

وجب

بني



members

Name	ID
Ghala Albishri (leader)	443007832
Renad Alsaedi	443009572
Asma Al-Asmari	443005870
Razan Alqurashai	443000158
Arwa Asiri	443008129
Atyaf Almatrfi	443000655

content table

CHAPTER 1 : Introduction

1.1 Abstract paragraph	6
1.2 Problem definition.....	7
1.3 Solution.....	8
1.4 Aims and objectives of the project....	9

CHAPTER 2 : System analysis

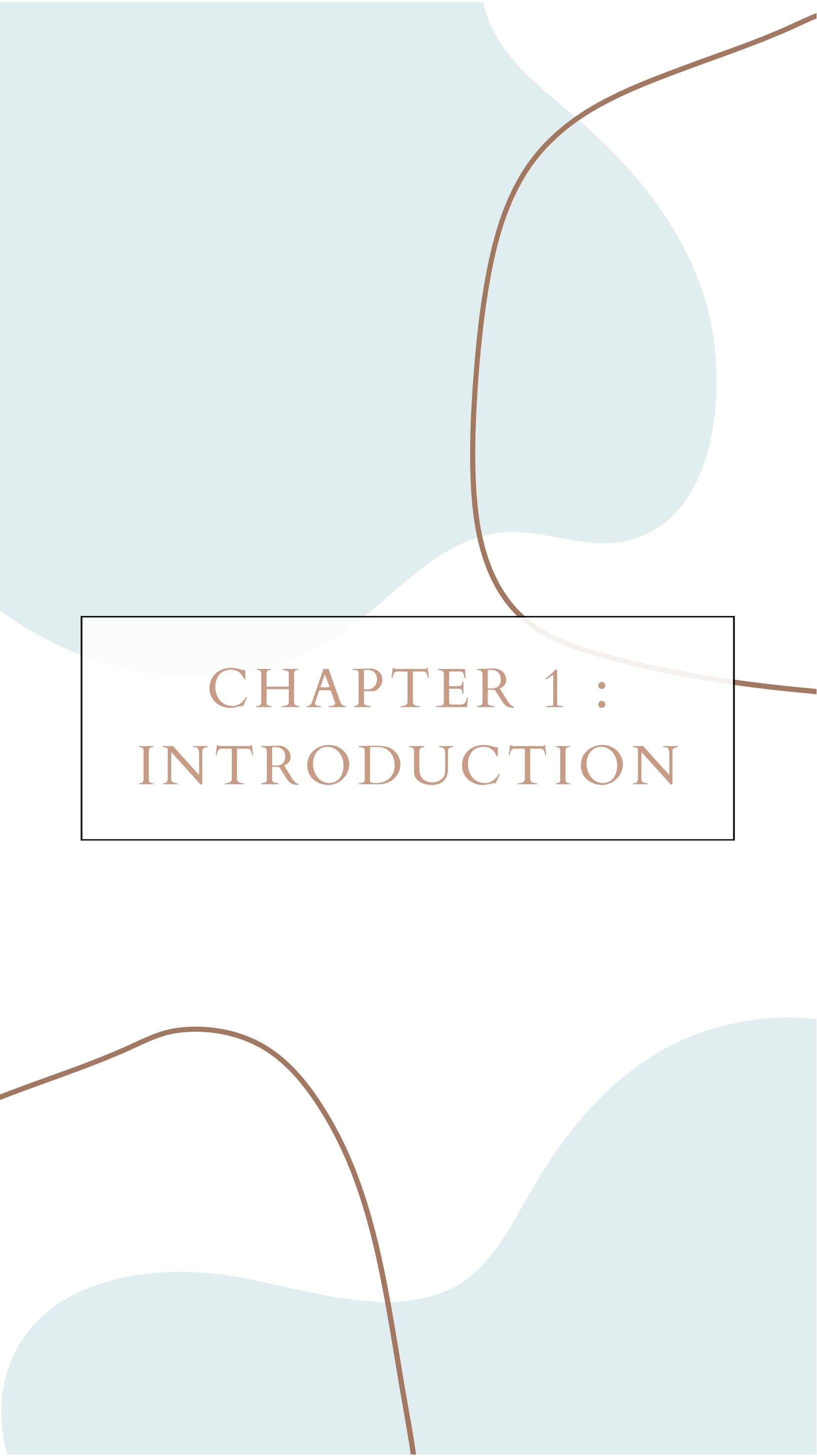
2.1 Services	11
2.2 Requirements specification.....	12
2.2.1 Functional requirements	12
2.2.1 Non-functional requirements..	19

CHAPTER 3 : System design

3.1 Context Diagram.....	20
3.2 DFD	21
3.3 Use Case Diagram	23
3.4 Scenarios.....	24
3.5 Class diagram.....	37
3.6 Activity diagram.....	38
3.7 Sequence diagram.....	46

CHAPTER 4 :Prototype

4.1 GUI.....	54
--------------	----



CHAPTER 1 : INTRODUCTION

1.1

Abstract paragraph

Wajjehni provides a notification service for events and activities at the university so that students will be informed from their first day of everything that is happening at the university.

We were inspired by the idea of the project from the problem that occurs in real life in terms of the lack of knowledge of buildings and facilities in public places, and because our scope is focused on the university, we added a solution to the problem of lack of information and sources for the academic subject.

Wajjehni provides a notification service for events and activities at the university so that students will be informed from their first day of everything that is happening at the university.

1.2

Problem definition

When visiting any facility, many of us face the problem of not knowing the correct route that leads to the desired destination. After studying this issue, it became apparent that female students face this problem on a daily basis when heading to their lecture halls, due to the unavailability of a guide to lead them to the university facilities when needed.

and Given the importance of good academic achievement, some students suffer from a problem in understanding some courses and the lack of a clear and good explanation but because some curricula differ from one university to another, and also the teaching of the subject may differ from one instructor to another, the student should receive an excellent explanation from someone who has studied the same curriculum with the same instructor so that she can she conveys the information well, as some courses do not have sources of explanation and there is difficulty in searching for the appropriate explanation, the feature of academic guidance by volunteers will be an advantage of our application and provide a service to students.

and Given the importance of activities and events The list at the university that helps and contributes to raising students' academic achievement. Because these activities and events are sent via university mail to students, the large number of messages in the mail caused many messages to be ignored by some, which made them struggle to know the available activities and events. It is necessary to provide a way for students to receive notifications that will contribute to their knowledge of these activities, their participation in them, and their benefit from them.

1 . 3

Solution

Our project depends on two types of guidance: guiding students to university buildings and facilities and advising them academically.

In terms of guidance to university facilities, the previous solutions were simple, including the student asking someone to direct them to the specific location or some universities placing a picture of the university map on the walls or the university website

As for academic guidance, many applications and websites help students understand the curriculum Such as the “علم” initiative offered by the Computer Club to help computer students at Umm Al-Qura University.

finally. The solution that we have come up with in our project is to have several volunteers from within the university who volunteer at specific times that they choose to help anyone who needs to get to the classroom or needs an explanation of a specific subject. So, when the user logs into the application, he will see the volunteers available during the current period, so he can contact them to request assistance This helps in providing assistance in the shortest time and in the best possible way.

1.4

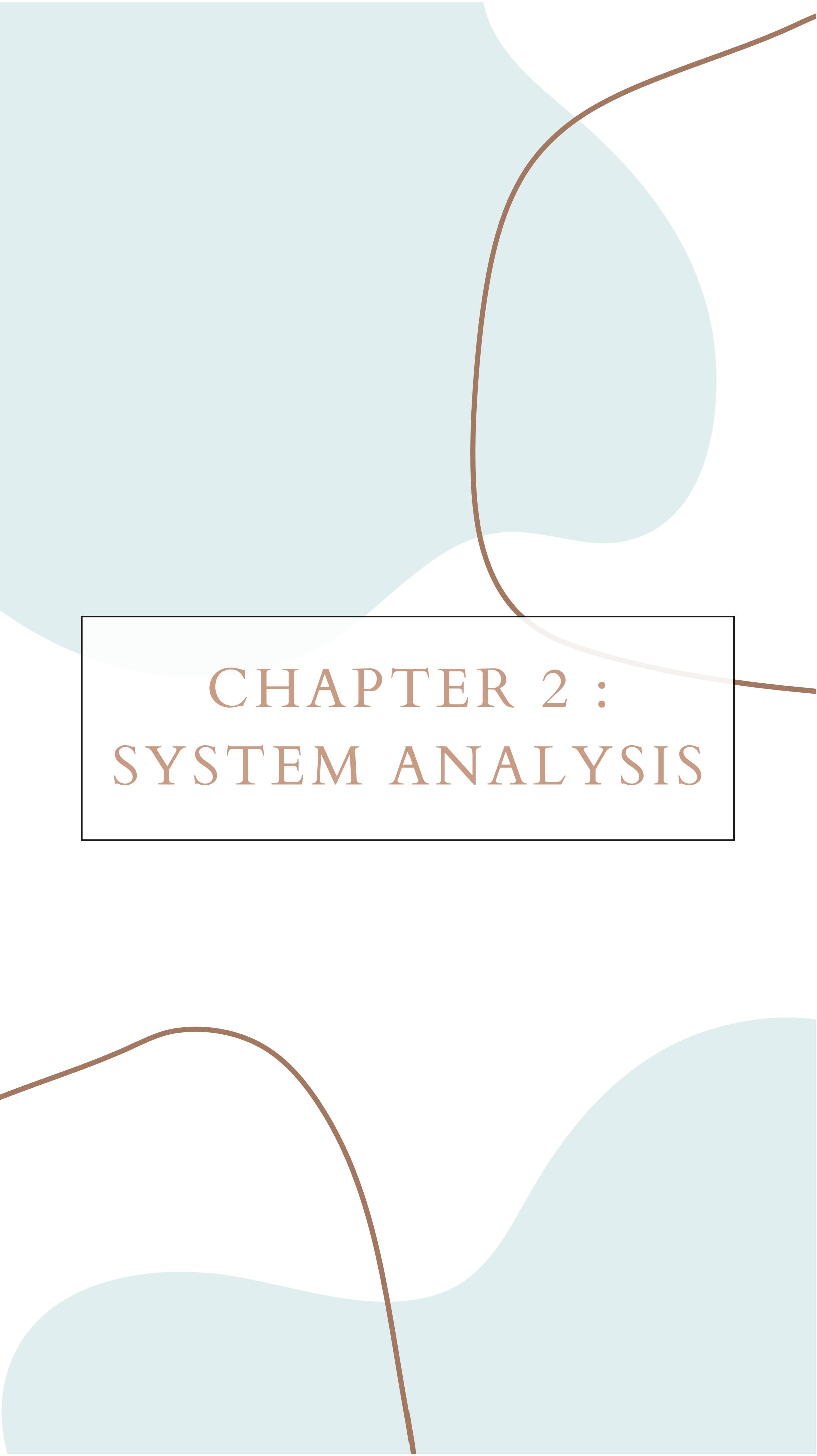
Aims and objectives of the project

1- Enhancing the students' experience on campus and providing them with a sustainable and comfortable environment.

2- Promoting communication and collaboration among students and encouraging their active participation in university life.

3- Providing volunteering opportunities for students.

4- Facilitating the process of guidance and learning in one time and place.



CHAPTER 2 : SYSTEM ANALYSIS

2.1

Services

1-Sign Up or Create an Account Feature

2-Login and Logout

3-View Interactive Map

4-Fill Volunteering Form

5-Filtering

6-Display Volunteer Information

7- Provide Safety Features

8-Rating and giving Feedback System

9- View Campus Events and Notifications

10-Provide Campus Deals and Discounts

2.2.1

Requirements specification Functional requirements

1-Sign up

User requirement:

1-the user shall be able to create a new account.

System requirement:

1.1-the system is able to display sign up form to the user which contains list of fields :

(first name, last name, UQU email, password, confirmation of the password)

1.2-The system is able to send a verification code to the registered email to verify the validity of the email

1.3- the system is able to validate the entered data with these constrains:

- * the email must be unique and there is no other record in the database has the same values.
- * The email should be in UQU database
- * the password should be at least 8 characters that contains letters in uppercase and lowercase and numbers.

1.4- the system is able to display pop-up window to the user with a message if the validation was successful verification. otherwise, it return the user to the form and ask to fill the form again with the specified constraints.

2.2.1

Requirements specification Functional requirements

2-Log in

-User requirement:

2-The user shall be able to log in to her Account.

-System requirements:

2.1-the system is able to display login form to the user which contain these fields: (UQU email and password)

2.2- the system is able to check from the entered data with the database. if there is a match retrieve user information and navigate to the main interface. otherwise, display pop-up message contains (there is no such account, try again or sign up).

3-Fill Volunteering Form

User requirement:

3-The user shall be able to fill out the volunteer form

System requirement:

3.1- The system shall be able to display the ready-made form for inserting the Volunteer data. If the user chooses study guidance the form must contain (the type of guidance, Academic level, Subject, department, and time) and if the user chooses guidance to university facilities the form must contain (Location, time).

3.2- The system shall be able to save data in the database.

3.3- The system shall be able to show the new volunteer in the user interface.

3.4-The system should save the number of volunteer times for each volunteer

2.2.1

Requirements specification Functional requirements

4- View Interactive Map

User requirement:

4-The user shall be able to use the interactive map

System requirements:

4.1-The system should be able to show available volunteers at the specified time along with their current locations

4.2-The system should be able to differentiate between types of volunteering by displaying different colors. pink is for guidance for facilities and blue is for study guidance

4.3-The system should be able to display all of the volunteer's information, including his name, department, email, and rate, when clicked on

4.4-The system should be able to remove the volunteer from the map when the pre-determined volunteer period ends

2.2.1

Requirements specification Functional requirements

5- filtering

User requirement:

5-The user should be able to filter volunteers and services.

System requirement:

5.1- The system should be able to filter to reduce search time.

5. 2- The system should be able to filter based on the type of volunteering (university facilities, academic guidance)and if the user chooses campus events, the filtering should be based on event location and type (educational or entertainment).

6- Display Volunteer Information

User Requirement:

6-The user can see the volunteer information

System Requirements:

6.1The system displays the volunteer's information, including full name, department, specialty, and email and rating

2.2.1

Requirements specification Functional requirements

7-Provide Safety Features

User requirements:

7-The user can use the security button.

System requirements:

7.1-The system is able to display the security form to the user that contains a list of fields (red button, yellow button, green button).

* The user can use the red button if he needs high-level assistance, such as medical assistance.

* The user can use the yellow button if he needs help at an intermediate level, such as getting a phone charger.

*The user can use the green button if he needs low-level assistance, such as getting a pen.

7.2-The system is able to display a pop-up window through which one can press the desired button.

7.3 The system can display a pop-up window to the user with a message in which he writes the type of help he needs.

2.2.1

Requirements specification Functional requirements

8-Rating and giving Feedback System

User requirements

8-The user should be able to view the previous ratings of the volunteers and can add his own rating.

System requirements

8.1 The system will allow users to choose from the ratings list:

- Cooperative
- Brief explanation
- Detailed explanation
 - Respectful
 - Uses various teaching methods.
- Impolite
- Poor interaction
- Inadequate explanation
- Poor ethics
- Takes more time than necessary.

Also, user can add additional comments , then display it in the volunteer's rating page.

2.2.1

Requirements specification

Functional requirements

9-View Campus Events and Notifications

User requirement:

9-The user should be able to receive notifications

System requirement:

9.1-The System shall be able to give notifications about upcoming events, workshops, seminars, and gatherings taking place on the university campus.

10-Provide Campus Deals and Discounts

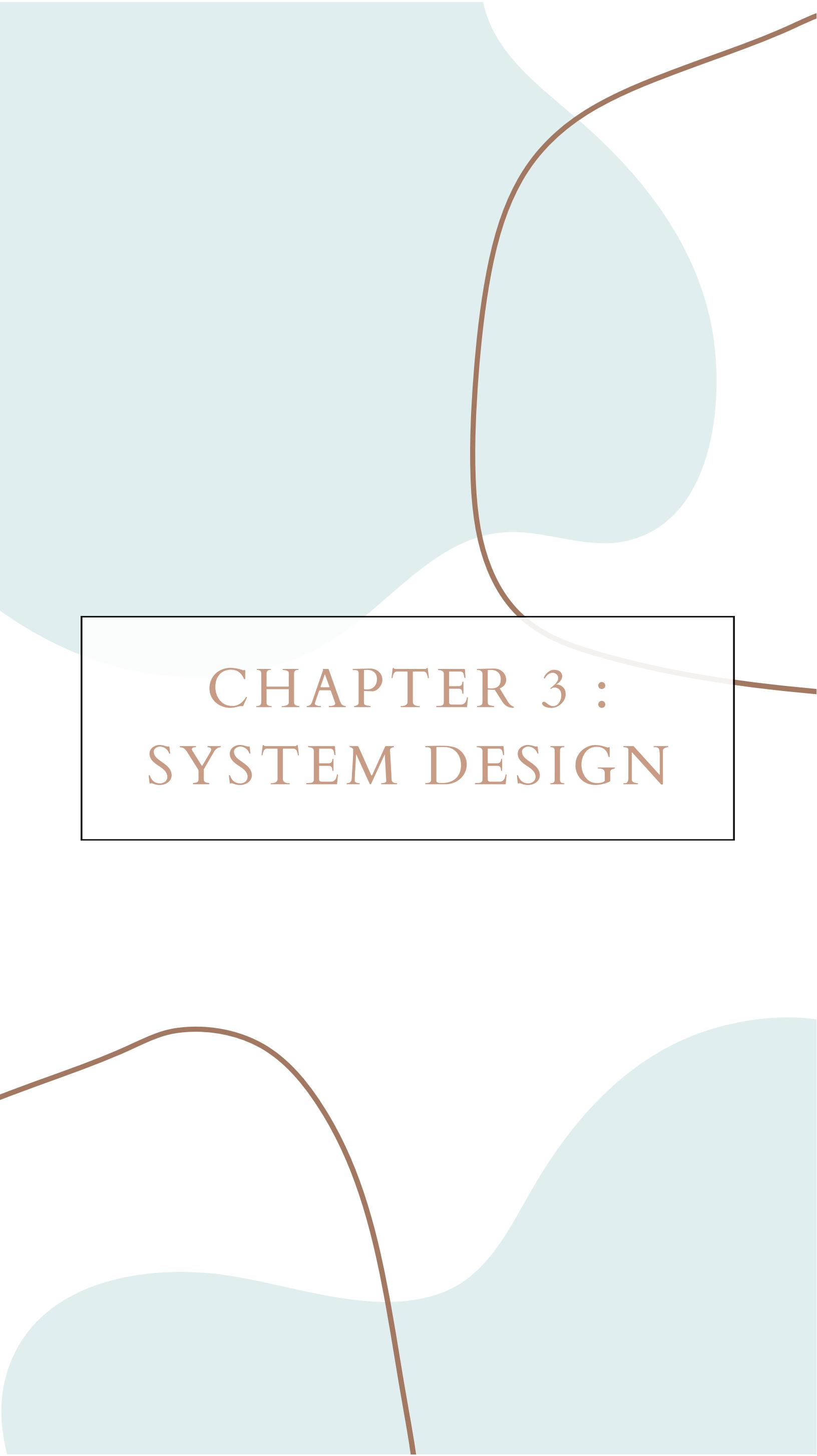
User Requirements:

10-The user must be able to access discounts near or around the university

System Requirements:

10.1-The system must provide students with access to exclusive discounts and offers at the university.

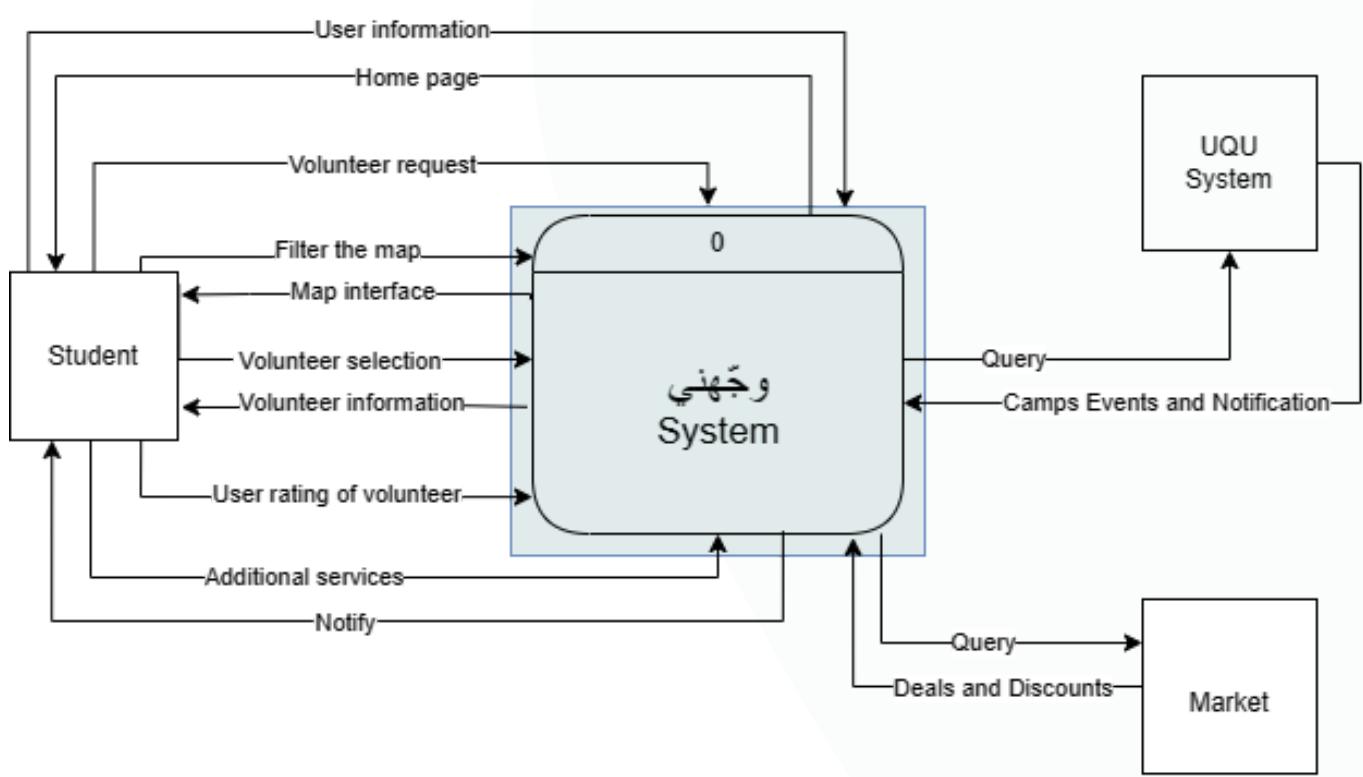
10.2-The system must allow volunteers with high points to benefit from special discounts and offers.



CHAPTER 3 : SYSTEM DESIGN

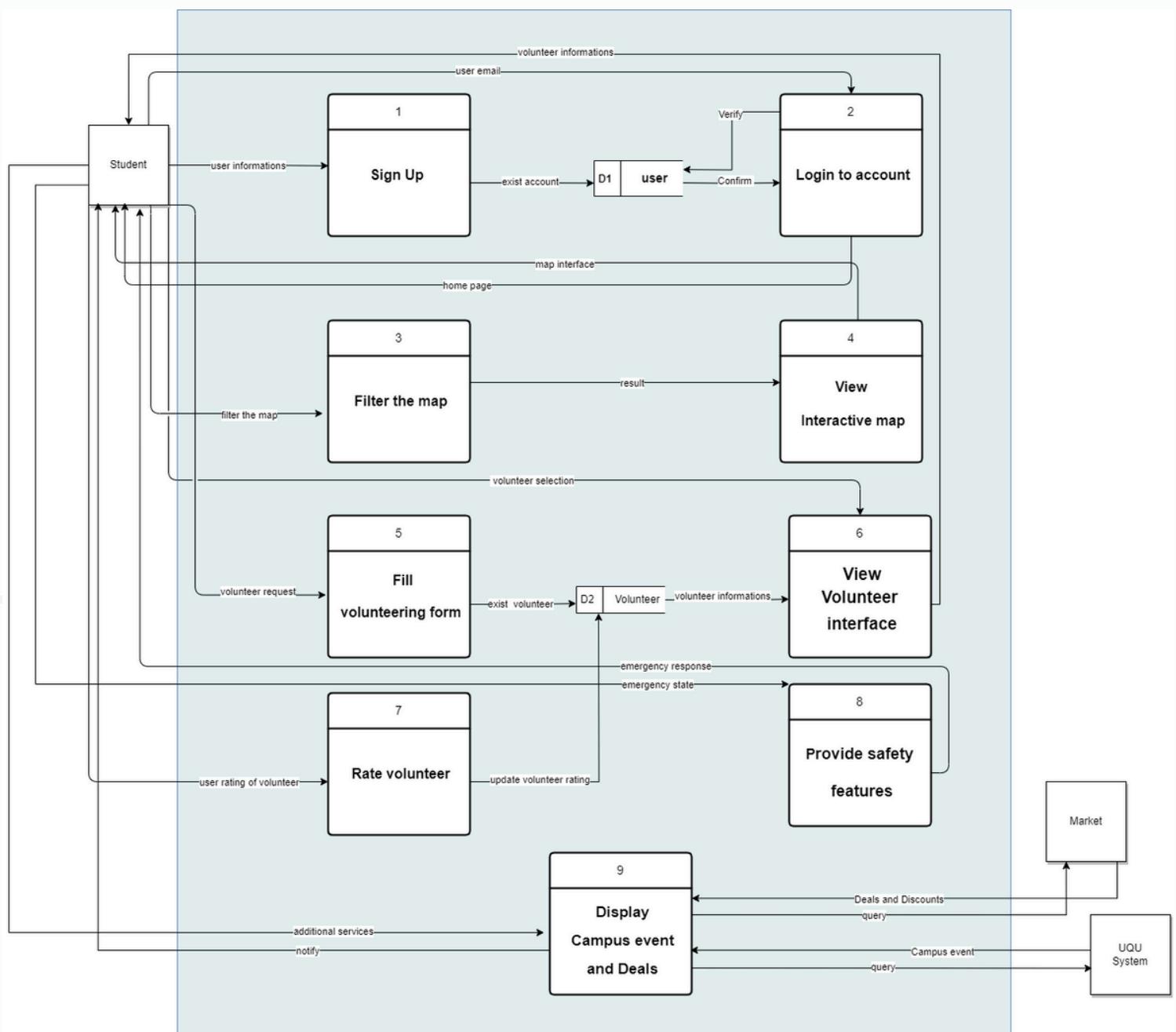
3.1

Context Diagram



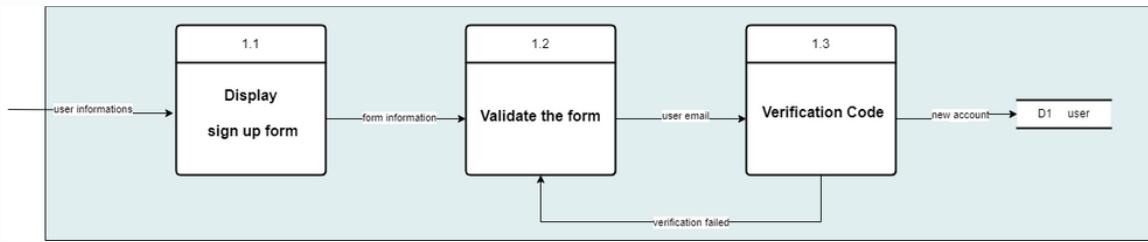
3.2

DFD 1 Diagram



3.2

Child Diagram 1



3.3

Use Case Diagram



Scenarios

Sign up

Actors	Student
Description	<p>-the system is able to display sign up form to the user which contains list of fields : (first name, last name, UQU email, password, confirmation of the password)</p> <p>- the system is able to display pop-up window to the user with a message if the validation was successful verification. otherwise, it return the user to the form and ask to fill the form again with the specified constraints.</p>
Data	(first name, last name, UQU email, password, confirmation of the password)
Stimulus	the trigger for this scenario is user's click on (sign up) button
Response	move to the next step of the registration (verification code) then to home interface
Comments	only the managers has the accessibility to the database

Scenarios

verify Email

Actors	Student
Description	<ul style="list-style-type: none"> -The system is able to send a verification code to the registered email to verify the validity of the email - the system is able to validate the entered data with these constraints: <ul style="list-style-type: none"> * the email must be unique and there is no other record in the database has the same values. * The email should be in UQU database * the password should be at least 8 characters that contains letters in uppercase and lowercase and numbers.
Data	User email
Stimulus	The user enters the email in the designated space
Response	The system verifies that the email is correct and exists in the university's database
Comments	If verification fails, a failure message is sent and the user enters the email again

Scenarios

Log In

Actors	Student
Description	<ul style="list-style-type: none"> -the system is able to display login form to the user which contain these fields: (UQU email and password) - the system is able to check from the entered data with the database, if there is a match retrieve user information and navigate to the main interface, otherwise, display pop-up message contains (there is no such account, try again or sign up).
Data	(UQU email and password)
Stimulus	users clicks on the login button.
Response	The user will be navigated to the home page
Comments	If the user forgot the password then move to the recovery screen that has one field (email or phone number) by sending a verification code

3.4

Scenarios

Fill Volunteering Form

Actors	Student
Description	<ul style="list-style-type: none">- The system shall be able to display the ready-made form for inserting the Volunteer data. If the user chooses study guidance the form must contain (the type of guidance, Academic level, Subject, department, email, date, and time) and if the user chooses guidance to university facilities the form must contain (the type of guidance, email, department, location, time).- The system shall be able to save data in the database.- The system shall be able to show the new volunteer in the user interface. The volunteer will appear in green if he is online and ready to help at present.-The system should save the number of volunteer times for each volunteer.
Data	If the user chooses study guidance the form must contain (the type of guidance, Academic level, Subject, department, email, date, and time) and if the user chooses guidance to university facilities the form must contain (the type of guidance, email, department, date, time).
Stimulus	users clicks on the Volunteering Form button.
Response	The user will be navigated to the Volunteering Form
Comments	None

Scenarios

Display Volunteer Information

Actors	Student
Description	-The system displays the volunteer's information, including full name, department, specialty, and email and rating
Data	None
Stimulus	Users click on the volunteer whose data they want to see
Response	The user will be navigated to the Volunteering interface
Comments	None

Scenarios

View Campus Events and Notifications

Actors	Student , UQU system
Description	-The System shall be able to give notifications about upcoming events, workshops, seminars, and gatherings taking place on the university campus.
Data	None
Stimulus	Users click on Additional Services and then select upcoming events
Response	The user will be informed of upcoming events by sending notifications
Comments	None

Scenarios

Provide Campus Events and Notifications

Actors	Student , UQU system
Description	-The system inquires about the latest events, workshops, seminars, etc. that are happening now from the university system, and then the data is transferred to the program system for display.
Data	latest events, workshops, seminars, etc.
Stimulus	The system inquires about events
Response	The system receives the required information
Comments	None

Scenarios

get special discount and offers

Actors	Student , market
Description	<ul style="list-style-type: none">-The system must provide students with access to exclusive discounts and offers at the university-The system must allow volunteers with high points to benefit from special discounts and offers.
Data	None
Stimulus	Users click on Additional Services and then select discount and offers
Response	The user will be informed of discount and offers by sending notifications
Comments	None

Scenarios

Provide Campus Deals and Discounts

Actors	Student , market
Description	<ul style="list-style-type: none"> -The system inquires about the latest discounts and offers from nearby markets to provide additional service to students -If the volunteer has high points due to his good work, he will be provided with special offers
Data	latest discounts and offers
Stimulus	The system inquires about offers
Response	The system receives the required information
Comments	None

Scenarios

Provide Safety Features

Actors	Student
Description	<ul style="list-style-type: none"> -The system is able to display the security form to the user that contains a list of fields (red button, yellow button, green button). * The user can use the red button if he needs high-level assistance, such as medical assistance. * The user can use the yellow button if he needs help at an intermediate level, such as getting a phone charger. *The user can use the green button if he needs low-level assistance, such as getting a pen. -The system is able to display a pop-up window through which one can press the desired button. -The system can display a pop-up window to the user with a message in which he writes the type of help he needs.
Data	What type of emergency state
Stimulus	users clicks on the safety buttons.
Response	Help is provided to her by a volunteer
Comments	None

Scenarios

Filter

Actors	Student
Description	<ul style="list-style-type: none">- The system should be able to filter to reduce search time.- The system should be able to filter based on the type of volunteering (university facilities, academic guidance)and if the user chooses campus events, the filtering should be based on event location and type (educational or entertainment).
Data	Filter classifications
Stimulus	Users click the filter button and select the category they want
Response	Show the interactive map interface after filtering
Comments	None

Scenarios

View Map

Actors	Student
Description	<ul style="list-style-type: none"> -The system should be able to show available volunteers at the specified time along with their current locations -The system should be able to differentiate between types of volunteering by displaying different colors. Blue is for guidance for facilities and pink is for study guidance -The system should be able to display all of the volunteer's information, including his name, department, email, and others when clicked on -The system should be able to remove the volunteer from the map when the pre-determined volunteer period ends
Data	None
Stimulus	users clicks on the interactive map button.
Response	Show the interactive map interface
Comments	None

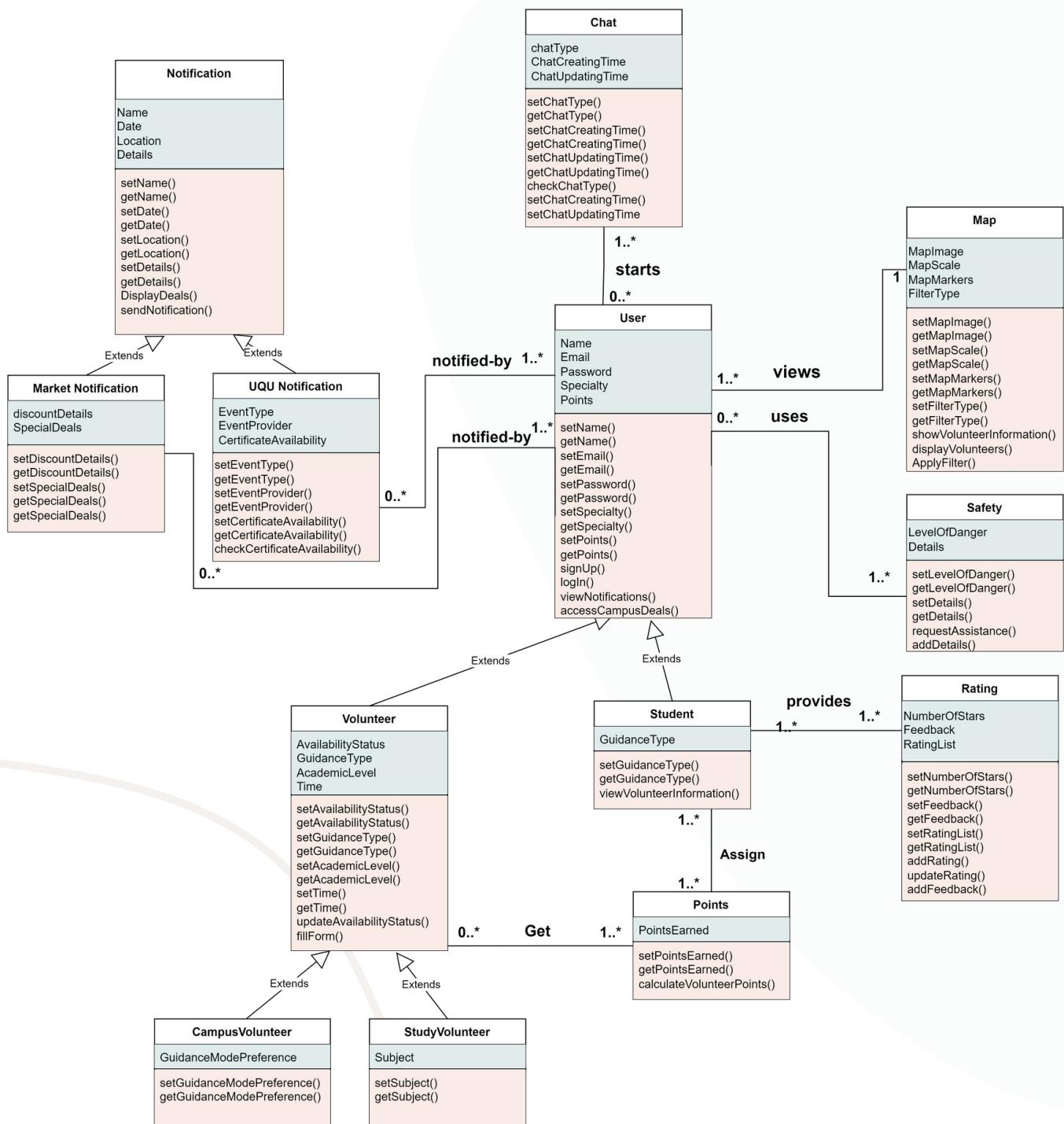
Scenarios

rate and give feedback

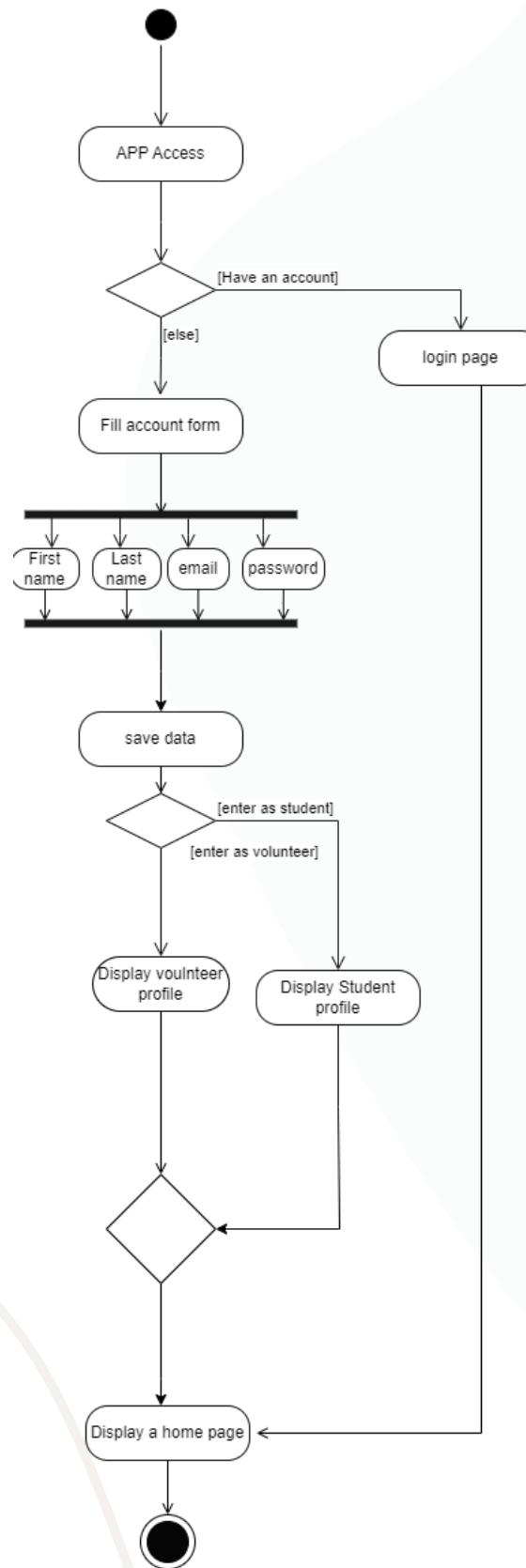
Actors	Student
Description	<ul style="list-style-type: none"> -The system will allow users to choose from the ratings list: -Cooperative -Brief explanation -Detailed explanation - Respectful - Uses various teaching methods. -Impolite -Poor interaction -Inadequate explanation -Poor ethics -Takes more time than necessary. <p>Also, user can add additional comments , then display it in the volunteer's rating page.</p>
Data	ratings list and comments
Stimulus	The user clicks the rating button
Response	The Systems rating criteria page appears
Comments	None

