UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SECJ2154

OBJECT ORIENTED PROGRAMMING

Systematic Squad

PROJECT

Bluestone Food Support

LECTURER	SECTION
Dr. Nies Hui Wen	10

Prepared by:

NAME	MATRIC
Toya Lazmin Khan	A20EC0284
Adnan Shafi	A20EC0255
Fayruz Tahzeeb Rahman	A20EC4019
GM Shaheen Shah Shimon	A20EC0266

1.0 Introduction

Near the end of 2019, the world faced a huge outbreak of a respiratory disease named COVID-19. It was later declared as the pandemic and is still an ongoing global pandemic. COVID-19 took the lives of many loved ones and urged the world to go into a strict lockdown. This resulted in a huge loss of economy and many small businesses as well as well established businesses ended up shutting down temporarily or completely. Not only that, every place involving social gathering was shut down until further notice. This made a huge chunk of the population lose their jobs and medical bills stripped money out of them due to covid-19. People having a standard lifestyle ended up living in poverty.

During the first of April 2022, Malaysia declared COVID-19 a an endemic disease and took steps to redo life the way it was. Unfortunately, even though a normal lifestyle commenced , there were many people who did not get their jobs back and are still struggling to make both ends meet.

The system we intend on making is targeted towards those people who are going through a hard time due to food insufficiency. Food is a prime need in a person's life and without it people cannot survive. Even though we cannot help the needy from other aspects, we are willing to take the initiative to reduce a person's hunger through our system. Our system is also focused towards the people who are willing to help people through the extra money they have. Moreover, people who have bought food in abundance and cannot finish it in time have the opportunity to donate it instead of letting it go to waste. The last feature of the system is to be able to volunteer in our noble act, for example: helping to deliver parcels from the donor to the seeker. This system is oriented towards the kind of process where a donor donates and the people in need receive their donation. Initially, our system will be based in Malaysia and we will be working with other organizations as well. The system is primarily implemented through a mobile application where all the aforementioned features will be available. Our system will be connected to multiple food supply chains and catering services for the continuous supply of food and to keep in stock. Our application will contain various features, the most prominent ones of which are 'Order food', where people or organizations will be able to order food and either pick up from locations or get them delivered,
'Donate', where people or restaurants can either donate money or leftover food items, and 'Volunteering', where people can apply to be volunteers.

Even though the system will be based on the covid-19 affected people, it is free for any person in need to be able to seek help from the system.

2.0 Project Planning

2.1 Relation between classes

The volunteer, donor and receiver are associated with the organizer by aggregation. The object of the volunteer, donor and receiver are declared in the organizer class.
The class people and restaurant are the sub-classes of the class donor. Donor is the parent class, and the people and restaurant are the child classes. The relationship of donor with people and restaurant is called inheritance.

