UNIVERSITI TEKNOLOGI MALAYSIA

OBJECT ORIENTED PROGRAMMING (SECJ2154)

SEMESTER 2 2023/2024

GROUP PROJECT

VOLUNTEER OPPORTUNITY SYSTEM

OMAR MOHAMMED REZK EBID ELASHRY (A22EC0026)

HASAN HATEM HASAN (A22EC0013)

AHMED AMIR MOUSTAFA (A22EC0007)

HAMDAN SALEH OMAR ALMOHAMADI(A22EC4027)

2 SCSR

SECTION 04

Lecturer:

MADAM LIZAWATI MI YUSUF

20 JUNE 2024

The work in this page has been done by: **Hamdan Saleh Omar** *Date:* **6/20/2024**

SECTION A: PROJECT DESCRIPTION

Synopsis:

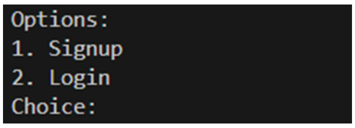
The Volunteer Opportunity System is a system which aims to link volunteers to several volunteering opportunities that are offered by the organizers. This system enables the organizers to put up short and long term positions and the volunteers can search and join opportunities and also get a response on the positions they applied for.

How to use the system:

Signup/Login:

Organizer: first the organizer needs to sign up with his personal information which are name, email, password, organization name. After that he can login with the information he used to sign up.

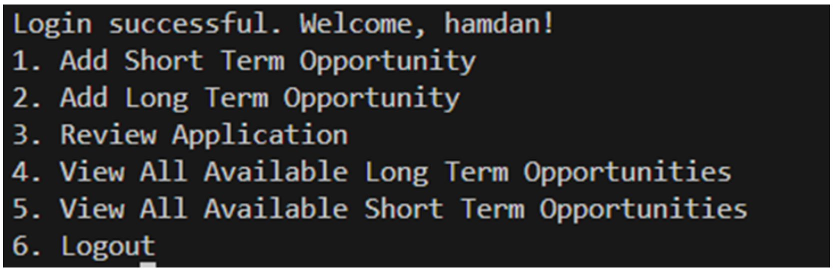
Volunteer: first the volunteer needs to sign up with his personal information which are name, email, password, skills and availability times. After that he can login with the information he used to sign up.



```
Options:
1. Signup
2. Login
Choice:
```

Functionalities for organizers:

- **Add short term opportunity:** organizers can create and add short term opportunities.
- **Add long term opportunity:** organizers can create and add long term opportunities.
- **Review applications:** organizers can evaluate volunteer applications and provide feedback.
- **View all available long term opportunities:** organizers can see all available long term opportunities.
- **View all available short term opportunities:** organizers can see all available short term opportunities.
- **Logout:** so the user can log out of the account he logged in with.



```
Login successful. Welcome, hamdan!
1. Add Short Term Opportunity
2. Add Long Term Opportunity
3. Review Application
4. View All Available Long Term Opportunities
5. View All Available Short Term Opportunities
6. Logout
```

Functionalities for volunteers:

- **Join long term opportunity:** for volunteers to apply for available long term opportunities.

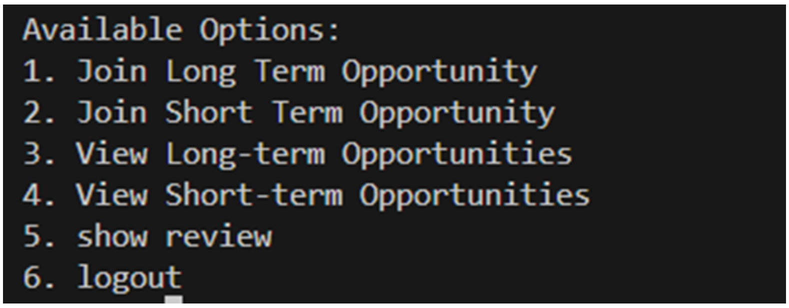
- **Join short term opportunity:** for volunteers to apply for available short term opportunities.

- **View long term opportunities:** for volunteers to see through the list of available long term opportunities.

- **View short term opportunities:** for volunteers to browse through the list of available short term opportunities.

- **Show review:** for volunteers to access feedbacks provided by organizers.

- **Logout:** so the user can log out of the account he logged in with.



```
Available Options:
1. Join Long Term Opportunity
2. Join Short Term Opportunity
3. View Long-term Opportunities
4. View Short-term Opportunities
5. show review
6. logout
```

Objective and Scope:

Objective:

The primary objective of the volunteer opportunity system is the process of matching volunteers with suitable opportunities based on their skills and availability, and to enable organizers to manage and review volunteer applications effectively.

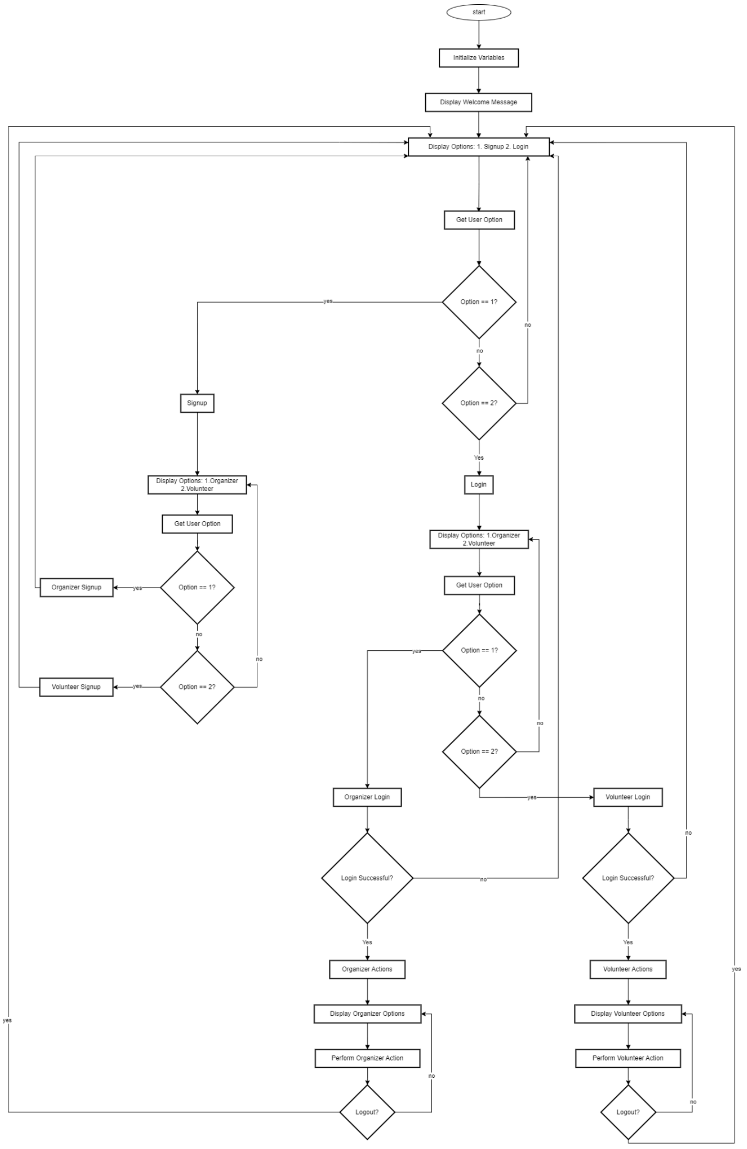
Scope:

· The system allows volunteers to sign up, log in, view, and apply for both long-term and short-term opportunities.

· Organizers can create, view, and manage volunteer opportunities, and provide feedback on volunteer applications.

· The system includes functionalities for handling user authentication, opportunity creation, application management, and giving feedback.

Workflow:



1. The system will ask the user to sign up if he is new or login if he has an account already, if the information he entered is wrong the system will take him to the beginning and ask to sign up or log in again.