3.5

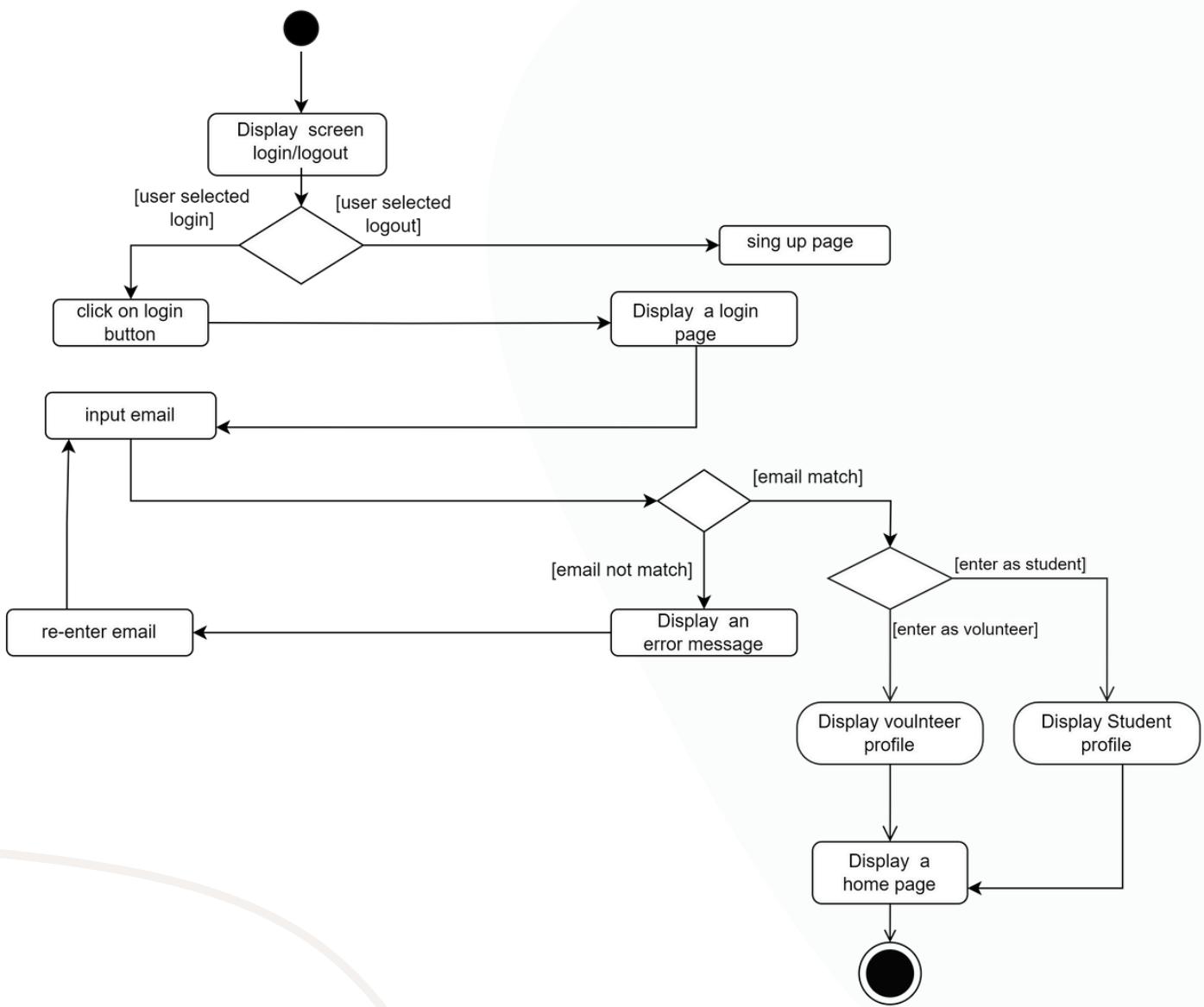
Class Diagram



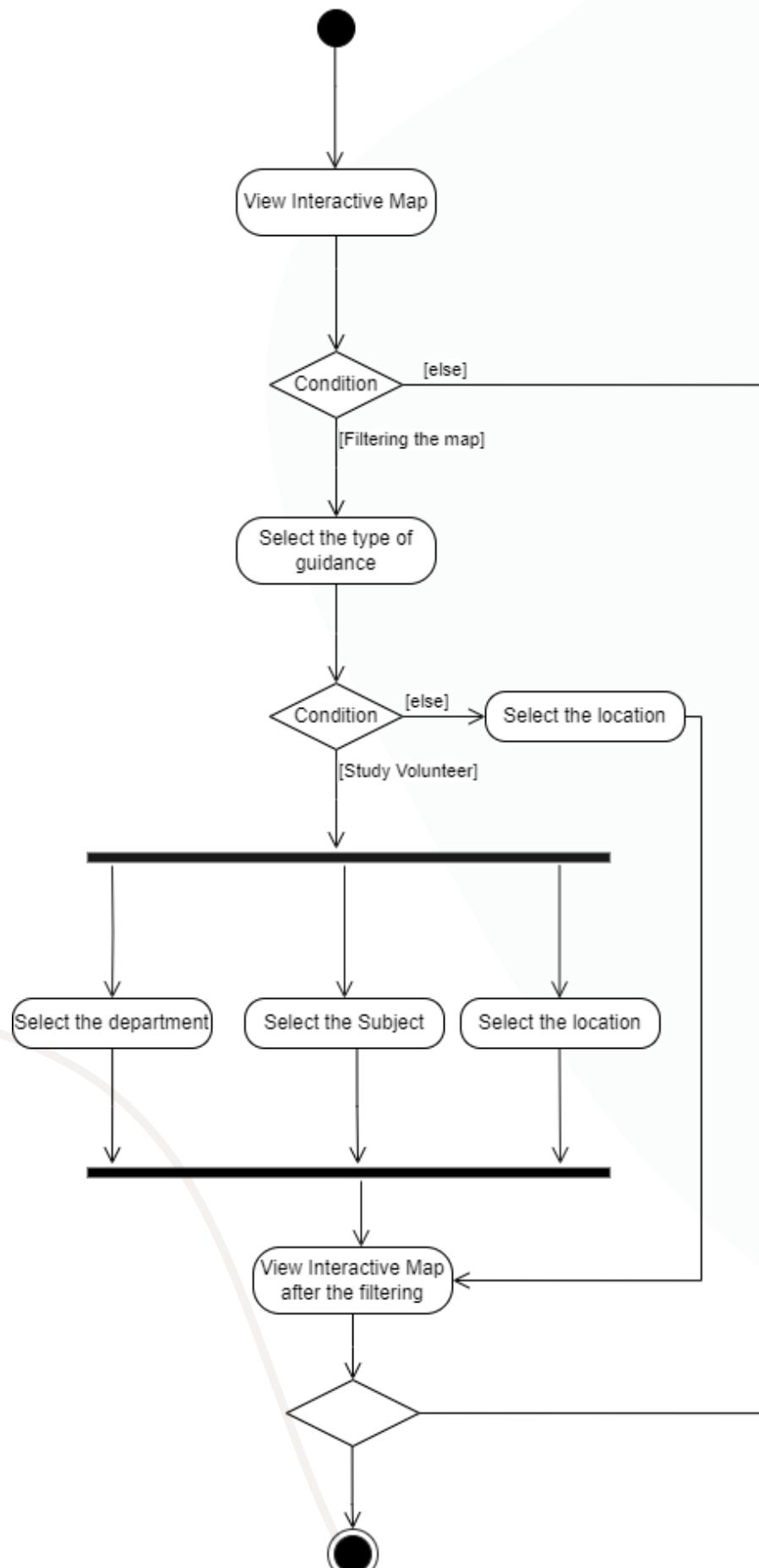
Activity Diagram (Sign up)



Activity Diagram (Login)

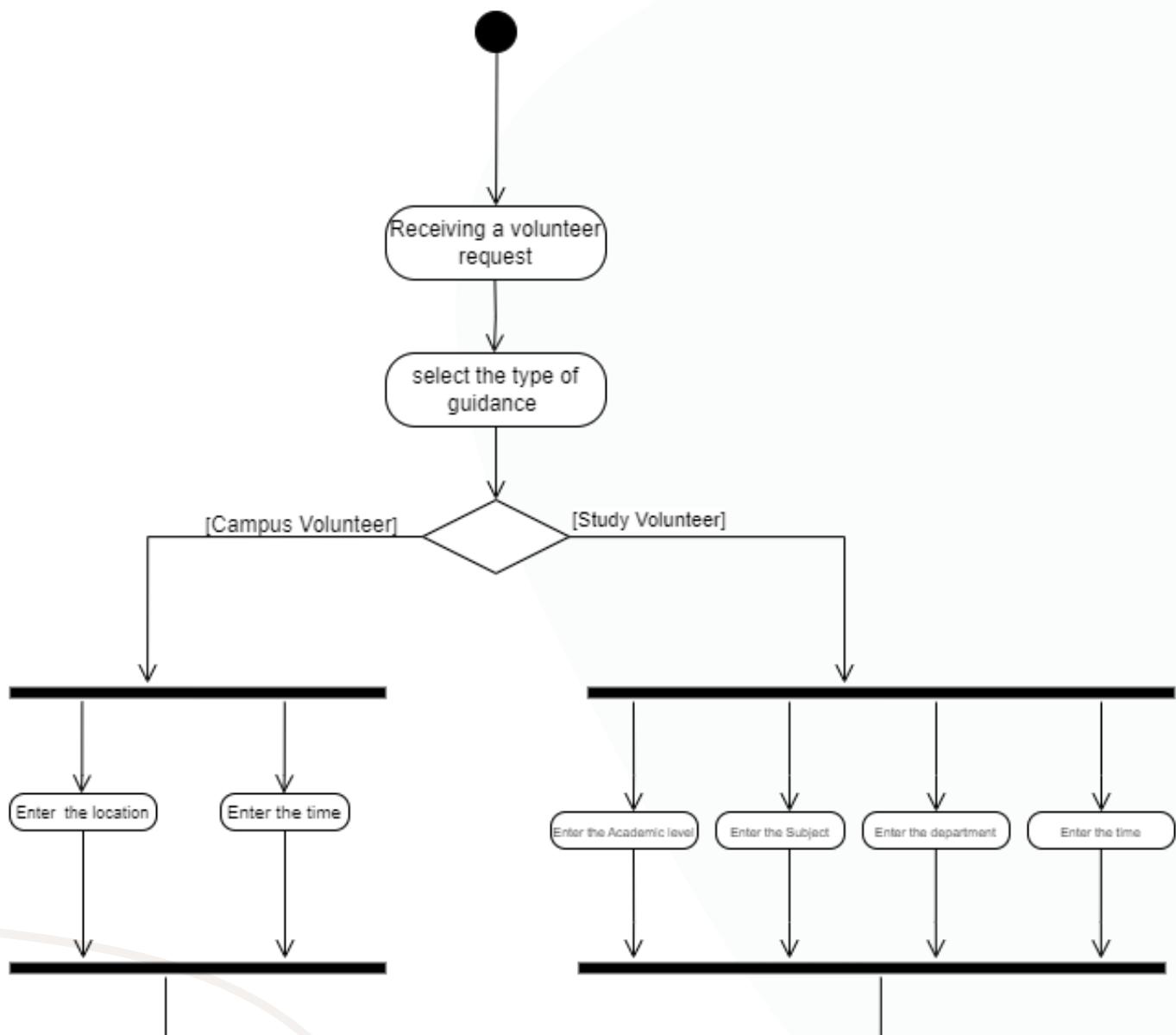


Activity Diagram (View Interactive map and Filtering)



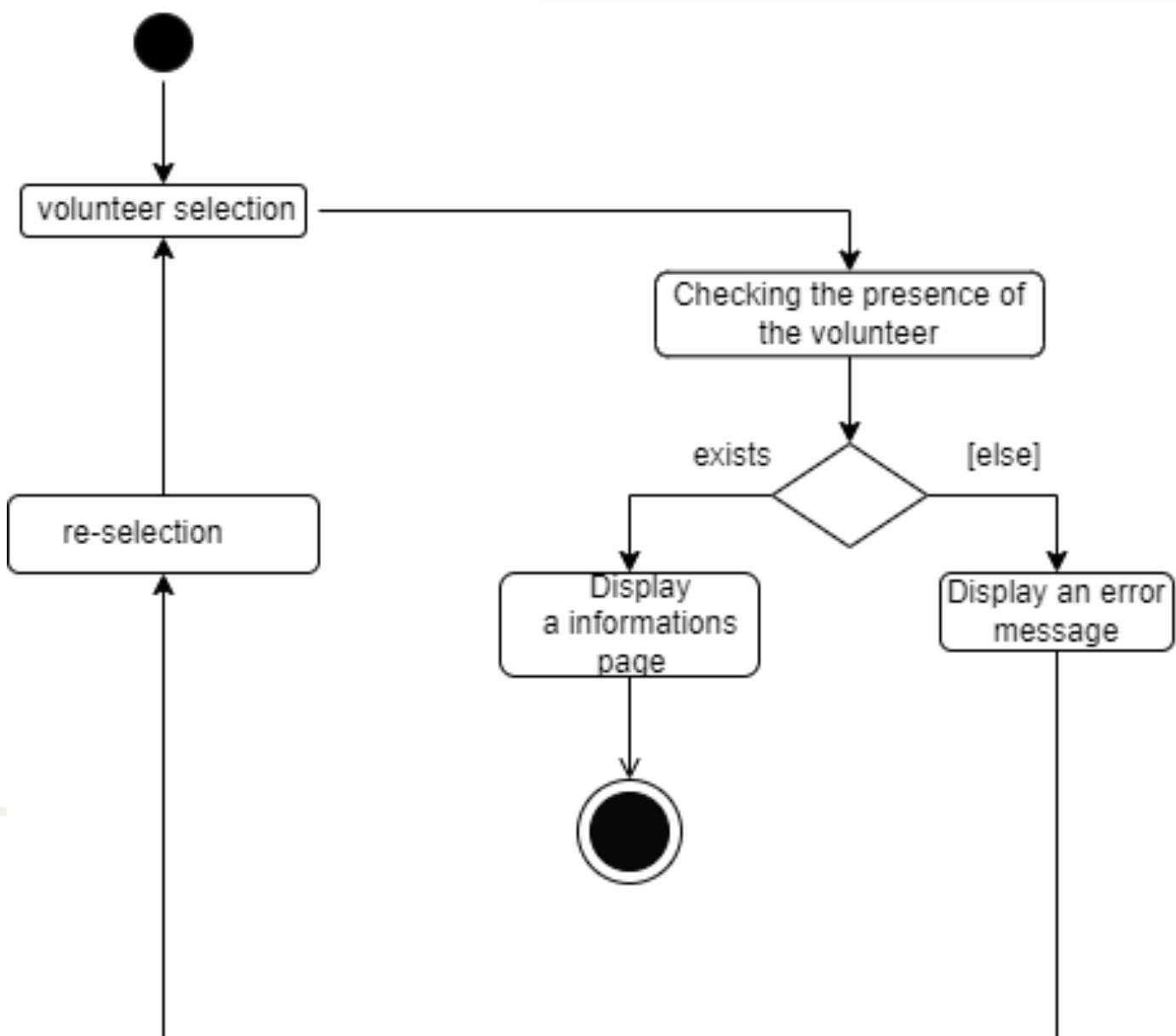
3.6

Activity Diagram (Fill volunteering form)



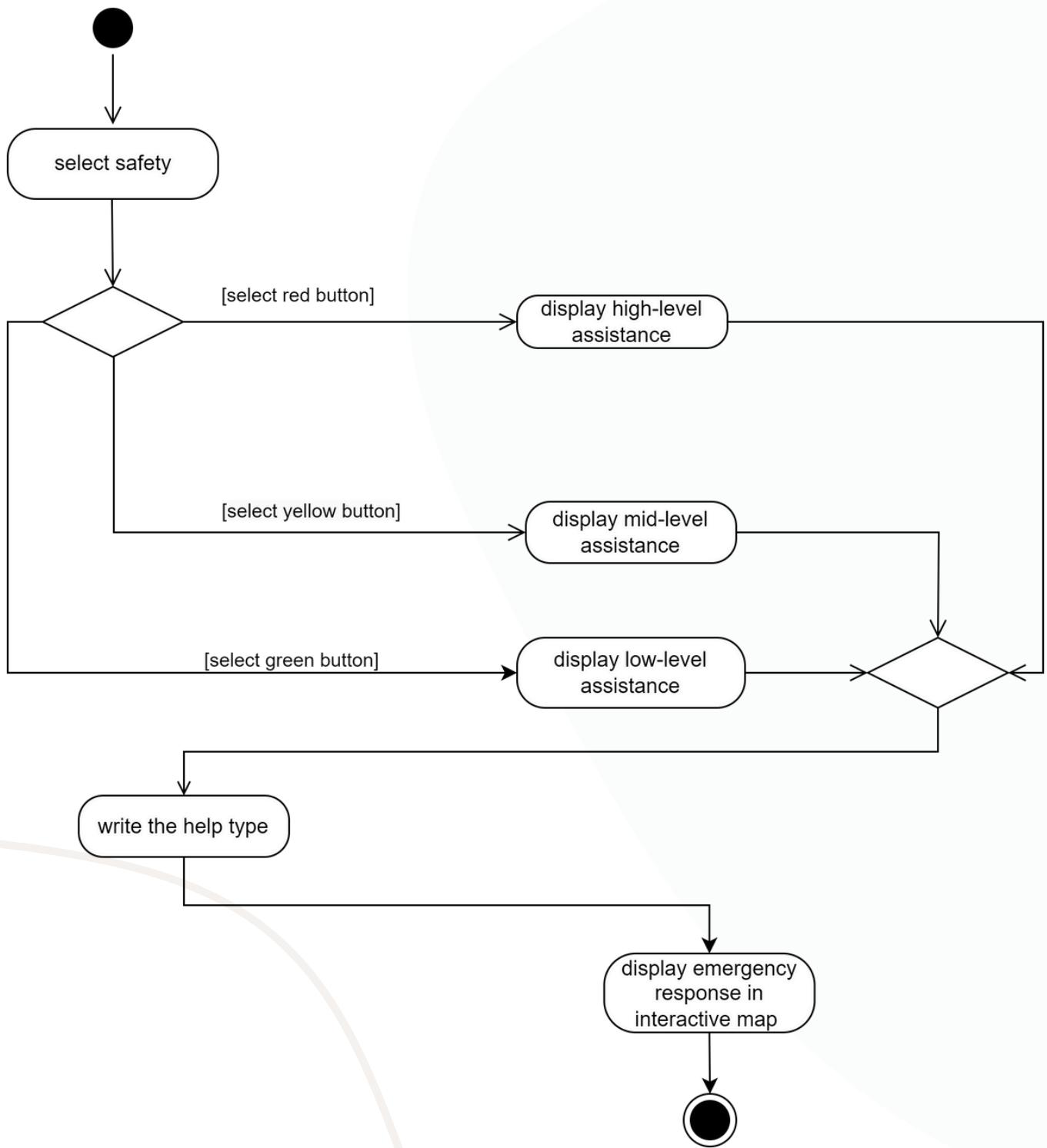
3.6

Activity Diagram (Display Volunteer Information)



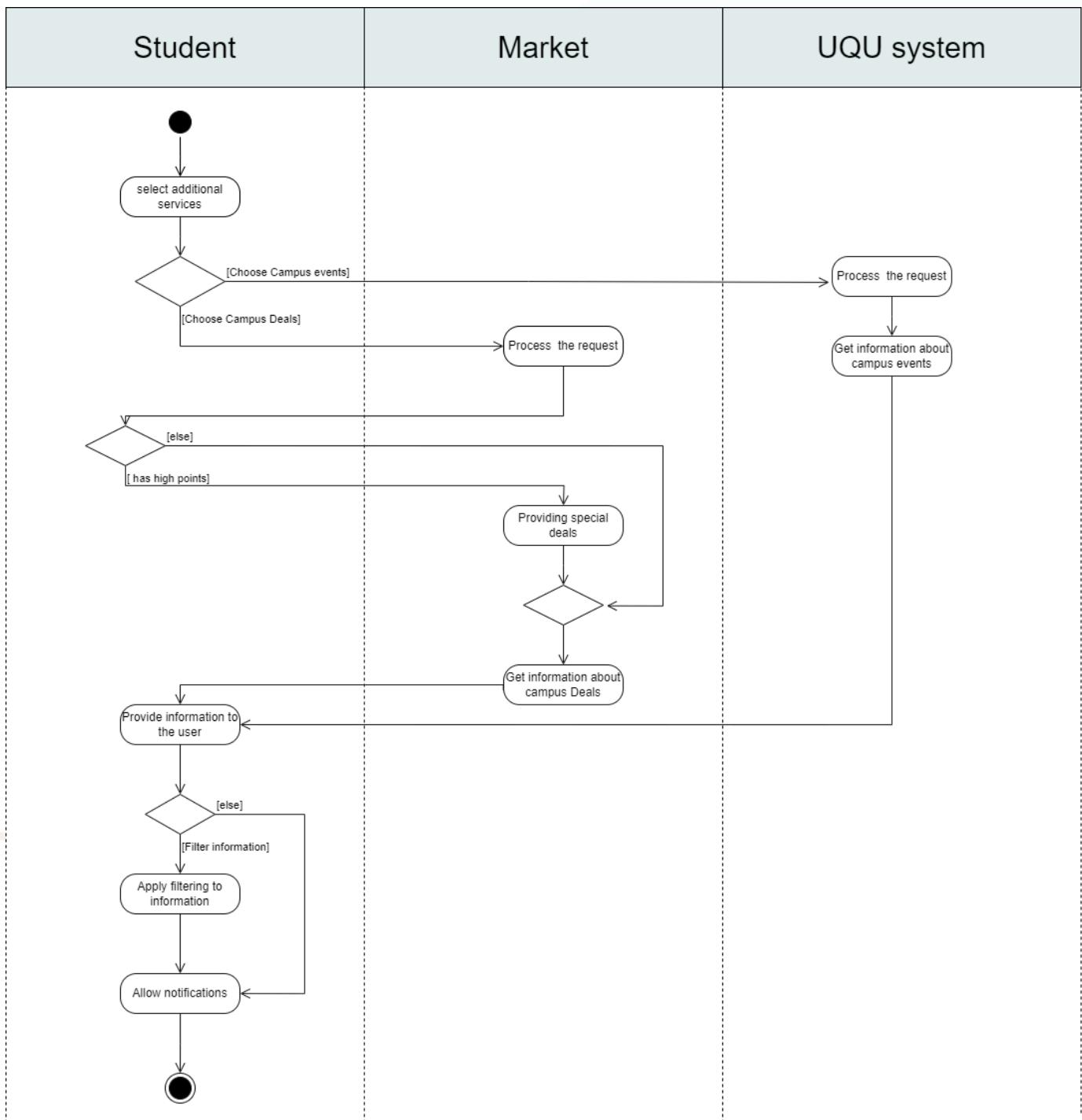
3.6

Activity Diagram (Provide Safety features)



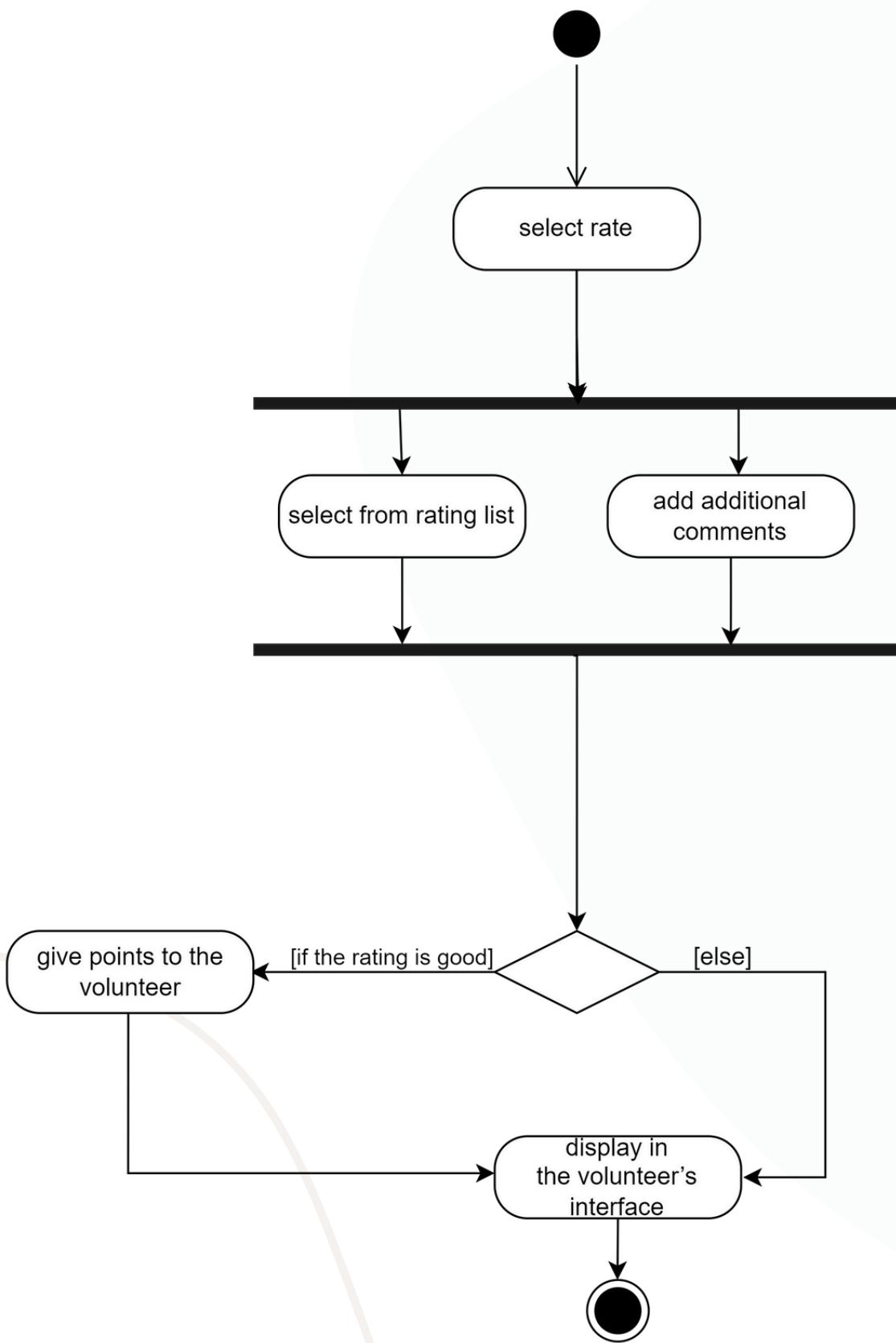
3.6

Activity Diagram (View Campus event and Provide Campus Deals)



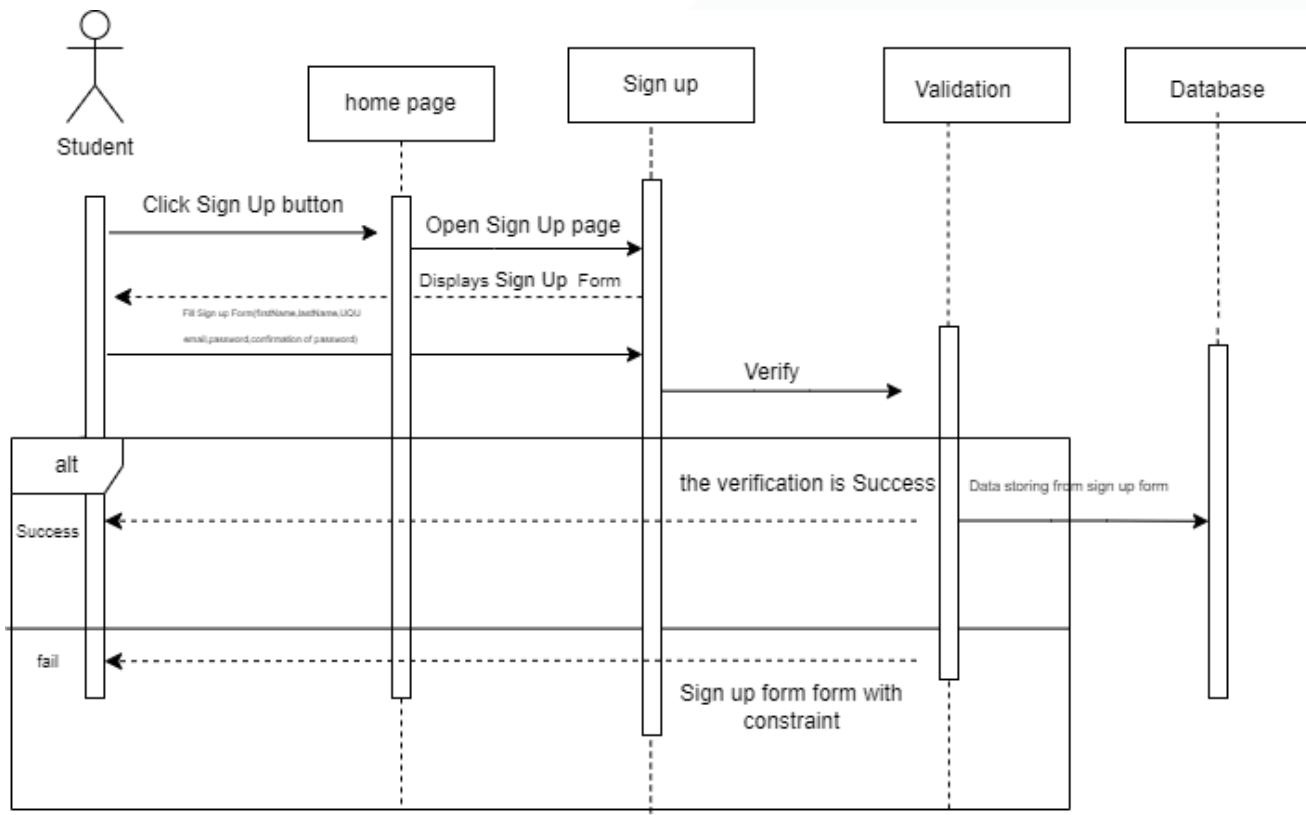
3.6

Activity Diagram(Rate and give Feedback)

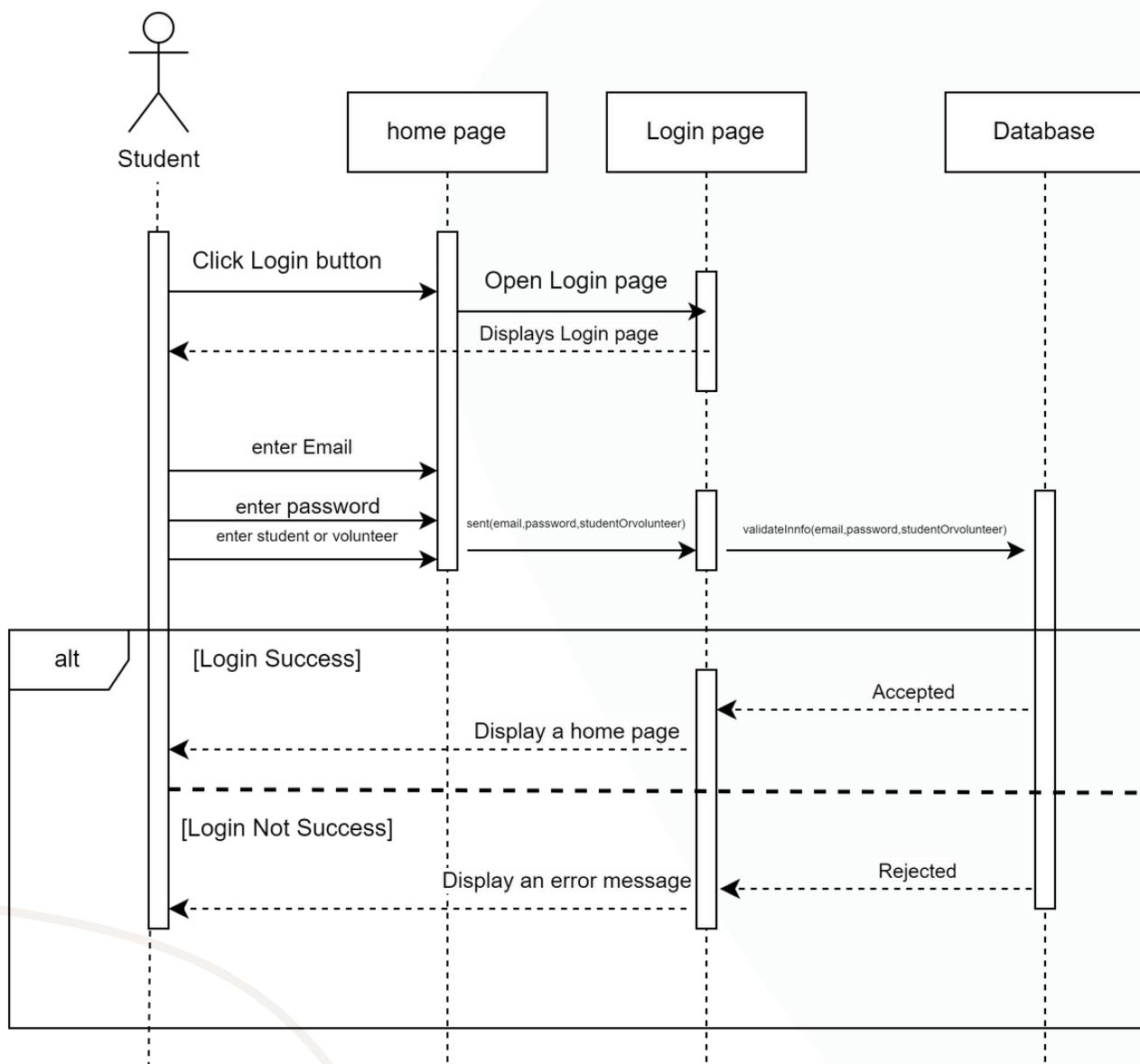


3.7

Sequence Diagram(Sign up)

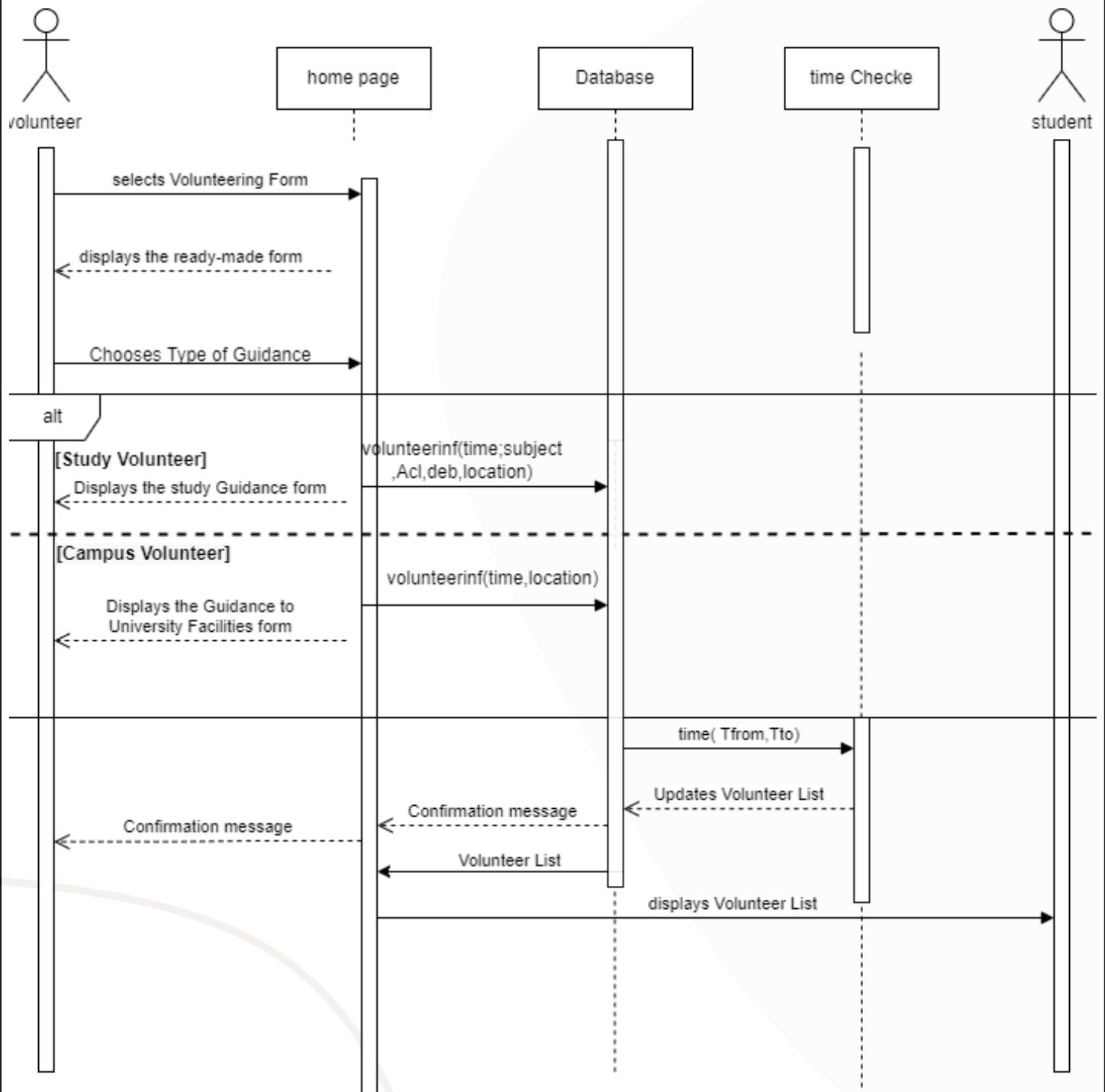


Sequence Diagram(Login)



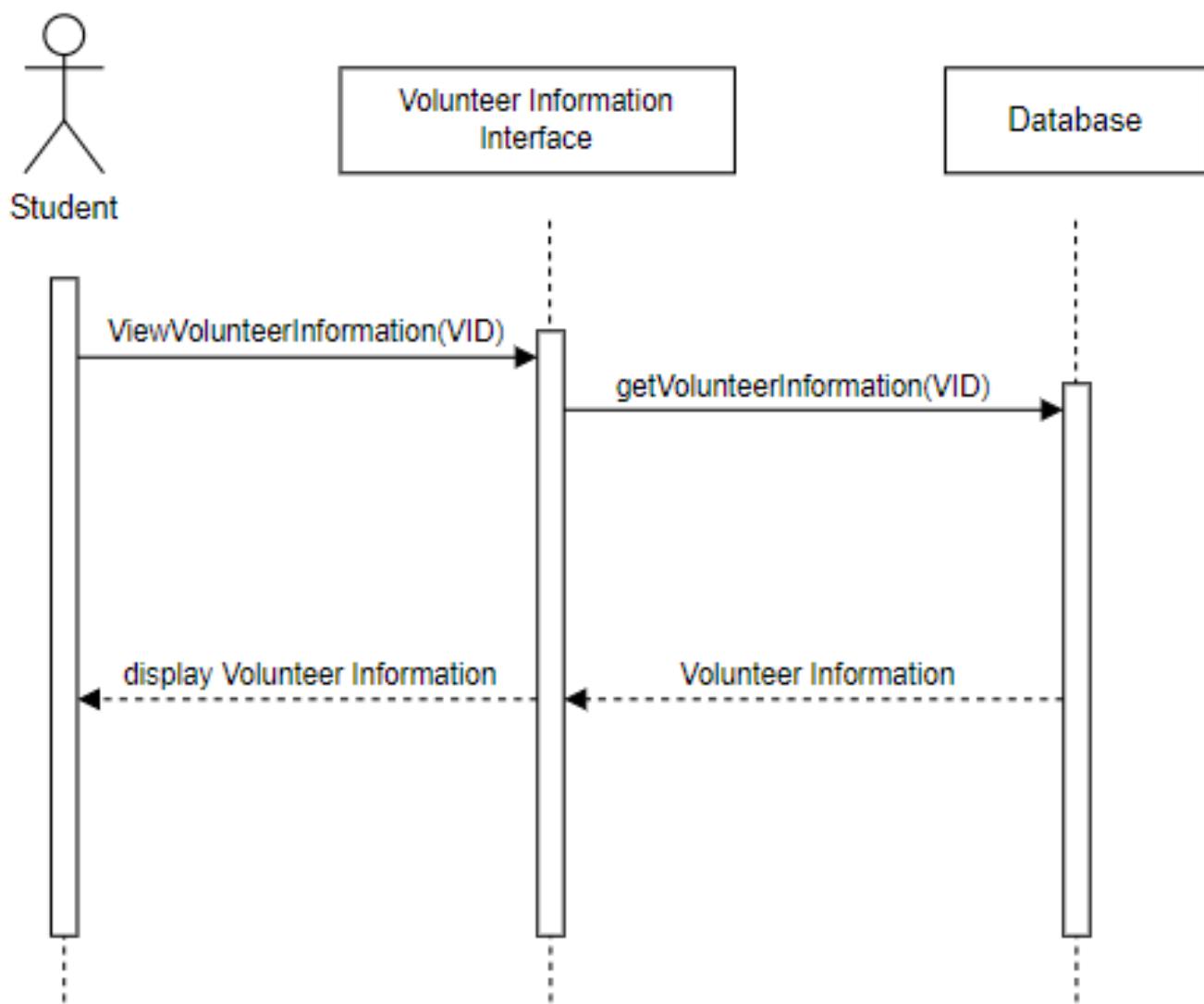
3.7

Sequence Diagram(Fill volunteer form)



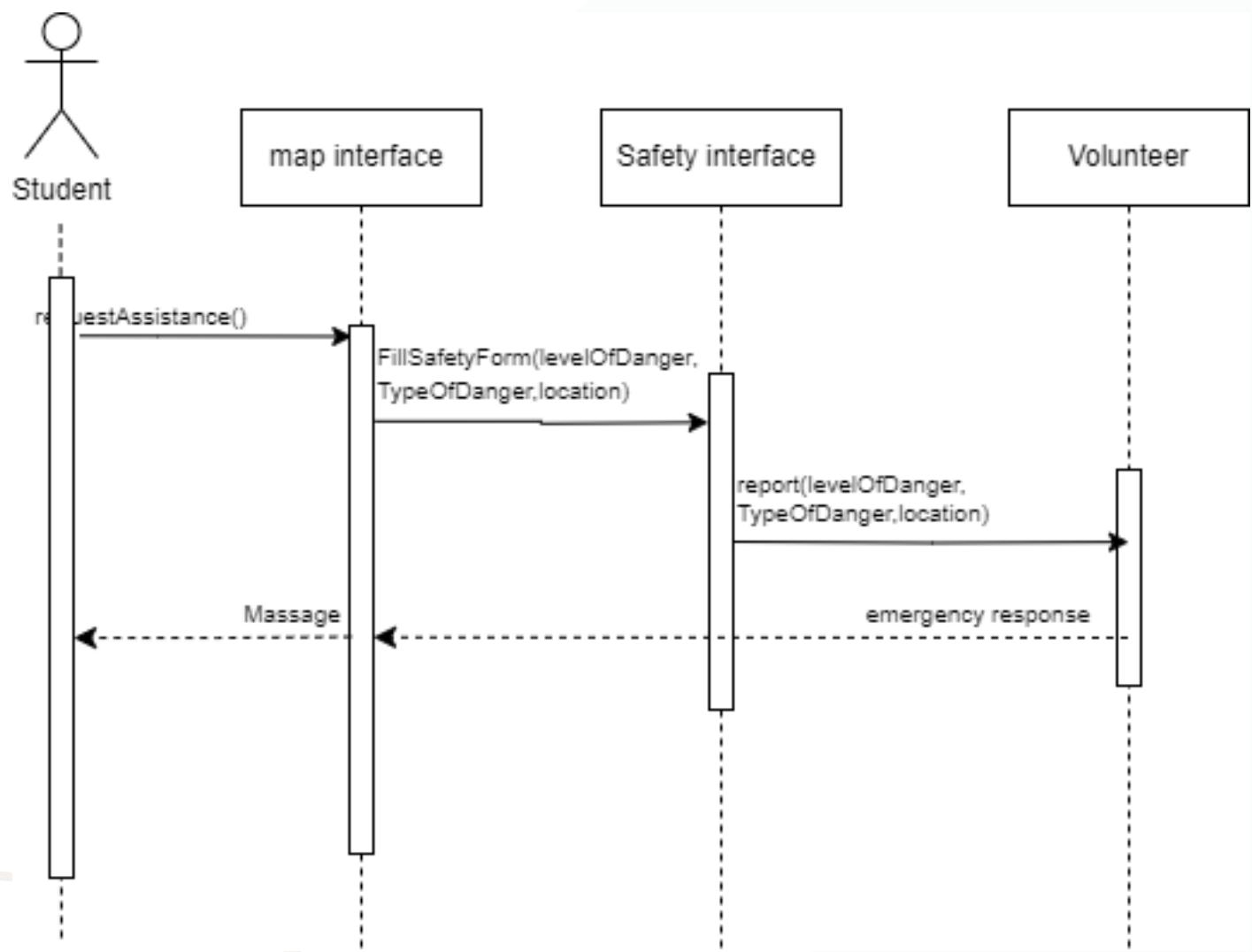
3.7

Sequence Diagram(View volunteer information)



