



BACS2063 Data Structures and Algorithms

ASSIGNMENT 202405

Declaration

- I confirm that I have read and complied with all the terms and conditions of Tunku Abdul Rahman University of Management and Technology's plagiarism policy.
- I declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own properly derived work.

| Student Name | Student ID | Prog / Tut.Grp | Signature |
|--------------|------------|----------------|-----------|
| Ong Yi Xin | 24WMR09097 | RSD2 G5 | |
| Chia Ming Yi | 24WMR09040 | RSD2 G5 | |
| Lim Jun Wei | 24WMR09078 | RSD2 G5 | |
| Ng Ian Kai | 24WMR09090 | RSD2 G5 | |
| | | | |

Note: The submission date and time will be according to the timestamp recorded in Google Classroom for Assignment Submission.

Ong Yi Xin

Chia Ming Yi

Lim Jun Wei

Ng Ian Kai

12
11
10
9
8
7
6
5
4
3
2
1

1
2
3
4
5
6
7
8
9
10
11

Outline

Part A: TEAM REPORT

1. Abstract Data Type (ADT)

1.1 ADT Specification

1.2 ADT Implementation

INDIVIDUAL REPORTS

2. Use of ADTs

Ong Yi Xin

Subsystem : Donee Management Subsystem

Donee Subsystem Menu

1.1 Create a new donee

1.2 View and Update Donee

1.2.2 Previous and Next Page of Donee List

1.2.3 Filter Donee List

1.2.4 Sort Donee List

1.2.5 Ascending or Descending List

1.2.6 Search Donee

1.3 View / Update / Remove Donee Detail

1.3.2 Edit Donee Detail

1.3.3 Remove Donee

1.3.4 Previous or Next Donee in List

1.3.5 Donation Distribution List of Donee Received

1.4 Clear Filter of List

1.5 Donee Summary Report

1.5.1 Total Value Donee Received Report

1.5.2 Top 10 Donee Received Selected Item\

1.5.3 Specific Donee Summary Report

Chia Ming Yi

Lim Jun Wei

Ng Ian Kai



Table of Contents

| | |
|--|-----------|
| Part A: TEAM REPORT | 5 |
| 1. Abstract Data Type (ADT) | 5 |
| 1.1 ADT Specification | 5 |
| 1.2 ADT Implementation | 9 |
| A. INDIVIDUAL REPORTS | 10 |
| 2. Use of ADTs | 10 |
| Ong Yi Xin | 10 |
| Subsystem : Donee Management Subsystem | 10 |
| Donee Subsystem Menu | 57 |
| 1.1 Create a new donee | 58 |

| | |
|---|-----|
| 1.2 View and Update Donee | 59 |
| 1.2.2 Previous and Next Page of Donee List | 60 |
| 1.2.3 Filter Donee List | 61 |
| 1.2.4 Sort Donee List | 63 |
| 1.2.5 Ascending or Descending List | 65 |
| 1.2.6 Search Donee | 66 |
| 1.3 View / Update / Remove Donee Detail | 70 |
| 1.3.2 Edit Donee Detail | 71 |
| 1.3.3 Remove Donee | 73 |
| 1.3.4 Previous or Next Donee in List | 73 |
| 1.3.5 Donation Distribution List of Donee Received | 74 |
| 1.4 Clear Filter of List | 76 |
| 1.5 Donee Summary Report | 76 |
| 1.5.1 Total Value Donee Received Report | 77 |
| 1.5.2 Top 10 Donee Received Selected Item\ | 77 |
| 1.5.3 Specific Donee Summary Report | 77 |
| Chia Ming Yi | 78 |
| Lim Jun Wei | 79 |
| Subsystem : Donation Management Subsystem | 79 |
| 1. Source Codes Listing | 80 |
| Controller: DonationControl.java | 80 |
| Entity class: Donation.java | 131 |
| 2. Screenshot | 134 |
| 1.1 Add New Donation | 135 |
| 1.2 View Donation | 138 |
| 1.2.1 View donation details (view next and previous donation details) | 140 |
| 1.2.2 View donation list by criteria | 143 |
| 1.3 Search Donation | 150 |
| 1.3.1 Search by donation ID | 150 |
| 1.3.2 Search by donor ID | 151 |
| 1.4 Update Donation | 152 |
| 1.4.1 Switch donor | 153 |
| 1.4.2 Modify quantity of current item | 154 |
| 1.4.3 Add new item | 155 |
| 1.4.4 Remove Item | 158 |
| 1.4.5 Update Notes | 159 |
| 1.5 Remove Donation | 160 |
| 1.6 Donation Report | 162 |
| 1.6.1 Top 10 Major Donation Report | 162 |
| 1.6.2 Top 10 Donated Items Report | 163 |
| 1.6.3 Top 10 Critical Shortages Report | 164 |
| 1.7 Create new donation item | 166 |
| Ng Ian Kai | 167 |
| 3. Source codes Listing. | 167 |
| a. Controller: DonationDistributionControl.java | 167 |

| | |
|--------------------------------------|-----|
| b. Entity: DonationDistribution.java | 186 |
| 4. Screenshots | 190 |

Part A: TEAM REPORT

1. Abstract Data Type (ADT)

1.1 ADT Specification

ADT List

List is a collection of entries of type T with a specific sequence that allows duplicates of entry. It allows for element accessing, insertion, removal and modification on any position based on the index of elements in the list. It can also be flexible for sorting and searching actions. Position starts from 1.

Boolean add(T newEntry)

Description: Add newEntry to the end of the list.

Postcondition: newEntry has been added to the end of the list.

Returns: A boolean value indicating whether the adding process is successful.

Boolean insertBefore(T existingData, T newData)

Description: Insert newData before the position of existing data.

Postcondition: newData has been inserted before the existing data in the list.

Returns: A boolean value indicating whether the insert process is successful.

Boolean insertAfter(T existingData, T newData)

Description: Insert newData after the position of the existing data.

Postcondition: newData has been inserted after the existing data in the list.

Returns: A boolean value indicating whether the insert process is successful.

Boolean remove(T anEntry)

Description: Remove the entry from the list that contains the matched data.

Postcondition: The matched data has been removed from the list.

Returns: A boolean value indicating whether the removing process is successful.

Boolean update(T updatedEntry)

Description: Find out the matched data from the list and replace it with the updatedEntry.

Postcondition: The matched data in the list has been updated.

Returns: A boolean value indicating whether the update process is successful.

T search(T data)

Description: Search and retrieve the data which is matched with the parameter data from the list.

Postcondition: The matched data in the list is retrieved.

Returns: The matched object of type T in the list.

Node<T> searchNode(Integer position)

Description: Search and retrieve the node with the specified **position** in the list.

Precondition: The specified position must be between 0 and the size of the list, excluding 0 and the size of the list.

Postcondition: The entry stored at the specified **position** in the list is retrieved.

Returns: The Node of type T storing the entry at the specified **position**.

Boolean contains(T anEntry)

Description: Determines whether the list contains **anEntry** or not.

Postcondition: The list remains unchanged.

Returns: true if **anEntry** is in the list, false otherwise.

bubbleSort(Boolean ascending, Comparator<T> comparator)

Description: Sorts the entries of a list in either ascending or descending order, based on the provided **comparator** implementation. This operation repeatedly steps through the list, compares adjacent entries, and swap them according to the **comparator** logic.

Postcondition: The list is sorted either ascending or descending, based on the comparator logic.

T getFirst()

Description: Get the first entry from the list.

Postcondition: The first entry object is retrieved from the list.

Returns: The first entry object of type T in the list.

T getLast()

Description: Get the last entry from the list.

Postcondition: The last entry object is retrieved from the list.

Returns: The last entry object of type T in the list.

T getPreviousData(T currentData)

Description: Get the entry which is before the position of currentData from the list.

Postcondition: Retrieve the object which is before the position of currentData.

Returns: The previous entry object of Type T of the currentData in the list.

T getNextData(T currentData)

Description: Get the entry which is after the position of currentData from the list.

Postcondition: Retrieve the object which is after the position of currentData.

Returns: The next entry object of Type T of the currentData in the list.

Integer getSize()

Description: Returns the integer size of elements in the list.

Postcondition: The integer size of elements in the list is returned.

Returns: The integer size of elements in the list.

Boolean isEmpty()

Description: Checks whether the list is empty.

Postcondition: Returns true if the list contains no elements, otherwise returns false.

Returns: A boolean value indicating whether the list is empty.

Boolean clear()

Description: Remove all the elements within the list.

Postcondition: Returns true if all the elements within the list are removed successfully, otherwise returns false.

Returns: A boolean value indicating whether all the elements within the list are removed successfully.

String toString()

Description: Retrieve the detail of the object in the form of String.

Postcondition: A string which contains the details of the list is retrieved.

Returns: A String.

swap(T data1, T data2)

Description: Switch the position between two objects in the list.

Postcondition: The two objects' positions have been switched.

ListInterface<T> intersect(ListInterface<T> otherList)

Description: Find out the similar objects between two lists then add them into a new list.

Postcondition: A list with similar objects between the two lists is retrieved.

Returns: A list with objects of Type T.

ListInterface<T> merge(ListInterface<T> otherList)

Description: Combine two lists by adding the objects of the second list from the end of the first list.

Postcondition: A combination list of two lists is retrieved.

Returns: A list with objects of Type T.

ListInterface<T> minus(ListInterface<T> otherList)

Description: Eliminates the similar objects between two lists.

Postcondition: The list which has excluded the similar objects between two lists is retrieved.

Returns: A list with objects of Type T.

ListInterface<T> filterList(T keyword)

Description: Find out all the objects which have the specified attributes.

Postcondition: The filtered list is retrieved.

Returns: A list with objects of Type T.

ListInterface<T> pagination(ListInterface<T> list, Integer page, Integer pageSize)

Description: Creates a new list containing only the objects from a specific page of an original list.

Postcondition: A list contains objects from the specified page of the original list based on the provided page and pageSize.

Returns: A list with objects of Type T.

1.2 ADT Implementation

<INSTRUCTIONS> (please delete all instructions before submitting your assignment report)

In NetBeans, use the print function to print to PDF the following:

- a. **Complete source code** with proper indentation for the Java interface of your collection ADT from 1.1.
- b. **Complete source code** with proper indentation for the Java implementation class of your collection ADT from 1.1.

IMPORTANT: If the above was taken from the sample source code or other sources, please indicate this information clearly in the comment section on top of your file.

</INSTRUCTIONS>

Copy paste adt interface class code and the adt java class code, use blue color for the method way different from lecture note or tutor sample. Remember to put the author name for each adt method.

b. Java Implementation Class: DoublyLinkedList.java

A. INDIVIDUAL REPORTS

2. Use of ADTs

| Name | Student ID | Prog / Tut.Grp | Signature |
|------------|------------|----------------|---|
| Ong Yi Xin | 24WMD09097 | RSD1 G5 |  |

Subsystem : Donee Management Subsystem

1. Source codes for Control classes
 - a. Controller: DoneeControl.java

```
package control;

import adt.DoublyLinkedList;
import adt.ListInterface;
import static boundary.CLONCharity.exitStr;
import static boundary.CLONCharity.option;
import static boundary.CLONCharity.optionYN;
import static boundary.CLONCharity.scan;
import static boundary.CLONCharity.h;
import static boundary.CLONCharity.v;
import static boundary.CLONCharity.exitFlag;
import dao.Initializer;
import entity.Address;
import entity.DonationDistribution;
import entity.Donee;
import entity.InventoryMovement;
import entity.Person;
import entity.Item;
import utility.PaginationResult;

/**
 *
 * @Ong Yi Xin
 */
public class DoneeControl {

    // Declare list
    private static ListInterface<Donee> doneeListP = Initializer.doneeListP;
    private static ListInterface<Donee> doneeListO = Initializer.doneeListO;
    private static ListInterface<Donee> doneeListF= Initializer.doneeListF;
    private static ListInterface<DonationDistribution> distributionList = Initializer.distributionList;
    private static ListInterface<Item> foodBeverageList = Initializer.foodBeverageList;
```

```

private static ListInterface<Item> moneyList = Initializer.moneyList;

private static ListInterface<Donee> doneeList = new DoublyLinkedList<>();
private static ListInterface<Donee> tempDoneeList = new DoublyLinkedList<>();
private static ListInterface<DonationDistribution> tempDistributionList = new DoublyLinkedList<>();
private static ListInterface<InventoryMovement> stockOutList = new DoublyLinkedList<>();

// Decalre object
private static Donee tempDonee = new Donee(); // For temparary store keyword of search that
input from user
private static Donee doneeFound = new Donee(); // For temporary store the donee data that search
sucess
private static DonationDistribution distribution = new DonationDistribution();
private static DonationDistribution tempDistribution = new DonationDistribution();

// Declare variable help for easier control
private static String searchByStr = null, keyword = null, searchBy = null, sortByStr = "ID", sortBy =
"id", orderBy = "Asc";
private static int currentPage = 1;
private static String filterByGender = null, filterByType = null, filterByState = null,
filterByRegisterDate = null;
private static int filterByRangeMin = -1, filterByRangeMax = -1;

// Main
public static boolean doneeMenu(){
    updateTotalDonationDistribution(null, null); // update the total received donation of donee to
newest
    option = "1";

    while(!option.equals("0")){
        option = "1";

        System.out.print(doneeMenuDisplay());
        option = scan.nextLine();

        switch(option.toLowerCase()){
            case "0":
                System.err.println(exitStr);
                return true;
            case "1":
                addDonee();
                break;
            case "2":
                viewDoneeList();
                break;
        }
    }
}

```

```

        case "3":
            reportMenu();
            break;
        default:
            System.err.println(v.errOptionInt);
        }
        option = "1"; // To prevent over exit
    } // while

    return false;
}

// Add
public static int addDonee(){ // Control for add donee input

do{
    String name = null, icNo = null, phoneNumber = null, email = null;
    char gender = '0', type = '0';
    Address newAddress = new Address();
    exitFlag = false; // reset exit flag to run condition

    System.out.print("\n\n New Donee Detail\n"
        + " ----- \n");

    // enter name
    while(exitFlag == false) {
        //Get donee name
        name = h.inputName();

        if (name == null){
            optionYN = h.inputYN(" Do you confirm to discard add new donee?");
            if(optionYN.equals("Y")) return -1;
            else exitFlag = false; //Exit Condition
        } else break;
    }

    // enter ic number
    while(exitFlag == false) {
        //Get donee IC Number
        icNo = h.inputIcNo();
        if (icNo == null) {
            optionYN = h.inputYN(" Do you confirm to discard add new donee?");
            if (optionYN.equals("Y")) return -1;
            else exitFlag = false; //Exit Condition
        } else break;
    }
}

```

```
// enter gender
while(exitFlag == false) {
    //Get donee gender
    gender = h.inputGender();
    if (gender == '0') {
        optionYN = h.inputYN(" Do you confirm to discard add new donee?");
        if (optionYN.equals("Y")) return -1;
        else exitFlag = false; //Exit Condition
    } else break;
}

// enter phone number
while(exitFlag == false) {
    //Get donee phone number
    phoneNumber = h.inputPhoneNumber();
    if (phoneNumber == null) {
        optionYN = h.inputYN(" Do you confirm to discard add new donee?");
        if (optionYN.equals("Y")) return -1;
        else exitFlag = false; //Exit Condition
    } else break;
}

// enter email
while(exitFlag == false) {
    //Get donee email
    email = h.inputEmail();
    if (email == null) {
        optionYN = h.inputYN(" Do you confirm to discard add new donee?");
        if (optionYN.equals("Y")) return -1;
        else exitFlag = false; //Exit Condition
    } else break;
}

// enter type
while(exitFlag == false) {
    //Get donee type
    type = h.inputType();
    if (type == '0') {
        optionYN = h.inputYN(" Do you confirm to discard add new donee?");
        if (optionYN.equals("Y")) return -1;
        else exitFlag = false; //Exit Condition
    } else break;
}

// enter address
while(exitFlag == false) {
```

```

//Get donee address
newAddress = h.inputAddress();
if (newAddress == null) {
    optionYN = h.inputYN(" Do you confirm to discard add new donee?");
    if (optionYN.equals("Y")) return -1;
    else exitFlag = false; //Exit Condition
} else break;
}

// if all the data needed have be enter
if (exitFlag == false) {
    //Create Donee object
    Donee newDonee = new Donee(h.capitalizeInput(name), icNo, gender, phoneNumber, email,
type, newAddress);

    /* prevent unlogic loop when user don't want to add another donee,
       if using optionYN at here, it will loop again on this situation. */
    String optionDiscard = "N";
    while(optionDiscard.equals("N")){
        // Display the data after edit for confirmation
        System.out.println("\n\n" + newDonee.toStringDetail());

        // Confirmation to add new donee?
        optionYN = h.inputYN(" Do you confirm to add this donee?");

        if (optionYN.equals("Y")){
            // Add new donee to doneeList
            if(Character.toUpperCase(type) == 'P') doneeListP.add(newDonee);
            if(Character.toUpperCase(type) == 'O') doneeListO.add(newDonee);
            if(Character.toUpperCase(type) == 'F') doneeListF.add(newDonee);

            System.err.println("\n Donee add successful. \n");
            optionYN = h.inputYN(" Do you want to add another donee?");
            break;
        } else {
            /* At here, optionYN now is "N", if confirm discard,
               the optionYN and optionDiscard both will change to "Y" */
            optionDiscard = h.inputYN(" Do you confirm to discard add the donee?");
        }
    }
}

} //while
}// if
}while(optionYN.equals("Y"));

```

```

        return 1;
    }

// View Donee List
// Menu and list for all flow of read, update, remove, donee's donation distribution list, filter,
sorting, and searching
public static void viewDoneeList() {
    String option = "1";

    clearFilter(); //Reset All Filter
    doneeList = new DoublyLinkedList<>(); // Reset the doneeList to avoid null pointer
    doneeList = doneeListP.merge(doneeListO.merge(doneeListF)); // merged the list to get newest
data
    tempDoneeList = doneeList; // Reset the tempDoneeList to doneeList

    while (!option.equals("0")) {
        exitFlag = false; // Reset exit flag to prevent over exit
        option = "1"; // Reset the option to prevent wrong exit

        // sort the donee list base on user input, default is sort by id in asc
        tempDoneeList = bubbleSortDonee(sortBy, orderBy, tempDoneeList);

        // generate the pagination list including total page
        PaginationResult paginatedResult = h.generatePaginationResult(tempDoneeList, currentPage,
7);

        //Display
        System.out.print("\n\n" + filterDisplay());
        System.out.print(doneePaginatedListDisplay(paginatedResult, tempDoneeList.getSize()));
        System.out.print(" Option ('0' - Back): ");

        option = scan.nextLine().toUpperCase();

        switch (option) {
            case "P": // Previous page
                if (currentPage > 1) {
                    currentPage--;
                } else {
                    System.err.println(h.firstPageInform);
                    currentPage = paginatedResult.getTotalPages();
                }
                break;
            case "N": // Next page
                if (currentPage < paginatedResult.getTotalPages()) {
                    currentPage++;
                }
        }
    }
}

```

```

} else {
    System.err.println(h.lastPageInform);
    currentPage = 1;
}
break;
case "F": // Filter
filterDonee(tempDoneeList);
break;
case "H": // Search
tempDoneeList = searchDoneeMenu(tempDoneeList);
currentPage = 1;
break;
case "M": // Edit Donee
tempDoneeList = doneeSearchSuccess(tempDoneeList, paginatedResult);
break;
case "L": // Distribution list of donee
doneeDistributionList(tempDoneeList);
break;
case "S": // Sort
tempDoneeList = sortDoneeMenu(tempDoneeList);
break;
case "A": // Asc
if (orderBy.toLowerCase().equals("asc")) {
    break;
} else {
    orderBy = "Asc";
    tempDoneeList = bubbleSortDonee(sortBy, orderBy, tempDoneeList);
}
break;
case "D": // Desc
if (orderBy.toLowerCase().equals("desc")) {
    break;
} else {
    orderBy = "Desc";
    tempDoneeList = bubbleSortDonee(sortBy, orderBy, tempDoneeList);
}
break;
case "C": // Clear Filter
tempDoneeList = doneeListP.merge(doneeListO.merge(doneeListF));
clearFilter();
break;
case "0": // Back condition
tempDoneeList = doneeList; // reset tempDoneeList
System.err.println(exitStr);
break;
default:
System.err.println(v.errInput);

```

```

} // switch

if(exitFlag == true) break; // Exit condition
} // while
}

// Control the filter flow and process
public static int filterDonee(ListInterface<Donee> list) {
    Address tempAddress = new Address();
    ListInterface<Donee> originalDoneeList = new DoublyLinkedList<>();
    ListInterface<Donee> filterList = new DoublyLinkedList<>();
    ListInterface<Donee> searchList = new DoublyLinkedList<>();

    doneeList = new DoublyLinkedList<>(); // Reset the list to avoid null pointer
    doneeList = doneeListP.merge(doneeListO.merge(doneeListF));

    do {
        exitFlag = true; // Reset exitFlag to prevent over exit
        String keyword = null;
        int[] minMax = {0, 0};
        doneeFound = new Donee();
        boolean runFilter = true;

        System.out.format(
            "\n\n Select filter\n"
            + " %15s %15s\n"
            + " %15s %15s\n"
            + " %-31s\n"
            + " Filter donee by? ('0' - Back, 'X' - Exit to Susbsystem menu): ",
            "1. Gender", "2. Type",
            "3. State", "4. Register Date",
            "5. Num of Donation Received");
        option = scan.nextLine();

        switch (option.toLowerCase()) {
            case "1": // Filter by gender
                keyword = h.inputSearchKeyword("gender", 0);
                if (keyword != null) filterByGender = keyword.toUpperCase();
                else {runFilter = false; minMax = null;}
                break;
            case "2": // Filter by type
                keyword = h.inputSearchKeyword("type", 0);
                if (keyword != null) filterByType = keyword.toUpperCase();
                else {runFilter = false; minMax = null;}
                break;
            case "3": // Filter by state
                keyword = h.inputSearchKeyword("state", 0);
                if (keyword != null) filterByState = keyword.toUpperCase();
        }
    } while (!exitFlag && runFilter);
}

```

```

        else {runFilter = false; minMax = null;}
        break;
    case "4": // Filter by register date
        keyword = h.inputSearchKeyword("registerdate", 0);
        if (keyword != null) filterByRegisterDate = keyword;
        else {runFilter = false; minMax = null;}
        break;
    case "5": // Filter by number of donation received
        minMax = h.inputMinMaxOfRange();
        if (minMax != null) {
            filterByRangeMin = minMax[0];
            filterByRangeMax = minMax[1];
        }
        else {runFilter = false; minMax = null;}
        break;
    case "0": // Back condition
        exitFlag = false;
        option = "1"; // Rset option to prevent exit more than one time
        return -1;
    case "x": // Exit condition
        exitFlag = true;
        return -1;
    default:
        System.err.println(v.errInput);
        exitFlag = false; // reset exit flag to re-loop
        runFilter = false;
    }

if (keyword == null && minMax == null) exitFlag = false; // Exit condition

if(runFilter == true){
    // Identify get the merged list or specific list of type
    tempDoneeList = doneeList;

    if (filterByType != null) {
        tempDoneeList = new DoublyLinkedList<>();
        switch (filterByType.toUpperCase()) {
            case "P":
                tempDoneeList = doneeListP;
                break;
            case "O":
                tempDoneeList = doneeListO;
                break;
            case "F":
                tempDoneeList = doneeListF;
                break;
        }
    }
}