2. After he log in to his account if he is an organizer he will be offered with actions to choose from by typing the number next to the action and the actions are explained previously you can check

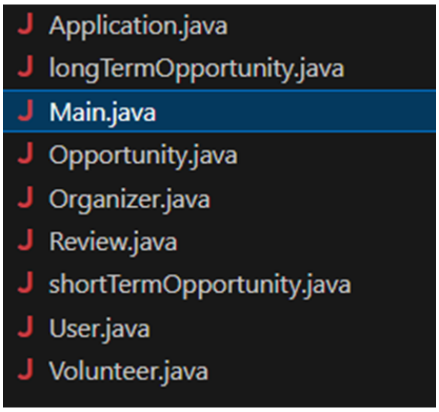
what each action do in the synopsis, but if he is a volunteer he will be offered with options to choose from by typing the number next to the action which are also explained in the synopsis previously also.

3. After that the user can choose whether he wants to continue or log out of his account, if he logged out the system will take to the beginning and ask him to sign up or login.

OO Concepts:

1. Classes and objects:

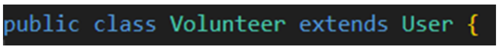
The system is a multifile program and uses classes such as Application, longTermOpportunity, Opportunity, Organizer, Review, shortTermOpportunity, User, and Volunteer.



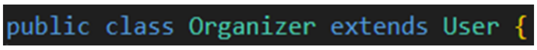
```
Application.java
longTermOpportunity.java
Main.java
Opportunity.java
Organizer.java
Review.java
shortTermOpportunity.java
User.java
Volunteer.java
```

2. Inheritance:

The Volunteer and Organizer classes are inherited from the User class, inheriting common properties and methods.

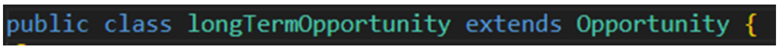


```
public class Volunteer extends User {
```

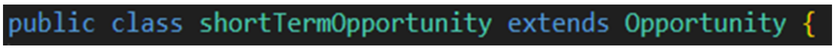


```
public class Organizer extends User {
```

The long Term Opportunity and short Term Opportunity classes are inherited from the Opportunity class.



```
public class longTermOpportunity extends Opportunity {
```

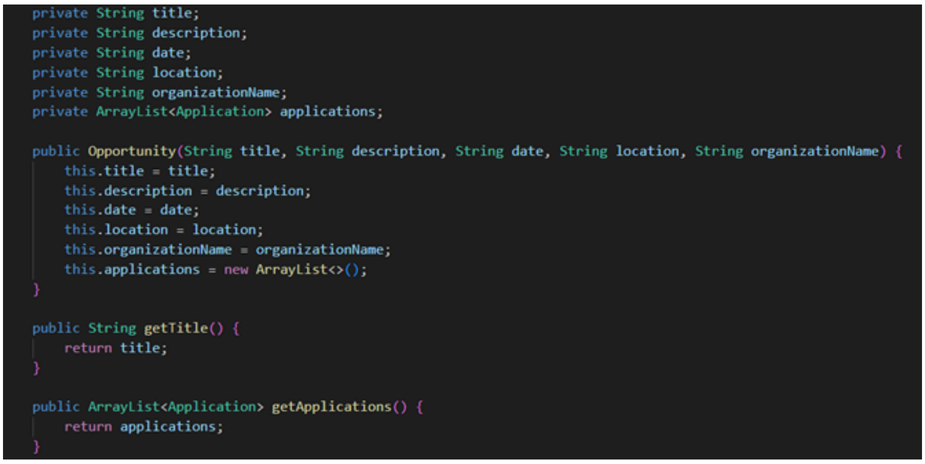


```
public class shortTermOpportunity extends Opportunity {
```

3. Encapsulation:

Each class has private attributes and provides public methods to access and modify these attributes, ensuring that the internal state of objects is protected.

Example from Opportunity class:

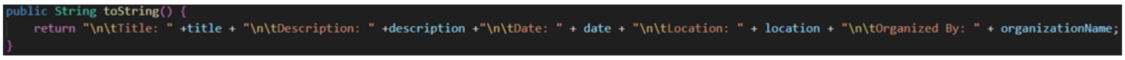


```
private String title;
private String description;
private String date;
private String location;
private String organizationName;
private ArrayList<Application> applications;

public Opportunity(String title, String description, String date, String location, String organizationName) {
    this.title = title;
    this.description = description;
    this.date = date;
    this.location = location;
    this.organizationName = organizationName;
    this.applications = new ArrayList<>();
}

public String getTitle() {
    return title;
}

public ArrayList<Application> getApplications() {
    return applications;
}
```

4. Polymorphism:

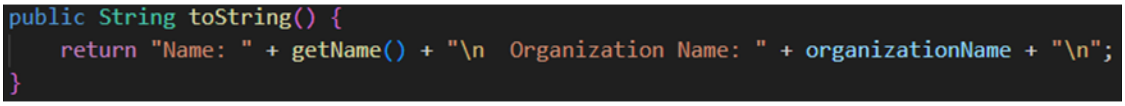
The system uses polymorphism using toString methods. Each class overrides the toString method to provide a string representation of its instances. This allows for a unified way to print object details while each class can customize its output format.

In opportunity class:



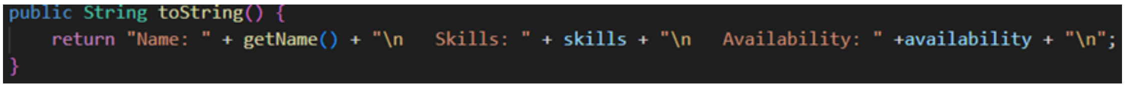
```
public String toString() {
    return "\n\tTitle: " +title + "\n\tDescription: " +description +"\n\tDate: " + date + "\n\tLocation: " + location + "\n\tOrganized By: " + organizationName;
}
```

In organizer class:



```
public String toString() {
    return "Name: " + getName() + "\n  Organization Name: " + organizationName + "\n";
}
```

In volunteer class:



```java
public String toString() {
    return "Name: " + getName() + "\n   Skills: " + skills + "\n   Availability: " +availability + "\n";
}
```

5. Abstraction:

The system abstracts common functionalities in base classes (User, Opportunity) and extends them in derived classes (Volunteer, Organizer, longTermOpportunity, shortTermOpportunity), providing a clear and modular structure.

6. Association:

We can find the association between the following classes:

a. Opportunity and longTermOpportunity and shortTermOpportunity:

opportunity is the super class while longTermOpportunity and shortTermOpportunity are the subclasses of it.

b. User and volunteer and organizer:

User is the super class while volunteer and organizer are the subclasses of it.

c. Volunteer and application:

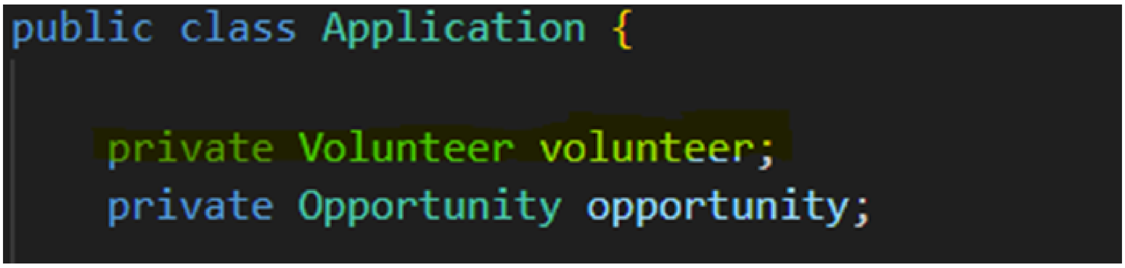
The association between these class is that the volunteer can make application to register for opportunities.