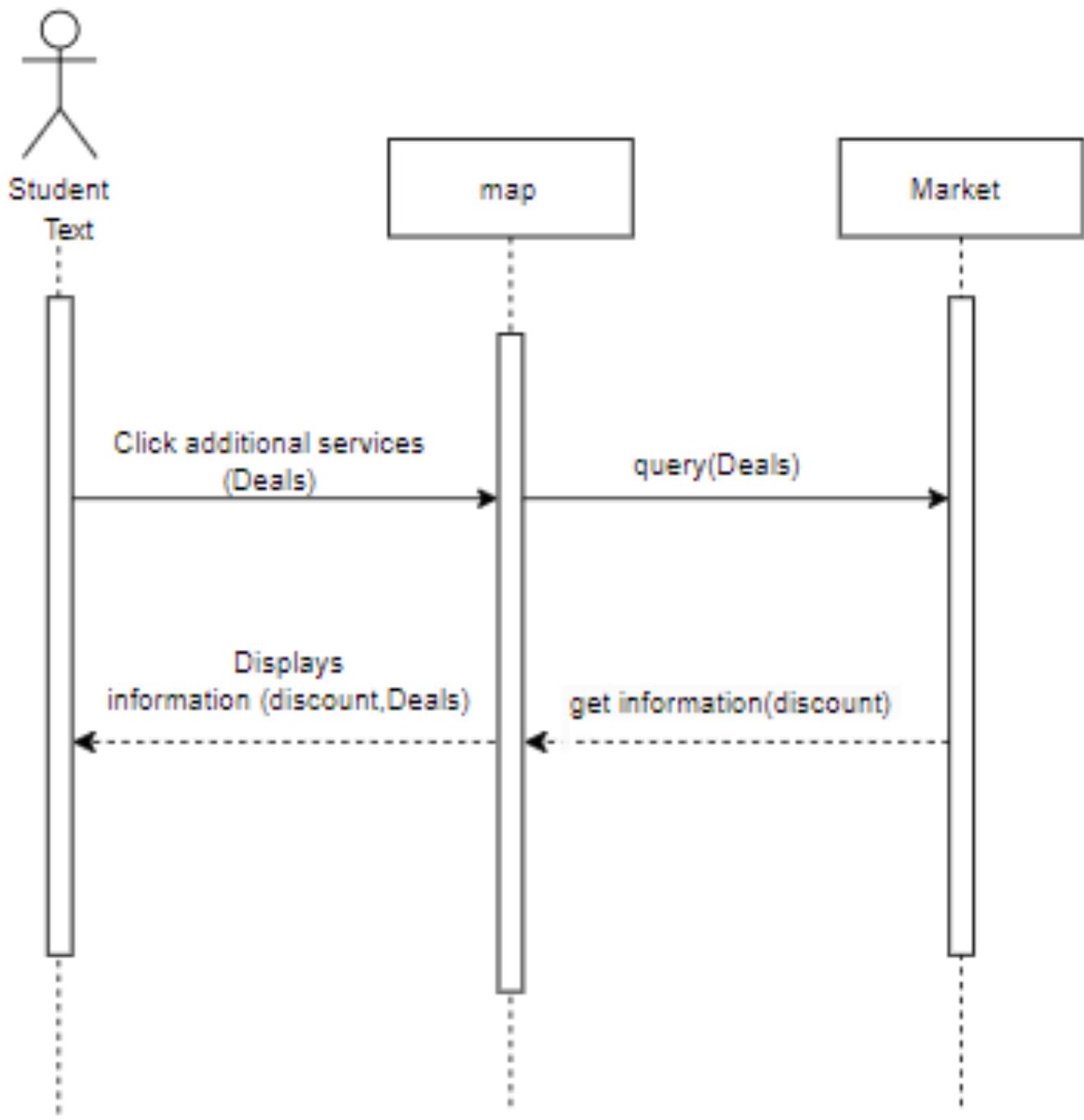
3.7

Sequence Diagram(Provide Safety Features)



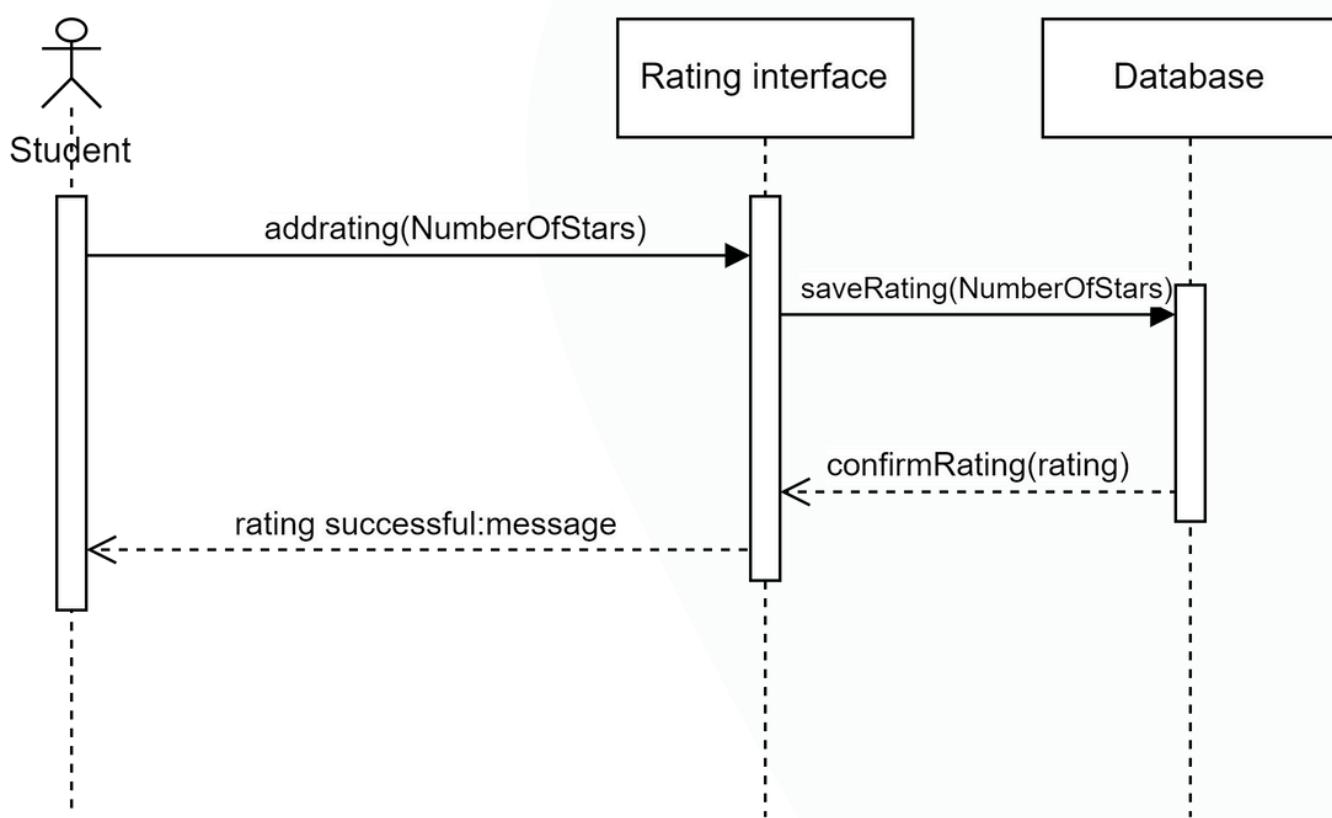
3.7

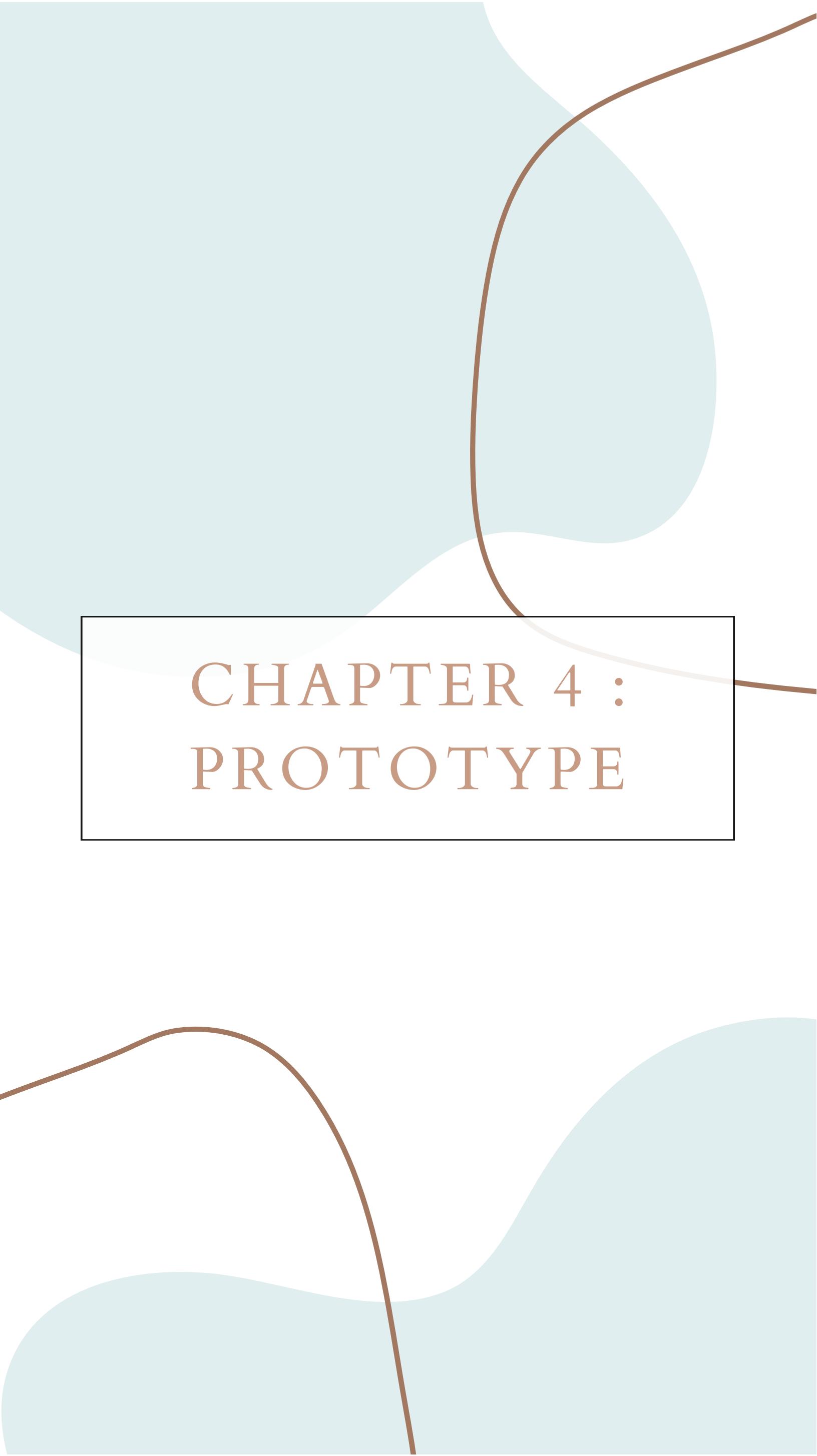
Sequence Diagram(Provide Campus Deals and Discounts)



3.7

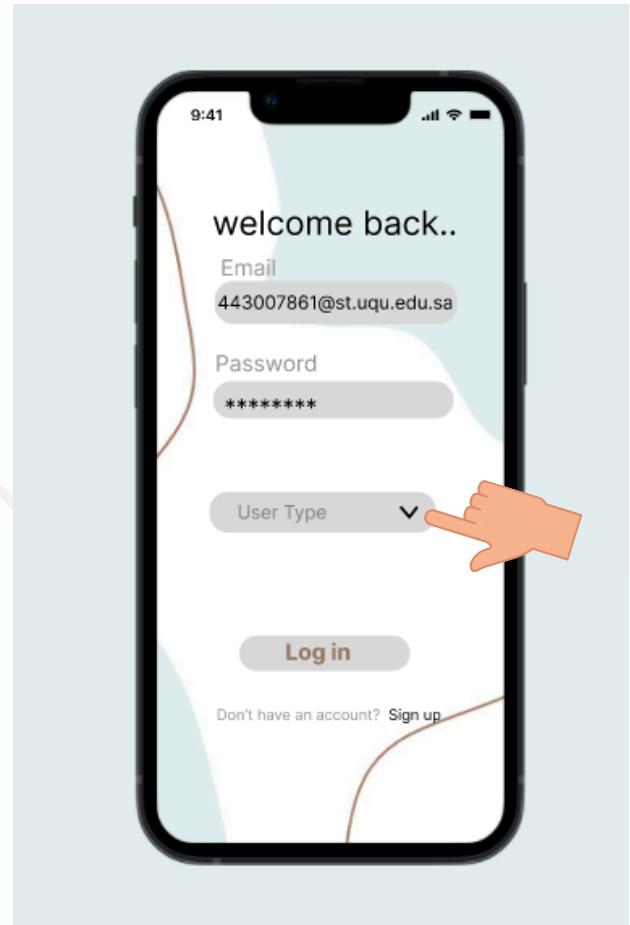
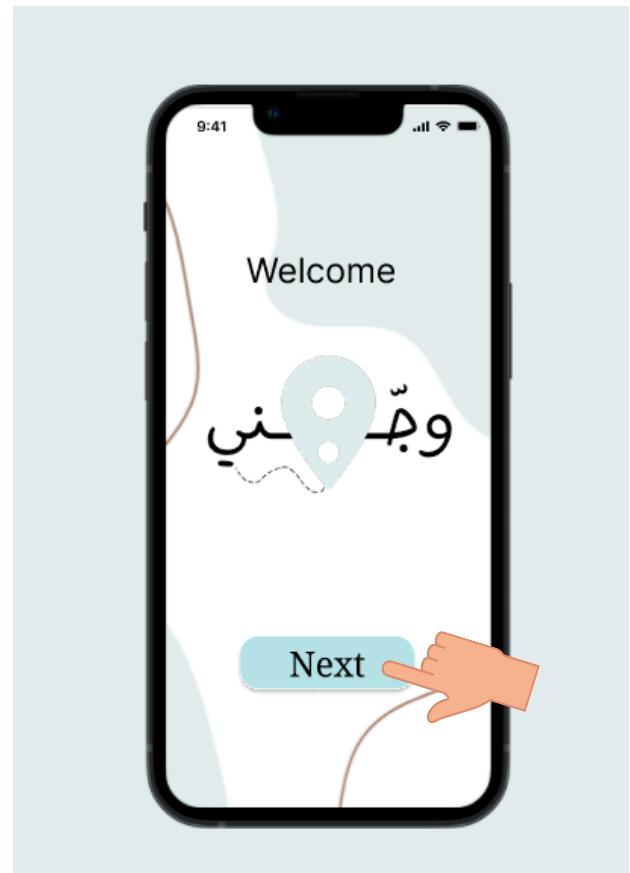
Sequence Diagram(Rate and give Feedback)



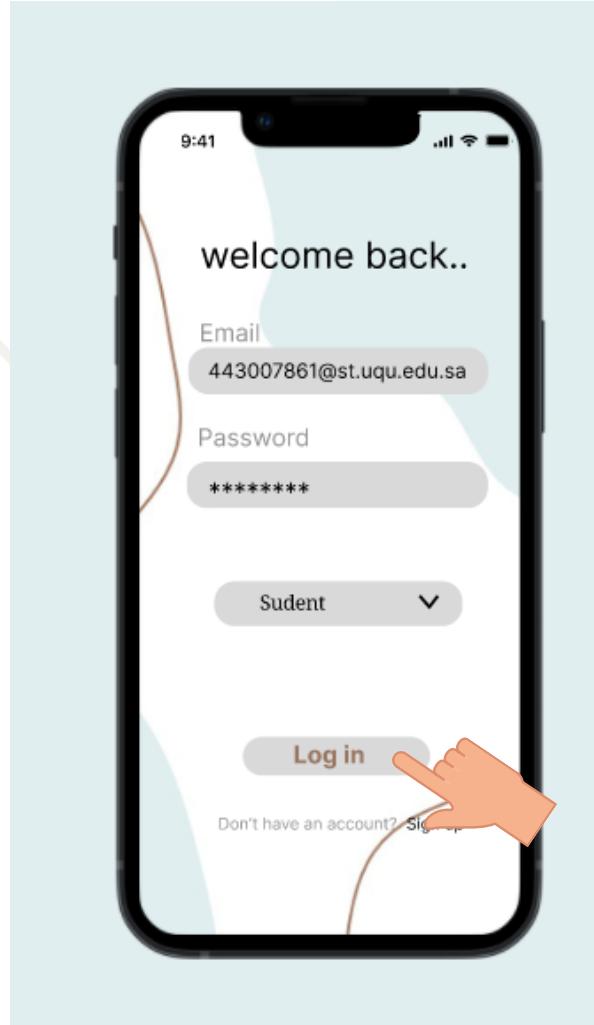
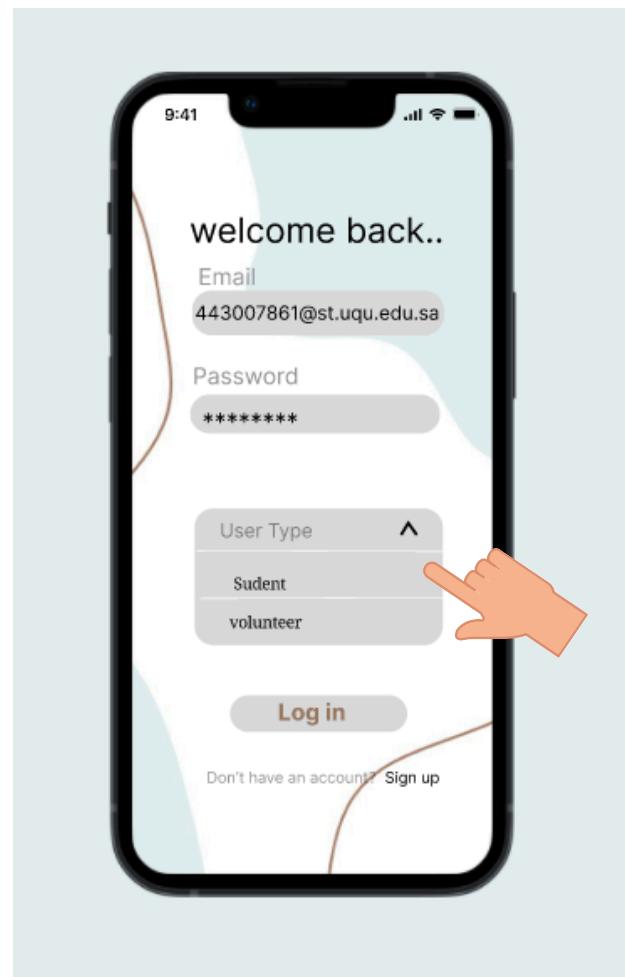


CHAPTER 4 : PROTOTYPE

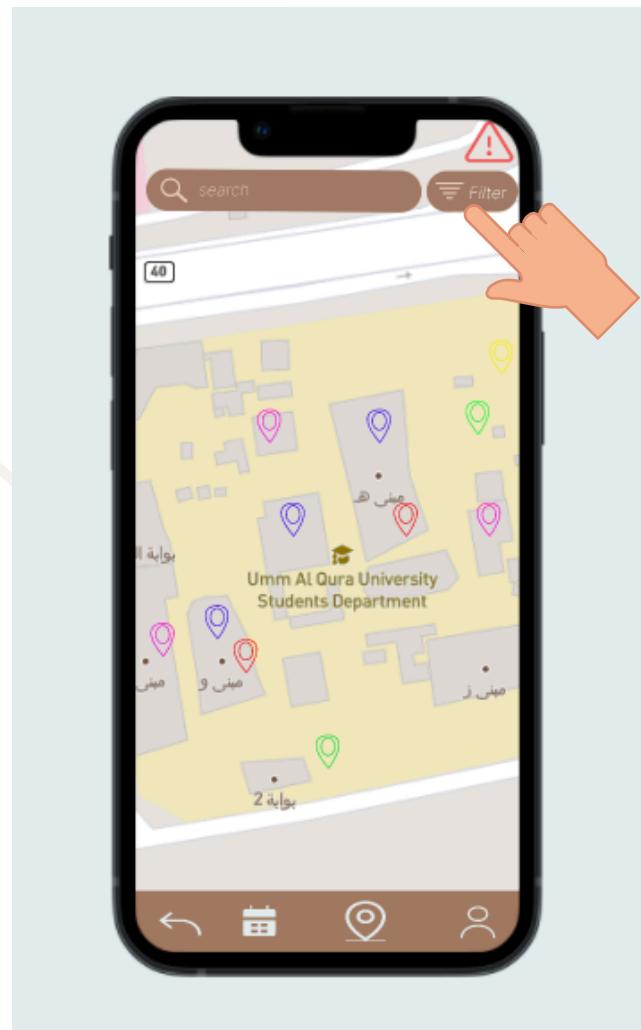
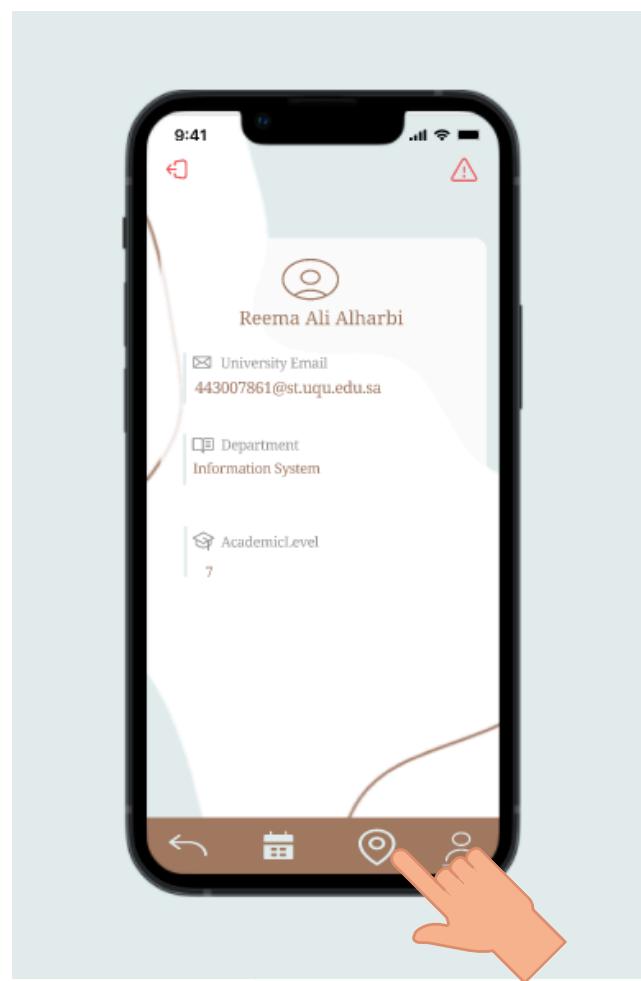
4.1



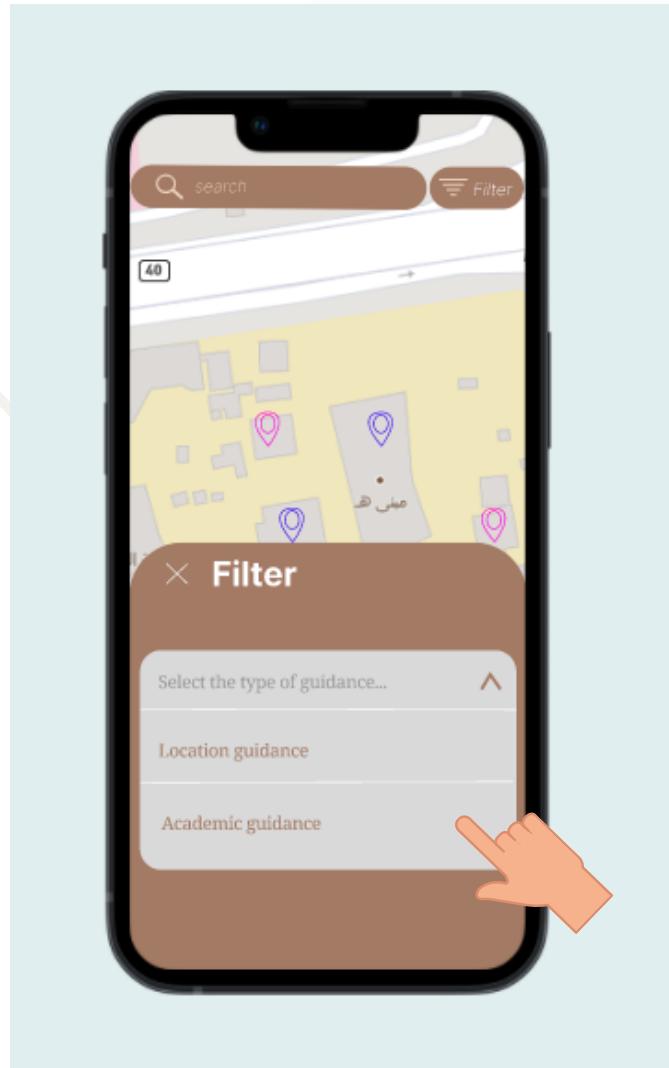
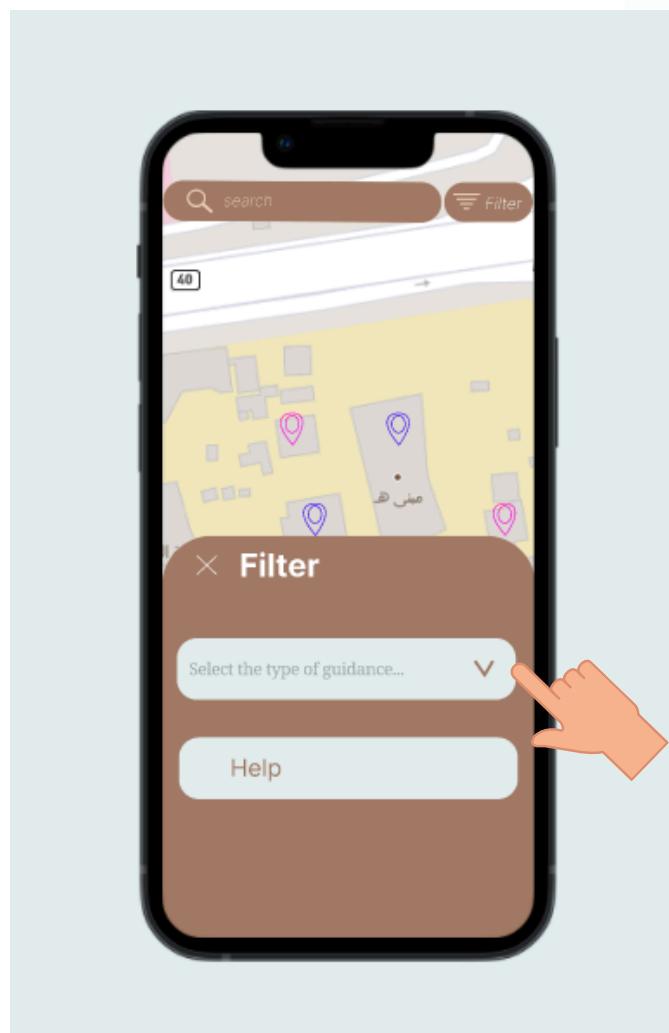
4.1



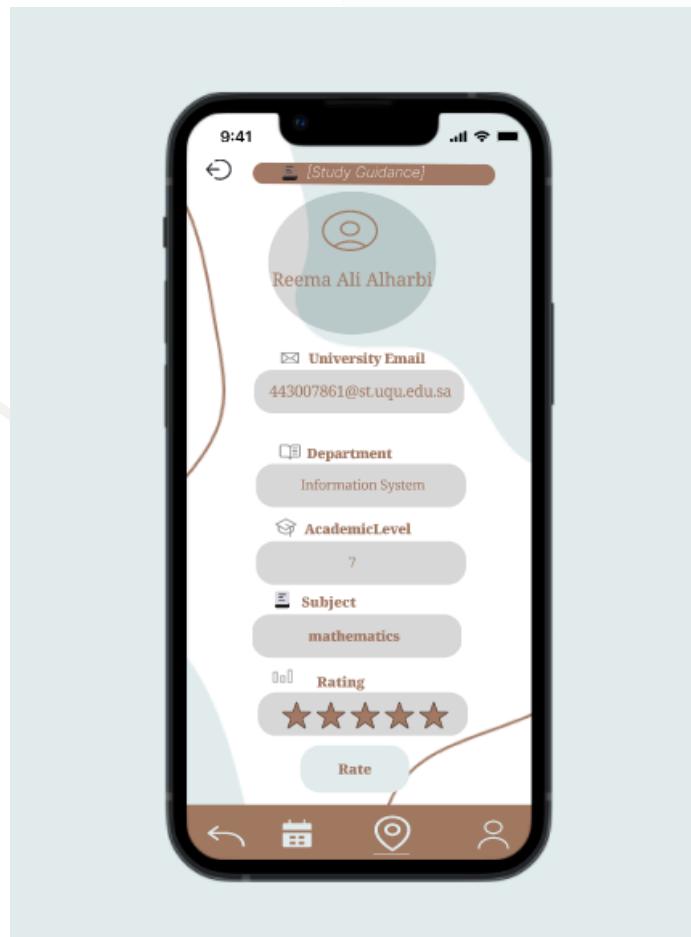
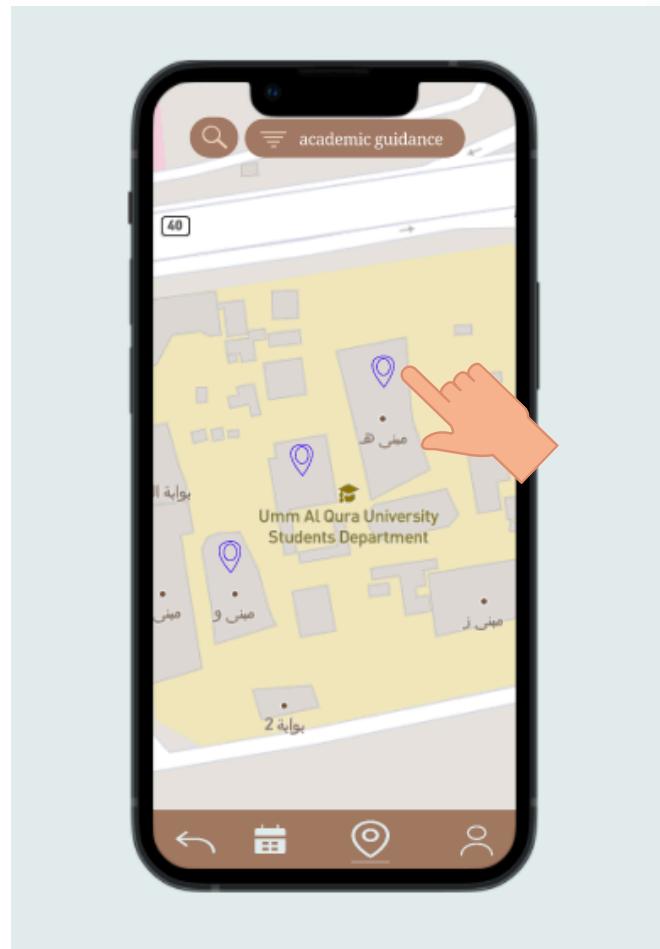
4.1



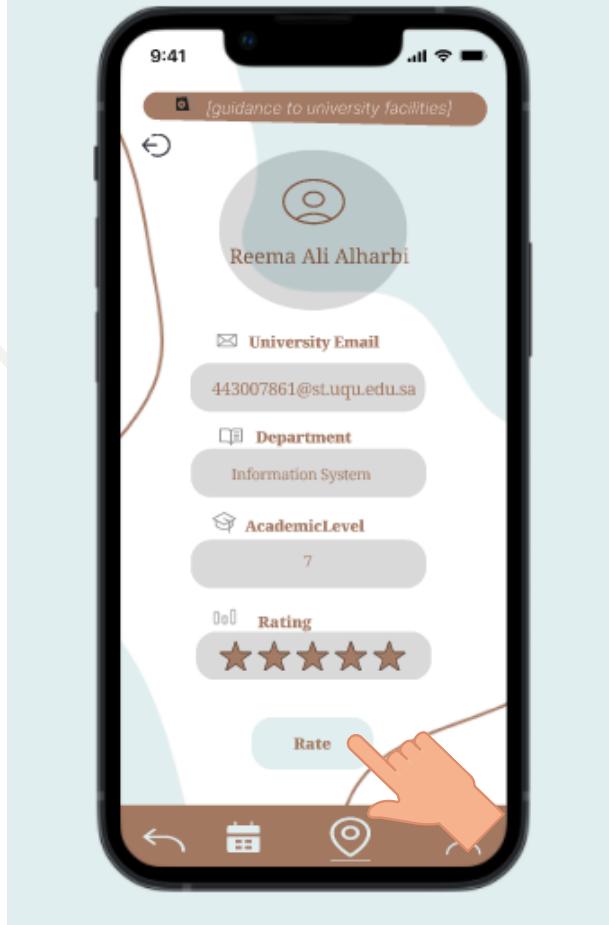
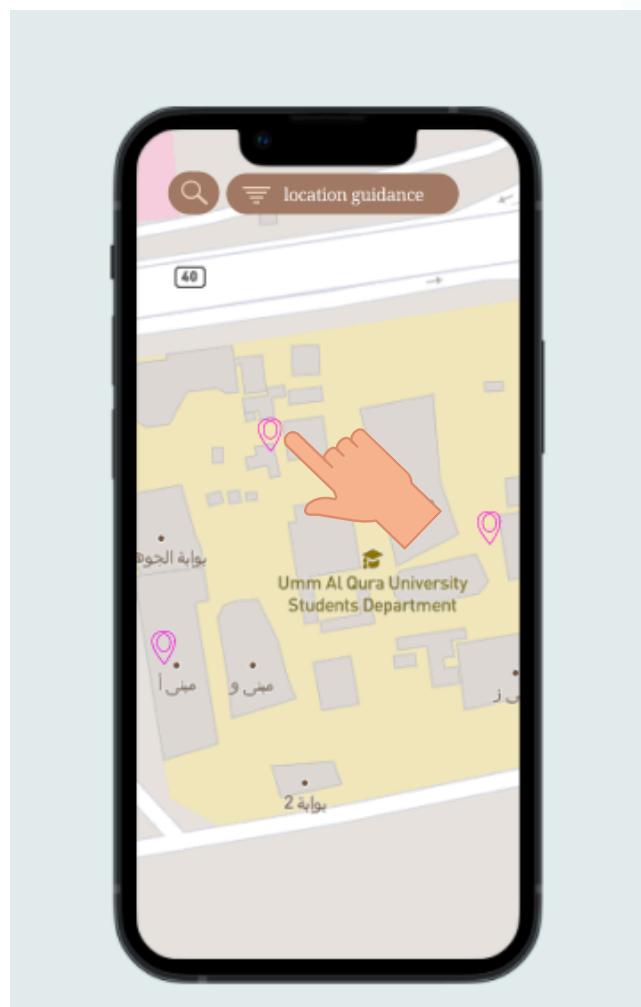
4.1



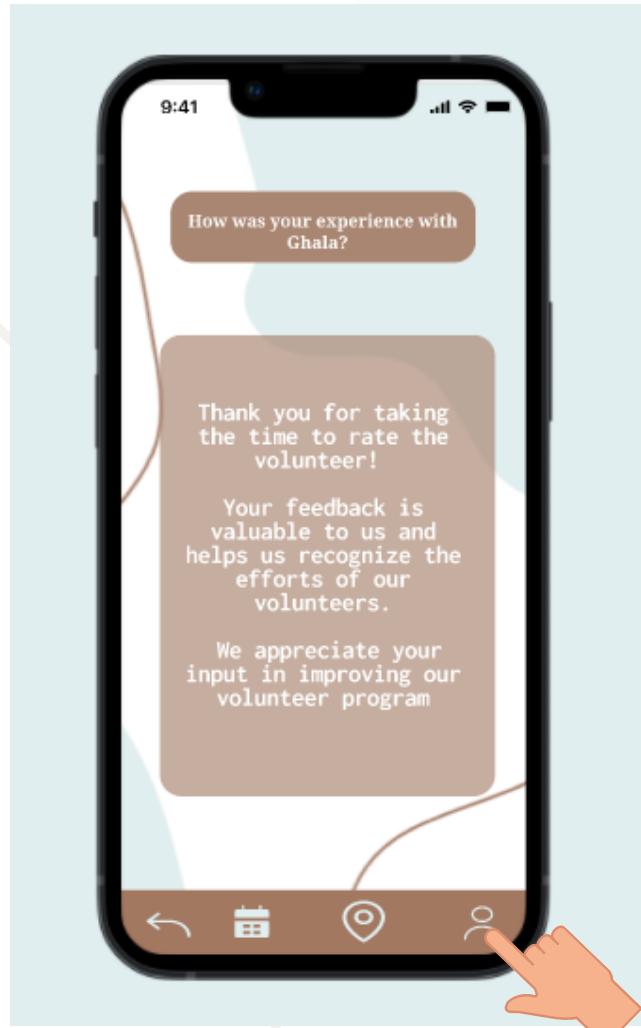
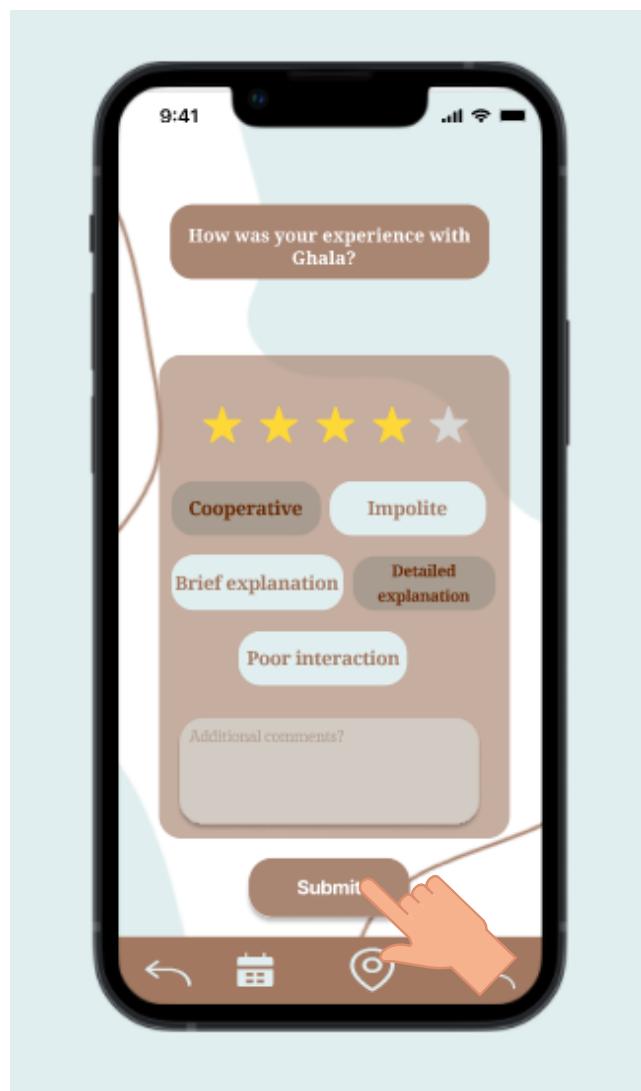
4.1



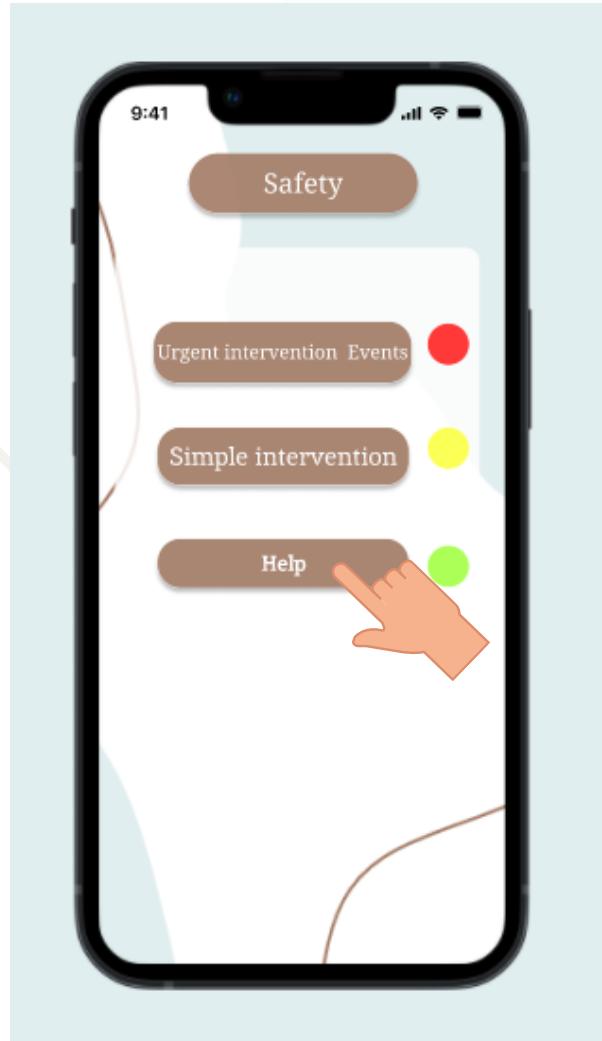
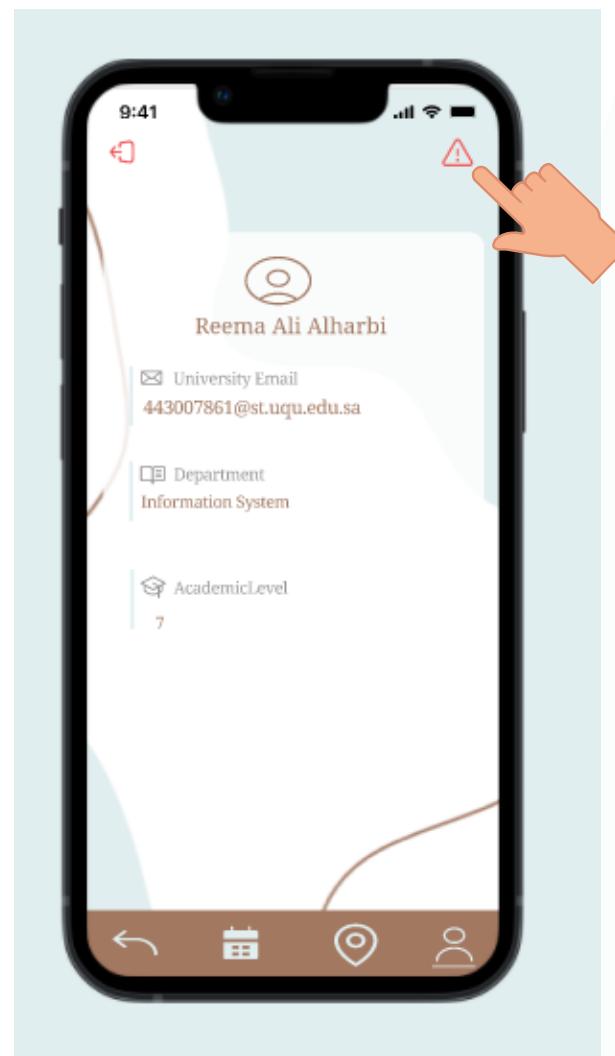
4.1



