Design Document

CodeFury: Contacts and Networking Application

WFS-5

Noice-Toit-Smort

Rajat Puri

Susmit Hongekar

Ritika Sethi

Swati Singh

Shradha Lohumi

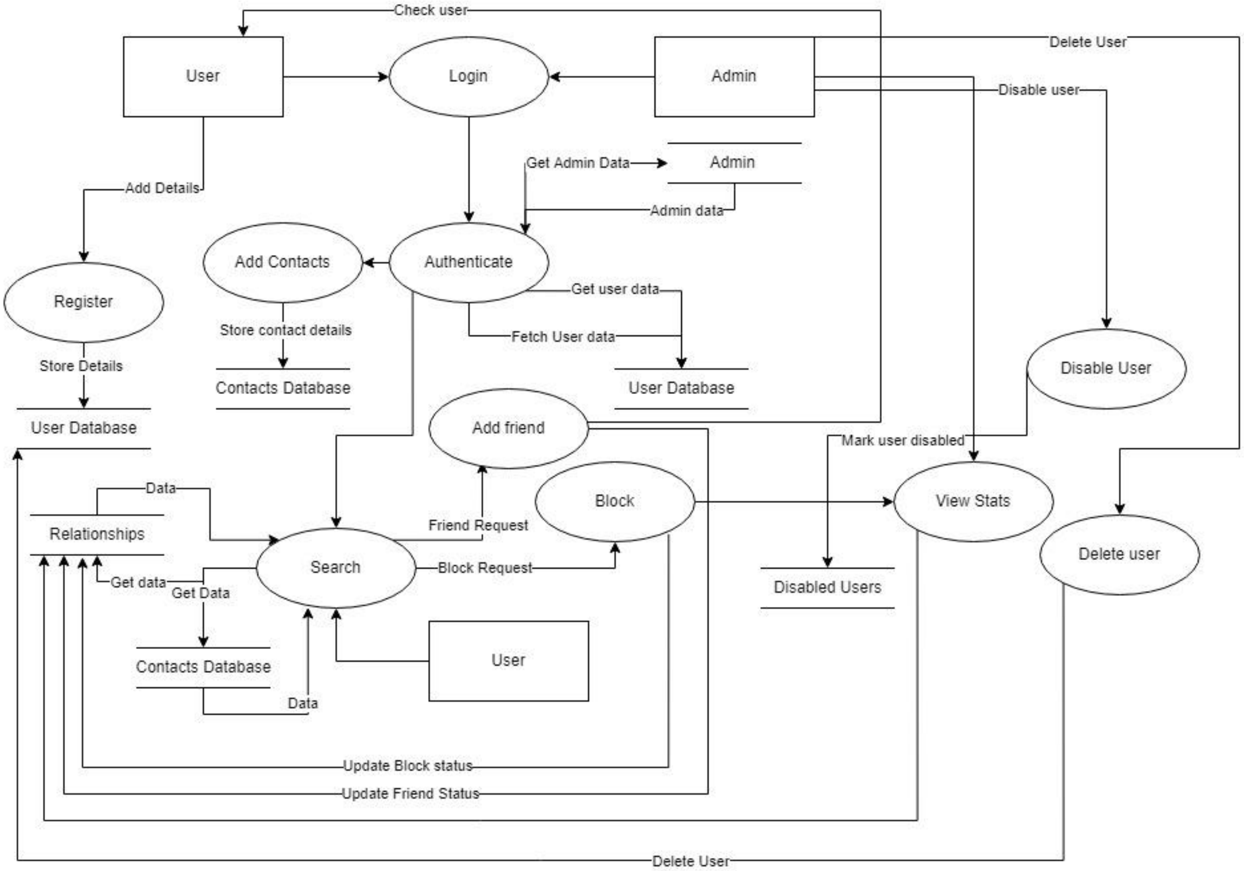
Rajatendu Dey

Prakhar Agrawal

