COMP 1430 SYSTEM DESIGN AND DEVELOPMENT

COURSEWORK

MARIATA HOME

OLUBUNMI ADEOLU ADENEKAN

001278112

Table of Contents

Table of Figures

1.0 Introduction

Systems Design and Development course has afforded me the opportunity to understand the development life cycle of information systems and database applications.

1.1 Overview of the Scenario

In this scenario, Mariata Homes is a charity organization that provides subsidized accommodation to individuals, primarily those from lower socio-economic backgrounds. While most clients fall into this category, there are instances where unexpected middle-class individuals may also require temporary housing assistance. The typical duration of stay for clients is short, usually a few weeks.

To streamline the process of seeking assistance, Mariata Homes currently uses a paper-based form system. However, the manual documentation has become challenging to maintain, prompting the management to transition to a digital and online system.

As part of this transition, a functional prototype of a web-based application is required.

This application should allow users (clients) to register and log in, providing the necessary details for accommodation assistance. The required user details include:

- Name(s)
- DOB (Date of Birth)
- Email (if applicable)
- Telephone (if applicable)
- Next of Kin (if applicable)
- Age:
- Recent Passport Photograph
- Any Illness:
- Last Residence Address (if applicable)

The users seeking accommodation assistance will submit request details that include the requested date, recommended source, and recommended source address.

All users' details and accommodation requests are stored in the back-end database. Additionally, an admin interface should be implemented, allowing administrators to log in and view all users who have registered for assistance. The admin should have the capability to perform CRUD operations (Create, Read, Update, and Delete) on any user record.

1.2. Aim

The goal of this web-based application is to automate and streamline the client registration process, making it more efficient and manageable for Mariata Homes.

2.0 PART 1

To produce a demonstration of the process of Normalization to the 3rd normal form and identify the entities that would be used to produce a database at the backend for the ADMIN to perform the CRUD operation.

The identified identities are:

1. User
2. Admin
3. Accommodation Assistance

Prior to the process of normalization techniques, the identified entities with their attributes in unnormalized form are given below:

S/n	Entities
1	User
2	Admin
3	Accommodation Assistance

Figure 1: The identified Entities.

The identified entities with their attributes are as follows:

1. User:

S/N	User
	Name(s)
	Date of Birth (DOB)
	Email (if applicable)
	Telephone (if applicable)
	Next of Kin (if applicable)
	Age
	Recent Passport Photograph
	Illness
	Last Residence Address

Figure 2: User's entity with the attributes.

2. Admin

S/N	Admin
	Username
	Password
	Email

Figure 3: Admin's entity with the attributes

3. Accommodation Request

S/N	Accommodation Request
	Date of Request
	Accommodation Source
	Accommodation Source Address

Figure 4: Accommodation Request entity with the attributes.

The following assumptions are made to produce the process of data normalization to the 3NF:

1. The name(s) of the users /clients is the full name.
2. The date of birth is a unique field and age is also a unique field.
3. The address is pointing to one location.
4. The User has only one email address.
5. The recommended source and recommended source address are associated with the type of accommodation assistance request.

Database normalization is a process used in database design to organize data and eliminate redundancy to make it robust and efficient. The raw dataset for the User entity in an unnormalized form is represented below:

Name(s)	Date of Birth	Email	Next of Kin	Age	Photo	Illness	Last Residence Address
Ajayi Olaoluwa	27/10/1974	Aola12@yahoo.com	Bola Olaoluwa	49	Image.jpg	N/A	12, Crescent Avenue, Lagos.
Olabode Adeoti	20/07/1980	Olaode1234@gmail.com	Aina Adeoti	43	Image.jpg	N/A	14, Saka Tinbu, Victoria Island.
Adeolu Amos	10/04/1985	Adeoluamos@gmail.com	Victoria Amos	38	Image.jpg	N/A	14 Frobisher Street, Lagos.

Figure 5: The unnormalized form of the User's table.

In the above table, the rules of 1NF are there must be no repeated data, the data must be atomic, each field must have a unique name and each row must be unique, i.e., it must have a primary key.

Therefore,

The 1st normalization form(1NF) is produced as follows:

- **User Table**

UserID(PK)	Name	DateofBirth	Email	NextOfKin	Age	Photo	Illness	Last Residence Address
001	Ajayi Olaoluwa	27/10/1974	Aola12@yahoo.com	Bola Olaoluwa	49	Image.jpg	N/A	12, Crescent Avenue, Lagos.
002	Olabode Adeoti	20/07/1980	Olaode1234@gmail.com	Aina Adeoti	43	Image.jpg	N/A	14, Saka Tinbu, Victoria Island.
003	Adeolu Amos	10/04/1985	Adeoluamos@gmail.com	Victoria Amos	38	Image.jpg	N/A	14 Frobisher Street, Lagos.

Figure 6: INF of the User's Table.

To proceed with the 2nd normal form, the following rules must be satisfied:

- The table must already be in INF.
- The non-key attributes in the data set must depend on every part of the primary key.

The above table satisfies these rules and therefore it is already in 2NF.

To obtain the 3NF of the above User's table, the following rules apply:

- If it is already in second normal form (2NF).
- There are non-key attributes that depend on another key attribute.

We can see from the User's table that the attribute, Age could be obtained from Date of Birth, therefore that field, Age must be removed for redundancy and efficient data robustness.

UserID(PK)	Name	DateOfBirth	Email	NextofKin	Photo	Illness	Line1Address	Line2Address
001	Ajayi Olaoluwa	27/10/1974	Aola12@yahoo.com	Bola Olaoluwa	Image.jpg	N/A	12, Crescent Avenue	Lagos.
002	Olabode Adeoti	20/07/1980	Olaode1234@gmail.com	Aina Adeoti	Image.jpg	N/A	14, Saka Tinubu.	Victoria Island
003	Adeolu Amos	10/04/1985	Adeoluamos@gmail.com	Victoria Amos	Image.jpg	N/A	14, Frobisher Street	Lagos

Figure 7: 3NF of the User's Table.

To obtain the 3NF for the admin entity, it follows the same procedure as represented below:

Username	Password	Email
Ajanny4ever*	**********	Ajanny@yahoo.com
Bold4me@#	**********	Boldme@gmail.com
Obagol*	**********	Obagol@yahoo.com

Figure 8: The unnormalized form of the Admin's Table.

To proceed with the normalization process of the above Admin table, the rules of the 1NF must be satisfied, and as shown above, in the case of the User's table where the primary key will be introduced with no repeated data, the data must be atomic, and the field must have a unique name.

Therefore, the below table shows the 2NF of the Admin's Table.

AdminID(PK)	Username	Password	Email
001	Ajanny4ever*	**********	Ajanny@yahoo.com
002	Bold4me@#	**********	Boldme@gmail.com
003	Obagol*	**********	Obagol@yahoo.com

Figure 9: 2NF of the admin's entity.

It is imperative to know that the above table satisfies all the rules of 3NF, i.e., the table must be in 2NF and there are non-key attributes that depend on another key attribute.

Therefore, the below table shows the 3NF of the admin's table.

AdminID(PK)	Username	Password	Email
001	Ajanny4ever*	**********	Ajanny@yahoo.com
002	Bold4me@#	**********	Boldme@gmail.com

003	Obagol*	**********	Obagol@yahoo.com

Figure 10: 3NF of the admin's entity.

2.1 The Entity Relationship Diagram

The below diagram shows the entity relationship diagram for the system. Each of the entities has an attribute associated with it.

There is a one-to-many relationship between the User and the Accommodation Request entity which means that one user can make multiple accommodation requests, but each request is associated with only one user.

