# Contact Management System


**Team: Wizards at Work**

**Advance Data Base**

**603-C**

**Dr. Reza Sadeghi**

**Thursday, 24 March 2022**

# Outline

# Team Description

**Prudhvi sai akhil Thumu-** Graduate Computer Science student at Sacred Heart University. Completed my Under graduation from Gitam University in India and, I acquired experience in IT industry as a cloud DevOps Engineer. My role in this project is to be Monitoring the team activities as a team lead, involving in the development of GUI, API and Database system for Contact management System and Creating Different Tables which are essential for the Contact Management System (CMS).

**Sandeep Yepuri -** Graduate Computer Science student at Sacred Heart University. Completed my Under graduation from KLU University in India and, I acquired experience in IT industry as a full stack developer. My role in this project to develop UI interface for contact management system and will share some work in SQL.

**Chetan Sai Tallamudi-** Graduate Computer Science student at Sacred Heart University. Completed my Under graduation from St Joseph in India. My role in this project to develop API in order to connect the GUI with backend database and will do some part of work in database.

# Team Description(Cont)

**Nikhilender Reddy Baddam-** Graduate Computer Science student at Sacred Heart University. Completed my Under graduation from VNR VJIET institute of Technology in India. I acquired experience in IT industry as a Python developer. My role in this project to develop the database  and will share work in developing the GUI for Contact Management System.

# System Description

Contact Management System (CMS) stores different types of information such as Telephone number, Mailing Address, Contact name, Phone number, Fax Number, Home number and Address. The CMS will store the user data in distinct SQL tables with different user types.

- Admin user or root user can add new user to the CMS by adding the username and password to the database which cannot done by the normal user.
- Admin user can remove user from the CMS by removing username, password and any other related data.
- Every user can be able to add the contact information like first name, sur name, phone number, workplace number, workplace address, home address, zip code, fax number, email address, gender and age.
- Every user can be able to remove the contact information

# System Description(Cont.)

- CMS GUI shows a beautiful welcome page
- CMS GUI can show all available options and functions to the end user
- CMS can display the information using various search options where users are able to search with contact number to get other details like name and mail address
- CMS graphical user interface (GUI) is very user friendly where it can show the warnings where the user is trying to put the contact information which is exist in tables.
- CMS can be able to provide the exit functionality on the GUI

