

Coursework of
COMP1430 Systems Design and Development
Module Leader: Dr Solomon H. Ebenuwa

Name: Abdullah Al Sohel

ID: 001287370

Programme: MSc Computing and Information Systems

University of Greenwich

Table of Contents

Table of Contents.....	2
Table of Figures.....	3
Introduction.....	5
Part 1: Systems Design.....	6
1. Database Normalization to 3rd Normal Form (3NF).....	6
2. Entity Relationship Diagram (ERD).....	10
3. Unified Modeling Language (UML) Class Diagram.....	12
4. Use Case Diagrams.....	15
5. User Interface (UI) Front-end Forms.....	16
Part 2: Systems Development.....	22
Technologies and Tools Used.....	22
2(a) Registration of User and Admin.....	23
2(b) Input User Details and Information.....	25
2(c) Admin User's Login.....	28
2(d) Admin Users Are Able to Perform CRUD Operations on All Users.....	30
2(e) Other Functionalities (Search Users, Generate Reports).....	38
Testing.....	42
Conclusion.....	43
References.....	44

Table of Figures

- [Figure 1: Entity Relationship Diagram](#)
- [Figure 2: UML Class Diagram](#)
- [Figure 3: Use Case Diagram](#)
- [Figure 4: Registration Form](#)
- [Figure 5: Login Form](#)
- [Figure 6: Form for creating client user](#)
- [Figure 7: Client User Create Form for admin](#)
- [Figure 8: Search Form](#)
- [Figure 9: User created successfully](#)
- [Figure 10: Admin created successfully](#)
- [Figure 11: Created User in database](#)
- [Figure 12: Created Admin in database](#)
- [Figure 13: User Password confirms](#)
- [Figure 14: Check empty input field for user](#)
- [Figure 15: Password length check for user registration](#)
- [Figure 16: Password length check for admin registration](#)
- [Figure 17: Check empty input field for admin](#)
- [Figure 18: checking inputs validation](#)
- [Figure 19: Showing error in frontend](#)
- [Figure 20: Redirects admin to their profile page with flash message](#)
- [Figure 21: Redirects client to their profile page with flash message](#)
- [Figure 22: login checking in admin controller](#)
- [Figure 23: Crud operation for users in admin dashboard](#)
- [Figure 24: Crud operation for admin](#)
- [Figure 25: User creation form](#)
- [Figure 26: Admin creation form](#)
- [Figure 27: User page from admin dashboard](#)
- [Figure 28: Admin profile](#)
- [Figure 29: Client User edit form](#)
- [Figure 30: Admin edit form](#)
- [Figure 31: Before updating the user in user profile](#)
- [Figure 32: After updating the user](#)
- [Figure 33: user delete confirmation](#)
- [Figure 34: Gate facade in app/Providers/AuthService](#)
- [Figure 35: Specifying all the routes for CRUD of all type of users that should be performed with admin access](#)
- [Figure 36: Before updating in database](#)
- [Figure 38: Client user search](#)
- [Figure 39: Admin user search](#)
- [Figure 40: Before searching recommended source](#)
- [Figure 41: After searching source](#)
- [Figure 42: Report download button](#)

[Figure 43: Report downloaded](#)

[Figure 44: Report in pdf](#)

[Figure 45: Search logic in controller](#)

[Figure 46: All 18 tests passed.](#)

Introduction

This coursework for COMP1430 Systems Design and Development involves designing and developing a web-based CRUD system for Mariata Homes. It covers everything from conceptual design to practical application, emphasizing structured database design, web interface development, and backend functionality.

Part 1 focuses on system design theory. This includes database normalization for efficient data storage, creating an Entity Relationship Diagram (ERD), and developing a Unified Modeling Language (UML) class diagram. Use case diagrams map user-system interactions, laying the groundwork for a system tailored to Mariata Homes' needs.

Part 2 moves from theory to implementation. It brings the database schema, ERD, and UML class diagrams to life through front-end and back-end development. This phase includes coding the web application, focusing on user interface for clients and administrators and robust back-end data management, demonstrating the practical application of theoretical design principles.

Part 1: Systems Design

1. Database Normalization to 3rd Normal Form (3NF)

1.1 Unnormalized Data Structure Description

The initial structure for the Mariata Homes database contains a single table, 'Clients,' which includes comprehensive client information and their recommended sources. This table, in its unnormalized state, has fields that are not atomic and contains repeating groups, violating the principles of the First Normal Form (1NF).

Unnormalized Form (UNF):

Table: Clients

- ID (Primary Key)
 - Username
 - Email
 - Password
 - User Role (default 'client')
 - Name(s)
 - Date of Birth
 - Telephone
 - Next of Kin
 - Age
 - Recent Passport Photograph
 - Any Illness
 - Last Residence Address
 - Recommended Source Type
 - Recommended Source Address

ID (PK)	User name	Email	Pass word	User Role	Nam e	DOB	Telep hone	Next of Kin	Age	Phot ograp h	Illnes s	Addr ess	Sour ce Type	Sour ce Addr ess
------------	--------------	-------	--------------	--------------	----------	-----	---------------	----------------	-----	--------------------	-------------	-------------	--------------------	---------------------------

Issues in UNF:

Because the 'Name(s)' field is a composite field, it must be separated for atomicity. 'Recommended Source Type' and 'Recommended Source Address' appear to be repeating groups, indicating a many-to-one relationship that should be separated into a distinct table.

1.2 Transition to 1NF

Normalization starts with making sure that each field is atomic and that there are no repeated groups (Codd, 1970). This is consistent with Codd's principles, which emphasize the value of atomicity in relational database design (Date, 2004).

- The 'Name(s)' field is split into 'First Name' and 'Last Name'.
- 'Recommended Source' information is separated into a distinct table.

The resultant tables are as follows:

Table 1: Users (1NF)

- ID (Primary Key)
- Username
- Email
- Password
- User Role

ID(PK)	username	email	password	User role
--------	----------	-------	----------	-----------

Table 2: Profiles (1NF)

- ID (Primary Key)
- First Name
- Last Name
- Date of Birth
- Telephone
- Next of Kin
- Age
- Recent Passport Photograph

- Any Illness
- Last Residence Address
- User ID (Foreign Key)

ID(PK)	First Name	Last Name	DOB	Telep hone	Next of Kin	Age	Photo graph	Illness	Address	User ID (FK)
---------	------------	-----------	-----	------------	-------------	-----	-------------	---------	---------	--------------

Table 3: Recommended Source (1NF)

- ID (Primary Key)
- Source Type
- Source Address
- Profile ID (Foreign Key)

ID(PK)	Source type	Source Address	Profile ID (FK)
--------	-------------	----------------	-----------------

1.3 Transition to 2NF

Once in 1NF, the next step is to make sure that all attributes that aren't keys rely on the primary key in every way. Elmasri and Navathe (2016) say that 2NF is reached when all non-key attributes rely on the primary key in order to work.

All non-key attributes must be fully dependent on the main key in order to achieve 2NF. Because the 'Personal Information' and 'Recommended Sources' tables already have their non-key characteristics entirely dependent on their respective main keys, they are deemed to be in 2NF.

1.4 Transition to 3NF

To reach 3NF, a table must not only be in 2NF, but all of its characteristics must be completely dependent on the primary key, eliminating transitive dependencies (Date, 2004). Codd (1970) says that this step is very important to get rid of duplicate data and keep the purity of the data.

A table can only be in 3NF if it is also in 2NF and all of its attributes depend on the main key and not on any other attributes that aren't keys. The 'Users,' 'Profiles,' and 'Recommended Source' tables all meet these criteria, indicating that they are in 3NF.

After reviewing the tables, it is evident that:

- Users Table: All attributes are dependent only on the primary key (ID).
- Personal Information Table: All attributes are dependent only on the primary key (ID).
- Recommended Sources Table: All attributes are dependent only on the primary key (ID).