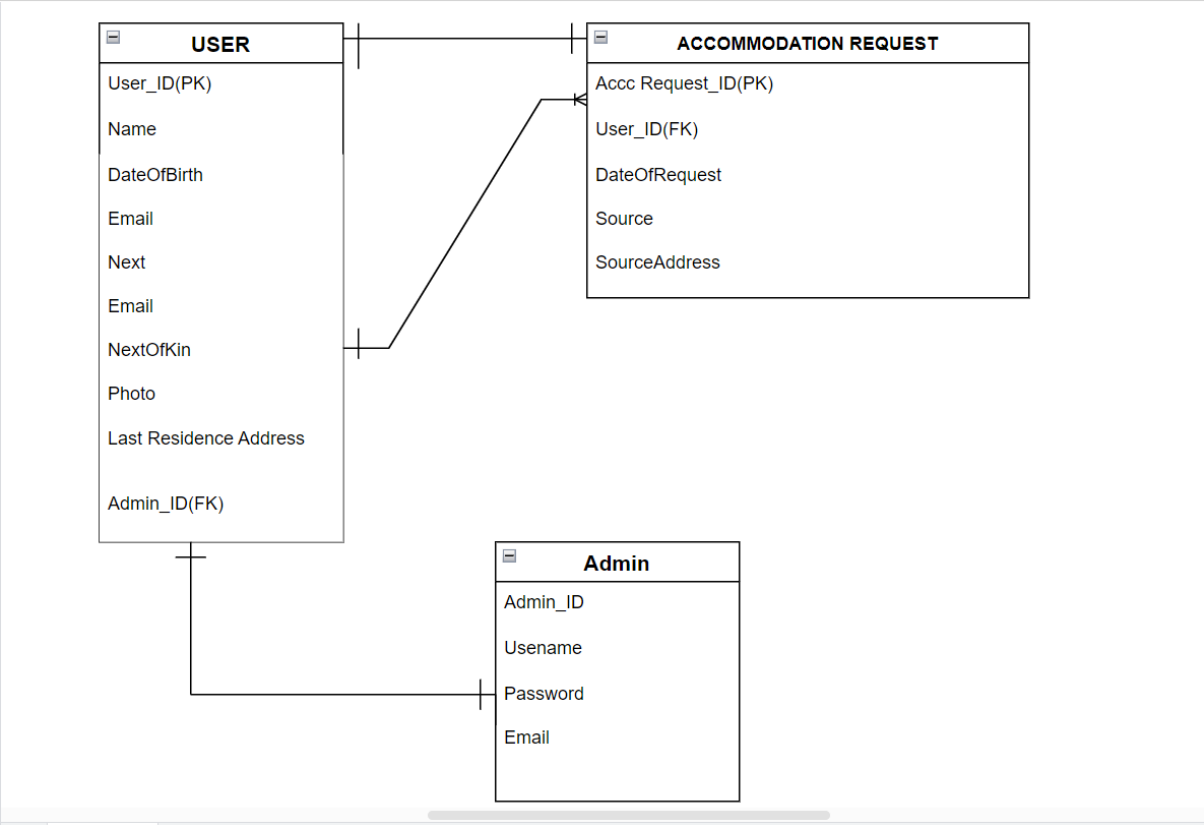


Figure 11: The Entity Relationship Diagram (ERD).

2.2 Unified Modelling Language (UML) Class Diagram

The below diagram provides each class as it is represented in the boxes, i.e., User, Accommodation Request, Client, and Admin with their associated attributes. The second line shows the attributes with the data types,

such as int, string, and date. The third line which is the class line provides functionalities with either private or public signs indicating meeting all necessary indicators.

Also, the visibility, inheritance, association, composition, and multiplicity are ably represented.

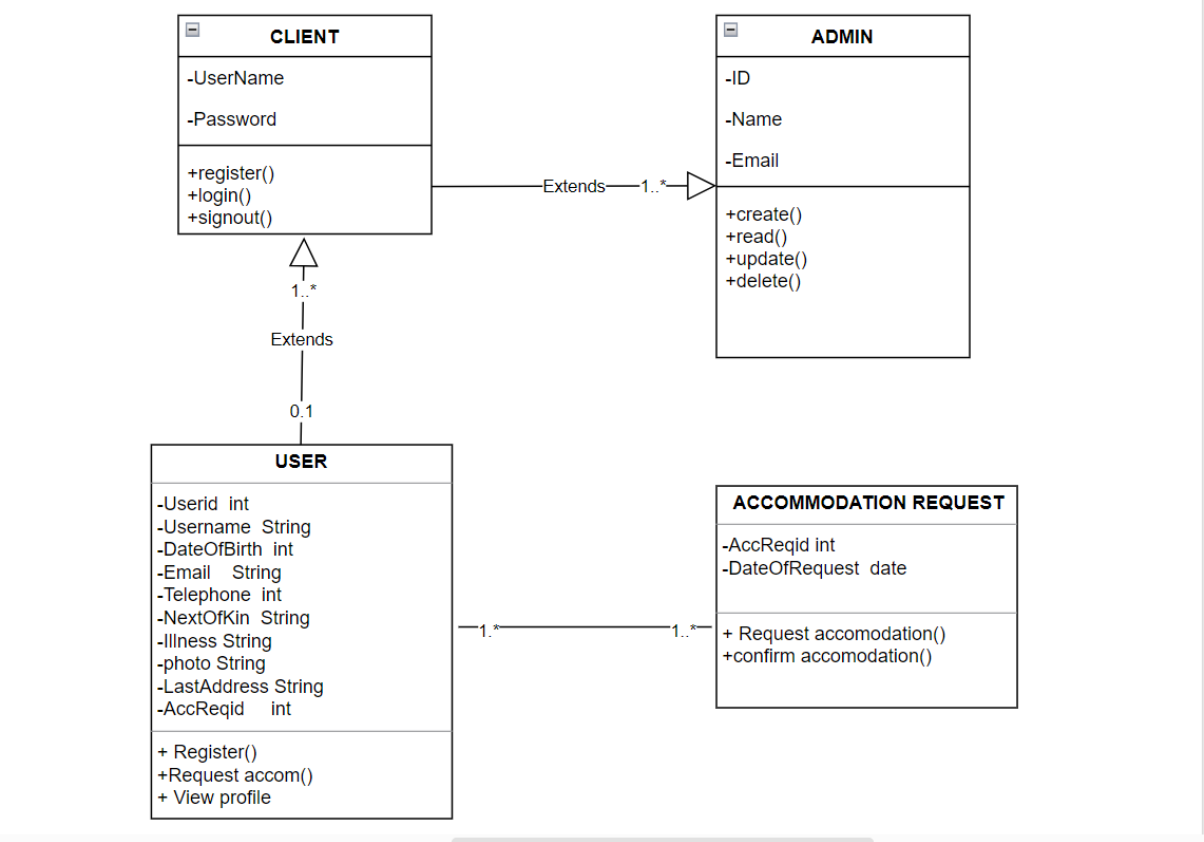


Figure 12: The (UML) Class Diagram.

2.3 Use Case Diagram

The use case diagram depicts the system activities, admin functionalities, and the user/client registration process. The users/clients register by providing personal details, after registration the users can log in and submit an accommodation request. The admin logs into the system using a username and password to access administrative rights to create, read, update, and delete users' records. The System manages registrations and allot accommodations to clients who meet the criteria.

These use cases cover the major interactions and goals for both users/clients and admins within the system.

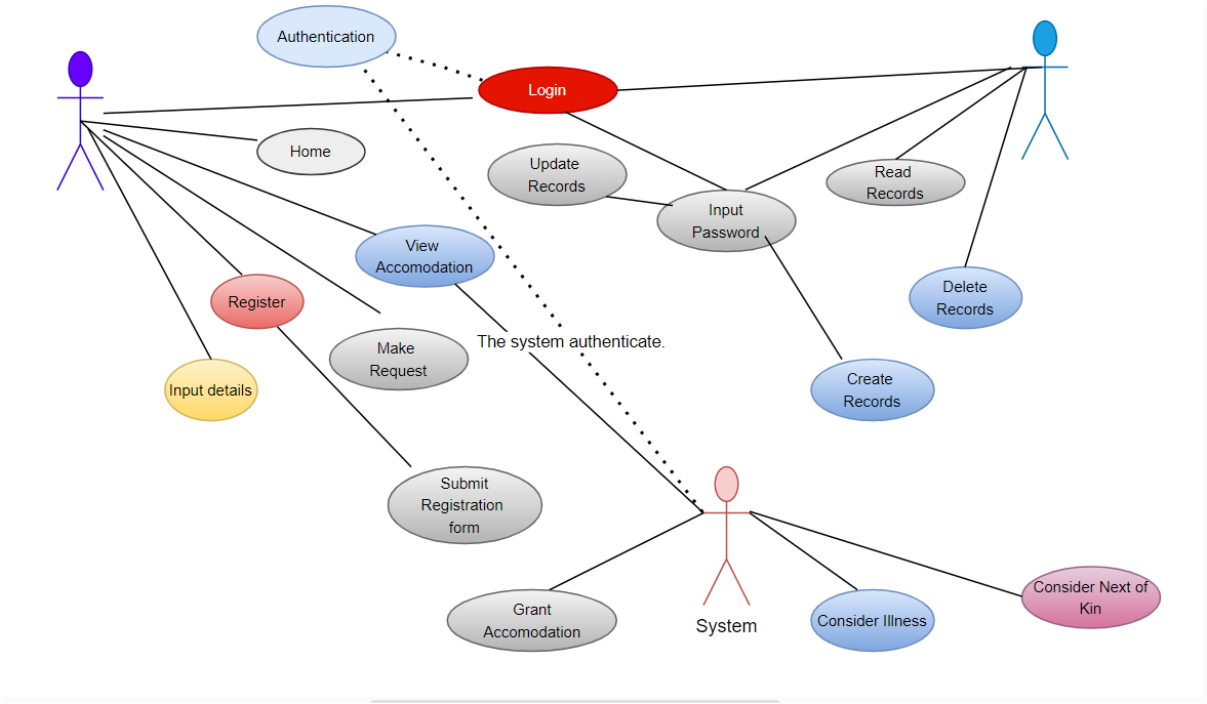


Figure 13: The use case diagram.

