google Chrome/Firefox

4- PROJECT DESCRIPTION:

The system supports two types of users

1. System administrators
   a)Can monitor activities of the user, they do not have access to the users and contacts database information.
   b)Can Disable Users, if the user is blocked by more than 3 Contacts.
   c) Can Delete Users, if user is inactive from a long time

2.Application users

Can create a profile in the system and add their contacts to the same. They can have friends and perform various activities.

5- TERMINOLOGIES

1. System administrator: The administrator of the application
2. Users: Those who have registered to the system and can login and perform certain tasks.
3. Contacts: These are contacts added by the user to the system which will reside in the contacts list of that user only
   a. A contact for a user could also be another registered user of the system
4. Friends: A(registered) User can search for other users and send friend requests. If the request is accepted, they become friends

. Friends are(registered) Users of the system.

6- APPLICATION FEATURES

1. Users must register to the system
2. Registered users can add and maintain their contacts in the system
    a. This information is private to that user, no other user or administrator can view the contact list information
3. Registered users can search the system for other registered users based on name/location etc. a. The system will show the basic information of users in the search results like name and profile image.
    b. A friend request can be sent to anyone in the system unless they have blocked the user.
    c. c)If the request is accepted, then the user can view the full information of the newly added friend.

System administrators cannot view or modify the users database; however, they can view statistical data like number of users, most active users etc. They can disable or delete users. The system should store all the information in the database.

Data points:

1. User Authentication
2. Users
3. Contacts
4. Friend List
5. Friend Request
6. Blocked Users
7. Deactivated Users
8. Disabled Users

7- MODULES:

Following are the basic modules of our system:

1. Front-end:
    a. View - Provides an interactive UI for users to input data and show respective output.
    b. Controller - Responsible for managing and routing data to necessary layers i.e. from view to service and vise-versa.
2. Back-end:
    a. Service - This layer includes the business logic needed for all the functionalities.
    b. DAO - Consists of an interface which enables connection to the database and processes the data as required.
    c. Database - A relational database(SQL) used to store all the necessary data which is managed using the DAO layer.

Entities:

1. User:
    a. User_id
    b. Full Name,
    c. Email
    d. Phone No
    e. Gender
    f. Date of Birth
    g. Address
    h. City
    i. State
    j. Country
    k. Company
    l. Profile image(upload)
    m. Username

n. Password

2. FriendRequest:
    a. User_id
    b. Sender_id
    c. Message

Functionalities:

1. User:

User module enables users to perform all the functionalities regarding all the users present on the system. This includes:

a. Onboarding (Register / Login)

i. Users registering to the system will trigger a DAO layer functionality which connects to the Users table to create a new User.

ii. Service layer ensures that those who are disabled are not allowed to register again.

iii. The Login function of DAO layer will verify the credentials provided by the user (username, password) with the UserAuth entries.

iv. The validation of all the user input is done in the controller and view layer.

b. Search

i. DAO function call returns a list of all the users present on the system.

ii. Service layer is responsible to filter all the users which have blocked the target user.

iii. Users will be able to send requests/ block any user using the output of controller and view layer.

c. Block/Unblock

i. Users will be able to see the list of users he/she has

blocked on the view.

ii. Block/Unblock functionality is implemented in service and dao layers to interact with the controller.

2. Contacts:

Main functionality of our system is storing and managing contacts. Users will be able to perform all the operations on his/her own contacts. This includes:

a. CRUD operations

i. View layer provides an interface to edit/manage contacts stored by the user.

ii. Service layer checks if the user is already present on the system and enables the user to send friend requests.

3. Friends:

Friends includes all the functionalities regarding interaction of all the users which are present on the system. Which include:

a. View/Manage friends

i. Friend list is displayed on the view layer which enables users to perform functions like remove/block friend.

ii. Information stored in the database excluding confidential information is retrieved in DAO to be visible to friends.

b. Send/Respond requests

i. Pending friend requests are displayed on the view layer along with the message from the sender.

ii. Service layer checks if the request is already pending for a certain user, thus preventing the user from creating another request.