All of the tables are in 3NF because there are no transitive relationships and each attribute only depends on its own primary key.

Three separate tables are made during the standardization process: Users, Profiles, and Recommended Source. This structure significantly decreases data redundancy, prevents CRUD abnormalities, and assures data integrity. Using foreign keys in the "Profiles" and "Recommended Source" tables keeps important connections alive while letting each table keep its own data.

2. Entity Relationship Diagram (ERD)

2.1 Detailed ERD of the System

For the Mariata Homes system, the Entity Relationship Diagram (ERD) shows how the entities Users, Profiles, and Recommended Source are connected to each other. According to Chen's (1976) ideas, this ERD follows the rules for showing data connections in a clear and organized way. Chen came up with the idea of ERDs.

Here is the Entity Relationship Diagram for this project

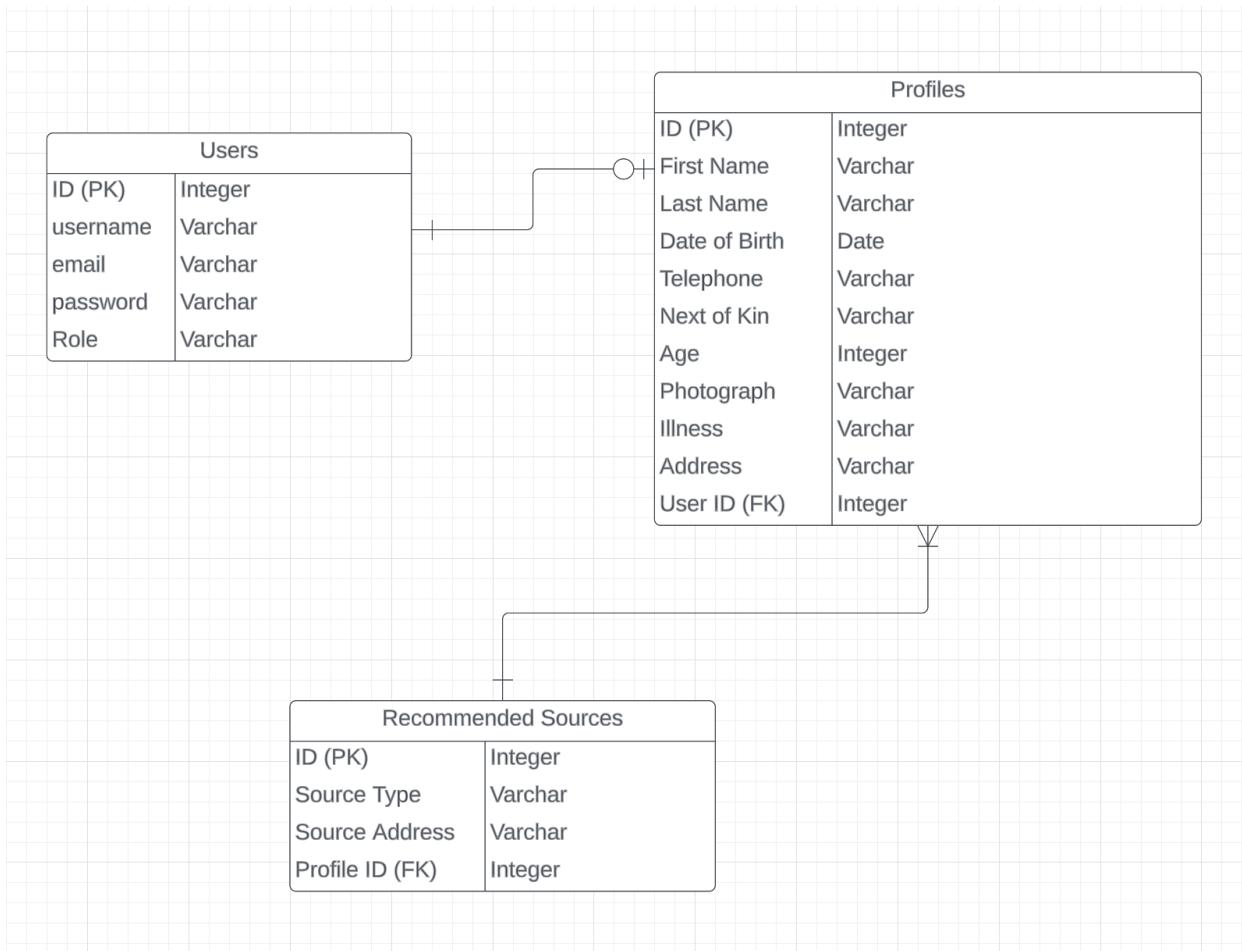


Figure 1: Entity Relationship Diagram

Each User is linked to a Profile which is a One to Zero or One relationship since the client will have a profile but admin does not need to have a profile. Each Recommended Source is linked to exactly one Profile, reflecting a one-to-many relationship as per ERD standards (Teorey et al., 2011). Each User's Profile will have one recommended source but one recommended source can have more than one user's profile.

Explanation of Entities, Attributes, and Relationships

- **Users:** This entity, as per the ERD model, comprises essential user information (Chen, 1976).
- **Profiles:** Reflects detailed user data, linked to the Users entity, as per ERD modeling practices (Teorey et al., 2011).
- **Recommended Sources:** Captures referral sources, a key aspect in understanding user backgrounds.
- **Relationships:**
 - A User may have zero or one Profile. This optional relationship is a fundamental ERD concept (Chen, 1976).
 - Each Recommended Source is linked to exactly one profile, reflecting a one-to-many relationship as per ERD standards (Teorey et al., 2011).

3. Unified Modeling Language (UML) Class Diagram

UML Diagram with Classes, Attributes, Data Types, and Methods

The Mariata Homes system's Unified Modeling Language (UML) Class Diagram shows the system's organization, including its classes, attributes, and methods, using a picture. It was made following the rules in UML 2.0 (Object Management Group, 2017). The diagram is made up of the following classes, which are derived from the identified entities:

Class: User:

Attributes:

- ID (Integer, Private)
- Username (String, Private)
- Email (String, Private)
- Password (String, Private)
- UserRole (String, Private)

Methods:

- login() (Public)
- register() (Public)
- changePassword() (Public)
- deleteUser() (Public)

Class: Profile

Attributes:

- ID (Integer, Private)
- FirstName (String, Private)
- LastName (String, Private)
- DateOfBirth (Date, Private)
- Telephone (String, Private)
- NextOfKin (String, Private)
- Age (Integer, Private)
- PassportPhoto (String, Private)
- Illness (String, Private)
- LastResidenceAddress (String, Private)
- UserID (Integer, Private)

Methods:

- createProfile() (Public)
- updateProfile() (Public)
- deleteProfile() (Public)

Class: RecommendedSource:

Attributes:

- ID (Integer, Private)
- SourceType (String, Private)
- SourceAddress (String, Private)
- ProfileID (Integer, Private)

Methods:

- addSource() (Public)
- addAddress() (Public)
- editSource() (Public)
- editAddress() (Public)
- deleteSource() (Public)
- deleteAddress() (Public)

Each class represents a different part of the system, reflecting the database structure described earlier. The attributes are marked as private to keep the data safe, and the methods are marked as public to let users connect with the objects.

Public and Private Methods Indication

- **Public Methods:** These methods are accessible from outside the class and are used to interact with the system. For instance, 'login()' and 'register()' in the User class allow for user authentication and registration.
- **Private Attributes:** The attributes within each class are set to private, adhering to the principle of encapsulation in object-oriented design (Booch et al., 2005).