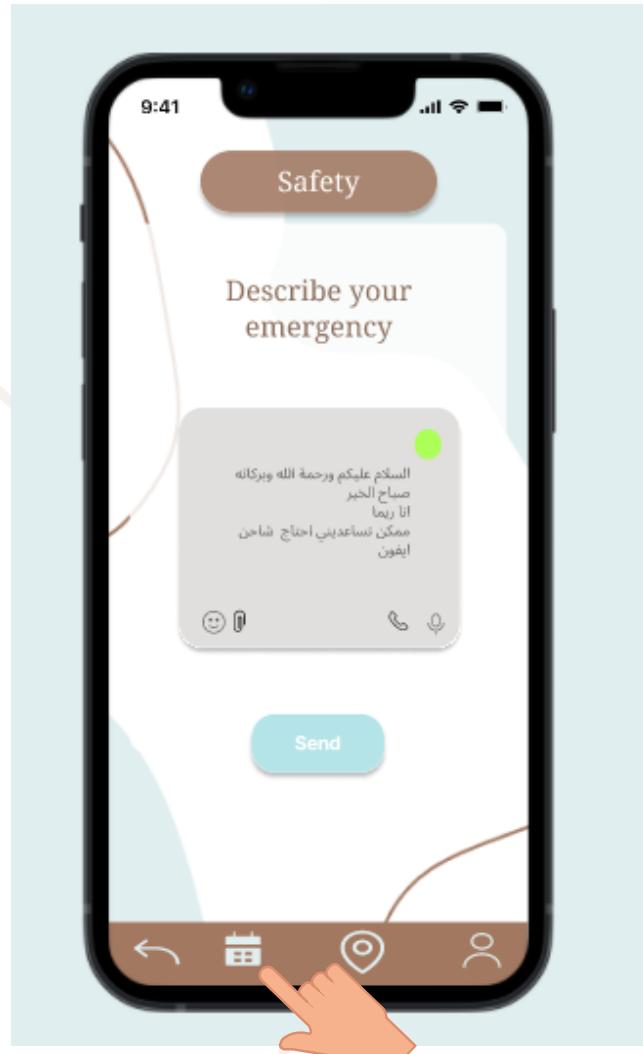
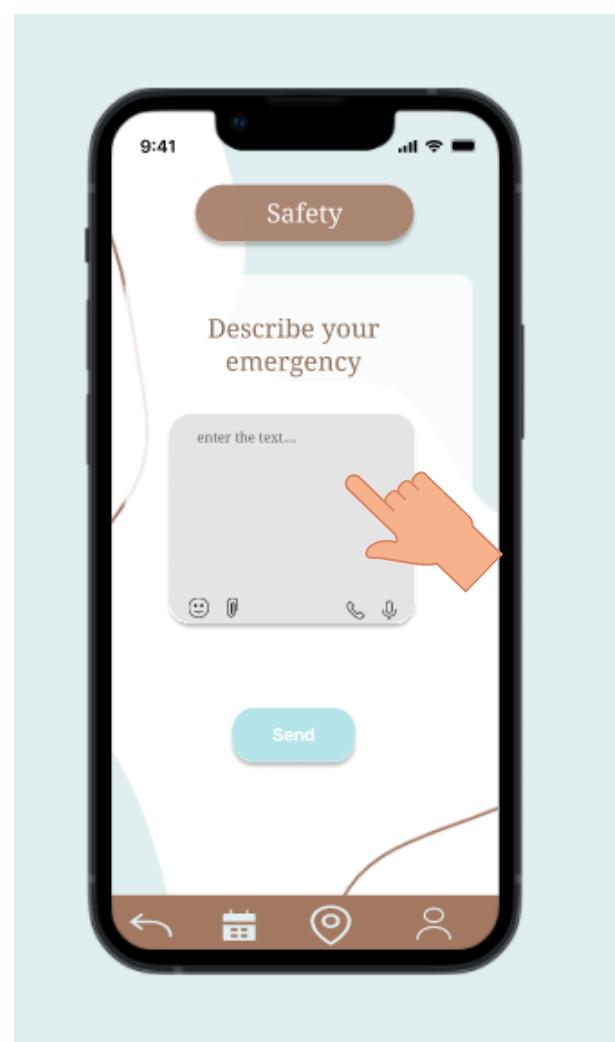
4.1



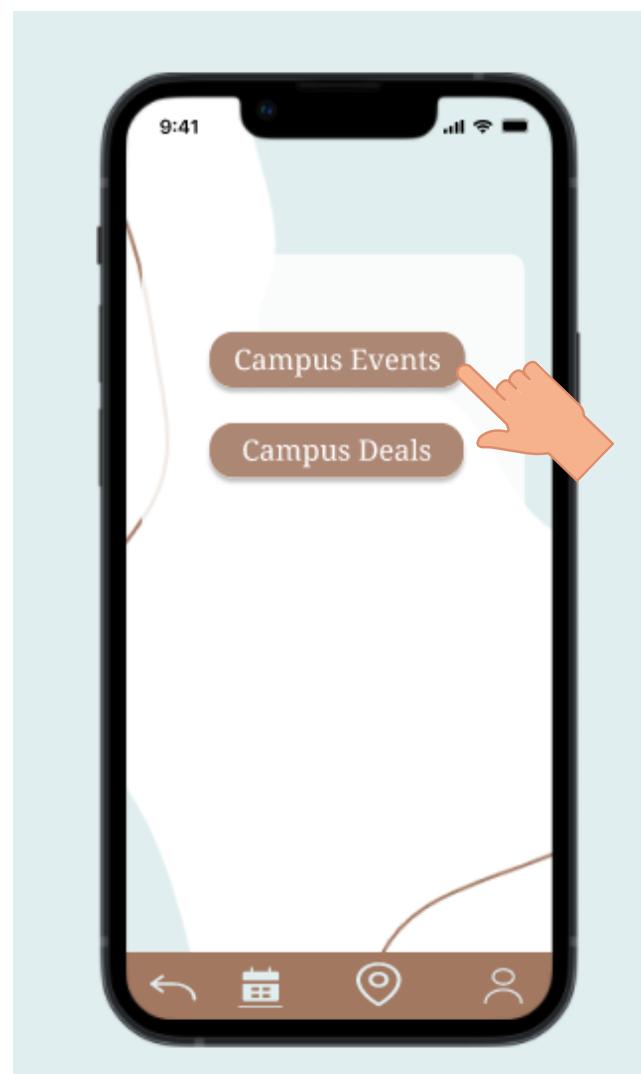
4.1



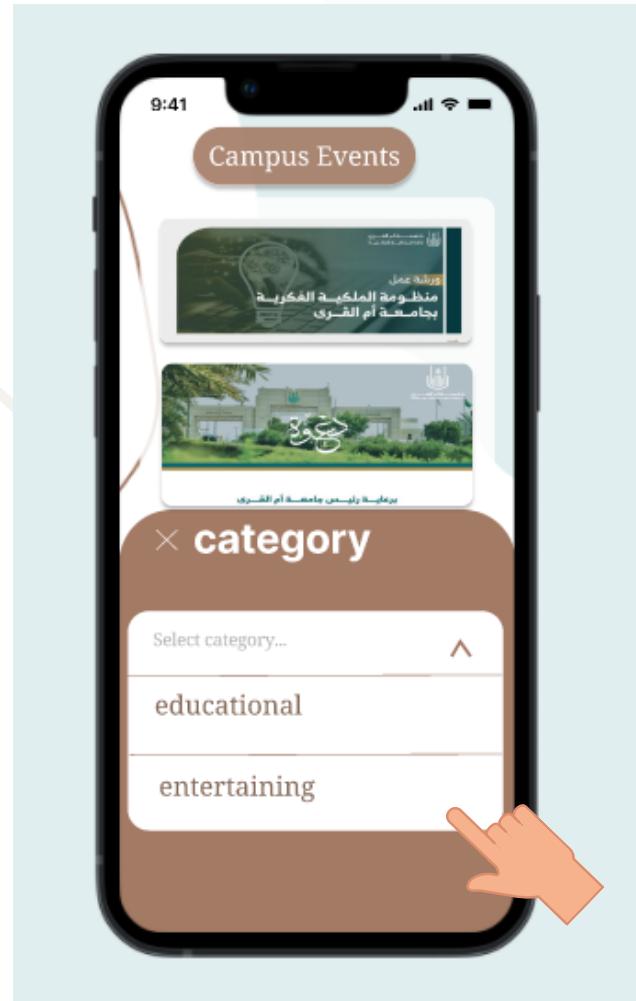
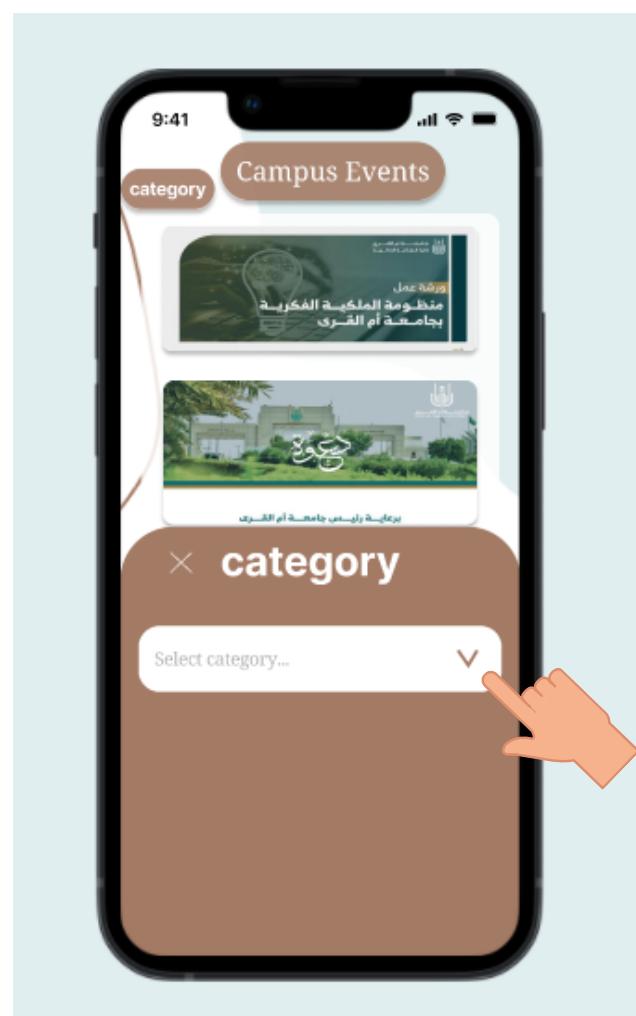
4.1



4.1



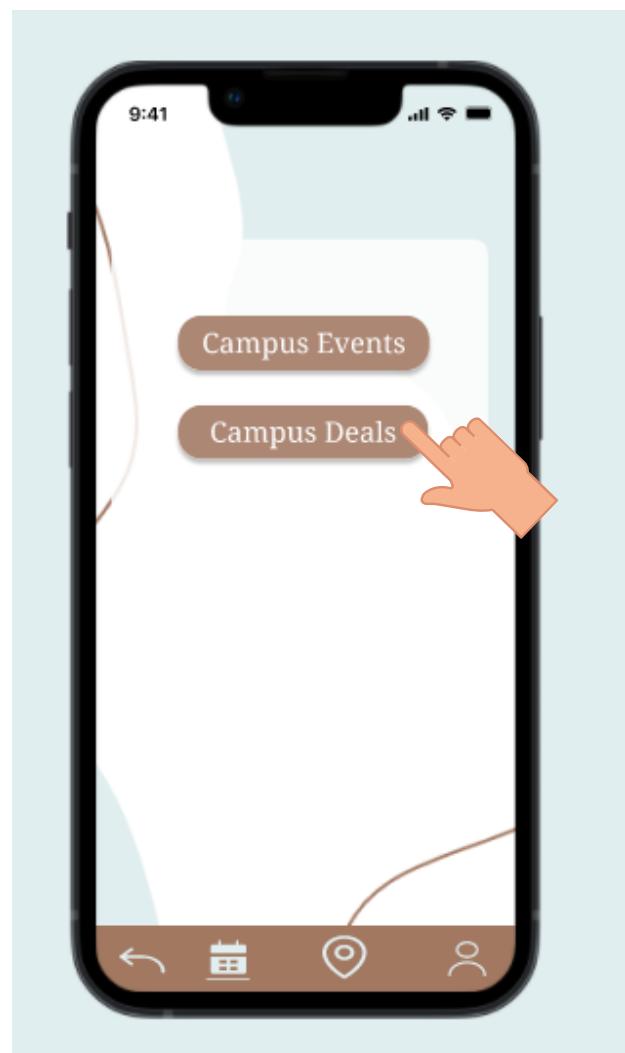
4.1



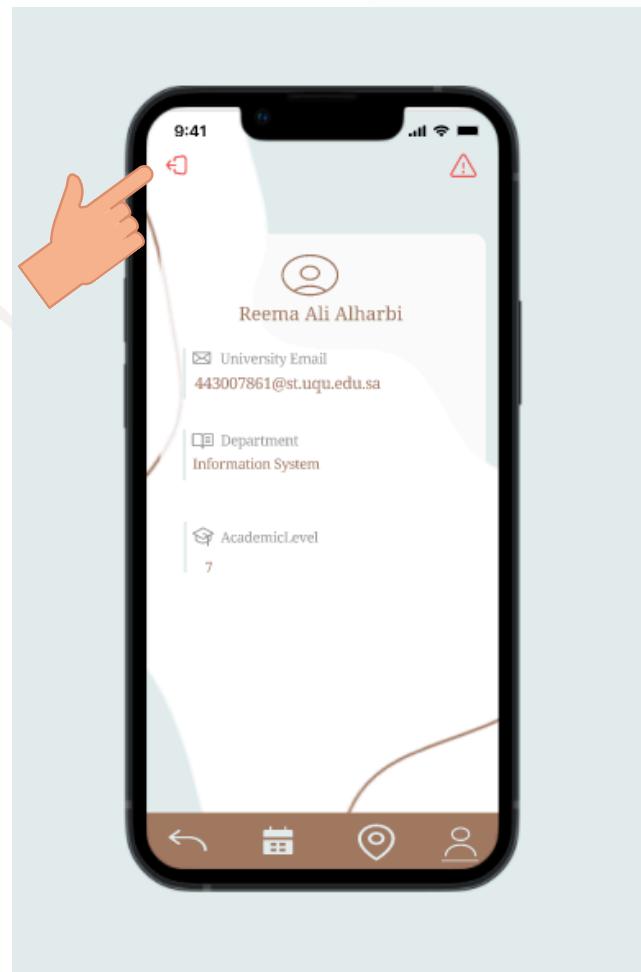
4.1



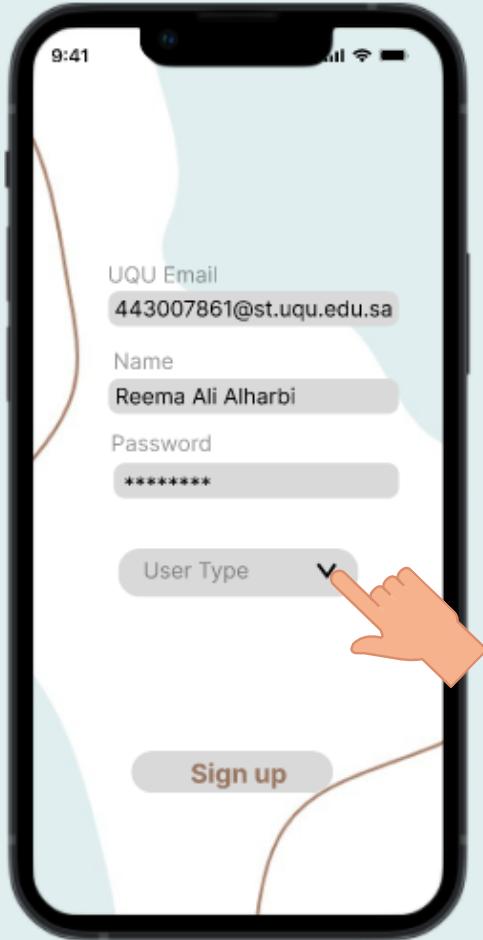
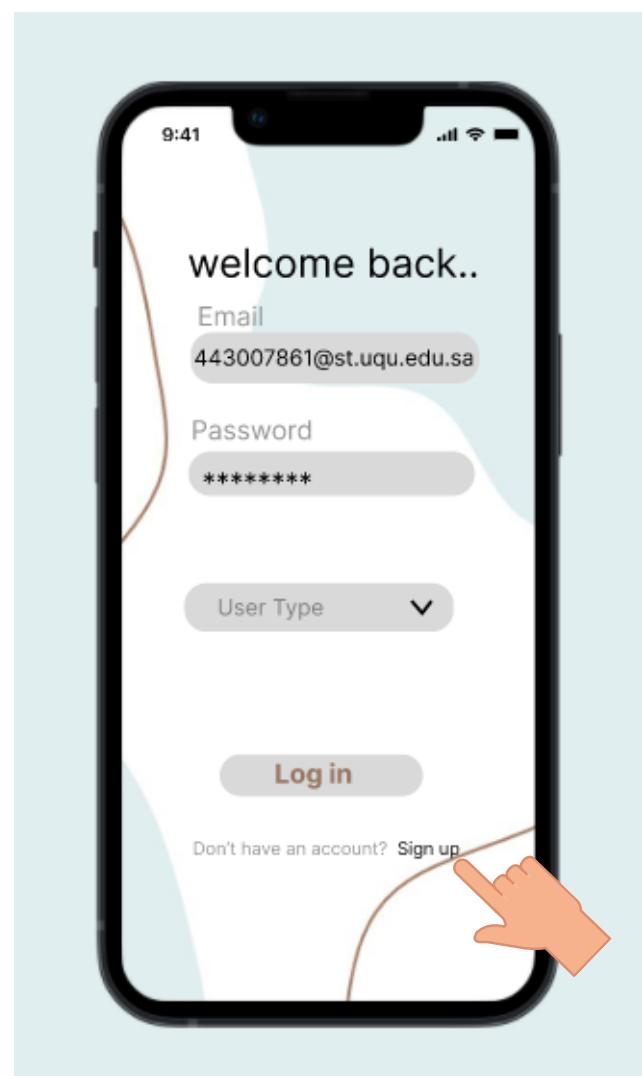
4.1



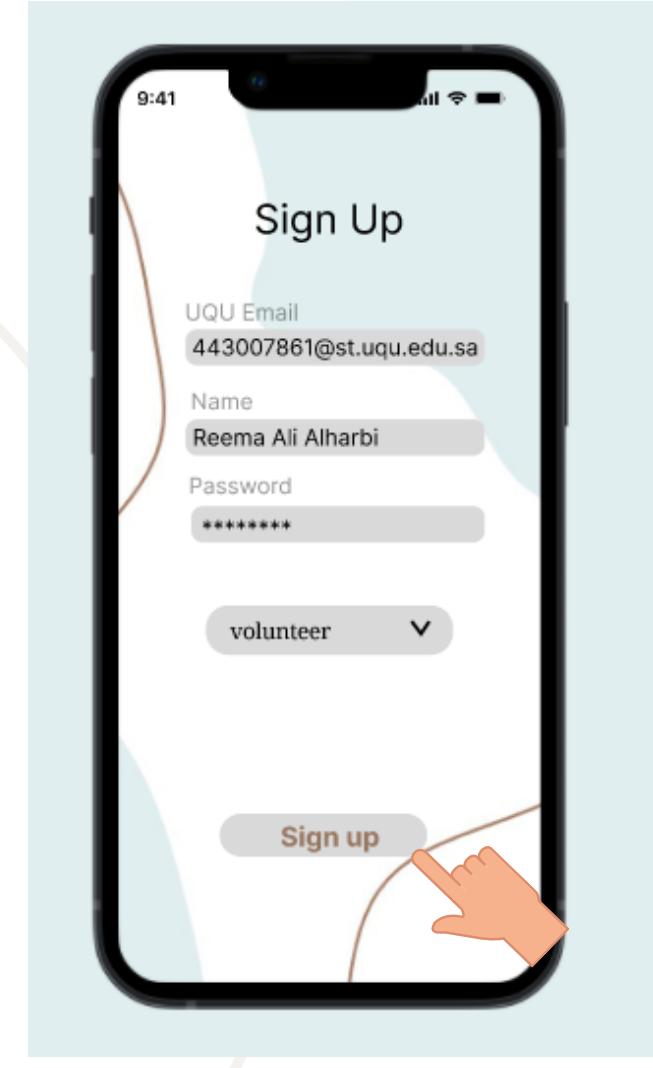
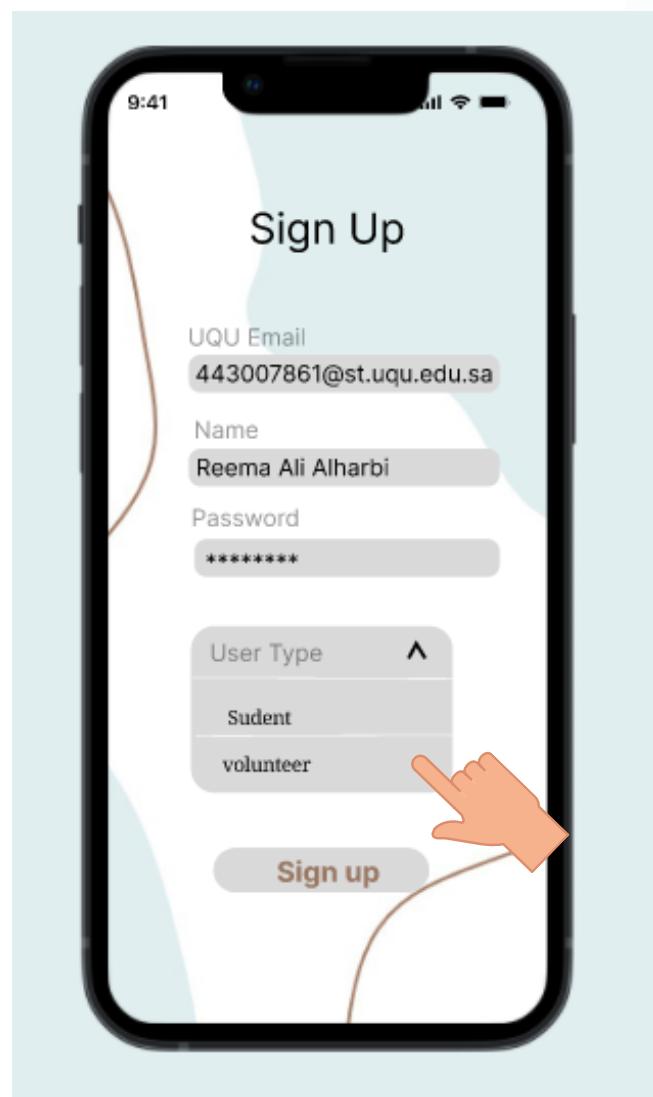
4.1



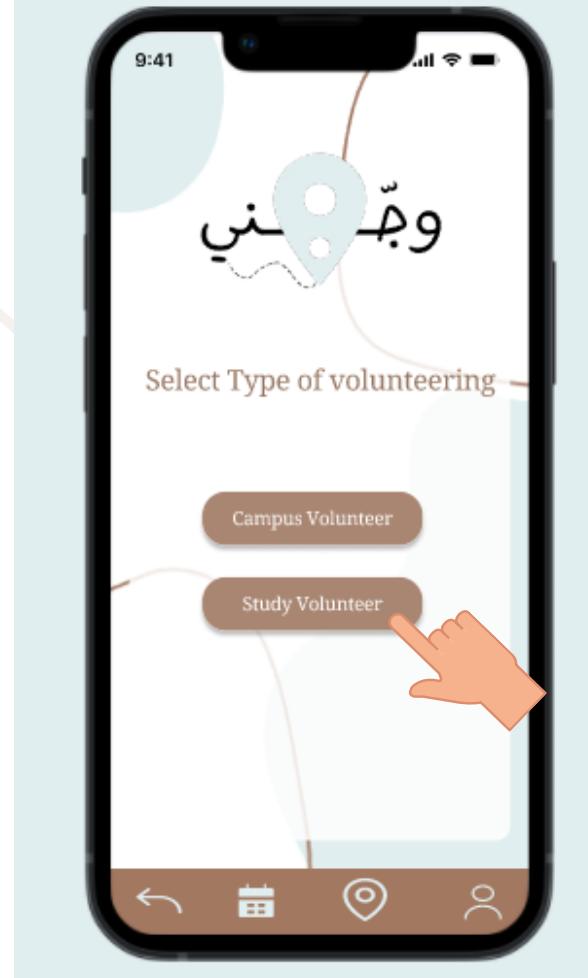
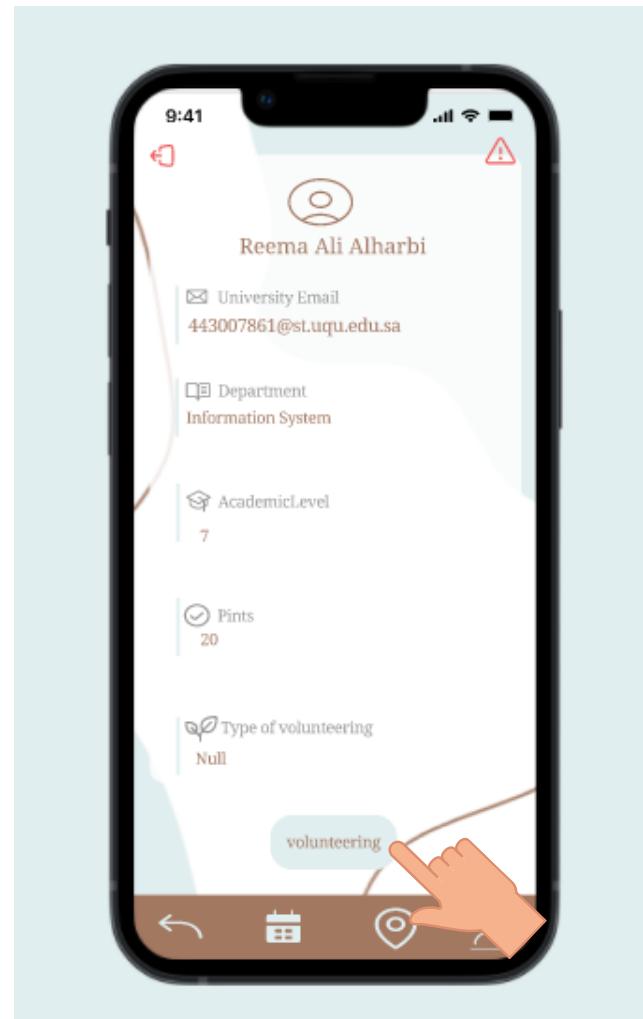
4.1



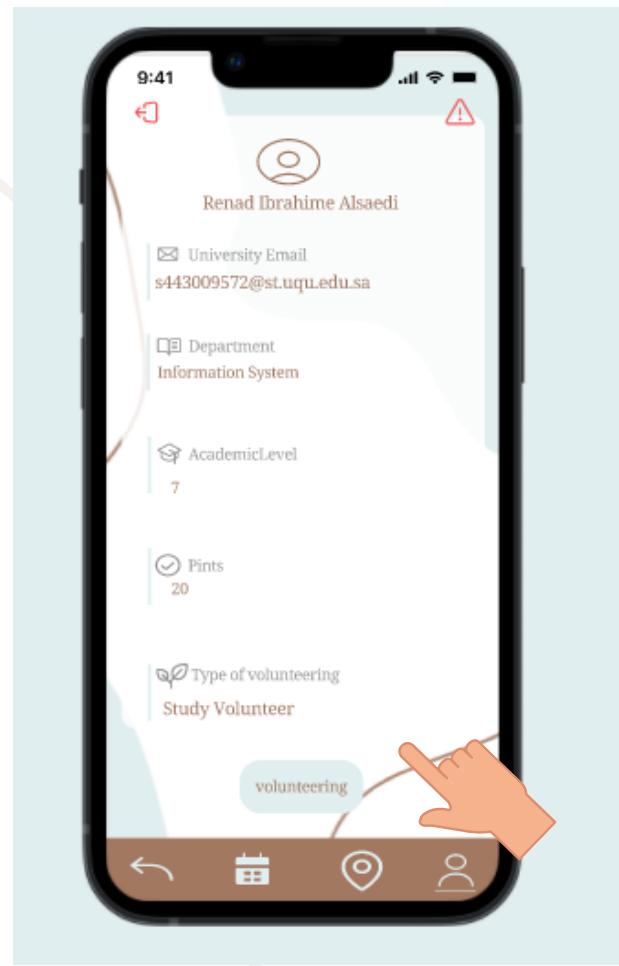
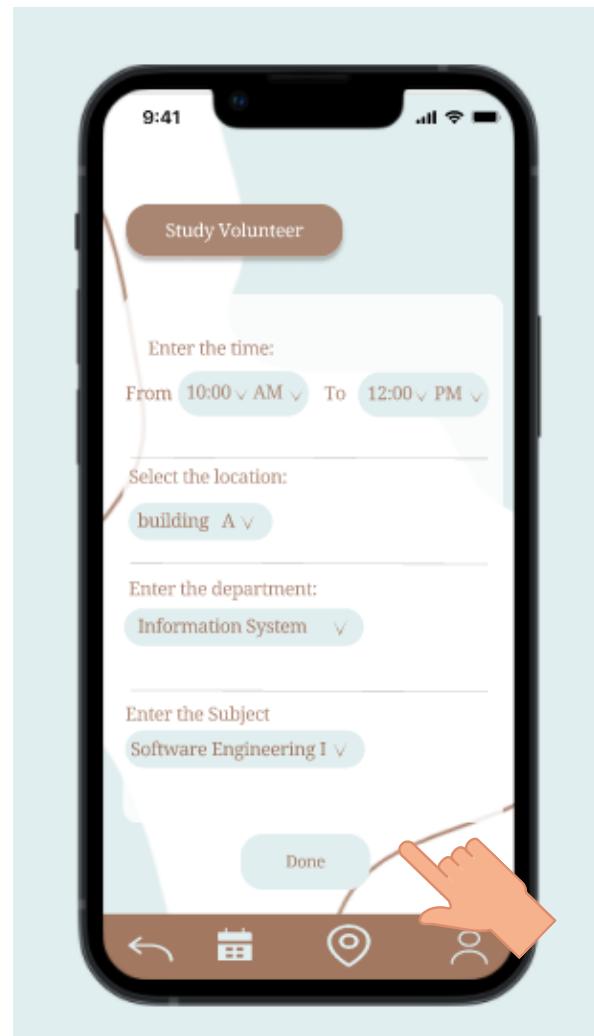
4.1



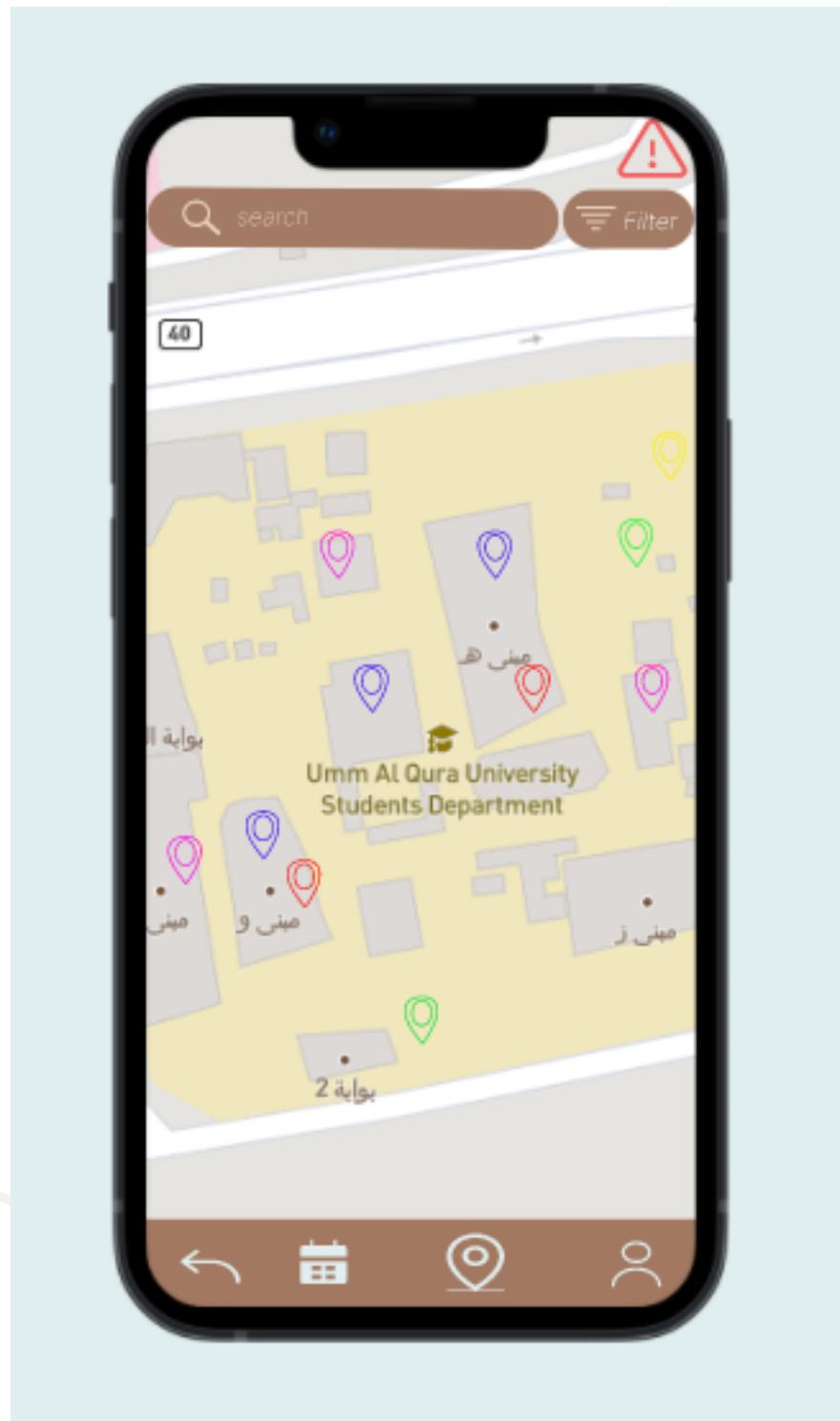
4.1



4.1

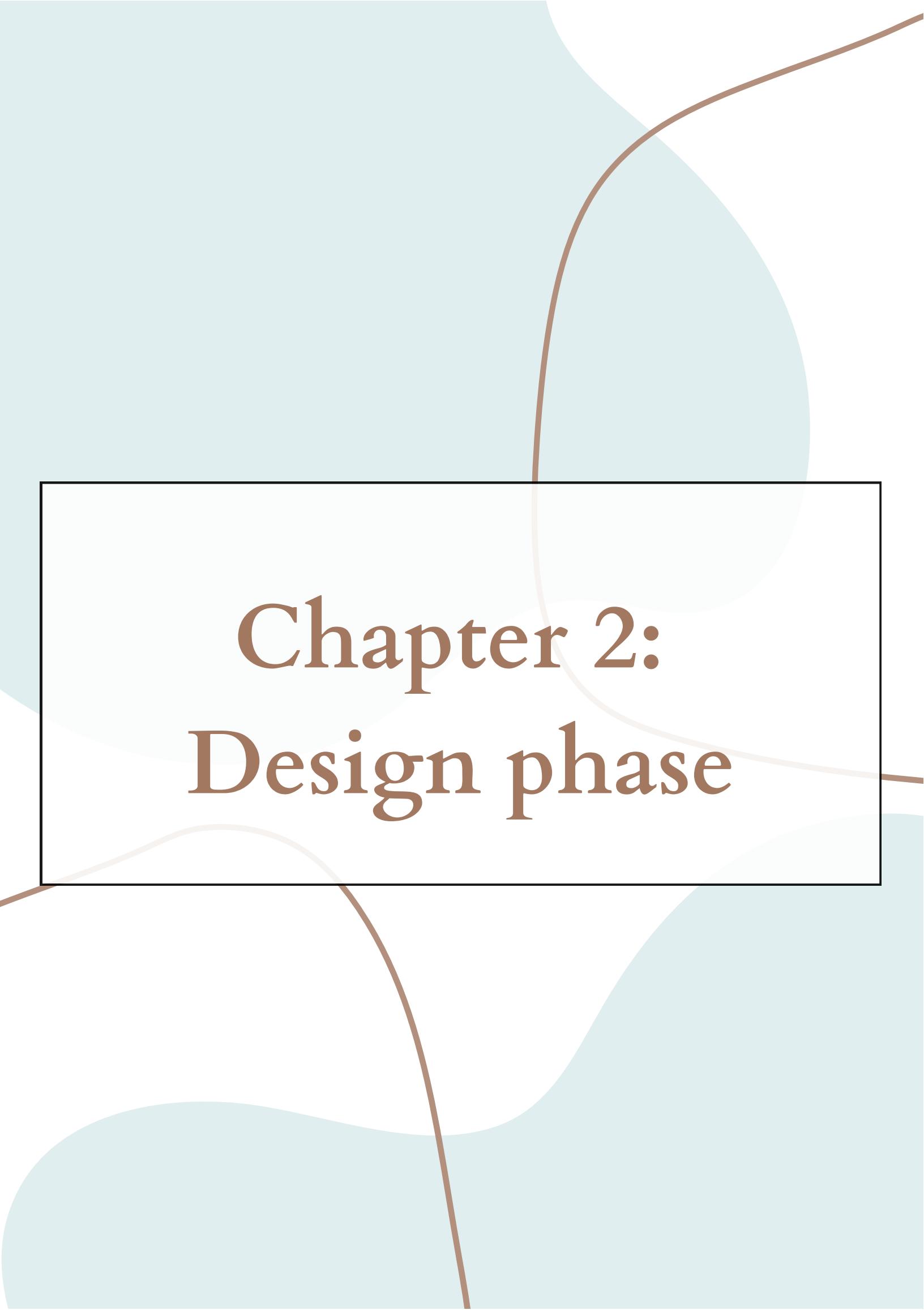


4.1





TO VIEW
OUR
PROTOTYPE

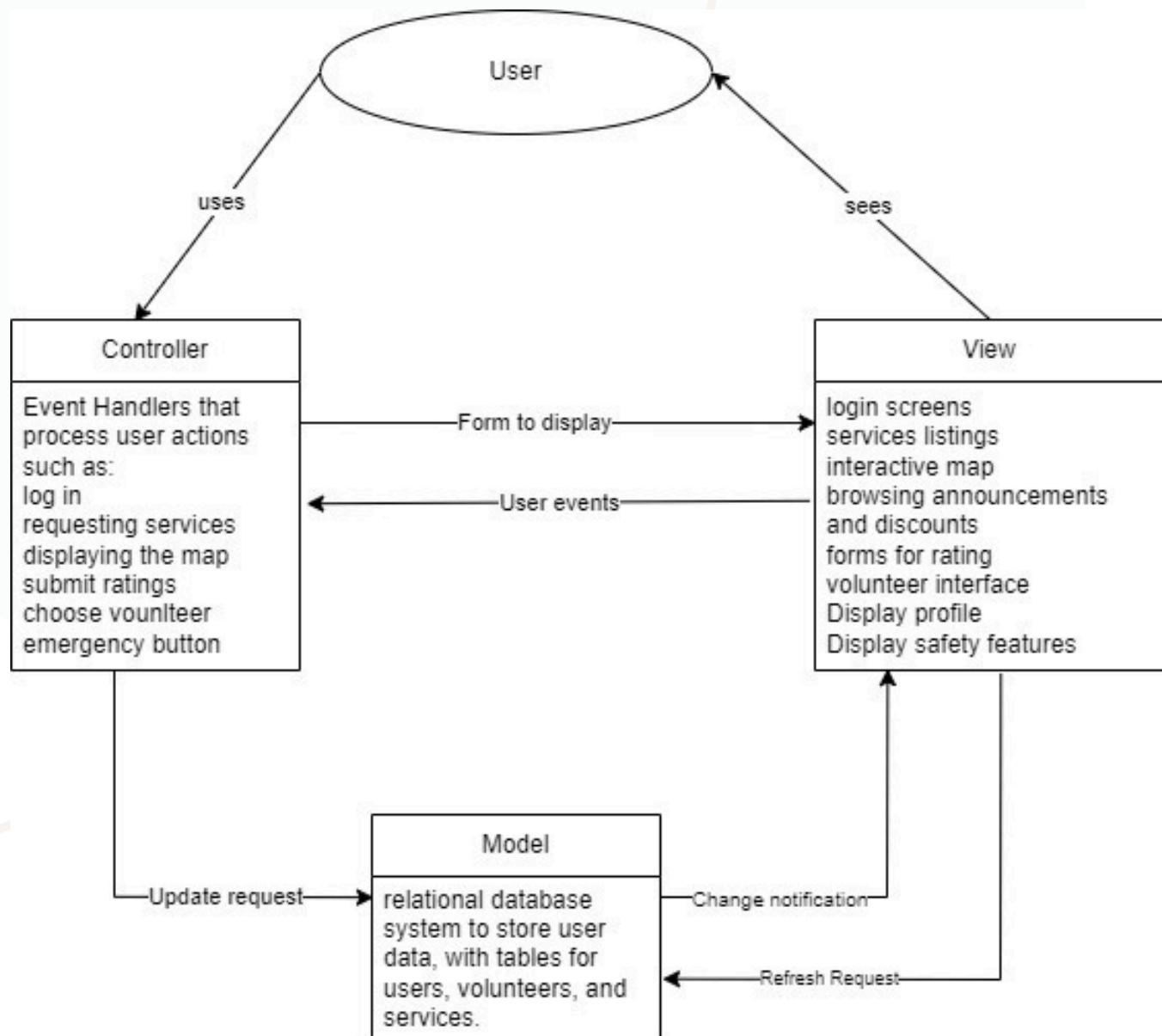


Chapter 2: Design phase

table tasks

TASK:	Renad	Ghala	Asma	Razan	Arwa	Atyaf
Architectural diagram				✓	✓	
Component Diagram		✓	✓			
Interface design	✓					✓
ERD	✓	✓	✓	✓	✓	✓

Architectural diagram:



[click here](#)

Using the specified requirements and applying them to the MVC pattern we'll describe each View and how it interacts with the rest of the system:

1-Login Screen

View: The interface design includes fields for entering username and password, along with a login button.