7. Aggregation:

We can find the aggregation between the following classes:

a. Volunteer and application:

The volunteer must have a application to register for an opportunity, and we can find the aggregation in the code here:



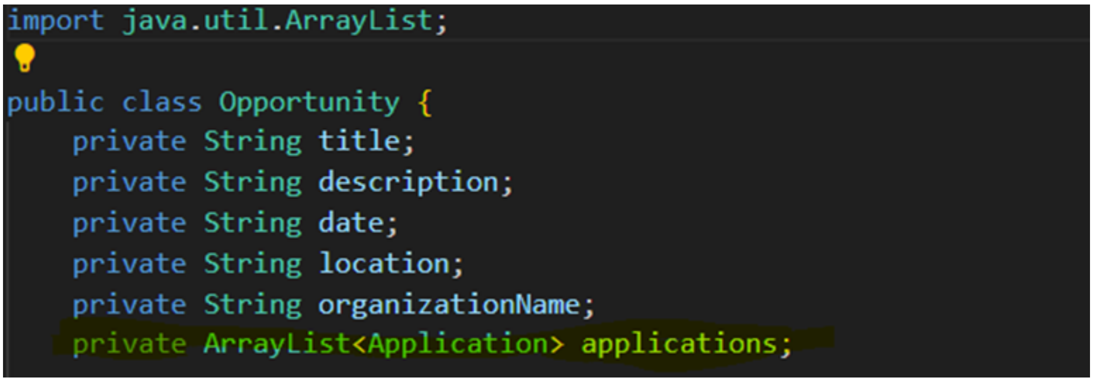
```java
public class Application {

    private Volunteer volunteer;
    private Opportunity opportunity;
```

8. Composition:

We can find the composition between the following classes:

a. Opportunity and application:

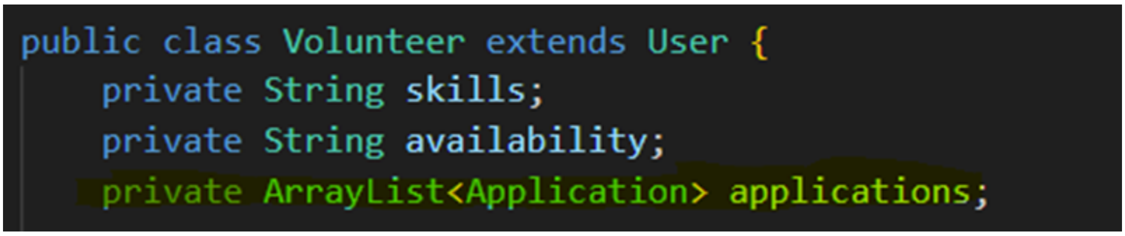
The class opportunity contains multiple applications that have been submitted by the volunteers, and we can find the composition in the code here:



```
import java.util.ArrayList;

public class Opportunity {
    private String title;
    private String description;
    private String date;
    private String location;
    private String organizationName;
    private ArrayList<Application> applications;
```

b. Application and volunteer:

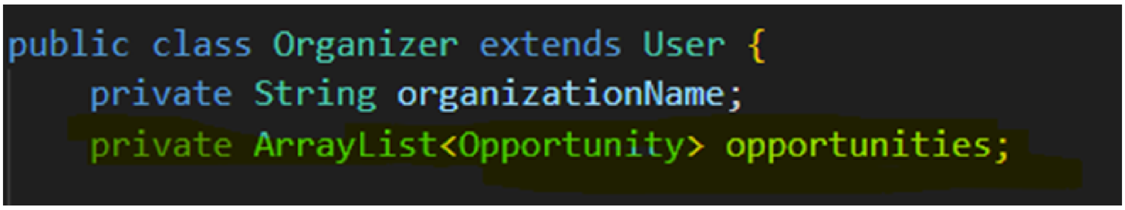
The class volunteer contains multiple application that have been made by volunteers to register for opportunities, and we can find the composition in the code here:



```
public class Volunteer extends User {
    private String skills;
    private String availability;
    private ArrayList<Application> applications;
```

c. Opportunity and organizer:

The class organizer contains multiple opportunities that have been made by them, and we can find it in the code here:

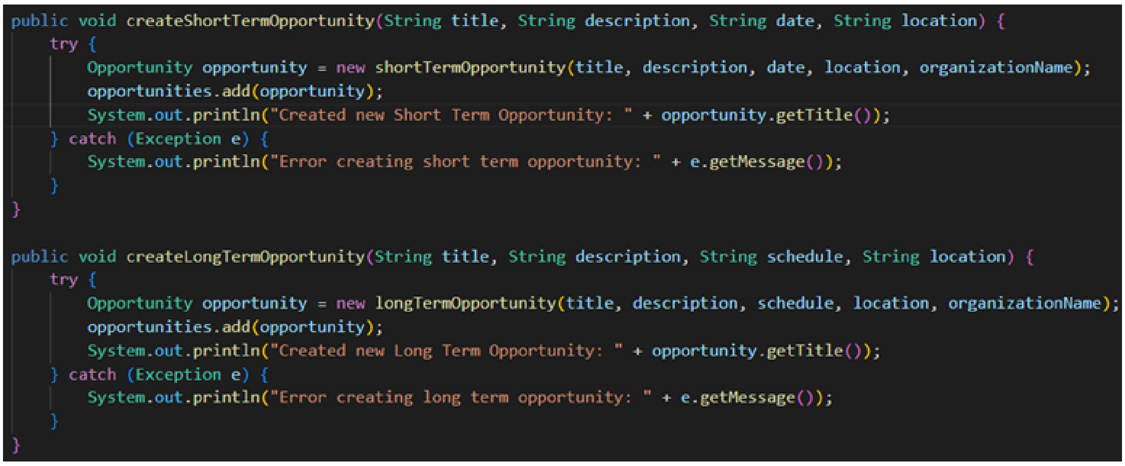


```
public class Organizer extends User {
    private String organizationName;
    private ArrayList<Opportunity> opportunities;
```

9. Exception handling:

Exception handling is used to manage errors gracefully and ensure the application runs smoothly without crashing.

- And we can find it in the organizer class when creating the short and long term opportunities here:



```
public void createShortTermOpportunity(String title, String description, String date, String location) {
    try {
        Opportunity opportunity = new shortTermOpportunity(title, description, date, location, organizationName);
        opportunities.add(opportunity);
        System.out.println("Created new Short Term Opportunity: " + opportunity.getTitle());
    } catch (Exception e) {
        System.out.println("Error creating short term opportunity: " + e.getMessage());
    }
}

public void createLongTermOpportunity(String title, String description, String schedule, String location) {
    try {
        Opportunity opportunity = new longTermOpportunity(title, description, schedule, location, organizationName);
        opportunities.add(opportunity);
        System.out.println("Created new Long Term Opportunity: " + opportunity.getTitle());
    } catch (Exception e) {
        System.out.println("Error creating long term opportunity: " + e.getMessage());
    }
}
```

What these 2 exception handling blocks will do:

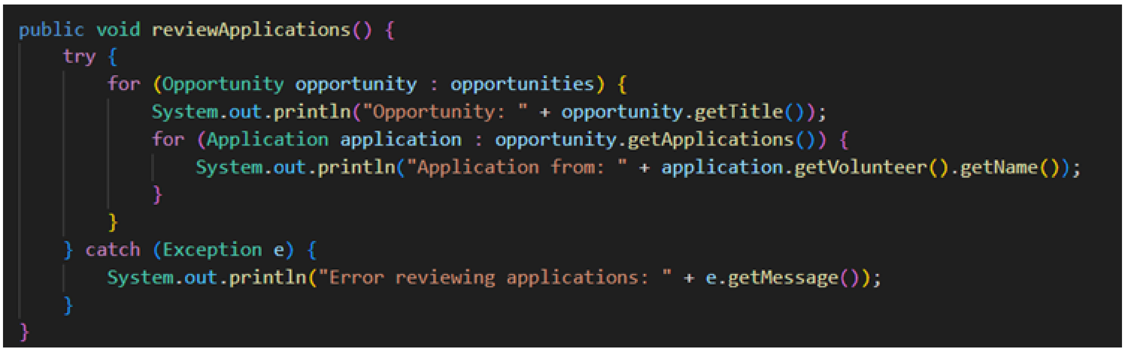
Try Block:

Attempts to create a new short or long term Opportunity object with the given parameters, and then adds the newly created opportunity to the opportunities list, after that it Prints a success message with the title of the created opportunity.