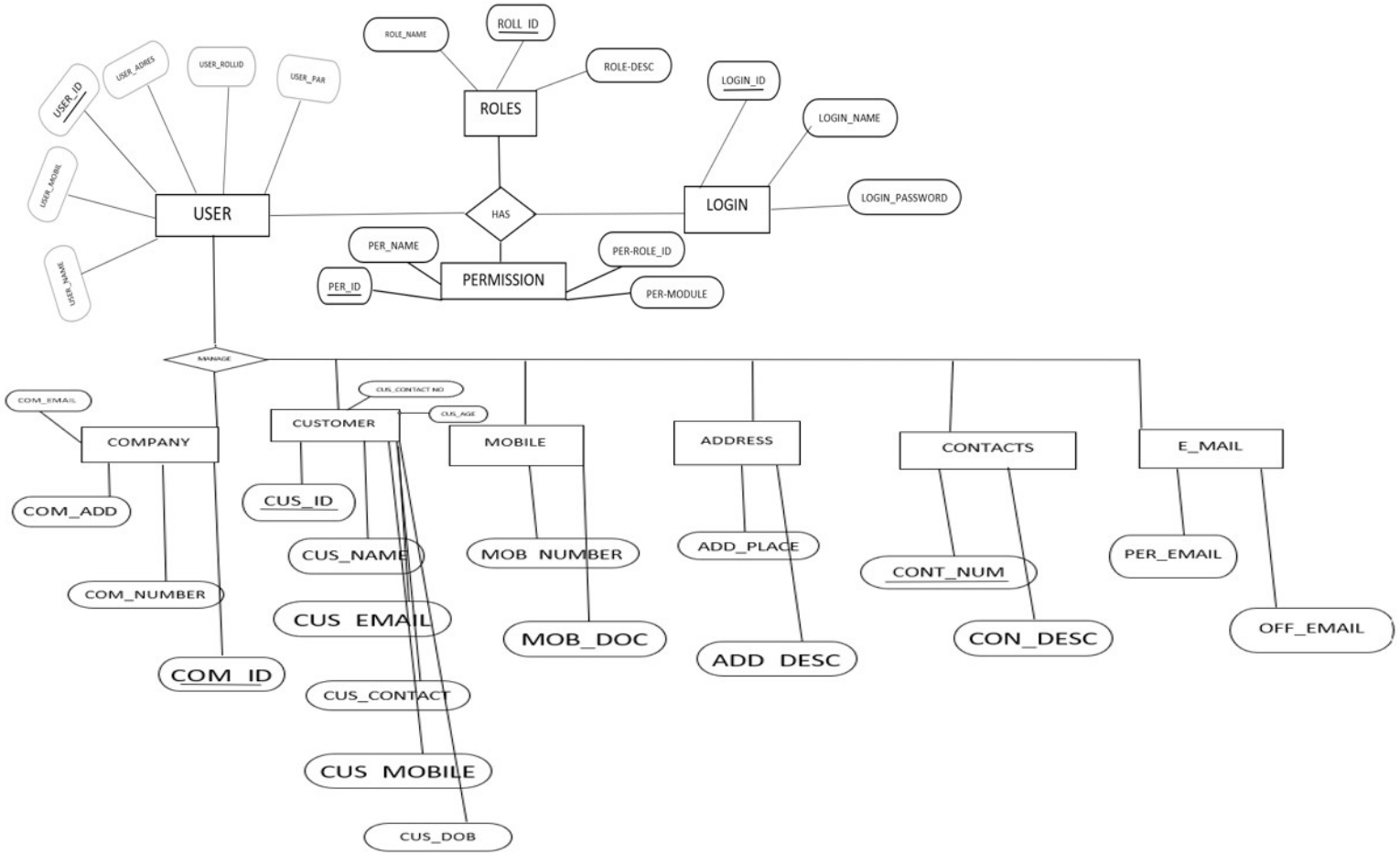
# Entity Relationship Model

Entity Relation diagram  represents the basic design upon which database is built. The main entities of contact management system are the User, Login, Permission, Company, Customer, Address etc. Each entity will have the primary key and some entities will contain the foreign keys. In our entity relationship diagram, we have also showed different functionalities like partial participation, total participation, multivalued attribute, composite attribute etc.

- Partial Participation - User table
- Total Participation – Customer table
- Multivalued attribute – emails and contacts (single user may have multiple emails and multiple contacts)
- Composite attribute – Name (F_name, M_name, L_name)
- Derived attribute – age of the user can be derived from the date of birth of the user.
- Weak entities – Email table and contact’s table

# Entity Relationship Model(Cont.)

Entity Relationship model diagram



# Enhanced Entity Relationship Model

Enhanced Entity Diagram helps us create and maintain detailed databases through high level models and they are developed based on the ER diagram. All the tables shown below are to implement the database for Contact Management System (CMS). Data which is going to store inside the CMS database is purely end user information who are working in different organizations and having different roles. For CMS to maintain the relationship between among the entities in the tables used one to many relationships and one to one relationship.

Enhanced Entity Relation diagram for CMS consists of 10 tables which are stated below