Controller: Handles incoming requests for login, verifies user data in the database.

Model: Stores user data in dedicated tables.

2-Services Listings

View: A list of available services displaying information such as map, profile, and service ratings.

Controller: Processes requests for browsing services and updates listings based on user interactions.

Model: Service tables contain all relevant information about the offered services.

3-Interactive Map

View: An interactive map showing locations of volunteer or events.

Controller: Manages map display and user interactions, such as selecting locations.

Model: Location data stored in the database can be used to display on the map.

4-Browsing Announcements and Discounts

View: Displays announcements and available discounts for users.

Controller: Handles updates and displays of announcements and discounts.

Model: Tables dedicated to announcements and discounts store relevant information.

5-Forms for Rating

View: Provides forms for users to rate volunteers .

Controller: Processes submission of ratings and updates them in the database.

Model: Tables contain ratings and comments from users.

6-Volunteer Interface

View: A dedicated interface for volunteers to manage their tasks and updates.

Controller: Handles all interactions related to volunteers, such as logging time and completed tasks.

Model: Tables specific to volunteers and their activities.

7-Display Profile

View: Displays the user's profile with information such as name, specialty, and other details.

Controller: Processes requests to update or display user profiles.

Model: Tables contain user information.

8-Display Safety Features

View: Presents safety features as emergency buttons.

Controller: Manages activation of safety features like emergency buttons.

Model: May include information about safety procedures and emergency responses.

All these interfaces interact with the Controller and Model to provide a seamless and efficient user experience while maintaining data safety and privacy.

We chose the MVC architecture

Because Layered architecture is suitable for programs that need layers of security, and our program does not need high security

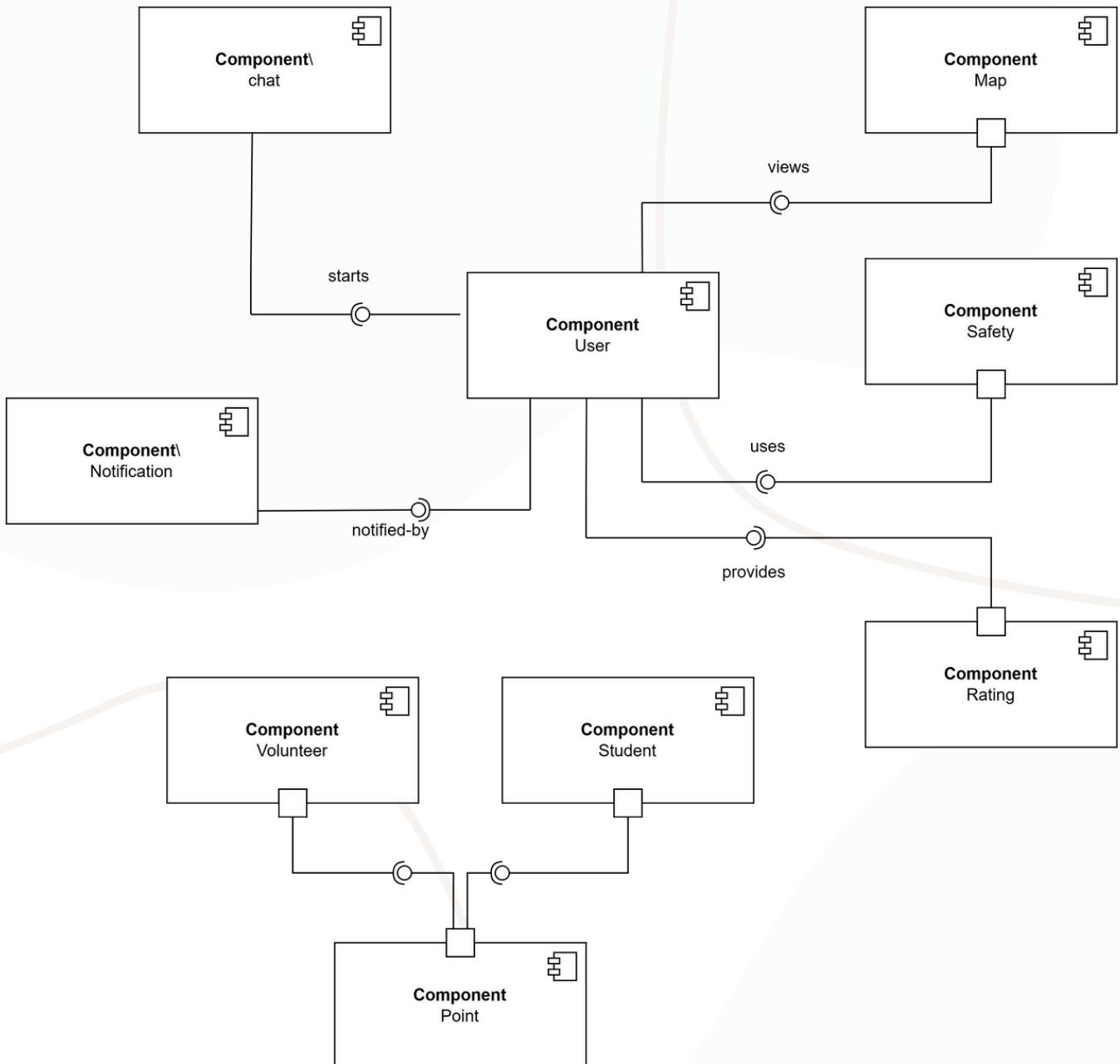
As for the Repository architecture, it is used when there is a large amount of data and our program does not contain much information

As for the Client-server architecture, it is used if there are several servers providing different services, and our program does not have many servers

As for the Pipe and filter architecture, it does not support interaction with the user, and our program relies heavily on interaction

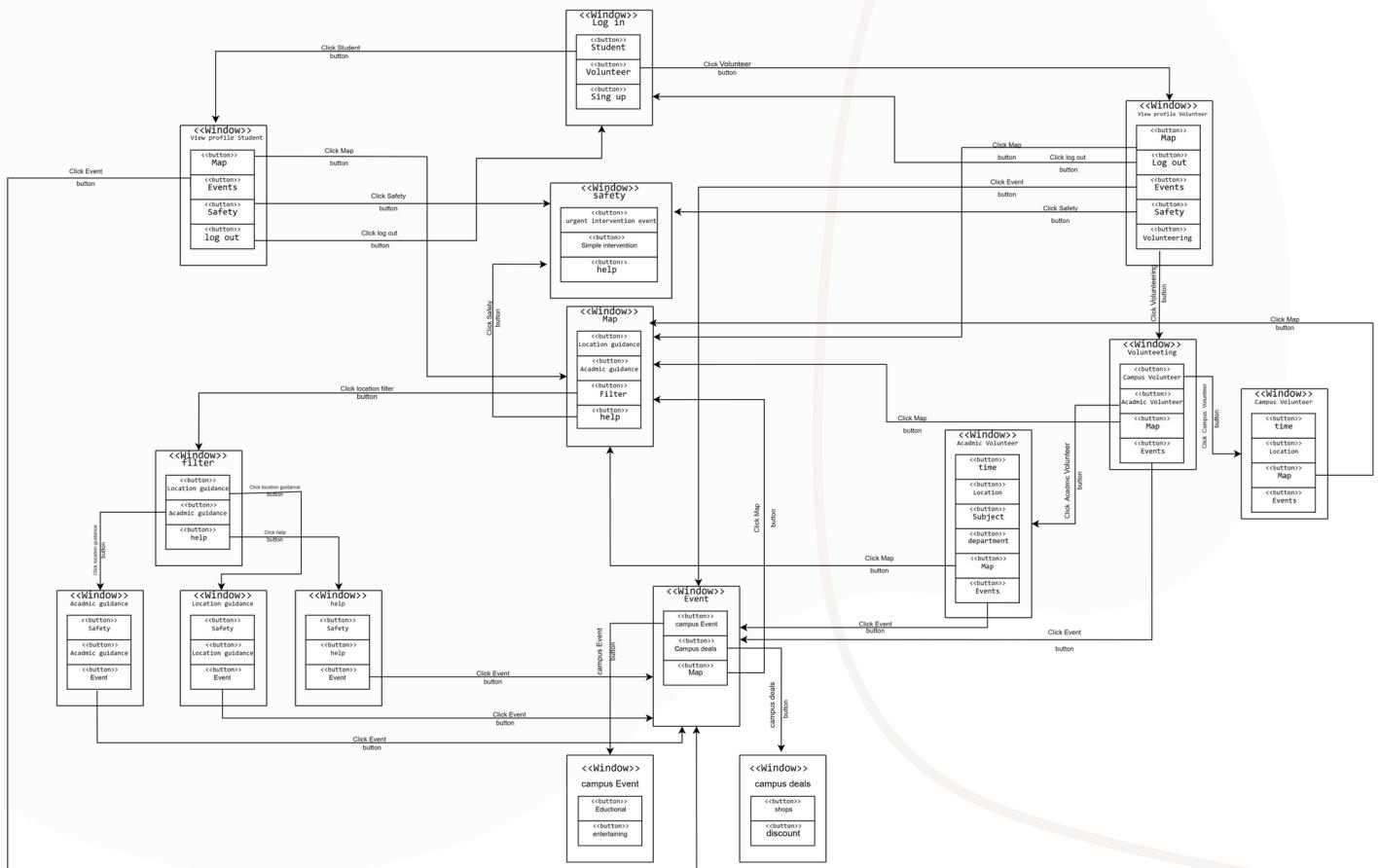
Therefore, we chose the most suitable architecture, which is the Model-View-Controller

Component Diagram:



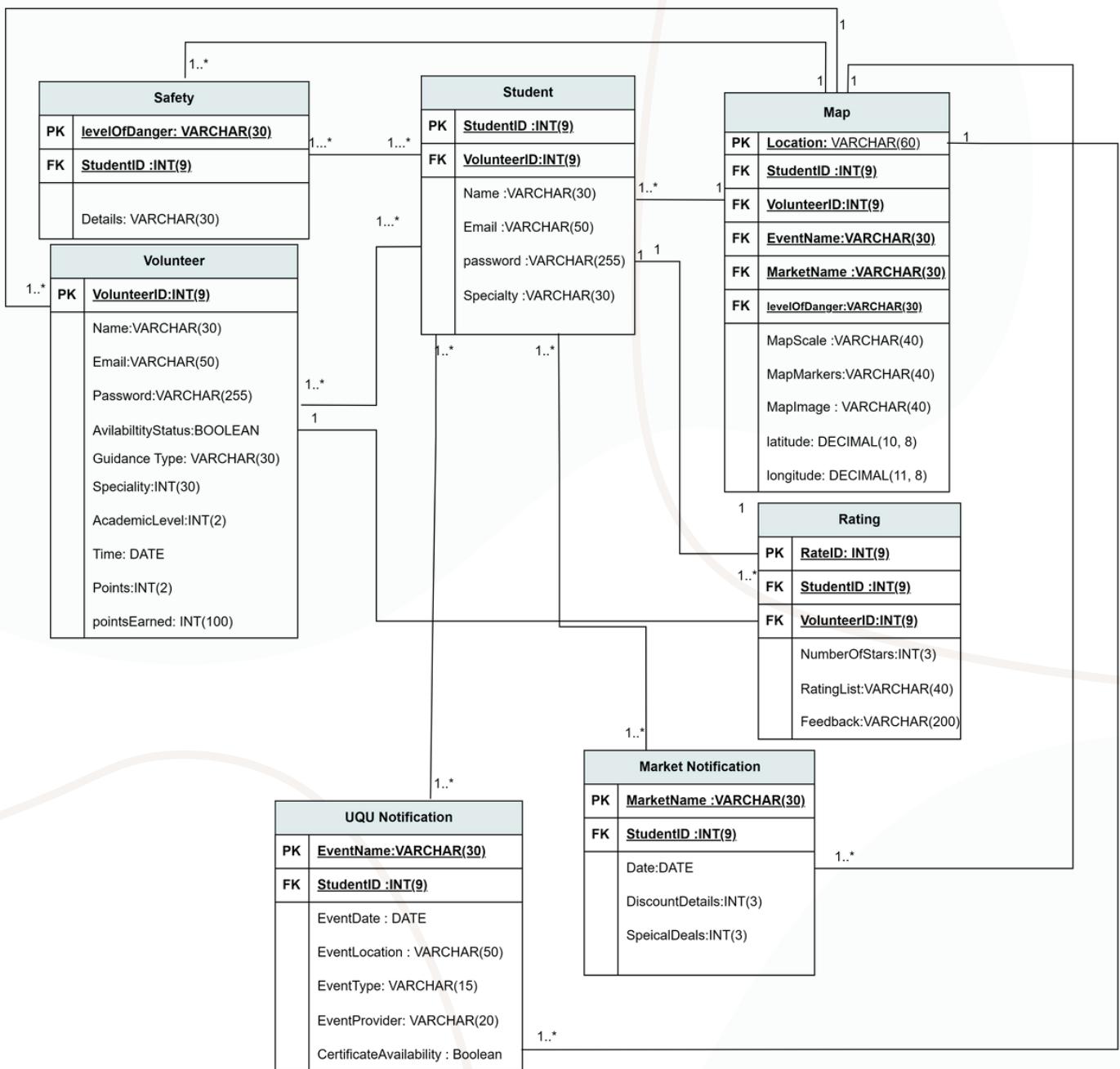
[click here](#)

Interface design:



click here

ERD:



click here

alteration

In the volunteer and student table, we changed the password size to 255 because when we were linking the system to the database, it did not allow this until we increased the password size to the maximum size.

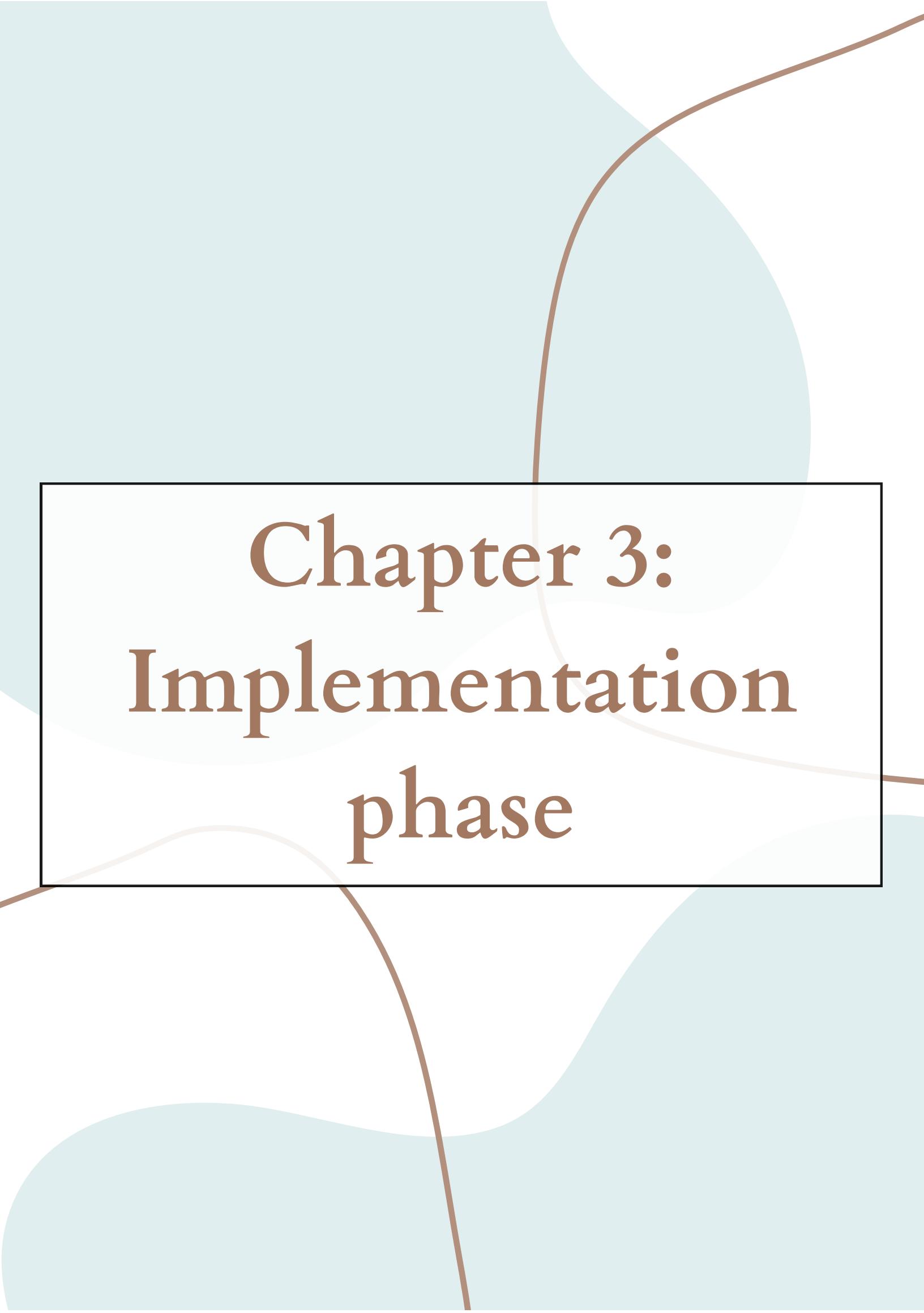
In the map table, we added a point attribute of longitude and latitude so that we can accurately determine the location of users

ERD:

In our project, volunteers are the backbone, assisting students with finding classrooms and understanding their lessons. We employ denormalization techniques to streamline information storage for easier access. By strategically duplicating essential data, we ensure quick retrieval without the need for complicated searches.

This approach simplifies and speeds up the process, benefiting both students and volunteers. With denormalization, important details like classroom locations and lesson information are readily available, reducing the time spent searching, ensuring that everyone finds what they need swiftly and effortlessly.

Moreover, by utilizing denormalization, we enhance the scalability of our system, accommodating future growth without sacrificing performance. This ensures that our project remains effective and efficient as it continues to evolve.

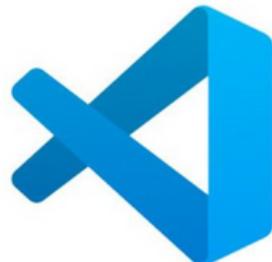


Chapter 3: Implementation phase

table tasks

TASK:	Renad	Ghala	Asma	Razan	Arwa	Atyaf
Map	✓					
Login		✓				
Campus Event				✓		
Safety						✓
Rating and Feedback					✓	
Volunteer form			✓			
Connecting to Database	✓	✓	✓	✓	✓	✓

Tools used in implementation and connection to the database.



Visual Studio Code



XAMPP



ChatGPT

We write the code from scratch but we used Chat GPT in some parts of the code to make the code more readable and more professional

Log in

by Ghala

وجّه سني

Email
Password
Select user type

Log in

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

Log in

by Ghala

HTML

```
<!DOCTYPE html>
<html>

<head>
    <title>Login Demo</title>
    <link rel="stylesheet" href="demo.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

<body>

    <div class="wrapper">
        <form onsubmit="return validateForm()" method="POST" action="http://localhost/W/Dphp.php">
            

            <div class="input-cont">
                <input type="email" id="email" name="email" placeholder="Email" required>
            </div>

            <div class="input-cont">
                <input type="password" id="password" name="password" placeholder="Password" required>
            </div>

            <div class="input-cont">
                <select id="usertype" name="usertype" required>
                    <option value="" disabled selected>Select user type</option>
                    <option value="student">Student</option>
                    <option value="volunteer">Volunteer</option>
                </select>
            </div>
        </form>
    </div>

```

```
        <button type="submit" class="bt">Log in</button>

        <p class="sentence">Don't have an account? <a href="invVolunteer.html">Sign up</a></p>
    </div>

    <script>
        function validateForm() {
            var email = document.getElementById("email").value;
            var password = document.getElementById("password").value;

            // Email validation
            var emailPattern = /^[^@\s]+@[^\s]+\.[^\s]+\$/;
            if (!emailPattern.test(email)) {
                alert("Please enter a valid email address.");
                return false;
            }

            // Password validation
            if (password.length < 8) {
                alert("Password must be at least 8 characters long.");
                return false;
            }
            alert("Form submitted successfully.");
            return true;
        }
    </script>

```

Log in

by Ghala

CSS

```
body {
    font-family: Arial, sans-serif; background-image: url('pics/BGse.png');
    background-size: cover; background-position: center; background-repeat: no-repeat;
}

.wrapper {
    display: flex; align-items: center;
    justify-content: center; height: 100vh; text-align: center;
}
.logo {
    margin-bottom: 20px;
}

.input-cont {
    width: 250px; max-width: 300px; display: flex;
    flex-direction: row; align-items: center; text-align: center;
    margin-bottom: 10px; margin-right: auto; margin-left: auto;
}

input {
    width: 100%; padding: 10px;
}

form {
    text-align: center;
}
.sentence {
    margin-top: 20px; color: #a7a7a7;
}
```

```
.sentence a {
    color: #rgb(33, 33, 33); text-decoration: none;
}
.sentence a:hover {
    text-decoration: underline;
}
.select-container {
    position: relative; width: 200px;
}

select {
    width: 100%; padding: 10px; color: #000; border: 1px solid #070303;
    border-radius: 4px; appearance: none;
}

select:focus {
    outline: none;
}

select option {
    color: #gray;
}

.bt {
    width: 40%; height: 50px; background: #E1EEEF; border: none;
    border-radius: 40px; box-shadow: 0 0 5px #rgb(0, 0, .1);
    cursor: pointer; font-size: 16px; color: #A27861;
    font-weight: 500; margin-bottom: 1%; margin-top: 10%;
```

Log in

by Ghala

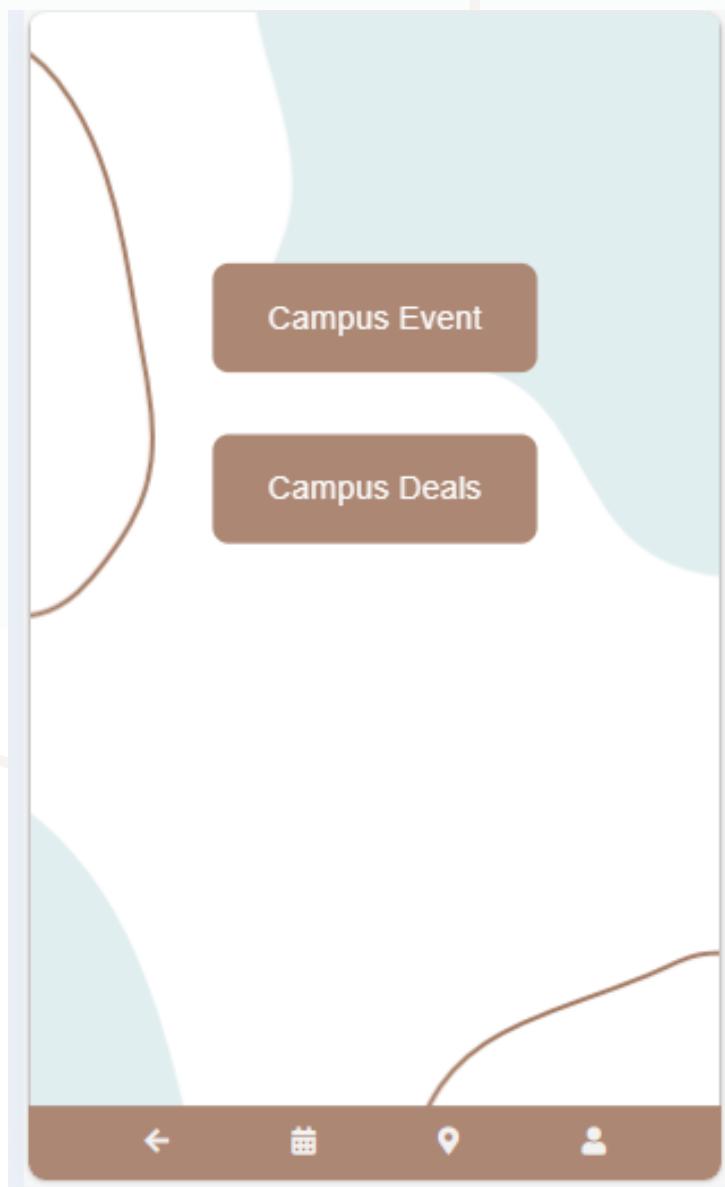
PHP

Connection to the Database

```
Dphp.php
1 <?php
2 error_reporting(E_ALL); ini_set('display_errors', 1);
3 $email = $_POST['email'];
4 $userPassword = $_POST['password']; // Change variable name to avoid conflict
5 // Database credentials
6 $hostname = "localhost"; // Change this to your database hostname
7 $username = "root"; // Change this to your database username
8 $password = "Razan443000"; // Change this to your database password
9 $database = "Wajjehni2"; // Change this to your database name
10 // Create connection
11 $connection = mysqli_connect($hostname, $username, $password, $database);
12 // Check connection
13 if (!$connection) {
14     die("Connection failed: " . mysqli_connect_error());
15 } else {
16     echo "Connected successfully";
17 }
18
19 $sql = "SELECT * FROM Student WHERE Email='$email' AND password='$userPassword'"; // Change variable name to avoid
20 $result = mysqli_query($connection, $sql);
21
22 if (mysqli_num_rows($result) > 0) {
23     header("Location: test.html");
24     exit();
25 } else {
26     // User not found or password incorrect
27     echo "Login failed. Please check your email and password.";
28 }
29 mysqli_close($connection);
30 ?>
31
```

Campus event

by Razan



Campus event by Razan

CSS

```
1  /* Style for the navigation bar */
2  .navbar {
3      position: fixed; bottom: 0; left: 0; width: 100%; background-color: #AC8774;
4      text-align: center; /* Align text in the center */
5  }
6  /* Style for the navigation bar links */
7  .navbar a {
8      display: inline-block; color: #f2f2f2; padding: 14px 36px; text-decoration: none;
9  }
10 /* Change color on hover */
11 .navbar a:hover {
12     background-color: #ddd; color: black;
13 }
14 /* Background image */
15 body {
16     background-image: url('Sc1.png'); /* URL of the background image */ background-size: cover; /* Cover the entire background */
17     background-position: center; /* Center the background image */ background-repeat: no-repeat; /* Do not repeat the background image */
18     background-attachment: fixed; /* Fixed background image */ text-align: center; /* Aligning the content of the entire page to the center */
19     margin: 0; /* Remove default margin */ padding: 0; /* Remove default padding */
20 }
21 button {
22     margin: 10px; background-color: #AC8774; /* Change button color #E1EEE9*/ border: none; /* Remove button border */
23     padding: 22px 34px; /* Increase button padding */ color: white; /* button text color*/ font-size: 20px; /* Increase button font size */
24     font-family: Arial, sans-serif; /* Change button font */ cursor: pointer; /* Change cursor to pointer on hover */
25     border-radius: 10px; /* Make button edges round */ transition: background-color 0.3s; /* Smooth transition for hover effect */
26 }
27 /* Hover effect */
28 button:hover {
29     background-color: #B3D6E6; /* Change button color on hover */
30 }
```

HTML

```
W1.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6      <link rel="stylesheet" href="W1.css" >
7      <script src="W1.js" ></script>
8      <title>Campus Event and Campus Deals</title>
9      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
10 </head>
11 <body>
12     <!-- Image -->
13     <br><br>
14     <br><br>
15     <br><br>
16     <br><br>
17     <!-- Button for Campus Event -->
18     <button onclick="selectOption('event')>Campus Event</button>
19     <br><br>
20     <!-- Button for Campus Deals -->
21     <button onclick="selectOption('deals')>Campus Deals</button>
22     <!-- Navigation Bar -->
23     <div class="navbar">
24         <a href="test.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
25         <a href="W1.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
26         <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
27         <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
28     </div>
29     </body>
30 </html>
```

Campus event

by Razan



Campus event

by Razan

CSS

```
Insert.php map.php test.html W1.html W1.css W2.html W2.css
W2.css > .navbar
1  /* Style for the navigation bar */
2  .navbar {
3      position: fixed;
4      bottom: 0;
5      left: 0;
6      width: 100%;
7      background-color: #AC8774;
8      text-align: center; /* Align text in the center */
9
10 /* Style for the navigation bar links */
11 .navbar a {
12     display: inline-block;
13     color: #f2f2f2;
14     padding: 14px 36px;
15     text-decoration: none;
16 }
17 /* Change color on hover */
18 .navbar a:hover {
19     background-color: #ddd;
20     color: black;
21 }
22 /* Background image */
23 body {
24     background-image: url('Sc1.png'); /* URL of the background image */
25     background-size: cover; /* Cover the entire background */
26     background-position: center; /* Center the background image */
27     background-repeat: no-repeat; /* Do not repeat the background image */
28     background-attachment: fixed; /* Fixed background image */
29     text-align: center; /* Aligning the content of the entire page to the center */
30     margin: 0; /* Remove default margin */
31     padding: 0; /* Remove default padding */
32 }
```

```
W2.css > .navbar
34 /* Style for the colorful frame */
35 .title {
36     background-color: #AC8774; /* Brown background color */
37     padding: 12px; /* Add padding */
38     display: inline-block; /* Make it inline block to fit the content */
39     border: 2px solid #E1EEE9; /* Add border around the frame */
40     border-radius: 16px; /* Make border edges round */
41 }
42
43 /* White text color for the frame content */
44 .title h2 {
45     color: white; /* White text color */
46 }
47 /* Style the dropdown button */
48 .dropbtn {
49     background-color: #AC8774; /* New color */
50     color: white; /* Text color is white */
51     padding: 16px;
52     font-size: 16px;
53     border: none;
54     cursor: pointer;
55     border-radius: 10px; /* Rounded corners */
56 }
57
58 /* The container <div> - needed to position the dropdown content */
59 .dropdown {
60     position: fixed; /* Keeps the dropdown at the top left */
61     top: 0;
62     left: 0;
63     display: inline-block;
64 }
```

Campus event

by Razan

CSS

```
W2.css > .navbar
66  /* Dropdown Content (Hidden by Default) */
67  .dropdown-content {
68    display: none;
69    position: absolute;
70    background-color: #f9f9f9;
71    min-width: 160px;
72    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
73    z-index: 1;
74  }
75
76  /* Links inside the dropdown */
77  .dropdown-content a {
78    color: black;
79    padding: 12px 16px;
80    text-decoration: none;
81    display: block;
82  }
83
84  /* Change color of dropdown links on hover */
85  .dropdown-content a:hover {
86    background-color: #f1f1f1;
87  }
88
89  /* Show the dropdown menu on hover */
90  .dropdown:hover .dropdown-content {
91    display: block;
92  }
93
94  /* Change the background color of the dropdown button when the dropdown content is shown */
95  .dropdown:hover .dropbtn {
96    background-color: #92705f; /* A slightly darker shade for hover effect */
97  }
```

```
W2.css > .image-container
99  .image-container {
100    display: flex;
101    flex-direction: column; /* Stack images vertically */
102    gap: 30px; /* Spacing between images */
103    padding: 20px;
104    justify-content: center; /* Center items vertically */
105    align-items: center; /* Center items horizontally */
106    height: 80%;
107  }
108
109
110  .image-container img {
111    width: 70%; /* Makes images responsive */
112    transition: transform 0.2s; /* Smooth transform effect */
113    border: none;
114    cursor: pointer;
115    border-radius: 8px;
116  }
117
118  .image-container img:hover {
119    transform: scale(1.1); /* Slightly enlarges the image on hover */
120  }
121
122
```

Campus event

by Razan

HTML

```
W2.html > html > body > div.image-container > img
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <link rel="stylesheet" href="W2.css" >
6  <title>Campus Event and Campus Deals</title> <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fo
7  </head>
8  <body>
9  <div class="title">
10 | <h2>Campus Event</h2>
11 </div>
12 <div class="dropdown">
13 | <button class="dropbtn">category</button>
14 | <div class="dropdown-content">
15 | | <a href="W3.html">entertaining</a>
16 | | <a href="#">educational </a>
17 |
18 | </div>
19 </div>
20 <div class="image-container">
21 |   
25 | <a href="W1.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
26 | <a href="W1.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
27 | <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
28 | <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
29 </div>
30 </body>
31 </html>
```

Campus event

by Razan



Campus event

by Razan

CSS

The screenshot shows a code editor with two tabs open: 'W2.css' and 'W2.css .navbar'. The top bar includes tabs for 'Insert.php', 'map.php', 'test.html', 'W1.html', 'W1.css', 'W2.html', and 'W2.css'. The 'W2.css' tab is active.

W2.css:

```
/* Style for the navigation bar */
.navbar {
    position: fixed;
    bottom: 0;
    left: 0;
    width: 100%;
    background-color: #AC8774;
    text-align: center; /* Align text in the center */
}

/* Style for the navigation bar links */
.navbar a {
    display: inline-block;
    color: #f2f2f2;
    padding: 14px 36px;
    text-decoration: none;
}
/* Change color on hover */
.navbar a:hover {
    background-color: #ddd;
    color: black;
}

/* Background image */
body {
    background-image: url('Sci.png'); /* URL of the background image */
    background-size: cover; /* Cover the entire background */
    background-position: center; /* Center the background image */
    background-repeat: no-repeat; /* Do not repeat the background image */
    background-attachment: fixed; /* Fixed background image */
    text-align: center; /* Aligning the content of the entire page to the center */
    margin: 0; /* Remove default margin */
    padding: 0; /* Remove default padding */
}
```

W2.css .navbar:

```
/* Style for the colorful frame */
.title {
    background-color: #AC8774; /* Brown background color */
    padding: 12px; /* Add padding */
    display: inline-block; /* Make it inline block to fit the content */
    border: 2px solid #E1EEE1; /* Add border around the frame */
    border-radius: 16px; /* Make border edges round */
}

/* White text color for the frame content */
.title h2 {
    color: white; /* White text color */
}

/* Style the dropdown button */
.dropbtn {
    background-color: #AC8774; /* New color */
    color: white; /* Text color is white */
    padding: 16px;
    font-size: 16px;
    border: none;
    cursor: pointer;
    border-radius: 10px; /* Rounded corners */
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
    position: absolute; /* Keeps the dropdown at the top left */
    top: 0;
    left: 0;
    display: inline-block;
}
```

Campus event

by Razan

CSS

```
W2.css > .navbar
66  /* Dropdown Content (Hidden by Default) */
67  .dropdown-content {
68    display: none;
69    position: absolute;
70    background-color: #f9f9f9;
71    min-width: 160px;
72    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
73    z-index: 1;
74  }
75
76  /* Links inside the dropdown */
77  .dropdown-content a {
78    color: black;
79    padding: 12px 16px;
80    text-decoration: none;
81    display: block;
82  }
83
84  /* Change color of dropdown links on hover */
85  .dropdown-content a:hover {
86    background-color: #f1f1f1
87  }
88
89  /* Show the dropdown menu on hover */
90  .dropdown:hover .dropdown-content {
91    display: block;
92  }
93
94  /* Change the background color of the dropdown button when the dropdown content is shown */
95  .dropdown:hover .dropbtn {
96    background-color: #92705f; /* A slightly darker shade for hover effect */
97  }
```

```
W2.css > .navbar
99  .image-container {
100    display: flex;
101    flex-direction: column; /* Stack images vertically */
102    gap: 30px; /* Spacing between images */
103    padding: 20px;
104    justify-content: center; /* Center items vertically */
105    align-items: center; /* Center items horizontally */
106    height: 80%;
107  }
108
109
110  .image-container img {
111    width: 70%; /* Makes images responsive */
112    transition: transform 0.2s; /* Smooth transform effect */
113    border: none;
114    cursor: pointer;
115    border-radius: 8px;
116  }
117
118  .image-container img:hover {
119    transform: scale(1.1); /* Slightly enlarges the image on hover */
120  }
121
122
```

Campus event

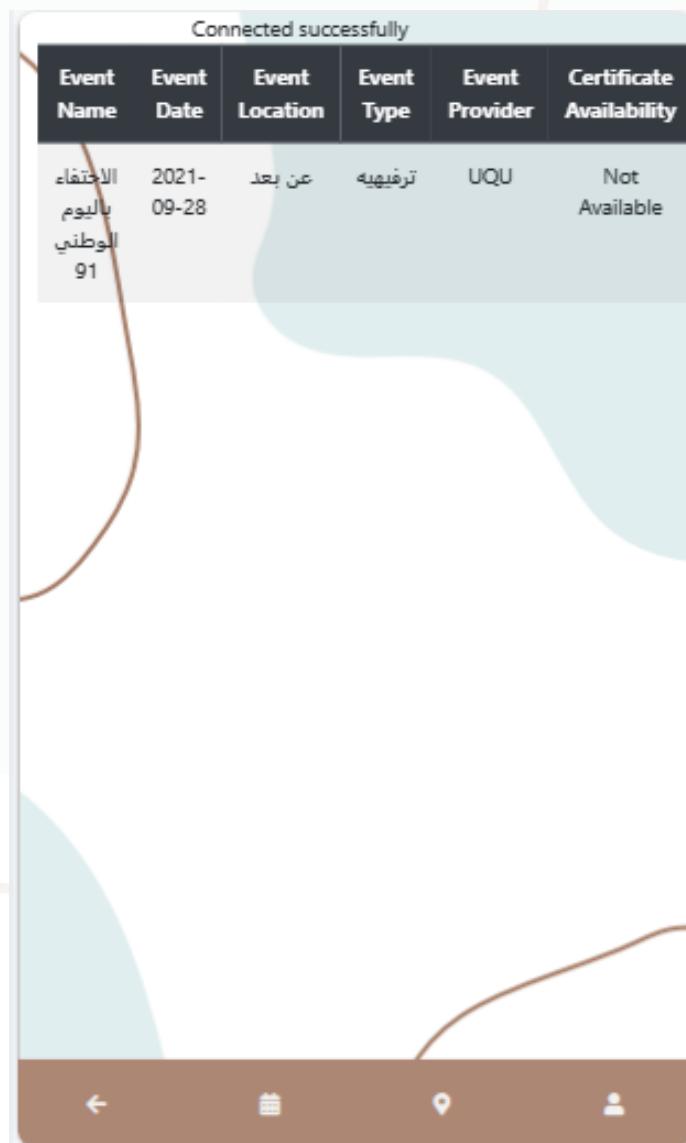
by Razan

HTML

```
W3.html > html > head > link
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <link rel="stylesheet" href="W3.css" >
7  <title>Campus Event and Campus Deals</title>
8  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
9  </head>
10 <body>
11 <div class="title">
12 | <h2>Campus Event</h2>
13 </div>
14 <div class="dropdown">
15 | <button class="dropbtn">category</button>
16 | <div class="dropdown-content">
17 | | <a href="#">educational</a>
18 | | <a href="#">entertaining</a>
19 | </div>
20 </div>
21 <div class="image-container">
22 | <a href="http://localhost/W/display_event.php"> </a>
23 </div>
24 <!-- Navigation Bar -->
25 <div class="navbar">
26 | <a href="W2.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
27 | <a href="W1.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
28 | <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
29 | <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
30 </div>
31 </body>
32 </html>
```

Campus event

by Razan



Campus event

by Razan

PHP & HTML

Connection to the Database

```
display_event.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Display Information</title>
7      <!-- Bootstrap CSS -->
8      <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
9      | <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
10     <style>
11         /* Style for the navigation bar */
12     .navbar {
13         position: fixed;
14         bottom: 0;
15         left: 0;
16         width: 100%;
17         background-color: #AC8774;
18         text-align: center; /* Align text in the center */
19     }
20     /* Style for the navigation bar links */
21     .navbar a {
22         display: inline-block;
23         color: #f2f2f2;
24         padding: 14px 36px;
25         text-decoration: none;
26     }
27     /* Change color on hover */
28     .navbar a:hover {
29         background-color: #ddd;
30         color: black;
31     }
32 
```

```
52     /* Background image */
53     body {
54         background-image: url('Sci.png'); /* URL of the background image */
55         background-size: cover; /* Cover the entire background */
56         background-position: center; /* Center the background image */
57         background-repeat: no-repeat; /* Do not repeat the background image */
58         background-attachment: fixed; /* Fixed background image */
59         text-align: center; /* Aligning the content of the entire page to the center */
60         margin: 0; /* Remove default margin */
61         padding: 0; /* Remove default padding */
62     }
63     </style>
64 </head>
65 <div class="navbar">
66     <a href="W3.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon --&gt;
67     &lt;a href="M1.html"&gt;&lt;i class="fas fa-calendar-alt"&gt;&lt;/i&gt;&lt;/a&gt; <!-- Calendar icon --&gt;
68     &lt;a href="test.html"&gt;&lt;i class="fas fa-map-marker-alt"&gt;&lt;/i&gt;&lt;/a&gt; <!-- Location icon --&gt;
69     &lt;a href="#"&gt;&lt;i class="fas fa-user"&gt;&lt;/i&gt;&lt;/a&gt; <!-- Profile icon --&gt;
70 &lt;/div&gt;
71 &lt;body&gt;
72 &lt;?php
73 // Database credentials
74 $hostname = "localhost"; // Change this to your database hostname
75 $username = "root"; // Change this to your database username
76 $password = "Razan443000"; // Change this to your database password
77 $database = "Wajjehni2"; // Change this to your database name
78
79 // Create connection
80 $connection = mysqli_connect($hostname, $username, $password, $database);
81 </pre>
```