Catch Block:

If the parameters are invalid the block catches the exception, and then Prints an error message indicating that there was an error creating the short or long opportunity, along with the specific exception message.

- Also in the class we can find another exception handling block for reviewing the applications of the volunteers:



```
public void reviewApplications() {
    try {
        for (Opportunity opportunity : opportunities) {
            System.out.println("Opportunity: " + opportunity.getTitle());
            for (Application application : opportunity.getApplications()) {
                System.out.println("Application from: " + application.getVolunteer().getName());
            }
        }
    } catch (Exception e) {
        System.out.println("Error reviewing applications: " + e.getMessage());
    }
}
```

What This Block Does:

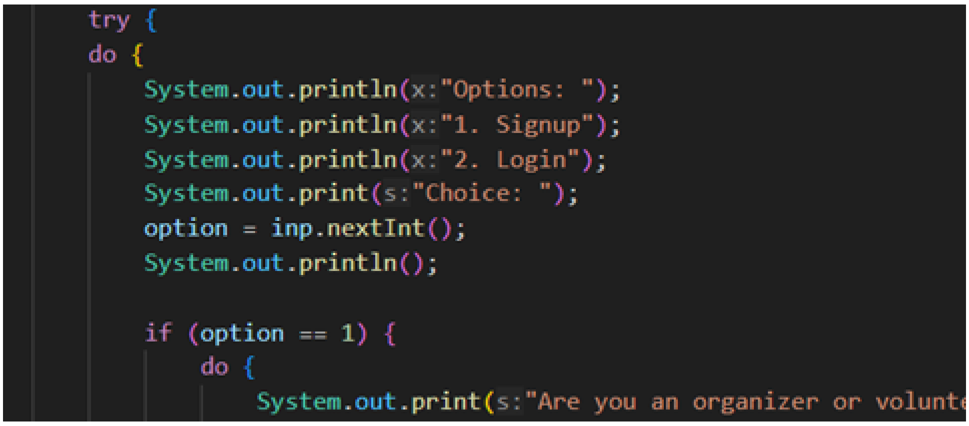
Try Block:

This block Iterates each Opportunity in the opportunities list, and for each Opportunity, it prints the title of the opportunity, and also iterates over each Application associated with the current Opportunity, and then prints the name of the volunteer who applied for the opportunity.

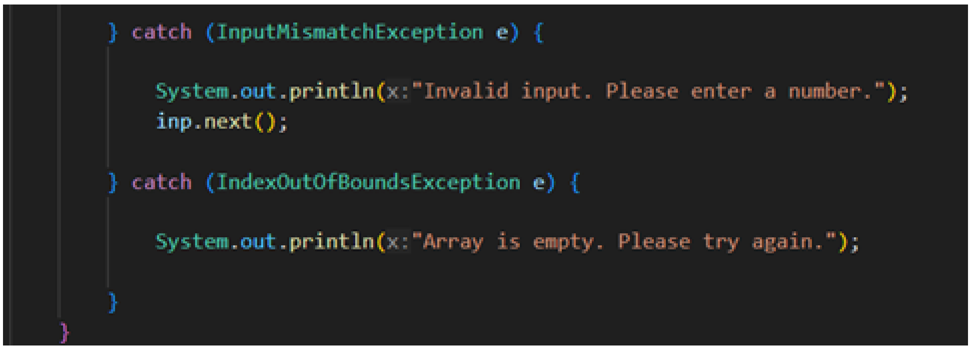
Catch Block:

If the opportunities list is empty, the catch block catches the exception, and then prints an error message indicating that there was an error reviewing the applications, along with the exception message.

- And also in the main function the exception handling is implemented as we can see in the pictures below:



```
try {
do {
    System.out.println(x:"Options: ");
    System.out.println(x:"1. Signup");
    System.out.println(x:"2. Login");
    System.out.print(s:"Choice: ");
    option = inp.nextInt();
    System.out.println();

    if (option == 1) {
        do {
            System.out.print(s:"Are you an organizer or volunte
```



```
    } catch (InputMismatchException e) {

        System.out.println(x:"Invalid input. Please enter a number.");
        inp.next();

    } catch (IndexOutOfBoundsException e) {

        System.out.println(x:"Array is empty. Please try again.");

    }
}
```

What this block will do:

Try Block:

In this block the code will displays the options for signup or login, and will handle the user input for signing up as an organizer or volunteer, and also eases the login process for organizers and volunteers, and also allows organizers to create short term or long term opportunities, and enables organizers to review applications, also allows volunteers to join opportunities and view reviews.

Catch Block:

In this block there is 2 exceptions to catch:

- **InputMismatchException**:

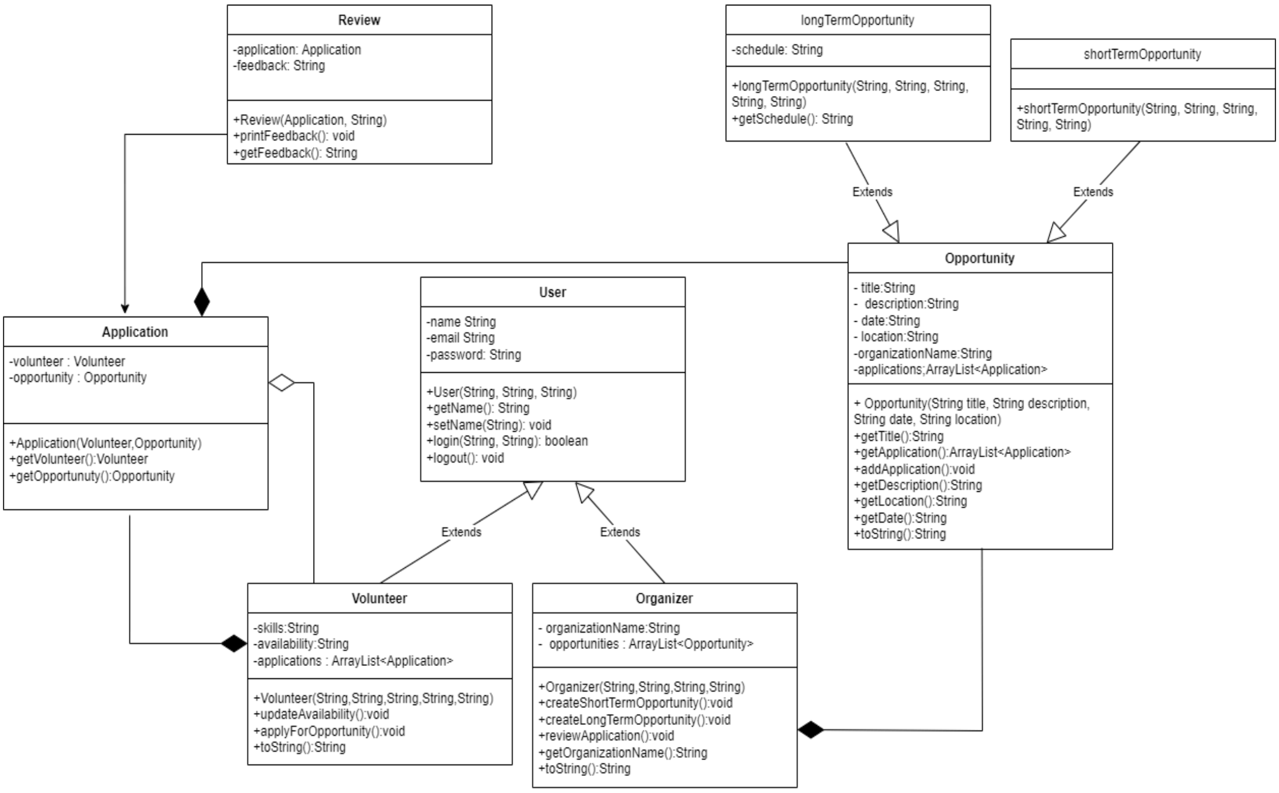
  In this one it will catch the exceptions where the input type does not match the expected type of the inputs and prints an error message: "Invalid input. Please enter a number.", and after that it clears the invalid input from the scanner to prevent an infinite loop of exceptions.