2.4 USER INTERFACE FRONT END FORMS OF MARIATA HOME

All the user interface front and end forms for new clients/users are shown below. The user inputs all the requested personal details and all the data are protected.

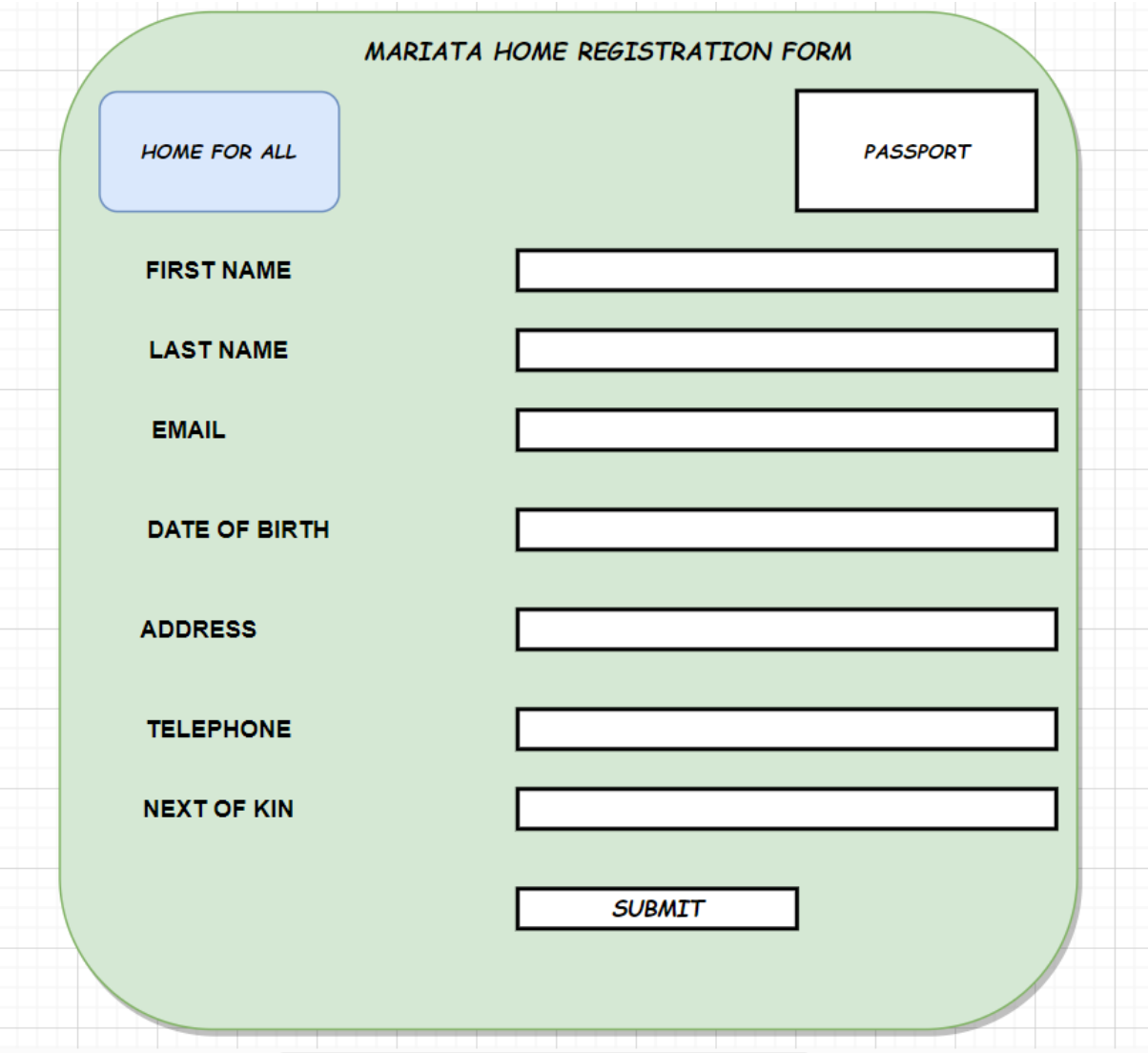


Figure 14: Registration form of Mariata Home.

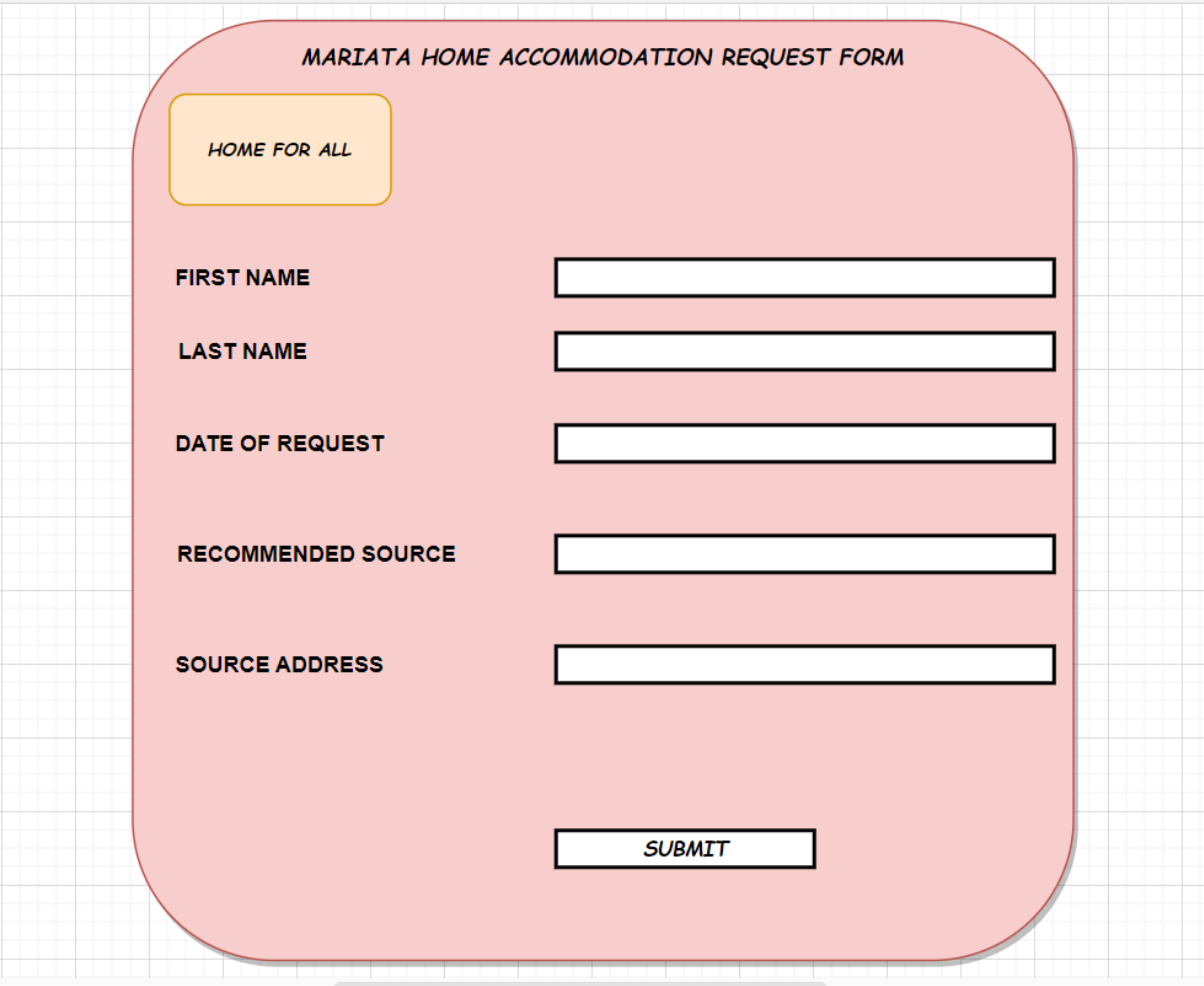


Figure 15: Accommodation Request Form.