2.2 UML Diagram

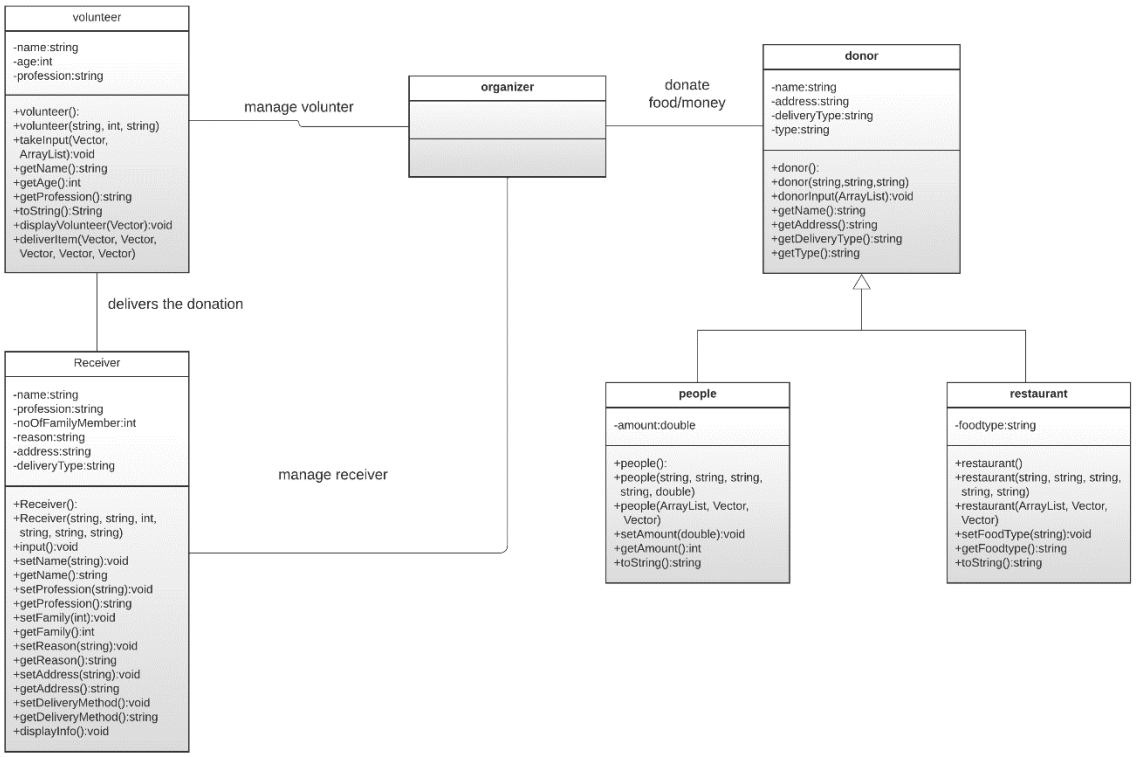

Fig: UML Class Diagram

2.1 Classes

1.volunteer
Attributes: name, age, profession
Methods: takeInput(), getName(), getAge(), getProfession(), toString(), displayVolunteer(), deliverItem() ;

2.receiver
Attributes: name, profession, noOfFamilyMember, reason,address , deliveryType
Methods:input(), getName(), getProfession(),getFamily(), getReason(), getAddress(), getDeliveryMethod(),setName(),setProfession(),setFamily(),setReason(),setAddress(),set DeliveeryType(),toString(),displayInfo()

3.donor
Attributes: name, address, deliveryType, type
Methods: donarInput(), getName(), getType(), getDeliveryType()

4.people
Attributes*:*amount
Methods:setAmount(), getAmount(),toString()

5.restaurant
Attributes:foodType
Methods:setFoodType(), getFoodType(),toString()

3.0 Project Design

For the project design, we used many java concepts like array, arrayList, vector , scanner etc.

Our project's main class, organizer, contains the public main method. This class will be used to execute our code. In order to conduct all of the functions, the objects of the volunteer, receiver, and donor classes are declared inside the organizor class.

For the class volunteer, we have included a method previously mentioned in the UML diagram takeInput(), here we use the Scanner class to get input from the user. Through the scanner object the user inputs the volunteer name, age, profession. We use vectors to store the inputs taken from the user. The attributes of the volunteer are first stored in one vector and then an ArrayList is created to store the vectors in it. Other than that, another ArrayList is formed where only the volunteer's name is stored. When the menu is displayed and the volunteer information is asked, the user can type in the information. When all the typed volunteer information is asked to print, the system prints are the typed information.

For the donor class, we use the method donorInput() to take input from the user. The same concept is used in the donor class as well. Inputs are stored in the scanner object and then the donor information is saved in a vector and then stored as a group of vector objects in ArrayList.Donor name is separately stored in an ArrayList. When the menu is displayed and the donor information is asked, the user can type in the information. When all the typed donor information is asked to print, the system prints are the typed information. The donorInput()method is called in 2 of donors inherited class People and Restaurant.

For class people and restaurants, inheritance is used to form a relationship between the donor class and these 2 classes. Where the donor is the parent class and people and restaurants are the children classes. People and restaurants specify which group of donors are they i.e they are either people or food given from restaurants.

To take input from the user, we use the Scanner class inside the input () method for the receiver class. It will also assign the input to the class variable. Furthermore, the getter method is used to return the value allocated to the instant variable. Because only one user can apply for donation at a time, we didn't use an arrayList or vector. We then use the display () method to show the user's information as well as the delivery type.