- **IndexOutOfBoundsException**:

  In this one it will catch the exceptions where an invalid index is accessed the empty list, and then will print an error message: "Array is empty. Please try again."

The work in this page has been done by: **HASAN HATEM HASAN** *Date:* **6/20/2024**

SECTION B: CLASS DIAGRAMS


Review
-application: Application
-feedback: String
+Review(Application, String)
+printFeedback(): void
+getFeedback(): String
longTermOpportunity
-schedule: String
+longTermOpportunity(String, String, String, String, String)
+getSchedule(): String
shortTermOpportunity
+shortTermOpportunity(String, String, String, String, String)
Extends
Extends
Opportunity
- title:String
- description:String
- date:String
- location:String
-organizationName:String
-applications;ArrayList<Application>
+ Opportunity(String title, String description, String date, String location)
+getTitle():String
+getApplication():ArrayList<Application>
+addApplication():void
+getDescription():String
+getLocation():String
+getDate():String
+toString():String
Application
-volunteer : Volunteer
-opportunity : Opportunity
+Application(Volunteer,Opportunity)
+getVolunteer():Volunteer
+getOpportunuty():Opportunity
User
-name String
-email String
-password: String
+User(String, String, String)
+getName(): String
+setName(String): void
+login(String, String): boolean
+logout(): void
Extends
Extends
Volunteer
-skills:String
-availability:String
-applications : ArrayList<Application>
+Volunteer(String,String,String,String,String)
+updateAvailability():void
+applyForOpportunity():void
+toString():String
Organizer
- organizationName:String
- opportunities : ArrayList<Opportunity>
+Organizer(String,String,String,String)
+createShortTermOpportunity():void
+createLongTermOpportunity():void
+reviewApplication():void
+getOrganizationName():String
+toString():String

Class User:

Attributes	Description
name	name of user
email	email of user
password	password for the user to login
Methods	**Description**
User	initialize the values for the user Attributes
getName	return user name
setName	sets a name for the user
login	verify if the user email and password is valid.
logout	log the user out of the system.

The work in this page has been done by: **HASAN HATEM HASAN** *Date:* **6/20/2024**

Class Organizer:

Attributes	**Description**
organizationName	name of the Organization
opportunities	Array that save the list of Opportunities
Methods	**Description**
Organizer	initialize the values for the Organizer Attributes.
createShortTermOpportunity	create a new short term opportunity.
createLongTermOpportunity	create a new long term opportunity.
reviewApplications	view the available Applications.
getOrganizationName	return organizationName.
toString	return organizer name and organization name.

Class Volunteer:

Attributes	**Description**
skills	volunteers skills
availability	when is the volunteer free
applications	list of Applications to apply for opportunity.
Methods	**Description**
Volunteer	initialize the values for the Volunteer Attributes.
applyForOpportunity	allow volunteers to apply for an opportunity.
updateAvailability	update the user Availability
toString	return organizer name and organization name.

Class longTermOpportunity:

Attributes	**Description**
schedule	the schedule for the long term opportunity.
Methods	**Description**
longTermOpportunity	initialize the values for the **longTermOpportunity** Attributes.
getSchedule	return the schedule.

The work in this page has been done by: **Ahmed Amir moustafa** *Date:* **6/20/2024**

Class shortTermOpportunity:

Attributes	Description
Methods	**Description**
shortTermOpportunity	initialize the values for the shortTermOpportunity Attributes.

Class Application:

Attributes	Description
volunteer	object of volunteer used in application
opportunity	object of opportunity used in opportunity
Methods	**Description**
Application	initialize the values for the **Application** Attributes.
getVolunteer	return the volunteer.
getOpportunity	return opportunity.

Class Review:

Attributes	Description
application	object of application used to attach the feedback to applications.
feedback	store the feedback on the application.
Methods	**Description**
Review	initialize the values for the **Review** Attributes.
printFeedback	print the application volunteer name and the feedback.
getFeedback	return the feedback.

The work in this page has been done by: **Omar Mohammed Rezk Ebid Elashry** *Date:* **20/June/2024**

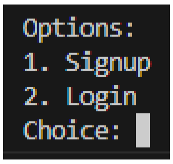
SECTION C: SOURCE CODE AND USER MANUAL

In this section you need to provide the source code of your system. The source code serves as a way to present the object oriented concepts used in your system. You are required to provide the user manual of your system. The user manual describe how to use the system and the flow of your system, i.e show the example of input and the expected output.

Source Code:

https://drive.google.com/drive/folders/1mBoSv1dsRvoJQsM1m0arfva0B41kn3he?usp=sharing

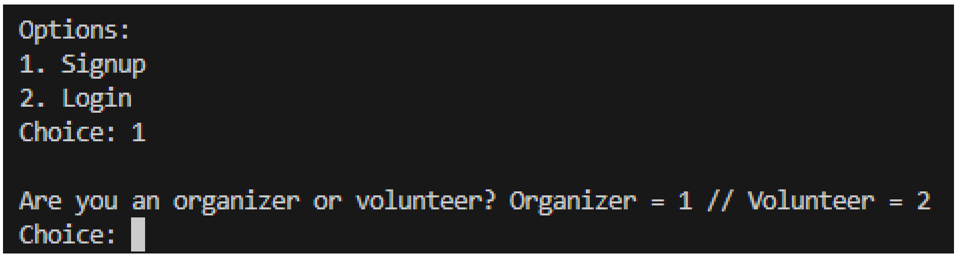
Main Screen:



```
Options:
1. Signup
2. Login
Choice: 
```

So this is the Main Screen that appears. The user is either going to Login or Signup.

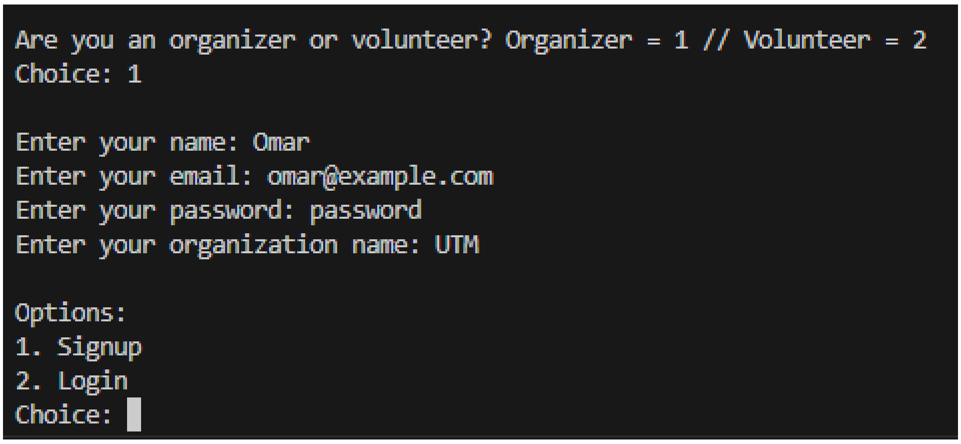
Screen 1a:



```
Options:
1. Signup
2. Login
Choice: 1

Are you an organizer or volunteer? Organizer = 1 // Volunteer = 2
Choice: 
```

User gets to choose if they are either an organizer or a volunteer and proceed to signup.