The new users/clients provide all the details requested before the accommodation assistance can be granted.

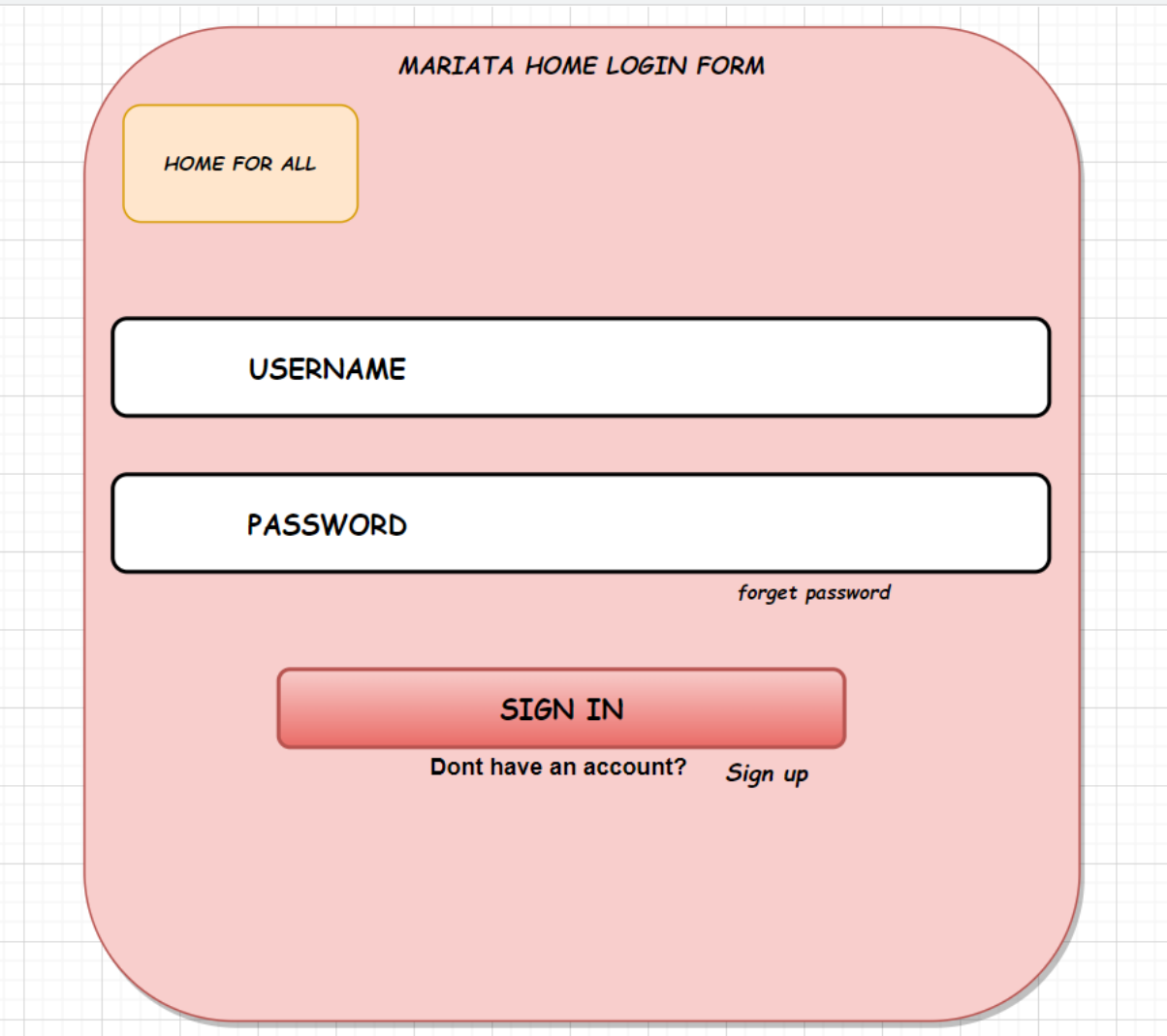


Figure 16: The Login form for the existing users/clients.

The login form shows that the existing users/clients would need to input the login details before accessing the services rendered by Mariata Home.

3.0 Overview of the Scenario

The scenario is about developing a web-based content management system (CMS) that allows users or clients to register before they can log in using a username and password. After logging in, the user should be able to input their details to accommodate the scenario and should not be able to delete or update his or other users' information.

The user login details, and information should be stored permanently in the backend database. The admin login user should be able to see all registered users in a web interface and can perform the Create, Read, Update, and Delete (CRUD) operation on them.

In summary, the developed systems should do the following among others.

(a) Registration of user and admin
(b) Input user details and information
(c) Admin user's login
(d) Admin users can access all users' information.
(e) Admin user able to perform CRUD operations on all users.
(f) Other as you deemed necessary.

It is important that the web-based content management system (CMS) should be developed as a standalone, which means that it must not be hosted on the university server. The section below provides the system and the CRUD operations with clear screenshot annotation and explanation.

3.1 Registration of User and Admin

First, before the registration of users and admin task was carried out, the web design front page of **Mariata Home** was designed using Bootstrap.

The Users and the Admins would have to first register but after the registrations, the admin would be given administrative rights to perform CRUD operations on the users.

Figure 17: The Main Home Page.

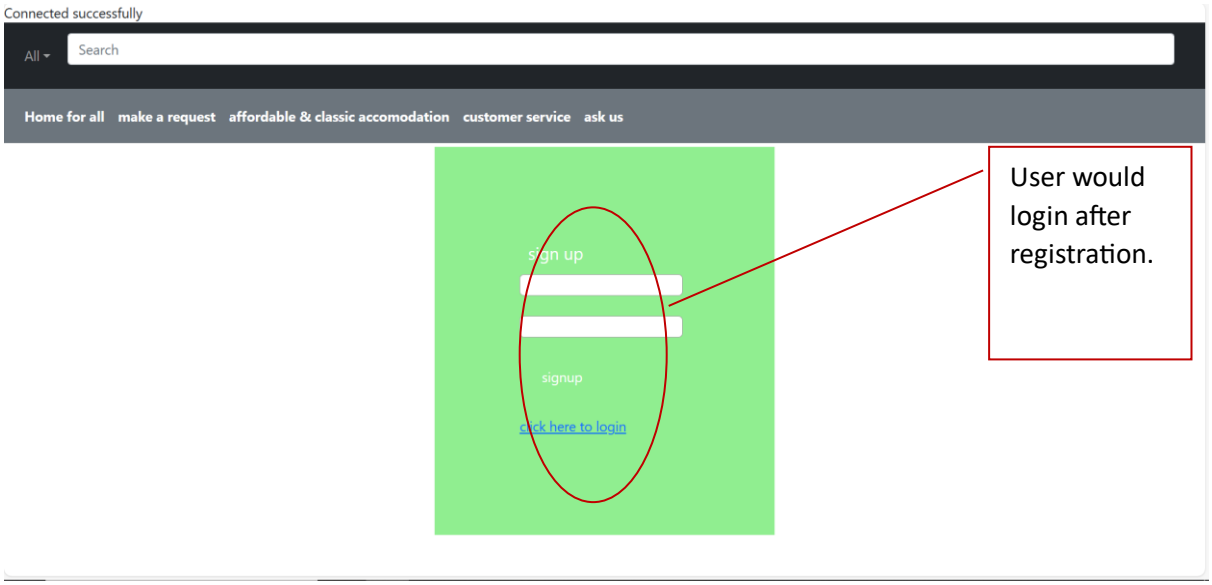


Figure 18: Signup page for all new users.