```

```

    }

}

if (filterByGender != null) {
    doneeFound.setGender(filterByGender.toUpperCase().charAt(0));
    tempDoneeList = h.filterDoneeList(tempDoneeList, "gender", doneeFound);
}

if (filterByState != null) {
    tempAddress.setState(filterByState.toUpperCase());
    doneeFound.setDefaultAddress(tempAddress);
    tempDoneeList = h.filterDoneeList(tempDoneeList, "state", doneeFound);
}

if (filterByRegisterDate != null) {

    doneeFound.setName(filterByRegisterDate);
    tempDoneeList = h.filterDoneeList(tempDoneeList, "registerdate", doneeFound);
}

if (filterByRangeMin > -1 && filterByRangeMax > -1) {
    minMax[0] = filterByRangeMin;
    minMax[1] = filterByRangeMax;

    // Update overall donee donation received and filter out donee donation received out of
the range
    filterList = filterRangeOfDonationReceived(minMax);

    // minus the list of donee that out of range from the donee list
    filterList = tempDoneeList.minus(filterList);

    // intersect the list after minus with the filtered list which may contain multi filter
    tempDoneeList = tempDoneeList.intersect(filterList);

    sortByStr = "Number of Donation Received";
    sortBy = "donationreceived";
    orderBy = "Asc";

    // sort the list by number of donation receiveds
    if (minMax[0] != minMax[1] && tempDoneeList.getSize() > 0)
        tempDoneeList = bubbleSortDonee(sortBy, orderBy, tempDoneeList);

}

if (!searchByStr.equals("None")) {
    tempDonee = new Donee();
}

```

```

tempDonee = caseOfSearchBy(searchBy, DoneeControl.keyword);
tempDoneeList = h.filterDoneeList(tempDoneeList, searchBy, tempDonee);

}

// if the list after filter is null and no record
if (tempDoneeList.getSize() <= 0) {

    // Prompt inform message to user
    System.err.println("\n\n \"No donee match. Redirect you back to list before filter
action.\"");

    // Reset the list to list before filter for avoid null pointer
    tempDoneeList = list;

} else currentPage = 1;
}

} while (exitFlag == false);

exitFlag = false; // Reset the exit flag to prevent over exit
return 1;
}

public static ListInterface<Donee> filterRangeOfDonationReceived(int[] minMax) {
    ListInterface<Donee> tempDoneeList = new DoublyLinkedList<>();

    updateTotalDonationDistribution(null, null);
    tempDonee = doneeList.getFirst();

    while (tempDonee != null) {

        if (tempDonee.getTotalDonationReceived() < minMax[0] ||
tempDonee.getTotalDonationReceived() > minMax[1])
            tempDoneeList.add(tempDonee); // add the donee to list if the donation received is out of
the range

        tempDonee = doneeList.getNextData(tempDonee);
    }

    return tempDoneeList; // return the list of donee that is out of the require range
}

```

```

// Edit & Remove
// Control selection for futher action to a specific donee, (edit and removes)
public static ListInterface<Donee> doneeSearchSuccess(ListInterface<Donee> list, PaginationResult
paginatedResult) {
    String digit = null;
    optionYN = "Y";

    while (optionYN.equals("Y")) {

        // reset variable for prevent error
        exitFlag = false;
        doneeFound = new Donee();

        digit = h.inputIdByStr(list.getLast().getId()); // Get id from user

        if (digit != null) {
            // set the id to donee for serching purpose
            doneeFound.setId(Integer.parseInt(digit));
            doneeFound = list.search(doneeFound); // search the donee in list

            if (doneeFound != null) {
                option = "1"; // reset option for looping

                while (!option.equals("0")) {
                    doneeFound.setSearchBy(""); // reset the seachBy to prepare for next or previous
donees

                    System.out.print("\n\n\n\n" + doneeFound.toStringDetail()); // Display detail of donee
                    System.out.format(
                        " %-5s - %-20s %-5s - %-20s \n"
                        + " %-5s - %-20s %-5s - %-20s\n",
                        "'E'", "Edit", "'R'", "Remove", "'P'",
                        "Previous donee", "'N'", "Next donee");
                    System.out.print("\n Select option for futher action to the donee? "+
                        "('0' - Back, 'X' - Exit to Subsystem Menu): ");
                    option = scan.nextLine().toUpperCase();

                    switch (option.toUpperCase()) {
                        case "E": // Edit
                            editDonee(doneeFound);
                            if (exitFlag == true) return list; // Exit conditon
                            break;
                        case "R": // Remove
                            optionYN = h.inputYN(" Do you confirm to delete Donee " + doneeFound.getIdStr() +
                            "?");
                            if (optionYN.equals("Y")) { // If confirm removr the donee

```

```

doneeFound.setSearchBy("id"); // Set the searchBy for compareTo function use

// remove donee for both displaying list and the dao donee list
list.remove(doneeFound);

if(doneeListP.search(doneeFound) != null) doneeListP.remove(doneeFound);
else if(doneeListO.search(doneeFound) != null) doneeListO.remove(doneeFound);
else if(doneeListF.search(doneeFound) != null) doneeListF.remove(doneeFound);

System.err.println(" Donee remove successful.");

// set the variable to exit loop and back to donee list as donee already removed
option = "0";
optionYN = "N";
}
break;
case "P": // Previous donee in list of current donee
doneeFound = list.getPreviousData(doneeFound);
if (doneeFound == null) {
    System.err.println(h.firstDoneeInform);

    // set current donee as last donee if current donee is the first donee in list
    doneeFound = list.getLast();
}
break;
case "N": // Next donee in list of current donee
doneeFound = list.getNextData(doneeFound);
if (doneeFound == null) {
    System.err.println(h.lastDoneeInform);

    // set current donee to first donee if the current donee is last donee in list
    doneeFound = list.getFirst();
}
break;
case "0": // Back condition
exitFlag = false;
return list;
case "X": // Exit condition
exitFlag = true;
return list;
default:
    System.err.println(v.errCharInput);
} // switch
}// while
option = "1"; // reset option to prevent over exit
} else {
    System.err.println(v.errDoneeNotFound); // When the donee not found
}

```

```

        doneeFound = new Donee(); // reset donee to prevent null pointer
    }
} else optionYN = "N"; // Back condition

} // while

return list;
}

// Process for edit donee
public static int editDonee(Donee doneeFound){ // Handle for specific donee information change
    boolean changeType = false, p = false, o = false, f = false;
    char type = 'N';
    if (doneeListP.search(doneeFound) != null) p = true;
    else if (doneeListO.search(doneeFound) != null) o = true;
    else if (doneeListF.search(doneeFound) != null) f = true;

    do{ // Loop again while not confirm edit

        do{ // Loop while want to edit other information of the donee
            changeType = false;
            optionYN = "Y"; // Reset the optionYN
            exitFlag = false; // Reset the exitFlag

            // Display donee current information
            System.out.print("\n\n Donee " + doneeFound.getIdStr() + "\n" +
doneeFound.toStringDetail());

            System.out.format(
                "\n"
                + " %-15s %-15s\n"
                + " %-15s %-15s\n"
                + " %-15s %-15s\n"
                + " %-15s\n"
                + " Option ('0' - Back, 'X' - Exit to Subsystem Menu): ",
                "1. Name", "2. IC Number", "3. Gender", "4. Phone Number",
                "5. Email", "6. Type", "7. Address", "0. Back");
            option = scan.nextLine();

            System.out.println("");


switch(option.toLowerCase()){
    case "1":
        String name = h.inputName();
        if(name == null) exitFlag = true; // Back Condition
        else doneeFound.setName(h.capitalizeInput(name));
        break;
    case "2":

```

```

String icNo = h.inputIcNo();
if(icNo == null) exitFlag = true; // Back Condition
else doneeFound.setIcNo(icNo);
break;
case "3":
    char gender = h.inputGender();
    if(gender == 'N') exitFlag = true; // Back Condition
    else doneeFound.setGender(gender);
    break;
case "4":
    String phoneNumber = h.inputPhoneNumber();
    if(phoneNumber == null) exitFlag = true; // Back Condition
    else doneeFound.setPhoneNumber(phoneNumber);
    break;
case "5":
    String email = h.inputEmail();
    if(email == null) exitFlag = true; // Back Condition
    else doneeFound.setEmail(email);
    break;
case "6":
    type = h.inputType();
    if(type == 'N') exitFlag = true; // Back Condition
    else {
        doneeFound.setType(Character.toUpperCase(type));

        switch (Character.toUpperCase(type)){
            case 'P':
                if (p != true) changeType = true;
                break;
            case 'O':
                if (o != true) changeType = true;
                break;
            case 'F':
                if (f != true) changeType = true;
                break;
        }
    }
    break;
case "7":
    Address address = h.inputAddress();
    if(address == null) exitFlag = true; // Back Condition
    else doneeFound.setDefaultValueAddress(address);
    break;
case "0":
    // Ask for confirm discard change?
    optionYN = h.inputYN(" Are you sure to discard edit");

```

```

if(optionYN.equals("Y")){
    System.err.println(exitStr);
    option = "1";
    return 0;
}

exitFlag = true; // If not confirm, set the exitFlag to true to loop back to re-select option
break;
case "x":
    optionYN = h.inputYN(" Are you sure to discard edit");
    if(optionYN.equals("Y")){
        System.err.println(exitStr);
        exitFlag = true; // set again the exit flag
        return 0;
    }
    exitFlag = true;
    break;
default:
    System.err.println(v.errInput);
    exitFlag = true;
}

// When user not select to exit the action, ask for edit another information?
if (exitFlag == false) optionYN = h.inputYN(" Do you want to change other information?");

// Loop while want to edit other information or exit from the input
}while(optionYN.equals("Y") || exitFlag == true);

// Display donee's information after modify
System.out.println(doneeFound.toStringDetail());

// Ask for confirmation to edit the donee
optionYN = h.inputYN(" Do you confirms to edit Donee " + doneeFound.getIdStr() + " information?");

if(optionYN.equals("Y")){
    // If confirm edit the donee
    doneeFound.setSearchBy(""); // Set searchBy to run the compareTo in update function
}

if (changeType == true){
    switch (Character.toUpperCase(type)) {
        case 'P':
            doneeListP.add(doneeFound);
            break;
        case 'O':
            doneeListO.add(doneeFound);
            break;
        case 'F':
}

```

```

        doneeListF.add(doneeFound);
        break;
    }
    if(p == true) doneeListP.remove(doneeFound);
    if(o == true) doneeListO.remove(doneeFound);
    if(f == true) doneeListF.remove(doneeFound);
} else {
    if(p == true) doneeListP.update(doneeFound);
    if(o == true) doneeListO.update(doneeFound);
    if(f == true) doneeListF.update(doneeFound);
}

System.out.println(" Donee " + doneeFound.getIdStr() + " information edit successful.\n");
}

}while(optionYN.equals("N"));// Do while not confrim edit the donee information

exitFlag = false;
return 1;
}

// Donee's Donation Distribution List
public static boolean doneeDistributionList(ListInterface<Donee> list){
    int verify = 0;

    // update the total amount and times of donation received
    updateTotalDonationDistribution(null, null);

    option = "1";
    while(!option.equals("0")){
        // Get donee id from user input
        doneeFound = selectDonee(list);

        if(option.equals("0")) break; // Back condition

        if(doneeFound != null){
            // get the distribution of the specific donee
            tempDistributionList = getSpecificDoneeDistribution(null, doneeFound);

            while(!option.equals("0")){
                // Display the donation distribution list that found
                doneeDistributionDisplay(tempDistributionList, doneeFound);

                if(tempDistributionList.getSize() > 0){ // if the list not null, let user to select futher action
                    System.out.print("\n\n Enter distribution ID to view futher detail "

```

```

        + "('0' - Back, 'X' - Exit to Subsystem Menu ):");
option = scan.nextLine();

switch (option.toLowerCase()){
    case "0": // Back condition
        exitFlag = false; // prevent over exit
        return false;
    case "x": // Exit Condition
        exitFlag = true; return true;
    default:
        // verify to the user input
        verify = v.verifyIsDigit(option, 1, 4);

        if(verify == 1)
            verify = v.isValidID(Integer.parseInt(option), tempDistributionList.getLast().getId());
        if(verify == 1){
            tempDistribution = new DonationDistribution(); // reset list to prevent null pointer
            tempDistribution.setId(Integer.parseInt(option));

            // serch and get the distribution of user want
            distribution = tempDistributionList.search(tempDistribution);

            if (distribution != null) {
                while(!option.equals("0")){
                    // Display distribution detail
                    System.out.println("\n\n" + distribution.toString());

                    System.out.println(" 'P' - Previous distribution, 'N' - Next distribution of "
                        + doneeFound.getName());
                    System.out.print(" Action ('0' - Back, 'X' - Exit to Subsystem Menu): ");
                    option = scan.nextLine();

                    switch (option.toLowerCase()) {
                        case "p": // Previous distribution of the donee in lisr
                            distribution = tempDistributionList.getPreviousData(distribution);
                            if (distribution == null) {

                                /* If the distribution already is the first in the list,
                                redirect user to the last distribution */
                                distribution = tempDistributionList.getLast();

                                // inform to user is the first distribution ady
                                System.err.println(h.firstDistributionInform);

                            }
                            break;
                    }
                }
            }
        }
    }
}

```

```

        case "n": // next distribution of the donee in list
            distribution = tempDistributionList.getNextData(distribution);
            if (distribution == null) {

                /* If the distribution already is the last in the list,
                   redirect user to the first distribution */
                distribution = tempDistributionList.getFirst();

                // inform to user is the last distribution ady
                System.err.println(h.lastDistributionInform);

            }
            break;
        case "0": // Back condition
            break;
        case "x": // Exit condition
            exitFlag = true; return true;
        default:
            System.err.println(v.errInput);
    } // switch

} // while

// Reset the option for loop to select donee when back from viewing
distribution detail
if (option.equals("0")) option = "1";

} else System.err.println(v.errDistributionNotFound);

} // if the id enter are in available range

}// switch option for futher action
} else { // if the list are null, let user select back or exit
    System.out.print("\n\n Enter '0' to back, 'X' for exit to Subsystem Menu: ");
    option = scan.nextLine();
    switch(option.toLowerCase()){
        case "x":
            exitFlag = true; return true;
        case "0":
            exitFlag = false; return false;
        default: System.err.println(v.errInput);
    }
}
} // while

} else System.err.println(v.errDoneeNotFound);

```

```

// Reset the option for loop to select donee when back from viewing distribution detail
if (option.equals("0")) option = "1";

} // while

return false;
}

// Search
// Menu Control for search
public static ListInterface<Donee> searchDoneeMenu(ListInterface<Donee> list){

do{
    exitFlag = true; // Reset exitFlag to prevent over exit

    System.out.format(
        "\n\n Search Donee By\n"
        + "%-15s %-15s %-15s\n"
        + "%-15s %-15s %-15s\n"
        + " Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu): ",
        "1. Donee ID", "2. Name", "3. IC Number",
        "4. Phone Number", "5. Email", "6. Address");
    option = scan.nextLine();

    switch(option.toLowerCase()){
        case "1":
            tempDoneeList = searchBy("id", list);
            break;
        case "2":
            tempDoneeList = searchBy("name", list);
            break;
        case "3":
            tempDoneeList = searchBy("icno", list);
            break;
        case "4":
            tempDoneeList = searchBy("phonenumer", list);
            break;
        case "5":
            tempDoneeList = searchBy("email", list);
            break;
        case "6":
            tempDoneeList = searchBy("address", list);
            break;
        case "0":
            exitFlag = true;
    }
}

```

```

        break;
    case "x":
        exitFlag = true;
        return list;
    default:
        System.err.println(v.errInput);
        exitFlag = false;
    }
    if(tempDoneeList == null) exitFlag = false;
}while(exitFlag == false);

if(option.equals("0")) option = "1"; // Rset option to prevent exit more than one time

exitFlag = false; // Reset exitFlag for prevent error exit
return tempDoneeList;
}
// Process control for search
public static ListInterface<Donee> searchBy(String searchBy, ListInterface<Donee> list){
    optionYN = "Y"; // reset optionYN for looping purpose
    clearFilter(); // clear all the filter

    doneeList = new DoublyLinkedList<>(); // Reset the list to avoid null pointer
    doneeList = doneeListP.merge(doneeListO.merge(doneeListF));

    while(optionYN.equals("Y")){
        tempDonee = new Donee(); // reset tempDonee to prevent null pointer

        System.out.println("\n Search by\n" + " -----");
        keyword = h.inputSearchKeyword(searchBy, doneeList.getLast().getId());

        if(keyword != null){

            tempDonee = caseOfSearchBy(searchBy, keyword);
            tempDoneeList = new DoublyLinkedList<>(); // Reset the tempDoneeList to prevent null
pointer
            tempDoneeList = h.filterDoneeList(doneeList, searchBy, tempDonee); //Get the list after filter

            // return list after filter if the list are not null
            if (tempDoneeList.getSize() != 0 && tempDoneeList != null) return tempDoneeList;
            else System.err.println(v.errDoneeNotFound);

        } else return null;

    }// while

    return tempDoneeList;
}

```

```

public static Donee caseOfSearchBy(String searchBy, String keyword){
    switch (searchBy.toLowerCase()) {
        case "id":
            tempDonee.setId(Integer.parseInt(keyword));
            searchByStr = "ID : " + keyword;
            break;
        case "name":
            tempDonee.setName(keyword);
            searchByStr = "Name : " + keyword;
            break;
        case "icno":
            tempDonee.setIcNo(keyword);
            searchByStr = "IC Number : " + keyword;
            break;
        case "phonenumerber":
            tempDonee.setPhoneNumber(keyword);
            searchByStr = "Phone Number : " + keyword;
            break;
        case "email":
            tempDonee.setEmail(keyword);
            searchByStr = "Email : " + keyword;
            break;
        case "address":
            tempDonee.setSearchAddressStr(keyword);
            searchByStr = "Address : " + keyword;
            break;
    }
    DoneeControl.searchBy = searchBy;
    return tempDonee;
}

```

```

// Report
public static boolean reportMenu(){
    do{
        exitFlag = false;
        System.out.print(reportMenuDisplay());

        option = scan.nextLine();

        switch(option.toLowerCase()){
            case "1":
                exitFlag = totalValueReport();
                break;
            case "2":

```

```

        exitFlag = top10ReceivedOfSelectedItem();
        break;
    case "3":
        exitFlag = doneeSummaryReport();
        break;
    case "0":
        return true; // Exit Condition
    default:
        System.err.println(v.errInput);
    }

}while(exitFlag == false);

return false;
}

public static boolean totalValueReport(){
    sortBy = "totalValue";
    orderBy = "Desc";
    String date = null;
    char doneeType = 'N';

    doneeList = new DoublyLinkedList<>(); // Reset the list to avoid null pointer

    do{
        do{
            // Get the type of donee for generate report
            System.out.println("\n\n" + doneeTypeMenuDisplay());
            System.out.print(" Select type of donee for generate report. ('0' - Back): ");
            option = scan.nextLine();

            switch (option.toLowerCase()){
                case "p":
                    doneeType = 'P';
                    doneeList = doneeListP;
                    break;
                case "o":
                    doneeType = 'O';
                    doneeList = doneeListO;
                    break;
                case "f":
                    doneeType = 'F';
                    doneeList = doneeListF;
                    break;
                case "0":
                    return false;
                default:

```

```

System.err.println(v.errType);
option = "0";
}
}while(option.equals("0"));

do{
    option = h.inputDate();

    if(option != null){

        date = option; // store the date from option to prevent overwrite
        while(!option.equals("0")){

            // update the total donation received of a specific date range
            updateTotalDonationDistribution(date, null);

            // sort the updated donee list by the total value they received
            tempDoneeList = bubbleSortDonee(sortBy, orderBy, doneeList);

            // display report
            totalValueReportDisplay(date, doneeType, tempDoneeList);

            System.out.println(" 'A' - Ascending, 'D' - Descending");
            System.out.print(" View report of other date? ( 'Y' - Yes, 'N' - No, "
                + "'0' to Back, 'X' to Subsystem Menu ): ");
            option = scan.nextLine();

            switch(option.toLowerCase()){

                case "y":
                    optionYN = "Y";
                    option = "0";
                    break;
                case "n":
                    option = "0";
                    optionYN = "N";
                    break;
                case "x":
                    return true; // Exit condition
                case "a":
                    orderBy = "Asc";
                    sortBy = "totalValue";
                    break;
                case "d":
                    orderBy = "Desc";
                    sortBy = "totalValue";
                    break;
            }
        }
    }
}

```

```

        case "0":
            optionYN = "Y";
            break;
        default:
    }
}// while

} else break;
}while (optionYN.equals("Y"));

if(optionYN.equals("N")) optionYN = h.inputYN(" Do you want to view report for other type of
donee?");

}while(optionYN.equals("Y"));

return false;

}

public static boolean top10ReceivedOfSelectedItem(){
    Item item = new Item();
    String date;
    int verify = 0, itemId = 0;
    boolean loop = true;

    while (loop == true) {
        // reset variable needed
        option = "0";
        loop = true;
        ListInterface<Item> templist = new DoublyLinkedList<>();
        doneeList = doneeListP.merge(doneeListO.merge(doneeListF));

        // User select item category
        while(option.equals("0")){
            DonationControl.categoryMenu();
            System.out.print("\n Select catogery of item that you want to generate report "
                + "('0' to back, 'X' to Exit to subsystem menu): ");
            option = scan.nextLine();

            switch (option.toLowerCase()) {
                case "0":
                    return false; // back conditon
                case "1":
                    templist = moneyList;
                    break;
                case "2":

```

```

        templItemList = foodBeverageList;
        break;
    case "X":
        System.err.println(exitStr); // Exit condition
        return true;
    default:
        option = "0";
        System.err.println(v.errCategoryNotFound);
    }
}

// User select item want
while(!option.equals("0")){

    optionYN = "N";

    DonationControl.donationItemList(templItemList);
    System.out.print("\n Select item for generate report ('0' to back, 'X' to Exit to subsystem
menu): ");
    option = scan.nextLine();

    switch (option.toLowerCase()) {
        case "0": // back condition
            break;
        case "X": // Exit condition
            System.err.println(exitStr);
            return true;
        default:
            verify = 0;
            verify = v.verifyIsDigit(option, 1, 4);

            // If option all is digit
            if(verify == 1) verify = v.isValidID(Integer.parseInt(option), templItemList.getLast().getId());

            // if option in a valid range of id
            if(verify == 1){
                item = new Item();
                item.setId(Integer.parseInt(option));
                item = templItemList.search(item);
                if(item == null) System.err.println(v.errItemNotFound);
                else {
                    optionYN = "Y";
                    option = "0";
                }
            }
        }
    }
}