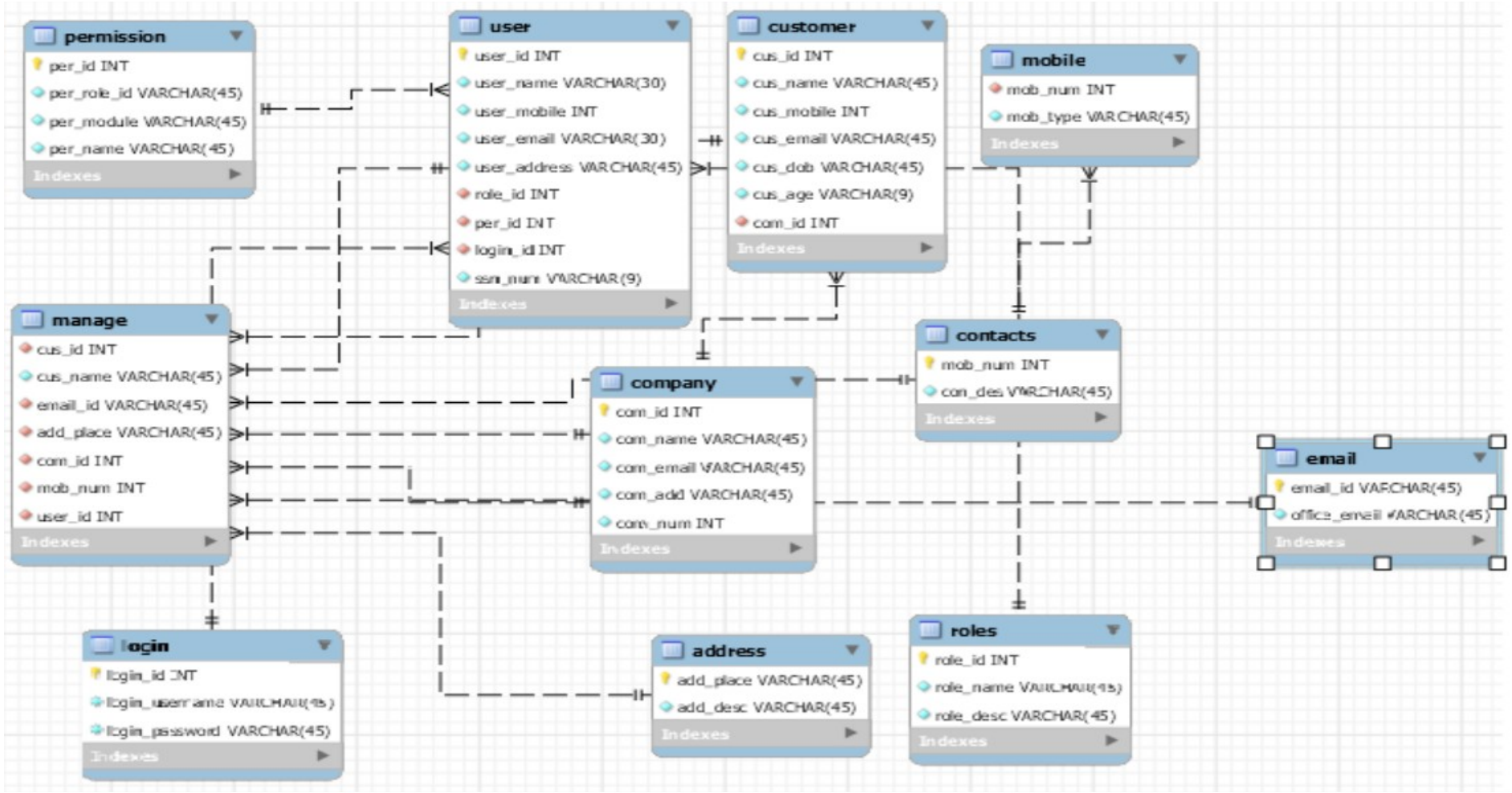
- Login (login_id, login_role_id, login_username, user_password)
- User (user_id, user_name, user_mobile, user_address, user_email, user_rolid, user_perid)
- Roles (role_id, role_name, role_desc)
- Address (add_place, add_desc, add_zip)

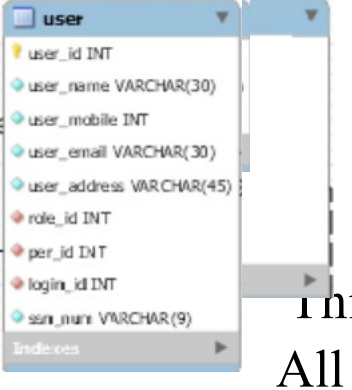
# Enhanced Entity Relationship Model(cont)

- Mobile (mobile_des, mob_num)
- Permission (per_id, per_role_id, per_name, per_module)
- Contacts (con_num, con_des)
- Company (company_id, company_name, company_add, company_num, company_mail, company_lev)
- Email (per_email,office_mail)
- Customer (cus_id, cus_name, cus_mobile, cus_email, cus_add, cus_comid, company_com_id, cus_dob,cus_age)

# Enhanced Entity Relationship Model(cont)

Enhanced Entity Relationship model diagram

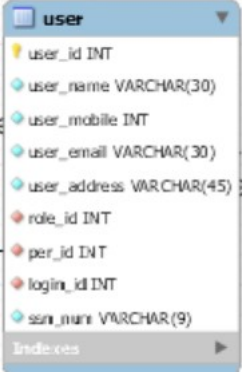





# SQL Database Development

This Database is designed as per the EER model diagram which is shown above. All the tables and entities are created with the help SQL queries Containing Primary Keys, foreign keys and Null keys.

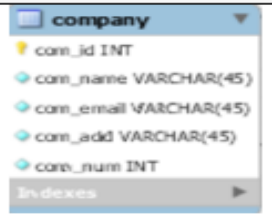
In all most all the tables Primary keys (Pkey) and Foreign Keys (Fkey) are defined which are having a unique value for each column.



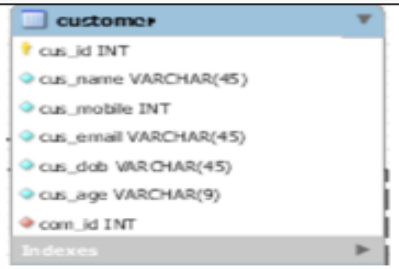
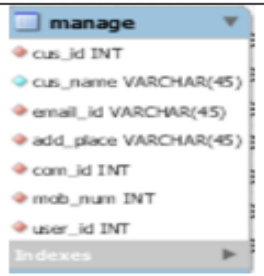