Campus event

by Razan

PHP & HTML

Connection to the Database

```
display_event.php
63 // Check connection
64 if (!$connection) {
65     die("Connection failed: " . mysqli_connect_error());
66 } else {
67     echo "Connected successfully";
68 }
69 $sql = "SELECT `EventName`, `EventDate`, `EventLocation`, `EventType`, `EventProvider`, `CertificateAvailability` FROM `UQU Notification`";
70 $result = mysqli_query($connection, $sql);
71 // Check if any rows were returned
72 if (mysqli_num_rows($result) > 0) {
73     // Output data as a Bootstrap table
74     echo '<div class="container">';
75     echo '<table class="table table-striped">';
76     echo '<thead class="thead-dark">';
77     echo '<tr><th>Event Name</th><th>Event Date</th><th>Event Location</th><th>Event Type</th><th>Event Provider</th><th>Certificate Availability</th></tr>';
78     echo '</thead>';
79     echo '<tbody>';
80     while($row = mysqli_fetch_assoc($result)) {
81         echo "<tr>";
82         echo "<td>" . $row['EventName'] . "</td>";
83         echo "<td>" . $row['EventDate'] . "</td>";
84         echo "<td>" . $row['EventLocation'] . "</td>";
85         echo "<td>" . $row['EventType'] . "</td>";
86         echo "<td>" . $row['EventProvider'] . "</td>";
87         echo "<td>" . ($row['CertificateAvailability'] ? "Available" : "Not Available") . "</td>";
88         echo "</tr>";
89     }
90     echo '</tbody>';
91     echo '</table>';
92     echo '</div>';
93 } else {
94     echo "0 results";
95 }
```

```
display_event.php
72 if (mysqli_num_rows($result) > 0) {
73 }
74     echo '</tbody>';
75     echo '</table>';
76     echo '</div>';
77 } else {
78     echo "0 results";
79 }
80 // Close connection
81 mysqli_close($connection);
82 ?>
83 <!-- Bootstrap JS -->
84 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
85 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
86 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
87 </body>
88 </html>
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
```

safety

by Atyaf

Safety

Urgent intervention

simple intervention

Help



safety

by Atyaf

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="safetytest1.css" >
<script src="button.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />

<title>Safety</title>
</head>
<body>
<!--button -->
<div class="dropdown1">
<button onclick="selectOption2('Urgent intervention')">Urgent intervention</button>
</div>
<div class="title"><h2>Safety</h2></div>

<div class="dropdown2">
| <button onclick ="selectOption1('simple intervention')">simple intervention</button>
</div>
<div class="dropdown3">
| <button onclick ="selectOption('help')">Help</button>
</div>
<!--circle-->
<div class="circle1"></div>
<div class="circle2"></div>
<div class="circle3"></div>
</div>
```

```
<!-- Navigation Bar -->
<ul>
<li><a href="#news"><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor"
   class="bi bi-arrow-90deg-left" viewBox="0 0 16 16">
   <path fill-rule="evenodd"
         d="M1.146 4.854a.5.5 0 0 1 -.70814-4a.5.5 0 1 1 .708.708L2.707 4H12.5A2.5 2.5 0 0 1 15 6.5v8a.5.5 0 0 1-1 0v-8A1.5 1.5
   </svg></a></li>
<li><a href="#contact"><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor"
   class="bi bi-calendar-event" viewBox="0 0 16 16">
   <path d="M11 6.5a.5.5 0 0 1 .5-.5h1a.5.5 0 0 1 -.5.5h-1a.5.5 0 0 1 -.5-.5z" />
   <path d="M3.5 0a.5.5 0 0 1 .5.5v8V.5a.5.5 0 0 1 1 0V1h2a2 2 0 0 1 2 2v1h2a2 2 0 0 1 2 2V3a2 2 0 0 1 2
   </svg></a></li>
<li><a href="#about"><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor"
   class="bi bi-geo-alt" viewBox="0 0 16 16">
   <path
         d="M12.166 8.94c-.524 1.062-1.234 2.12-1.96 3.07A32 32 0 0 1 8 14.58a32 32 0 0 1-2.206-2.57c-.726-.95-1.436-2.008-1.96-
   <path d="M8 8a2 2 0 1 1 0-4 2 2 0 1 0 4m0 1a3 3 0 1 0 0-6 3 3 0 0 0 0 6" />
   </svg></a></li>
<li><a class="active" href="#home"> <svg xmlns="http://www.w3.org/2000/svg" width="25" height="25"
   fill="currentColor" class="bi bi-person" viewBox="0 0 16 16">
   <path
         d="M8 8a3 3 0 1 0 0-6 3 3 0 0 0 6m2-3a2 2 0 1 1-4 0 2 2 0 1 4 0m4 8c0 1-1 1-1 1H3s-1 0-1-1 1-4 6-4 6 3 6 4m-1-.004c
   </svg></a></li>
</ul>
```

safety

by Atyaf

CSS

```
ul {
    position: fixed;
    list-style-type: none;
    width: 100%;
    top: 680px;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #A27861;
}

li {
    float:left;
}

li a {
    display:flow-root;
    color: white;
    text-align: center;
    padding: 16px 40px;
    text-decoration: none;
}

li a:hover {
    background-color: #A27861;
}
```

```
.title {
    background-color: #AC8774; /* Brown background color */
    padding: 12px; /* Add padding */
    display: inline-block; /* Make it inline block to fit the content */
    border: 2px solid #E1EEEF; /* Add border around the frame */
    border-radius: 16px; /* Make border edges round */
}

/* White text color for the frame content */
.title h2 {
    color: white; /* White text color */
}

/* The container <div> - needed to position the dropdown content */
.dropdown1 {
    position:fixed; top:200px;
    left: 130px; display:inline-block;
}
.dropdown2 {
    position:fixed; top:270px;
    left: 130px; display:inline-block;
}
.dropdown3 {
    position:fixed; top:340px;
    left: 130px; display: inline-block;
}
```

```
.circle1{
    position: absolute; height: 30px;
    width: 30px; border: 5px;
    left:130%; top: 210px;
    transform: translateX(-805%); background-color: red;
    border-radius: 5000px;
}
.circle2{
    position: absolute; height: 30px;
    width: 30px; border: 5px;
    left:130%; top:280px;
    transform:translateX(-236px); background-color: yellow;
    border-radius: 5000px;
}
.circle3{
    position: absolute; height: 30px;
    width: 30px; border: 5px;
    left:130%; top:350px;
    transform: translateX(-788%); background-color: green;
    border-radius: 500px;
}
```

safety

by Atyaf

JS

```
1 // JavaScript function to handle button clicks and navigation
2 function selectOption(option) {
3     if (option === 'help') {
4         window.location.href = "help.html";
5     }
6 }
7 function selectOption1(option1) {
8     if (option1 === 'simple intervention') {
9         window.location.href = "simple intervention.html";
10    }
11 }
12 function selectOption2(option2) {
13     if (option2 === 'Urgent intervention') {
14         window.location.href = "Urgent intervention.html";
15    }
16 }
```

safety

by Atyaf

Safety

enter the text

Submit



safety

by Atyaf

HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="safetytest1.css" >
7   <title>Urgent</title>
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
9 </head>
10 <body>
11   <div class="title">
12     <h2>Safety</h2>
13   </div>
14   <div class="box">
15     <form action="http://localhost/W/insert_safety1.php" method="post">
16       <input type="text" name="urgent" placeholder="enter the text"><br>
17       <br>
18       <button class="button submit" type="submit">Submit</button>
19     </form>
20   </div>
21   <!-- Navigation Bar -->
22   <div class="navbar">
23     <a href="safetytest1.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
24     <a href="W1.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
25     <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
26     <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
27   </div>
28 </body>
29 </html>
```

safety

by Atyaf

CSS

```
1 /* Style for the navigation bar */
2 .navbar {
3     position: fixed;
4     bottom: 0;
5     left: 0;
6     width: 100%;
7     background-color: #AC8774;
8     text-align: center; /* Align text in the center */
9 }
10 /* Style for the navigation bar links */
11 .navbar a {
12     display: inline-block;
13     color: #f2f2f2;
14     padding: 14px 36px;
15     text-decoration: none;
16 }
17 /* Change color on hover */
18 .navbar a:hover {
19     background-color: #ddd;
20     color: black;
21 }
22 /* Background image */
23 body {
24     background-image: url('Sci.png'); /* URL of the background image */
25     background-size: cover; /* Cover the entire background */
26     background-position: center; /* Center the background image */
27     background-repeat: no-repeat; /* Do not repeat the background image */
28     background-attachment: fixed; /* Fixed background image */
29     text-align: center; /* Aligning the content of the entire page to the center */
30 }
```

```
30     margin: 0; /* Remove default margin */
31     padding: 0; /* Remove default padding */
32 }
33 /* Style for the colorful frame */
34 .title {
35     background-color: #AC8774; /* Brown background color */
36     padding: 12px; /* Add padding */
37     display: inline-block; /* Make it inline block to fit the content */
38     border: 2px solid #E1E1FF; /* Add border around the frame */
39     border-radius: 16px; /* Make border edges round */
40 }
41 /* White text color for the frame content */
42 .title h2 {
43     color: white; /* White text color */
44 }
45 /* Style the dropdown button */
46 button{
47     background-color: #Ac8774; /* New color */
48     color: white; /* Text color is white */
49     padding: 16px;
50     font-size: 16px;
51     border: none;
52     cursor: pointer;
53     border-radius: 10px; /* Rounded corners */
54 }
55 .box {
56     border-radius: 20px;
57     padding: 20px;
```

```
.box {
    border-radius: 20px;
    padding: 20px;
    color: white;
    width: 80%;
    max-width: 600px;
    margin: 200px auto;
    text-align: center;
}
```

safety

by Atyaf

Safety

We Will Provide you with help Soon!!



safety

by Atyaf
html

```
53  .navbar a:hover {  
54      background-color: #ddd;  
55      color: black;  
56  }  
57  </style>  
58 </head>  
59 <body>  
60 <div class="title">  
61   <h2>Safety</h2>  
62 </div>  
63 <div class="box">  
64   <h3>We Will Provide you with help Soon!!</h3>  
65 </div>  
66 <div class="navbar">  
67   <a href="safetytest1.html"><i class="fas fa-arrow-left"></i></a>  
68   <a href="W1.html"><i class="fas fa-calendar-alt"></i></a>  
69   <a href="test.html"><i class="fas fa-map-marker-alt"></i></a>  
70   <a href="#"><i class="fas fa-user"></i></a>  
71 </div>  
72  
73 </body>  
74 </html>  
75
```

safety

by Atyaf

CSS

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Submit</title>
7      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
8      <style>
9          body {
10              margin: 0;
11              padding: 0;
12              font-family: Arial, sans-serif;
13              background-color: #F4F4F4;
14              background-image: url('Sc1.png'); /* URL of the background image */
15              background-size: cover; /* Cover the entire background */
16              background-position: center; /* Center the background image */
17              background-repeat: no-repeat; /* Do not repeat the background image */
18              background-attachment: fixed; /* Fixed background image */
19              text-align: center; /* Aligning the content of the entire page to the center */
20              margin: 0; /* Remove default margin */
21              padding: 0; /* Remove default padding */
22          }
23          .title {
24              text-align: center;
25              margin-top: 50px;
26          }
27          .box {
28              background-color: #C5AD9F;
29              border-radius: 20px;
30              padding: 20px;
31              color: white;
32              width: 80%;
33              max-width: 600px;
34              margin: 50px auto;
35              text-align: center;
36          }
37          .navbar {
38              position: fixed;
39              bottom: 0;
40              left: 0;
41              width: 100%;
42              background-color: #AC8774;
43              text-align: center; /* Align text in the center */
44          }
45          /* Style for the navigation bar links */
46          .navbar a {
47              display: inline-block;
48              color: #f2f2f2;
49              padding: 14px 36px;
50              text-decoration: none;
51          }
52          /* Change color on hover */
53          .navbar a:hover {
54              background-color: #ddd;
```

safety

by Atyaf

PHP

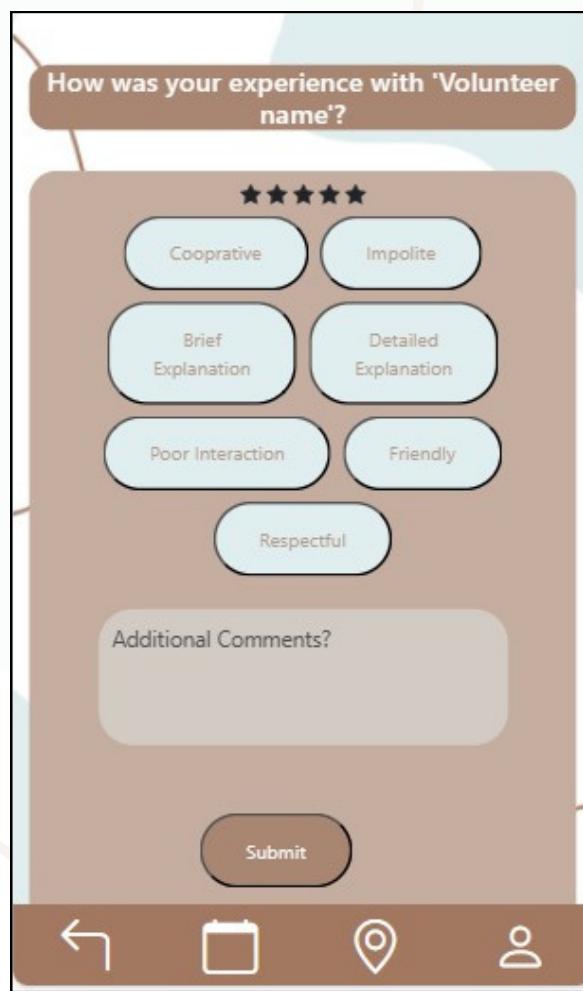
Connection to the Database

```
insert_safety1.php
1 <?php
2 // Database credentials
3 $hostname = "localhost"; // Change this to your database hostname
4 $username = "root"; // Change this to your database username
5 $password = "Razan443000"; // Change this to your database password
6 $database = "Wajjehni2"; // Change this to your database name
7 // Create connection
8 $connection = mysqli_connect($hostname, $username, $password, $database);
9 // Check connection
10 if (!$connection) {
11     die("Connection failed: " . mysqli_connect_error());
12 } else {
13     echo "Connected successfully";
14 }
15
16 // Insertion for Form 1
17 if(isset($_POST['urgent'])) {
18     $levelOfDanger = "urgent";
19     $studentID = 123456789;
20     $details = $_POST['urgent'];
21
22     $sql1 = "INSERT INTO safety (levelOfDanger, StudentID, Details) VALUES ('$levelOfDanger', '$studentID', '$details')";
23     if (mysqli_query($connection, $sql1)) {
24         echo "Data inserted successfully for Form 1<br>";
25         header("Location: submit.html");
26         exit();
27     } else {
28         echo "Error inserting data for Form 1: " . mysqli_error($connection) . "<br>";
29     }
30 }
```

```
insert_safety1.php
30 }
31 // Insertion for Form 2
32 if(isset($_POST['simple'])) {
33     $levelOfDanger = "simple";
34     $studentID = 123456789;
35     $details = $_POST['simple'];
36     $sql2 = "INSERT INTO safety (levelOfDanger, StudentID, Details) VALUES ('$levelOfDanger', '$studentID', '$details')";
37     if (mysqli_query($connection, $sql2)) {
38         echo "Data inserted successfully for Form 2<br>";
39         header("Location: submit.html");
40         exit();
41     } else {
42         echo "Error inserting data for Form 2: " . mysqli_error($connection) . "<br>";
43     }
44 }
45 // Insertion for Form 2
46 if(isset($_POST['help'])) {
47     $levelOfDanger = "Help";
48     $studentID = 123456789;
49     $details = $_POST['help'];
50     $sql3 = "INSERT INTO safety (levelOfDanger, StudentID, Details) VALUES ('$levelOfDanger', '$studentID', '$details')";
51     if (mysqli_query($connection, $sql3)) { // Corrected $sql2 to $sql3
52         echo "Data inserted successfully for Form 3<br>";
53         header("Location: submit.html");
54         exit();
55     } else {
56         echo "Error inserting data for Form 3: " . mysqli_error($connection) . "<br>";
57     }
58 }
59 // Close the database connection
60 mysqli_close($connection);
61 ?>
```

Rating & feedback

by Arwa



Limitations: Due to complications with the database, the current implementation does not support the display of volunteers' previous ratings. However, volunteer information, including their rating (Number of stars), will still be visible on the map interface.

Rating & feedback

by Arwa

The image shows five code snippets from a web browser's developer tools, likely from a file named Rating.html. The snippets are arranged in a grid-like layout.

- Top Left:** A snippet of CSS and JavaScript. The CSS includes meta tags for charset and viewport, links to Bootstrap and Bootstrap Icons, and a background image URL. The JavaScript is a single-line function definition.
- Top Right:** A snippet of CSS. It sets the body's background image to fixed, aligns content center, and removes default padding. It also defines styles for checked checkboxes and buttons, including colors and borders.
- Middle Left:** A snippet of CSS. It styles a button with text-align: center, inline-block display, and a green background color (#A98672). It also styles a header with a green background color (#A98672) and white text.
- Middle Right:** A snippet of CSS. It styles a text area container with flex display, center justify-content, and a margin-top of 20px. It also styles a text area with a width of 300px, height of 100px, rounded corners, and a white background color (#D2CBC4).
- Bottom Left:** A snippet of CSS. It styles a text area with a black background color (#C5AD9F), rounded corners, and a padding of 5px. It also styles a nav element with a fixed position, a light blue background color (#A27861), and a margin-top of 110px.
- Bottom Right:** A snippet of CSS. It defines active states for buttons and spans, changing their background colors to #d4c9bc and #d7d758 respectively. It also includes a large block of CSS for a card-based layout, including container, col, header, rateCard, row-md-6, card, and card-body classes.

Rating & feedback

by Arwa

The image shows a code editor with three tabs open, all titled "Rating.html".

- Rating.html (Top Tab):** This tab displays the main HTML structure. It includes a container with a card containing five star icons, each with an onclick event to call "toggleStar(this)". Below the card is a row of four buttons: "Cooperative", "Impolite", "Brief Explanation", and "Detailed Explanation".
- Rating.html (Middle Tab):** This tab shows the "Rating.html" page with additional content. It includes a form with a text area for "Additional Comment" and a submit button. The "Detailed Explanation" button is highlighted with a yellow box.
- Rating.html (Bottom Tab):** This tab contains the JavaScript code for the rating logic. It defines functions "toggleStar" and "toggleButton". The "toggleStar" function iterates through all star icons and toggles their checked status. The "toggleButton" function toggles the active class on the button itself and its corresponding star icon.

```
Rating.html (Top Tab)
W > Rating.html > html > body > script
2   <html lang="en">
105  <body>
106    <div class="container">
107      <div class="col justify-content-center">
108        <div class="row-md-6">
109          <div class="row-md-6">
110            <card>
111              <card-body class="justify-content-center">
112                <span class="fa fa-star" onclick="toggleStar(this)"></span>
113                <span class="fa fa-star" onclick="toggleStar(this)"></span>
114                <span class="fa fa-star" onclick="toggleStar(this)"></span>
115                <span class="fa fa-star" onclick="toggleStar(this)"></span>
116                <span class="fa fa-star" onclick="toggleStar(this)"></span>
117            </card-body>
118          </card>
119        </div>
120      </div>
121    </div>
122
123    <button class="button" onclick="toggleButton(this)">Cooperative</button>
124    <button class="button" onclick="toggleButton(this)">Impolite</button>
125    <button class="button" onclick="toggleButton(this)">Brief Explanation</button>
126    <button class="button" onclick="toggleButton(this)">Detailed Explanation</button>
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
```

```
Rating.html (Middle Tab)
W > Rating.html > html > body > script
2   <html lang="en">
105  <body>
106    <div class="container">
107      <div class="col justify-content-center">
108        <div class="row-md-6">
109          <button class="button" onclick="toggleButton(this)">Detailed Explanation</button>
110          <button class="button" onclick="toggleButton(this)">Poor Interaction</button>
111          <button class="button" onclick="toggleButton(this)">Friendly</button>
112          <button class="button" onclick="toggleButton(this)">Respectful</button>
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
```

```
Rating.html (Bottom Tab)
W > Rating.html > html > body > script > toggleStar
2   <html lang="en">
105  <body>
106    <div class="container">
107      <div class="col">
108        <a href="#">i class="bi bi-arrow-90deg-left"></a>
109      </div>
110      <div class="col">
111        <a href="#">i class="bi bi-calendar"></a>
112      </div>
113      <div class="col">
114        <a href="#">i class="bi bi-geo-alt"></a>
115      </div>
116      <div class="col">
117        <a href="#">i class="bi bi-person"></a>
118      </div>
119    </div>
120    <script>
121      function toggleStar(star) {
122        var stars = document.querySelectorAll('.fa-star');
123        var index = Array.prototype.indexOf.call(stars, star);
124        for (var i = 0; i < index; i++) {
125          stars[i].classList.add("checked");
126        }
127        for (var i = index + 1; i < stars.length; i++) {
128          stars[i].classList.remove("checked");
129        }
130      }
131
132      function toggleButton(button) {
133        button.classList.toggle("active");
134      }
135
136      function toggleStar(star) {
137        var stars = document.querySelectorAll('.fa-star');
138        var index = Array.prototype.indexOf.call(stars, star);
139        for (var i = 0; i < index; i++) {
140          stars[i].classList.add("checked");
141        }
142        for (var i = index + 1; i < stars.length; i++) {
143          stars[i].classList.remove("checked");
144        }
145      }
146
147      function toggleButton(button) {
148        button.classList.toggle("active");
149      }
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
```

Rating & feedback

by Arwa

*Java script code is AI generated using chatGPT to make the buttons change color when clicked and to make a loop that makes the rating stars interactive and fills up all the stars from the leftmost one up to the star that was clicked.

```
Rating.html X

W > Rating.html > html > body > script > toggleStar
  2   <html lang="en">
  3     <body>...
  4       ...
  5
  6       </nav>
  7       <script>
  8         function toggleButton(button) {
  9           button.classList.toggle("active");
 10         }
 11
 12         function toggleStar(star) {
 13           var stars = document.querySelectorAll('.fa-star');
 14           var index = Array.prototype.indexOf.call(stars, star);
 15           for (var i = 0; i <= index; i++) {
 16             stars[i].classList.add("checked");
 17           }
 18           for (var i = index + 1; i < stars.length; i++) {
 19             stars[i].classList.remove("checked");
 20           }
 21         }
 22       </script>
 23     </body>
 24   </html>
 25
```

PHP database connection:

```
C:\Users\usjy> Downloads > W > W > insert_rating.php
1  <?php
2
3
4  error_reporting(E_ALL);
5  ini_set('display_errors', 1);
6  // Database credentials
7  $hostname = "localhost";
8  $username = "root";
9  $password = "Razan443000";
10 $database = "Wajjehni2";
11
12 // Create connection
13 $connection = mysqli_connect($hostname, $username, $password, $database);
14
15 // Check connection
16 if (!$connection) {
17     die("Connection failed: " . mysqli_connect_error());
18 } else {
19     echo "Connected successfully";
20 }
21
22 $RateID=10;
23 $StudentID=123456789 ;
```

```
insert_rating.php X

C:> Users > dsgj > Downloads > W > W > insert_rating.php

22 $RateID=10;
23 $StudentID=123456789 ;
24 $VolunteerID=111111111 ;
25 $NumberOfStars = 5;
26 $RatingList="cooperative";
27 $Feedback= $_POST['additional_comments'];
28
29 // Prepare SQL query to insert the data into the database
30 $sql = "INSERT INTO Rating (RateID, StudentID, VolunteerID, NumberOfStars, RatingList, Feedback)
31 | VALUES ($RateID, $StudentID, $VolunteerID, $NumberOfStars, '$RatingList', '$Feedback')";
32
33 // Execute the SQL query
34 if (mysqli_query($connection, $sql)) {
35     echo "Data added to the database successfully";
36     header("Location: FeedbackConfirmed.html");
37     exit();
38 } else {
39     echo "Error: " . $sql . "<br>" . mysqli_error($connection);
40 }
41
42 // Close the database connection
43 mysqli_close($connection);
44 ?>
```

Rating & feedback

by Arwa

How was your experience with 'Volunteer name'?

Thank you for taking the time to rate our
volunteer!!
Your feedback is valuable to us and helps us
recognize the efforts of our volunteers.
We appreciate your input in improving our
volunteer program



Rating & feedback

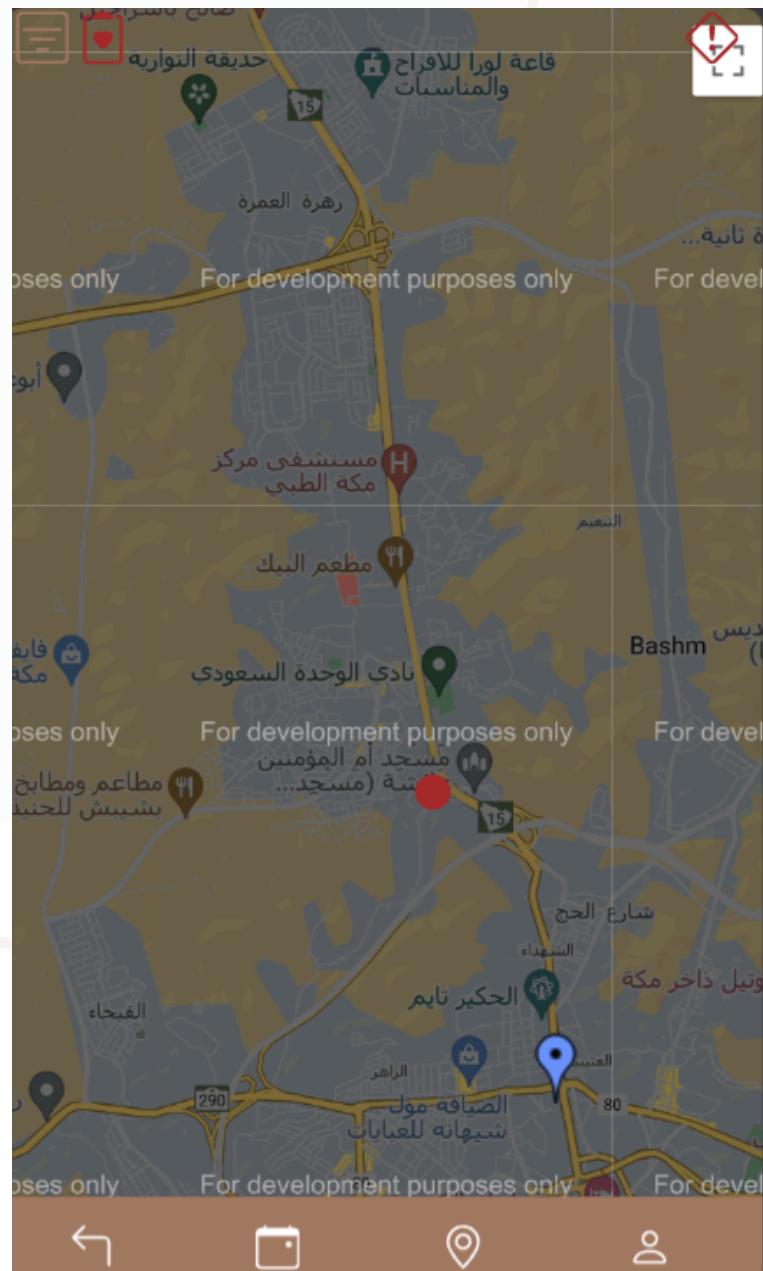
by Arwa

HTML code while using the same css as the previous interface:

```
insert_rating.php  FeedbackConfirmed.html X
C: > Users > اسجی > Downloads > W > W > FeedbackConfirmed.html > ...
2   <html lang="en">
71  <body>
72    <div class="container">
73      <div class="col justify-content-center">
74        <div class="row-md-6 ">
75          <h5 class="header">How was your experience with 'Volunteer name'?</h5>
76          <div class="rateCard">
77            <h3>Thank you for taking the time to rate our volunteer!!</h3>
78            <br>Your feedback is valuable to us and helps us recognize the efforts of our volunteers.
79            <br>We appreciate your input in improving our volunteer program</h3>
80          </div>
81        </div>
82      </div>
83    </div>
84
85    <div class="nav">
86      <a href="Rating.html"><i class="fas fa-arrow-left"></i></a>
87      <a href="W1.html"><i class="fas fa-calendar-alt"></i></a>
88      <a href="test.html"><i class="fas fa-map-marker-alt"></i></a>
89      <a href="#"><i class="fas fa-user"></i></a>
90    </div>
91  </body>
92 </html>
```

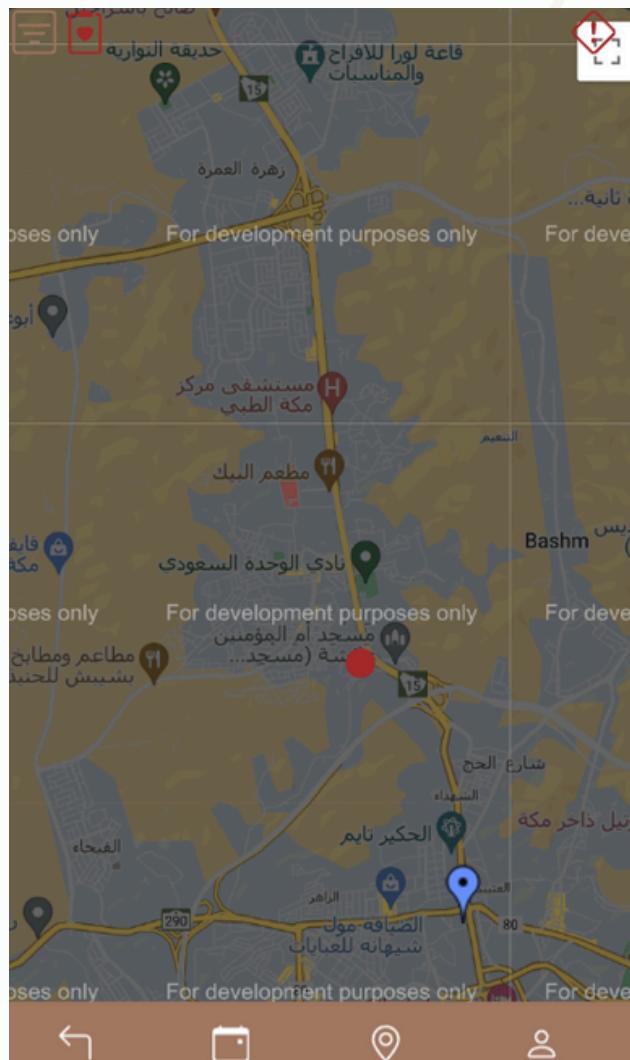
map

by Renad



Map

By Renad



I couldn't link the interface to the database, so I wasn't able to display the marker after saving the user's location and type of volunteering. Consequently, the marker wasn't shown at the time entered by the user, nor was it hidden after the entered time expired. As for the marker's color, I assigned coordinates for the volunteer as an attribute in the code and assumed the volunteer chose academic volunteering, so I set the marker color to blue.

map

by Renad

CSS

```
.icon-container {  
    position: fixed; top: 10px;  
    width: 100%; display:block;  
    gap: 210px; padding: 0 10px;  
    z-index: 1000;}  
#filter {  
    margin-left: 310px;}  
ul {  
    position: fixed;  
    list-style-type: none;  
    width: 100%;  
    top: 680px;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #A27861;}  
li {  
    float:left; }  
li a {  
    display:flow-root;  
    color: white;  
    text-align: center;  
    padding: 16px 40px;  
    text-decoration: none; }  
li a:hover {  
    background-color: #A27861; }
```

map by Renad

JS

```
<script>
  var map;
  var marker;
  var previousLocation;
  // Function to initialize the map
  function initMap() {
    var initialLocation = { lat: 37.7749, lng: -122.4194 }; // Default to San Francisco

    // Create a new map
    map = new google.maps.Map(document.getElementById('map'), {
      zoom: 14,
      center: initialLocation
    });

    var volunteerType = "guidance";
    var markerColor;

    if (volunteerType === "academic") {
      markerColor = "green";
    } else if (volunteerType === "guidance") {
      markerColor = "brown";
    } else {
      markerColor = "blue";
    }
  }
```

```
// Create a marker for the current location
marker = new google.maps.Marker({
  map: map,
  title: 'Current Location',
  icon: {
    path: google.maps.SymbolPath.CIRCLE,
    scale: 10,
    fillColor: markerColor,
    fillOpacity: 1,
    strokeWeight: 0
  }
});
```

```
  // Get the user's location and update the map
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(updatePosition, handleError);
  }

  // Function to update the marker position and center the map
  function updatePosition(position) {
    var lat = position.coords.latitude;
    var lng = position.coords.longitude;
    var currentLocation = { lat: lat, lng: lng };
    // Set marker for current location
    marker.setPosition(currentLocation);
    map.setCenter(currentLocation);
    // Show previous location if available
    if (previousLocation) {
      var previousMarker = new google.maps.Marker({
        position: previousLocation,
        map: map,
        title: 'Previous Location',
        icon: {
          url: 'http://maps.google.com/mapfiles/ms/icons/blue-dot.png',
          scaledSize: new google.maps.Size(30, 30)
        }
      });
    }
  }
}
```

```
  // Save current location as previous location
  previousLocation = currentLocation;
  // Show marker for provided coordinates
  var providedCoordinates = { lat: 21.43996424018569, lng: 39.80914295054868 };
  var providedMarker = new google.maps.Marker({
    position: providedCoordinates,
    map: map,
    title: 'Provided Coordinates',
    icon: {
      url: 'http://maps.google.com/mapfiles/ms/icons/blue-dot.png',
      scaledSize: new google.maps.Size(48, 48)
    }
  });
  // Add click event listener to the provided marker
  providedMarker.addListener('click', function () {
    // Open an info window with the desired content
    var infoWindow = new google.maps.InfoWindow({
      content: "<div>Name: Renad Ibrahim Alsaedi<br>Email: 443809572@st.uqu.edu.sa<br>Department: I"
    });
    infoWindow.open(map, providedMarker);
  });
  // Function to handle errors
  function handleError(error) {
    console.error('Error getting location:', error);
  }
  // Initialize the map when the page loads
  window.onload = initMap;
```

map

by Renad

HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Interactive Map</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDRDP0wFgd61DC0185Noex2d01hi5vii0"></script>
  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
    integrity="sha384-ITEBVVOJWfrrHfPjPvp/LzJvgyl6FvVRXk/VK+wC4jQKc/HVqc2ph8CDewa9r"
    crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
    integrity="sha384-0pUZvbkaxF6gxjElmu0rXVNuzTqB8avLwCsOgabYfzo8Tto5qErlyt"
    crossorigin="anonymous"></script>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script> ...
  </script>

  <style> ...
  </style>
</head>
<body>
```

```
<!-- Map -->
<div id="map" style="width: 100%; height: 1000px;"></div>
<!-- Navigation Bar --&gt;
<ul>
- <a href="#news"><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor"
    class="bi bi-arrow-90deg-left" viewBox="0 0 16 16">
        <path fill-rule="evenodd">
            d=M1.146 4.854a.5.5 0 0 1 -.708L4-.5.5 0 1 1 .708.708L2.707 4H12.5A2.5 2.5 0 0 1 15 6.5v8a.5.5 0 0 1-1 0v-8A1.5 1.5
        </path>
    </svg></a></li>
- <a href="#contact"><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor"
    class="bi bi-calendar-event" viewBox="0 0 16 16">
        <path d="M11 6.5a.5.5 0 0 1 .5-.5v1a.5.5 0 0 1-.5.5h-1a.5.5 0 0 1-.5-.5z" />
        <path d="M3.5 0a.5.5 0 0 1 .5.5V1h8.5a.5.5 0 0 1 0V1h2a 2 0 0 1 2 2v1a2 2 0 0 1 2 2H2a2 2 0 0 1 2-2V3a2 2 0 0 1 2
    </svg></a></li>
- <a href="#about"><svg xmlns="http://www.w3.org/2000/svg" width="25" height="25" fill="currentColor"
    class="bi bi-geo-alt" viewBox="0 0 16 16">
        <path
            d="M12.166 8.94c-.524 1.062-1.234 2.12-1.96 3.07A32 32 0 0 1 8 14.58a32 32 0 0 1-2.206-2.57c-.726-.95-1.436-2.008-1.96
        <path d="M8 8a2 2 0 1 1 0-4 2 2 0 0 1 0 4m1a3 3 0 1 0 0-6 3 3 0 0 0 6" />
    </svg></a></li>
- <a class="active" href="#home"> <svg xmlns="http://www.w3.org/2000/svg" width="25" height="25"
    fill="currentColor" class="bi bi-person" viewBox="0 0 16 16">
        <path
            d="M8 8a3 3 0 1 0 0-6 3 3 0 0 0 6m2-3a2 2 0 1 1-4 0 2 2 0 0 1 4 0m4 8c0 1-1 1-1 1H3s-1 0-1-1 1-4 6-4 6 3 6 4m-1-.004c
    </svg></a></li>

```

volunteer information

information volunteer page by Asma

Name: _____

University Email: _____

Department: _____

Academic Level: _____

Points: _____

Type of Volunteering: _____

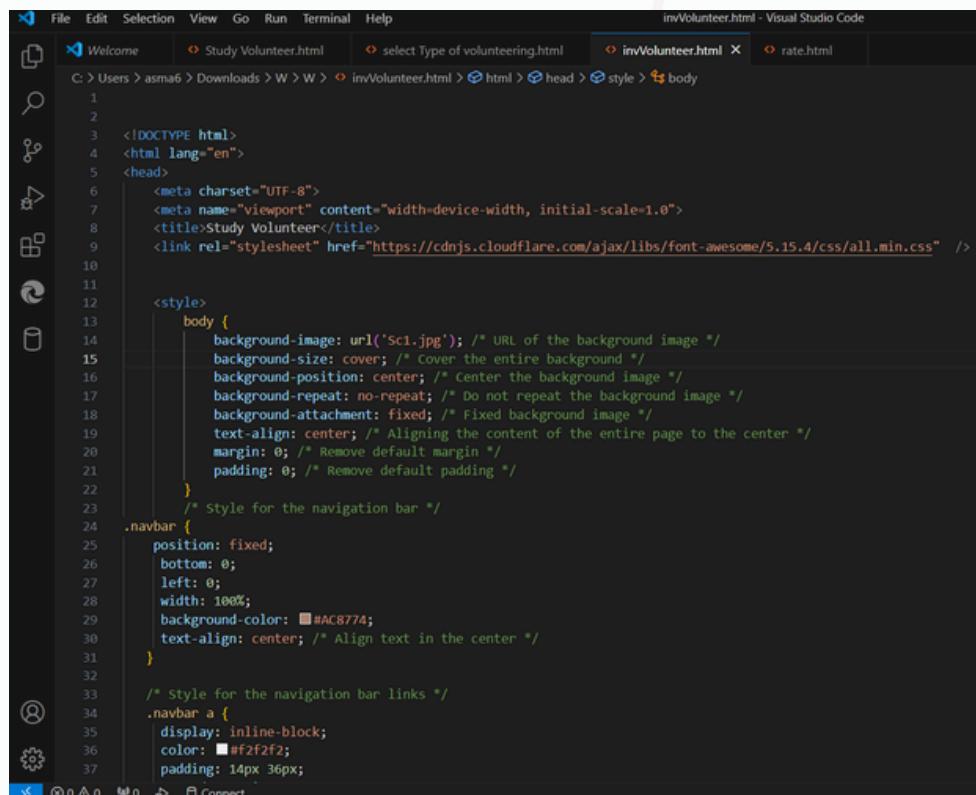
Volunteering

← ⏷ ⏸ ⏹ ⏺

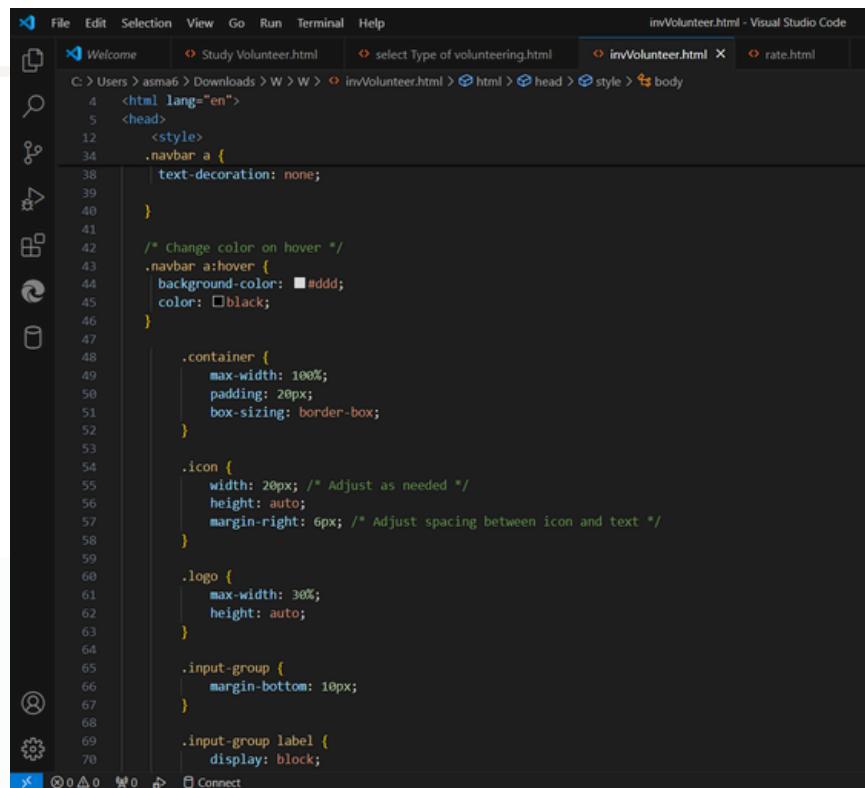
volunteer information

information volunteer page

by Asma



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Study Volunteer</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
<style>
    body {
        background-image: url('Sc1.jpg'); /* URL of the background image */
        background-size: cover; /* Cover the entire background */
        background-position: center; /* Center the background image */
        background-repeat: no-repeat; /* Do not repeat the background image */
        background-attachment: fixed; /* Fixed background image */
        text-align: center; /* Aligning the content of the entire page to the center */
        margin: 0; /* Remove default margin */
        padding: 0; /* Remove default padding */
    }
    /* Style for the navigation bar */
    .navbar {
        position: fixed;
        bottom: 0;
        left: 0;
        width: 100%;
        background-color: #AC8774;
        text-align: center; /* Align text in the center */
    }
    /* Style for the navigation bar links */
    .navbar a {
        display: inline-block;
        color: #f2f2f2;
        padding: 14px 36px;
    }
</style>
```

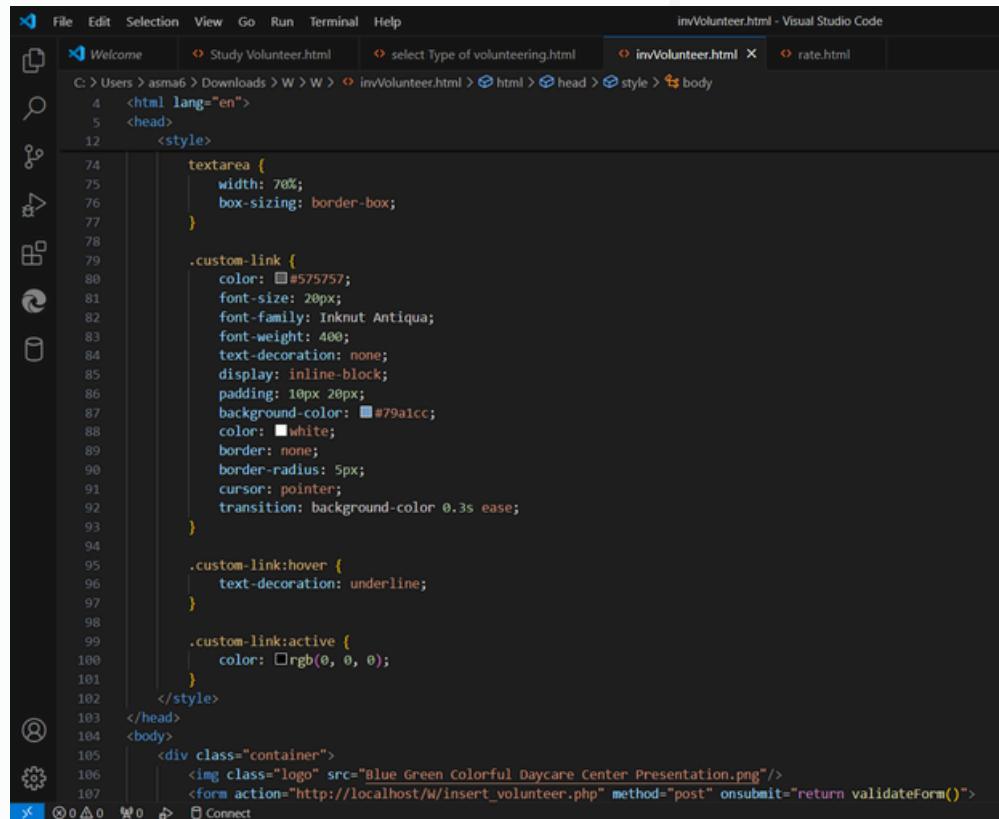