4. Admin:

a. Login

i. Admin details are stored in an XML file which is loaded when the application is initialized. These are used to verify the input provided by the admin.

b. List Users

i. Admin can view the users present on the system and their basic information like user_id, location.

c. Disable/Deactivate Users

i. The DAO layer fetches all the users which are eligible to be disabled (users blocked by multiple users) or deactivated (users who are not active for a long time).

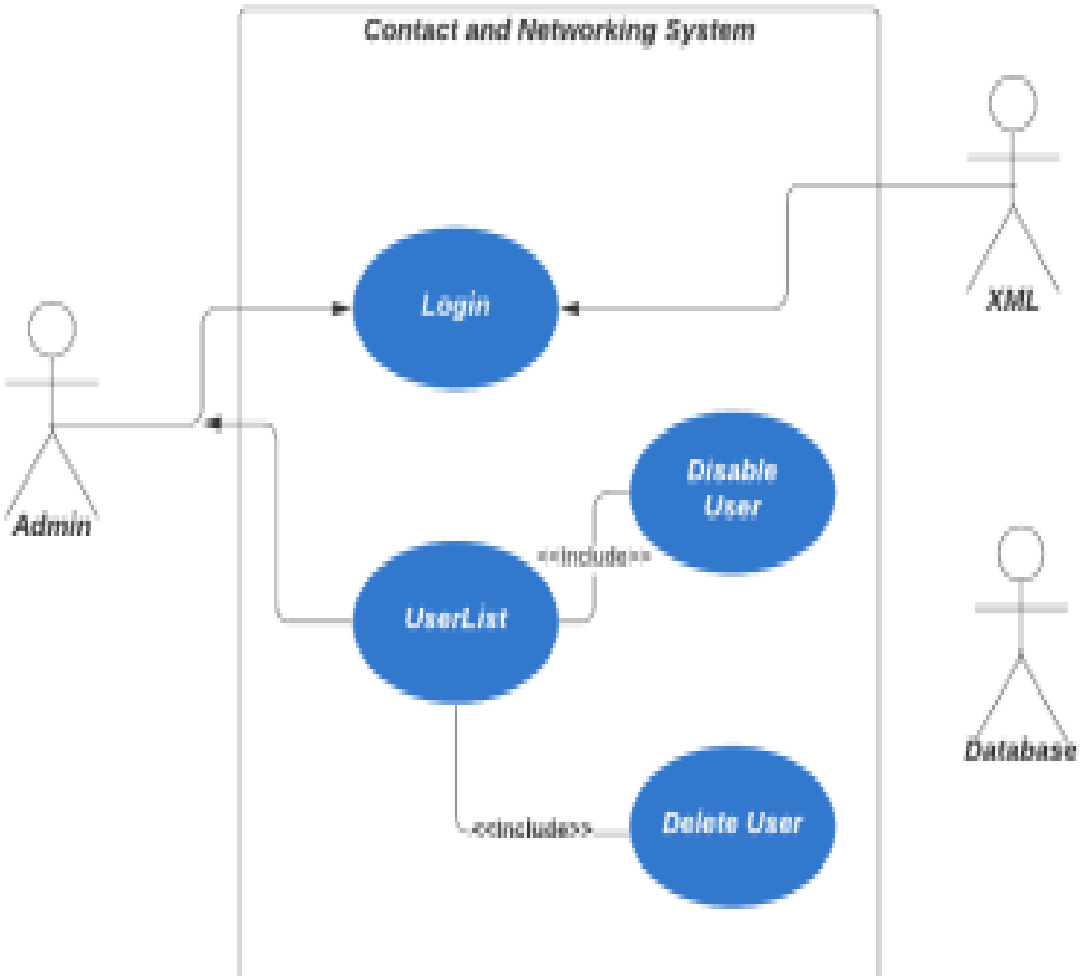
ii. The list is displayed on the view to enable the admin to deactivate/disable a certain user.