4.0 Project Outcome

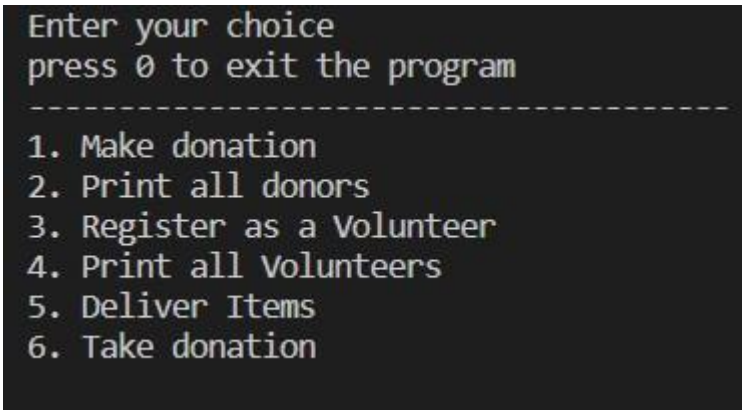

Fig: menu of the code .

When the option to make donation is selected by typing 1 as the user’s choice, the interface shows if the donation has to be made by people or from a restaurant .If people is selected , the interface will ask the people for their name , address, delivery type which will then be saved in the system. If the option restaurant is rather selected, the same information is asked from them.