3.2 Input users' details and information

The screenshot with annotation clearly shows when the user is inputting his data for registration to **MARIATA HOME.**

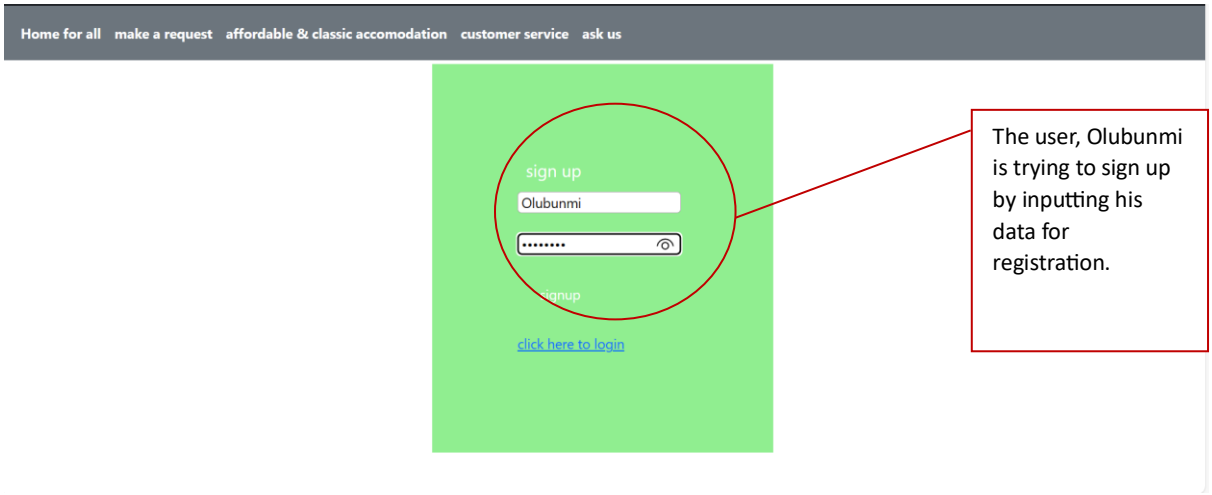


Figure 19: A new user assessing the sign-up page.

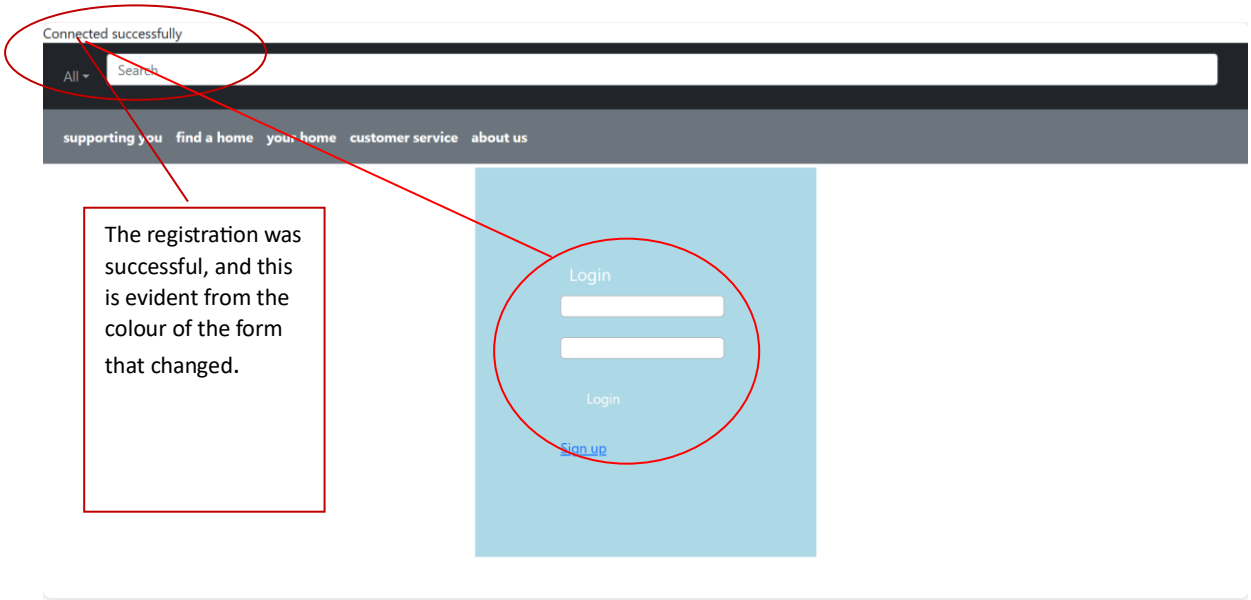


Figure 20: Successful registration of the user.

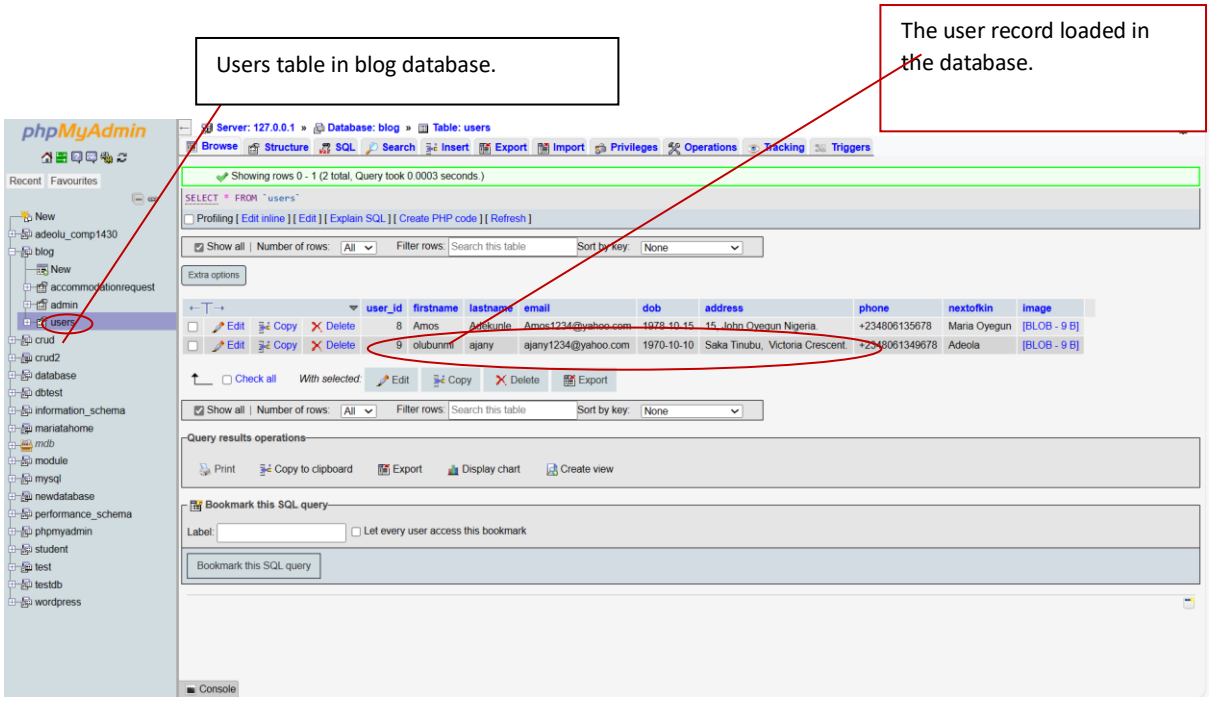


Figure 21: Database showing the successful user registration.

3.3 Registration of Admin with user's login

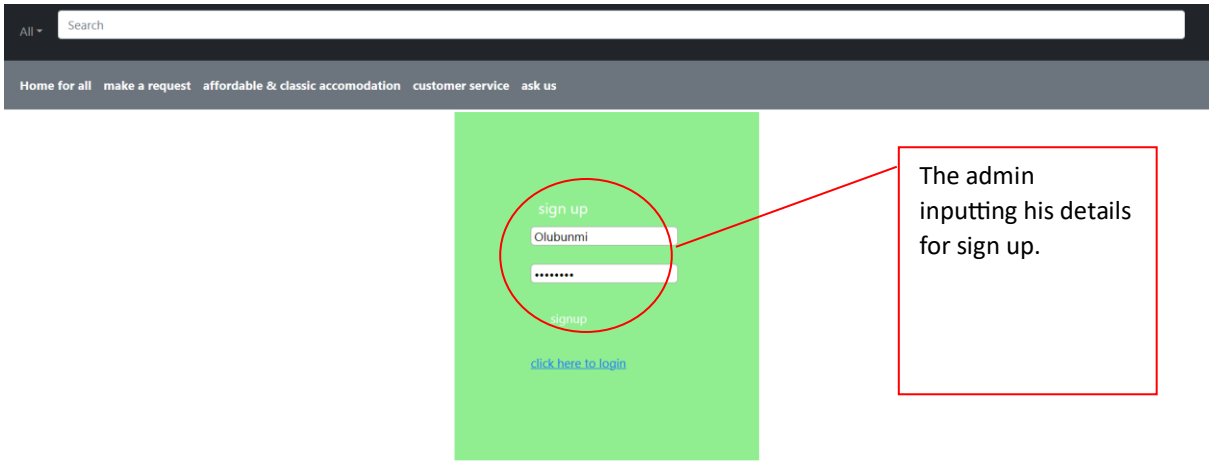


Figure 22: Admin sign-up.