Here is the UML Class Diagram

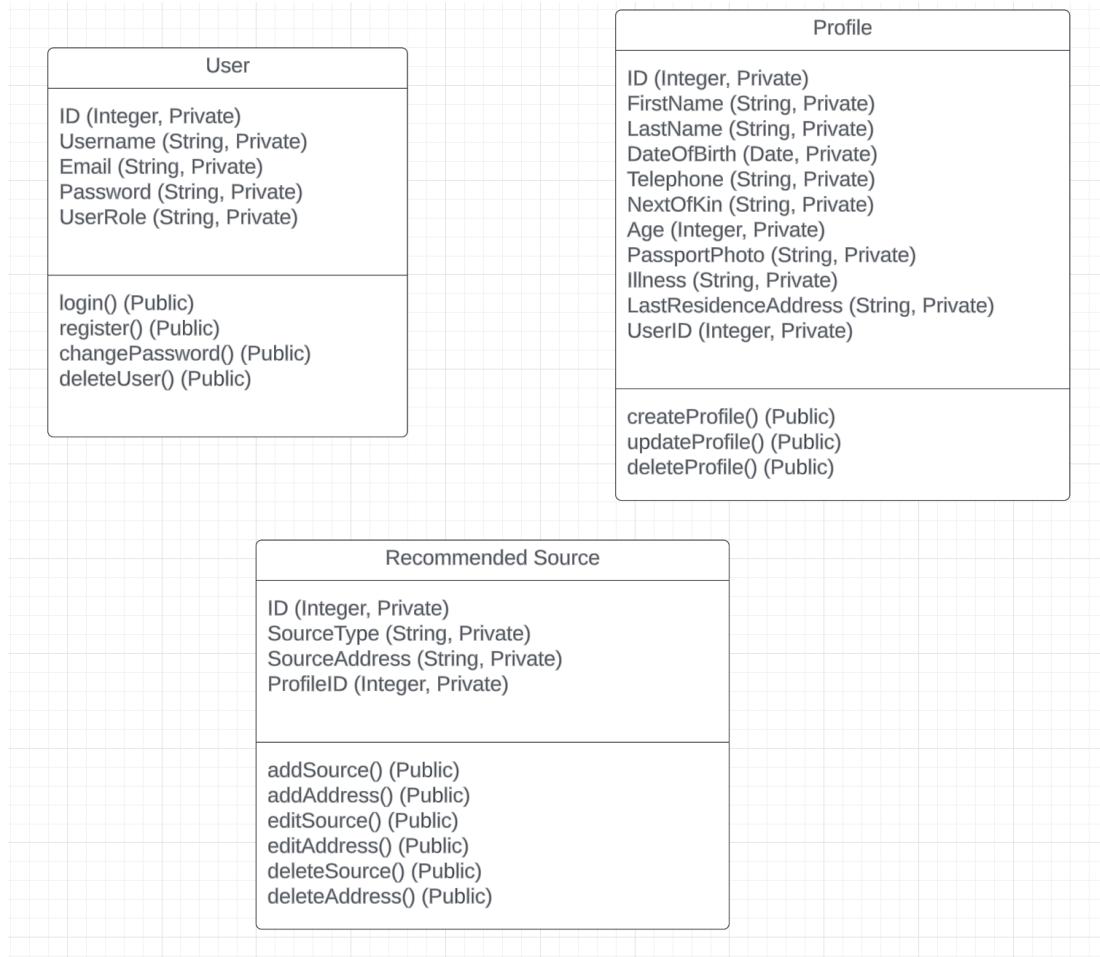


Figure 2: UML Class Diagram

This diagram shows how classes' methods and characteristics link, as proposed by Booch, Rumbaugh, and Jacobson (2005).

4. Use Case Diagrams

Main Use Cases and Actors

Mariota Homes' Use Case Diagram shows how Clients and Admins interact with the system. Jacobson et al. (1992), who invented the use case idea, defined this diagram as a tool for specifying the use of the system.

- **Actors:**
 - **Client:** An individual seeking accommodation assistance.
 - **Admin:** Responsible for system management and client information oversight.
- **Main Use Cases:**
 - **Register:** Allows Clients to register on the system.
 - **Login:** Enables both Clients and Admins to log into the system.
 - **Fill Profile:** Clients provide their detailed information after logging in.
 - **Upload Photo:** Clients can upload their passport photographs.
 - **View Profiles:** Admins can view the list of all client profiles.
 - **CRUD Operations:** Admins perform Create, Read, Update, and Delete operations.
 - **Generate Reports:** Admins generate reports based on client, admin and recommended source data.

Here is the Use Case Diagram

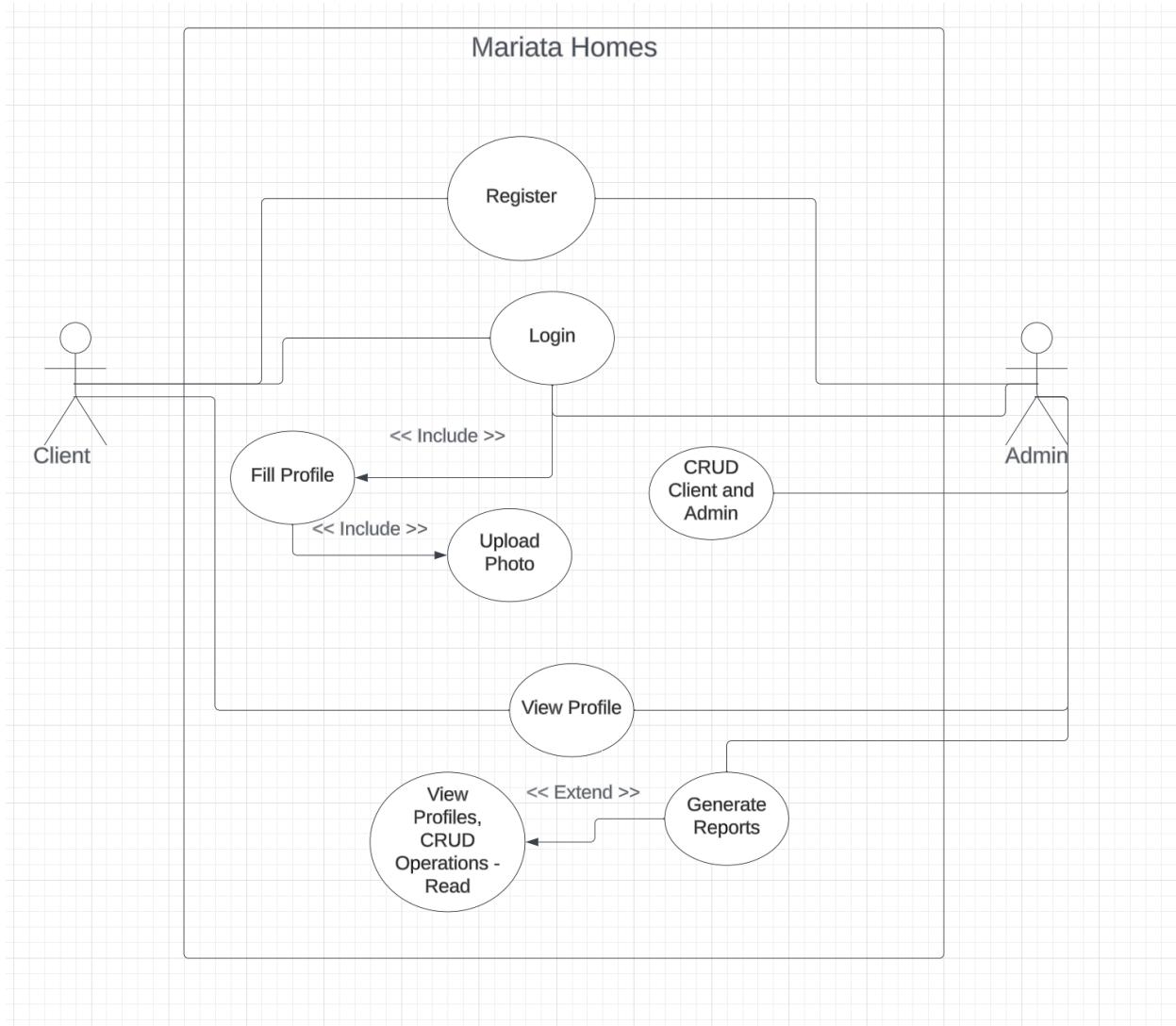


Figure 3: Use Case Diagram

This figure of Use Case Diagram visually depicts these relationships and interactions, aligning with the best practices in UML (Booch, Rumbaugh, and Jacobson, 2005).