# SQL Database Development(Cont.)

| Table Name | Query | EER Model for Table | Description | Pkey | Fkey |
|---|---|---|---|---|---|
| user | CREATE TABLE `user` (<br>user_id int NOT NULL,<br>user_name varchar(30) NOT NULL,<br>user_mobile INT NOT NULL,<br>user_email varchar(30) NOT NULL,<br>user_address varchar(45) NOT NULL ,<br>role_id INT NOT NULL,<br>per_id INT NOT NULL,<br>login_id int NOT NULL,<br>ssn_num varchar(9) NOT NULL,<br>PRIMARY KEY (`user_id`),<br>FOREIGN KEY (role_id) REFERENCES ROLES(role_id),<br>FOREIGN KEY (login_id) REFERENCES login(login_id),<br>FOREIGN KEY (per_id) REFERENCES PERMISSION(per_id)); |  | The purpose of User table is to manage the customer's data | Yes | Yes |

# SQL Database Development(Cont.)

| | | | | | |
|---|---|---|---|---|---|
| company | CREATE TABLE `company` (<br>com_id int NOT NULL,<br>com_name varchar(45) NOT NULL,<br>com_email varchar(45) NOT NULL,<br>com_add varchar(45) NOT NULL,<br>com_num int NOT NULL,<br>PRIMARY KEY (`com_id`)); |  | The purpose of company table is to add or manipulate the customer's company data. | Yes | No |
| permission | CREATE TABLE `permission` (<br>per_id INT NOT NULL,<br>per_role_id varchar(45) NOT NULL,<br>per_module varchar(45) NOT NULL,<br>per_name varchar(45) NOT NULL,<br>PRIMARY KEY (`per_id`)); |  | The purpose of Permissions table to assign different permissions to end user. | Yes | No |
| login | CREATE TABLE `login` (<br>login_id int NOT NULL,<br>login_username varchar(45) NOT NULL,<br>login_password varchar(45) NOT NULL,<br>PRIMARY KEY (`login_id`)); |  | The purpose of Login table is to store the login credentials like username and password etc | Yes | No |
| roles | CREATE TABLE `roles` (<br>role_id INT NOT NULL,<br>role_name varchar(45) NOT NULL ,<br>role_desc varchar(45) NOT NULL ,<br>PRIMARY KEY (`role_id`)); |  | The purpose of roles table is to assign the different roles to user like admin, developer etc. | Yes | No |

# SQL Database Development(Cont.)

| | | | | | |
|---|---|---|---|---|---|
| customer | CREATE TABLE `customer` (<br>cus_id int NOT NULL,<br>cus_name varchar(45) NOT NULL,<br>cus_mobile INT NOT NULL,<br>cus_email varchar(45) NOT NULL,<br>cus_dob varchar(45) NOT NULL,<br>cus_age varchar(9) NOT NULL,<br>com_id INT NOT NULL,<br>PRIMARY KEY (`cus_id`),<br>FOREIGN KEY (com_id)<br>REFERENCES company(com_id)); |  | The Purpose of customer table is to store the customer data. | Yes | Yes |
| manage | CREATE TABLE `manage` (<br>cus_id int NOT NULL,<br>cus_name varchar(45) NOT NULL,<br>email_id varchar(45) NOT NULL,<br>add_place varchar(45) NOT NULL,<br>com_id INT NOT NULL,<br>mob_num int NOT NULL,<br>user_id INT NOT NULL,<br>FOREIGN KEY (cus_id)<br>REFERENCES customer(cus_id),<br>FOREIGN KEY (email_id)<br>REFERENCES email(email_id),<br>FOREIGN KEY (add_place)<br>REFERENCES address(add_place),<br>FOREIGN KEY (com_id)<br>REFERENCES company(com_id),<br>FOREIGN KEY (com_id)<br>REFERENCES company(com_id),<br>FOREIGN KEY (mob_num)<br>REFERENCES contacts(mob_num), |  | The purpose of manage table is for user to manage the customer data. | No | Yes |

| Table Name | Query | Description |
|---|---|---|
| Login | insert into login(login_id,login_username,login_password) values('32354','sacreadheart','SHU@345'),<br>('90282','kites','Pru@345'),<br>('67533','university','Ak@6323'),<br>('63276','posst','mypass'),<br>('821629','kiytes','uni@1P'),<br>('91222','leosa','post%$'),<br>('78632','lionds','Peudjj@sj'),<br>('92329','school','Liqjs'),<br>('83728','hshes','Insjs'),<br>('656757','yrsyg','$^uyt'); | Here we are inserting the data into the login table. In this table we have information like login ids, username, and password. |
| Permission | Insert into permission(per_id, per_role_id, per_module, per_name)<br>values (1,2,'ALL','Access to all modules/pages'),(2,3,'Limited','Access to only some pages'),(3,4,'No Permission','No Access to any modules'),<br>(4,2,'hello','Access to all modules/pages'),<br>(5,3,'Limited','Access to only some pages'),<br>(7,8,'No Permission','No Access to any modules'),<br>(9,2,'ALL','Access to all modules/pages'),<br>(10,2,'Limited','Access to only some pages'),<br>(11,4,'No Permission','No Access to any modules'); | Here we are inserting the data into the permission table, and we have data like permission id, permission role, permission name. There will be different permissions for different users. |

# Manipulating the data

In this section, we manipulated the data which exist in the tables with the help of ALTER and UPDATE commands.