```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        .navbar a {
            text-decoration: none;
        }
        /* Change color on hover */
        .navbar a:hover {
            background-color: #ddd;
            color: black;
        }
        .container {
            max-width: 100%;
            padding: 20px;
            box-sizing: border-box;
        }
        .icon {
            width: 20px; /* Adjust as needed */
            height: auto;
            margin-right: 6px; /* Adjust spacing between icon and text */
        }
        .logo {
            max-width: 30%;
            height: auto;
        }
        .input-group {
            margin-bottom: 10px;
        }
        .input-group label {
            display: block;
        }
    </style>
```

volunteer information

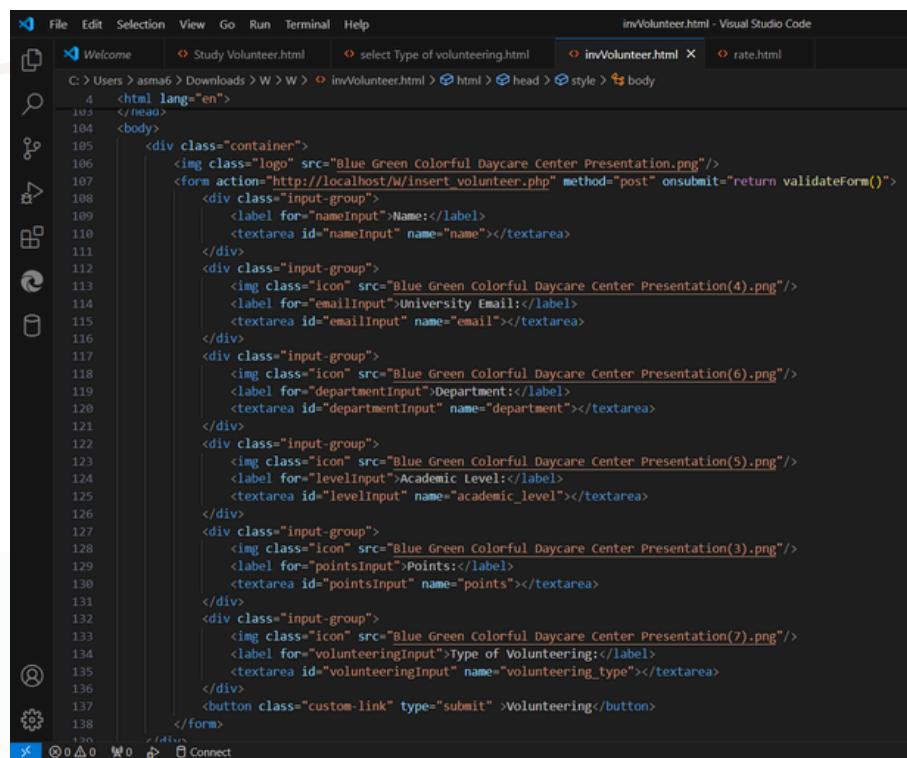
information volunteer page

by Asma



```
<html lang="en">
</head>
<body>
<div class="container">

<form action="http://localhost/W/insert_volunteer.php" method="post" onsubmit="return validateForm()">
```



```
<html lang="en">
</head>
<body>
<div class="container">

<form action="http://localhost/W/insert_volunteer.php" method="post" onsubmit="return validateForm()">
<div class="input-group">
<label for="nameInput">Name:</label>
<textarea id="nameInput" name="name"></textarea>
</div>
<div class="input-group">

<label for="emailInput">University Email:</label>
<textarea id="emailInput" name="email"></textarea>
</div>
<div class="input-group">

<label for="departmentInput">Department:</label>
<textarea id="departmentInput" name="department"></textarea>
</div>
<div class="input-group">

<label for="levelInput">Academic Level:</label>
<textarea id="levelInput" name="academic_level"></textarea>
</div>
<div class="input-group">

<label for="pointsInput">Points:</label>
<textarea id="pointsInput" name="points"></textarea>
</div>
<div class="input-group">

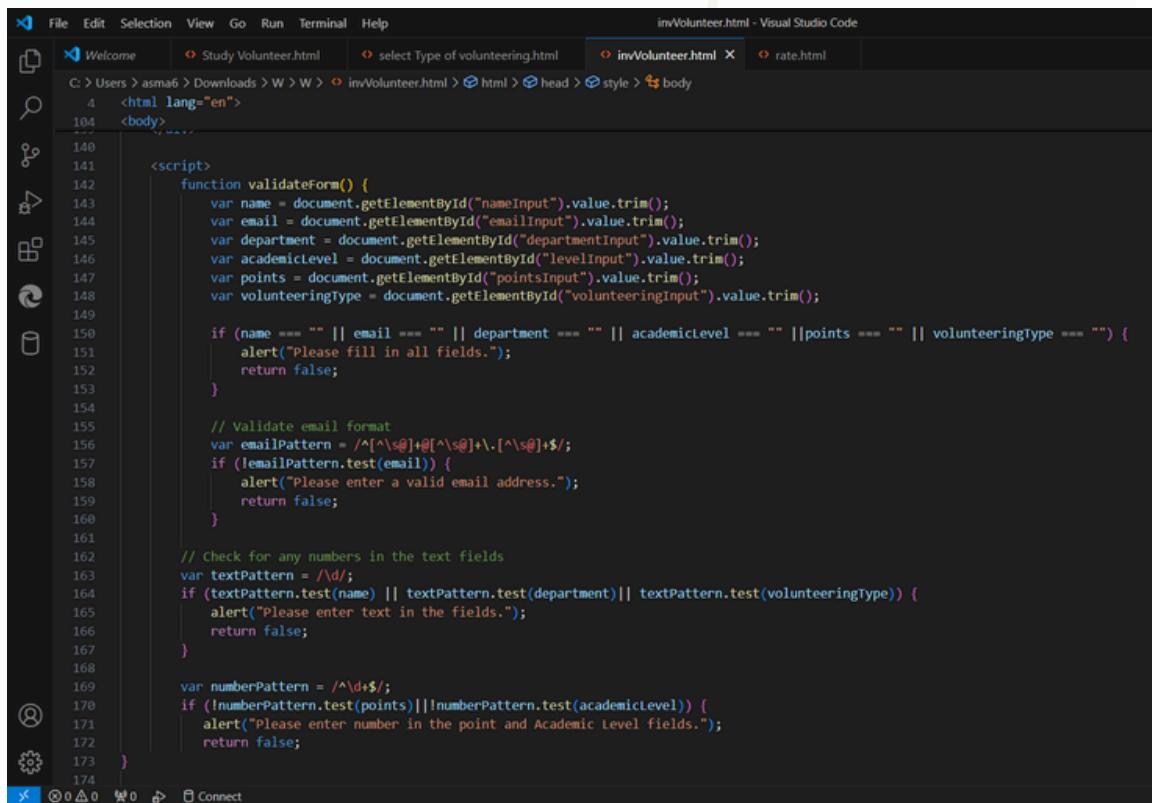
<label for="volunteeringInput">Type of Volunteering:</label>
<textarea id="volunteeringInput" name="volunteering_type"></textarea>
</div>
<button class="custom-link" type="submit" >Volunteering</button>
</form>

```

volunteer information

information volunteer page

by Asma



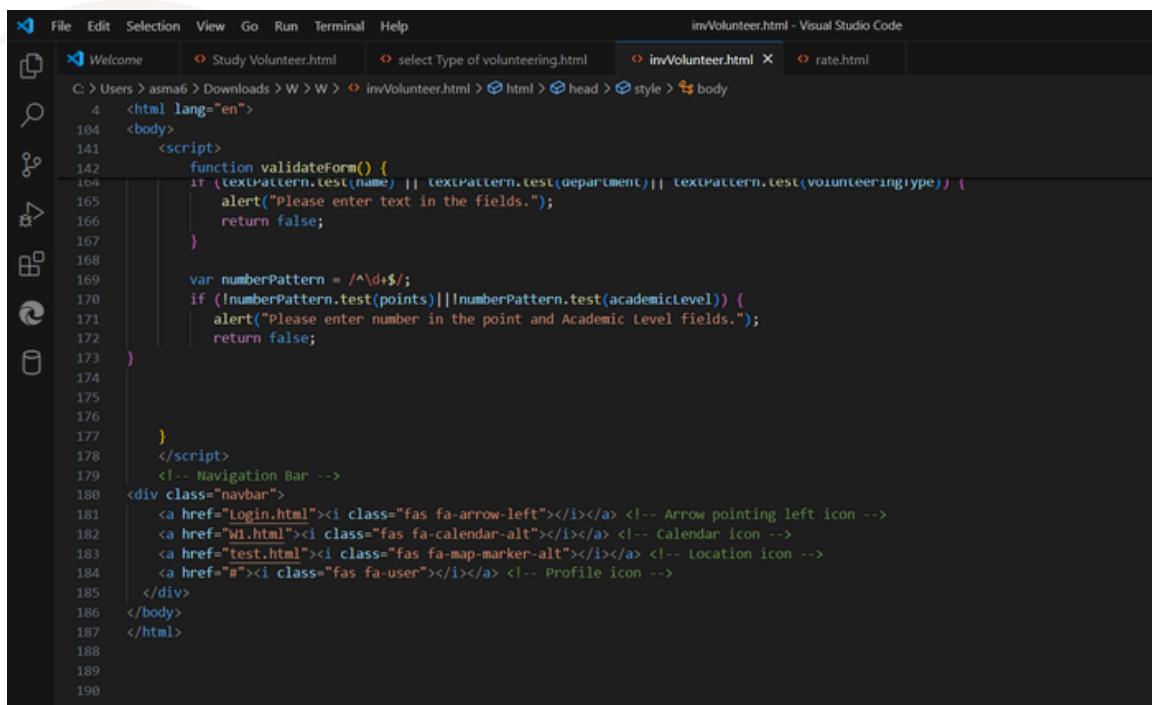
```
invVolunteer.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C:\> Users > asma6 > Downloads > W > W > invVolunteer.html > html > head > style > body
<html lang="en">
<head>
<script>
    function validateForm() {
        var name = document.getElementById("nameInput").value.trim();
        var email = document.getElementById("emailInput").value.trim();
        var department = document.getElementById("departmentInput").value.trim();
        var academicLevel = document.getElementById("levelInput").value.trim();
        var points = document.getElementById("pointsInput").value.trim();
        var volunteeringType = document.getElementById("volunteeringInput").value.trim();

        if (name === "" || email === "" || department === "" || academicLevel === "" || points === "" || volunteeringType === "") {
            alert("Please fill in all fields.");
            return false;
        }

        // Validate email format
        var emailPattern = /^[^s@]+@[^\s@]+\.\[^s@]+\$/;
        if (!emailPattern.test(email)) {
            alert("Please enter a valid email address.");
            return false;
        }

        // Check for any numbers in the text fields
        var textPattern = /\d/;
        if (textPattern.test(name) || textPattern.test(department) || textPattern.test(volunteeringType)) {
            alert("Please enter text in the fields.");
            return false;
        }

        var numberPattern = /^\d+$/;
        if (!numberPattern.test(points) || !numberPattern.test(academicLevel)) {
            alert("Please enter number in the point and Academic Level fields.");
            return false;
        }
    }
</script>
```



```
invVolunteer.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C:\> Users > asma6 > Downloads > W > W > invVolunteer.html > html > head > style > body
<html lang="en">
<head>
<script>
    function validateForm() {
        if (!textPattern.test(name) || !textPattern.test(department) || !textPattern.test(volunteeringType)) {
            alert("Please enter text in the fields.");
            return false;
        }

        var numberPattern = /\d+$/;
        if (!numberPattern.test(points) || !numberPattern.test(academicLevel)) {
            alert("Please enter number in the point and Academic Level Fields.");
            return false;
        }
    }
</script>
<!-- Navigation Bar -->
<div class="navbar">
    <a href="Login.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
    <a href="W1.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
    <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
    <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
</div>
</body>
</html>
```

volunteer information

information volunteer page

by Asma

PHP

Connection to the Database

```
insert_volunteer.php
1 <?php
2 // Database credentials
3 $hostname = "localhost"; // Change this to your database hostname
4 $username = "root"; // Change this to your database username
5 $password = "Razan443000"; // Change this to your database password
6 $database = "Wajjehni2"; // Change this to your database name
7
8 // Create connection
9 $connection = mysqli_connect($hostname, $username, $password, $database);
10
11 // Check connection
12 if (!$connection) {
13 | die("Connection failed: " . mysqli_connect_error());
14 } else {
15 | echo "Connected successfully";
16 }
17 $VolunteerID =14; // Assuming VolunteerID is retrieved from session
18 $Name = $_POST['name']; // Assuming Name is provided via form or any other source
19 $Email = $_POST['email']; // Assuming Email is provided via form or any other source
20 $password = 1234567890; // Assuming password is provided via form or any other source
21 $AvailabiltyStatus=TRUE;// Assuming AvailabiltyStatus is provided via form or any other source
22 $GuidanceType ="Academic"; // Assuming GuidanceType is provided via form or any other source
23 $Specialty = $_POST['department']; // Assuming Specialty is provided via form or any other source
24 $AcademicLevel = $_POST['academic_level']; // Assuming AcademicLevel is provided via form or any other source
25 // $Time =0; // Assuming Time is provided via form or any other source
26 $Points = $_POST['points']; // Assuming Points is provided via form or any other source
27 $pointsEarned =0; // Assuming pointsEarned is provided via form or any other source
```

```
insert_volunteer.php
28 // Insert data into the database
29 $sql = "INSERT INTO `volunteer` (VolunteerID, Name, Email, password, AvailabiltyStatus, `Guidance Type`, Specialty, AcademicLevel, Time, Points,
30 | VALUES ('$VolunteerID', '$Name', '$Email', '$password', '$AvailabiltyStatus', '$GuidanceType', '$Specialty', '$AcademicLevel', NOW(), '$
31
32 if (mysqli_query($connection, $sql)) {
33 | echo "Data inserted successfully";
34 | header("Location: select Type of volunteering.html");
35 | exit();
36 } else {
37 | echo "Error: " . $sql . "<br>" . mysqli_error($connection);
38 }
39 // Close the database connection
40 mysqli_close($connection);
41
42 ?>
```

volunteer information

type of volunteering page by Asma

Select Type
of
Volunteering

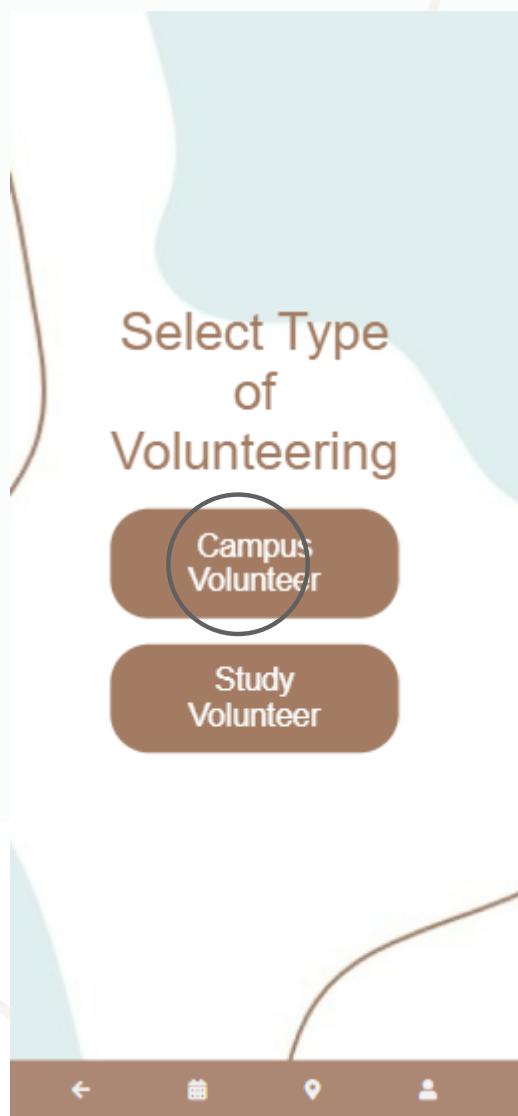
Campus
Volunteer

Study
Volunteer



volunteer information

type of volunteering page by Asma

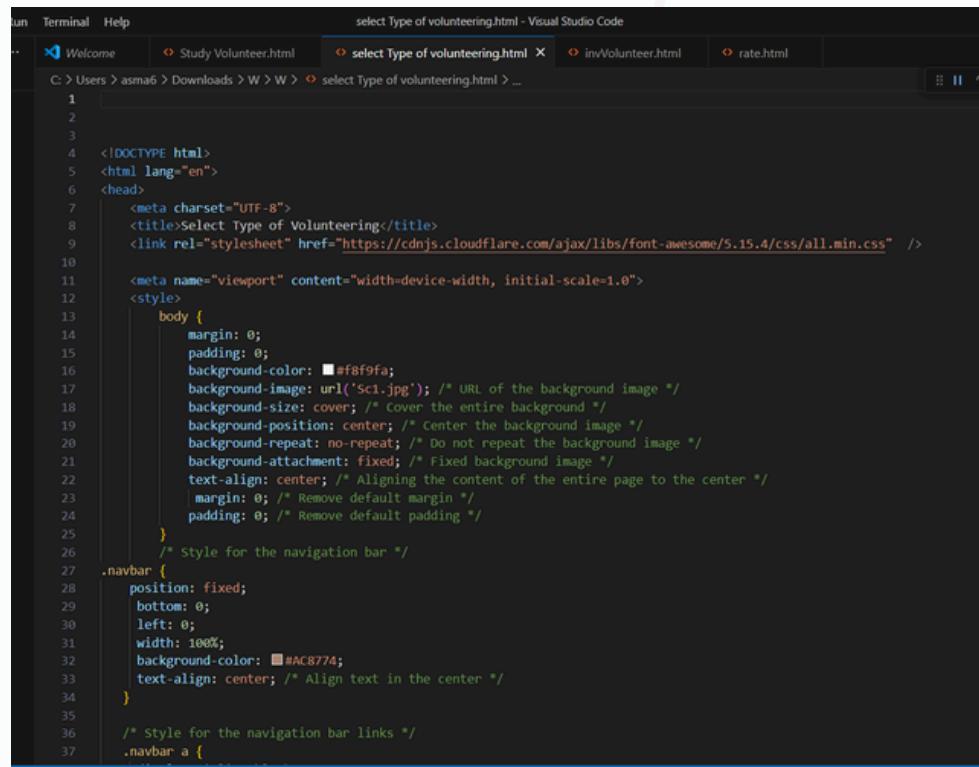


I was unable to complete this part due to a lack of time, as my primary focus was on linking the database with the HTML code. Therefore, I couldn't finish it. I hope to complete this project with all its requirements in the future.

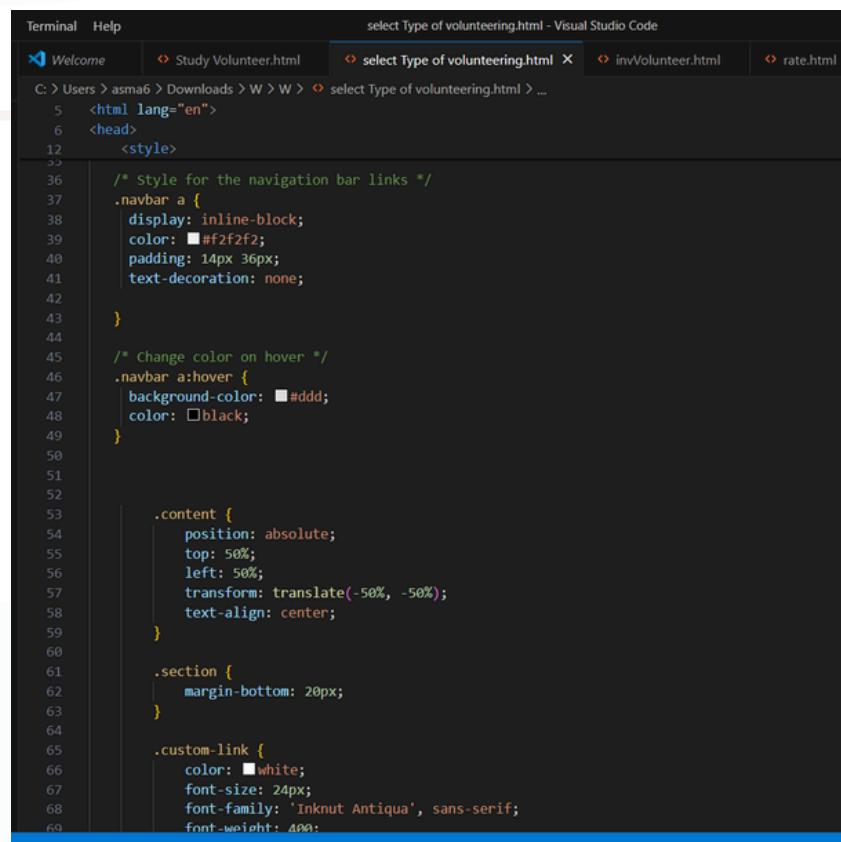
volunteer information

type of volunteering page

by Asma



```
File Terminal Help select Type of volunteering.html - Visual Studio Code
..  Welcome  Study Volunteer.html  select Type of volunteering.html  invVolunteer.html  rate.html
C:\> Users > asma6 > Downloads > W > W > select Type of volunteering.html ...
1
2
3
4 <!DOCTYPE html>
5 <html lang="en">
6 <head>
7   <meta charset="UTF-8">
8   <title>Select Type of Volunteering</title>
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
10  <meta name="viewport" content="width=device-width, initial-scale=1.0">
11  <style>
12    body {
13      margin: 0;
14      padding: 0;
15      background-color: #f8f9fa;
16      background-image: url('sc1.jpg'); /* URL of the background image */
17      background-size: cover; /* cover the entire background */
18      background-position: center; /* center the background image */
19      background-repeat: no-repeat; /* Do not repeat the background image */
20      background-attachment: fixed; /* Fixed background image */
21      text-align: center; /* Aligning the content of the entire page to the center */
22      margin: 0; /* Remove default margin */
23      padding: 0; /* Remove default padding */
24    }
25    /* Style for the navigation bar */
26    .navbar {
27      position: fixed;
28      bottom: 0;
29      left: 0;
30      width: 100%;
31      background-color: #AC8774;
32      text-align: center; /* Align text in the center */
33    }
34    /* Style for the navigation bar links */
35    .navbar a {
36      /* Style for the navigation bar links */
37    }
38    .navbar a {
39      display: inline-block;
40      color: #f2f2f2;
41      padding: 14px 36px;
42      text-decoration: none;
43    }
44    /* Change color on hover */
45    .navbar a:hover {
46      background-color: #ddd;
47      color: black;
48    }
49
50
51
52
53    .content {
54      position: absolute;
55      top: 50%;
56      left: 50%;
57      transform: translate(-50%, -50%);
58      text-align: center;
59    }
60
61    .section {
62      margin-bottom: 20px;
63    }
64
65    .custom-link {
66      color: white;
67      font-size: 24px;
68      font-family: 'Inknut Antiqua', sans-serif;
69      font-weight: 400;
```



```
File Terminal Help select Type of volunteering.html - Visual Studio Code
..  Welcome  Study Volunteer.html  select Type of volunteering.html  invVolunteer.html  rate.html
C:\> Users > asma6 > Downloads > W > W > select Type of volunteering.html ...
5 <html lang="en">
6 <head>
12 <style>
35
36    /* Style for the navigation bar links */
37    .navbar a {
38      display: inline-block;
39      color: #f2f2f2;
40      padding: 14px 36px;
41      text-decoration: none;
42    }
43
44
45    /* Change color on hover */
46    .navbar a:hover {
47      background-color: #ddd;
48      color: black;
49    }
50
51
52
53    .content {
54      position: absolute;
55      top: 50%;
56      left: 50%;
57      transform: translate(-50%, -50%);
58      text-align: center;
59    }
60
61    .section {
62      margin-bottom: 20px;
63    }
64
65    .custom-link {
66      color: white;
67      font-size: 24px;
68      font-family: 'Inknut Antiqua', sans-serif;
69      font-weight: 400;
```

volunteer information

type of volunteering page

by Asma

```
Terminal Help select Type of volunteering.html - Visual Studio Code
  Welcome Study Volunteer.html select Type of volunteering.html invVolunteer.html
C:\Users\asma6\Downloads\W\W> select Type of volunteering.html ...
5   <html lang="en">
6   <head>
12     <style>
65       .custom-link {
66         color: #white;
67         font-size: 24px;
68         font-family: 'Inknut Antiqua', sans-serif;
69         font-weight: 400;
70         text-decoration: none;
71     }
72
73     .custom-link:hover {
74       text-decoration: underline;
75     }
76
77     .custom-link:active {
78       color: #white;
79     }
80
81     .button {
82       display: inline-block;
83       padding: 15px 30px;
84       background-color: #A27862;
85       color: #white;
86       border-radius: 30px;
87       text-decoration: none;
88       transition: background-color 0.3s ease;
89     }
90
91     .button:hover {
92       background-color: #875f4d;
93     }
94   </style>
95   </head>
96   <body>
97
98
99   <div class="content">
100     <div class="section">
101       <div style="font-size: 40px; color: #A27862; font-family: 'Inknut Antiqua', sans-serif;">Select Type of Volunteering</div>
102     <div class="section">
103       <a class="custom-link button" href="#">Campus Volunteer</a>
104     </div>
105
106     <div class="section">
107       <a class="custom-link button" href="Study Volunteer.html">Study Volunteer</a>
108     </div>
109
110   </div>
111
112 </div>
113
114 <!-- Navigation Bar -->
115 <div class="navbar">
116   <a href="invVolunteer.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
117   <a href="Wi.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
118   <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
119   <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
120
121 </div>
122 </body>
123 </html>
```

```
Terminal Help select Type of volunteering.html - Visual Studio Code
  Welcome Study Volunteer.html select Type of volunteering.html invVolunteer.html rate.html
C:\Users\asma6\Downloads\W\W> select Type of volunteering.html ...
5   <html lang="en">
95   </head>
96   <body>
97
98
99   <div class="content">
100     <div class="section">
101       <div style="font-size: 40px; color: #A27862; font-family: 'Inknut Antiqua', sans-serif;">Select Type of Volunteering</div>
102     <div class="section">
103       <a class="custom-link button" href="#">Campus Volunteer</a>
104     </div>
105
106     <div class="section">
107       <a class="custom-link button" href="Study Volunteer.html">Study Volunteer</a>
108     </div>
109
110   </div>
111
112 </div>
113
114 <!-- Navigation Bar -->
115 <div class="navbar">
116   <a href="invVolunteer.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
117   <a href="Wi.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
118   <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
119   <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
120
121 </div>
122 </body>
123 </html>
```

volunteer information

study volunteer page

by Asma

Study Volunteer

Enter the time:

--- : ---

to:

--- : ---

Enter the location:

Enter the department:

Enter the Subject:

Done

← ⏷ ⏅ ⏆

volunteer information

study volunteer page by Asma

```
C:\> Users > asma6 > Downloads > W > W > Study Volunteer.html > HTML > head > style > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Study Volunteer</title>
7   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
8
9
10 <style>
11   body {
12     margin: 0;
13     padding: 0;
14     font-family: 'Inknuut Antiqua';
15     background-color: #f0f9fa;
16     background-image: url('S1.jpg'); /* URL of the background image */
17     background-size: cover; /* Cover the entire background */
18     background-position: center; /* Center the background image */
19     background-repeat: no-repeat; /* Do not repeat the background image */
20     background-attachment: fixed; /* Fixed background image */
21     text-align: center; /* Aligning the content of the entire page to the center */
22   }
23   /* Style for the navigation bar */
24 .navbar {
25   position: fixed;
26   bottom: 0;
27   left: 0;
28   width: 100%;
29   background-color: #ACB774;
30   text-align: center; /* Align text in the center */
31 }
32
33 /* Style for the navigation bar links */
34 .navbar a {
35   display: inline-block;
36   color: #f2f2f2;
37   padding: 14px 36px;
38 }
```

```
File Edit Selection View Go Run Terminal Help
Study Volunteer.html
C:\> Users > asma6 > Downloads > W > W > Study Volunteer.html > HTML > head > style > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <head>
5 <style>
6
7   /* Change color on hover */
8   .navbar a:hover {
9     background-color: #addc0d;
10    color: black;
11  }
12
13  .container {
14    position: relative;
15    width: 80%;
16    max-width: 600px;
17    margin: 0 auto;
18    text-align: center;
19  }
20
21  .section {
22    margin-bottom: 30px;
23  }
24
25  .input-field {
26    width: calc(100% - 40px);
27    height: 40px;
28    margin: 10px 20px;
29    border: none;
30    border-radius: 20px;
31    padding: 10px;
32    box-sizing: border-box;
33    background-color: #c2d6d6;
34  }
35
36  .button {
37    display: inline-block;
38    background-color: #A27B62;
39    color: white;
40    border: none;
41    padding: 10px 20px;
42    border-radius: 20px;
43    text-decoration: none;
44    font-size: 16px;
45    font-weight: bold;
46  }
47
48  .text {
49    font-size: 14px;
50    font-weight: bold;
51    color: #333;
52  }
53
54  .text p {
55    font-size: 14px;
56    font-weight: normal;
57    color: #666;
58  }
59
60  .text a {
61    color: #333;
62    text-decoration: underline;
63  }
64
65  .text a:hover {
66    color: #A27B62;
67  }
68
69  .text h3 {
70    font-size: 18px;
71    font-weight: bold;
72    color: #333;
73  }
74
75  .text p {
76    font-size: 14px;
77    font-weight: normal;
78    color: #666;
79  }
80
81  .text a {
82    color: #333;
83    text-decoration: underline;
84  }
85
86  .text a:hover {
87    color: #A27B62;
88  }
89
90  .text h3 {
91    font-size: 18px;
92    font-weight: bold;
93    color: #333;
94  }
95
96  .text p {
97    font-size: 14px;
98    font-weight: normal;
99    color: #666;
100 }
```

volunteer information

study volunteer page

by Asma

A screenshot of a code editor showing the HTML code for a study volunteer page. The code includes a CSS section for button styling, a main container with a heading, and three sections for time, location, and department input fields. The code is numbered from 1 to 110.

```
C:\> Users > asma6 > Downloads > W > W > Study Volunteer.html > HTML > Head > Style > Body
3   <html lang="en">
4     <head>
10       <style>
11         .button {
12           transition: background-color 0.3s ease;
13         }
14         .button:hover {
15           background-color: #875f4d;
16         }
17       </style>
18     </head>
19   <body>
20
21     <div class="container">
22       <div class="section">
23         <div style="width: 100%; height: 100%; background: #A27B62; border-radius: 25px;">
24           <p style="margin: 10px; color: white; font-size: 24px;">Study Volunteer</p>
25         </div>
26       </div>
27
28     <form action="test.html" method="POST" id="volunteerForm">
29       <div class="section">
30         <label for="timeFrom">Enter the time:</label>
31         <input type="time" id="timeFrom" name="timeFrom" class="input-field">
32         <label for="timeTo">to:</label>
33         <input type="time" id="timeTo" name="timeTo" class="input-field">
34       </div>
35
36       <div class="section">
37         <label for="location">Enter the location:</label>
38         <input type="text" id="location" name="location" class="input-field">
39       </div>
40
41       <div class="section">
42         <label for="department">Enter the department:</label>
43         <input type="text" id="department" name="department" class="input-field">
44       </div>
45     </form>
46   </div>
47
48 </body>
49
50 </html>
```

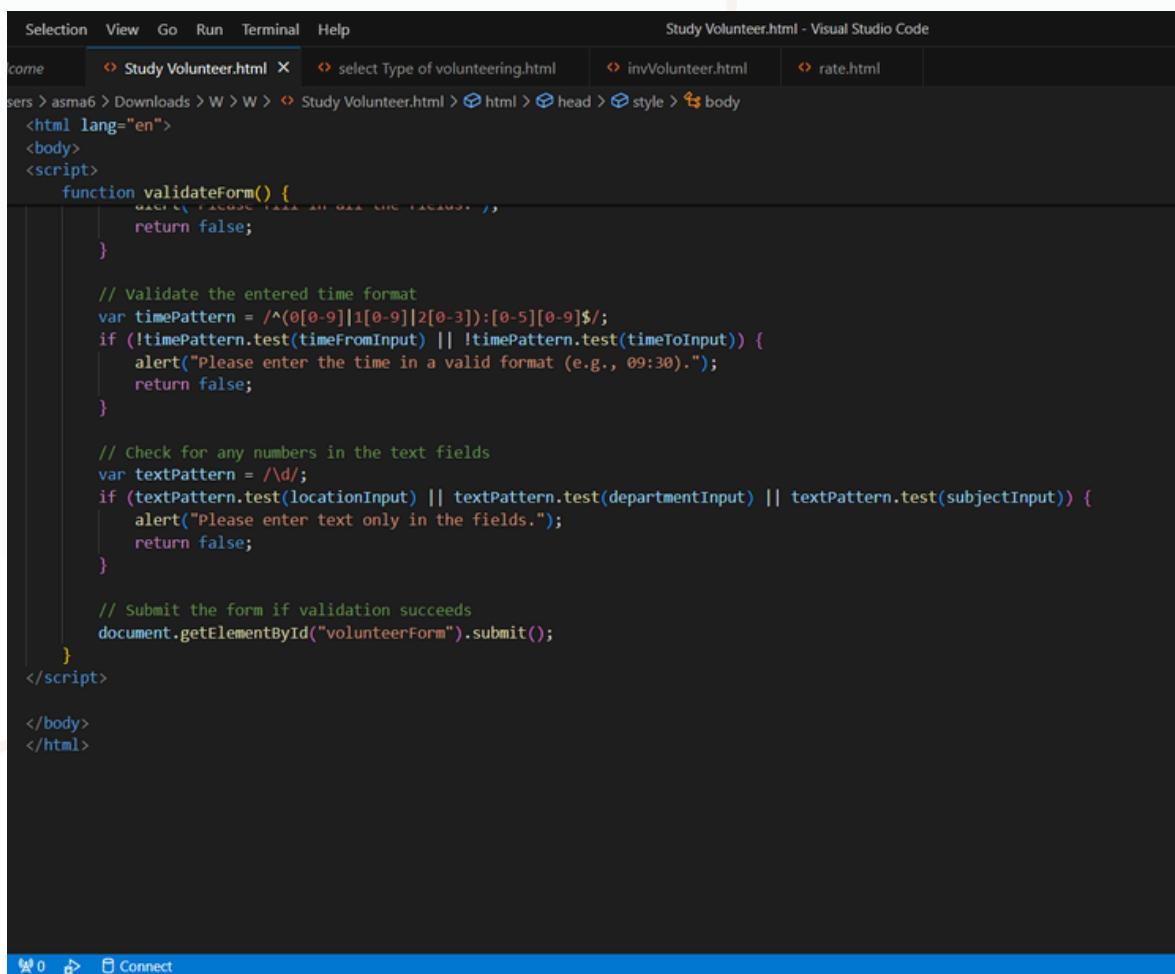
A screenshot of Visual Studio Code showing the HTML code for a study volunteer page. The code includes a CSS section for button styling, a main container with a heading, and three sections for time, location, and department input fields. A script block at the bottom contains a validateForm() function that checks if all fields are filled. The code is numbered from 1 to 145.