5. User Interface (UI) Front-end Forms

Design of UI Forms for Data Collection

The User Interface (UI) tools for the Mariata Homes system were thoughtfully designed to make sure that they collect data quickly and easily. These forms are essential for user interaction with the web application and serve various user journey functions.

- **Registration Form:** This form is essential for new users to create an account. It includes fields for username, email, and password (with a confirmation field for password verification). The registration process is initiated by the “Create Account” button.

A wireframe diagram of a Registration form. The form is titled "Registration" at the top. It contains four input fields: "username", "email", "password", and "confirm password". Below these fields is a large rectangular button labeled "Create Account".

Figure 4: Registration Form

- **Login Form:** Designed for existing users to access their accounts. It requires the user's username and password, with a 'Login' button to submit credentials.

The diagram shows a wireframe representation of a login interface. It consists of a central rectangular box with rounded corners, set against a background of a grid pattern. The top section of the box contains the word "Login" in a bold, sans-serif font. Below this, the word "username" is centered above a horizontal input field, which is represented by a thin rectangular border. Further down, the word "password" is centered above another similar input field. At the bottom of the box, there is a large, prominent rectangular button with the word "Login" centered on it.

Figure 5: Login Form

- **Client Profile Form:** A detailed form used by clients to enter or update their personal information. It includes fields for first name, last name, date of birth, telephone number, next of kin, age, any illnesses, and last residence address, with a 'Submit' button for completing the process.

Create Profile

First Name	<input type="text"/>
Last Name	<input type="text"/>
Date of Birth	<input type="text"/>
Telephone	<input type="text"/>
Next of Kin	<input type="text"/>
Passport Photograph	
Choose File	No File Chosen
Any Illness	
<input type="text"/>	
Next of Kin	
<input type="text"/>	
Last Address	
<input type="text"/>	
Recommended Source	
Choose	<input type="button" value="▼"/>
Recommended Address	
<input type="text"/>	
<input type="button" value="Create Profile"/>	

Figure 6: Form for creating client user

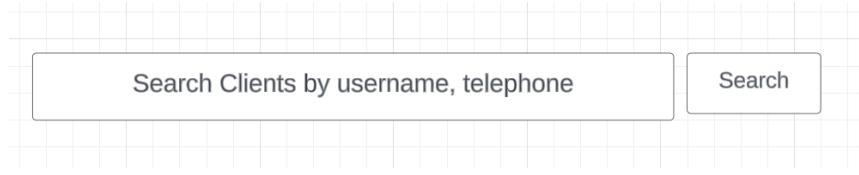
- **Photo Upload Form:** This form allows clients to upload their passport photographs. It is included in the Client Profile Form.
- **Client User Create Form for admin:** Another form only for the admin users to create a client with fields such as username, email, first name, last name, date of birth, telephone number, next of kin, age, any illnesses, and last residence address. This form post request will create a client and their profile entity in one go.

The form is titled "Create User". It includes the following fields:

- username: Text input field
- email: Text input field
- First Name: Text input field
- Last Name: Text input field
- Date of Birth: Text input field
- Telephone: Text input field
- Next of Kin: Text input field
- Passport Photograph: File input field with "Choose File" button and "No File Chosen" label
- Any Illness: Text input field

Figure 7: Client User Create Form for admin

- **Search form:** A specialized interface for searching tables like clients, admins, and recommended sources entities.



A screenshot of a search form interface. It features a search input field with the placeholder text "Search Clients by username, telephone" and a "Search" button to its right. The entire form is set against a background of light gray horizontal grid lines.

Figure 8: Search Form

These figures demonstrate online application forms. By highlighting important parts of each form, like input fields, buttons, and options, it follows the best practices in user interface design (Nielsen, 1993; Shneiderman, 1998).

Part 2: Systems Development

Mariata Homes' web-based CRUD system used multiple front-end and back-end technologies. This method produced a reliable, scalable, and user-friendly app.

Technologies and Tools Used

Backend Development:

- PHP: Chosen for its compatibility with HTML and database support.
- Laravel: Used for MVC support, security, and efficient development.
- Eloquent ORM: Laravel's tool for simplified database interactions.

Database:

- MySQL: Selected for its reliability and PHP compatibility, it handles data storage.

Frontend Development:

- HTML: Forms the core of the user interface.
- JavaScript: Enhances web page interactivity and user experience.
- Bootstrap 4: Used for responsive, mobile-first design.
- Blade Template: Integrates PHP with HTML for efficient page rendering.

Version Control and Collaboration:

- Git: Manages source code changes.
- GitHub: Facilitates codebase backup, sharing, and project management.

The combination of technologies provides performance, security, and usability. The PHP-Laravel backend provides server-side security. Git and GitHub improve coding collaboration. HTML, JavaScript, and Bootstrap 4 create a responsive frontend. Using Eloquent ORM with MySQL improves database interactions, helping the project succeed.

2(a) Registration of User and Admin

Front-End

After both the user and the admin have registered, a success message flashes on the screen. This leads people to the profile creation form. Admins are redirected to their dashboard.

The screenshot shows a web application interface. At the top, there is a pink header bar. Below it, a green success message box contains the text "You have been registered and login successfully.". The main content area is a white box titled "Create Your Profile" with the sub-instruction "Enter your personal details to create a profile". It contains several input fields: "First Name" (empty), "Last Name" (empty), "Date of Birth" (empty), "dd/mm/yyyy" (placeholder text) with a small calendar icon, "Telephone" (empty), "Next of Kin" (empty), "Passport Photograph" (empty), and a file upload field labeled "Choose file" with the sub-label "No file chosen". There is also a text area for "Any Illness" which is currently empty.

Figure 9: User created successfully



admin [See Profile](#)

Clients: 3 Admins: 2 Recommendation Sources: 3

[Add Admin](#) [Download Admins Report](#)

Search Admin by username [Search](#)

admin created at 11/29/2023	
sohel473 created at 11/29/2023	

Figure 10: Admin created successfully

Back-End

It checks the validation of the laravel controller, and creates the user entity in the database. Also passwords are stored in Hash encrypted way.

id	username	email	password	role	created_at	updated_at
4	sohel	sohel@gmail.com	\$2y\$12\$9qiA1GXBT0PzLLa4Oqo9BuPrH2pJ3OnpRdJ.BEsyFtHsQzOs... 5	client	2023-11-29 15:31:04	2023-11-29 15:31:04
5	admin	HULL	\$2y\$12\$ZN9JvJkVD8R7ALjvL4.1.vNWpMC7CH.8wk4QvpoVfpraWk6...	admin	2023-11-29 15:31:04	HULL
6	alyss	alyss@wonderland.com	\$2y\$12\$eQhsC.n0iUkv/yBNWoFeZSM2gpWN5/zWq9f2idaW5J.5l5/vWkq 7	client	2023-11-29 19:00:07	2023-11-29 19:00:07
7	roman	HULL	\$2y\$12\$0VzgF2FVwom7luA3Sj5.Td3igIVSAZ2j3NIVTSPqCpg59ATvB2 8	client	2023-11-29 19:02:32	2023-11-29 19:02:32
8	sabeel	HULL	\$2y\$12\$4c1dit12itvH3lVKfgzT.IODwyWEGL3AtLzV9aobhRODO3/EbKW 9	client	2023-11-29 19:25:54	2023-11-29 19:25:54
HULL	HULL	HULL	HULL	HULL	HULL	HULL