Using ALTER command, we created a new column in the user table and updating the new column with some data with the help of UPDATE command.

**Table Name**: User

| user_id | user_name | user_mobile | user_email | user_address | role_id | per_id | login_id | ssn_num |
|---|---|---|---|---|---|---|---|---|
| 1 | Sandeep | 2 | sandeep214@gmail.com | 186, lincoln | 3 | 1 | 91222 | 564776 |
| 2 | Akhil | 2 | akhndeep214@gmail.com | 186, lincoln | 3 | 1 | 91222 | 123 |
| 3 | NIkhil | 2 | nikhil@gmail.com | 186, lincoln | 3 | 1 | 91222 | 564776 |
| 4 | Chethan | 4 | chethan14@gmail.com | 186, lincoln | 4 | 1 | 91222 | 564776 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Query with Alter:

Here we are adding the new column to the user table.

**Alter table user**

**add New_SSN varchar(10);**

After Manipulation of data with Alter:

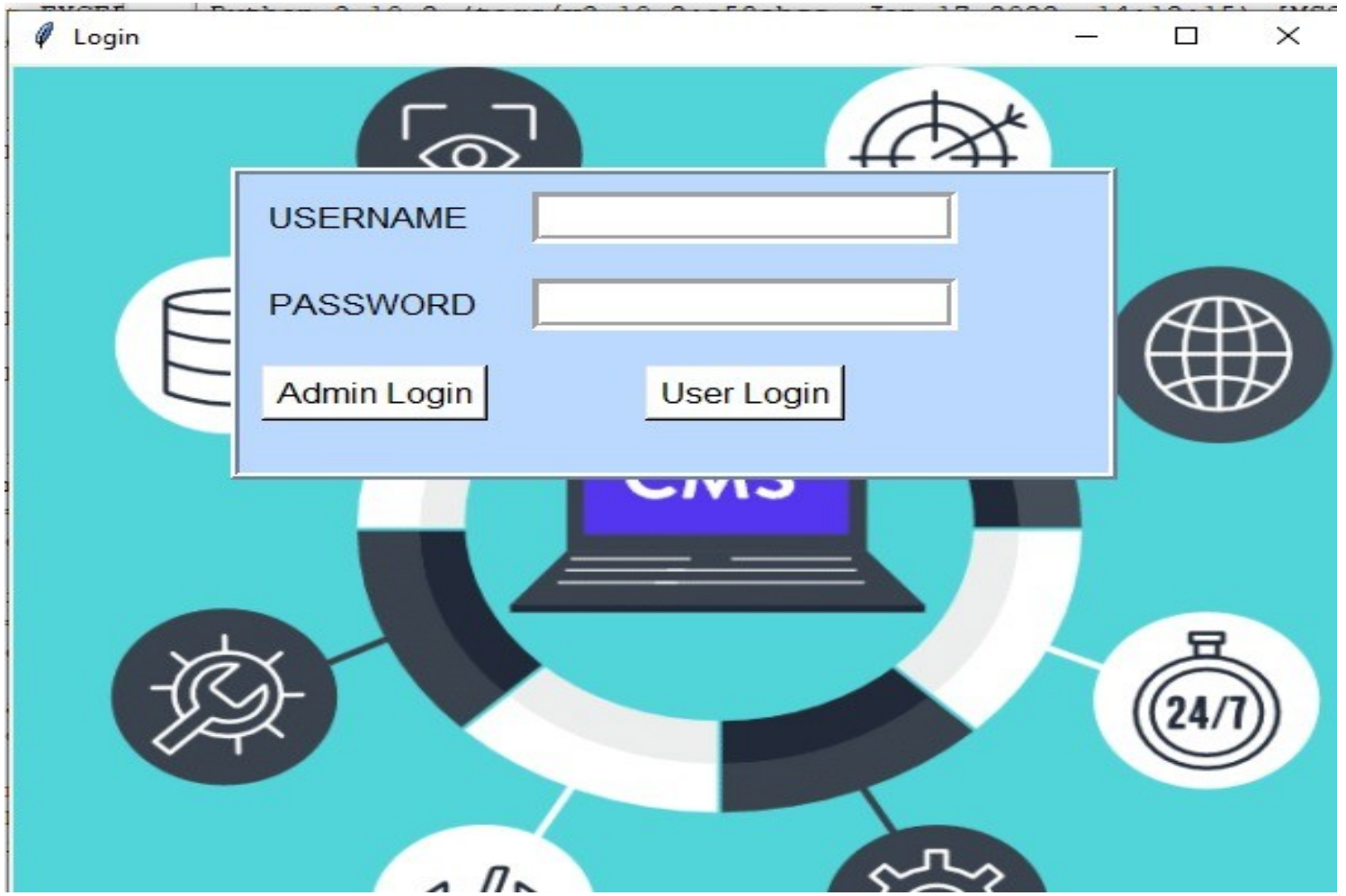
# Manipulating the data(Cont.)