```

```

// User select date of the report
orderBy = "Desc";
while (optionYN.equals("Y")){
    option = h.inputDate();

    if(option != null){
        date = option;
        while (!option.equals("0")) {

            sortBy = "itemValue";

            // Get and update the total amt and donation received of donee based on date and item
require
            updateTotalDonationDistribution(date, item);

            // sort the donee list based of the total item value of user received
            tempDoneeList = bubbleSortDonee(sortBy, orderBy, doneeList);

            // Display report
            top10ReceivedOfSelectedItemDisplay(date, item, tempDoneeList);

            System.out.println(" 'A' - Ascending, 'D' - Descending");
            System.out.print(" View report of other date? ( 'Y' - Yes, 'N' - No ): ");
            option = scan.nextLine();

            switch (option.toLowerCase()) {
                case "y":
                    optionYN = "Y"; // reset optionYN to loop back to reenter date of report
                    option = "0"; // set the option to exit the loop to display report nad get option
                    break;
                case "n":
                    loop = false;
                    optionYN = "N";
                    option = "0";
                case "a":
                    orderBy = "Asc";
                    break;
                case "d":
                    orderBy = "Desc";
                    break;
                default:
            }
        }// while
    } else optionYN = "N";
}

```

```

}

if(loop == false){
    optionYN = h.inputYN(" View report of other items?");
    if (optionYN.equals("N")) return true;
    loop = true;
}

}// while

return false;
}

// Generate the every month donation received of a donee
public static boolean doneeSummaryReport(){
    String month[] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
                      "Oct", "Nov", "Dec"};

    do{
        // Let user select donee type and get the list of the type
        doneeList = selectDoneeType();
        if(doneeList == null) return false; // Back Condition

        // Display the donee in list and let user to select
        displayDoneeListForGenerateReport(doneeList);
        doneeFound = selectDonee(doneeList); // Back Condition

        // When donee is found
        if(doneeFound != null){

            // Get the year to generate report
            String year = h.inputYear();
            if(year == null) return false; // Back condition

            // Get all the distribution received of the donee
            tempDistributionList = getSpecificDoneeDistribution(year, doneeFound);

            // Declare variable needs
            int max = 1, min = 1, count = 0;
            double maxValue = 0, minValue = 0;
            double[] total = {0,0,0};

            // Report Header
            System.out.format(
                "\n\n Summary Report of " + doneeFound.getIdStr() + " " + doneeFound.getName() +
                " in " + year + "\n" +
                " +-----+-----+-----+\n"
            );
        }
    }
}

```

```

" | %5s | %30s | %30s | %20s |\n" +
" +-----+-----+-----+-----+\n",
"Month", "Money Received (RM)", "Item Value Received (RM)", "Total (RM)"
);

// When there were donation received of donee in the year
if(tempDistributionList.getSize() > 0){

    for (int i = 1; i <= 12; i++) {

        // Get all the amount of distribution in month of the year
        String date = String.format("%s-%02d", year, i);
        double[] amount = calSpecificDoneeDonationDistribution(date, tempDistributionList);

        // When the total amount received in the month are not 0
        if (amount[2] > 0) {

            // Assign the first month that have received amount as minimum received
            if(count < 1) {
                min = i;
                minValue = amount[2];
            }

            // Report table body
            System.out.format(
                " | %5s | %30.2f | %30.2f | %20.2f |\n"
                + " +-----+-----+-----+-----+\n"
                + "+-----+\n",
                month[i], amount[0], amount[1], amount[2]
            );
        }

        // Update the month and value of the minimun and maximum donation received in a
month
        if (amount[2] > maxValue) {
            max = i;
            maxValue = amount[2];
        } else if (amount[2] < minValue && amount[2] <= 0) {
            min = i;
            minValue = amount[2];
        }
        count++;
        total[0] += amount[0]; // Total money received
        total[1] += amount[1]; // Total item value received
        total[2] += amount[2]; // Total received
    }

}

```

```

// Report footer and final summary
System.out.format(
    " %5s : %30.2f %30.2f %20.2f\n"
    + "\n\n Summary \n"
    + " =====\n"
    + " Total Received Month: %d\n"
    + " Average Amount Received per Month : RM %.2f\n"
    + " Maximum Donation Received: %-5s RM %.2f\n"
    + " Minimum Donation Received: %-5s RM %.2f\n\n",
    "Total", total[0], total[1], total[2],
    count, (double) total[2] / count,
    month[max], maxValue, month[min], minValue
);

}

// When no distribution of the donee found in the year
if (count <= 0) {
    System.out.format(
        " | %-94s |\n"
        + " +-----+"
        + "-----+\n",
        "No donation received in " + year
    );
}

System.out.println("\n ps: \"Month that did not received donation will not show\"");
optionYN = h.inputYN(" Do you want to generate another donee summary report?");
}
}while (optionYN.equals("Y"));

return false;
}

// Sort
public static ListInterface<Donee> sortDoneeMenu(ListInterface<Donee> list){

do{
    exitFlag = true; // reset exit flag
    orderBy = "Asc"; // reset order by
    System.out.format(
        "\n Sort By\n" +

```

```

    " %-20s %-20s\n" +
    " %-20s\n" +
    " Enter option ('0' - Back, 'X' - Exit to Subsystem Menu): ",
    "1. Donee ID", "2. Times of donation received",
    "3. Register Date"
);

option = scan.nextLine();

switch (option.toLowerCase()){
    case "1":
        sortBy = "id";
        sortByStr = "ID";
        break;
    case "2":
        sortBy = "donationReceived";
        sortByStr = "Number of Donation Received";
        break;
    case "3":
        sortBy = "registerDate";
        sortByStr = "Register Date";
        break;
    case "0": // back condition, back to previous step
        break;
    case "x":
        exitFlag = true;
        return list; // Exit condition
    default:
        System.err.println(v.errInput);
        exitFlag = false;
}
tempDoneeList = bubbleSortDonee(sortBy, orderBy, list); // sort the list and store the sorted list
}while (exitFlag == false);

exitFlag = false; // reset exit flag to prevent over exit
return tempDoneeList;
}

public static ListInterface<Donee> bubbleSortDonee(String sortBy, String orderBy,
ListInterface<Donee> list){
    Donee currentDonee = list.getFirst();
    Donee rightDonee = list.getNextData(currentDonee);
    Donee doneeBefore = null;

    int pass = list.getSize();
    boolean condition = false;
}

```

```

orderBy = orderBy.toLowerCase();
sortBy = sortBy.toLowerCase();

if(currentDonee != null && rightDonee != null){
    while (pass != 0) {
        currentDonee = list.getFirst();
        rightDonee = list.getNextData(currentDonee);

        for (int i = 0; i < pass - 1; i++) {
            condition = false; // reset condition

            switch (sortBy.toLowerCase()){
                case "id": // condition of id
                    if (currentDonee.getId() != rightDonee.getId()) {
                        switch (orderBy.toLowerCase()) {
                            case "asc":
                                condition = currentDonee.getId() > rightDonee.getId();
                                break;
                            case "desc":
                                condition = currentDonee.getId() < rightDonee.getId();
                                break;
                        } // switch
                    } // if
                    break;
                case "registerdate": // condition of register date
                    if (currentDonee.getRegisterDate() != rightDonee.getRegisterDate()) {
                        switch (orderBy.toLowerCase()) {
                            case "asc":
                                condition =
currentDonee.getRegisterDate().isAfter(rightDonee.getRegisterDate());
                                break;
                            case "desc":
                                condition =
currentDonee.getRegisterDate().isBefore(rightDonee.getRegisterDate());
                                break;
                        } // switch
                    } // if
                    break;
                case "itemvalue": // condition of total item value
                    if(currentDonee.getTotalItemValueReceived() !=
rightDonee.getTotalItemValueReceived()){
                        switch (orderBy.toLowerCase()) {
                            case "asc":
                                condition = currentDonee.getTotalItemValueReceived() >
rightDonee.getTotalItemValueReceived();
                                break;
                            case "desc":

```

```

        condition = currentDonee.getTotalItemValueReceived() <
rightDonee.getTotalItemValueReceived();
        break;
    }
}
break;
case "donationreceived": // condition of total number of donation received
    if(currentDonee.getTotalDonationReceived() !=
rightDonee.getTotalDonationReceived()){

        switch (orderBy.toLowerCase()) {

            case "asc":
                condition = currentDonee.getTotalDonationReceived() >
rightDonee.getTotalDonationReceived();
                break;
            case "desc":
                condition = currentDonee.getTotalDonationReceived() <
rightDonee.getTotalDonationReceived();
                break;
            } // switch
        }
        break;
case "totalvalue": // condition of total value of all donation received
    double[] total = {0,0};
    total[0] = currentDonee.getTotalItemValueReceived() +
currentDonee.getTotalMoneyReceived();
    total[1] = rightDonee.getTotalItemValueReceived() +
rightDonee.getTotalMoneyReceived();

    if(total[0] != total[1]){
        switch (orderBy.toLowerCase()) {
            case "asc": condition = total[0] > total[1];
            break;
            case "desc": condition = total[0] < total[1];
            break;
        } // switch
    }
    break;
} // switch

if (condition) {
    // when the conditon are true, swap the data of the two node
    doneeBefore = rightDonee;
    rightDonee = list.getNextData(rightDonee); // Get the next data first to avoid override
    list.swap(currentDonee, doneeBefore);

} else { // else just go to the next data
}

```

```

        currentDonee = rightDonee;
        rightDonee = list.getNextData(rightDonee);
    }

} // for

pass--; // minus out the index that already done sort

}// while
}

return list;
}

// Additional function
public static Donee selectDonee(ListInterface<Donee> list) {
    String digitIdStr = null;
    Donee doneeFound = new Donee(), tempDonee = new Donee();

    do {
        exitFlag = false; // Reset the exit flag

        digitIdStr = h.inputIdByStr(list.getLast().getId());

        if (digitIdStr != null) {
            tempDonee.setSearchBy(""); // set the searchBy in entity to let adt search the adt can run
compareTo
            tempDonee.setId(Integer.parseInt(digitIdStr));
            doneeFound = list.search(tempDonee); // search the donee in list

            if (doneeFound != null) return doneeFound;

            System.err.println(v.errDoneeNotFound);

        } else {
            option = "0";
            break;
        } // Exit condition
    } while (exitFlag == false);

    return null;
}

public static ListInterface<Donee> selectDoneeType(){
    do {

```

```

// Get the type of donee for generate report
System.out.println("\n\n" + doneeTypeMenuDisplay());
System.out.print(" Select type of donee for generate report. ('0' - Back): ");
option = scan.nextLine();

switch (option.toLowerCase()) {
    case "p":
        doneeList = doneeListP;
        break;
    case "o":
        doneeList = doneeListO;
        break;
    case "f":
        doneeList = doneeListF;
        break;
    case "0":
        return null;
    default:
        System.err.println(v.errType);
        option = "0";
}
} while (option.equals("0"));

return doneeList;
}

// Update all donee total donation received in overall or in date range
public static void updateTotalDonationDistribution(String date, Item item) {

    // reset the variable needed
    ListInterface<InventoryMovement> stockOutList = new DoublyLinkedList<>();
    doneeListP = Initializer.doneeListP;
    doneeListO = Initializer.doneeListO;
    doneeListF = Initializer.doneeListF;
    distributionList = Initializer.distributionList;

    Item templItem = new Item();
    InventoryMovement templItemMovement = new InventoryMovement(null, 0, 0);
    tempDistributionList = distributionList;
    int loopTime = 0; // count the time loop for identify update different donee list data

    while(loopTime < 3){

        // Identify which doneeList are updating
        if(loopTime == 0) tempDonee = doneeListP.getFirst();
        else if (loopTime == 1) tempDonee = doneeListO.getFirst();

```

```

else if (loopTime == 2) tempDonee = doneeListF.getFirst();

while (tempDonee != null) {
    int count = 0;
    double[] amount = {0, 0}, received = {0, 0};

    // Direct get the specific donee distributionList first
    tempDistributionList = getSpecificDoneeDistribution(date, tempDonee);

    // Check is it the donee have received any distribution, else direct amount to 0
    if(tempDistributionList.getSize() > 0){
        if (date != null && item != null) {

            DonationDistribution distribution = tempDistributionList.getFirst();

            while (distribution != null) {
                if (distribution.getDate().toString().contains(date)) {

                    templItemMovement = distribution.getMovementItems().getFirst();
                    if (templItemMovement != null) {
                        templItem = templItemMovement.getItem();
                    }

                    // Compare the is item that want to find is exist in the inventory movement of the
                    distribution
                    while (templItemMovement != null) {
                        if (templItem.compareTo(item) == 0) {
                            count++; // number of donation received that contain selected item in specific
date range

                            received = calAmtPerItem(templItemMovement);
                            amount[0] += received[0]; // money amount
                            amount[1] += received[1]; // amt of item value

                            break;
                        }

                        templItemMovement =
distribution.getMovementItems().getNextData(templItemMovement);
                        if (templItemMovement != null) {
                            templItem = templItemMovement.getItem();
                        }
                    } // while
                } // if

                distribution = tempDistributionList.getNextData(distribution);
            } // while
        }
    }
}

```

```

} else if(date != null){
    count++;
    amount = calSpecificDoneeDonationDistribution(date, tempDistributionList);
}
else {
    count++;
    amount = calSpecificDoneeDonationDistribution(null, tempDistributionList);
}
}

tempDonee.setTotalDonationReceived(count);
tempDonee.setTotalMoneyReceived(amount[0]);
tempDonee.setTotalItemValueReceived(amount[1]);

if(loopTime == 0) {
    doneeListP.update(tempDonee);
    tempDonee = doneeListP.getNextData(tempDonee);
}
else if (loopTime == 1) {
    doneeListO.update(tempDonee);
    tempDonee = doneeListO.getNextData(tempDonee);
}
else if (loopTime == 2) {
    doneeListF.update(tempDonee);
    tempDonee = doneeListF.getNextData(tempDonee);
}

} // while tempDonee != null

loopTime++; // count the time loop for identify update different donee list data
}

}

// Update specific donee total donation received
public static double[] calSpecificDoneeDonationDistribution(String date,
ListInterface<DonationDistribution> distributionList){

int count = 0;
double[] amount = {0, 0, 0}, received = {0,0};
distribution = distributionList.getFirst(); // Reset the distribution for run again the second loop

while(distribution != null){
    if (date != null) { // For report total value
        if (distribution.getDate().toString().contains(date)) {
            // Get total times of donation received
            count++;
        }
    }
}
}

```

```

// Get total amount of donation received
received = calAmtReceived(distribution);

}

} else {
    // Get total times of donation received
    count++;
    // Get total amount of donation received
    received = calAmtReceived(distribution);
}

amount[0] += received[0]; // totalMoneyAmount
amount[1] += received[1]; // totalItemValue
distribution = distributionList.getNextData(distribution);
} // while

amount[2] = amount[0] + amount[1];
return amount;
}

// Get the donation distribution of a specific donee
public static ListInterface<DonationDistribution> getSpecificDoneeDistribution(String date, Donee donee) {
    ListInterface<DonationDistribution> tempDistributionList = new DoublyLinkedList<>();
    distributionList = Initializer.distributionList;

    DonationDistribution tempDistribution = distributionList.getFirst();

    while (tempDistribution != null) {
        if (tempDistribution.getDonee().compareTo(donee) == 0) {

            if(date != null){
                if (tempDistribution.getDate().toString().contains(date)) {
                    tempDistributionList.add(tempDistribution);
                }
            } else {
                tempDistributionList.add(tempDistribution);
            }
        }

        tempDistribution = distributionList.getNextData(tempDistribution);
    }

    return tempDistributionList;
}

```

```

// Get the donation distribution tha contain selected item of a specific donee
public static ListInterface<DonationDistribution> getSelectedItemOccurInDistribution(Item item,
Donee donee){
    ListInterface<DonationDistribution> resultList = new DoublyLinkedList<>();
    tempDistributionList = distributionList;
    Item templItem = new Item();
    InventoryMovement templItemMovement = null;

    distribution = tempDistributionList.getFirst();

    while(distribution != null){
        if(tempDonee.getId() == distribution.getDonee().getId()){
            templItemMovement = distribution.getMovementItems().getFirst();
            templItem = templItemMovement.getItem();

            // Compare the is item that want to find is exist in the invetory movement of the distribution
            while(templItem != null){
                stockOutList = new DoublyLinkedList<>(); // Reset the list for prevent null pointer

                // direct record the inventory movement that match the item selected and break to
                // improve efficient
                if(templItem.compareTo(item) == 0){
                    tempDistribution.setId(distribution.getId());
                    stockOutList.add(templItemMovement);
                    tempDistribution.setMovementItems(stockOutList);

                    resultList.add(tempDistribution);
                    break;
                }

                templItemMovement =
distribution.getMovementItems().getNextData(templItemMovement);
                if(templItemMovement != null) templItem = templItemMovement.getItem();
                else templItem = null;
            } // while
        } //if
        distribution = tempDistributionList.getNextData(distribution);
    }

    return resultList;
}

// Calculate totalAmountReceived of an distribution
private static double[] calAmtReceived(DonationDistribution distribution){
    double[] amount = {0, 0}, received;
    InventoryMovement movementItem = null;
    ListInterface<InventoryMovement> stockOutList = new DoublyLinkedList<>();
}

```

```

// Get total amount of donation received
stockOutList = distribution.getMovementItems();

movementItem = stockOutList.getFirst();
while (movementItem != null) {

    received = calAmtPerItem(movementItem);
    amount[0] += received[0]; // total money amt
    amount[1] += received[1]; // total item value amt

    movementItem = stockOutList.getNextData(movementItem);
} // while item != null

return amount;
}

// Calculate amount per item
private static double[] calAmtPerItem(InventoryMovement movementItem){
    double[] amount = {0, 0};

    if(movementItem.getItemQuantity() <= 0) amount[0] = movementItem.getMoneyAmount();
    else amount[1] += (double) movementItem.getItemQuantity() *
movementItem.getMoneyAmount();

    return amount;
}

//Display
private static String doneeMenuDisplay(){

    return String.format(
        "\n\tDonee Subsystem\n\n"
        + " +-----+-----+\n"
        + " | %-35s |   %-7s  |\n"
        + " +-----+-----+\n"
        + " | %-35s | \t1\t|\n"
        + " +-----+-----+\n"
        + " | %-35s | \t2\t|\n"
        + " +-----+-----+\n"
        + " | %-35s | \t3\t|\n"
        + " +-----+-----+\n"
        + " | %-35s | \t0\t|\n"
        + " +-----+-----+\n"
        + " Please select option (0 - 3): ",

```

```

    "Function", "Option",
    "Add new donee",
    "View & Update donee",
    "Donee Summary Report",
    "Exit");

}

private static String reportMenuDisplay(){
    return String.format(
        "\n\tDonee Report\n\n"
        + " +-----+-----+\n"
        + " | %-35s | %7s | \n"
        + " +-----+-----+\n"
        + " | %-35s | t1| \n"
        + " +-----+-----+\n"
        + " | %-35s | t2| \n"
        + " +-----+-----+\n"
        + " | %-35s | t3| \n"
        + " +-----+-----+\n"
        + " | %-35s | t0| \n"
        + " +-----+-----+\n"
        + " Please select option (0 - 3): ",

        "Report", "Option",
        "Total Value Donee Received Report",
        "Top 10 Donee Received Selected Item",
        "Specific Donee Summary Report",
        "Exit");
}

private static String doneeTypeMenuDisplay(){
    return String.format(
        " +-----+\n"
        " |t\tDonee Type %16s |\n" +
        " +-----+\n"
        " |tP| %25s |\n" +
        " |tO| %25s |\n" +
        " |tF| %25s |\n" +
        " +-----+\n",
        "", "Individual", "Organization", "Family"
    );
}

private static void displayDoneeListForGenerateReport(ListInterface<Donee> doneeList){
    System.out.format(

```

```

    "
    _____ \n"
+ " | | | | |
+ " | %8s | %20s | %20s | %20s |\n"
+ "
|_____ |_____ |_____ |_____ |\n"
,
    "Donee ID", "Name", "Phone Number", "Email"
);

tempDonee = doneeList.getFirst();
while (tempDonee != null){
    System.out.format(
        " | %8s | %20s | %20s | %20s |\n"
        +
        "|_____ |_____ |_____ |_____ |\n"
),
    tempDonee.getIdStr(), tempDonee.getName(), tempDonee.getPhoneNumber(),
tempDonee.getEmail()
);

tempDonee = doneeList.getNextData(tempDonee);
}
}

private static String filterDisplay(){

return String.format(
    " %-10s%-20s %30s: %5s %5s\n" +
    " %-10s: %15s -> %-15s %18s -> %-15s %15s -> %-15s \n" +
    " %11s %15s -> %-15s %18s -> %-15s\n",
    "Search ", searchByStr == null ? ": None" : searchByStr,
    "Sort By", sortByStr == null ? "None": sortByStr, orderBy, // row 1
    "Filter", "Gender", filterByGender == null ? "All" :
Person.getGenderStr(filterByGender.charAt(0)), // row 2
    "Type", filterByType == null ? "All" : Person.getTypeStr(filterByType.charAt(0)),
    "State", filterByState == null ? "All" : filterByState, // row 2
    "", "Register Date", filterByRegisterDate == null ? "All" : filterByRegisterDate,
    "Donation Received",
    filterByRangeMin == -1 && filterByRangeMax == -1 ? "All" : filterByRangeMin + " - " +
filterByRangeMax // row 3
);
}

private static void doneeDistributionDisplay(ListInterface<DonationDistribution> list, Donee donee){
    System.out.format(
        "\n\n Distribution of Donee " + donee.getIdStr() + "\n" +

```

```

" +-----+-----+-----+
" +-----+\n" +
" | %7s | %30s | %20s | %20s | %20s | \n" +
" +-----+-----+-----+-----"
" +-----+\n",
"ID", "Category Item Received", "Value(RM)", "Distribute Date", "Status"
);

if(list.getSize() > 0){
    distribution = list.getFirst();
    double[] received = {0,0};
    String [] category = {"Money", "Food & Beverage"};

    while(distribution != null){
        received = calAmtReceived(distribution);
        int money = 0, fb = 0;
        InventoryMovement tempMoveItem = new InventoryMovement(null, 0,0);
        stockOutList = distribution.getMovementItems();
        tempMoveItem = stockOutList.getFirst();

        while(tempMoveItem != null){
            // times of the food and beverage item exit in the distribution
            if(moneyList.search(tempMoveItem.getItem()) != null) money++;
            // times of the food and beverage item exit in the distribution
            else if(foodBeverageList.search(tempMoveItem.getItem()) != null) fb++;

            tempMoveItem = stockOutList.getNextData(tempMoveItem);
        } // while

        if( (money + fb) >= 2){ // if both category item also exit at least 1 time
            for (int i = 0; i < 2; i++) { // display the two category in the list
                if (i == 0) {
                    System.out.format(
                        " | %7s | %30s | %20.2f | %20s | %20s |\n",
                        distribution.getIdStr(), category[i], received[0] + received[1],
                        distribution.getDate().toString(), distribution.getCurrentTrackingStatus()
                    );
                } else System.out.format(" | %7s | %30s | %20s | %20s | %20s |\n", "", category[i], "",
                "", "");
            } // for
        } else {
            System.out.format(
                " | %7s | %30s | %20.2f | %20s | %20s |\n",
                distribution.getIdStr(), money >= 1 ? category[0] : category[1], received[0] +
received[1],
                distribution.getDate().toString(), distribution.getCurrentTrackingStatus())
        }
    }
}