The admin sign up by inputting his details and this was confirmed in the database as shown.

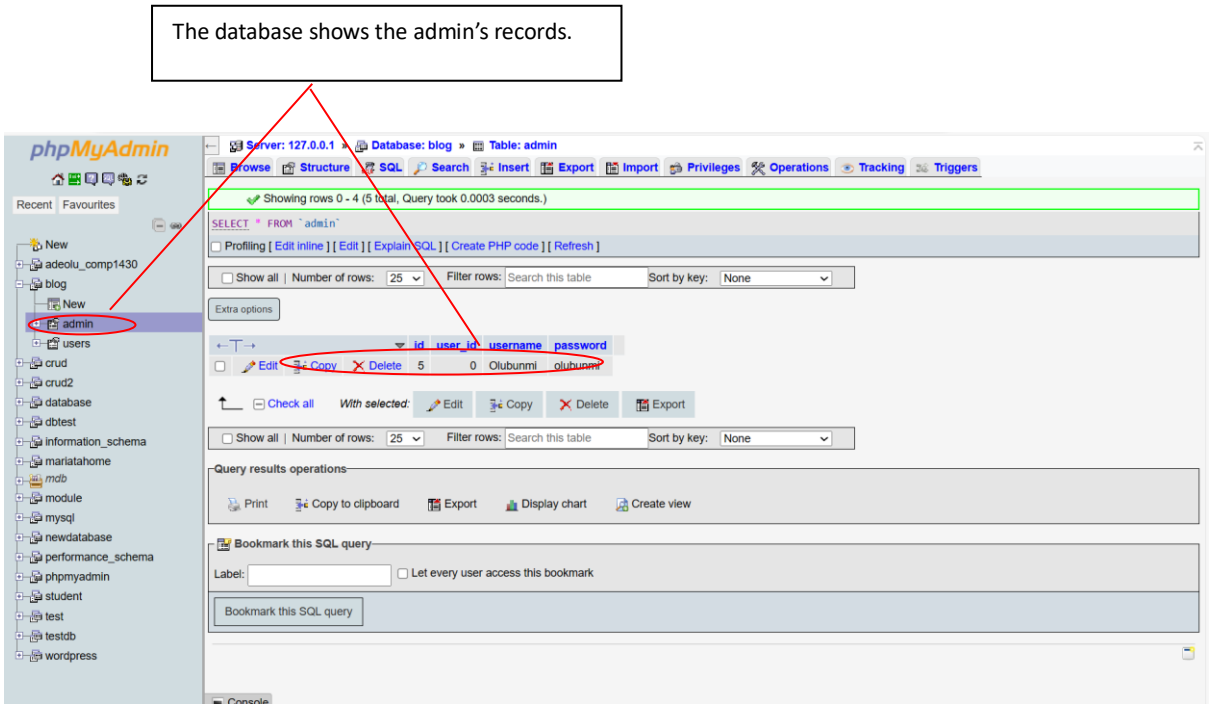


Figure 23: Admin records in the database.

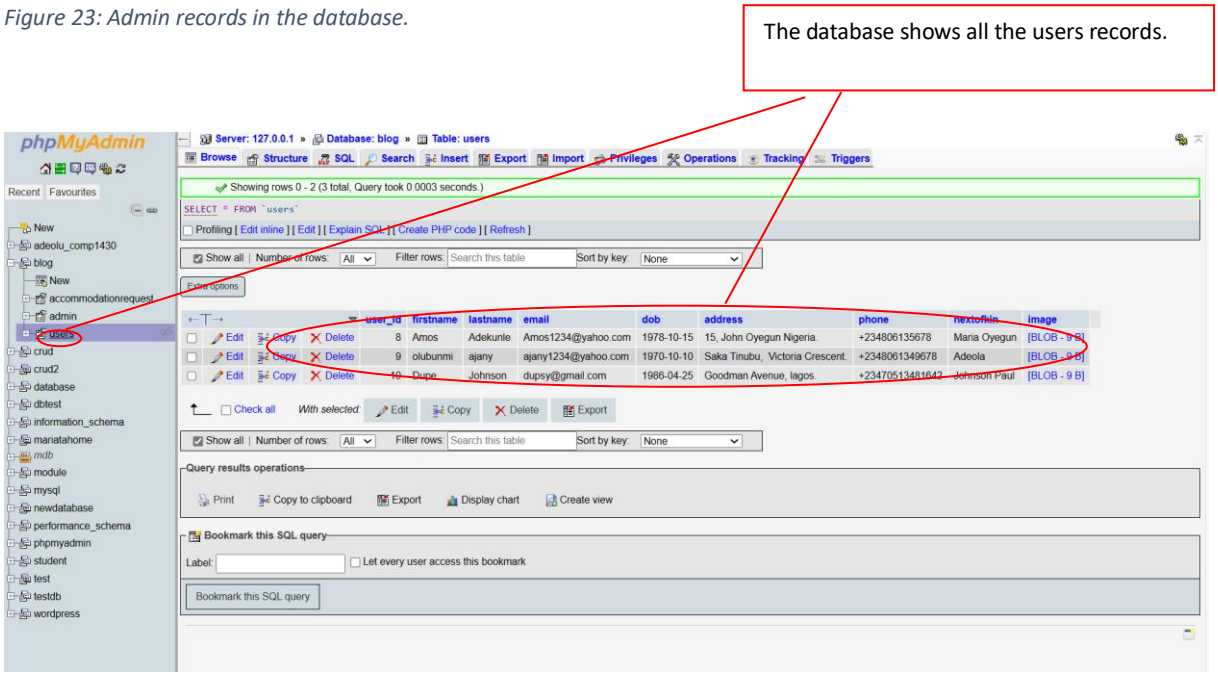


Figure 24: Users record in the database.

3.4. Admin users access all users' records.

This is evident in the below annotated screenshot that the admin can view, delete and edit users' information as deemed necessary.

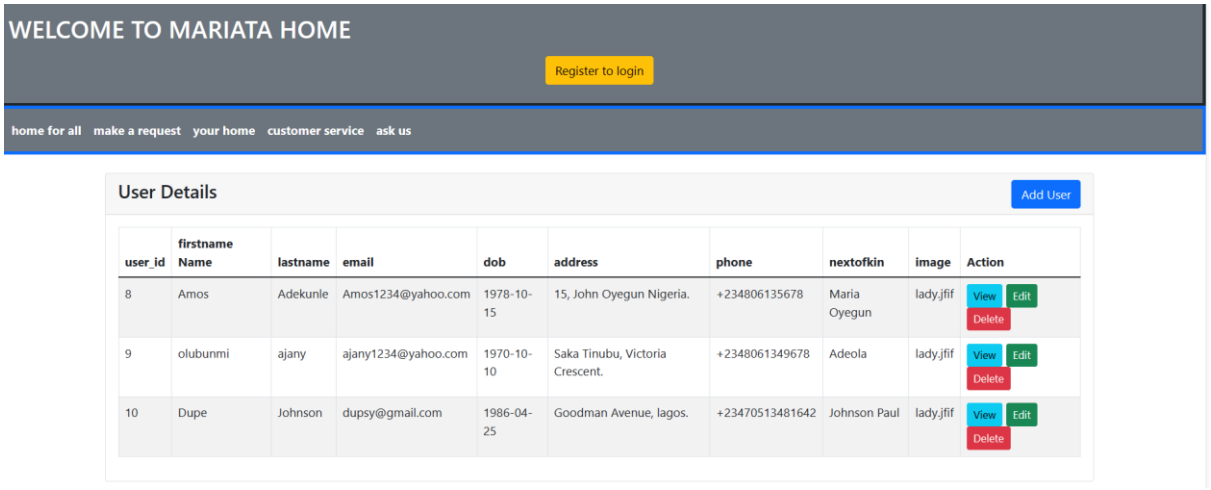


Figure 25: A page where Admin can assess all users' records.

3.5 CRUD operations performed on all users by Admin.

- **Create.**

The admin should be able to perform create, read, update, and delete operations on all the users as he has the absolute rights to do that. The operation is captured below in the screenshot with annotations.