d. User Activity

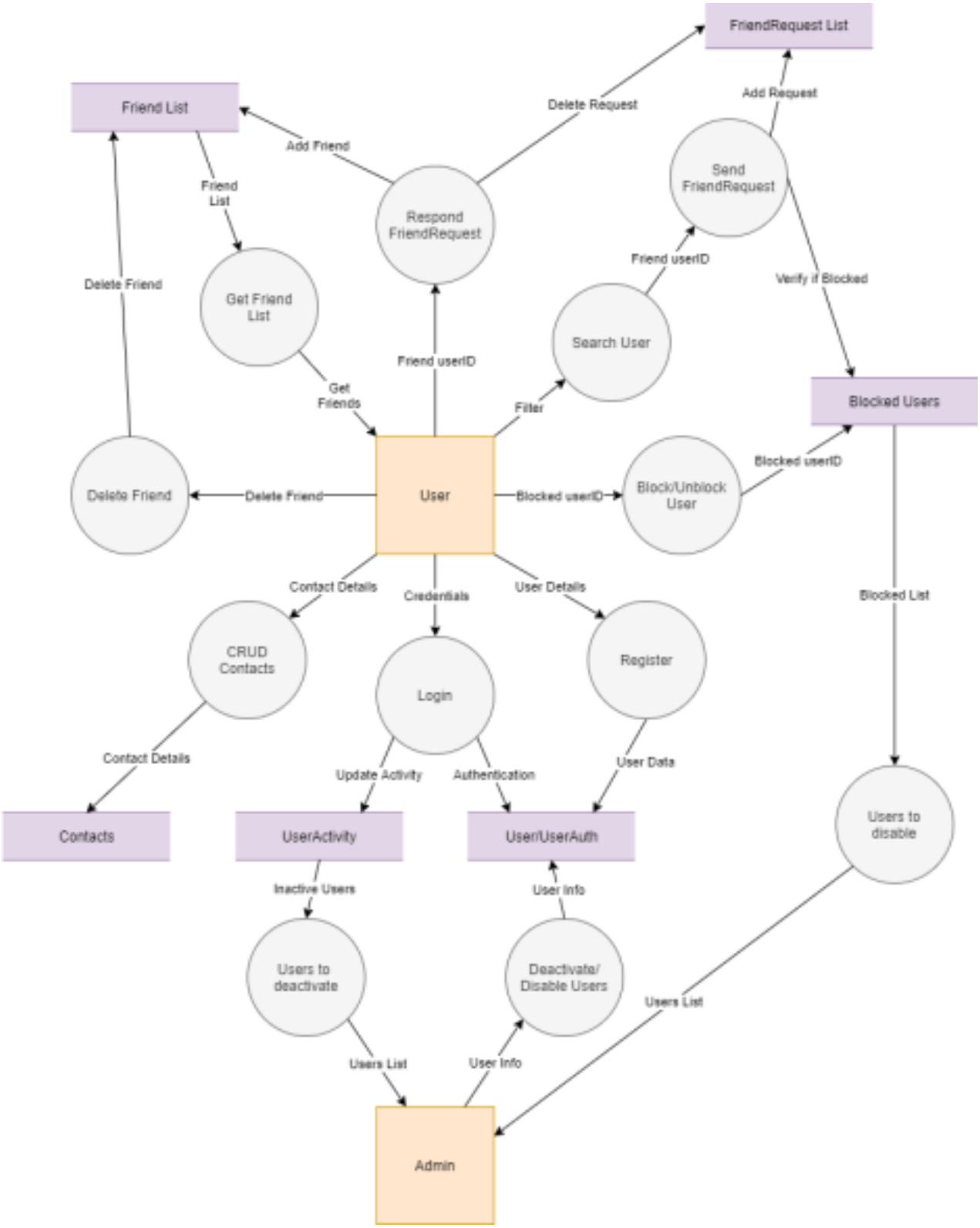
i. This is a server functionality which is triggered whenever a user is logged in and makes a DAO functionality call to update user activity in the database.