```

```

    );
}

System.out.print(" +-----+-----+-----+-----" +
    "+-----+\n");
distribution = list.getNextData(distribution);
}
} else System.out.print(" No record(s) found\n");

System.out.format(
"\n" +
" %-10s (RM): %6.2f %10s %20s: %-12d\n",
"Total Amount", donee.getTotalItemValueReceived() + donee.getTotalMoneyReceived(), "",
"Total Received", donee.getTotalDonationReceived()
);

}

private static void totalValueReportDisplay(String date, char doneeType, ListInterface<Donee> list){
double[] total = {0,0,0};

System.out.format(
"\n\n %-50s %32s: %10s\n" +
" +-----+-----+-----+-----+-----+-----+\n" +
" | %6s | %15s | %20s | %20s | %20s | \n" +
" +-----+-----+-----+-----+-----+\n",
"Total Value of " + Donee.getTypeStr(doneeType) + " Donee Received in " + date, "Order By",
orderBy,
"ID", "Name", "Item Value (RM)", "Money Value (RM)", "Total (RM)"
);

if(list.getSize() > 0){
tempDonee = list.getFirst();
while(tempDonee != null){
System.out.format(
" | %6s | %15s | %20.2f | %20.2f | %20.2f |\n" +
" +-----+-----+-----+-----+-----+\n",
tempDonee.getIdStr(), tempDonee.getName(),
tempDonee.getTotalItemValueReceived(),
tempDonee.getTotalMoneyReceived(),
tempDonee.getTotalItemValueReceived() + tempDonee.getTotalMoneyReceived()
);

total[0] += tempDonee.getTotalItemValueReceived();
total[1] += tempDonee.getTotalMoneyReceived();
total[2] += tempDonee.getTotalItemValueReceived() + tempDonee.getTotalMoneyReceived();
}
}
}

```

```

        tempDonee = list.getNextData(tempDonee);
    } // while
} else System.out.print(" 0 record(s) found\n");

System.out.format(
    " | %24s | %20.2f | %20.2f | %20.2f |\n" +
    " +-----+-----+-----+-----+\n" +
    " %d Result(s)\n\n",
    "Total (RM)", total[0], total[1], total[2], list.getSize()
);

}

private static void top10ReceivedOfSelectedItemDisplay(String date, Item item,
ListInterface<Donee> doneeList){
    double total = 0;
    int qty = 0, count = 0;
    InventoryMovement templItem = new InventoryMovement(null, 0, 0);

    System.out.format(
        "\n\n %50s %25s: %10s\n"
        + " +-----+-----+-----+-----+-----+\n" +
        " | %10s | %15s | %20s | %15s | %15s | \n"
        + " +-----+-----+-----+-----+-----+\n",
        "Top 10 Donee Received " + item.getName() + " in " + date, "Order By", orderBy,
        "Donee ID", "Name", "Distribution ID", "Received(Qty)", "Value (RM)"
    );
}

tempDonee = doneeList.getFirst();
count = 0;
while(tempDonee != null){

    /* the total received time of selected item of a donee already be filter before come to display,
       so just ignore the donee that received 0 times */
    if(tempDonee.getTotalDonationReceived() > 0){

        // Get the distribution list that have the selected item with the sorted sequence
        tempDistributionList = getSelectedItemOccurInDistribution(item, tempDonee);

        if(tempDistributionList.getSize() > 0){
            distribution = tempDistributionList.getFirst();

            int loop = 0;
            while(distribution != null){

                // direct display the first item of the distribution as the list already filter
                templItem = distribution.getMovementItems().getFirst();

```

```

// if is the row if the first row of donee, will print the name and id else will be null
System.out.format(
    " | %10s | %15s | %20s | %15d |",
    loop == 0 ? tempDonee.getIdStr() : "", loop == 0 ? tempDonee.getName() : "",
    distribution.getIdStr(), templItem.getItemQuantity()
);

// when the item is not money
if(moneyList.search(item) == null) {
    System.out.format(" %15.2f |\n", templItem.getMoneyAmount() *
templItem.getItemQuantity());
    total += templItem.getMoneyAmount() * templItem.getItemQuantity(); // Final total
item value received
} else {
    System.out.format(" %15.2f |\n", templItem.getMoneyAmount() );
    total += templItem.getMoneyAmount(); // Final total item value received
}

qty += templItem.getItemQuantity(); // Final total item qty received
loop++;
count++; // Total times of all the donee received the selected item
distribution = tempDistributionList.getNextData(distribution);
} // while

System.out.print(" +-----+-----+-----+-----+
" +-----+-----+\n");
} // if
}
if(count == 10) break; // as a top 10 report, just loop for first 10 donee base on the sorted list
tempDonee = doneeList.getNextData(tempDonee);
} // while

if(count <= 0){
    System.out.format(
        " | %-87s |\n" +
        " +-----+-----+-----+-----" +
        " +-----+-----+\n",
        " No record(s) found"
    );
}

System.out.format(
    " | %28s | %20d | %15d | %15.2f |\n" +
    " +-----+-----+-----+" +
    " +-----+-----+\n",
    "Total (RM)", count, qty, total
)

```

```

    );
}

public static String doneePaginatedListDisplay(PaginationResult paginatedResult, int originalListSize)
{
    return String.format(
        "
        + " _____ \n"
        + " | | | | | | "
        + " | \n"
        + " | %8s | %20s | %20s | %20s | %20s | %20s | %20s | \n"
        + "
        |_____|_____|_____|_____|_
        + " _____ | \n"
        + " | | | | | | "
        + " | \n"
        + "%s"
        +
        |_____|_____|_____|_____|_
        + " _____ | \n"
        + " Page %d of %d %4d of %4d Records\n\n"
        + " %-3s - %-15s %-3s - %-15s %-3s - %-15s %-3s - %-15s\n"
        + " %-3s - %-15s %-3s - %-15s %-3s - %-15s %-3s - %-15s %-3s - %-15s\n",
        "Donee ID", "Name", "Ic Number", "Phone Number", "Email", "Register Date", "Donation
Received",

        paginatedResult.getPaginatedList().getSize() > 0 ?
        paginatedResult.getPaginatedList().toString() : " | No Record(s) found      " +
        "\n",
        paginatedResult.getCurrentPage(), paginatedResult.getTotalPages(),
        paginatedResult.getRecordCount(), originalListSize,
        "'P'", "Previous page", "'F'", "Filter", "'S'", "Sort",
        "'A'", "Ascending", "'M'", "View / Update / Remove donee details",
        "'N'", "Next page", "'C'", "Clear filter", "'H'", "Search",
        "'D'", "Descending", "'L'", "Donation Distribute List of donee"
    );
}

// reset all the filter, sort and search
private static void clearFilter(){
    filterByGender = null;
}

```

```

filterByType = null;
filterByState = null;
filterByRegisterDate = null;
filterByRangeMin = -1;
filterByRangeMax = -1;
searchBy = "";
keyword = null;
searchByStr = "None";
sortBy = "id";
orderBy = "Asc";
sortByStr = "ID";
}
}

}

```

2. Screenshot

Donee Subsystem Menu

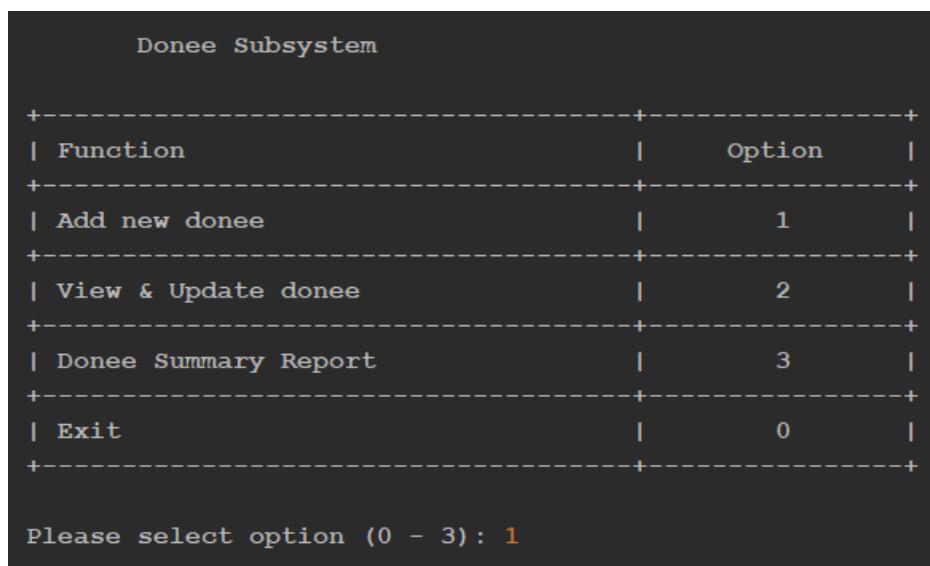


Figure 1: Menu of donee subsystem

1.1 Create a new donee

New Donee Detail

Name ('0' - Back): **jane lim**

IC Number (Eg. 123456121234; '0' - Back: **123456789012**

Gender ('F' - Female, 'M' - Male; '0' - Back): **f**

Phone Number (Eg. 01XXXXXXXX; '0' - Back): **01222334455**

Email (Eg. example@mail.com; '0' - Back): **2kdjds**

"Invalid format. Please enter follow format 'example@mail.com'."

Email (Eg. example@mail.com; '0' - Back): **a@gmail.com**

Type ('P' - Individual, 'O' - Organization, 'F' - Family; '0' - Back): **o**

Street Address (Eg. No, Street; '0' - Back): **1, Jalan Klang, Jane**

City ('0' - Back): **Kepong**

Postcode (Eg. '12345'; '0' - Back): **12345**

| | | |
|-----------------------|--|--------------------|
| JHR - Johor | | KDH - Kedah |
| KTN - Kelantan | | MLK - Melaka |
| NSN - Negeri Sembilan | | PHG - Pahang |
| PNG - Pulau Pinang | | PRK - Perak |
| PLS - Perlis | | SBH - Sabah |
| SWK - Sarawak | | SGR - Selangor |
| TRG - Terengganu | | KUL - Kuala Lumpur |
| LBN - Labuan | | PJY - Putrajaya |

State (Eg. 'JHR'; '0' - Back): **kul**

Figure 1.1.1: Input page of user add new donee.

```

+-----+
| DE0031 - Jane Lim | Register : 2024-08-17 |
+-----+
| Personal Detail: |
| IC Number : 123456789012 |
| Gender : Female |
| Phone Number : 01222334455 |
| Email : a@gmail.com |
| Type : Organization |
+-----+
| 1, Jalan Klang, Jane | Total Received: |
| Kepong, 12345 | Times : 0 |
| Kuala Lumpur | Item Value : 0.00 |
| | Money : 0.00 |
| |-----+ |
| | Total Value : 0.00 |
+-----+
Do you confirm to add this donee? ('Y' - Yes; 'N' - No): y
Donee add successful.

Do you want to add another donee? ('Y' - Yes; 'N' - No):

```

Figure 1.1.2: Confirmation of user add new donee

1.2 View and Update Donee

| Search Filter | None | Sort By: | | ID Asc | Type -> All | State -> All |
|---------------|--------|--------------|--------------|-----------------|---------------|-------------------|
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 |

Page 1 of 5 7 of 30 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back):

Figure 1.2.1.1: Display list of donee

1.2.2 Previous and Next Page of Donee List

| Search : None | | Sort By: ID Asc | | State -> All | | | |
|------------------------|--------|-----------------|--------------------------|-----------------|---------------|-------------------|--|
| Filter : Gender -> All | | Type -> All | Donation Received -> All | | | | |
| Register Date -> All | | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | |

Page 1 of 5 7 of 29 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back): p

You already at first page. Now redirecting you to last page."

| Search : None | | Sort By: ID Asc | | State -> All | | | |
|------------------------|------|-----------------|--------------------------|---------------|---------------|-------------------|--|
| Filter : Gender -> All | | Type -> All | Donation Received -> All | | | | |
| Register Date -> All | | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0030 | Hadi | 894567890123 | 01123456785 | hadi@mail.com | 2021-11-22 | 1 | |

Page 5 of 5 1 of 29 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back):

Figure 1.2.2.1: Previous page of list displaying

| Search : None | | Sort By: ID Asc | | State -> All | | | |
|------------------------|------|-----------------|--------------------------|---------------|---------------|-------------------|--|
| Filter : Gender -> All | | Type -> All | Donation Received -> All | | | | |
| Register Date -> All | | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0030 | Hadi | 894567890123 | 01123456785 | hadi@mail.com | 2021-11-22 | 1 | |

Page 5 of 5 1 of 29 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back): n

You already at last page. Now redirecting you to first page."

| Search : None | | Sort By: ID Asc | | State -> All | | | |
|------------------------|--------|-----------------|--------------------------|-----------------|---------------|-------------------|--|
| Filter : Gender -> All | | Type -> All | Donation Received -> All | | | | |
| Register Date -> All | | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | |

Page 1 of 5 7 of 29 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back):

Figure 1.2.2.2: Next page of list displaying

1.2.3 Filter Donee List

| Select filter | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------|----------------------|--------------|--------------------------|---------------|-------------------|--|----------|------|-----------|--------------|-------|---------------|-------------------|--|--------|-----|--------------|-------------|--------------|------------|---|--|--------|--------|--------------|-------------|-----------------|------------|---|--|--------|-------|--------------|-------------|----------------|------------|---|--|--------|-------|--------------|-------------|----------------|------------|---|--|--------|--------|--------------|-------------|-----------------|------------|---|--|--------|------|--------------|-------------|---------------|------------|---|--|--------|-------|--------------|-------------|----------------|------------|---|--|
| 1. Gender | | 2. Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. State | | 4. Register Date | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5. Num of Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter donee by? ('0' - Back, 'X' - Exit to Subsystem menu): 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Gender ('F' - Female, 'M' - Male; '0' - Back): m | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search | None | Gender -> Male | | Sort By: | ID Asc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter | : | Register Date -> All | | Type | -> All | State -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th><th></th></tr> </thead> <tbody> <tr><td>DE0001</td><td>Jay</td><td>123456789012</td><td>01123456789</td><td>jay@mail.com</td><td>2023-01-10</td><td>1</td><td></td></tr> <tr><td>DE0004</td><td>Hassan</td><td>901234567890</td><td>01123456781</td><td>hassan@mail.com</td><td>2024-03-07</td><td>0</td><td></td></tr> <tr><td>DE0006</td><td>Rahim</td><td>112345678901</td><td>01123456770</td><td>rahim@mail.com</td><td>2023-07-15</td><td>0</td><td></td></tr> <tr><td>DE0008</td><td>Azman</td><td>312345678901</td><td>01123456772</td><td>azman@mail.com</td><td>2021-11-12</td><td>0</td><td></td></tr> <tr><td>DE0010</td><td>Ismail</td><td>512345678901</td><td>01123456774</td><td>ismail@mail.com</td><td>2022-05-18</td><td>0</td><td></td></tr> <tr><td>DE0012</td><td>Ravi</td><td>567890123456</td><td>01123456785</td><td>ravi@mail.com</td><td>2023-10-24</td><td>0</td><td></td></tr> <tr><td>DE0015</td><td>Chong</td><td>034567890123</td><td>01123456779</td><td>chong@mail.com</td><td>2023-02-10</td><td>0</td><td></td></tr> </tbody> </table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | | DE0010 | Ismail | 512345678901 | 01123456774 | ismail@mail.com | 2022-05-18 | 0 | | DE0012 | Ravi | 567890123456 | 01123456785 | ravi@mail.com | 2023-10-24 | 0 | | DE0015 | Chong | 034567890123 | 01123456779 | chong@mail.com | 2023-02-10 | 0 | |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0010 | Ismail | 512345678901 | 01123456774 | ismail@mail.com | 2022-05-18 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0012 | Ravi | 567890123456 | 01123456785 | ravi@mail.com | 2023-10-24 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0015 | Chong | 034567890123 | 01123456779 | chong@mail.com | 2023-02-10 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 2 7 of 13 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page | | 'F' - Filter | | 'S' - Sort | | 'A' - Ascending | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'N' - Next page | | 'C' - Clear filter | | 'H' - Search | | 'D' - Descending | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'M' - View / Update / Remove donee details | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'L' - Donation Distribute List of donee | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Option ('0' - Back): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.3.1: Filter donee by gender and display the list

| Select filter | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------|----------------------|--------------|--------------------------|---------------|-------------------|--|----------|------|-----------|--------------|-------|---------------|-------------------|--|--------|-----|--------------|-------------|--------------|------------|---|--|--------|------|--------------|-------------|---------------|------------|---|--|--------|--------|--------------|-------------|-----------------|------------|---|--|--------|--------|--------------|-------------|-----------------|------------|---|--|--------|------|--------------|-------------|---------------|------------|---|--|--------|-------|--------------|-------------|----------------|------------|---|--|--------|------|--------------|-------------|---------------|------------|---|--|
| 1. Gender | | 2. Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. State | | 4. Register Date | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5. Num of Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter donee by? ('0' - Back, 'X' - Exit to Subsystem menu): 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Type ('P' - Individual, 'O' - Organization, 'F' - Family; '0' - Back): p | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search | None | Gender -> All | | Sort By: | ID Asc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter | : | Register Date -> All | | Type | -> Individual | State -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th><th></th></tr> </thead> <tbody> <tr><td>DE0001</td><td>Jay</td><td>123456789012</td><td>01123456789</td><td>jay@mail.com</td><td>2023-01-10</td><td>1</td><td></td></tr> <tr><td>DE0002</td><td>Siti</td><td>456789012345</td><td>01123456786</td><td>siti@mail.com</td><td>2024-04-11</td><td>0</td><td></td></tr> <tr><td>DE0003</td><td>Aishah</td><td>678901234567</td><td>01123456784</td><td>aishah@mail.com</td><td>2023-08-31</td><td>0</td><td></td></tr> <tr><td>DE0004</td><td>Hassan</td><td>901234567890</td><td>01123456781</td><td>hassan@mail.com</td><td>2024-03-07</td><td>0</td><td></td></tr> <tr><td>DE0005</td><td>Nina</td><td>012345678901</td><td>01123456780</td><td>nina@mail.com</td><td>2024-03-08</td><td>0</td><td></td></tr> <tr><td>DE0006</td><td>Rahim</td><td>112345678901</td><td>01123456770</td><td>rahim@mail.com</td><td>2023-07-15</td><td>0</td><td></td></tr> <tr><td>DE0007</td><td>Lina</td><td>212345678901</td><td>01123456771</td><td>lina@mail.com</td><td>2022-02-05</td><td>0</td><td></td></tr> </tbody> </table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | | DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | | DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | | DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 2 7 of 10 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page | | 'F' - Filter | | 'S' - Sort | | 'A' - Ascending | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'N' - Next page | | 'C' - Clear filter | | 'H' - Search | | 'D' - Descending | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'M' - View / Update / Remove donee details | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'L' - Donation Distribute List of donee | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Option ('0' - Back): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.3.2: Filter donee by type and display the list

| Select filter | | | | | | |
|--|--------------------------|--------------|--------------|---------------|---------------|-------------------|
| 1. Gender | 2. Type | | | | | |
| 3. State | 4. Register Date | | | | | |
| 5. Num of Donation Received | | | | | | |
| Filter donee by? ('0' - Back, 'X' - Exit to Subsystem menu): 3 | | | | | | |
| JHR - Johor | KDH - Kedah | | | | | |
| KW'L - Kelantan | MLK - Melaka | | | | | |
| NSN - Negeri Sembilan | PHG - Pahang | | | | | |
| PNG - Pulau Pinang | PRK - Perak | | | | | |
| PLS - Perlis | SBH - Sabah | | | | | |
| SWK - Sarawak | SGR - Selangor | | | | | |
| TRG - Terengganu | KUL - Kuala Lumpur | | | | | |
| LBN - Labuan | PJY - Putrajaya | | | | | |
| State (Eg. 'JHR'; '0' - Back): jhr | | | | | | |
| Search None | Sort By: ID Asc | | | | | |
| Filter : Gender -> All | Type -> All | | | | | |
| Register Date -> All | State -> JHR | | | | | |
| | Donation Received -> All | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 |
| DE0014 | Ling | 923456789012 | 01123456790 | ling@mail.com | 2022-07-29 | 0 |
| DE0027 | Zara | 591234567890 | 01123456788 | zara@mail.com | 2022-09-30 | 0 |

Figure 1.2.3.3: Filter donee by state and display the list

| Select filter | | | | | | |
|---|--------------------------|--------------|--------------|---------------|---------------|-------------------|
| 1. Gender | 2. Type | | | | | |
| 3. State | 4. Register Date | | | | | |
| 5. Num of Donation Received | | | | | | |
| Filter donee by? ('0' - Back, 'X' - Exit to Subsystem menu): 4 | | | | | | |
| "Eg. '2023-08-00' - Report of August 2023, '2024-00-00' - Report of 2024" | | | | | | |
| Enter a date (yyyy-MM-dd; '0' - Back): 2022-00-00 | | | | | | |
| Search None | Sort By: ID Asc | | | | | |
| Filter : Gender -> All | Type -> All | | | | | |
| Register Date -> 2022 | State -> JHR | | | | | |
| | Donation Received -> All | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 |
| DE0014 | Ling | 923456789012 | 01123456790 | ling@mail.com | 2022-07-29 | 0 |
| DE0027 | Zara | 591234567890 | 01123456788 | zara@mail.com | 2022-09-30 | 0 |

Figure 1.2.3.4: Filter donee by register date and display the list

| Select filter | | | | | | | |
|--|----------------------|----------------------|-----------------------------|--|---------------|-------------------|--|
| 1. Gender | 2. Type | | | | | | |
| 3. State | 4. Register Date | | | | | | |
| 5. Num of Donation Received | | | | | | | |
| Filter donee by? ('0' - Back, 'X' - Exit to Subsystem menu): | 5 | | | | | | |
| Enter the minimum of the range you want (0 - 9999; '-1' - Back): | | | | | | | |
| Enter the maximum of the range you want (0 - 9999; '-1' - Back): | | | | | | | |
| Search | None | Sort By: | Number of Donation Received | Asc | | | |
| Filter : | Gender -> All | Type -> | All | | State -> All | | |
| | Register Date -> All | Donation Received -> | 1 - 1 | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0030 | Hadi | 894567890123 | 01123456785 | hadi@mail.com | 2021-11-22 | 1 | |
| Page 1 of 1 2 of 2 Records | | | | | | | |
| 'P' - Previous page | 'F' - Filter | 'S' - Sort | 'A' - Ascending | 'M' - View / Update / Remove donee details | | | |
| 'N' - Next page | 'C' - Clear filter | 'H' - Search | 'D' - Descending | 'L' - Donation Distribute List of donee | | | |
| Option ('0' - Back): | | | | | | | |