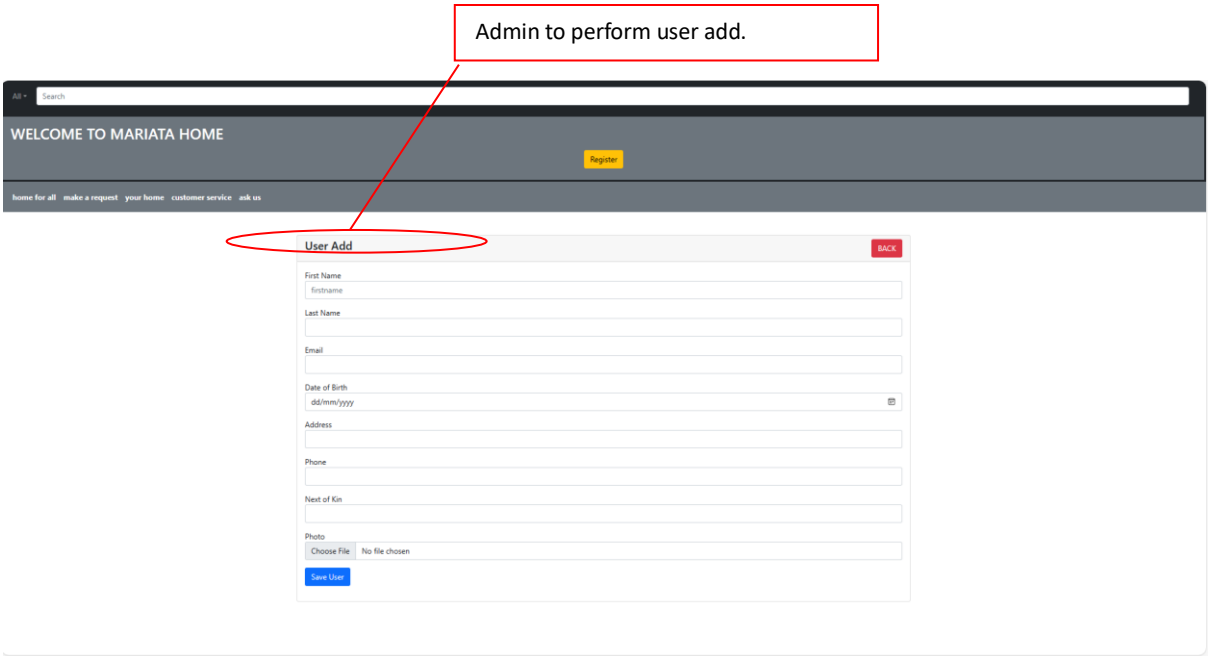


Figure 26: Admin's add operation on the user.

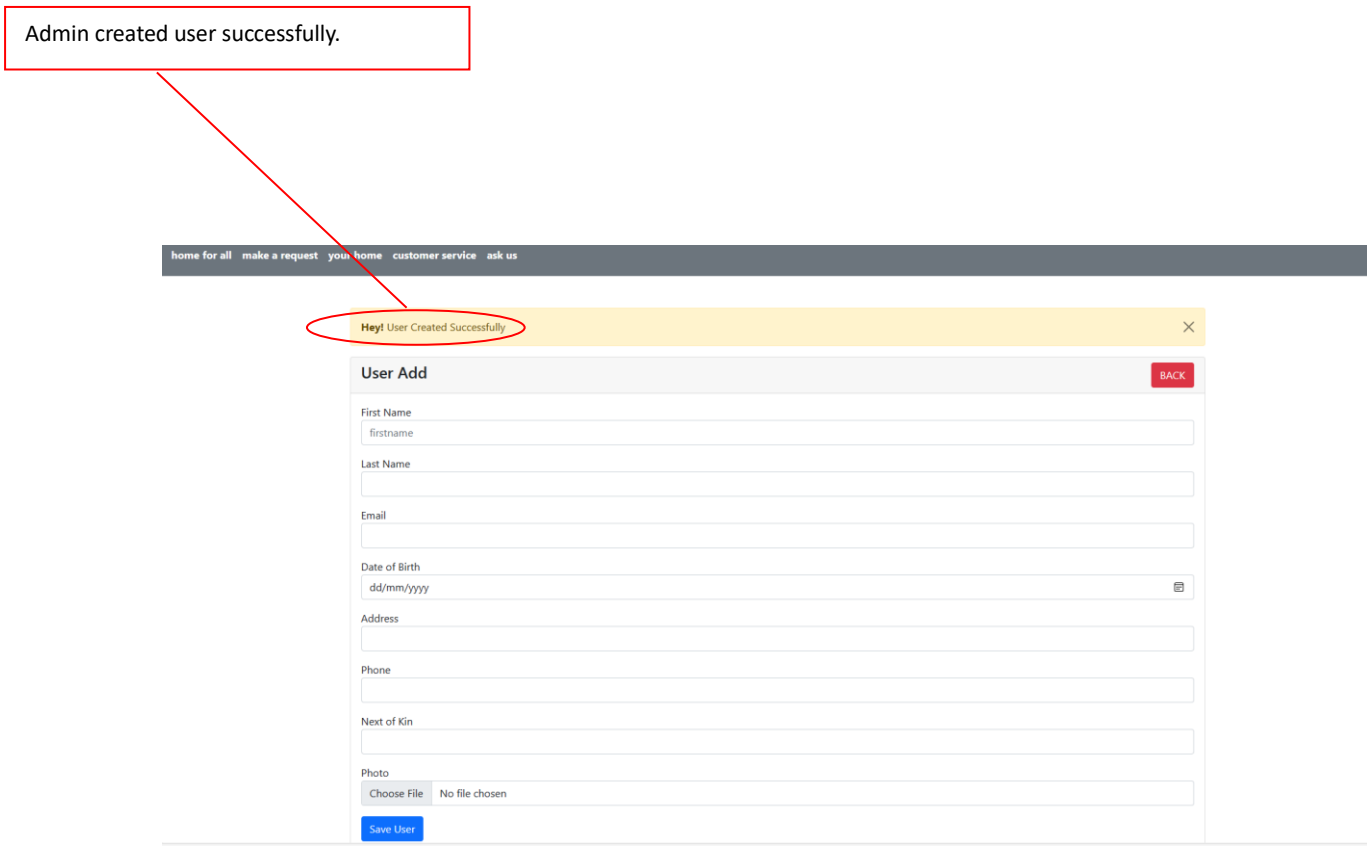
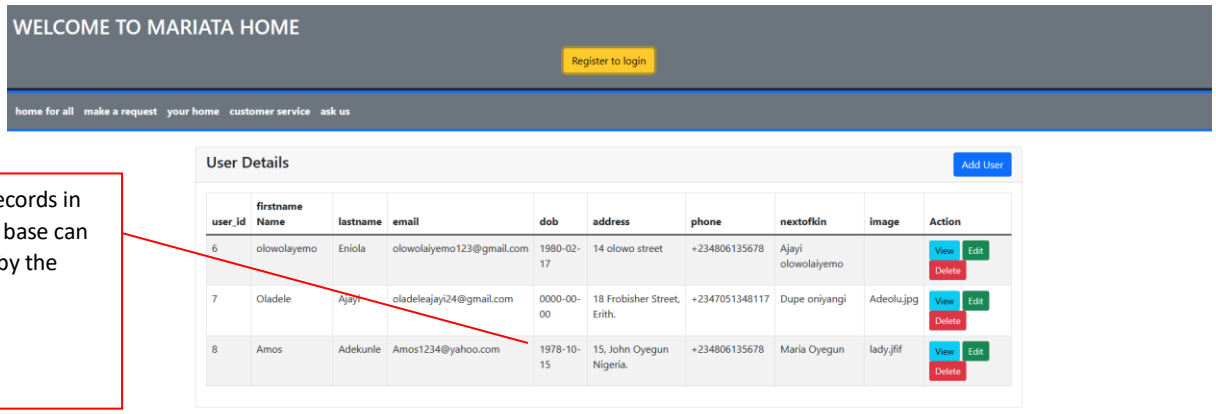


Figure 27: Admins create operation on the user.

- **Read Operation**



All the records in the data base can be read by the admin.

Figure 28: Read operation performed by the admin.