Figure 11: Created User in database

id	username	email	password	role	created_at	updated_at
4	sohel	sohel@gmail.com	\$2y\$12\$9qiA1GXBT0PzLLa4Oqo9BuPrH2pJ3OnpRdJ.BEsyFtHsQzOs... 5	client	2023-11-29 15:31:04	2023-11-29 15:31:04
5	admin	HULL	\$2y\$12\$ZN9JvJkVD8R7ALjvL4.1.vNWpMC7CH.8wk4QvpoVfpraWk6...	admin	2023-11-29 15:31:04	HULL
6	alyss	alyss@wonderland.com	\$2y\$12\$eQhsC.n0iUkv/yBNWoFeZSM2gpWN5/zWq9f2idaW5J.5l5/vWkq 7	client	2023-11-29 19:00:07	2023-11-29 19:00:07
7	roman	HULL	\$2y\$12\$0VzgF2FVwom7luA3Sj5.Td3igIVSAZ2j3NIVTSPqCpg59ATvB2 8	client	2023-11-29 19:02:32	2023-11-29 19:02:32
8	sabeel	HULL	\$2y\$12\$4c1dit12itvH3lVKfgzT.IODwyWEGL3AtLzV9aobhRODO3/EbKW 9	client	2023-11-29 19:25:54	2023-11-29 19:25:54
HULL	HULL	HULL	HULL	HULL	HULL	HULL

Figure 12: Created Admin in database

2(b) Input User Details and Information

Front-End

In order to add users as well as admin, there is a registration form for adding both user (client) and admin. It also validates fields such as checking empty value, min passwords length, checking both password fields, and password type input etc.

Create an Account

Enter your personal details to create account

Username

Email

Password

The password field confirmation does not match.

Confirm Password

Create Account

Already have an account? [Log in](#)

Figure 13: User Password confirms

Create an Account

Enter your personal details to create account

Username

Email

 ! Please fill in this field.

Password

The password field confirmation does not match.

Confirm Password

Create Account

Already have an account? [Log in](#)

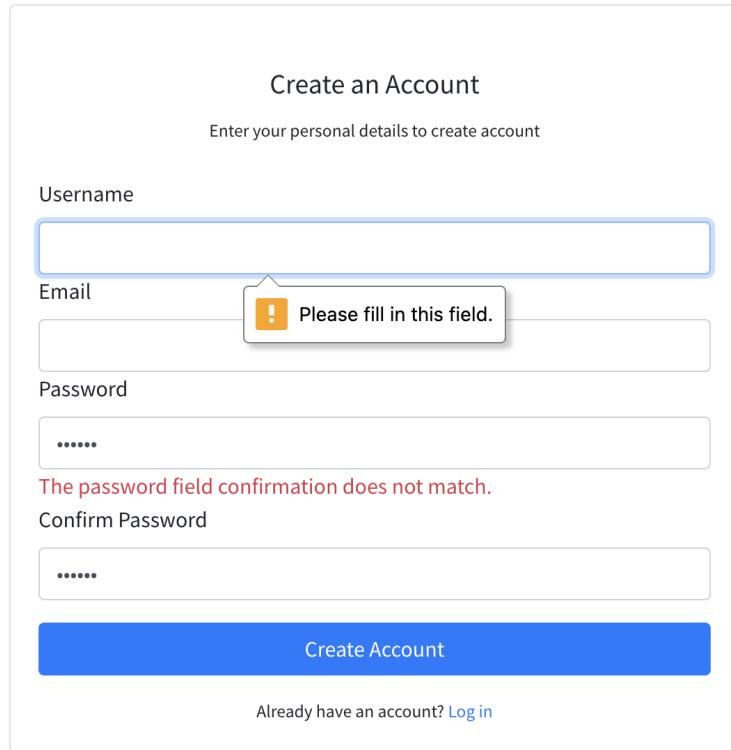


Figure 14: Check empty input field for user

Create an Account

Enter your personal details to create account

Username

 sabeel

Email

Password

The password field must be at least 6 characters.

Confirm Password

Create Account

Already have an account? [Log in](#)

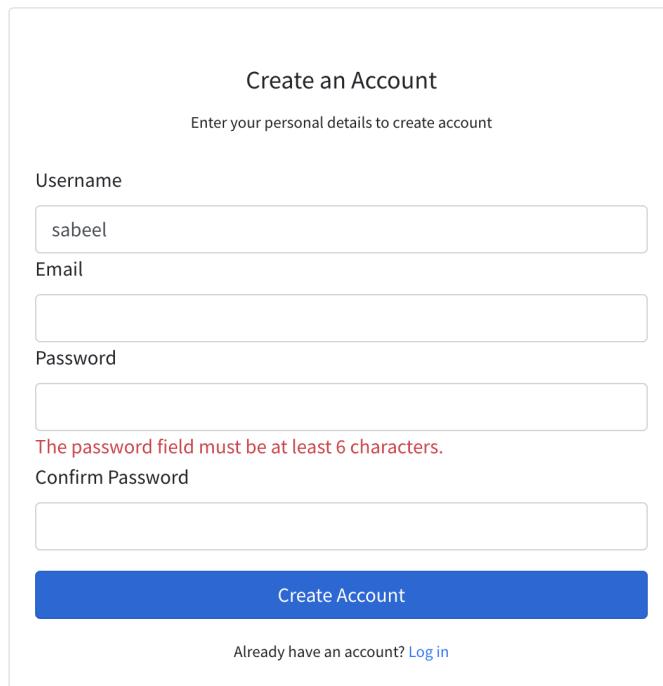


Figure 15: Password length check for user registration

Create Admin User

Enter details to create a new admin user

Username

Email

Password

The password field must be at least 5 characters.

Confirm Password

Create Admin User

Figure 16: Password length check for admin registration

Create Admin User

Enter details to create a new admin user

Username

Email

Please fill in this field.

Password

Confirm Password

Create Admin User

Figure 17: Check empty input field for admin

Back-End

The laravel controller lets you check validation with php instead of html.

```
public function register(Request $request) {
    $request->validate([
        'username' => ['required', 'min:3', 'max:255', Rule::unique('users', 'username')],
        'email' => ['max:255'],
        'password' => ['required', 'min:6', 'max:255', 'confirmed'],
    ]);

    $user = User::create([
        'username' => $request->username,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);

    Auth::login($user);

    session()->flash('success', 'You have been registered and login successfully.');

    return redirect('/create_profile');
}
```

Figure 18: checking inputs validation

2(c) Admin User's Login

Front-End

In both admin and user login form, if it matches the credential, it will redirect you to their profile page, if not then it will show you error message in flash message

The provided credentials do not match our records.

Login to Your Account
Enter your username & password to login

Username

Password

Don't have an account? [Create an account](#)

Figure 19: Showing error in frontend

You have been logged in successfully.

Admin Profile

Username: sohel473

Email: Email not set

created at: 29-11-2023

Edit Profile

Figure 20: Redirects admin to their profile page with flash message

You have been logged in successfully.