After Manipulation of data with Alter:

| user_id | user_name | user_mobile | user_email | user_address | role_id | per_id | login_id | ssn_num | New_SSN |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Sandeep | 2 | sandeep214@gmail.com | 186, lincoln | 3 | 1 | 91222 | 564776 | NULL |
| 2 | Akhil | 2 | akhndeep214@gmail.com | 186, lincoln | 3 | 1 | 91222 | 123 | NULL |
| 3 | NIkhil | 2 | nikhil@gmail.com | 186, lincoln | 3 | 1 | 91222 | 564776 | NULL |
| 4 | Chethan | 4 | chethan14@gmail.com | 186, lincoln | 4 | 1 | 91222 | 564776 | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

# Graphical User Interface

Login Page:

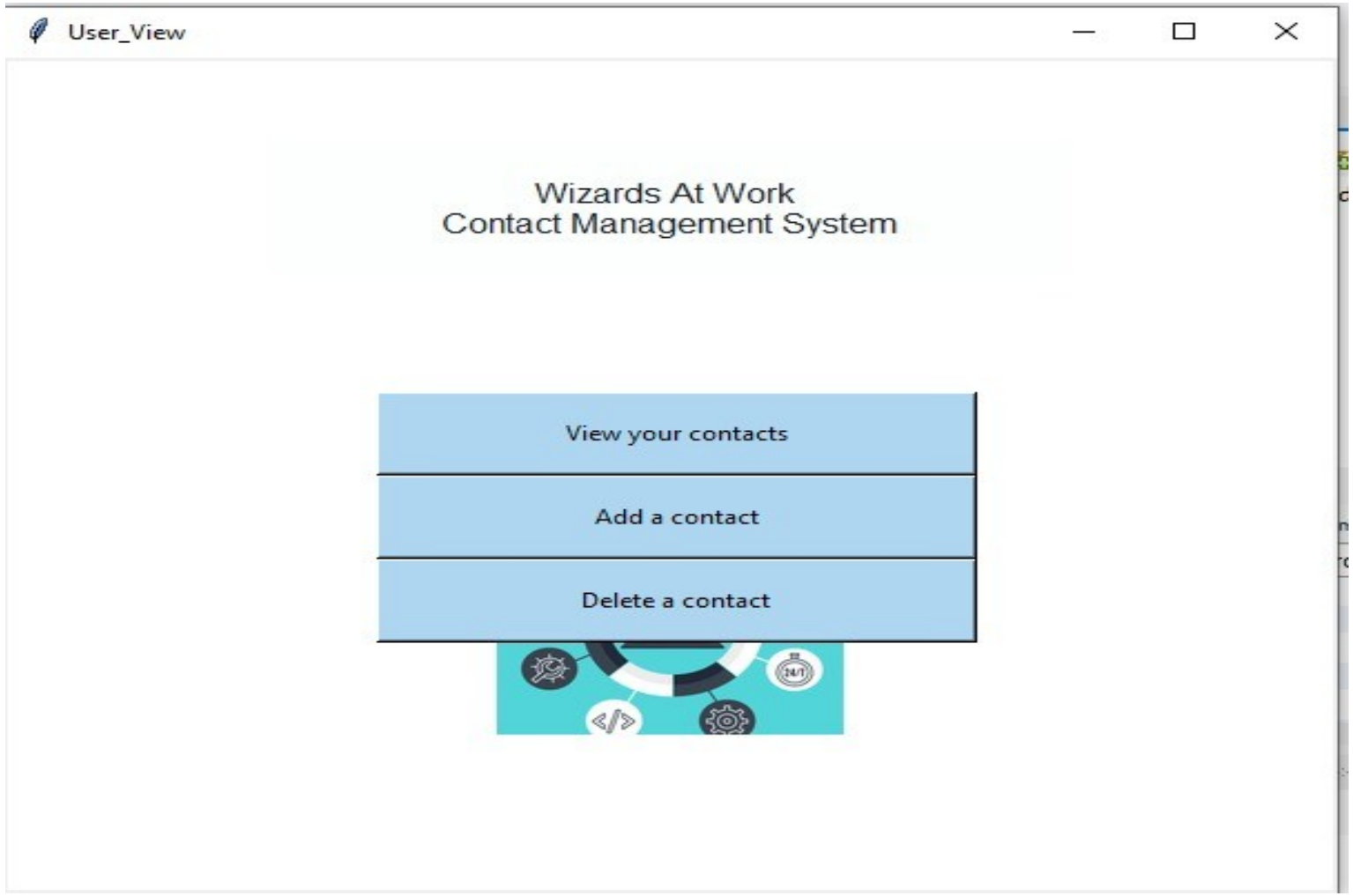


# Graphical User Interface(Cont)

Admin View:

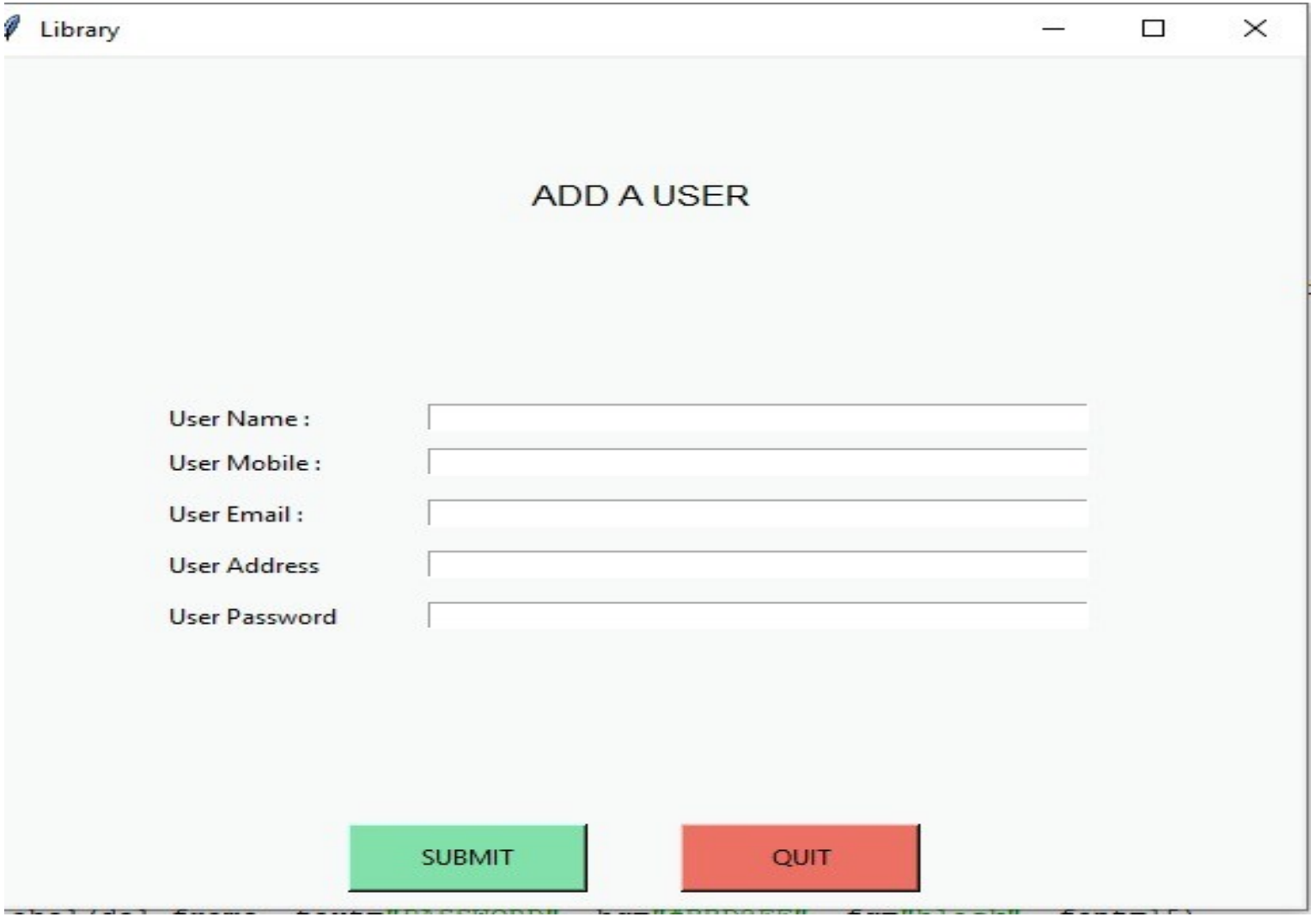
# Graphical User Interface(Cont.)