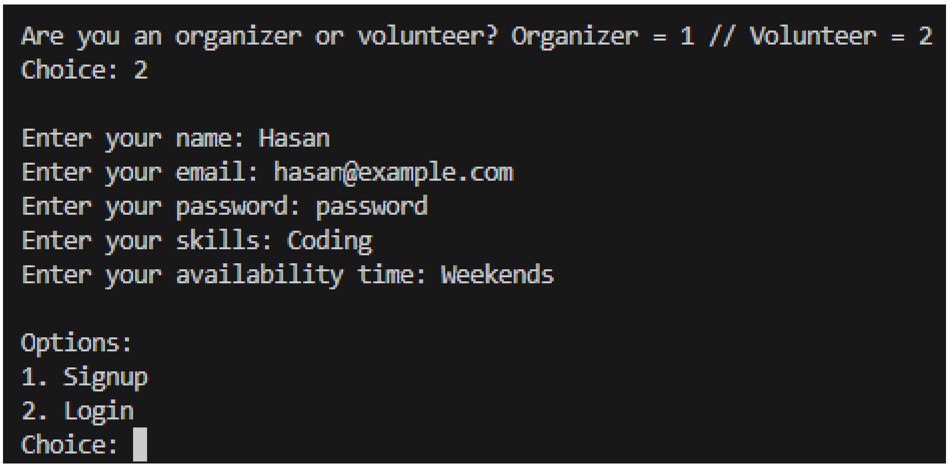
Screen 1aa:



```
Are you an organizer or volunteer? Organizer = 1 // Volunteer = 2
Choice: 1

Enter your name: Omar
Enter your email: omar@example.com
Enter your password: password
Enter your organization name: UTM

Options:
1. Signup
2. Login
Choice: 
```

User signs up as an organizer and is required to enter their name, email, password and organization name and then they are returned to the main screen.

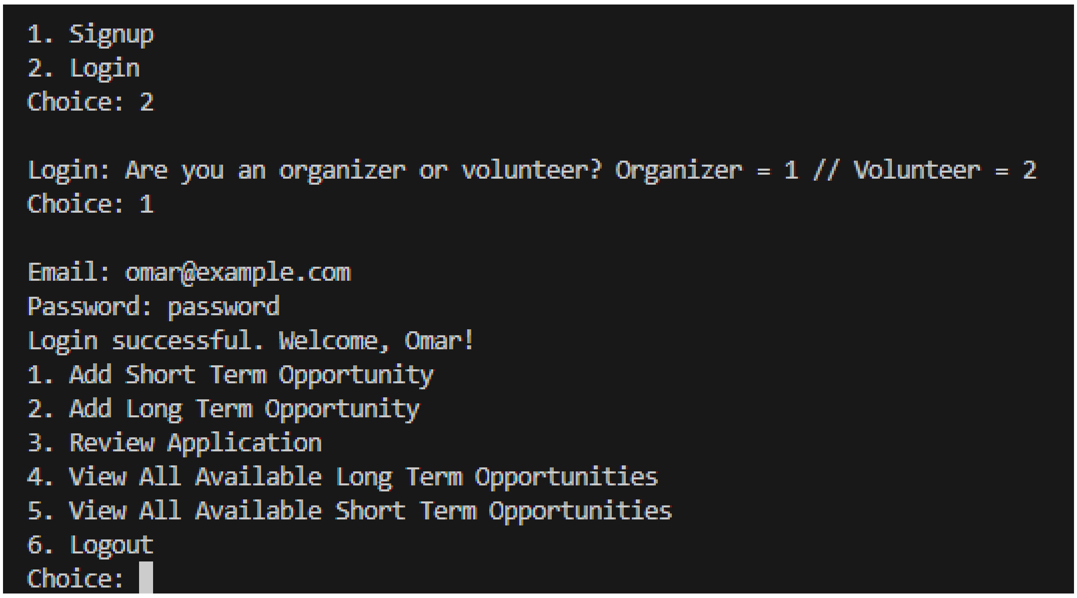
Screen 1ab:



```
Are you an organizer or volunteer? Organizer = 1 // Volunteer = 2
Choice: 2

Enter your name: Hasan
Enter your email: hasan@example.com
Enter your password: password
Enter your skills: Coding
Enter your availability time: Weekends

Options:
1. Signup
2. Login
Choice: 
```

User signs up as a volunteer and is required to enter their name, email, password, skills and availability time and then they are returned to the main screen.

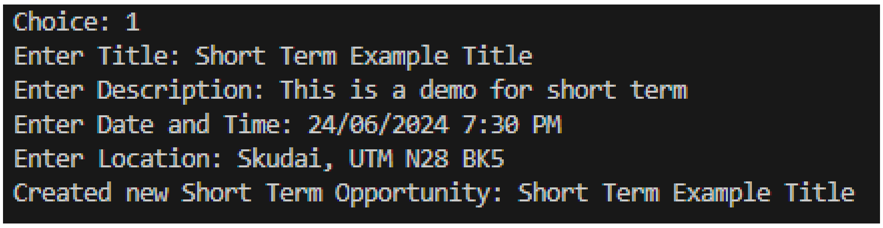
Screen 1b:



```
1. Signup
2. Login
Choice: 2

Login: Are you an organizer or volunteer? Organizer = 1 // Volunteer = 2
Choice: 1

Email: omar@example.com
Password: password
Login successful. Welcome, Omar!
1. Add Short Term Opportunity
2. Add Long Term Opportunity
3. Review Application
4. View All Available Long Term Opportunities
5. View All Available Short Term Opportunities
6. Logout
Choice: 
```

Organizer proceeds to login after signing up and has a list of options to choose from.

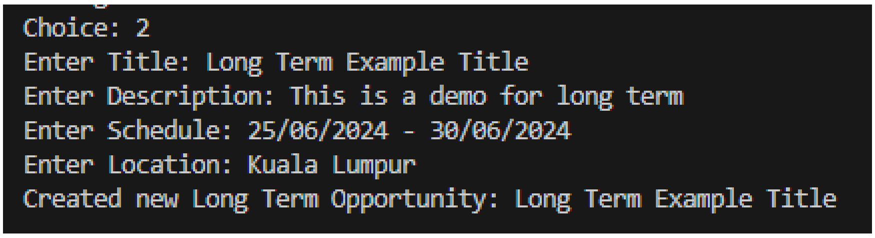
Screen 1baa:



```
Choice: 1
Enter Title: Short Term Example Title
Enter Description: This is a demo for short term
Enter Date and Time: 24/06/2024 7:30 PM
Enter Location: Skudai, UTM N28 BK5
Created new Short Term Opportunity: Short Term Example Title
```

Organizer chooses to add a short term opportunity and is required to a title, description, date and time and the location.

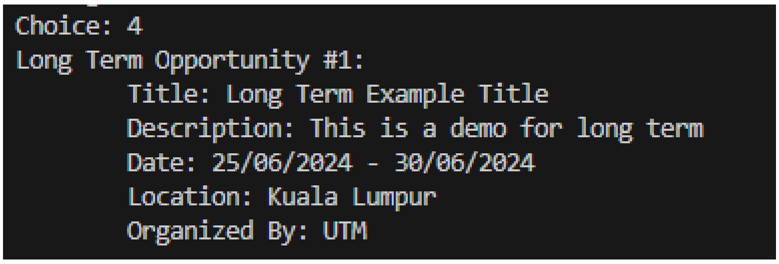
Screen 1bab:



```
Choice: 2
Enter Title: Long Term Example Title
Enter Description: This is a demo for long term
Enter Schedule: 25/06/2024 - 30/06/2024
Enter Location: Kuala Lumpur
Created new Long Term Opportunity: Long Term Example Title
```

Organizer chooses to add a short term opportunity and is required to a title, description, schedule and the location.

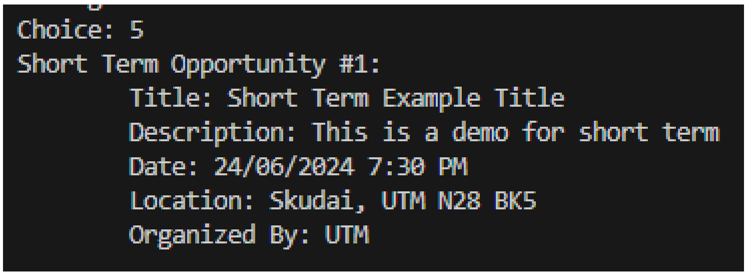
Screen 1bac:



```
Choice: 4
Long Term Opportunity #1:
        Title: Long Term Example Title
        Description: This is a demo for long term
        Date: 25/06/2024 - 30/06/2024
        Location: Kuala Lumpur
        Organized By: UTM
```

Organizer gets to see all the Long Term Opportunities available with the title, description, date, location and who it is organized by.