Passport Photograph

Full Name Abdullah Al Sohel

Date of Birth 12-08-1995

Telephone 07594792783

Next of Kin Tarek

Age 28

Any Illness

Last Residence Address 14, Wentworth Road

Recommended Source immigration

Type

Recommended Source Address Kensington Immigration Consultancy

Figure 21: Redirects client to their profile page with flash message

Back-End

In the backend, the session with the token is used to check if the user is authenticated.

```

public function login(Request $request) {
    $request->validate([
        'username' => 'required',
        'password' => 'required',
    ]);

    $user = User::where('username', $request->username)->first();

    if ($user && Hash::check($request->password, $user->password)) {
        // Log the user in
        Auth::login($user);

        $request->session()->regenerate();

        session()->flash('success', 'You have been logged in successfully.');
        return redirect('/profile');
    }

    session()->flash('failure', 'The provided credentials do not match our records.');

    return back()->withInput($request->only('username'));
}

```

Figure 22: login checking in admin controller

2(d) Admin Users Are Able to Perform CRUD Operations on All Users

Front-End:

According to coursework's requirements, admin should be able to perform CRUD operation on all types of users be it clients or admin.

The screenshot shows an admin dashboard interface. At the top, it displays the user 'admin' and a 'See Profile' button. Below this, there are statistics: 'Clients: 3' and 'Admins: 2'. To the right of these, a red box highlights a button labeled 'For Read, Update and Delete'. Below the stats, there is a search bar with a placeholder 'For Create Operation' and a 'Search' button. The main area lists three user entries, each with a red circle around its edit and delete icons. The entries are: 'sohel created at 11/29/2023', 'alyss created at 11/29/2023', and 'roman created at 11/29/2023'.

Figure 23: Crud operation for users in admin dashboard

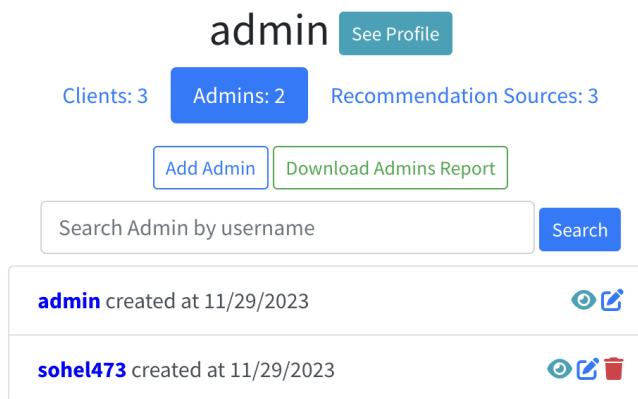


Figure 24: Crud operation for admin

The form that is used for the admin is different from the one used for the user. It lets you add a username, email address, and password, as well as personal information like names, dates of birth, locations, and so on, when you add a user.

Please fill in this field.

Create User

Enter details to create a new user

Username

Email

Password

Confirm Password

First Name

Last Name

Date of Birth
 dd/mm/yyyy

Telephone

Next of Kin

Passport Photograph

Figure 25: User creation form

For admin, it just requires the username, email and password since admin doesn't need to have a profile.

Create Admin User

Enter details to create a new admin user

Username

Email

Password

Confirm Password
 ! Please fill in this field.

Create Admin User

Figure 26: Admin creation form

The view page of client and admin is different. For user it will show you all the necessary information and for admin it will show you only username, email and registration date.

Mariata Homesadmin[Admin Dashboard](#)[Sign Out](#)



A placeholder image for a passport photograph, showing a green and blue design with a small photo area.

Full Name	Abdullah Al Sohel
Date of Birth	12-08-1995
Telephone	07594792783
Next of Kin	Tarek
Age	28
Any Illness	
Last Residence Address	14, Wentworth Road
Recommended Source	immigration
Type	Kensington Immigration Consultancy
Address	Kensington Immigration Consultancy

Figure 27: User page from admin dashboard

The screenshot shows a web browser window with the URL 127.0.0.1:8000/admin_user/9. The title bar says 'Mariata Homes'. The top right has 'admin' and 'Admin Dashboard' buttons, and a 'Sign Out' link. The main content area is titled 'Admin Profile' and contains the following information:

- Username: sohel473
- Email: Email not set
- created at: 29-11-2023

At the bottom is a yellow 'Edit Profile' button.

Figure 28: Admin profile

For updating a user, it runs a query to get all the saved information from the database and displays on the form along with the photograph of the passport.

The screenshot shows a web browser window with the URL 127.0.0.1:8000/user/4/edit. The title bar says 'Edit User'. The main content area is titled 'Edit User' and has the sub-instruction 'Update the details of the user'. It contains the following fields:

- Username: sohel
- Email: sohel@gmail.com
- Password (empty field)
- Confirm Password (empty field)
- First Name: Abdullah
- Last Name: Al Sohel
- Date of Birth: 12/08/1995
- Telephone: 07594792783
- Next of Kin: Tarek
- Passport Photograph: A placeholder image of a passport card is shown, with a 'Choose file' button below it.

Figure 29: Client User edit form

Same goes for the admin as well.

Edit Admin User

Update the details of the admin user

Username

Email

Password

Confirm Password

Update Admin User

Figure 30: Admin edit form

Passport Photograph 	Full Name Abdullah Al Sohel Date of Birth 12-08-1995 Telephone 07594792783 Next of Kin Tarek Age 28 Any Illness Last Residence Address 14, Wentworth Road Recommended Source Type immigration Recommended Source Address Kensington Immigration Consultancy
---	--

Figure 31: Before updating the user in user profile

User updated successfully.

Passport Photograph	Full Name	Abdullah Al Sohel
	Date of Birth	12-08-1995
	Telephone	07594792783
	Next of Kin	Tarek
	Age	28
	Any Illness	Asthma
	Last Residence Address	14, Wentworth Road
	Recommended Source Type	police
	Recommended Source Address	London Bridge Police Office

Figure 32: After updating the user

Clicking the trash icon prompts a confirmation alert. It will remove the clients, admins, sources if the answer is ok.

127.0.0.1:8000 says

Are you sure you want to delete this client?

Cancel OK

admin Admin Dashboard

admin See Profile

Clients: 3 Admins: 2 Recommendation Sources: 3

Add Client Download Clients Report

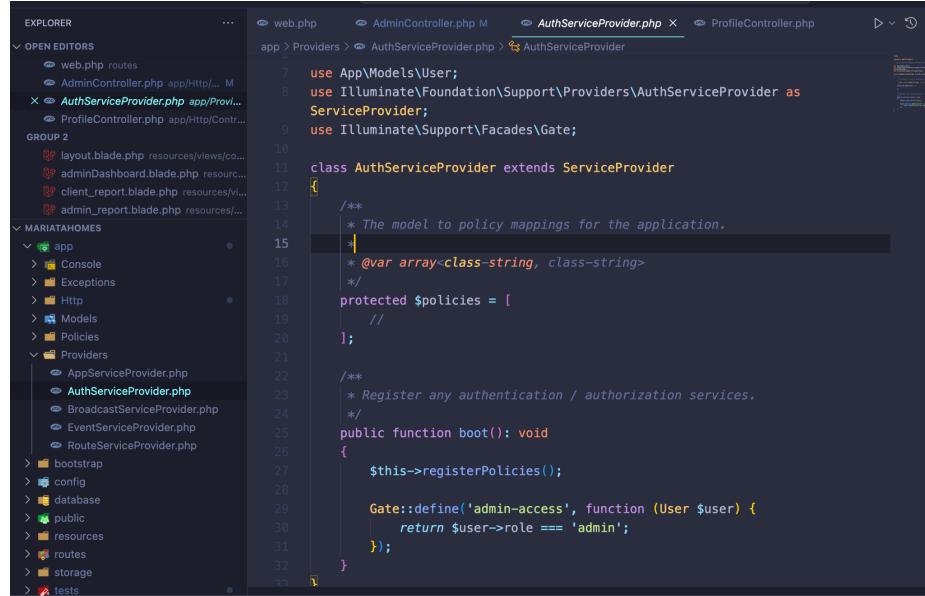
Search Clients by username, telephone Search Delete

sohel created at 11/29/2023		
alyss created at 11/29/2023		
roman created at 11/29/2023		