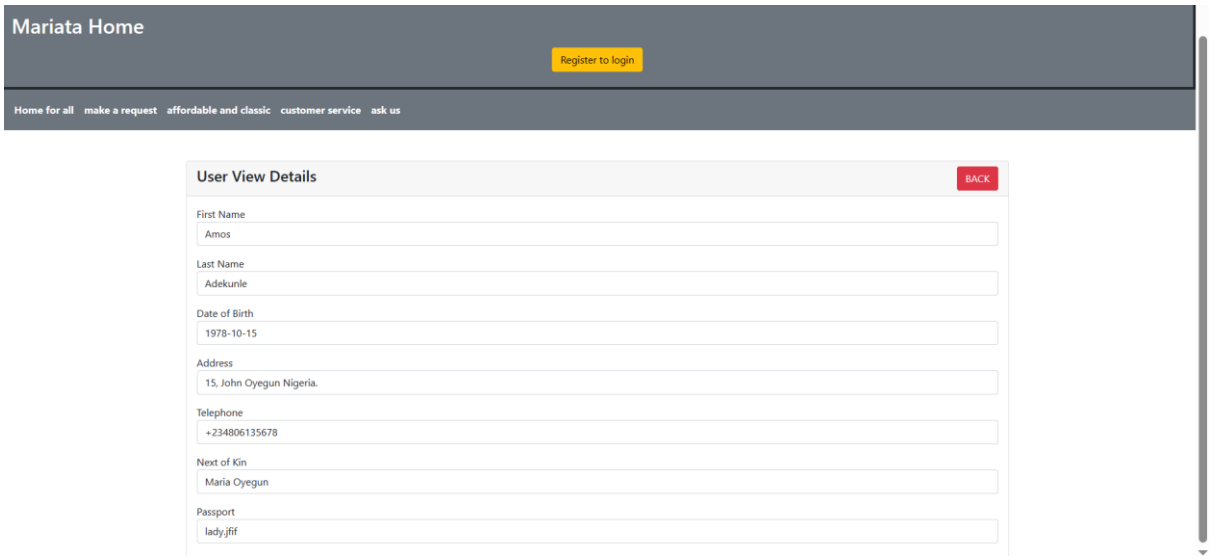


Figure 29: Admin can view all the records in the database.

Delete operation performed by the admin.

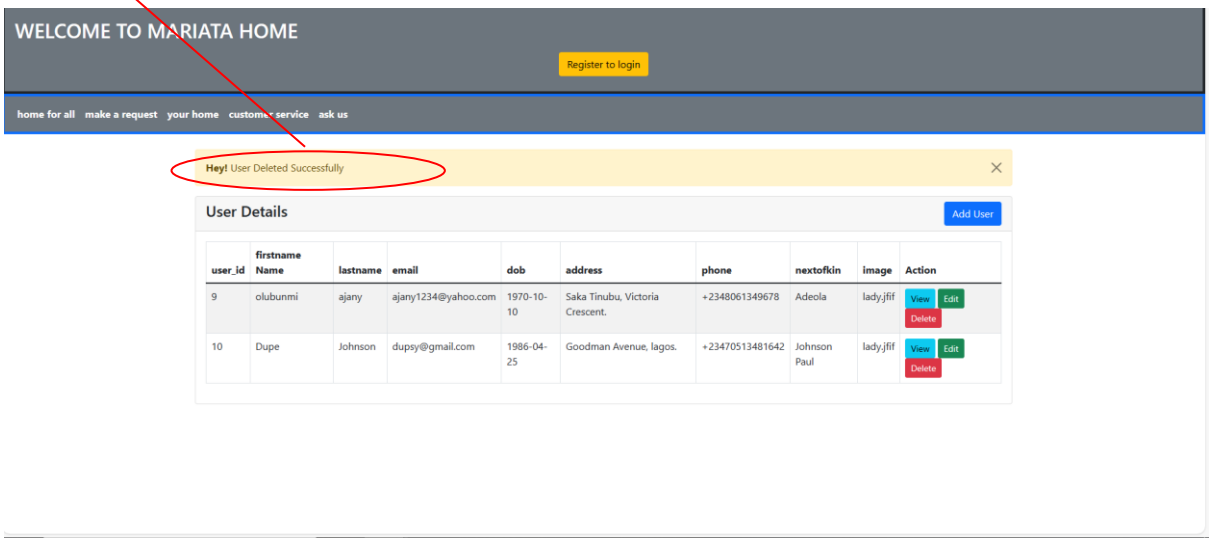
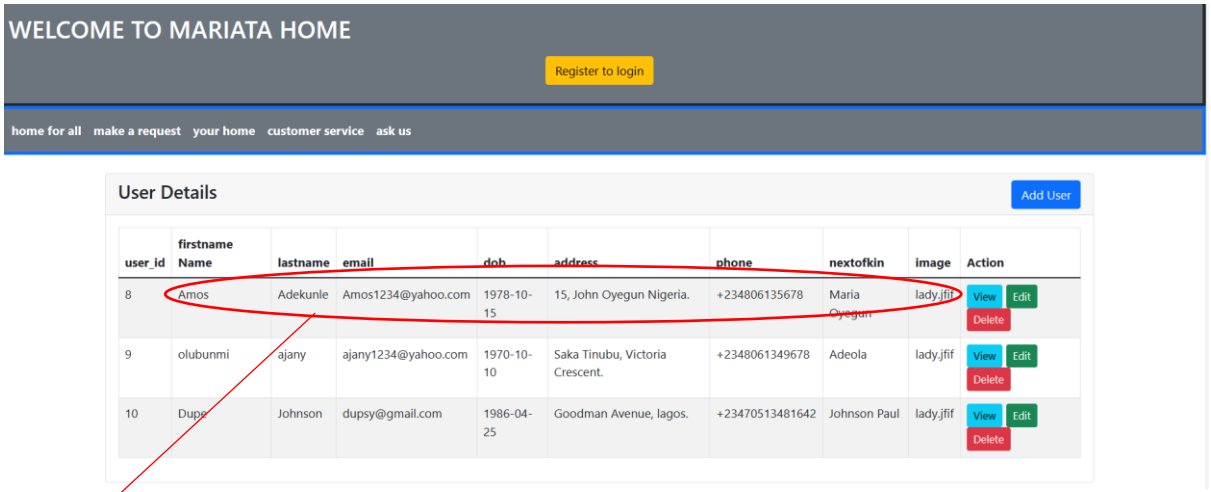


Figure 30: Admin deleted a user successfully.



The records of the user in the database before it was deleted by admin.

Figure 31: User records in the database.

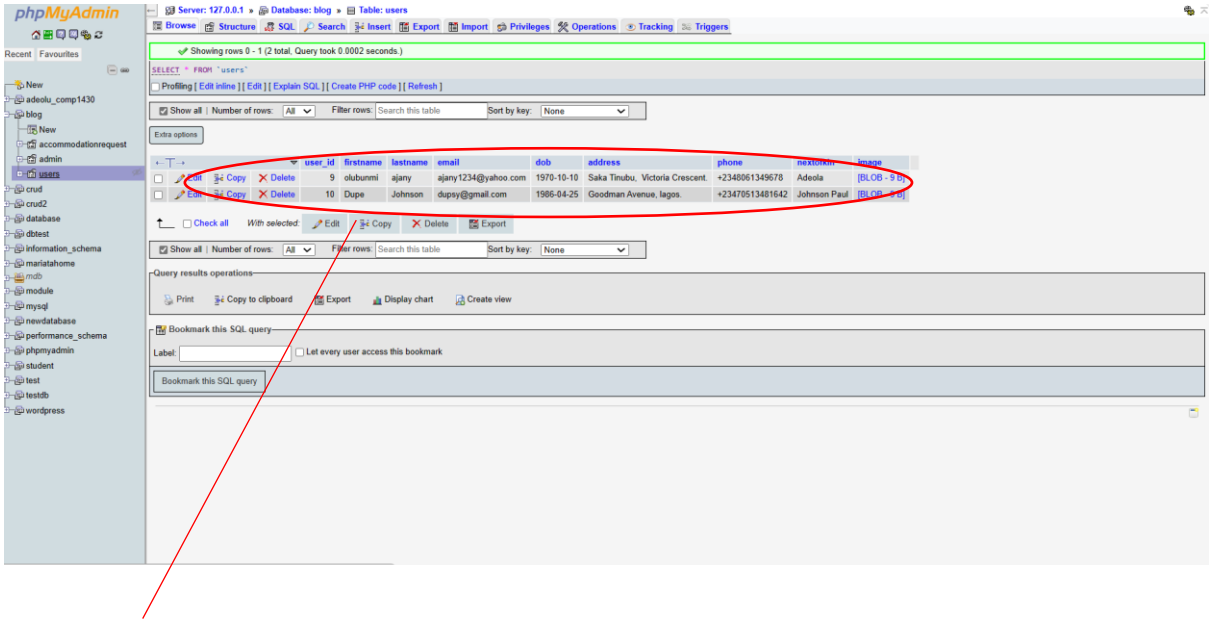


Figure 32: Database showing the records have been deleted.

4.0 Conclusion

System Design and Development has a very rich content in software development life cycle of information systems and what I have learned in this course has really given me an insight into data analytics. The course has made me to understand the CRUD system at the front end and back end.

5.0 References.

Chong Lip Phang (2022). *HTML, Boostrap, CSS, Tailwind, & Cordova.* Chong Lip Phang.