8- DIAGRAMS

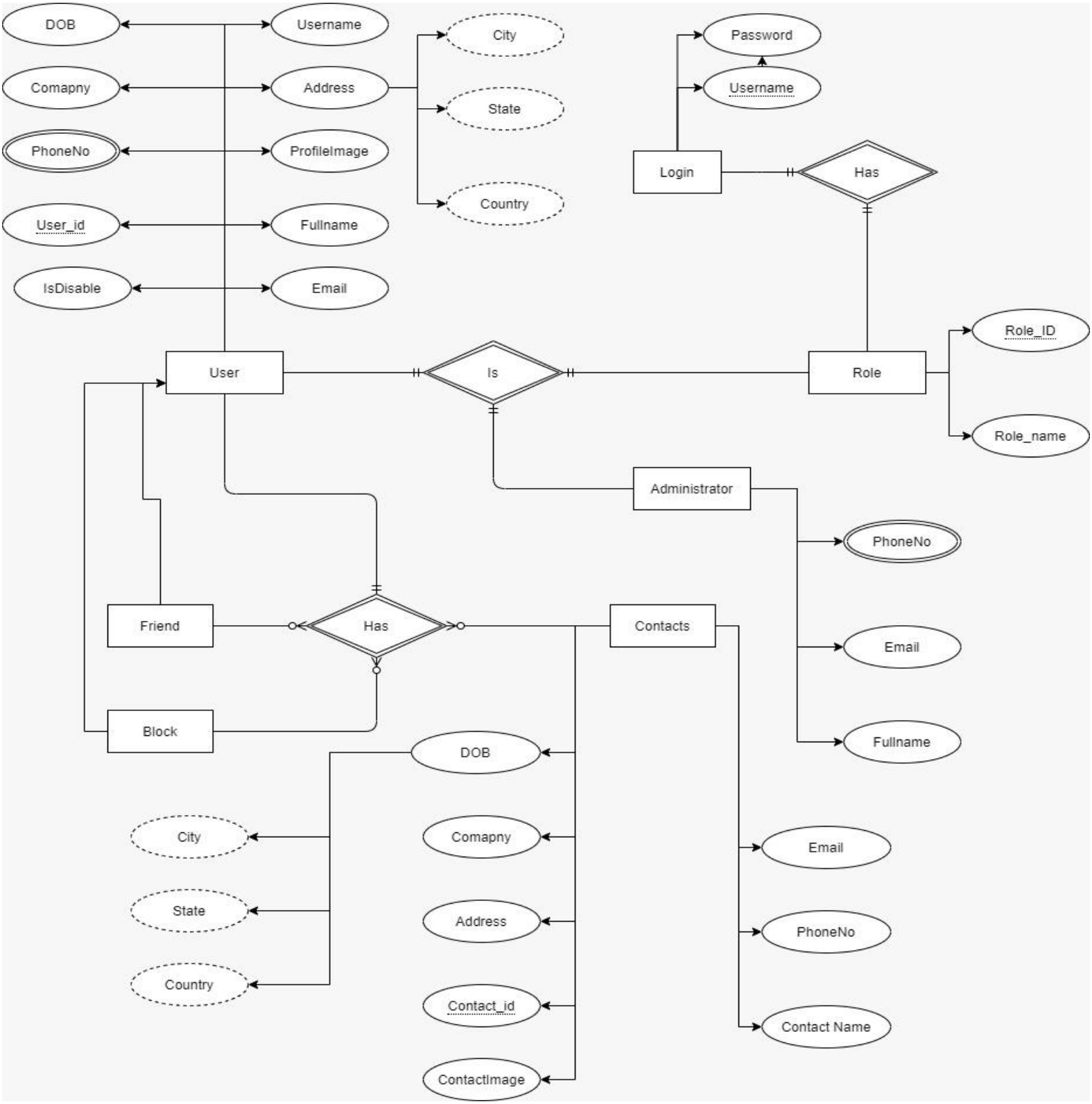
1. Admin Use Case:



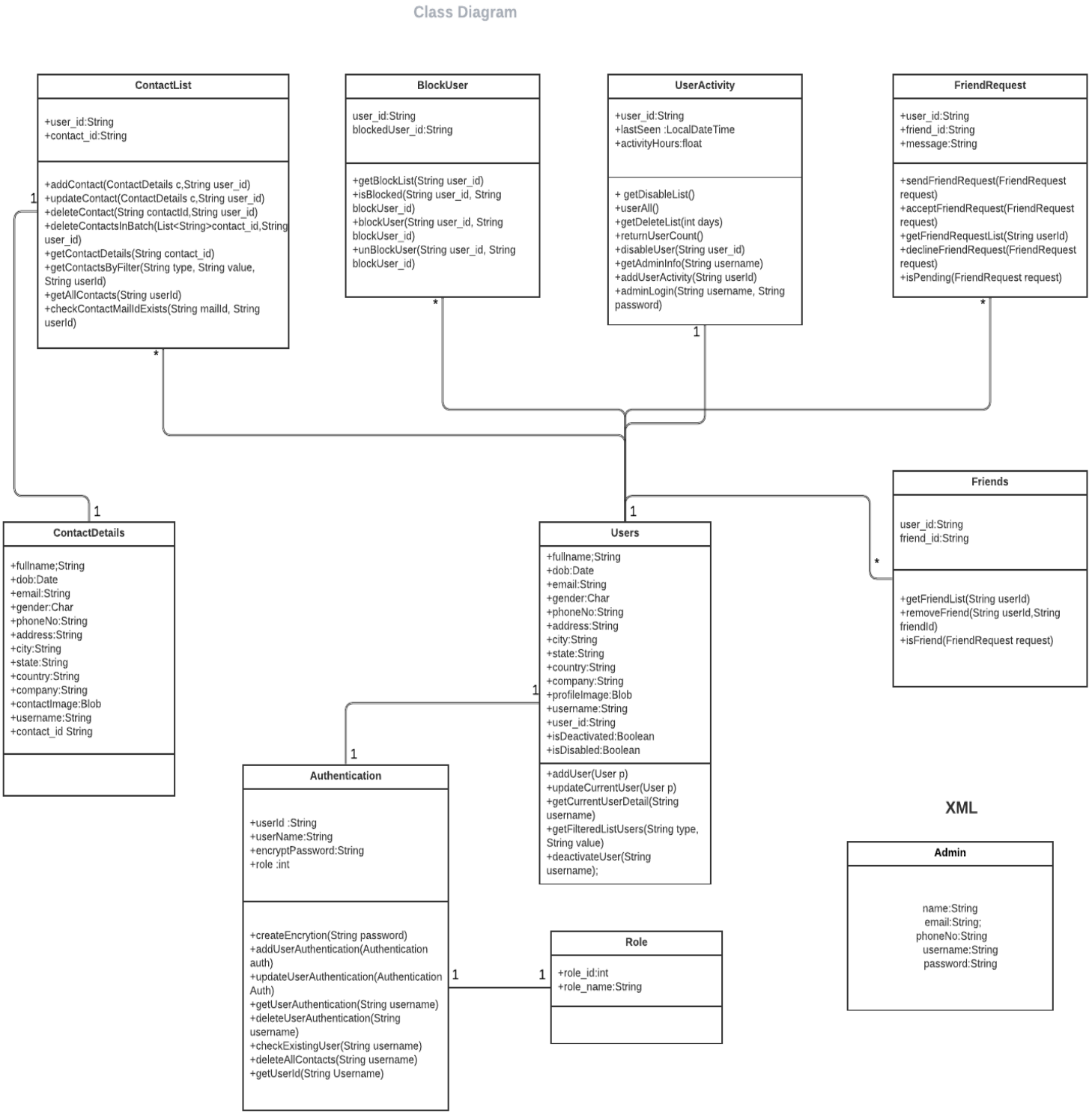
2. Data Flow Diagram



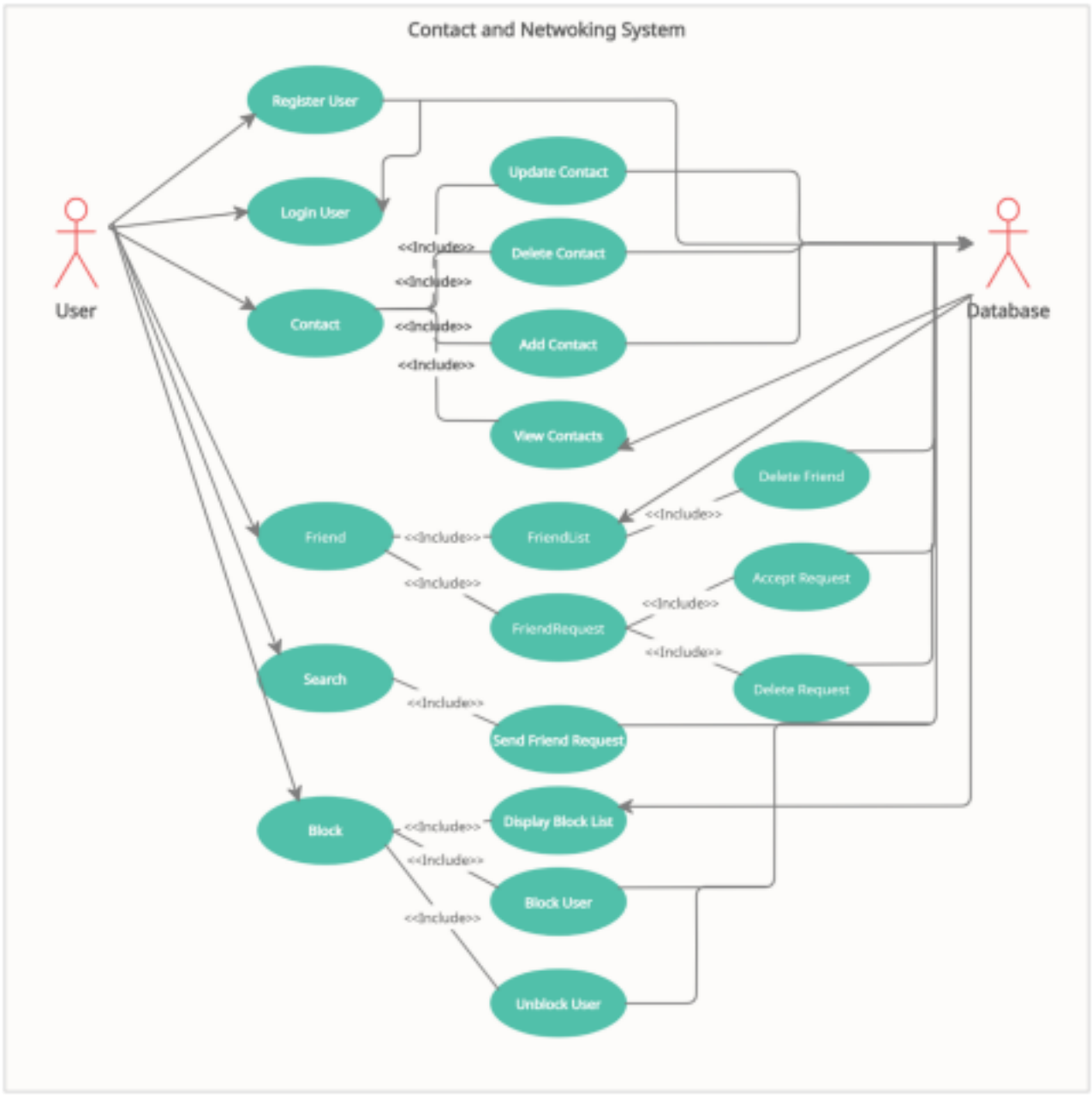
3. ER Diagram


DOB
Username
City
Password
Comapny
Address
State
Username
PhoneNo
ProfileImage
Login
Has
Country
User_id
Fullname
IsDisable
Email
Role_ID
User
Is
Role
Role_name
Administrator
PhoneNo
Friend
Has
Contacts
Email
Block
Fullname
DOB
City
Comapny
Email
State
Address
PhoneNo
Country
Contact_id
Contact Name
ContactImage

4. Class Diagram



5. User Use Case Diagram



9- CONCLUSION

Thus we have proposed a website where users can interact with each other and the admin can have specific features which are provided through frontend and backend and also we learned how to work as a team and contribute towards the project individually as well as collectively.