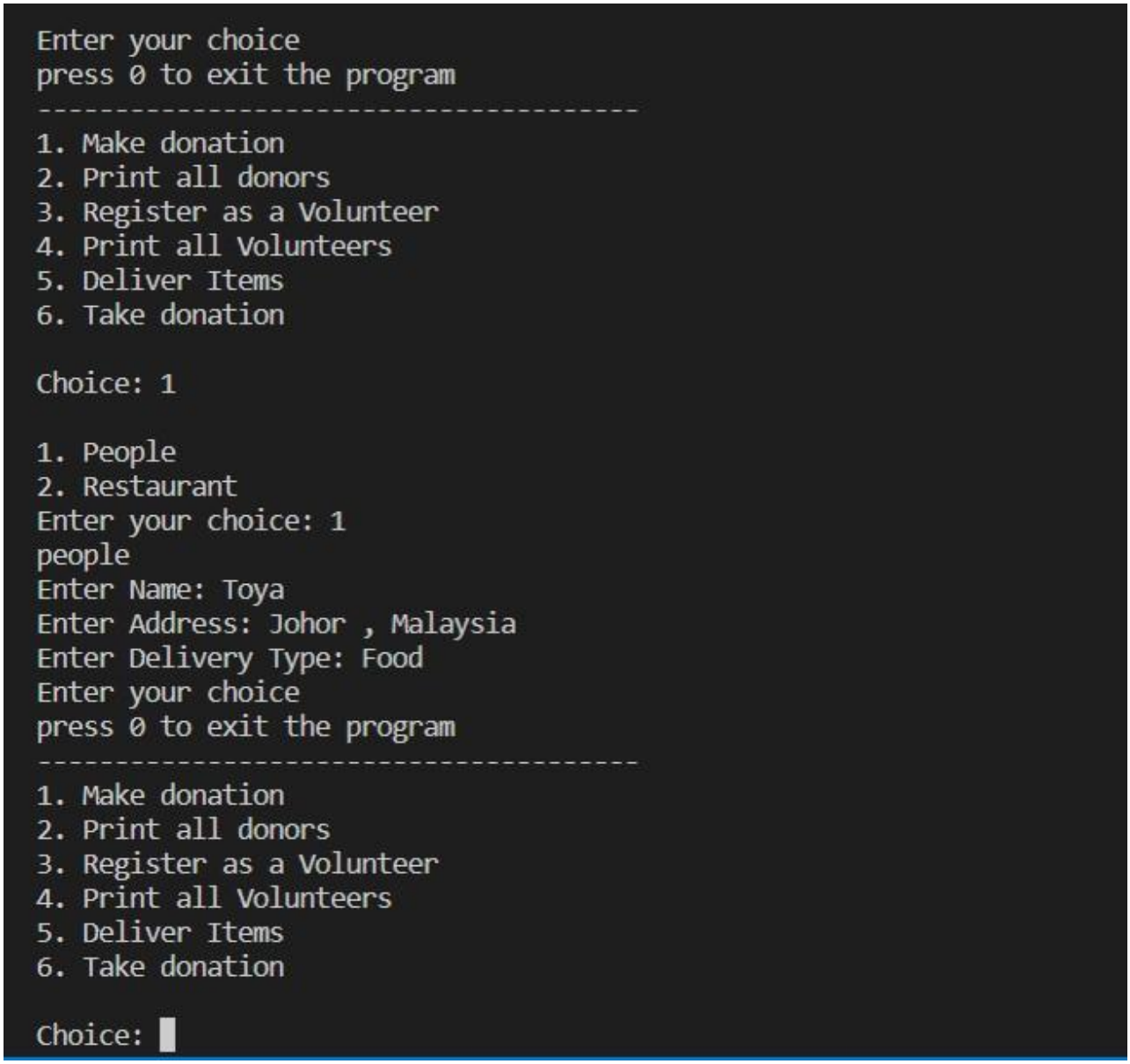

Fig: Choice 1

When choice 2 is selected all the donor information are shown in the display

```
Choice: 2
All Donor Information is as follows:
[Name: Toya, Address: Johor , Malaysia, Delivery Type: Food, Donor type: People]
[Name: Shimon, Address: Khulna 17 no , Bangladesh, Delivery Type: Food, Donor type: Restaurant
[Name: Tahzeeb, Address: 23 Skudai Johor , Johor, Delivery Type: Money, Donor type: People]
Enter your choice
press 0 to exit the program
---------------------------------------
1. Make donation
2. Print all donors
3. Register as a Volunteer
4. Print all Volunteers
5. Deliver Items
6. Take donation

Choice:
```

Fig: Choice 2

When choice 3 is selected , the system lets the Volunteer be registered as a member of the organization.

```
Choice: 3
Enter Name:
Amanullah Ahmed
Enter Age:
23
Enter Prof:
Student
Enter Role:
Deliver Items
Enter your choice
press 0 to exit the program
---------------------------------------
1. Make donation
2. Print all donors
3. Register as a Volunteer
4. Print all Volunteers
5. Deliver Items
6. Take donation

Choice:
```

Fig:Choice 3

When choice 4 is selected all the volunteer information are shown in the display

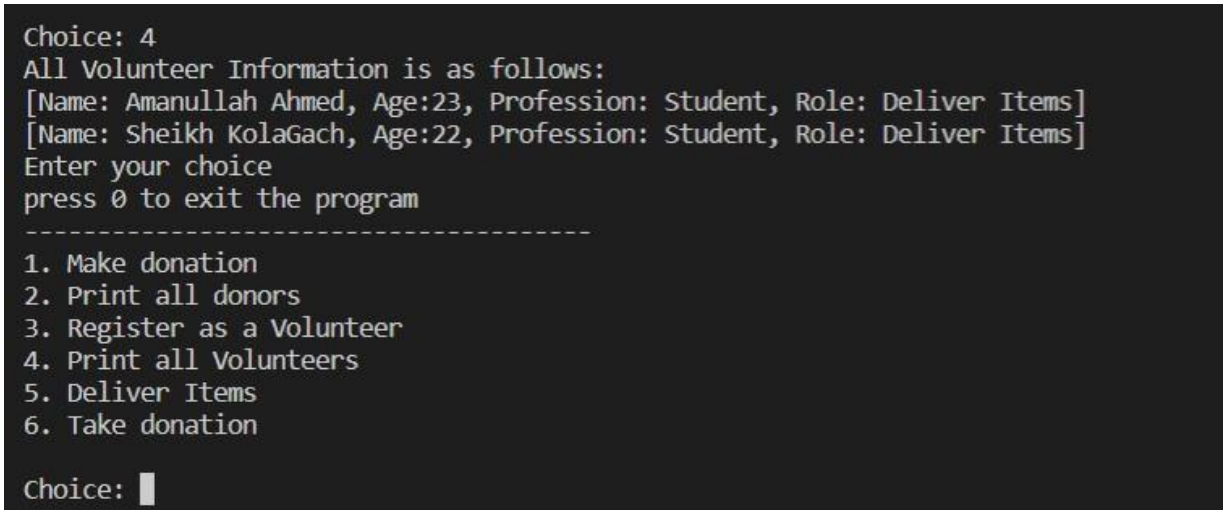

Fig: Choice 4

Option 5 will be activated once all the field have a person to make the delivery possible. Option 1 is needed for having donated goods. When use select option 4 a volunteer will be assigned to deliver the donation to the receiver. Option 6 is for receiver, those who want to take donation from the organization. Once it is selected, the user will be asked to provide the name, profession, number of family, the reason why user is requesting for donation and finally the address of the user. Afterwards, system will display the deliver option, according user choice the deliver option will be assigned.