Figure 33: user delete confirmation

Back-End:

In order to make sure only admin can perform CRUD operation, middleware has been used in routes. In laravel, you will have to create Gate Facade and there it will check if the user role is admin. Then in routes, you can specify which operations can be performed by the admin.



```
use App\Models\User;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
use Illuminate\Support\Facades\Gate;

class AuthServiceProvider extends ServiceProvider
{
    /**
     * The model to policy mappings for the application.
     *
     * @var array<class-string, class-string>
     */
    protected $policies = [
        //
    ];

    /**
     * Register any authentication / authorization services.
     */
    public function boot(): void
    {
        $this->registerPolicies();

        Gate::define('admin-access', function (User $user) {
            return $user->role === 'admin';
        });
    }
}
```

Figure 34: Gate facade in app/Providers/AuthService



```
// admin routes
Route::middleware('can:admin-access')->group(function () {
    // get routes
    // admin routes
    Route::get('/admin', [AdminController::class, 'showAdminPage']);

    Route::get('/admin/download_clients_report', [AdminController::class, 'downloadClientsReport'])->name('admin.download_clients_report');
    Route::get('/admin/download_admin_users_report', [AdminController::class, 'downloadAdminUsersReport'])->name('admin.download_admins_report');
    Route::get('/admin/download_recommended_sources_report', [AdminController::class, 'downloadRecommendedSourcesReport'])->name('admin.download_sources_report');

    // users routes
    Route::get('/create_user', [AdminController::class, 'showCreateUserPage']);
    Route::get('/user/{user}', [AdminController::class, 'showUserPage']);
    Route::get('/user/{user}/edit', [AdminController::class, 'showEditUserPage']);
    // admin users routes
    Route::get('/create_admin_user', [AdminController::class, 'showCreateAdminUserPage']);
    Route::get('/admin_user/{admin_user}', [AdminController::class, 'showAdminUserPage']);
    Route::get('/admin_user/{admin_user}/edit', [AdminController::class, 'showEditAdminUserPage']);
    // recommended sources routes
    Route::get('/create_recommended_source', [AdminController::class, 'showCreateRecommendedSourcePage']);
    Route::get('/recommended_source/{recommended_source}', [AdminController::class, 'showEditRecommendedSourcePage']);
    Route::get('/recommended_source/{recommended_source}/edit', [AdminController::class, 'showEditRecommendedSourcePage']);
});
```

Figure 35: Specifying all the routes for CRUD of all type of users that should be performed with admin access

<u>id</u>	<u>user_id</u>	<u>recommended_source_id</u>	<u>first_name</u>	<u>last_name</u>	<u>date_of_birth</u>	<u>telephone</u>	<u>next_of_kin</u>	<u>passport_photograph</u>	<u>any_illness</u>	<u>last_residence_address</u>	<u>updated_at</u>
1	4	1	Abdullah	Al Sohel	1995-08-12	07594792783	Tarek	4_sohel_1701294072.jpg	NULL	14, Wentworth Road	2023-11-29 21:44:27
2	6	1	Alys	Soe	2000-06-07	01303912124	NULL	6_alys_1701284407.jpg	NULL	Fiat: 2A, House: 752, Roa...	2023-11-29 19:00:07
3	7	2	ROman	Ahmed	2021-12-12	21312321312	Sohel	7_roman_1701284647....	NULL	NULL	2023-11-29 19:04:07

Figure 36: Before updating in database

<u>id</u>	<u>user_id</u>	<u>recommended_source_id</u>	<u>first_name</u>	<u>last_name</u>	<u>date_of_birth</u>	<u>telephone</u>	<u>next_of_kin</u>	<u>passport_photograph</u>	<u>any_illness</u>	<u>last_residence_address</u>	<u>updated_at</u>
1	4	4	Abdullah	Al Sohel	1995-08-12	07594792783	Tarek	4_sohel_1701294072.jpg	Cough	14, Wentworth Road, E12 5BD	2023-11-29 21:45:33
2	6	1	Alys	Soe	2000-06-07	01303912124	NULL	6_alys_1701284407.jpg	NULL	Fiat: 2A, House: 752, Road: 16,...	2023-11-29 19:00:07
3	7	2	ROman	Ahmed	2021-12-12	21312321312	Sohel	7_roman_1701284647....	NULL	NULL	2023-11-29 19:04:07

Figure 37: After updating in database

2(e) Other Functionalities (Search Users, Generate Reports)

Front-End:

For enhancing user quality experience, two extra functionalities have been added to Mariata Homes project. All types of Users like Clients as well admin, all the recommended sources can be searched by their respective attributes.

The screenshot shows the Admin Dashboard at http://127.0.0.1:8000/admin?client_search=sohel. The search bar contains 'sohel'. Below it, a table lists a single result: 'sohel created at 11/29/2023'. There are edit and delete icons next to the result. The dashboard also shows statistics: Clients: 1, Admins: 2, Recommendation Sources: 3, and buttons for Add Client, Download Clients Report, and Search.

Figure 38: Client user search

The screenshot shows the Admin Dashboard at http://127.0.0.1:8000/admin?admin_search=so. The search bar contains 'so'. Below it, a table lists a single result: 'sohel473 created at 11/29/2023'. There are edit and delete icons next to the result. The dashboard also shows statistics: Clients: 3, Admins: 1, Recommendation Sources: 3, and buttons for Add Admin, Download Admins Report, and Search.

Figure 39: Admin user search



Figure 40: Before searching recommended source



Figure 41: After searching source

Also admin can generate reports for the client users, admin users and recommended sources that are stored in the database