Figure 1.2.3.5: Filter donee by range of number donation received

| Sort By: ID Asc | | | | | | | |
|-------------------------------|-----------------------|----------------------|------------------|--|---------------|-------------------|--|
| Search | None | Sort By: | ID Asc | | | | |
| Filter : | Gender -> Male | Type -> | Individual | | State -> All | | |
| | Register Date -> 2024 | Donation Received -> | All | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| Page 1 of 1 1 of 1 Records | | | | | | | |
| 'P' - Previous page | 'F' - Filter | 'S' - Sort | 'A' - Ascending | 'M' - View / Update / Remove donee details | | | |
| 'N' - Next page | 'C' - Clear filter | 'H' - Search | 'D' - Descending | 'L' - Donation Distribute List of donee | | | |
| Option ('0' - Back): | | | | | | | |

Figure 1.2.3.6: Filter donee by multi filter

1.2.4 Sort Donee List

| Sort By | | | | | | | |
|--|-------------------------------|----------------------|------------------|--|---------------|-------------------|--|
| 1. Donee ID | 2. Times of donation received | | | | | | |
| 3. Register Date | | | | | | | |
| Enter option ('0' - Back, 'X' - Exit to Subsystem Menu): | 1 | | | | | | |
| Search | None | Sort By: | ID Asc | | | | |
| Filter : | Gender -> All | Type -> | All | | State -> All | | |
| | Register Date -> All | Donation Received -> | All | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |
| Page 1 of 5 7 of 30 Records | | | | | | | |
| 'P' - Previous page | 'F' - Filter | 'S' - Sort | 'A' - Ascending | 'M' - View / Update / Remove donee details | | | |
| 'N' - Next page | 'C' - Clear filter | 'H' - Search | 'D' - Descending | 'L' - Donation Distribute List of donee | | | |
| Option ('0' - Back): | | | | | | | |

Figure 1.2.4.1: Sort donee by ID in ascending order

| Sort By | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------|---|--------------|------------------------|---------------|-------------------|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|------|--------------|-------------|---------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|
| 1. Donee ID | | 2. Times of donation received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. Register Date | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Enter option ('0' - Back, 'X' - Exit to Subsystem Menu) : 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter : Gender -> All | | Sort By: Number of Donation Received Asc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Register Date -> All | | Type -> All | | State -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Donee ID</th><th>Name</th><th>Ic Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr> </thead> <tbody> <tr><td>DE0002</td><td>Siti</td><td>456789012345</td><td>01123456786</td><td>siti@mail.com</td><td>2024-04-11</td><td>0</td></tr> <tr><td>DE0003</td><td>Aishah</td><td>678901234567</td><td>01123456784</td><td>aishah@mail.com</td><td>2023-08-31</td><td>0</td></tr> <tr><td>DE0004</td><td>Hassan</td><td>901234567890</td><td>01123456781</td><td>hassan@mail.com</td><td>2024-03-07</td><td>0</td></tr> <tr><td>DE0005</td><td>Nina</td><td>012345678901</td><td>01123456780</td><td>nina@mail.com</td><td>2024-03-08</td><td>0</td></tr> <tr><td>DE0006</td><td>Rahim</td><td>112345678901</td><td>01123456770</td><td>rahim@mail.com</td><td>2023-07-15</td><td>0</td></tr> <tr><td>DE0007</td><td>Lina</td><td>212345678901</td><td>01123456771</td><td>lina@mail.com</td><td>2022-02-05</td><td>0</td></tr> <tr><td>DE0008</td><td>Azman</td><td>312345678901</td><td>01123456772</td><td>azman@mail.com</td><td>2021-11-12</td><td>0</td></tr> </tbody> </table> | | | | | | | | Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 5 7 of 30 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Option ('0' - Back) : | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.4.2: Sort donee By number of donation received in ascending order

| Sort By | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----------|------------------------------------|--------------|------------------------|---------------|-------------------|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|--------|--------------|-------------|-----------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|----------|--------------|-------------|-------------------|------------|---|
| 1. Donee ID | | 2. Times of donation received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. Register Date | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Enter option ('0' - Back, 'X' - Exit to Subsystem Menu) : 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search None | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter : Gender -> All | | Sort By: Register Date Asc | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Register Date -> All | | Type -> All | | State -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Donee ID</th><th>Name</th><th>Ic Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr> </thead> <tbody> <tr><td>DE0018</td><td>Suresh</td><td>367890123456</td><td>01123456776</td><td>suresh@mail.com</td><td>2021-09-21</td><td>0</td></tr> <tr><td>DE0008</td><td>Azman</td><td>312345678901</td><td>01123456772</td><td>azman@mail.com</td><td>2021-11-12</td><td>0</td></tr> <tr><td>DE0030</td><td>Hadi</td><td>894567890123</td><td>01123456785</td><td>hadi@mail.com</td><td>2021-11-22</td><td>1</td></tr> <tr><td>DE0007</td><td>Lina</td><td>212345678901</td><td>01123456771</td><td>lina@mail.com</td><td>2022-02-05</td><td>0</td></tr> <tr><td>DE0010</td><td>Ismail</td><td>512345678901</td><td>01123456774</td><td>ismail@mail.com</td><td>2022-05-18</td><td>0</td></tr> <tr><td>DE0014</td><td>Ling</td><td>923456789012</td><td>01123456790</td><td>ling@mail.com</td><td>2022-07-29</td><td>0</td></tr> <tr><td>DE0024</td><td>Khadijah</td><td>278901234567</td><td>01123456791</td><td>khadijah@mail.com</td><td>2022-08-11</td><td>0</td></tr> </tbody> </table> | | | | | | | | Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | DE0018 | Suresh | 367890123456 | 01123456776 | suresh@mail.com | 2021-09-21 | 0 | DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | DE0030 | Hadi | 894567890123 | 01123456785 | hadi@mail.com | 2021-11-22 | 1 | DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | DE0010 | Ismail | 512345678901 | 01123456774 | ismail@mail.com | 2022-05-18 | 0 | DE0014 | Ling | 923456789012 | 01123456790 | ling@mail.com | 2022-07-29 | 0 | DE0024 | Khadijah | 278901234567 | 01123456791 | khadijah@mail.com | 2022-08-11 | 0 |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0018 | Suresh | 367890123456 | 01123456776 | suresh@mail.com | 2021-09-21 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0030 | Hadi | 894567890123 | 01123456785 | hadi@mail.com | 2021-11-22 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0010 | Ismail | 512345678901 | 01123456774 | ismail@mail.com | 2022-05-18 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0014 | Ling | 923456789012 | 01123456790 | ling@mail.com | 2022-07-29 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0024 | Khadijah | 278901234567 | 01123456791 | khadijah@mail.com | 2022-08-11 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 5 7 of 30 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Option ('0' - Back) : | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.4.3: Sort donee by register date in ascending order

1.2.5 Ascending or Descending List

| Search : None | | Sort By: ID Asc | | | | | |
|------------------------|--------|--------------------------|--------------|-----------------|---------------|-------------------|--|
| Filter : Gender -> All | | Type -> All | | State -> All | | | |
| | | Donation Received -> All | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |

Page 1 of 5 7 of 30 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back): **d**

| Search : None | | Sort By: ID Desc | | | | | |
|------------------------|----------|--------------------------|--------------|-------------------|---------------|-------------------|--|
| Filter : Gender -> All | | Type -> All | | State -> All | | | |
| | | Donation Received -> All | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0030 | Hadi | 894567890123 | 01123456785 | hadi@mail.com | 2021-11-22 | 1 | |
| DE0029 | Siti | 793456789012 | 01123456786 | siti@mail.com | 2024-06-03 | 0 | |
| DE0028 | Aminah | 692345678901 | 01123456787 | aminah@mail.com | 2023-04-14 | 0 | |
| DE0027 | Zara | 591234567890 | 01123456788 | zara@mail.com | 2022-09-30 | 0 | |
| DE0026 | Azizah | 490123456789 | 01123456789 | azizah@mail.com | 2023-05-17 | 0 | |
| DE0025 | Sofia | 389012345678 | 01123456790 | sofia@mail.com | 2024-01-09 | 0 | |
| DE0024 | Khadijah | 278901234567 | 01123456791 | khadijah@mail.com | 2022-08-11 | 0 | |

Page 1 of 5 7 of 30 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back):

Figure 1.2.5.1: Order donee list in descending order based on current sorting

| Search None | | Sort By: Register Date Desc | | | | State -> All | |
|-------------|----------------------|-----------------------------|--------------------------|-----------------|---------------|-------------------|--|
| Filter : | Gender -> All | Type -> All | Donation Received -> All | | | | |
| | Register Date -> All | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0013 | Maya | 890123456789 | 01123456782 | maya@mail.com | 2024-06-25 | 0 | |
| DE0029 | Siti | 793456789012 | 01123456786 | siti@mail.com | 2024-06-03 | 0 | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0019 | Vani | 478901234567 | 01123456775 | vani@mail.com | 2024-03-02 | 0 | |
| DE0025 | Sofia | 389012345678 | 01123456790 | sofia@mail.com | 2024-01-09 | 0 | |

Page 1 of 5 7 of 30 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back) : a

| Search None | | Sort By: Register Date Asc | | | | State -> All | |
|-------------|----------------------|----------------------------|--------------------------|-------------------|---------------|-------------------|--|
| Filter : | Gender -> All | Type -> All | Donation Received -> All | | | | |
| | Register Date -> All | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0018 | Suresh | 367890123456 | 01123456776 | suresh@mail.com | 2021-09-21 | 0 | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | |
| DE0030 | Hadi | 894567890123 | 01123456785 | hadi@mail.com | 2021-11-22 | 1 | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |
| DE0010 | Ismail | 512345678901 | 01123456774 | ismail@mail.com | 2022-05-18 | 0 | |
| DE0014 | Ling | 923456789012 | 01123456790 | ling@mail.com | 2022-07-29 | 0 | |
| DE0024 | Khadijah | 278901234567 | 01123456791 | khadijah@mail.com | 2022-08-11 | 0 | |

Page 1 of 5 7 of 30 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back) :

Figure 1.2.5.2: Order donee list in ascending order based on current sorting

1.2.6 Search Donee

| Search Donee By | | | | | | | | | |
|--|----------------------|--------------|--------------------------|----------|--------------|---|--|--|--|
| 1. Donee ID | 2. Name | 3. IC Number | 4. Phone Number | 5. Email | 6. Address | Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu) : 1 | | | |
| Search by | | | | | | | | | |
| ----- | | | | | | | | | |
| Digit of ID (Eg. 'DE0001' enter 1 ;'0' - Back) : 1 | | | | | | | | | |
| Search ID : 1 | Sort By: ID Asc | | | | State -> All | | | | |
| Filter : | Gender -> All | Type -> All | Donation Received -> All | | | | | | |
| | Register Date -> All | | | | | | | | |

| | | | | | | | |
|----------|------|--------------|--------------|--------------|---------------|-------------------|--|
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |

Page 1 of 1 1 of 1 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back) :

Figure 1.2.6.1: Search donee by Id

| Search Donee By | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----------|--------------------------|-----------------|-----------------|---------------|--|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|-----|--------------|-------------|--------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|
| 1. Donee ID | 2. Name | 3. IC Number | 4. Phone Number | 5. Email | 6. Address | Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu): 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search by | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name ('0' - Back): a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search : Filter | Name : a | Sort By: ID Asc | Type -> All | State -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter : Register Date -> All | | Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr> </thead> <tbody> <tr><td>DE0001</td><td>Jay</td><td>123456789012</td><td>01123456789</td><td>jay@mail.com</td><td>2023-01-10</td><td>1</td></tr> <tr><td>DE0003</td><td>Aishah</td><td>678901234567</td><td>01123456784</td><td>aishah@mail.com</td><td>2023-08-31</td><td>0</td></tr> <tr><td>DE0004</td><td>Hassan</td><td>901234567890</td><td>01123456781</td><td>hassan@mail.com</td><td>2024-03-07</td><td>0</td></tr> <tr><td>DE0005</td><td>Nina</td><td>012345678901</td><td>01123456780</td><td>nina@mail.com</td><td>2024-03-08</td><td>0</td></tr> <tr><td>DE0006</td><td>Rahim</td><td>112345678901</td><td>01123456770</td><td>rahim@mail.com</td><td>2023-07-15</td><td>0</td></tr> <tr><td>DE0007</td><td>Lina</td><td>212345678901</td><td>01123456771</td><td>lina@mail.com</td><td>2022-02-05</td><td>0</td></tr> <tr><td>DE0008</td><td>Azman</td><td>312345678901</td><td>01123456772</td><td>azman@mail.com</td><td>2021-11-12</td><td>0</td></tr> </tbody> </table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 4 7 of 25 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Option ('0' - Back): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.6.2: Search donee by name

| Search Donee By | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-----------------|--------------------------|-----------------|-----------------|---------------|--|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|-----|--------------|-------------|--------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|
| 1. Donee ID | 2. Name | 3. IC Number | 4. Phone Number | 5. Email | 6. Address | Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu): 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search by | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IC Number ('0' - Back): 123 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search : Filter | IC Number : 123 | Sort By: ID Asc | Type -> All | State -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filter : Register Date -> All | | Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr> </thead> <tbody> <tr><td>DE0001</td><td>Jay</td><td>123456789012</td><td>01123456789</td><td>jay@mail.com</td><td>2023-01-10</td><td>1</td></tr> <tr><td>DE0002</td><td>Siti</td><td>456789012345</td><td>01123456786</td><td>siti@mail.com</td><td>2024-04-11</td><td>0</td></tr> <tr><td>DE0003</td><td>Aishah</td><td>678901234567</td><td>01123456784</td><td>aishah@mail.com</td><td>2023-08-31</td><td>0</td></tr> <tr><td>DE0004</td><td>Hassan</td><td>901234567890</td><td>01123456781</td><td>hassan@mail.com</td><td>2024-03-07</td><td>0</td></tr> <tr><td>DE0005</td><td>Nina</td><td>012345678901</td><td>01123456780</td><td>nina@mail.com</td><td>2024-03-08</td><td>0</td></tr> <tr><td>DE0006</td><td>Rahim</td><td>112345678901</td><td>01123456770</td><td>rahim@mail.com</td><td>2023-07-15</td><td>0</td></tr> <tr><td>DE0007</td><td>Lina</td><td>212345678901</td><td>01123456771</td><td>lina@mail.com</td><td>2022-02-05</td><td>0</td></tr> </tbody> </table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 4 7 of 27 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Option ('0' - Back): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.6.3: Search donee by IC number

| Search Donee By | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------|--------------|--------------|-----------------|---------------|-------------------|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|-----|--------------|-------------|--------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|
| 1. Donee ID | | 2. Name | | 3. IC Number | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4. Phone Number | | 5. Email | | 6. Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu): 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search by | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Phone Number (Max 11 digit; '0' - Back): 011 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search Phone Number : 011 Sort By: ID Asc Filter : Type -> All State -> All Register Date -> All Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"><thead><tr><th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr></thead><tbody><tr><td>DE0001</td><td>Jay</td><td>123456789012</td><td>01123456789</td><td>jay@mail.com</td><td>2023-01-10</td><td>1</td></tr><tr><td>DE0002</td><td>Siti</td><td>456789012345</td><td>01123456786</td><td>siti@mail.com</td><td>2024-04-11</td><td>0</td></tr><tr><td>DE0003</td><td>Aishah</td><td>678901234567</td><td>01123456784</td><td>aishah@mail.com</td><td>2023-08-31</td><td>0</td></tr><tr><td>DE0004</td><td>Hassan</td><td>901234567890</td><td>01123456781</td><td>hassan@mail.com</td><td>2024-03-07</td><td>0</td></tr><tr><td>DE0005</td><td>Nina</td><td>012345678901</td><td>01123456780</td><td>nina@mail.com</td><td>2024-03-08</td><td>0</td></tr><tr><td>DE0006</td><td>Rahim</td><td>112345678901</td><td>01123456770</td><td>rahim@mail.com</td><td>2023-07-15</td><td>0</td></tr><tr><td>DE0007</td><td>Lina</td><td>212345678901</td><td>01123456771</td><td>lina@mail.com</td><td>2022-02-05</td><td>0</td></tr></tbody></table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 5 7 of 30 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee Option ('0' - Back): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.6.4: Search donee by phone number

| Search Donee By | | | | | | | | | | | | | | | | | | | | | |
|---|-------|--------------|--------------|----------------|---------------|-------------------|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|-------|--------------|-------------|----------------|------------|---|
| 1. Donee ID | | 2. Name | | 3. IC Number | | | | | | | | | | | | | | | | | |
| 4. Phone Number | | 5. Email | | 6. Address | | | | | | | | | | | | | | | | | |
| Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu): 5 | | | | | | | | | | | | | | | | | | | | | |
| Search by | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | |
| Email ('0' - Back): so | | | | | | | | | | | | | | | | | | | | | |
| Search Email : so Sort By: ID Asc Filter : Type -> All State -> All Register Date -> All Donation Received -> All | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"><thead><tr><th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr></thead><tbody><tr><td>DE0025</td><td>Sofia</td><td>389012345678</td><td>01123456790</td><td>sofia@mail.com</td><td>2024-01-09</td><td>0</td></tr></tbody></table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | DE0025 | Sofia | 389012345678 | 01123456790 | sofia@mail.com | 2024-01-09 | 0 |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | |
| DE0025 | Sofia | 389012345678 | 01123456790 | sofia@mail.com | 2024-01-09 | 0 | | | | | | | | | | | | | | | |
| Page 1 of 1 1 of 1 Records | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee Option ('0' - Back): | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.6.5: Search donee by email

| Search Donee By | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------|--------------|-----------------|-----------------|---------------|---|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|-----|--------------|-------------|--------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|
| 1. Donee ID | 2. Name | 3. IC Number | 4. Phone Number | 5. Email | 6. Address | Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu) : 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search by | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Address ('0' - Back): Kuala Lumpur | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search Address : Kuala Lumpur Sort By: ID Asc Filter : Type -> All State -> All Register Date -> All Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"><thead><tr><th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr></thead><tbody><tr><td>DE0001</td><td>Jay</td><td>123456789012</td><td>01123456789</td><td>jay@mail.com</td><td>2023-01-10</td><td>1</td></tr><tr><td>DE0002</td><td>Siti</td><td>456789012345</td><td>01123456786</td><td>siti@mail.com</td><td>2024-04-11</td><td>0</td></tr><tr><td>DE0015</td><td>Chong</td><td>034567890123</td><td>01123456779</td><td>chong@mail.com</td><td>2023-02-10</td><td>0</td></tr><tr><td>DE0016</td><td>Ahmad</td><td>145678901234</td><td>01123456778</td><td>ahmad@mail.com</td><td>2022-12-07</td><td>0</td></tr><tr><td>DE0018</td><td>Suresh</td><td>367890123456</td><td>01123456776</td><td>suresh@mail.com</td><td>2021-09-21</td><td>0</td></tr><tr><td>DE0023</td><td>Aina</td><td>167890123456</td><td>01123456792</td><td>aina@mail.com</td><td>2023-10-28</td><td>0</td></tr><tr><td>DE0028</td><td>Aminah</td><td>692345678901</td><td>01123456787</td><td>aminah@mail.com</td><td>2023-04-14</td><td>0</td></tr></tbody></table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | DE0015 | Chong | 034567890123 | 01123456779 | chong@mail.com | 2023-02-10 | 0 | DE0016 | Ahmad | 145678901234 | 01123456778 | ahmad@mail.com | 2022-12-07 | 0 | DE0018 | Suresh | 367890123456 | 01123456776 | suresh@mail.com | 2021-09-21 | 0 | DE0023 | Aina | 167890123456 | 01123456792 | aina@mail.com | 2023-10-28 | 0 | DE0028 | Aminah | 692345678901 | 01123456787 | aminah@mail.com | 2023-04-14 | 0 |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0015 | Chong | 034567890123 | 01123456779 | chong@mail.com | 2023-02-10 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0016 | Ahmad | 145678901234 | 01123456778 | ahmad@mail.com | 2022-12-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0018 | Suresh | 367890123456 | 01123456776 | suresh@mail.com | 2021-09-21 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0023 | Aina | 167890123456 | 01123456792 | aina@mail.com | 2023-10-28 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0028 | Aminah | 692345678901 | 01123456787 | aminah@mail.com | 2023-04-14 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 2 7 of 9 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee Option ('0' - Back): | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.6.6: Search donee by entering keyword of address

| Search Donee By | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------|--------------|-----------------|-----------------|---------------|---|--|----------|------|-----------|--------------|-------|---------------|-------------------|--------|-----|--------------|-------------|--------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|--------|--------------|-------------|-----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|--------|------|--------------|-------------|---------------|------------|---|--------|-------|--------------|-------------|----------------|------------|---|
| 1. Donee ID | 2. Name | 3. IC Number | 4. Phone Number | 5. Email | 6. Address | Search donee by? ('0' - Back, 'X' - Exit to Subsystem Menu) : 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search by | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ----- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name ('0' - Back): a | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Search Name : a Sort By: ID Asc Filter : Type -> All State -> All Register Date -> All Donation Received -> All | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"><thead><tr><th>Donee ID</th><th>Name</th><th>IC Number</th><th>Phone Number</th><th>Email</th><th>Register Date</th><th>Donation Received</th></tr></thead><tbody><tr><td>DE0001</td><td>Jay</td><td>123456789012</td><td>01123456789</td><td>jay@mail.com</td><td>2023-01-10</td><td>1</td></tr><tr><td>DE0003</td><td>Aishah</td><td>678901234567</td><td>01123456784</td><td>aishah@mail.com</td><td>2023-08-31</td><td>0</td></tr><tr><td>DE0004</td><td>Hassan</td><td>901234567890</td><td>01123456781</td><td>hassan@mail.com</td><td>2024-03-07</td><td>0</td></tr><tr><td>DE0005</td><td>Nina</td><td>012345678901</td><td>01123456780</td><td>nina@mail.com</td><td>2024-03-08</td><td>0</td></tr><tr><td>DE0006</td><td>Rahim</td><td>112345678901</td><td>01123456770</td><td>rahim@mail.com</td><td>2023-07-15</td><td>0</td></tr><tr><td>DE0007</td><td>Lina</td><td>212345678901</td><td>01123456771</td><td>lina@mail.com</td><td>2022-02-05</td><td>0</td></tr><tr><td>DE0008</td><td>Azman</td><td>312345678901</td><td>01123456772</td><td>azman@mail.com</td><td>2021-11-12</td><td>0</td></tr></tbody></table> | | | | | | | | Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 |
| Donee ID | Name | IC Number | Phone Number | Email | Register Date | Donation Received | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page 1 of 4 7 of 25 Records | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee Option ('0' - Back): f | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Select filter 1. Gender 2. Type 3. State 4. Register Date 5. Num of Donation Received Filter donee by? ('0' - Back, 'X' - Exit to Subsystem menu) : 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Gender ('F' - Female, 'M' - Male; '0' - Back): f | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 1.2.6.7: Combination of search and filter the list

1.3 View / Update / Remove Donee Detail

| Search Phone Number : 011 | | Sort By: ID Asc | | | | | |
|---------------------------|--------|--------------------------|--------------|-----------------|---------------|-------------------|--|
| Filter : Gender -> All | | Type -> All | | | State -> All | | |
| | | Donation Received -> All | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | |
| DE0003 | Aishah | 678901234567 | 01123456784 | aishah@mail.com | 2023-08-31 | 0 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |