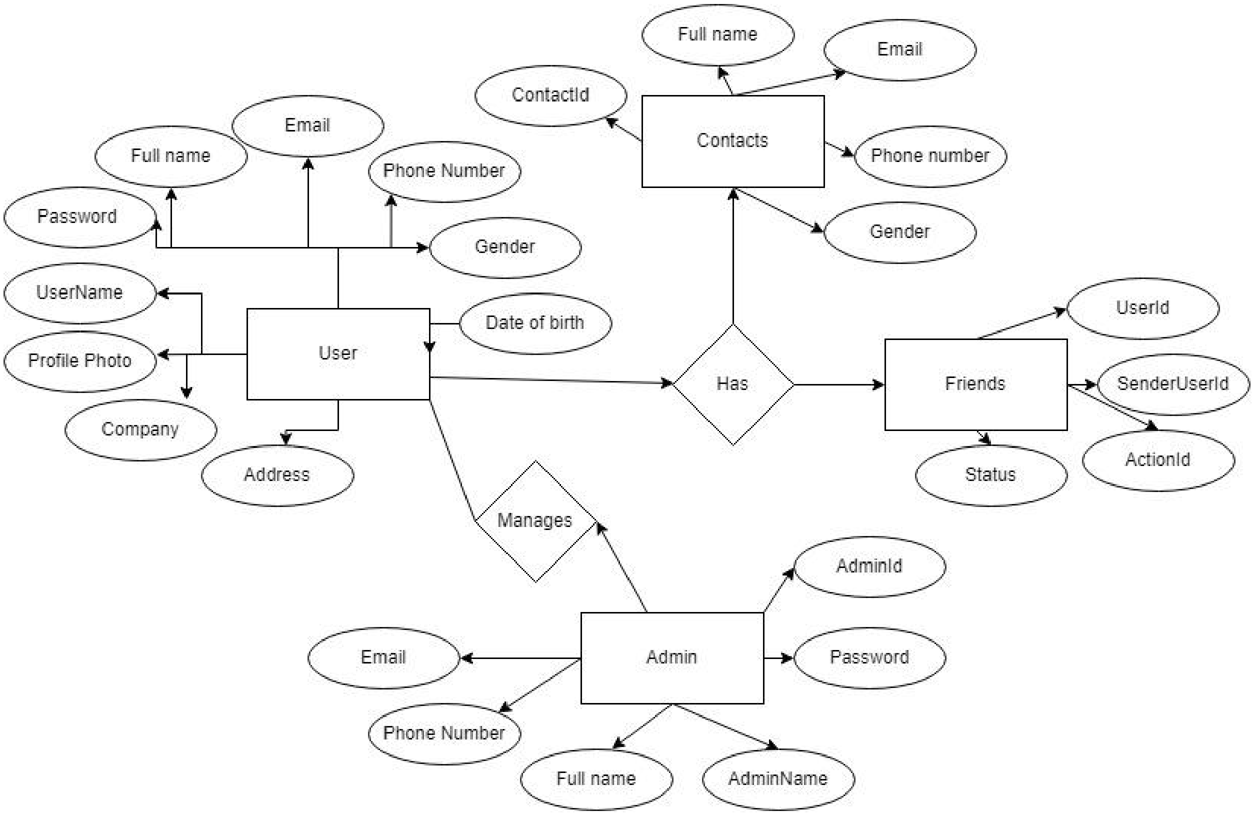
Ruchika Gosavi

Shreyoshi Ganguly

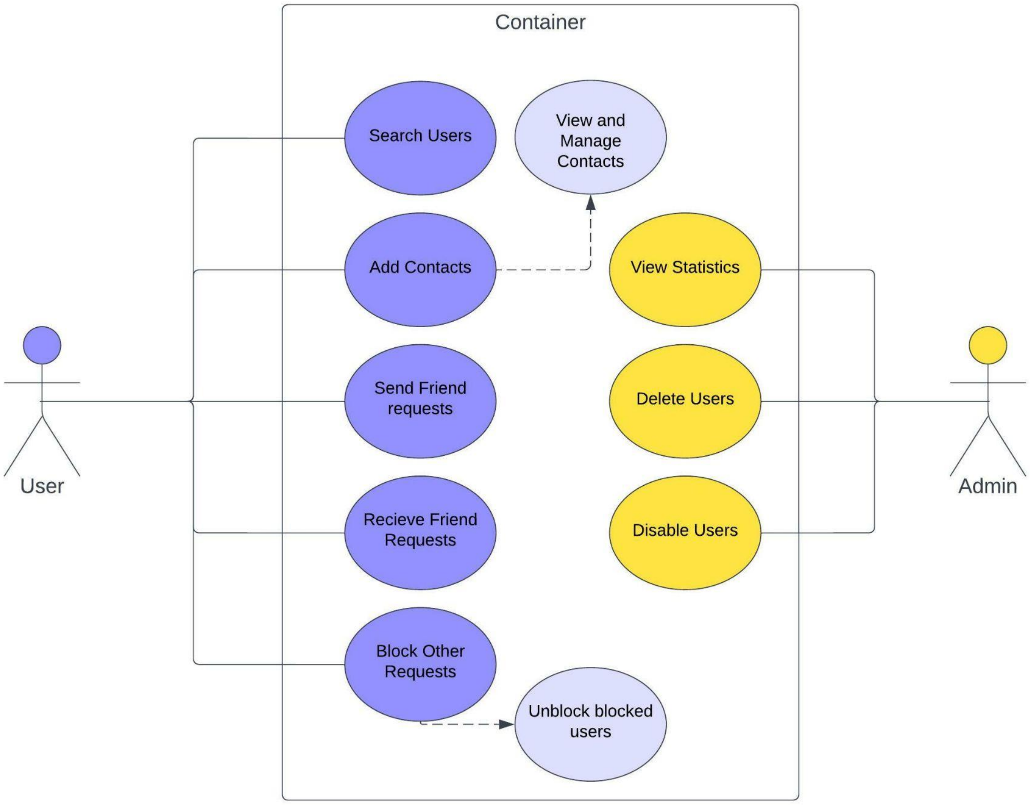
DataFlow Diagram



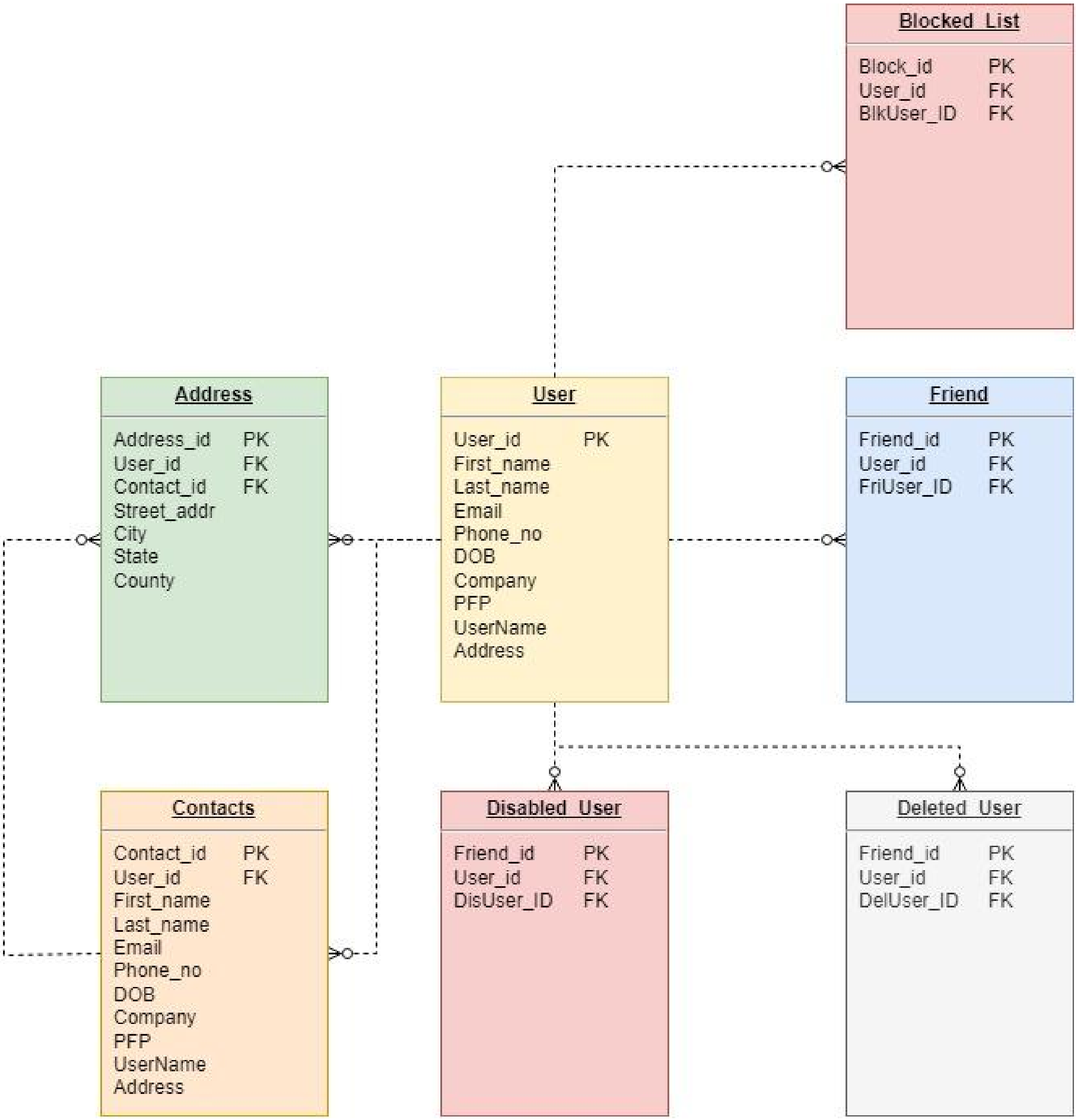
Enitity Relationship Diagram



Use Case Diagram



Database Design



Blocked_List

Block_id	PK
User_id	FK
BlkUser_ID	FK

Address

Address_id	PK
User_id	FK
Contact_id	FK
Street_addr	
City	
State	
County	

User

User_id	PK
First_name	
Last_name	
Email	
Phone_no	
DOB	
Company	
PFP	
UserName	
Address	