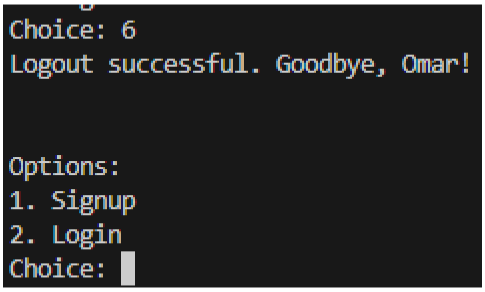
Screen 1bad:



```
Choice: 5
Short Term Opportunity #1:
        Title: Short Term Example Title
        Description: This is a demo for short term
        Date: 24/06/2024 7:30 PM
        Location: Skudai, UTM N28 BK5
        Organized By: UTM
```

Organizer gets to see all the Short Term Opportunities available with the title, description, date, location and who it is organized by.

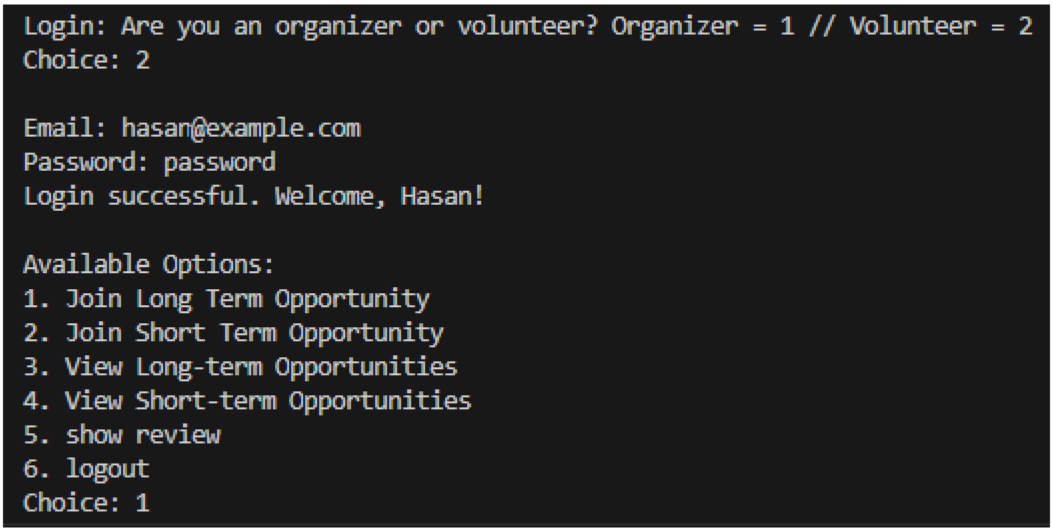
Screen 1bae:



```
Choice: 6
Logout successful. Goodbye, Omar!


Options:
1. Signup
2. Login
Choice: 
```

Organizer logs out and is returned to the main screen

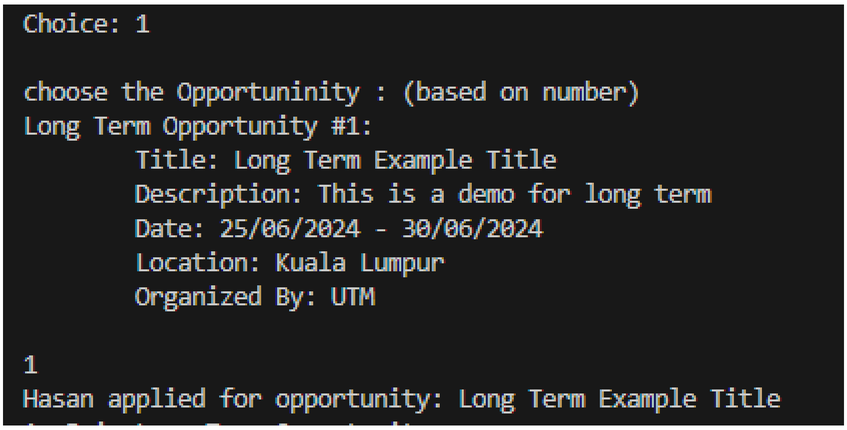
Screen 1bba:



```
Login: Are you an organizer or volunteer? Organizer = 1 // Volunteer = 2
Choice: 2

Email: hasan@example.com
Password: password
Login successful. Welcome, Hasan!

Available Options:
1. Join Long Term Opportunity
2. Join Short Term Opportunity
3. View Long-term Opportunities
4. View Short-term Opportunities
5. show review
6. logout
Choice: 1
```

Volunteer logs in and is giving several options to choose from

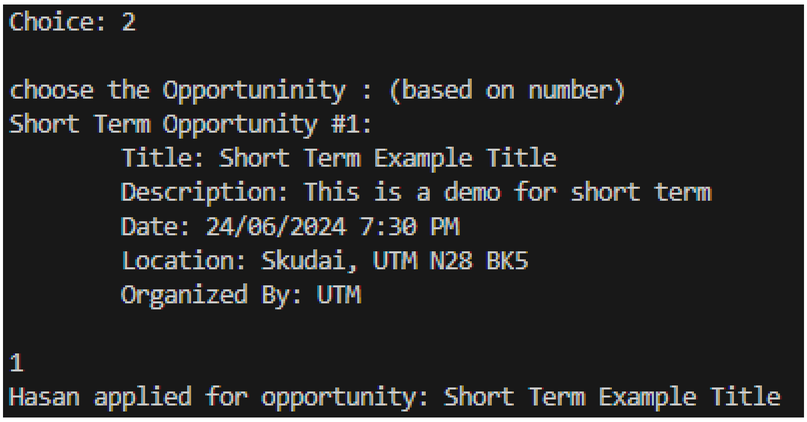
Screen 1bbb:



```
Choice: 1

choose the Opportuninity : (based on number)
Long Term Opportunity #1:
        Title: Long Term Example Title
        Description: This is a demo for long term
        Date: 25/06/2024 - 30/06/2024
        Location: Kuala Lumpur
        Organized By: UTM

1
Hasan applied for opportunity: Long Term Example Title
```

Volunteer decides to join long term opportunity and is given list of available long term opportunities to join.

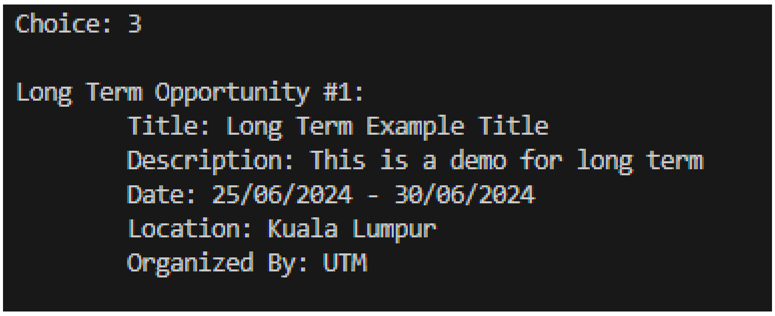
Screen 1bbc:



```
Choice: 2

choose the Opportuninity : (based on number)
Short Term Opportunity #1:
        Title: Short Term Example Title
        Description: This is a demo for short term
        Date: 24/06/2024 7:30 PM
        Location: Skudai, UTM N28 BK5
        Organized By: UTM

1
Hasan applied for opportunity: Short Term Example Title
```

Volunteer decides to join short term opportunity and is given list of available short term opportunities to join.