Page 1 of 5 7 of 30 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back): m

Enter digit of ID you want in the list for further action. (eg. 'AB0001' enter 1; '0' - Back: 3

Figure 1.3.1.1: Select donee for view, edit or remove

1.3.2 Edit Donee Detail

```

+-----+
| DE0003 - Aishah           Register : 2023-08-31 |
+-----+
| Personal Detail:
|   IC Number      : 678901234567
|   Gender         : Female
|   Phone Number   : 01123456784
|   Email          : aishah@mail.com
|   Type           : Individual
+-----+
| 6 Jalan Subang          | Total Received:
| Kuantan, 67890          |   Times       : 0
| Pahang                  |   Item Value  : 0.00
|                         |   Money       : 0.00
|                         |-----+
|                         |   Total Value : 0.00
+-----+
'E' - Edit             'R' - Remove
'P' - Previous donee  'N' - Next donee

Select option for futher action to the donee? ('0' - Back, 'X' - Exit to Subsystem Menu): e

Donee DE0003
+-----+
| DE0003 - Aishah           Register : 2023-08-31 |
+-----+
| Personal Detail:
|   IC Number      : 678901234567
|   Gender         : Female
|   Phone Number   : 01123456784
|   Email          : aishah@mail.com
|   Type           : Individual
+-----+
| 6 Jalan Subang          | Total Received:
| Kuantan, 67890          |   Times       : 0
| Pahang                  |   Item Value  : 0.00
|                         |   Money       : 0.00
|                         |-----+
|                         |   Total Value : 0.00
+-----+

1. Name      2. IC Number
3. Gender    4. Phone Number
5. Email     6. Type
7. Address

Option ('0' - Back, 'X' - Exit to Subsystem Menu): |

```

Figure 1.3.2.1: Display the detail of selected donee and select option for edit action.

```

Donee DE0003
+-----+
| DE0003 - Aishah           Register : 2023-08-31 |
+-----+
| Personal Detail:
|   IC Number      : 678901234567
|   Gender         : Female
|   Phone Number   : 01123456784
|   Email          : aishah@mail.com
|   Type           : Individual
+-----+
| 6 Jalan Subang          | Total Received:
| Kuantan, 67890          |   Times        : 0
| Pahang                   |   Item Value   : 0.00
|                           |   Money        : 0.00
|                           |   Total Value  0.00
+-----+
1. Name       2. IC Number
3. Gender     4. Phone Number
5. Email      6. Type
7. Address
Option ('0' - Back, 'X' - Exit to Subsystem Menu): 1

Name ('0' - Back): Aila

Do you want to change other information? ('Y' - Yes; 'N' - No): y

```

Figure 1.3.2.2: Edit the information of selected donee

```

Do you want to change other information? ('Y' - Yes; 'N' - No): n
+-----+
| DE0003 - Aila           Register : 2023-08-31 |
+-----+
| Personal Detail:
|   IC Number      : 123123121234
|   Gender         : Female
|   Phone Number   : 01123456784
|   Email          : aishah@mail.com
|   Type           : Individual
+-----+
| 6 Jalan Subang          | Total Received:
| Kuantan, 67890          |   Times        : 0
| Pahang                   |   Item Value   : 0.00
|                           |   Money        : 0.00
|                           |   Total Value  0.00
+-----+
Do you confirms to edit Donee DE0003 information? ('Y' - Yes; 'N' - No): y
Donee DE0003 information edit successful.

```

Figure 1.3.2.3: Select confirmation option for confirm edit, edit successful message show

1.3.3 Remove Donee

```

+-----+
| DE0003 - Aila                         Register : 2023-08-31 |
+-----+
| Personal Detail:
|   IC Number      : 123123121234
|   Gender         : Female
|   Phone Number   : 01123456784
|   Email          : aishah@mail.com
|   Type           : Individual
+-----+
| 6 Jalan Subang          | Total Received:
| Kuantan, 67890          |   Times       : 0
| Pahang                  |   Item Value  : 0.00
|                           |   Money       : 0.00
|                           |   -----
|                           |   Total Value : 0.00
+-----+
'E' - Edit      'R' - Remove
'P' - Previous donee  'N' - Next donee

Select option for futher action to the donee? ('0' - Back, 'X' - Exit to Subsystem Menu): r

Do you confirm to delete Donee DE0003? ('Y' - Yes; 'N' - No): Y
Donee remove successful.

```

Figure 1.3.3.1: Enter the remove option and confirm remove, remove successful message display.

1.3.4 Previous or Next Donee in List

```

+-----+
| DE0005 - Nina                          Register : 2024-03-08 |
+-----+
| Personal Detail:
|   IC Number      : 012345678901
|   Gender         : Female
|   Phone Number   : 01123456780
|   Email          : nina@mail.com
|   Type           : Individual
+-----+
| 10 Jalan Pasir             | Total Received:
| Kuala Terengganu, 01234    |   Times       : 0
| Terengganu                 |   Item Value  : 0.00
|                           |   Money       : 0.00
|                           |   -----
|                           |   Total Value : 0.00
+-----+
'E' - Edit      'R' - Remove
'P' - Previous donee  'N' - Next donee

Select option for futher action to the donee? ('0' - Back, 'X' - Exit to Subsystem Menu): n

+-----+
| DE0007 - Lina                          Register : 2022-02-05 |
+-----+
| Personal Detail:
|   IC Number      : 212345678901
|   Gender         : Female
|   Phone Number   : 01123456771
|   Email          : lina@mail.com
|   Type           : Individual
+-----+
| 12 Jalan Sejahtera        | Total Received:
| Johor Bahru, 56812        |   Times       : 0
| Johor                     |   Item Value  : 0.00
|                           |   Money       : 0.00
|                           |   -----
|                           |   Total Value : 0.00
+-----+
'E' - Edit      'R' - Remove
'P' - Previous donee  'N' - Next donee

Select option for futher action to the donee? ('0' - Back, 'X' - Exit to Subsystem Menu): |

```

Figure 1.3.4.1: Jump to next donee in the list displayed before to take action on the donee

```

+-----+
| DE0007 - Lina           Register : 2022-02-05 |
+-----+
| Personal Detail:
|   IC Number      : 212345678901
|   Gender         : Female
|   Phone Number   : 01123456771
|   Email          : lina@mail.com
|   Type           : Individual
+-----+
| 12 Jalan Sejahtera | Total Received:
| Johor Bahru, 56812 |   Times       : 0
| Johor               |   Item Value  : 0.00
|                      |   Money       : 0.00
|                      |   Total Value: 0.00
+-----+
'E' - Edit      'R' - Remove
'P' - Previous donee  'N' - Next donee

Select option for futher action to the donee? ('0' - Back, 'X' - Exit to Subsystem Menu): p

```



```

+-----+
| DE0005 - Nina           Register : 2024-03-08 |
+-----+
| Personal Detail:
|   IC Number      : 012345678901
|   Gender         : Female
|   Phone Number   : 01123456780
|   Email          : nina@mail.com
|   Type           : Individual
+-----+
| 10 Jalan Pasir        | Total Received:
| Kuala Terengganu, 01234 |   Times       : 0
| Terengganu             |   Item Value  : 0.00
|                      |   Money       : 0.00
|                      |   Total Value: 0.00
+-----+
'E' - Edit      'R' - Remove
'P' - Previous donee  'N' - Next donee

Select option for futher action to the donee? ('0' - Back, 'X' - Exit to Subsystem Menu): |

```

Figure 1.3.4.2: Jump to previous donee in the list displayed before to take action on the donee

1.3.5 Donation Distribution List of Donee Received

| Search | None | Sort By: | ID Asc | |
|----------|----------------------|--------------------------|--------------|-----------------|
| Filter | : Gender -> All | Type -> All | | State -> All |
| | Register Date -> All | Donation Received -> All | | |
| Donee ID | Name | Ic Number | Phone Number | Email |
| | | | | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com |
| | | | | |

Page 1 of 5 7 of 29 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back): 1

Enter digit of ID you want in the list for further action. (eg. 'AB0001' enter 1; '0' - Back: 1

Figure 1.3.5.1: Select option to view specific donee's donation distribution, enter the donee ID

| Distribution of Donee DE0001 | | | | | |
|------------------------------|------------------------|------------|-----------------|---------|--|
| ID | Category Item Received | Value (RM) | Distribute Date | Status | |
| DI0001 | Money | 4350.00 | 2024-08-17 | Pending | |
| | Food & Beverage | | | | |

Total Amount (RM): 4350.00 Total Received: 1

Enter distribution ID to view further detail ('0' - Back, 'X' - Exit to Subsystem Menu):

Figure 1.3.5.2: Display the selected donee's donation distribution list.

| Enter distribution ID to view further detail ('0' - Back, 'X' - Exit to Subsystem Menu) :1 | | | | | |
|---|--|--|--|--|--|
| +-----+ Distribution ID : DI0001 +-----+ | | | | | |
| +-----+ Donee : Jay - DE0001 +-----+ | | | | | |
| +-----+ Stock Out Item: +-----+ | | | | | |
| OU0001 Cash 1000.00 OU0002 Bank Transfer 750.00 OU0003 Canned Beans x 50 OU0004 Rice Pack x 100 Total value (RM) : 4350.00 +-----+ | | | | | |
| +-----+ Distribution Date : 2024-08-17 +-----+ | | | | | |
| +-----+ Status : Pending +-----+ | | | | | |

'P' - Previous distribution, 'N' - Next distribution of Jay
Action ('0' - Back, 'X' - Exit to Subsystem Menu):

Figure 1.3.5.3: Enter distribution ID of the donee to view the distribution detail.

1.4 Clear Filter of List

| Search Name : a | | Sort By: ID Asc | | | | | |
|-----------------|----------------------|--------------------------|--------------|-----------------|---------------|-------------------|--|
| Filter : | Gender -> Male | Type -> | Individual | State -> All | | | |
| | Register Date -> All | Donation Received -> All | | | | | |
| | | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | |
| DE0010 | Ismail | 512345678901 | 01123456774 | ismail@mail.com | 2022-05-18 | 0 | |

Page 1 of 1 5 of 5 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back): c

| Search None | | Sort By: ID Asc | | | | | |
|-------------|----------------------|--------------------------|--------------|-----------------|---------------|-------------------|--|
| Filter : | Gender -> All | Type -> | All | State -> All | | | |
| | Register Date -> All | Donation Received -> All | | | | | |
| | | | | | | | |
| Donee ID | Name | Ic Number | Phone Number | Email | Register Date | Donation Received | |
| DE0001 | Jay | 123456789012 | 01123456789 | jay@mail.com | 2023-01-10 | 1 | |
| DE0002 | Siti | 456789012345 | 01123456786 | siti@mail.com | 2024-04-11 | 0 | |
| DE0004 | Hassan | 901234567890 | 01123456781 | hassan@mail.com | 2024-03-07 | 0 | |
| DE0005 | Nina | 012345678901 | 01123456780 | nina@mail.com | 2024-03-08 | 0 | |
| DE0006 | Rahim | 112345678901 | 01123456770 | rahim@mail.com | 2023-07-15 | 0 | |
| DE0007 | Lina | 212345678901 | 01123456771 | lina@mail.com | 2022-02-05 | 0 | |
| DE0008 | Azman | 312345678901 | 01123456772 | azman@mail.com | 2021-11-12 | 0 | |

Page 1 of 5 7 of 29 Records

'P' - Previous page 'F' - Filter 'S' - Sort 'A' - Ascending 'M' - View / Update / Remove donee details
 'N' - Next page 'C' - Clear filter 'H' - Search 'D' - Descending 'L' - Donation Distribute List of donee
 Option ('0' - Back): |

Figure 1.4: Clear all the filter and search of the list

1.5 Donee Summary Report

| Donee Report | | |
|-------------------------------------|--------|--|
| Report | Option | |
| Total Value Donee Received Report | 1 | |
| Top 10 Donee Received Selected Item | 2 | |
| Specific Donee Summary Report | 3 | |
| Exit | 0 | |

Please select option (0 - 3): 1

Figure 1.5: Select option to view different report on Donee Report Menu.

1.5.1 Total Value Donee Received Report

```
+-----+  
|          Donee Type      |  
+-----+  
|     P      | Individual    |  
|     O      | Organization  |  
|     F      | Family        |  
+-----+  
  
Select type of donee for generate report. ('0' - Back): p  
  
"Eg. '2023-08-00' - Report of August 2023, '2024-00-00' - Report of 2024"  
Enter a date (yyyy-MM-dd; '0' - Back): 2024-00-00
```

Figure 1.5.1.1: Select the donee type and enter date for generate report

1.5.2 Top 10 Donee Received Selected Item\

1.5.3 Specific Donee Summary Report

<INSTRUCTIONS> (please delete all instructions before submitting your assignment report)

1. For the source code

a. **Source codes for Control class.**

<remove instructions>

- i. Ensure that the **author name** of each class is stated in the comment section of each classes,
- ii. **Copy & Paste** of your source code listing in NetBeans(source codes) that include the use of collection ADTs.
- iii. For Source Code listing, use **font color blue** to all lines of code that involve the **declaration or creation of collection ADT objects (control class, Dao)**, and the **invocation of collection ADT methods**.

2. **Screenshots**

</INSTRUCTIONS>

| Name | Student ID | Prog / Tut.Grp | Signature |
|---------------------|----------------|----------------|--|
| Chia Ming Yi | 24WMR090 40 | RSD1 G5 |  |

Subsystem : Donor Management Subsystem

<INSTRUCTIONS> (please delete all instructions before submitting your assignment report)

3. For the source code

a. **Source codes for Control class.**

<remove instructions>

- i. Ensure that the **author name** of each class is stated in the comment section of each classes,
- ii. **Copy & Paste** of your source code listing in NetBeans(source codes) that include the use of collection ADTs.
- iii. For Source Code listing, use **font color blue** to all lines of code that involve the **declaration or creation of collection ADT objects (control class, Dao)**, and the **invocation of collection ADT methods**.

4. Screenshots

</INSTRUCTIONS>

| Name | Student ID | Prog / Tut.Grp | Signature |
|-------------|------------|----------------|--|
| Lim Jun Wei | 24WMR09078 | RSD1 G5 |  |

Subsystem : Donation Management Subsystem

Business Rule:

- Donation item amount for money in each donation should between 0.01 and 999,999.99 only.
- Donation item quantity for food and beverage in each donation should between 1 and 99999 only.
- Donation notes should not exceed 30 characters.
- Donated money item in created donation cannot be updated.
- Created donation can only be updated before 24 hours from the time when users created the donation.
- Donation item name should not exceed 20 characters when creating new donation item.
- Donation item name should not be same with created donation item name when creating new donation item.
- Donation item initial quantity should not exceed 300 when creating new donation item.
- Donation item amount should between 0.01 and 999, 999.99 only when creating new donation item.

1. Source Codes Listing

Controller: DonationControl.java

```

package control;

import adt.DoublyLinkedList;
import adt.ListInterface;
import static boundary.CLONCharity.h;
import static boundary.CLONCharity.v;
import static boundary.CLONCharity.exitStr;
import static boundary.CLONCharity.optionInt;
import static boundary.CLONCharity.optionYN;
import static boundary.CLONCharity.scan;
import dao.Initializer;
import entity.Donor;
import entity.Donation;
import entity.Item;
import entity.Person;
import entity.StockIn;
import java.time.LocalDate;

/**
 *
 * @Lim Jun Wei
 */
public class DonationControl {
    private static ListInterface<Donor> donorListPrivate = Initializer.donorListPrivate;
    private static ListInterface<Donor> donorListPublic = Initializer.donorListPublic;
    private static ListInterface<Donor> donorListGovernment =
    Initializer.donorListGovernment;
    private static ListInterface<Donor> donorList =
    donorListPrivate.merge(donorListPublic).merge(donorListGovernment);
    private static ListInterface<Item> moneyList = Initializer.moneyList;
    private static ListInterface<Item> foodBeverageList = Initializer.foodBeverageList;
    private static ListInterface<StockIn> stockInList = Initializer.stockInList;
    private static ListInterface<Donation> donationList = Initializer.donationList;

    /*
     main menu
    */
    // show donation menu
    public static void donationMenu() {
        // update donor list changes
        donorList = donorListPrivate.merge(donorListPublic).merge(donorListGovernment);
        optionInt = -1; // reset option value
    }
}

```

```

while (optionInt < 0 || optionInt > 7) {
    mainMenu();

    optionInt = h.scanInt();
    switch (optionInt) {
        case 1: {addDonation(); break;}
        case 2: {viewDonation(); break;}
        case 3: {searchDonation(); break;}
        case 4: {updateDonation(); break;}
        case 5: {removeDonation(); break;}
        case 6: {donationReport(); break;}
        case 7: {createDonationItem(); break;}
        case 0: {System.out.println("\n Back to main menu page...\n"); return;}
        default:
            System.err.println(v.errOptionInt);
            System.out.println(exitStr);
    }
    optionInt = -1;
}
}

/*
 main function
*/
// add new donation
public static void addDonation() {
    // reset variable
    Donor useDonor = null;
    int useCategory = -1;
    Item useItem = null;
    StockIn useStockIn = null;

    while (useDonor == null) {
        ListInterface<Item> useMoneyList = moneyList.merge(new DoublyLinkedList<>());
        ListInterface<Item> useFoodBeverageList = foodBeverageList.merge(new
DoublyLinkedList<>());
        // reset temporary stockInList
        ListInterface<StockIn> useStockInList = new DoublyLinkedList<>();

        useDonor = selectDonor();
        if (useDonor == null) break; // exit

        useCategory = -1;
        while (useCategory == -1) {
            useCategory = selectCategory(); // select category
    }
}

```

```

optionYN = "Y";
while (optionYN.equals("Y")) {
    useStockIn = null;
    while (useStockIn == null) {
        if (useCategory == 0) { // exit
            optionYN = "N";
            useDonor = null;
            break; // back to selectDonor()
        }
        else if (useCategory == 1) useItem = selectItem(useMoneyList); // money
        else if (useCategory == 2) useItem = selectItem(useFoodBeverageList); //
other item

        if (useItem == null) { // exit
            useCategory = -1;
            break;
        }

        useStockIn = makeStockIn(useCategory, useItem); // prompt donation item
quantity / amount

        if (useStockIn != null) {
            useStockInList.add(useStockIn); // store stockIn object into temporary
stockInList

            if (moneyList.search(useStockIn.getItem()) != null)
useMoneyList.remove(useStockIn.getItem());
            if (foodBeverageList.search(useStockIn.getItem()) != null)
useFoodBeverageList.remove(useStockIn.getItem());
        }
    }
    if (useDonor == null) break;
    if (useCategory == -1) break;

    optionYN = "null";

    while (optionYN.equals("null")) {
        System.out.print("\n Do you want to donate other items? (Y/N): ");
        optionYN = h.scanOptionStr();

        // check whether option input is correct
        optionYN = v.verifyOptionYN(optionYN) ? optionYN : "null";
    }
}

if (useDonor == null) break;
if (useCategory == -1) continue;

```



```

// search donation
public static void searchDonation() {
    optionInt = -1;

    while (optionInt == -1) {
        donationSearchMenu();
        optionInt = h.scanInt();

        switch (optionInt) {
            case 0: {System.out.println(exitStr); return;}
            case 1: {searchByDonationID(); break;}
            case 2: {searchByDonorID(); break;}
            default:
                optionInt = -1;
                System.err.println(v.errOptionInt);
        }
        optionInt = -1;
    }
}

// update donation detail
public static void updateDonation() {
    Donation useDonation = null;
    optionInt = 0;

    while (optionInt == 0) {
        useDonation = null;
        while (useDonation == null) {
            useDonation = selectDonation();

            if (useDonation == null) { // exit
                return;
            } else {
                // check whether the donation has been created over 3 days
                if (LocalDate.now().isAfter(useDonation.getDonationDate().plusDays(1))) {
                    System.err.println(v.err1DayOver);
                    useDonation = null;
                }
            }
        }
    }

    // valid updateable donation
    optionInt = -1;
    while (optionInt == -1) {
        donationCard(useDonation);
        donationUpdateMenu();

        System.out.print("\n Select an option ('0' to exit): ");
    }
}

```

```

optionInt = h.scanInt();

switch (optionInt) {
    case 0: {System.out.println(exitStr); break;} // exit
    case 1: {updateDonor(useDonation); optionInt = 1; break;} // switch donor
    case 2: {updateQty(useDonation); optionInt = 2; break;} // add quantity of
current item
    case 3: {addNewItem(useDonation); optionInt = 3; break;} // add new item
    case 4: {removeItem(useDonation); optionInt = 4; break;} // remove
donated item
    case 5: {updateNotes(useDonation); optionInt = 5; break;}// update notes
    default: {optionInt = -1; System.err.println(v.errOptionInt);} // invalid option
}

if (optionInt > 0 && optionInt < 6) {
    // prompt user to continue update the donation
    optionYN = h.inputYN("\n Do you want to continue update the donation?");

    if (optionYN.equals("Y")) optionInt = -1;
    else optionInt = 0;
}

}

}

}

// remove donation
public static void removeDonation() {
    Donation useDonation = selectDonation();

    if (useDonation != null) {
        donationCard(useDonation);

        optionYN = h.inputYN("\n Are you sure to remove this donation (" +
useDonation.formatID() + ") ?: ");
        if (optionYN.equals("Y")) {
            ListInterface<StockIn> removeItemList = useDonation.getStockIn();
            StockIn currentStockIn = removeItemList.getFirst();

            for (int i = 0; i < removeItemList.getSize(); i++) {
                Item moneyItem = moneyList.search(currentStockIn.getItem());
                Item foodBeverageItem = foodBeverageList.search(currentStockIn.getItem());

                if (moneyItem != null) {
                    moneyItem.setAmount(moneyItem.getAmount() -
currentStockIn.getAmt());
                }
            }
        }
    }
}

```

```

        moneyList.update(moneyItem);
    } else if (foodBeverageItem != null) {
        foodBeverageItem.setQty(foodBeverageItem.getQty() -
currentStockIn.getQty());
        foodBeverageList.update(foodBeverageItem);
    }
    StockIn removeStockIn = currentStockIn;
    currentStockIn = stockInList.getNextData(currentStockIn);
    stockInList.remove(removeStockIn);
}
donationList.remove(useDonation);
donationViewList(donationList);
System.out.println("\n Donation has been deleted successfully.");
System.out.println(exitStr);
}
}
}

// generate donation report
public static void donationReport() {
    optionInt = -1; // reset variable
    optionYN = "Y";

    while (optionYN.equals("Y")) {
        optionInt = -1;
        while (optionInt == -1) {
            donationReportMenu();
            optionInt = h.scanInt();

            switch (optionInt) {
                case 0:
                    System.out.println(exitStr);
                    return;
                case 1:
                    top10DonationReport();
                    break;
                case 2:
                    top10DonatedItemReport();
                    break;
                case 3:
                    top10CriticalShortageReport();
                    break;
                default:
                    System.err.println(v.errOptionInt);
                    optionInt = -1;
            }
        }
    }
}