Friend

Friend_id	PK
User_id	FK
FriUser_ID	FK

Contacts

Contact_id	PK
User_id	FK
First_name	
Last_name	
Email	
Phone_no	
DOB	
Company	
PFP	
UserName	
Address	

Disabled_User

Friend_id	PK
User_id	FK
DisUser_ID	FK

Deleted_User

Friend_id	PK
User_id	FK
DelUser_ID	FK

Application Design

Model Classes:

The application uses four model classes as entity classes:

- **Person**
    - This class defines fields that are common to both the user and the contact user classes.
    - The attributes of the classes are:
        - Full name
        - Email
        - Phone Number
        - Gender
        - Date of birth
        - Address
        - Company
        - Profile Picture
- User
    - This class extends the <Person> class.
    - The added attributes that make the Person class User class are UserId, Username and password
    - Attributes of user class are:
        - UserId
        - Username
        - Password
- Contact
    - This class extends the <Person> class.
    - The contacts do not require any additional fields of their own.

- Admin
    - This class contains all the attributes of the administrator.
    - An object of this class is created by parsing an existing XML file that contains all the admin details.
    - The attributes of this class are:
        - AdminId
        - Fullname
        - Email
        - Phone number
        - AdminUsername
        - Password

Data Access Objects (DAO)

This application implements 3 DAO interfaces. These DAO interfaces have their own functionalities to perform and are responsible to talk to the Database

- **Admin DAO**
  - This DAO is responsible for handling all the Admin operations.
  - Admin DAO has the following methods:
    - disableUser
    - deleteUser
    - showSummary
    - viewUsersToDisable
    - viewUsersToDelete
- **UserDAO**
  - This DAO is responsible for handling all the Admin operations.
  - User DAO has the following methods:
    - boolean addContact(int id, Contact contact);
    - editContact
    - addFriend
    - viewContacts
    - viewFriends
    - viewBlockedUsers
    - searchUser
    - blockUser
    - unblockUser
    - sendRequest
    - declineRequest
    - viewFriendRequests
- **Verification**
  - This DAO contains methods for registration and login of Users and also the login of Admin

    - The methods are:
        - verifyAdmin
        - verifyUser
        - registerUser

These DAO interfaces are respectively implemented by:

- UserDaoImpl
- AdminDaoImpl
- VerificationService

Servlets

Servelts are used as a communication channel between Java Clsses and the JSP/HTML views. Servlets also act as a channel of communication between the JSP pages themselves.

The Servlets used for the implementation of this project are:

- AdminServlet
  - This servelt contains the methods to handle requests to and from the AdminDaoImpl and manages the pages responsible for the admin view.
- UserServlet
  - This servlet contains the methods to handle request to and from the UserDaoImpl and manages the view pages for a user.
- NavigationServlet
  - This servlet is designed to handle the home page and the pages directly connected to it, them being the registration page for users, user login page and the admin login page. It also handles the requests to the VerificationService to handle the Registration and Login.

Utility Classes

These classes are used for basic utility functions. They are:

- ConnectionUtility
  - This is used to Connect to the Database
- AdminParser
  - This is a parser to get the Admin credentials from the existing XML file.
- NameComparator
  - Used to sort Users in order of their name.

View Pages

The following are the view pages used in this implementation. These are used to display the UI and perform some validation tasks:

- Index.html
- adminDelete.jsp
- adminDisable.jsp
- adminHome.jsp
- adminLogin.jsp
- userHome.jsp
- userInfo.jsp
- userLogin.jsp
- userResult.jsp
- userSignUp.jsp
- viewBlockedList.jsp
- viewFriendList.jsp
- viewContactList.jsp
- viewFriendRequest.jsp
- addNewContact.jsp

Additional files:

The following are some additional files that are needed for the implementation to work as desired:

- Admin.xml : This file holds the credentials for the admin user.
- Default Profile picture : This is used as a placeholder image for users that do not upload a profile picture. This is also used as a placeholder for contacts with no contact image.