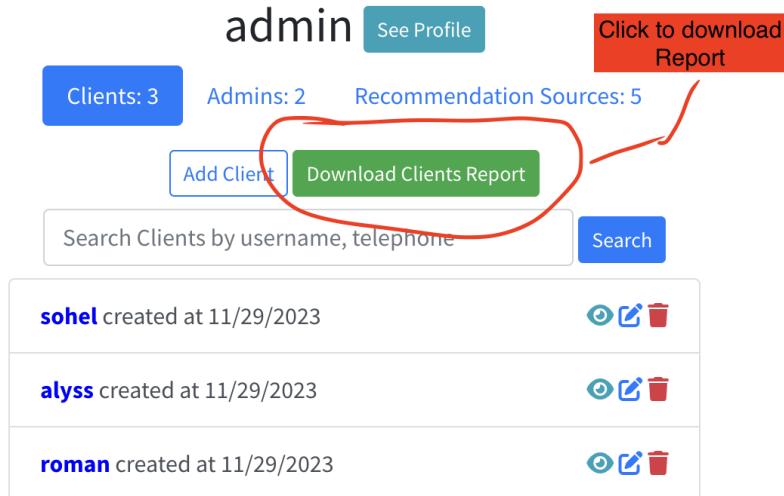


Figure 42: Report download button

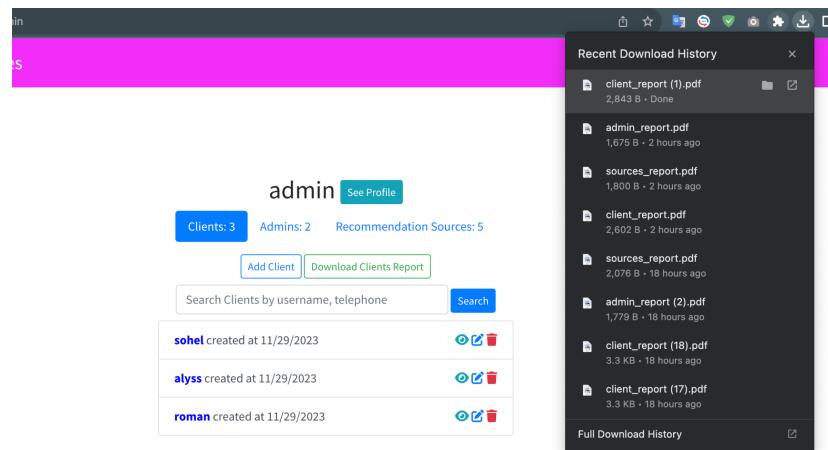


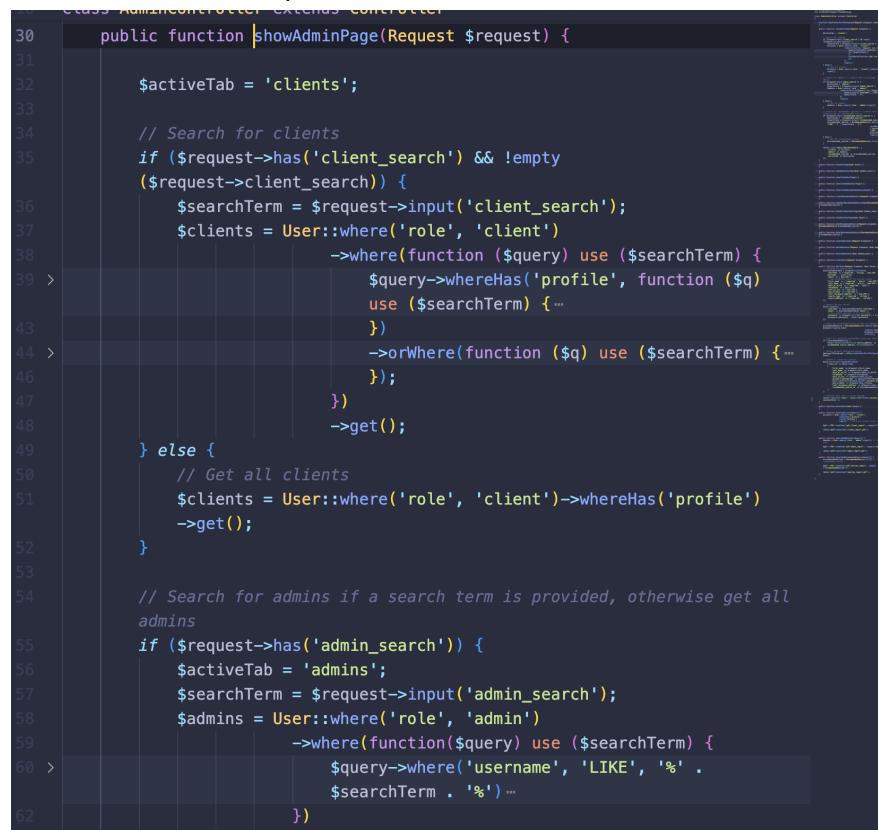
Figure 43: Report downloaded

Username	Email	First Name	Last Name	Date of Birth	Telephone	Next of Kin	Illness	Last Residence Address
sohel	sohel@gmail.com	Abdullah	Al Sohel	12-Aug-1995	07594792783	Tarek	N/A	14, Wentworth Road
alyss	alys@wonderland.com	Alys	Soe	07-Jun-2000	01303912124	N/A	N/A	Flat: 2A, House: 752, Road: 16, Block: K, Bashundhara R/A
roman	N/A	ROman	Ahmed	12-Dec-2021	21312321312	Sohel	N/A	N/A

Figure 44: Report in pdf

Back-End:

Behind the curtain, it checks if the request from search input has the specific type such as is it client search or admin search or recommendation search. And then query to to database and pass it to the admin dashboard template



```
30 class AdminController extends Controller
31 {
32     public function showAdminPage(Request $request) {
33
34         $activeTab = 'clients';
35
36         // Search for clients
37         if ($request->has('client_search') && !empty
38             ($request->client_search)) {
39             $searchTerm = $request->input('client_search');
40             $clients = User::where('role', 'client')
41                             ->where(function ($query) use ($searchTerm) {
42                                 $query->whereHas('profile', function ($q)
43                                     use ($searchTerm) { ...
44                                     })
45                                     ->orWhere(function ($q) use ($searchTerm) { ...
46                                     })
47                                     });
48             $clients->get();
49         } else {
50             // Get all clients
51             $clients = User::where('role', 'client')->whereHas('profile')
52                             ->get();
53         }
54
55         // Search for admins if a search term is provided, otherwise get all
56         // admins
57         if ($request->has('admin_search')) {
58             $activeTab = 'admins';
59             $searchTerm = $request->input('admin_search');
60             $admins = User::where('role', 'admin')
61                             ->where(function($query) use ($searchTerm) {
62                                 $query->where('username', 'LIKE', '%' .
63                                     $searchTerm . '%') ...
64                             });
65         }
66     }
67 }
```

Figure 45: Search logic in controller

Testing

Testing Methodologies Used

For the Mariata Homes system, a comprehensive testing approach was employed to ensure the application's functionality, reliability, and user experience. The primary methodology used was automated feature testing.

Automated Feature Testing: Utilizing **php artisan test**, total 18 tests were conducted such as all user login, registration, logout, CRUD operation of admin.

```
● sohel@Sohel-MacBook MariataHomes % php artisan test tests/Feature

    PASS Tests\Feature\AdminControllerTest
    ✓ admin can access admin dashboard          0.30s
    ✓ admin can create user                   0.05s
    ✓ admin can download clients report      0.05s
    ✓ admin can edit user                     0.01s
    ✓ admin can delete user                  0.01s
    ✓ admin can create recommended source   0.01s
    ✓ admin can edit recommended source     0.01s
    ✓ admin can delete recommended source   0.01s
    ✓ admin can download admin users report 0.02s
    ✓ admin can download recommended sources report 0.02s

    PASS Tests\Feature\ProfileControllerTest
    ✓ user can view create profile page       0.01s
    ✓ user can create profile                 0.06s
    ✓ user can view profile page             0.02s

    PASS Tests\Feature\UserControllerTest
    ✓ user can view register page           0.01s
    ✓ user can register                    0.01s
    ✓ user can view login page             0.01s
    ✓ user can login                      0.01s
    ✓ user can logout                     0.01s

Tests:  18 passed (62 assertions)
Duration: 0.71s
```

Figure 46: All 18 tests passed.

Conclusion

Summary:

The Mariata Homes web-based CRUD system project used full-spectrum system design and development. Database normalization, design of ERD and UML Class Diagrams, and creation of Use Case Diagrams for user interface clarity were all important tasks. The backend used PHP, Laravel, and MySQL, while the frontend used HTML, JavaScript, and Bootstrap 4, with Git and GitHub version control. Despite hurdles, the development phase was a success, characterized by good testing and problem-solving. In summary, this project achieved its objectives and provided valuable insights into the practical application of system design and development theories.

References

1. Booch, G., Rumbaugh, J. and Jacobson, I., 2005. *The Unified Modeling Language User Guide*. 2nd ed. Addison-Wesley.
2. Chen, P.P., 1976. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1), pp.9-36.
3. Codd, E.F., 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6).
4. Date, C.J., 2004. *An Introduction to Database Systems*. 8th ed. Pearson Education.
5. Elmasri, R. and Navathe, S.B., 2016. *Fundamentals of Database Systems*. 7th ed. Pearson.
6. Jacobson, I., Christerson, M., Jonsson, P. and Overgaard, G., 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley.
7. Nielsen, J., 1993. *Usability Engineering*. Academic Press.
8. Object Management Group, 2017. *Unified Modeling Language (UML)*, Version 2.5.1.
9. Shneiderman, B., 1998. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 3rd ed. Addison-Wesley.
10. Teorey, T.J., Lightstone, S. and Nadeau, T., 2011. *Database Modeling & Design: Logical Design*. 5th ed. Morgan Kaufmann.