User View:



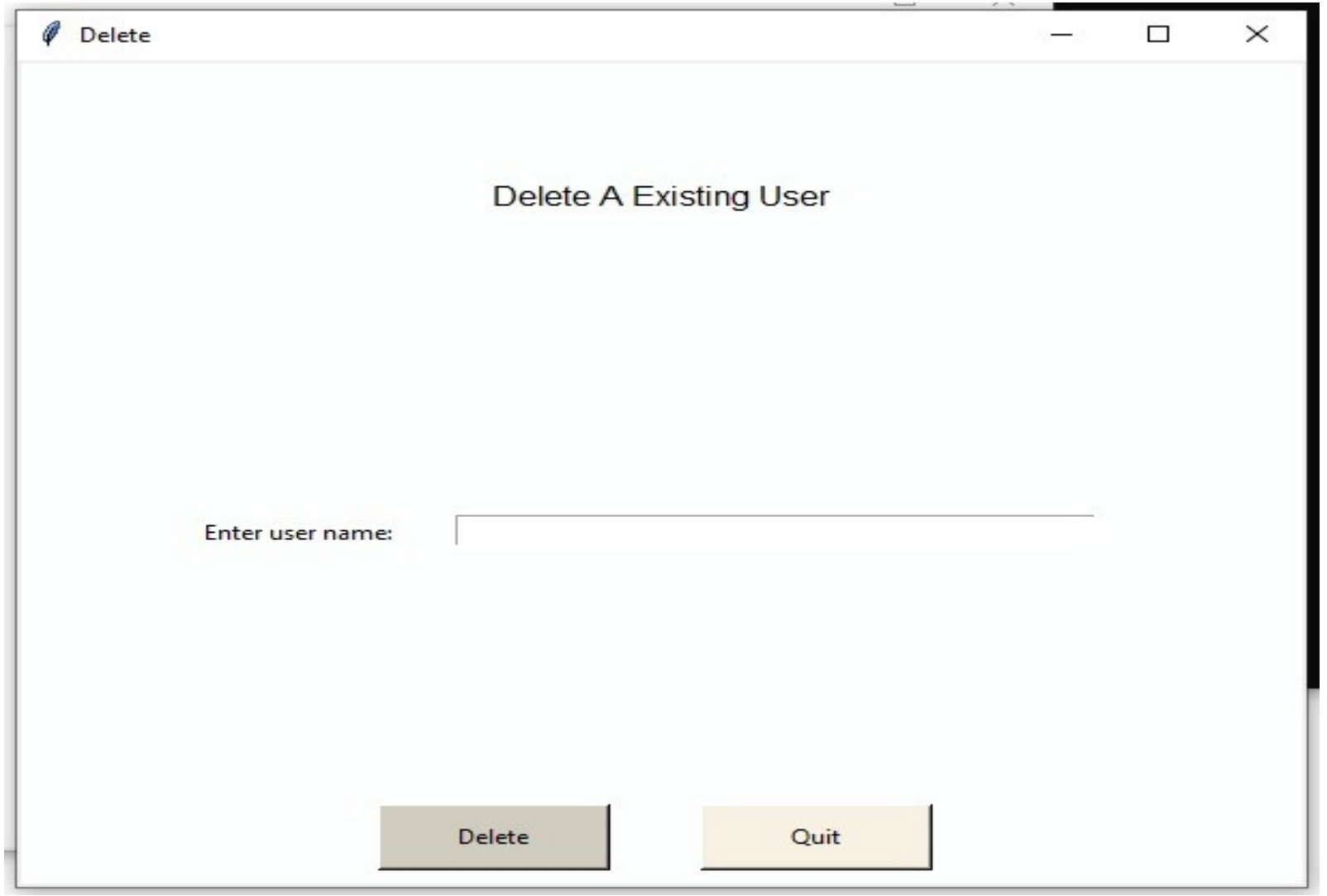
# Graphical User Interface(Cont.)

Add User Page:



# Graphical User Interface(Cont.)

Delete User:



# References


Contact Management System:

https://monday.com/blog/project-management/contact-management-database/

Entity Relationship Diagram: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

Enhanced Entity Relationship Diagram:

https://www.differencebetween.com/difference-between-er-and-vs-eer-diagram/

Primary Key and Foreign Key:

https://www.geeksforgeeks.org/difference-between-primary-key-and-foreign-key/

Optimization:

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_groupby1

CMS logo:

https://soffront.com/blog/features-free-contact-management-software/

Warehouse Management System:

https://github.com/RezaSadeghiWSU/Warehouse-Management-System

# Thank You!