```

```
optionYN = h.inputYN("\n Do you want to generate other report?");
if (optionYN.equals("N")) System.out.println(exitStr);
}

// create donation item
public static void createDonationItem() {
    optionYN = "Y";
    while (optionYN.equals("Y")) {
        ListInterface<Item> allItemList = null;
        String name = "";
        String desc = "";
        int category = 0;
        int qty = 0;
        double amt = 0.0;

        while (name.equals("")) {
            name = h.inputDonationItemName(); // prompt donation item name

            if (name.equals("")) return;

            // check whether donation item name redundant
            allItemList = moneyList.merge(foodBeverageList);
            Item currentItem = allItemList.getFirst();

            for (int i = 0; i < allItemList.getSize(); i++) {
                if (currentItem.getName().equals(name)) {
                    System.err.println(v.errItemNameRedundant);
                    name = "";
                    break;
                }
            }
        }

        while (category == 0) {
            category = selectCategory();

            if (category == 0) return;
        }

        while (desc.equals("")) {
            desc = h.inputDonationItemDesc();

            if (desc.equals("")) return;
        }
    }
}
```

```

if (category == 2) {
    while (qty == 0) {
        qty = h.inputDonationItemQty();

        if (qty == 0) return;
    }
}

while (amt == 0.0) {
    amt = h.inputDonationItemAmt();

    if (amt == 0) return;
}

Item newItem = new Item(name, desc, qty, amt);

if (category == 1) {
    moneyList.add(newItem);
    itemCard(moneyList.getLast());
} else if (category == 2) {
    foodBeverageList.add(newItem);
    itemCard(foodBeverageList.getLast());
}

System.out.println("\n The donation item has been created successfully.");
optionYN = h.inputYN("\n Do you want to continue create more item?");

if (optionYN.equals("N")) System.out.println(exitStr);
}

/*
 sub function
*/
// select donation
public static Donation selectDonation() {
    int useID = -1;
    Donation useDonation = null;

    while (useDonation == null) {
        donationViewList(donationList);
        System.out.print("\n Please enter the donation ID you want to select ('1' for
'DN0001')('0' to exit): ");
        useID = h.scanInt();
    }
}

```

```

if (useID == 0) {
    System.out.println(exitStr);
    return null;
} else {
    useDonation = searchDonation(donationList, useID); // search donation
}

if (useDonation == null) { // invalid donation
    System.err.println(v.errIdInput);
}
}

return useDonation;
}

// select donor
public static Donor selectDonor() {
    // initialize variable
    int id = -1;
    Donor selectedDonor = null;

    while (id == -1) {
        id = -1;
        // prompt donor ID input
        while (id == -1) {
            System.out.print("\n Please enter donor ID ('1' for 'DR0001')('0' to exit): ");
            prompt id input
            id = h.nextInt();
        }

        if (id == 0) {
            System.out.println("\n Back to subsystem menu page...\n");
            return null;
        } else {
            selectedDonor = searchDonor(id);

            if (selectedDonor == null) {
                id = -1;
            }
        }
    }

    // reset optionYN to null
    optionYN = "null";

    // prompt confirmation input
    while (optionYN.equals("null")) {

```

```

donorCard(selectedDonor);
System.out.print("\n Please confirm the donor info (Y/N); ");

optionYN = h.scanOptionStr();

// check whether option input is correct
optionYN = v.verifyOptionYN(optionYN) ? optionYN : "null";
}

if (optionYN.equals("N")) id = -1; // check whether confirm the donor detail
}
return selectedDonor;
}

// select donor name
public static Donor selectDonorName() {
    String donorName = "";
    Donor useDonor = new Donor();

    while (donorName.equals("")) {
        System.out.print("\n Enter donor name ('0' to exit): ");
        donorName = scan.nextLine();

        if (donorName.equals("0")) return null;
        if (donorName.equals("")) {
            System.err.println("\n \\" Donor name should not be empty. Please try
again.\\"");
        }
    }

    useDonor.setName(donorName);
    useDonor.setSearchBy("name");

    return useDonor;
}

// select item category
public static int selectCategory() {
    optionInt = -1;

    while (optionInt == -1) {
        categoryMenu();

        // prompt category option input
        System.out.print("\n Please select the category option of donation item ('0' to
back): ");
    }
}

```

```

optionInt = h.scanInt();

switch (optionInt) {
    case 0: {System.out.println(exitStr); return optionInt;}
    case 1:
    case 2: {return optionInt;}
    default: {System.err.println(v.errOptionInt); optionInt = -1; break;}
}
}

return optionInt; // exit
}

// select donation item
public static Item selectItem(ListInterface<Item> useItemList) {
    Item useItem = useItemList.getFirst(); // initialize item node (start from first node)
    int useItemID = -1; // selected item id

    // show item menu and select item
    while (useItemID == -1) {
        if (useItemList.isEmpty()) { // item not found
            System.out.println("\n No item here");
        } else {
            donationItemList(useItemList);

            // prompt category option input
            System.out.print("\n Please select the ID of donation item ('1' for 'IM0001')('0'
to exit): ");
        }
    }

    useItemID = h.scanInt();

    // validate item ID selected
    if (useItemID == 0) {
        return null;
    } else {
        useItem = searchItem(useItemList, useItemID);

        if (useItem == null) {
            System.err.println(v.errIdInput);
            useItemID = -1;
        }
    }
}

return useItem;
}

```

```

// select donation item name
public static Item selectItemName(ListInterface<Item> useItemList) {
    String useName = "";
    Item useItem = new Item();

    while (useName.equals("")) {
        System.out.print("\n Please enter donated item name ('0' to exit): ");
        useName = scan.nextLine();

        if (useName.equals("0")) return null;
        if (useName.equals("")) {
            System.err.println(v.errItemNameEmpty);
        }
    }
    useItem.setName(useName);
    return useItem;
}

// select stock in item
public static StockIn selectStockIn(ListInterface<StockIn> useStockInList) {
    int useStockInID = -1;
    StockIn useStockIn = null;

    stockInViewList(useStockInList); // display stock in list

    while (useStockIn == null) {
        // prompt stock in id
        System.out.print("\n Enter the stock in ID that you want to remove ('1' for
'IN0001')('0' to exit): ");
        useStockInID = h.scanInt();

        if (useStockInID == 0) {
            return null;
        } else {
            useStockIn = searchStockIn(useStockInList, useStockInID);
        }
    }
    return useStockIn;
}

// select date
public static LocalDate[] selectDonationDate() {
    LocalDate[] dateRange = new LocalDate[2];
    String startDateStr = "", endDateStr = "";

```

```

while (startDateStr.equals("")) {
    // prompt start date
    while (startDateStr.equals("")) {
        System.out.print("\n Enter donation start date (yyyy-mm-dd) ('0' to exit): ");
        startDateStr = scan.nextLine();

        if (startDateStr.equals("0")) { // exit
            return null;
        } else {
            dateRange[0] = v.verifyDonationDate(startDateStr);
            if (dateRange[0] == null) startDateStr = "";
        }
    }

    // prompt start date
    while (endDateStr.equals("")) {
        System.out.print("\n Enter donation end date (yyyy-mm-dd) ('0' to exit): ");
        endDateStr = scan.nextLine();

        if (endDateStr.equals("0")) { // back
            startDateStr = "";
            endDateStr = "";
            break;
        } else {
            dateRange[1] = v.verifyDonationDate(endDateStr);
            if (dateRange[1] == null) { // invalid end date input
                endDateStr = "";
                continue;
            }

            if (!dateRange[1].isAfter(dateRange[0]) &&
                !dateRange[1].isEqual(dateRange[0])) { // check whether the end date is after the start date
                System.err.println("\n\"End date must be after or equal to the start date.
Please try again.\\"");
                endDateStr = "";
            }
        }
    }

    return dateRange;
}

// search donation by id
public static void searchByDonationID() {
    Donation useDonation = null;
}

```

```

while (useDonation == null) {
    System.out.print("\n Please enter the donation ID ('1' for 'DN0001')('0' to exit): ");
    int useID = h.scanInt();

    if (useID == 0) {
        System.out.println(exitStr);
        return;
    } else {
        useDonation = searchDonation(donationList, useID); // search donation
object

        if (useDonation == null) { // invalid donation
            return;
        } else {
            donationCard(useDonation);
        }
    }
    optionYN = h.inputYN("\n Do you want to continue search other donation by
donation ID?");
    if (optionYN.equals("Y")) useDonation = null;
    else System.out.println(exitStr);
}
}

// search donation by donor id
public static void searchByDonorID() {
    Donor useDonor = null;
    Donation useDonation = null;
    ListInterface<Donation> useDonationList = new DoublyLinkedList<>();

    while (useDonation == null) {
        System.out.print("\n Please enter the donor ID ('1' for 'DR0001')('0' to exit): ");
        int useID = h.scanInt();

        if (useID == 0) {
            System.out.println(exitStr);
            return;
        } else {
            useDonor = searchDonor(useID);

            if (useDonor == null) { // invalid donation
                return;
            } else {
                useDonation = new Donation();
                useDonation.setDonor(useDonor);
                useDonationList = h.filterDonationList(donationList, "donorID",

```

```

useDonation);
}

if (useDonationList.isEmpty()) {
    System.out.print("\n No donation found.");
} else {
    viewDonationList(useDonationList);
}
}

optionYN = h.inputYN(" Do you want to continue search other donation by donor
ID?");

if (optionYN.equals("Y")) useDonation = null;
else System.out.println(exitStr);
}
}

// prompt user to enter item quantity / amount to donate and return stockIn object
public static StockIn makeStockIn(int useCategory, Item useItem) {
    optionYN = "N";

    if (useCategory == 1) { // money category
        // initialize money amount and validation status
        double donationMoney = -1;

        // reset optionYN
        optionYN = "N";

        while (optionYN.equals("N")) {
            optionYN = "null"; // reset optionYN
            donationMoney = -1;

            // prompt donation money amount input
            while (donationMoney == -1) {
                System.out.print("\n Please enter the donation amount (RM) ('0' to exit): ");
                donationMoney = h.scanDouble();

                if (donationMoney == 0.00) return null; // exit

                donationMoney = v.verifyMoneyAmt(donationMoney) ? donationMoney : -1;
            }
        }

        // confirmation dialog
        optionYN = donationConfirmation(useItem, String.format("RM %.2f",
donationMoney));
    }
}

```

```

        return new StockIn(useItem, 0, donationMoney); // return object
    } else { // other items
        // initialize money amount and validation status
        int donationQty = -1;

        // reset optionYN
        optionYN = "N";

        while (optionYN.equals("N")) {
            optionYN = "null"; // reset optionYN
            donationQty = -1;

            // prompt donation item quantity input
            while (donationQty == -1) {
                System.out.print("\n Please enter the donation quantity ('0' to exit): ");
                donationQty = h.nextInt();

                if (donationQty == 0) return null;

                donationQty = v.verifyQty(donationQty) ? donationQty : -1; // check
                whether the Qty amount is valid
            }

            // confirmation dialog
            optionYN = donationConfirmation(useItem, String.format("%s", donationQty));
        }

        return new StockIn(useItem, donationQty, 0); // return object
    }
}

// summarize donation
public static void summarizeDonation(Donor useDonor, ListInterface<StockIn>
useStockInList) {
    // Summarize donation
    System.out.println("\n Summarize Donation");
    System.out.println("\n ======");
}

// Prompt donation notes
String notes = "";

while (notes.equals("") || notes.length() > 30) {
    System.out.print("\n Any notes you want to write here? ('-' if no comment): \n ");
    notes = scan.nextLine();

    if (notes.equals("")) {

```

```

        System.err.println("\n Invalid notes. Please fill in the notes. If no notes, please
enter '-'.");
    } else if (notes.length() > 30) {
        System.err.println("\n The notes should not exceed 30 characters.");
    }
}

stockInList = stockInList.merge(useStockInList);
donationList.add(new Donation(useDonor, useStockInList, notes, LocalDate.now()));
// update donationList

// update moneyItemList or foodBeverageItem
StockIn useStockIn = useStockInList.getFirst();

for (int i = 0; i < useStockInList.getSize(); i++) {
    Item moneyItem = moneyList.search(useStockIn.getItem());
    Item foodBeverageItem = foodBeverageList.search(useStockIn.getItem());

    if (moneyItem != null) { // update moneyItem
        moneyItem.setAmount(moneyItem.getAmount() + useStockIn.getAmt());
    } else if (foodBeverageItem != null) { // update foodBeverageItem
        foodBeverageItem.setQty(foodBeverageItem.getQty() + useStockIn.getQty());
    }
    useStockIn = useStockInList.getNextData(useStockIn);
}

donationCard(new Donation(useDonor, useStockInList, notes, LocalDate.now()));
}

// view donation list
public static void viewDonationList(ListInterface<Donation> useDonationList) {
    Donation useDonation = useDonationList.getFirst();
    int useID = -1;

    // show donation list and prompt donation id for more details
    optionInt = 0; // reset variable value
    while (optionInt == 0) {
        donationViewList(useDonationList);

        // prompt donation id to get more details
        useID = -1;

        while (useID == -1) {
            System.out.print("\n You may enter donation ID to view more details of
donation ('1' for 'DN0001')('0' to exit): ");
            useID = h.scanInt();
        }
    }
}

```

```

if (useID == 0) {
    System.out.println(exitStr);
    return;
} else {
    useDonation = searchDonation(useDonationList, useID);

    if (useDonation == null) useID = -1;
}
}

// donation is found, show donationCard and donorCard
optionInt = -1;
while (optionInt == -1) {
    donationCard(useDonation);
    donorCard(useDonation.getDonor());

    // prompt for further action
    System.out.print("\n 1) View next donation (based on ID)"
        + "\n 2) View previous donation (based on ID)"
        + "\n 0) Back\n"
        + "\n Select option: ");

    optionInt = h.scanInt();

    switch (optionInt) {
        case 1:
            if (useDonationList.getNextData(useDonation) != null) {
                useDonation = useDonationList.getNextData(useDonation);
            } else {
                System.out.println("\n This is already the last donation.");
            }
            optionInt = -1;
            break;
        case 2:
            if (useDonationList.getPreviousData(useDonation) != null) {
                useDonation = useDonationList.getPreviousData(useDonation);
            } else {
                System.out.println("\n This is already the first donation.");
            }
            optionInt = -1;
            break;
        case 0:
            optionInt = 0;
            break;
        default:
            optionInt = -1;
            System.err.println(v.errOptionInt);
    }
}
}

```

```
        }
    }
}

// view donation list by donor
public static void viewDonationListByCriterias() {
    // initialize variable
    Donation useDonation = new Donation();
    String filterBy = "";
    Donor useDonor = null;
    ListInterface<StockIn> useStockInList = new DoublyLinkedList<>();
    StockIn useStockIn = new StockIn();
    Item useItem = null;
    LocalDate[] dateRange = null;

    optionYN = "Y";
    while (optionYN.equals("Y")) {
        optionInt = -1;
        while (optionInt == -1) {
            donationFilterMenu();
            optionInt = h.scanInt();
            switch (optionInt) {
                case 0: // exit
                    System.out.println(exitStr);
                    return;
                case 1: // filter by donor ID
                    filterBy = "donorID";
                    useDonor = selectDonor();
                    break;
                case 2: // filter by donor name
                    filterBy = "donorName";
                    useDonor = selectDonorName();
                    break;
                case 3: // filter by item ID
                    filterBy = "itemID";
                    useItem = selectItem(moneyList.merge(foodBeverageList));
                    break;
                case 4: // filter by donated item name
                    filterBy = "itemName";
                    useItem = selectItemName(moneyList.merge(foodBeverageList));
                    break;
                case 5: // filter by donation date
                    filterBy = "donationDate";
                    dateRange = selectDonationDate();
                    break;
                default: // invalid input
            }
        }
    }
}
```

```

        System.err.println(v.errOptionInt);
        optionInt = -1;
    }

    if (optionInt == 1 || optionInt == 2) {
        if (useDonor == null) {optionInt = -1; continue;}
        useDonation.setDonor(useDonor);
    } else if (optionInt == 3 || optionInt == 4) {
        if (useItem == null) {optionInt = -1; continue;}
        useStockIn.setItem(useItem);
        useStockInList.add(useStockIn);
        useDonation.setStockIn(useStockInList);
    } else if (optionInt == 5) {
        if (dateRange == null) {optionInt = -1; continue;}
        useDonation.setStartDate(dateRange[0]);
        useDonation.setEndDate(dateRange[1]);
    }
}

// filter and get the donation list
ListInterface<Donation> useDonationList = h.filterDonationList(donationList,
filterBy, useDonation);

if (useDonationList.isEmpty()) { // no donation found
    System.out.println("\n \"Donation not found.\"");
} else { // donation found
    viewDonationList(useDonationList);
}

// prompt for continue viewing other donation list
optionYN = h.inputYN("\n Do you want to continue viewing other donation
list?");

if (optionYN.equals("N")) {
    System.out.println(exitStr);
}
}

// view donation item list
public static void viewDonationItemList() {
    ListInterface<Item> allItemList = moneyList.merge(foodBeverageList);
    Item useItem = null;
    int useID;

    // display donation item list
    optionInt = 0; // reset variable
}

```

```
while (optionInt == 0) {
    donationItemList(allItemList);

    // prompt item id to get more details of the item
    useID = -1;

    while (useID == -1) {
        System.out.print("\n You may enter item ID to view more details of item ('1' for
'M0001')('0' to exit): ");
        useID = h.scanInt();

        if (useID == 0) { // exit
            System.out.println(exitStr);
            return;
        } else {
            useItem = searchItem(allItemList, useID);

            if (useItem == null) useID = -1;
        }
    }

    optionInt = -1;
    while (optionInt == -1) {
        itemCard(useItem);

        System.out.print("\n 1) View next item (based on ID)"
                        + "\n 2) View previous item (based on ID)"
                        + "\n 0) Back\n"
                        + "\n Select option: ");

        optionInt = h.scanInt();

        switch (optionInt) {
            case 1:
                if (allItemList.getNextData(useItem) != null) useItem =
                    allItemList.getNextData(useItem);
                else System.out.println("\n This is already the last donation.");

                optionInt = -1;
                break;
            case 2:
                if (allItemList.getPreviousData(useItem) != null) useItem =
                    allItemList.getPreviousData(useItem);
                else System.out.println("\n This is already the first donation.");

                optionInt = -1;
                break;
        }
    }
}
```

```

        case 0:
            System.out.println(exitStr);
            break;
        default:
            System.err.println(v.errOptionInt);
            optionInt = -1;
    }
}

}

// update donation (switch donor)
public static void updateDonor(Donation useDonation) {
    Donor useDonor = null;
    int useID = -1;
    optionInt = -1;
    optionYN = "null";

    while (optionInt == -1) {
        useID = -1;
        while (useID == -1) {
            // show current donor
            System.out.print("\n Current donor: ");
            donorCard(useDonation.getDonor());

            // prompt donor to be switched to
            System.out.print("\n Enter donor ID to switch the donor ('0' to exit): ");

            useID = h.scanInt();

            if (useID == 0) { // exit
                System.out.println(exitStr);
                return;
            } else {
                useDonor = searchDonor(useID); // search donor
                if (useDonor == null) useID = -1; // invalid donor
            }
        }

        optionYN = "null";
        while (optionYN.equals("null")) {
            donorCard(useDonor);
            // prompt switch donor confirmation
            optionYN = h.inputYN("\n Are you sure to switch to this donor (" +
useDonor.formatID() + ") for this donation " + useDonation.formatID() + "?");
        }
    }
}

```

```

        if (optionYN.equals("Y")) {
            useDonation.setDonor(useDonor);
            donationUpdatePart(useDonation); // update donation
            optionInt = 0;
        } else optionInt = -1;
    }
}

// update donation (modify quantity of donation item)
public static void updateQty(Donation useDonation) {
    ListInterface<StockIn> useStockInList = useDonation.getStockIn();
    StockIn useStockIn = null;
    int useID = -1;
    int newQty = -1;
    optionYN = "Y";

    while (optionYN.equals("Y")) {
        useID = -1;
        while (useID == -1) {
            // display stock in list
            stockInViewList(useStockInList);

            // prompt stock in ID to be modified
            System.out.print("\n Select the stock in ID to modify the quantity ('0' to exit):
");
            useID = h.scanInt();

            if (useID == 0) { // exit
                System.out.println(exitStr);
                return;
            } else {
                useStockIn = searchStockIn(useStockInList, useID);

                if (useStockIn == null) { // stockIn object not found
                    useID = -1;
                } else if (moneyList.search(useStockIn.getItem()) != null) { // item in money
category cannot be updated
                    System.err.println("\n Category 'Money' cannot be updated.");
                    useID = -1;
                }
            }
        }

        optionYN = "N";
        while (optionYN.equals("N")) {
    }
}

```

```

newQty = -1;
while (newQty == -1) {
    stockInCard(useStockIn); // display stockIn detail

    // prompt new quantity input
    System.out.print("\n Enter the new quantity ('0' to exit): ");
    newQty = h.scanInt();

    if (newQty == 0) break;
    else if (!v.verifyQty(newQty)) newQty = -1; // invalid new quantity
}
if (newQty == 0) break;

optionYN = h.inputYN("\n Are you sure to change the quantity of stock (" +
useStockIn.formatID() + ") to " + newQty + "?");

if (optionYN.equals("Y")) {
    Item useItem = foodBeverageList.search(useStockIn.getItem());
    useItem.setQty(useItem.getQty() - useStockIn.getQty() + newQty);
    foodBeverageList.update(useItem);

    stockInList.search(useStockIn).setQty(newQty);

    // show result
    donationCard(donationList.search(useDonation));
    donorCard(donationList.search(useDonation).getDonor());
    System.out.println("\n The donation detail has been updated successfully.");
}
}

optionYN = h.inputYN("\n Do you want to update the quantity of other
stockIn?");

if (optionYN.equals("N")) {
    System.out.println(exitStr);
}
}

// update donation (add new item)
public static void addNewItem(Donation useDonation) {
    ListInterface<Item> useDonationItemList = new DoublyLinkedList<>();
    ListInterface<Item> tempMoneyList = new DoublyLinkedList<>();
    ListInterface<Item> tempFoodBeverageList = new DoublyLinkedList<>();
    ListInterface<StockIn> useStockInList = new DoublyLinkedList<>();
    Item useItem = null;
    StockIn useStockIn = null;
}

```

```

int useCategory = -1;

// get all item from the donation stock in
StockIn tempStockIn = useDonation.getStockIn().getFirst();
for (int i = 0; i < useDonation.getStockIn().getSize(); i++) {
    useDonationItemList.add(tempStockIn.getItem());
    tempStockIn = useDonation.getStockIn().getNextData(tempStockIn);
}

tempMoneyList = useDonationItemList.minus(moneyList); // get available money
item
tempFoodBeverageList = foodBeverageList.minus(useDonationItemList); // get
available food beverage item

while (useCategory == -1) {
    useCategory = selectCategory(); // select category

    if ((tempMoneyList.isEmpty() && useCategory == 1) ||
(tempFoodBeverageList.isEmpty() && useCategory == 2)) {
        System.out.println("\n No available item can be added");
        useCategory = -1;
        continue;
    }

    optionYN = "Y";
    while (optionYN.equals("Y")) {
        switch (useCategory) {
            case 1: // money category
                useItem = selectItem(tempMoneyList);
                break;
            case 2: // food beverage category
                useItem = selectItem(tempFoodBeverageList);
                break;
            case 0: // exit
                return;
        }
    }

    if (useItem == null) {
        optionYN = "N";
        useCategory = -1;
    } else {
        useStockIn = makeStockIn(useCategory, useItem); // create StockIn object
        if (useStockIn != null) {
            useStockInList.add(useStockIn);

            if (useCategory == 1) tempMoneyList.remove(useStockIn.getItem()); // remove selected item from tempMoneyList
        }
    }
}