```
Edit Selection View Go Run Terminal Help
Study Volunteer.html - Visual Studio Code
C:\> Users > asma6 > Downloads > W > W > Study Volunteer.html > HTML > Head > Style > Body
3   <html lang="en">
4     <body>
8     <div class="container">
9       <form action="test.html" method="POST" id="volunteerForm">
10      <div class="section">
11        <label for="subject">Enter the Subject:</label>
12        <input type="text" id="subject" name="subject" class="input-field">
13      </div>
14
15      <div class="section">
16        <button type="button" class="button" onclick="validateForm()>Done</button>
17      </div>
18    </form>
19  </div>
20
21  <!-- Navigation Bar -->
22  <div class="navbar">
23    <a href="select Type of volunteering.html"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
24    <a href="W1.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
25    <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
26    <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
27  </div>
28
29  <script>
30    function validateForm() {
31      var timeFromInput = document.getElementById("timeFrom").value;
32      var timeToInput = document.getElementById("timeTo").value;
33      var locationInput = document.getElementById("location").value;
34      var departmentInput = document.getElementById("department").value;
35      var subjectInput = document.getElementById("subject").value;
36
37      if (timeFromInput === "" || timeToInput === "" || locationInput === "" || departmentInput === "" || subjectInput === "") {
38        alert("Please fill in all the fields.");
39        return false;
40      }
41
42      // Validate the entered time format
43      var timePattern = /(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]$/;
```

volunteer information

study volunteer page

by Asma



The screenshot shows a Visual Studio Code window with the title "Study Volunteer.html - Visual Studio Code". The code editor displays the following script:

```
Selection View Go Run Terminal Help
Study Volunteer.html - Study Volunteer.html - select Type of volunteering.html - invVolunteer.html - rate.html
Study Volunteer.html - Visual Studio Code

come
<html lang="en">
<body>
<script>
    function validateForm() {
        // Check for empty fields
        if (timeFromInput.value === "" || timeToInput.value === "") {
            alert("Please enter start and end times.");
            return false;
        }

        // Validate the entered time format
        var timePattern = /^(0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]$/;
        if (!timePattern.test(timeFromInput.value) || !timePattern.test(timeToInput.value)) {
            alert("Please enter the time in a valid format (e.g., 09:30).");
            return false;
        }

        // Check for any numbers in the text fields
        var textPattern = /\d/;
        if (textPattern.test(locationInput.value) || textPattern.test(departmentInput.value) || textPattern.test(subjectInput.value)) {
            alert("Please enter text only in the fields.");
            return false;
        }

        // Submit the form if validation succeeds
        document.getElementById("volunteerForm").submit();
    }
</script>

</body>
</html>
```

volunteer information

Rate page
by Asma

Please click on the
following button to
complete

Rate the Volunteer



volunteer information

Rate page by Asma

```
C:\> Users > asma6 > Downloads > W > W > rate.html > html > head > style > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Rate Volunteer</title>
7    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
8
9    <style>
10   body {
11     margin: 0;
12     padding: 0;
13     font-family: 'Arial', sans-serif;
14     background-color: #f8f9fa;
15     background-image: url('sc1.jpg'); /* URL of the background image */
16     background-size: cover; /* Cover the entire background */
17     background-position: center; /* Center the background image */
18     background-repeat: no-repeat; /* Do not repeat the background image */
19     background-attachment: fixed; /* Fixed background image */
20     text-align: center; /* Aligning the content of the entire page to the center */
21     margin: 0; /* Remove default margin */
22     padding: 0; /* Remove default padding */
23   }
24   /* Style for the navigation bar */
25   .navbar {
26     position: fixed;
27     bottom: 0;
28     left: 0;
29     width: 100%;
30     background-color: #AC8774;
31     text-align: center; /* Align text in the center */
32   }
33
34   /* Style for the navigation bar links */
35   .navbar a {
36     display: inline-block;
37     color: #f2f2f2;
```

△ 0 ━ 0 ⌂ Connect

```
C:\> Users > asma6 > Downloads > W > W > rate.html > html > head > style > heading
2  <html lang="en">
3  <head>
9  <style>
42
43   /* Change color on hover */
44   .navbar a:hover {
45     background-color: #ddd;
46     color: black;
47   }
48
49   .container {
50     position: absolute;
51     top: 50%;
52     left: 50%;
53     transform: translate(-50%, -50%);
54     text-align: center;
55   }
56
57   .heading {
58     font-size: 2rem;
59     color: #A27B62;
60     margin-bottom: 20px;
61   }
62
63   .button-container {
64     display: flex;
65     justify-content: center;
66     align-items: center;
67     margin-bottom: 20px;
68   }
69
70   .button {
71     background-color: #A27B62;
72     color: white;
73     border: none;
74     padding: 15px 30px;
75     border-radius: 50px;
```

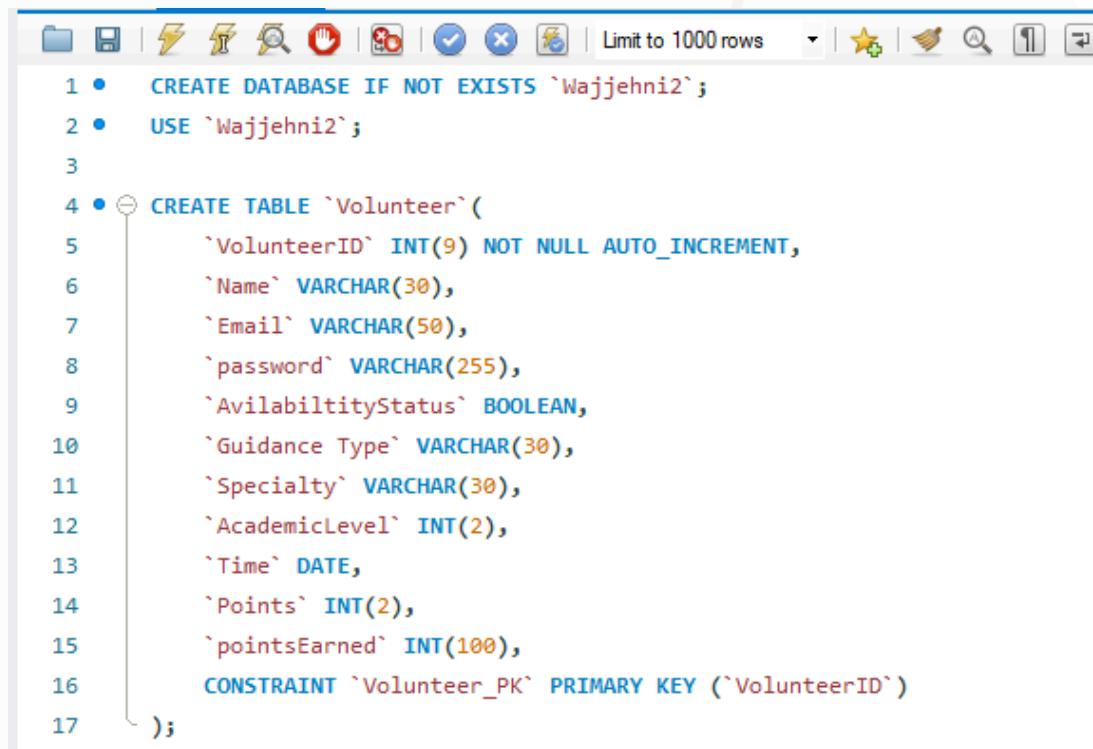
△ 0 ━ 0 ⌂ Connect

volunteer information

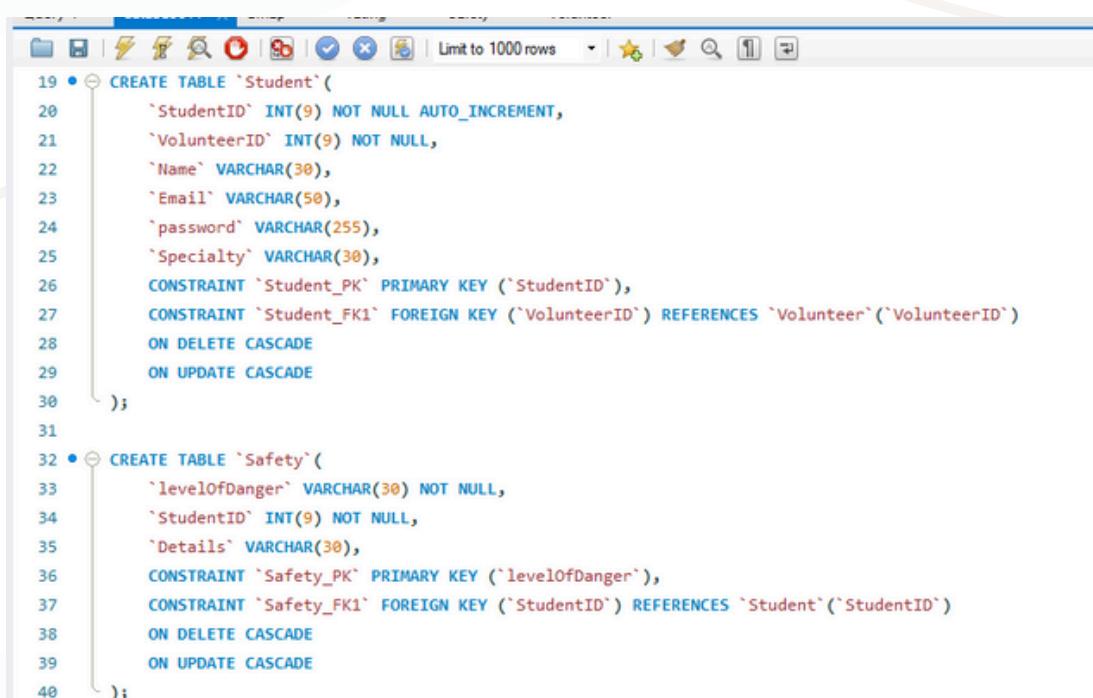
Rate page
by Asma

```
C: > Users > asma6 > Downloads > W > W > rate.html > html > head > style > .heading
 2   <html lang="en">
 3     <head>
 4       <style>
 5         .button:hover {
 6           background-color: #875f4d;
 7         }
 8
 9         @media (max-width: 768px) {
10           .heading {
11             font-size: 1.5rem;
12           }
13
14           .button {
15             font-size: 1rem;
16           }
17         }
18       </style>
19     </head>
20     <body>
21       <div class="container">
22         <div class="heading">Please click on the following button to complete</div>
23
24         <div class="button-container">
25           <a class="button" href="Rating.html">Rate the Volunteer</a>
26         </div>
27       </div>
28
29         <!-- Navigation Bar -->
30       <div class="navbar">
31         <a href="#"><i class="fas fa-arrow-left"></i></a> <!-- Arrow pointing left icon -->
32         <a href="W1.html"><i class="fas fa-calendar-alt"></i></a> <!-- Calendar icon -->
33         <a href="test.html"><i class="fas fa-map-marker-alt"></i></a> <!-- Location icon -->
34         <a href="#"><i class="fas fa-user"></i></a> <!-- Profile icon -->
35       </div>
36     </body>
37   </html>
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
```

Database table



```
1 • CREATE DATABASE IF NOT EXISTS `Wajjehni2`;
2 • USE `Wajjehni2`;
3
4 • CREATE TABLE `Volunteer`(
    `VolunteerID` INT(9) NOT NULL AUTO_INCREMENT,
    `Name` VARCHAR(30),
    `Email` VARCHAR(50),
    `password` VARCHAR(255),
    `AvailabiltyStatus` BOOLEAN,
    `Guidance Type` VARCHAR(30),
    `Specialty` VARCHAR(30),
    `AcademicLevel` INT(2),
    `Time` DATE,
    `Points` INT(2),
    `pointsEarned` INT(100),
    CONSTRAINT `Volunteer_PK` PRIMARY KEY (`VolunteerID`)
);
17
```



```
19 • CREATE TABLE `Student`(
    `StudentID` INT(9) NOT NULL AUTO_INCREMENT,
    `VolunteerID` INT(9) NOT NULL,
    `Name` VARCHAR(30),
    `Email` VARCHAR(50),
    `password` VARCHAR(255),
    `Specialty` VARCHAR(30),
    CONSTRAINT `Student_PK` PRIMARY KEY (`StudentID`),
    CONSTRAINT `Student_FK1` FOREIGN KEY (`VolunteerID`) REFERENCES `Volunteer`(`VolunteerID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
31
32 • CREATE TABLE `Safety`(
    `levelOfDanger` VARCHAR(30) NOT NULL,
    `StudentID` INT(9) NOT NULL,
    `Details` VARCHAR(30),
    CONSTRAINT `Safety_PK` PRIMARY KEY (`levelOfDanger`),
    CONSTRAINT `Safety_FK1` FOREIGN KEY (`StudentID`) REFERENCES `Student`(`StudentID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
40
```

Database table

```
42 • CREATE TABLE `UQU Notification`(
43     `EventName` VARCHAR(30) NOT NULL,
44     `StudentID` INT(9) NOT NULL,
45     `EventDate` DATE,
46     `EventLocation` VARCHAR(50),
47     `EventType` VARCHAR(15),
48     `EventProvider` VARCHAR(20),
49     `CertificateAvailability` BOOLEAN,
50     CONSTRAINT `UQU Notification_PK` PRIMARY KEY (`EventName`),
51     CONSTRAINT `UQU Notification_FK1` FOREIGN KEY (`StudentID`) REFERENCES `Student`(`StudentID`)
52     ON DELETE CASCADE
53     ON UPDATE CASCADE
54 );
55
56 • CREATE TABLE `Market Notification`(
57     `MarketName` VARCHAR(30) NOT NULL,
58     `StudentID` INT(9) NOT NULL,
59     `Date` DATE,
60     `DiscountDetails` INT(3),
61     `SpeicalDeals` INT(3),
62     CONSTRAINT `Market Notification_PK` PRIMARY KEY (`MarketName`),
63     CONSTRAINT `Market Notification_FK1` FOREIGN KEY (`StudentID`) REFERENCES `Student`(`StudentID`)
64     ON DELETE CASCADE
65     ON UPDATE CASCADE
```

```
68 • CREATE TABLE `Rating`(
69     `RateID` INT(9) NOT NULL AUTO_INCREMENT,
70     `StudentID` INT(9) NOT NULL,
71     `VolunteerID` INT(9) NOT NULL,
72     `NumberOfStars` INT(3),
73     `RatingList` VARCHAR(40),
74     `Feedback` VARCHAR(200),
75     CONSTRAINT `Rating_PK` PRIMARY KEY (`RateID`),
76     CONSTRAINT `Rating_FK1` FOREIGN KEY (`StudentID`) REFERENCES `Student`(`StudentID`),
77     CONSTRAINT `Rating_FK2` FOREIGN KEY (`VolunteerID`) REFERENCES `Volunteer`(`VolunteerID`)
78     ON DELETE CASCADE
79     ON UPDATE CASCADE
80 );
```

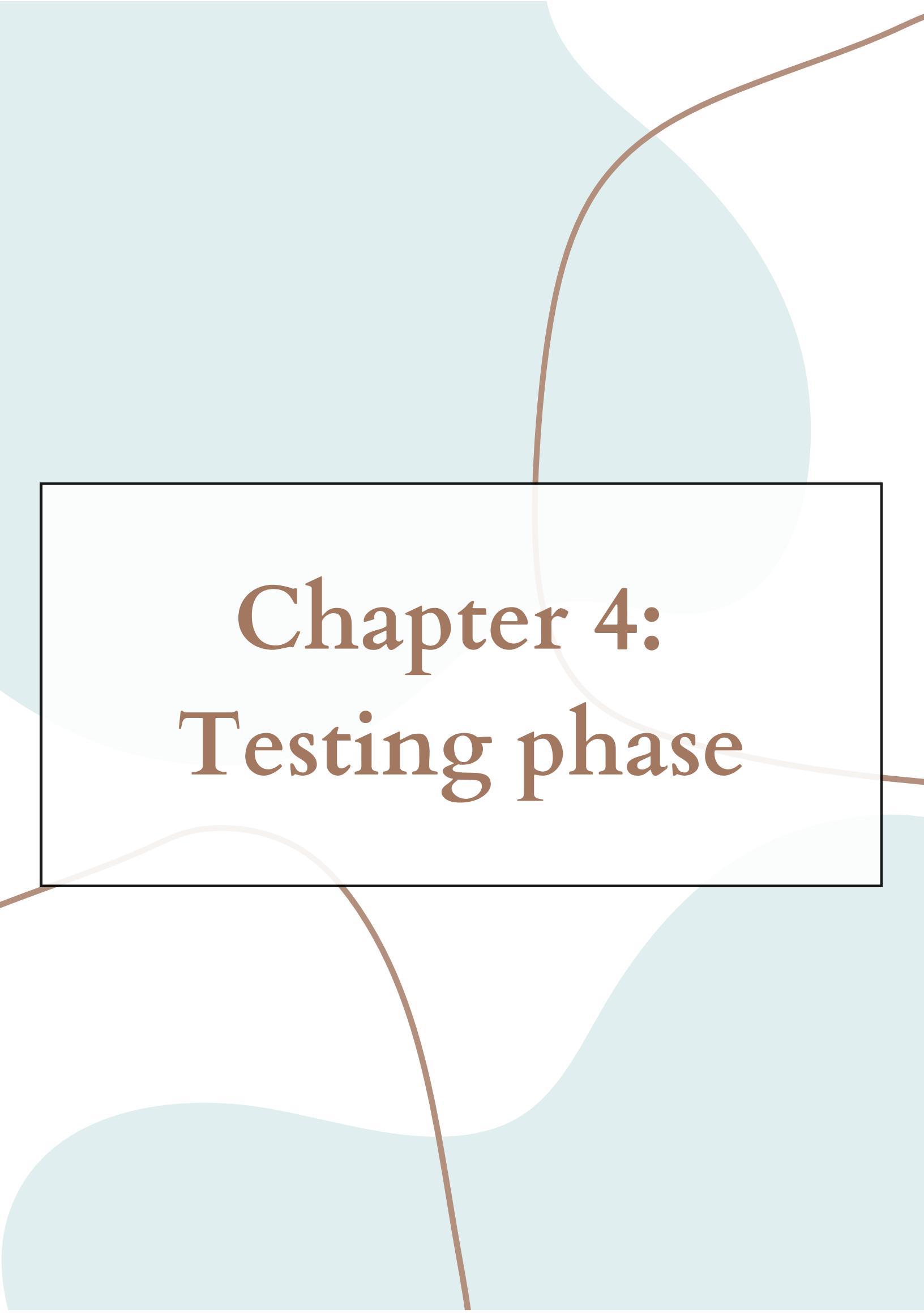
Database table

```
82 • CREATE TABLE `Map`(
83     `Location` VARCHAR(60) NOT NULL,
84     `StudentID` INT(9) NOT NULL,
85     `VolunteerID` INT(9) NOT NULL,
86     `EventName` VARCHAR(30) NOT NULL,
87     `MarketName` VARCHAR(30) NOT NULL,
88     `levelOfDanger` VARCHAR(30) NOT NULL,
89     `MapScale` VARCHAR(40),
90     `MapMarkers` VARCHAR(40),
91     `MapImage` VARCHAR(40),
92     `latitude` DECIMAL(10, 8) NOT NULL,
93     `longitude` DECIMAL(11, 8) NOT NULL,
94     CONSTRAINT `Map_PK` PRIMARY KEY (`Location`),
95     CONSTRAINT `Map_FK1` FOREIGN KEY (`StudentID`) REFERENCES `Student`(`StudentID`),
96     CONSTRAINT `Map_FK2` FOREIGN KEY (`VolunteerID`) REFERENCES `Volunteer`(`VolunteerID`),
97     CONSTRAINT `Map_FK3` FOREIGN KEY (`EventName`) REFERENCES `UQU Notification`(`EventName`),
98     CONSTRAINT `Map_FK4` FOREIGN KEY (`MarketName`) REFERENCES `Market Notification`(`MarketName`),
99     CONSTRAINT `Map_FK5` FOREIGN KEY (`levelOfDanger`) REFERENCES `Safety`(`levelOfDanger`)
100    ON DELETE CASCADE
101    ON UPDATE CASCADE
102 );
```

```
104 • insert into `Volunteer`(`VolunteerID`, `Name`, `Email`, `password`, `Specialty`, `AvailabiltyStatus`, `Guidance Type`, `AcademicLevel`, `Time`, `Points`, `pointsEarned`)
105 values
106     (1111111111, 'AAA', 'Arwa@example.com', 'password111', 'IS', true, 'Campus volunteer', 8, '2024-05-10', 10, 10);
107
108 • insert into `Student`(`StudentID`, `VolunteerID`, `Name`, `Email`, `password`, `Specialty`)
109 values
110     (123456789, 1111111111, 'Arwa', 'ArwaAsiri@example.com', 'password123', 'IS');
111
112 • INSERT INTO `UQU Notification` VALUES ('91,اطباء,جامعة الملك عبدالعزيز', '123456789', '2021-09-28', 'مكالمات', 'نعم', 'UQU', FALSE);
```

Demo

To view the demo
video,
[click here](#)



Chapter 4: Testing phase

Test Case

Razan

Test Case ID	TC01	TC02
Test Case Description	checking if the user click that button the system will display the right page.	checking if the user click the calendar icon from the navigation bar The system will display the notification page
Test Steps	1-Go to the Wajjehni application 2-go to the notification page 3-click campus event button	1-go to the Wajjehni application 2-go to navigation bar 3-Click the calendar icon
Input	Click the Campus Event Button	Click the calendar icon button.
Expected output	Display the campus event page.	Display the notification page.
Actual Output	The campus events are displayed.	the notification page appears
Status	Pass	Pass

Test Case

Ghala

Test Case ID	TC03	TC04	TC05
Test Case Description	Checking if the user enter invalid input	Checking if the user didn't choose from the user Type.	Checking if the user clicked logIn button without entering anything
Test Steps	1-Go to the Wajjehni application 2- Go to logIn page. 3- try to type an invalid email.	1-Go to the Wajjehni application 2- Go to log In page. 3- enter the email ,password, then click login button without choosing user Type.	1-Go to the Wajjehni application 2- Go to log In page. 3- clicked log In button without entering anything
Input	type an invalid email.	input email and password, click on login button without choosing user Type.	empty entry
Expected output	Display the error message.	Display the error message.	Display the error message.
Actual Output	The error message is displayed.	The error message is displayed.	The error message is displayed.
Status	Pass	Pass	Pass

Test Case

Atyaf

Test Case ID	TC06	TC07
Test Case Description	Verify whether the user has clicked the button based on the issue size, and then display the correct page.	Verify whether the user has clicked the button based on the issue size, and then display the correct page.
Test Steps	1-Go to the Wajihni application. 2-Go to the map page. 3-Click on the safety button. 4-Click on the Help.	1-go to the Wajjehni application. 2-go to the map page. 3-Click on the safety button. 4-click on the simple intervention.
Input	Click the button .	click the button.
Expected output	display help page.	display simple intervention.
Actual Output	Help page.	simple intervention.
Status	Pass.	Pass.

Test Case

Asma

Test Case ID	TC08	TC09	TC10
Test Case Description	Checking if the user entered all data correctly.	There are two buttons Specifies the type of volunteer	Checking if the user entered all data correctly.
Test Steps	1-Go to the Wajjehni application 1- enter the information, then Click the volunteering button. 2- Go to Type of volunteering page.	1-Go to the Wajjehni application 1- clicked Study Volunteer button 3- Go to Study Volunteer page.	1-Go to the Wajjehni application 2- enter the information, then Click the Done button. 3- Go to map page.
Input	The user enters the following data (name, email, Department, academic level, points, and type of volunteer)and Click the volunteering button.	Click the Study Volunteer button	The user enters the following data (time,location,department, subject)and Click the volunteering button.
Expected output	Display the Type of volunteering page.	Display the the Study Volunteer page.	Display the map page.
Actual Output	The Type of volunteering are displayed	The study Volunteer are displayed	The map are displayed
Status	Pass	Pass	Pass

Test Case

Arwa

Test Case ID	TC11	TC12	TC13
Test Case Description	Checking the responsiveness of the rating stars icons.	Inserting additional comments in the textarea and submitting the form.	Checking the responsiveness of the Feedback buttons.
Test Steps	<ol style="list-style-type: none"> 1-Go to the Wajjehni application 2- From the map interface, click on the rating icon on the top left corner. 2- click on the desired number of stars. 	<ol style="list-style-type: none"> 1-Go to the Wajjehni application 2- From the map interface, click on the rating icon on the top left corner. 3- Write comments in the textarea. 	<ol style="list-style-type: none"> 1-Go to the Wajjehni application 2- enter the information, then Click the Done button. 3- Selected needed feedback.
Input	The user clicks on the stars that represents the rating input.	Text input in the textarea, then clicking on submit button.	The user clicks on the buttons that represents desired feedback.
Expected output	Color of selected stars will be changed from black to yellow.	Feedback confirmed page is displayed when submit button is clicked.	Color of selected buttons will change into a darker shade.
Actual Output	Color of selected stars is changed.	The confirmation page is displayed.	Color of selected buttons is changed.
Status	Pass	Pass	Pass

Test Case

Renad

Test Case ID	TC14	TC15
Test Case Description	When the user enters the application, the map page appears to them and click the map mark	When the user enters the application, the map page appears to them and click the rate mark.
Test Steps	1-Go to the Wajjehni application 2-sign up/log in 3-Click the map mark	1-Go to the Wajjehni application 2-sign up/log in 3- Click the rate mark
Input	Click the map mark	Click the rate mark
Expected output	Show the popup page to Display volunteer information	Show the Rate page
Actual Output	Show the popup page to Display volunteer information	Show the Rate page
Status	Pass	Pass

Chapter 5 :

Evolution

Campus event

By Razan

W2.css

```
W2.css > .navbar
1 /* Style for the navigation bar */
2 .navbar {
3     position: fixed;
4     bottom: 0;
5     left: 0;
6     width: 100%;
7     background-color: #ACB774;
8     text-align: center; /* Align text in the center */
9 }
10 /* Style for the navigation bar links */
11 .navbar a {
12     display: inline-block;
13     color: #F2F2F2;
14     padding: 14px 36px;
15     text-decoration: none;
16 }
17 /* Change color on hover */
18 .navbar a:hover {
19     background-color: #ddd;
20     color: #black;
21 }
22 /* Background image */
23 body {
24     background-image: url('Sci.png'); /* URL of the background image */
25     background-size: cover; /* Cover the entire background */
26     background-position: center; /* Center the background image */
27     background-repeat: no-repeat; /* Do not repeat the background image */
28     background-attachment: fixed; /* Fixed background image */
29     text-align: center; /* Aligning the content of the entire page to the center */
30     margin: 0; /* Remove default margin */
31     padding: 0; /* Remove default padding */
32 }
```

W3.css

```
W3.css > .
1 /* Style for the navigation bar */
2 .navbar {
3     position: fixed;
4     bottom: 0;
5     left: 0;
6     width: 100%;
7     background-color: #ACB774;
8     text-align: center; /* Align text in the center */
9 }
10 /* Style for the navigation bar links */
11 .navbar a {
12     display: inline-block;
13     color: #F2F2F2;
14     padding: 14px 36px;
15     text-decoration: none;
16 }
17 /* Change color on hover */
18 .navbar a:hover {
19     background-color: #ddd;
20     color: #black;
21 }
22 /* Background image */
23 body {
24     background-image: url('Sci.png'); /* URL of the background image */
25     background-size: cover; /* Cover the entire background */
26     background-position: center; /* Center the background image */
27     background-repeat: no-repeat; /* Do not repeat the background image */
28     background-attachment: fixed; /* Fixed background image */
29     text-align: center; /* Aligning the content of the entire page to the center */
30     margin: 0; /* Remove default margin */
31     padding: 0; /* Remove default padding */
32 }
```

1-Bad smell: Duplicate code

Description: the two css code W2 &W3 are very similar code

Refactoring: we will use just one of them for html codes

Before

```
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>" . $row['EventName'] . "</td>";
    echo "<td>" . $row['EventDate'] . "</td>";
    echo "<td>" . $row['EventLocation'] . "</td>";
    echo "<td>" . $row['EventType'] . "</td>";
    echo "<td>" . $row['EventProvider'] . "</td>";
    echo "<td>" . ($row['CertificateAvailability'] ? "Available" : "Not Available") . "</td>";
    echo "</tr>";
}
```

After

```
while($row = mysqli_fetch_assoc($result)) {
    echo "<tr>";
    echo "<td>" . htmlspecialchars($row['EventName']) . "</td>";
    echo "<td>" . htmlspecialchars($row['EventDate']) . "</td>";
    echo "<td>" . htmlspecialchars($row['EventLocation']) . "</td>";
    echo "<td>" . htmlspecialchars($row['EventType']) . "</td>";
    echo "<td>" . htmlspecialchars($row['EventProvider']) . "</td>";
    echo "<td>" . ($row['CertificateAvailability'] ? "Available" : "Not Available") . "</td>";
    echo "</tr>";
}
```

2-Improved Security with `htmlspecialchars`:

Description: The code was vulnerable to attack by XSS When retrieving data from the database and displaying it on the page.

- Refactoring: Used “htmlspecialchars” to prevent XSS attacks when outputting data from the database.

Safety By Atyaf

Before

submit.html

After

```
1 <!DOCTYPE HTML>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <link rel="stylesheet" href="safetytest1.css" >
7     <title>Urgent</title>
8     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
9   </head>
10  <body>
11    <div class="title">
12      <h2>Safety</h2>
13    </div>
14    <div class="box">
15      <form action="http://localhost/M/insert_safety1.php" method="post">
16        <input type="text" name="urgent" placeholder="enter the text"><br>
17        <br>
18        <button class="button submit" type="submit">Submit</button>
19      </form>
20    </div>
```

```
2   <!DOCTYPE HTML>
3   <html lang="en">
4     <head>
5       <meta charset="UTF-8">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <link rel="stylesheet" href="safetytest1.css" >
8       <!--title-->
9       <title>Urgent</title>
10      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
11    </head>
12    <body>
13      <!--title for the brown box-->
14      <div class="title">
15        <h2>Safety</h2>
16      </div>
17      <!--Style the dropdown button -->
18      <div class="box">
19        <!--Create a form that allows users to submit data to a server-side script-->
20        <form action="http://localhost/M/insert_safety1.php" method="post">
21          <!--The box created to describe the problem-->
22          <input type="text" name="urgent" placeholder="enter the text"><br>
23          <br>
24          <!--Submit Issue to volunteer Button-->
25          <button class="button submit" type="submit">Submit</button>
26      </form>
```

1. Bad Smell: No Comments

- Description: The code has no comments .
- Refactoring: Add comments to make it easier for the code developer to modify in the future.

Map By Renad

Before

```
var map;
var marker;
var previousLocation;
// Function to initialize the map
function initMap() {
  var initialLocation = { lat: 37.7749, lng: -122.4194 }; // Default to San Francisco

  // Create a new map
  map = new google.maps.Map(document.getElementById('map'), {
    zoom: 14,
    center: initialLocation
  });

  var volunteerType = "guidance";
  var markerColor;

  if (volunteerType === "academic") {
    markerColor = "green";
  } else if (volunteerType === "guidance") {
    markerColor = "brown";
  } else {
    markerColor = "blue";
  }
}
```

After

```
var map; // Variable to store the map
var marker; // Variable to store the marker (location) on the map
var previousLocation; // Variable to store the previous location

// Function to initialize the map
function initMap() {
  // Default settings for the map (San Francisco)
  var initialLocation = { lat: 37.7749, lng: -122.4194 };

  // Create a new map
  map = new google.maps.Map(document.getElementById('map'), {
    zoom: 14,
    center: initialLocation
  });

  // Determine the volunteer type
  var volunteerType = "guidance";
  var markerColor;

  // Set marker color based on volunteer type
  if (volunteerType === "academic") {
    markerColor = "green";
  } else if (volunteerType === "guidance") {
    markerColor = "brown";
  } else {
    markerColor = "blue";
  }
}
```

1. Bad Smell: No Comments

- Description: The code has no comments .
- Refactoring: Add comments to make it easier for the code developer to modify in the future.



```
<!DOCTYPE html>
<html>

<head>
  <title>Interactive Map</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDROPDWFgd61DCD1B5Noex2dR1bi5Ovi1o"></script>
  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
    integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvyo16VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r"
    crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
    integrity="sha384-Q6EeZ8hUVIMTlrxqJF7yvHsDyRzjYDqyZDcZDwqOOGabYfZo0T0to5eqrptly"
    crossorigin="anonymous"></script>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>...
  </script>

  <style>...
  </style>
</head>
<body>
```

2. Bad Smell: Mixing HTML ,CSS and JS in Same File

- Description: HTML and CSS and JS code are merged into one file, resulting in reduced maintainability and readability.
- Refactoring: Separate HTML ,CSS and JS code into distinct files

Log in By Ghala

Before

CSS

After

```
body {
  font-family: Arial, sans-serif; background-image: url('pics/BGse.png');
  background-size: cover; background-position: center; background-repeat: no-repeat;
}

.wrapper {
  display: flex; align-items: center; justify-content: center; height: 100vh; text-align: center;
}

.logo {
  margin-bottom: 20px;
}

.input-cont {
  width: 250px; max-width: 300px; display: flex;
  flex-direction: row; align-items: center; text-align: center;
  margin-bottom: 10px; margin-right: auto; margin-left: auto;
}

input {
  width: 100%; padding: 10px;
}

form {
  text-align: center;
}

.sentence {
  margin-top: 20px; color: #a7a7a7;
}
```

```
/* Global Styles */
body {
  font-family: Arial, sans-serif;
  background-image: url('pics/BGse.png');
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
}

/* Layout Styles */
.wrapper {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  text-align: center;
}

/* Component Styles */
.logo {
  margin-bottom: 20px;
}

/* Input container Styles */
.input-cont {
  width: 250px;
  max-width: 300px;
  display: flex;
  flex-direction: row;
  align-items: center;
  text-align: center;
  margin: 0 auto 10px;
}

.input-container input {
  width: 100%;
  padding: 10px;
}

.form {
  text-align: center;
}
```

1. Bad Smell: No Comments

- Description: The code has no comments .
- Refactoring: Add comments to make it easier for the code developer to modify in the future.

Before

After

```
.input-cont {
  width: 250px; max-width: 300px; display: flex;
  flex-direction: row; align-items: center; text-align: center;
  margin-bottom: 10px; margin-right: auto; margin-left: auto;
}

input {
  width: 100%; padding: 10px;
}

form {
  text-align: center;
}

.sentence {
  margin-top: 20px; color: #a7a7a7;
}
```

```
.input-container {
  width: 250px;
  max-width: 300px;
  display: flex;
  flex-direction: row;
  align-items: center;
  text-align: center;
  margin: 0 auto 10px;
}

.input-container input {
  width: 100%;
  padding: 10px;
}

.form {
  text-align: center;
}
```

2. Bad Smell: Inefficient use of selectors

- Description: The CSS code uses overly specific selectors, which can lead to performance issues and make the code harder to maintain.
- Refactoring: Removed the inefficient use of selectors in the original code. For example, instead of `input { ... }`, the selector was changed to `.input-container input { ... }`

Log in By Ghala

Before

HTML

After

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Demo</title>
    <link rel="stylesheet" href="demo.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

<body>
    <div class="wrapper">
        <form onsubmit="return validateForm()" method="POST" action="http://localhost/W/Dphp.php">
            

            <div class="input-cont">
                <input type="email" id="email" name="email" placeholder="Email" required>
            </div>

            <div class="input-cont">
                <input type="password" id="password" name="password" placeholder="Password" required>
            </div>

            <div class="input-cont">
                <select id="userstype" name="userstype" required>
                    <option value="" disabled selected>Select user type</option>
                    <option value="student">Student</option>
                    <option value="volunteer">Volunteer</option>
                </select>
            </div>
        </form>
    </div>
</body>
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Demo</title>
    <link rel="stylesheet" href="demo.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
    <div class="wrapper">
        <form id="loginForm" method="POST" action="http://localhost/W/Dphp.php">
            <!-- Logo -->
            

            <!-- Email input -->
            <div class="input-cont">
                <input type="email" id="email" name="email" placeholder="Email" required>
                <span id="emailError" class="error-message"></span>
            </div>

            <!-- Password input -->
            <div class="input-cont">
                <input type="password" id="password" name="password" placeholder="Password" required>
                <span id="passwordError" class="error-message"></span>
            </div>

            <!-- User type selection -->
            <div class="input-cont">
                <select id="userstype" name="userstype" required>
                    <option value="" disabled selected>Select user type</option>
                </select>
            </div>
        </form>
    </div>
</body>
```

1. Bad Smell: No Comments

- Description: The code has no comments .
- Refactoring: Add comments to make it easier for the code developer to modify in the future.

Before

After

```
<html>
<body>
    <div class="wrapper">
        <form id="loginForm" method="POST" action="http://localhost/W/Dphp.php">

            <!-- Submit button -->
            <button type="submit" class="bt">log in</button>

            <!-- Sign up link -->
            <p class="sentence">Don't have an account? <a href="invVolunteer.html">Sign up</a></p>
        </form>
    </div>

    <script>
        function validateForm() {
            var email = document.getElementById("email").value;
            var password = document.getElementById("password").value;

            // Email validation
            var emailPattern = /^[^@\s]+@[^\s]+\.[^\s@]+\$/;
            if (!emailPattern.test(email)) {
                alert("Please enter a valid email address.");
                return false;
            }

            // Password validation
            if (password.length < 8) {
                alert("Password must be at least 8 characters long.");
                return false;
            }
        }
    </script>

```

```
File: Login.html
File: Login.js

Login.html
Login.js > validateForm

1  function validateForm() {
2      var email = document.getElementById("email").value;
3      var password = document.getElementById("password").value;
4
5
6      // Email validation
7      var emailPattern = /^[^@\s]+@[^\s]+\.[^\s@]+\$/;
8      if (!emailPattern.test(email)) {
9          alert("Please enter a valid email address.");
10         return false;
11     }
12
13     // Password validation
14     if (password.length < 8) {
15         alert("Password must be at least 8 characters long.");
16         return false;
17     }
18
19     // Form submission
20     // This is just an example to demonstrate form submission.
21     alert("Form submitted successfully.");
22     return true;
}
```

2. Bad Smell: Mixing HTML and JS in the Same File

- Description: HTML and JS code are merged into one file, resulting in reduced maintainability and readability.
- Refactoring: Separate HTML and JS code into distinct files

Rating By Arwa

ratingStyle.css

Ratin.html

```
<link rel="stylesheet" href="ratingStyle.css">
```

FeedbackConfirmed.html

```
<link rel="stylesheet" href="ratingStyle.css">
```

1. Bad Smell: Duplicat code

- Description: Ratin.html and FeedbackConfirmed.html has the same embedded styles.
- Refactoring: Combining the styles into one seperate css file and linking it to html files via `<link>` tag, then removing the `<style>` tags in the html files.

Before

```
function toggleButton(button) {
  button.classList.toggle("active");
}

function toggleStar(star) {
  var stars = document.querySelectorAll('.fa-star');
  var index = Array.prototype.indexOf.call(stars, star);
  for (var i = 0; i <= index; i++) {
    stars[i].classList.add("checked");
  }
  for (var i = index + 1; i < stars.length; i++) {
    stars[i].classList.remove("checked");
  }
}
```

After

```
function toggleStar(star) {
  var stars = document.querySelectorAll('.fa-star');
  var index = Array.prototype.indexOf.call(stars, star);

  stars.forEach((el, i) => {
    if (i <= index) {
      el.classList.add("checked");
    } else {
      el.classList.remove("checked");
    }
  });
}
```

2. Bad Smell: Complex functions

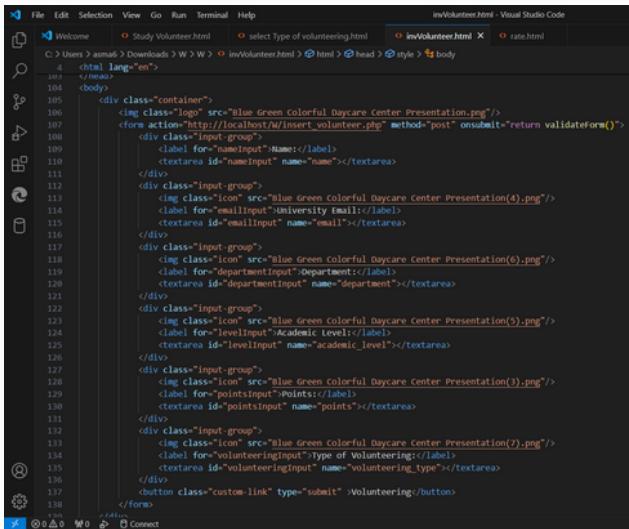
- Description: The `toggleStar` function iterates over the stars twice: once to add the "checked" class and once to remove it using two for loops.
- Refactoring: Rewrite the function using `forEach` to make the code more efficient and less complex.

information volunteer

By Asma

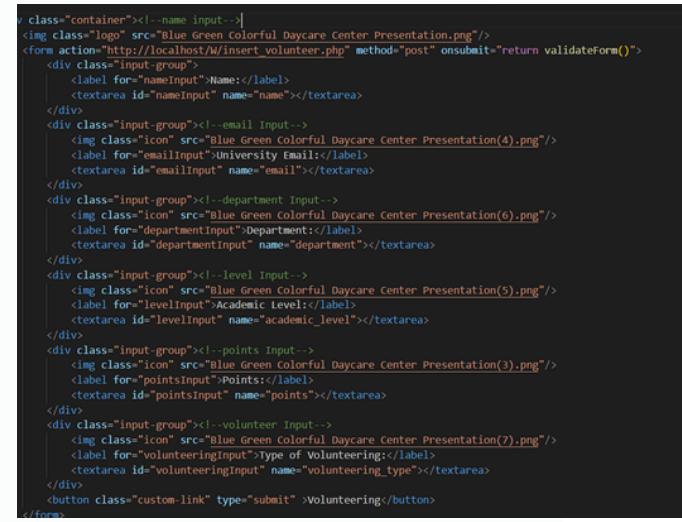
Before

After



```
C:\Users\Asma\Downloads>W>W> invVolunteer.html > select Type of volunteering.html > invVolunteer.html > rate.html
File Edit Selection View Go Run Terminal Help
Welcome Study Volunteer.html select Type of volunteering.html invVolunteer.html rate.html
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
    <div class="container">
        
        <form action="http://localhost/W/insert_volunteer.php" method="post" onsubmit="return validateForm()">
            <div class="input-group">
                <label for="nameInput">Name:</label>
                <textarea id="nameInput" name="name"></textarea>
            </div>
            <div class="input-group">
                
                <label for="emailInput">University Email:</label>
                <textarea id="emailInput" name="email"></textarea>
            </div>
            <div class="input-group">
                
                <label for="departmentInput">Department:</label>
                <textarea id="departmentInput" name="department"></textarea>
            </div>
            <div class="input-group">
                
                <label for="levelInput">Academic Level:</label>
                <textarea id="levelInput" name="academic_level"></textarea>
            </div>
            <div class="input-group">
                
                <label for="pointsInput">Points:</label>
                <textarea id="pointsInput" name="points"></textarea>
            </div>
            <div class="input-group">
                
                <label for="volunteeringInput">Type of Volunteering:</label>
                <textarea id="volunteeringInput" name="volunteering_type"></textarea>
            </div>
            <button class="custom-link" type="submit" >Volunteering</button>
        </form>
    </div>

```



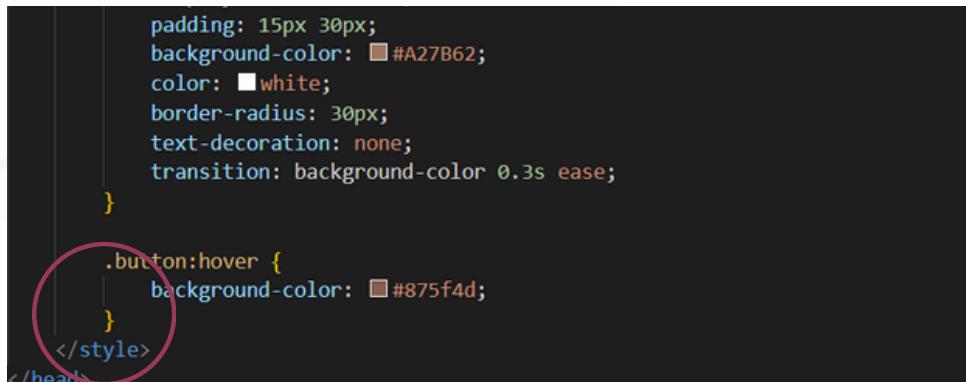
```
<div class="container"><!--name input-->

<form action="http://localhost/W/insert_volunteer.php" method="post" onsubmit="return validateForm()">
    <div class="input-group">
        <label for="nameInput">Name:</label>
        <textarea id="nameInput" name="name"></textarea>
    </div>
    <div class="input-group"><!--email input-->
        
        <label for="emailInput">University Email:</label>
        <textarea id="emailInput" name="email"></textarea>
    </div>
    <div class="input-group"><!--department Input-->
        
        <label for="departmentInput">Department:</label>
        <textarea id="departmentInput" name="department"></textarea>
    </div>
    <div class="input-group"><!--level Input-->
        
        <label for="levelInput">Academic Level:</label>
        <textarea id="levelInput" name="academic_level"></textarea>
    </div>
    <div class="input-group"><!--points Input-->
        
        <label for="pointsInput">Points:</label>
        <textarea id="pointsInput" name="points"></textarea>
    </div>
    <div class="input-group"><!--volunteer Input-->
        
        <label for="volunteeringInput">Type of Volunteering:</label>
        <textarea id="volunteeringInput" name="volunteering_type"></textarea>
    </div>
    <button class="custom-link" type="submit" >Volunteering</button>
</form>

```

1. Bad Smell: No Comments

- Description: The code has no comments .
- Refactoring: Add comments to make it easier for the code developer to modify in the future.



```

padding: 15px 30px;
background-color: #A27B62;
color: white;
border-radius: 30px;
text-decoration: none;
transition: background-color 0.3s ease;
}

.button:hover {
    background-color: #875f4d;
}
</style>
</head>
```

2. Bad Smell: Mixing HTML ,CSS in Same File

- Description: HTML and CSS code are merged into one file, resulting in reduced maintainability and readability.
- Refactoring: Separate HTML ,CSS code into distinct files