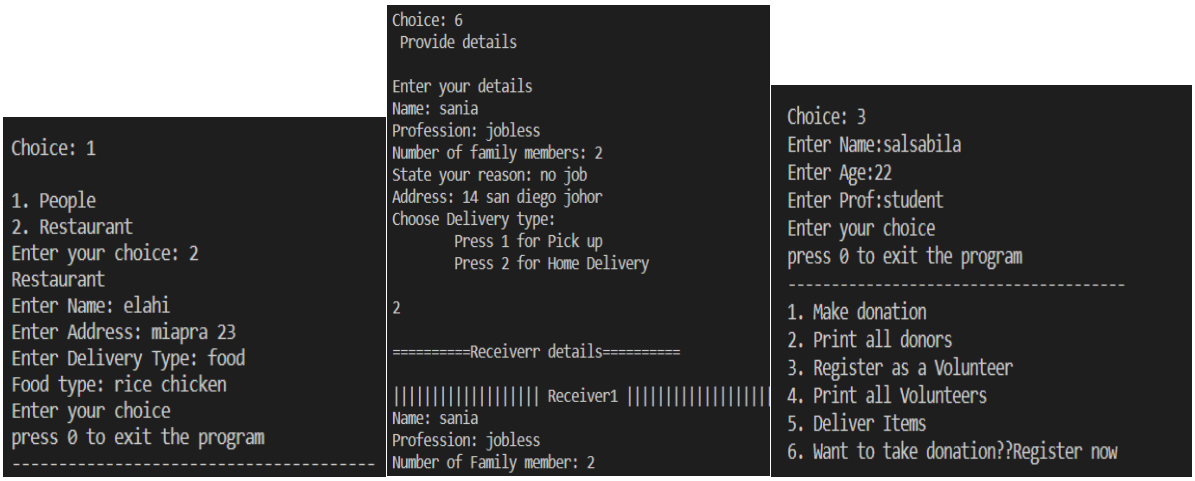

Fig: all choices needed for item delivery

When **option 5** is pressed , the delivery will be made

```
Choice: 5
sania Has got donation from elahi. Amount of money: 200.0
Enter your choice
press 0 to exit the program
--------------------------------------
1. Make donation
2. Print all donors
3. Register as a Volunteer
4. Print all Volunteers
5. Deliver Items
6. Want to take donation??Register now

Choice:
```

Fig : choice 5

```
Sorry we dont have any donation right now.. Later we will try to deliver the donation

Enter your choice
press 0 to exit the program
--------------------------------------
1. Make donation
2. Print all donors
3. Register as a Volunteer
4. Print all Volunteers
5. Deliver Items
6. Want to take donation??Register now

Choice:
```

Fig : If no donation available

```
Choice: 6
 Provide details

Enter your details
Name: Shimon
Profession: Student
Number of family members: 4
State your reason: lost job
Address: khulna
Choose Delivery type:
        Press 1 for Pick up
        Press 2 for Home Delivery

1
```

Figure: choice 6

After the above function, system will display the details of the user and deliver method.

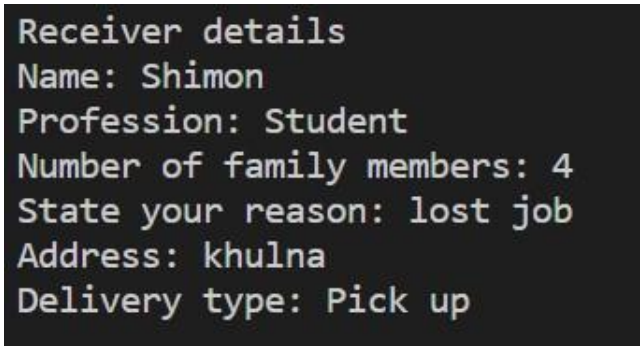

Fig: details of receiver