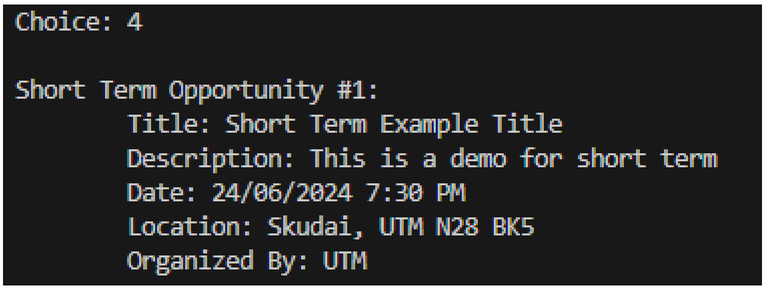
Screen 1bbd:



```
Choice: 3

Long Term Opportunity #1:
        Title: Long Term Example Title
        Description: This is a demo for long term
        Date: 25/06/2024 - 30/06/2024
        Location: Kuala Lumpur
        Organized By: UTM
```

Volunteer can see all available long term opportunities

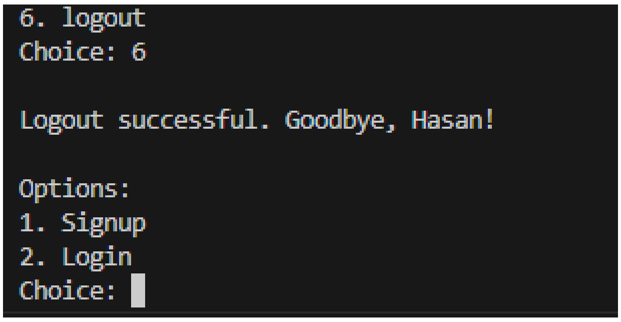
Screen 1bbe:



```
Choice: 4

Short Term Opportunity #1:
        Title: Short Term Example Title
        Description: This is a demo for short term
        Date: 24/06/2024 7:30 PM
        Location: Skudai, UTM N28 BK5
        Organized By: UTM
```

Volunteer can see all available short term opportunities

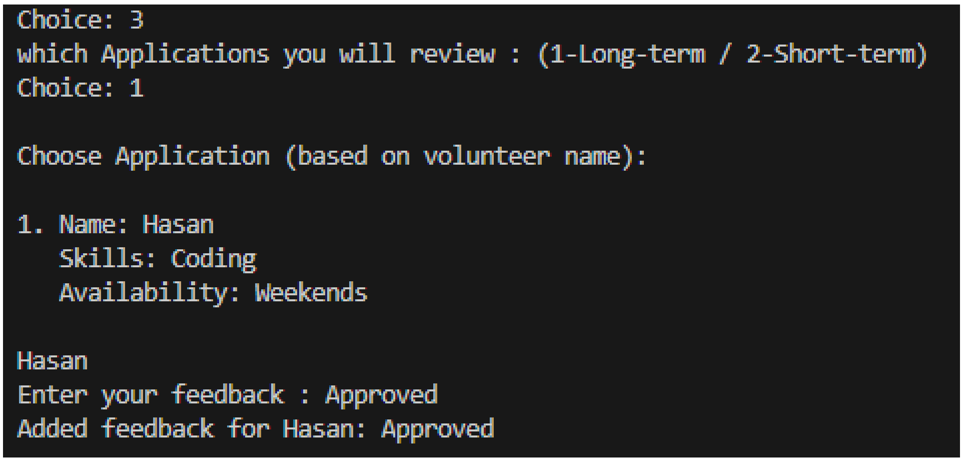
Screen 1bbf:



```
6. logout
Choice: 6

Logout successful. Goodbye, Hasan!

Options:
1. Signup
2. Login
Choice: 
```

Volunteer logs out and returned to main screen

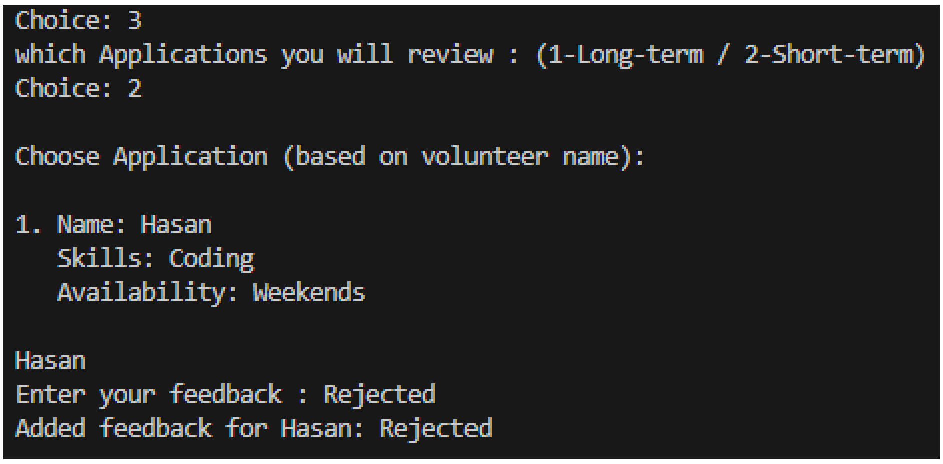
Screen 1baf:



```
Choice: 3
which Applications you will review : (1-Long-term / 2-Short-term)
Choice: 1

Choose Application (based on volunteer name):

1. Name: Hasan
   Skills: Coding
   Availability: Weekends

Hasan
Enter your feedback : Approved
Added feedback for Hasan: Approved
```

Organizer checks the available applications for the long term and gives feedback.

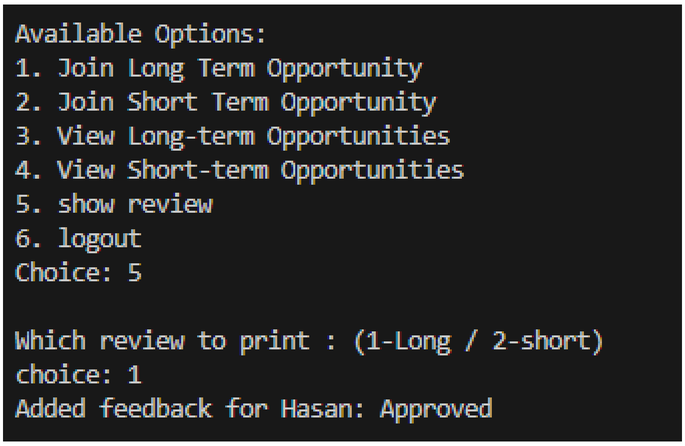
Screen 1bag:



```
Choice: 3
which Applications you will review : (1-Long-term / 2-Short-term)
Choice: 2

Choose Application (based on volunteer name):

1. Name: Hasan
   Skills: Coding
   Availability: Weekends

Hasan
Enter your feedback : Rejected
Added feedback for Hasan: Rejected
```

Organizer checks the available applications for the short term and gives feedback.

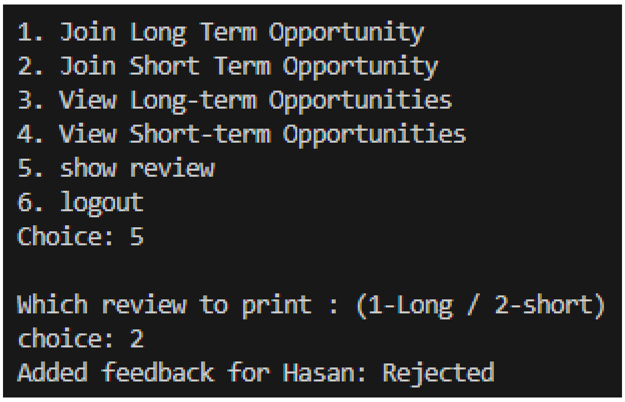
Screen 1bbg:



```
Available Options:
1. Join Long Term Opportunity
2. Join Short Term Opportunity
3. View Long-term Opportunities
4. View Short-term Opportunities
5. show review
6. logout
Choice: 5

Which review to print : (1-Long / 2-short)
choice: 1
Added feedback for Hasan: Approved
```

Volunteer checks for the review of their application

Screen 1bbh:



```
1. Join Long Term Opportunity
2. Join Short Term Opportunity
3. View Long-term Opportunities
4. View Short-term Opportunities
5. show review
6. logout
Choice: 5

Which review to print : (1-Long / 2-short)
choice: 2
Added feedback for Hasan: Rejected
```

Volunteer checks for the review of their application