```

```

        if (useCategory == 2)
tempFoodBeverageList.remove(useStockIn.getItem()); // remove selected item from
tempFoodBeverageList
            useDonationItemList.add(useStockIn.getItem());
        }

// prompt for adding more item
optionYN = h.inputYN("\n Do you want to add more item?");
if (optionYN.equals("N")) {
    stockInList = stockInList.merge(useStockInList); // update the stockInList

    tempStockIn = useStockInList.getFirst();
    for (int i = 0; i < useStockInList.getSize(); i++) {
        Item useMoneyItem = moneyList.search(tempStockIn.getItem());
        Item useFoodBeverageItem =
foodBeverageList.search(tempStockIn.getItem());
        if (useMoneyItem != null) { // selected item is belongs to money
category
            useMoneyItem.setAmount(useMoneyItem.getAmount() +
tempStockIn.getItem().getAmount());
            moneyList.update(useMoneyItem); // update moneyList
        } else if (useFoodBeverageItem != null) { // selected item is belongs to
foodBeverage category
            useFoodBeverageItem.setQty(useFoodBeverageItem.getQty() +
tempStockIn.getItem().getQty());
            foodBeverageList.update(useFoodBeverageItem); // update
foodBeverageList
        }
    }
}

useDonation.setStockIn(useDonation.getStockIn().merge(useStockInList));
donationUpdatePart(useDonation); // update donationList
}
}
}

}

// remove donated item (remove item)
public static void removeItem(Donation useDonation) {
ListInterface<StockIn> useStockInList = useDonation.getStockIn();
StockIn useStockIn = null;

while (useStockIn == null) {
    if (useStockInList.getSize() <= 1) {
System.err.println("\n You are not allowed to remove the donated item since

```

```

each donation must have at least 1 donation item.");
    return;
} else {
    useStockIn = selectStockIn(useStockInList);

    if (useStockIn == null) return;
    if (moneyList.search(useStockIn.getItem()) != null) {
        System.err.println("\n You are not allowed to remove the 'Money' category
item.");
        useStockIn = null;
        continue;
    }

    stockInCard(useStockIn);

    optionYN = h.inputYN("\n Are you sure to remove this item (" +
useStockIn.formatID() + ") from this donation?");
    if (optionYN.equals("Y")) {
        Item deductMoneyItem = moneyList.search(useStockIn.getItem());
        Item deductFoodBeverageItem =
foodBeverageList.search(useStockIn.getItem());

        if (deductMoneyItem != null) {
            deductMoneyItem.setAmount(deductMoneyItem.getAmount() -
useStockIn.getAmt());
            moneyList.update(deductMoneyItem); // update moneyList
        } else if (deductFoodBeverageItem != null) {
            deductFoodBeverageItem.setQty(deductFoodBeverageItem.getQty() -
useStockIn.getQty());
            foodBeverageList.update(deductFoodBeverageItem); // update
foodBeverageList
        }
        stockInList.remove(useStockIn); // update stockInList

        useStockInList.remove(useStockIn);
        useDonation.setStockIn(useStockInList);
        donationList.search(useDonation).setStockIn(useStockInList); // update
donationList

        System.out.print("\n The item has been removed successfully.");
    }
}

optionYN = h.inputYN("\n Do you want to remove other item within this
donation?");
if (optionYN.equals("N")) {
    System.out.println(exitStr);
}

```

```

        return;
    } else useStockIn = null;
}

}

// update donation (update notes)
public static void updateNotes(Donation useDonation) {
    String useNotes = "";
    System.out.println("\n Current Notes:"
        + "\n " + useDonation.getNotes());

    while (useNotes.equals("")) {
        System.out.print("\n Enter new notes ('0' to exit): \n ");
        useNotes = scan.nextLine();

        if (useNotes.equals("")) {
            System.err.println("\n Invalid notes. Please fill in the notes. If no notes, please
enter '-'");
        } else if (useNotes.equals("0")) {
            return;
        }
    }

    useDonation.setNotes(useNotes);
    donationUpdatePart(useDonation);
}

// top 10 major donation report
public static void top10DonationReport() {
    Donation useDonation = new Donation();
    LocalDate[] dateRange = selectDonationDate();
    if (dateRange == null) return;

    useDonation.setStartDate(dateRange[0]);
    useDonation.setEndDate(dateRange[1]);

    // filter and get the donation list
    ListInterface<Donation> useDonationList = h.filterDonationList(donationList,
"donationDate", useDonation);
    if (useDonationList.isEmpty()) {
        System.out.println("\n No data found.");
    } else {
        useDonationList = calcDonationAmt(useDonationList);
        useDonationList = donationSort(useDonationList, "donationTotalAmt");

        useDonation = useDonationList.getFirst();
    }
}

```

```

System.out.print("\n\t\t\t\t\tTop 10 Major Donation Report"
+ "\n"
=====
=====);
System.out.format("\n | %-15s | %-15s | %-40s | %-20s | %-20s |", "Donation
ID", "Donor ID", "Donor Name", "Total Amount", "Donation Date");
System.out.print("\n"
=====
=====);
for (int i = 0; i < 10; i++) {
    if (useDonation == null) break;
    System.out.format("\n | %-15s | %-15s | %-40s | %-20s | %-20s |", "", "", "", "",
", ");
    System.out.print(String.format("\n | %-15s | %-15s | %-40s | RM %17.2f |
%-20s |"
, useDonation.formatID()
, useDonation.getDonor().formatID()
, useDonation.getDonor().getName()
, useDonation.getTotalAmount()
, useDonation.formatDonationDate()));
    useDonation = useDonationList.getNextData(useDonation);
}
System.out.print("\n"
=====
=====);
}
}

// top 10 donated items report
public static void top10DonatedItemReport() {
    Donation useDonation = new Donation();
    int useCategory = -1;
    LocalDate[] dateRange = null;

    while (dateRange == null) {
        useCategory = selectCategory(); // select category
        if (useCategory == 0) return;
        dateRange = selectDonationDate(); // select donation donation date range
    }

    useDonation.setStartDate(dateRange[0]);
    useDonation.setEndDate(dateRange[1]);

    // filter and get the donation list
    ListInterface<Donation> useDonationList = h.filterDonationList(donationList,

```

```

"donationDate", useDonation);
if (useDonationList.isEmpty()) {
    System.out.println("\n No data found.");
    return;
}
ListInterface<Item> resultList = calcDonatedItem(useDonationList, useCategory);
if (useDonationList.isEmpty()) {
    System.out.println("\n No data found.");
    return;
}

Item currentItem = resultList.getFirst(); // Get the first item in the resultList

for (int i = 0; i < resultList.getSize(); i++) { // Iterate through each item
    ListInterface<Item> templist = new DoublyLinkedList<>(); // Temporary list
    to hold item and donor contributions

    Donation currentDonation = useDonationList.getFirst(); // Start with the first
    donation

    for (int j = 0; j < useDonationList.getSize(); j++) { // Iterate through each donation
        // Find stock-in lists for the specific item and donor
        ListInterface<StockIn> itemStockInList = findItemStockIn(useDonationList,
        currentItem);
        ListInterface<StockIn> donorStockInList = findDonorStockIn(useDonationList,
        currentDonation.getDonor());

        // Intersect both lists to find stock-in records for the specified item and donor
        ListInterface<StockIn> finalStockInList =
        itemStockInList.intersect(donorStockInList);
        if (!finalStockInList.isEmpty()) stockInViewList(finalStockInList);

        if (!finalStockInList.isEmpty()) {
            Item templItem = new Item();
            if (useCategory == 1) {
                templItem.setAmount(sumStockInAmt(finalStockInList));
            } else {
                templItem.setQty(sumStockInQty(finalStockInList)); // Sum the stock-in
                quantities
            }
            templItem.setContributor(currentDonation.getDonor()); // Set the donor as
            the contributor
            templItemList.add(templItem); // Add the temporary item to the list
        }

        currentDonation = useDonationList.getNextData(currentDonation); // Move to
        the next donation
    }
}

```

```

}

if (!tempItemList.isEmpty()) {
    // Find the donor who contributed the most to this item
    Item bestItem = findBestContributorItem(tempItemList, useCategory);
    currentItem.setContributor(bestItem.getContributor()); // Set the best
contributor for the current item
}

currentItem = resultList.getNextData(currentItem); // Move to the next item in
the resultList
}

if (useCategory == 1) {
    resultList = donatedItemSort(resultList, "moneyAmt");
    Item useItem = resultList.getFirst();

    System.out.print("\n\t\t\t\tTop 10 Donated Item Report (Money)"
        + "\n"
        =====
        ======");
    System.out.format("\n | %-10s | %-20s | %-20s | %-40s |", "Item ID", "Item
Name", "Total Amount", "Major Contributor");
    System.out.print("\n"
        =====
        ======");

    for (int i = 0; i < 10; i++) {
        if (useItem == null) break;
        System.out.format("\n | %-10s | %-20s | %-20s | %-40s |", "", "", "", "");
        System.out.print(String.format("\n | %-10s | %-20s | RM %17.2f | %-40s |"
            , useItem.formatID()
            , useItem.getName()
            , useItem.getTotalAmt()
            , useItem.getContributor().getName()));
        useItem = resultList.getNextData(useItem);
    }
    System.out.print("\n"
        =====
        ======"\\n");
}

} else if (useCategory == 2) {
    resultList = donatedItemSort(resultList, "foodBeverageQty");
    Item useItem = resultList.getFirst();

    System.out.print("\n\t\t\t\tTop 10 Donated Item Report (Food & Beverage)"
        + "\n"
        =====
        ======");
}

```

```

=====
System.out.format("\n | %-10s | %-30s | %-20s | %-20s | %-40s |", "Item ID",
"Item Name", "Quantity", "Total Amount", "Major Contributor");
System.out.print("\n
=====
=====");
for (int i = 0; i < 10; i++) {
    if (useItem == null) break;
    System.out.format("\n | %-10s | %-30s | %-20s | %-20s | %-40s |", "", "", "", "",
"");
    System.out.print(String.format("\n | %-10s | %-30s | %-20d | RM %17.2f |
%-40s |"
        , useItem.formatID()
        , useItem.getName()
        , useItem.getQty()
        , useItem.getTotalAmt()
        , useItem.getContributor().getName())));
    useItem = resultList.getNextData(useItem);
}
System.out.print("\n
=====
=====\\n");
}
}

// top 10 critical shortage report
public static void top10CriticalShortageReport() {
    int useCategory = -1;

    while (useCategory == -1) {
        useCategory = selectCategory();
        if (useCategory == 0) return;

        if (useCategory == 1) {
            double amtThreshold = -1;

            while (amtThreshold == -1) {
                System.out.print("\n Enter the threshold amount of donation item (RM)('0'
to exit): ");
                amtThreshold = h.scanDouble();

                if (amtThreshold == 0) {
                    useCategory = -1;
                    break;
                } else if (amtThreshold < 0.01 || amtThreshold > 999999.99) {
                    System.err.println(v.errItemAmtRange);
                }
            }
        }
    }
}

```

```
        amtThreshold = -1;
    }
}
if (useCategory == -1) continue;

Item useItem = new Item();
useItem.setAmount(amtThreshold);
ListInterface<Item> resultList = h.filterItemList(moneyList, "amtThreshold",
useItem);

if (resultList.isEmpty()) {
    System.out.println("\n All items have exceeded the threshold quantity");
} else {
    resultList = donatedItemSort(resultList, "thresholdAmt");

    useItem = resultList.getFirst();

    System.out.print("\n\t\t\t\tTop 10 Critical Shortage Report (Money)\n"
                    + "\n Threshold Amount (RM): " + String.format("%.2f",
amtThreshold)
                    + "\n"
=====);
    System.out.format("\n | %-10s | %-30s | %-30s | %-30s |", "Item ID", "Item
Name", "Current Amount", "Amount Required");
    System.out.print("\n"
=====);

    for (int i = 0; i < 10; i++) {
        if (useItem == null) break;
        System.out.format("\n | %-10s | %-30s | %-30s | %-30s |", "", "", "", "");
        System.out.print(String.format("\n | %-10s | %-30s | RM %27.2f | RM
%27.2f |"
                                         , useItem.formatID()
                                         , useItem.getName()
                                         , useItem.getAmount()
                                         , amtThreshold - useItem.getAmount()));
        useItem = resultList.getNextData(useItem);
    }

    System.out.print("\n"
=====);
}
} else {
    int qtyThreshold = -1;
```

```

while (qtyThreshold == -1) {
    System.out.print("\n Enter the threshold quantity of donation item ('0' to
exit): ");
    qtyThreshold = h.scanInt();

    if (qtyThreshold == 0) {
        useCategory = -1;
        break;
    } else if (qtyThreshold < 1 || qtyThreshold > 99999999) {
        System.err.println("\n \"Quantity of donation item should between 1 and
999,999,999.\"");
        qtyThreshold = -1;
    }
}
if (useCategory == -1) continue;

Item useItem = new Item();
useItem.setQty(qtyThreshold);
ListInterface<Item> resultList = h.filterItemList(foodBeverageList,
"qtyThreshold", useItem);

if (resultList.isEmpty()) {
    System.out.println("\n All items have exceeded the threshold quantity");
} else {
    resultList = donatedItemSort(resultList, "thresholdQty");
    useItem = resultList.getFirst();

    System.out.print("\n\t\t\tTop 10 Critical Shortage Report (Food &
Beverage)\n"
                    + "\n Threshold Quantity: " + qtyThreshold
                    + "\n"
                    + "=====");
    System.out.format("\n | %-10s | %-30s | %-20s | %-20s | ", "Item ID", "Item
Name", "Current Quantity", "Quantity Required");
    System.out.print("\n"
                    + "=====");
    for (int i = 0; i < 10; i++) {
        if (useItem == null) break;
        System.out.format("\n | %-10s | %-30s | %-20s | %-20s | ", "", "", "", "");
        System.out.print(String.format("\n | %-10s | %-30s | %-20d | %-20d | "
                , useItem.formatID()
                , useItem.getName()
                , useItem.getQty())
    }
}

```

```

                , qtyThreshold - uselItem.getQty()));
        uselItem = resultList.getNextData(uselItem);
    }

    System.out.print("\n
=====
=====\\n");
}

}

}

/*
    display menu/listing function
*/
// main menu
public static void mainMenu() {
    System.out.print("\n\t\tDonation Subsystem\n\n"
        + " +-----+\n"
        + " | Function      | Option   |\n"
        + " +=====+=====+\n"
        + " | Add new donation | 1 |\n"
        + " +-----+\n"
        + " | View donation | 2 |\n"
        + " +-----+\n"
        + " | Search donation | 3 |\n"
        + " +-----+\n"
        + " | Update donation | 4 |\n"
        + " +-----+\n"
        + " | Remove donation | 5 |\n"
        + " +-----+\n"
        + " | Donation Report | 6 |\n"
        + " +-----+\n"
        + " | Create new donation item | 7 |\n"
        + " +-----+\n"
        + " | Exit           | 0 |\n"
        + " +-----+\n"
        + " Please select function (0 - 7): ");
}

// category menu
public static void categoryMenu() {
    // show category menu
    System.out.print("\n +-----+
        + "\\n | Category      | Option |"

```

```

+ "\n +=====+====+\n"
+ "\n | Money      | 1  |"
+ "\n +-----+-----+"
+ "\n | Food and Beverages | 2  |"
+ "\n +-----+");}
}

// view donation menu
public static void donationViewMenu() {
    System.out.print("\n +-----+"
+ "\n | Type of list      | Option |"
+ "\n +=====+====+\n"
+ "\n | Donation list      | 1  |"
+ "\n +-----+-----+"
+ "\n | Donation list By criteria | 2  |"
+ "\n +-----+-----+"
+ "\n | Donation item list    | 3  |"
+ "\n +-----+\n"
+ "\n Please select list ('0' to exit): ");
}

// search donation menu
public static void donationSearchMenu() {
    System.out.print("\n +-----+"
+ "\n | Search by      | Option |"
+ "\n +=====+====+\n"
+ "\n | Donation ID      | 1  |"
+ "\n +-----+-----+"
+ "\n | Donor ID      | 2  |"
+ "\n +-----+-----+\n"
+ "\n Please select list ('0' to exit): ");
}

// update donation menu
public static void donationUpdateMenu() {
    System.out.print("\n +-----+"
+ "\n | Update      | Option |"
+ "\n +=====+====+\n"
+ "\n | Switch donor      | 1  |"
+ "\n +-----+-----+"
+ "\n | Modify quantity of current item | 2  |"
+ "\n +-----+-----+"
+ "\n | Add new item      | 3  |"
+ "\n +-----+-----+"
+ "\n | Remove item      | 4  |"
+ "\n +-----+-----+"
+ "\n | Update notes      | 5  |"

```

```

        + "\n +-----+\n");
    }

// donation list filter menu
public static void donationFilterMenu() {
    System.out.print("\n +-----+"
        + "\n | Filter By      | Option |"
        + "\n +=====+=====+"
        + "\n | Donor ID       | 1     |"
        + "\n +-----+-----+"
        + "\n | Donor Name     | 2     |"
        + "\n +-----+-----+"
        + "\n | Donated Item ID | 3     |"
        + "\n +-----+-----+"
        + "\n | Donated Item Name | 4     |"
        + "\n +-----+-----+"
        + "\n | Donation Date   | 5     |"
        + "\n +-----+-----+\n"
        + "\n Please select a filter ('0' to exit): ");
}

// donation report menu
public static void donationReportMenu() {
    System.out.print("\n +-----+"
        + "\n | Report          | Option |"
        + "\n +=====+=====+"
        + "\n | Top 10 Major Donation Report | 1     |"
        + "\n +-----+-----+"
        + "\n | Top 10 Donated Items Report  | 2     |"
        + "\n +-----+-----+"
        + "\n | Top 10 Critical Shortages Report | 3     |"
        + "\n +-----+-----+\n"
        + "\n Please select a report ('0' to exit): ");
}

// donation view card
public static void donationCard(Donation useDonation) {
    // display donation detail card
    System.out.print("\n +-----+");
    System.out.format("\n | %-14s | %-30s |", "Donation ID:", useDonation.formatID());
    System.out.print("\n +-----+");
    System.out.format("\n | %-47s |", "Stock In Item:");

    // display stock in detail
    StockIn useStockIn = useDonation.getStockIn().getFirst();

    for (int i = 0; i < useDonation.getStockIn().getSize(); i++) {
}

```

```

String id = useStockIn.formatID();
String name = String.format("%-20s", useStockIn.getItem().getName());
String quantity = String.format("%-10d", useStockIn.getQty());
double amount = useStockIn.getAmt();

if (useStockIn.getAmt() != 0) {
    System.out.format("\n | %-10s %-20s RM %-12.2f |", id, name, amount);
} else {
    System.out.format("\n | %-10s %-20s x %-13s |", id, name, quantity);
}

useStockIn = useDonation.getStockIn().getNextData(useStockIn); // shift to next
stockIn object
}

System.out.print("\n +-----+");
System.out.format("\n | %-14s | %-30s |", "Notes:", useDonation.getNotes());
System.out.print("\n +-----+");
System.out.format("\n | %-14s | %-30s |", "Donation date:",
useDonation.formatDonationDate());
System.out.print("\n +-----+");

// donor view card
public static void donorCard(Donor useDonor) {
    System.out.print("\n\n +-----+");
    System.out.format("\n | %-9s | %-40s |", "Donor ID:", useDonor.formatID());
    System.out.print("\n +-----+");
    System.out.format("\n | %-9s | %-40s |", "Name:", useDonor.getName());
    System.out.print("\n +-----+");
    System.out.format("\n | %-9s | %-40s |", "Type:", useDonor.formatType());
    System.out.print("\n +-----+");
}

// donation item card
public static void itemCard(Item useItem) {
    double totalAmount = 0.0;
    String category = "";

    if (moneyList.search(useItem) != null) {
        category = "Money";
        totalAmount = useItem.getAmount();
    } else if (foodBeverageList.search(useItem) != null) {
        category = "Food and Beverages";
        totalAmount = (double) useItem.getQty() * useItem.getAmount();
    }
}

```

```

System.out.print("\n +-----+");
System.out.format("\n | %-15s | %-50s |", "Item ID:", useItem.formatID());
System.out.print("\n +-----+");
System.out.format("\n | %-15s | %-50s |", "Item Name:", useItem.getName());
System.out.print("\n +-----+");
System.out.format("\n | %-15s | %-50s |", "Category:", category);
System.out.print("\n +-----+");
System.out.format("\n | %-15s | %-50s |", "Description:", useItem.getDesc());
System.out.print("\n +-----+");
System.out.format("\n | %-15s | %-50d |", "Quantity:", useItem.getQty());
System.out.print("\n +-----+");
System.out.format("\n | %-15s | RM %-47.2f |", "Price Per Item:",
useItem.getAmount());
System.out.print("\n +-----+");
System.out.format("\n | %-15s | RM %-47.2f |", "Total Amount:", totalAmount);
System.out.print("\n +-----+");
}

// stockIn card
public static void stockInCard(StockIn useStockIn) {
    System.out.print("\n +-----+");
    System.out.format("\n | %-19s | %-20s |", "Stock In ID:", useStockIn.formatID());
    System.out.print("\n +-----+");
    System.out.format("\n | %-19s | %-20s |", "Stock In Item Name:",
useStockIn.getItem().getName());
    System.out.print("\n +-----+");
    System.out.format("\n | %-19s | %-20s |", "Stock In Quantity:",
useStockIn.getQty());
    System.out.print("\n +-----+");
    System.out.format("\n | %-19s | RM %-17.2f |", "Total Amount:", ((double)
useStockIn.getQty() * useStockIn.getItem().getAmount()));
    System.out.print("\n +-----+");
}

// display item list
public static void donationItemList(ListInterface<Item> useItemList) {
    Item useItem = useItemList.getFirst();

    System.out.print("\n +-----+");
    System.out.format("\n | %-7s | %-20s | %-8s | %-20s |", "Item ID", "Item Name",
"Quantity", "Total Amount (RM)");
    System.out.print("\n +=====+");

    for (int i = 0; i < useItemList.getSize(); i++) {
        String totalAmountStr = "";

```

```

if (moneyList.search(useItem) != null) {
    totalAmountStr = String.format("%s", h.formatMoney(useItem.getAmount()));
} else if (foodBeverageList.search(useItem) != null) {
    totalAmountStr = String.format("%s", h.formatMoney(((double)
useItem.getQty() * useItem.getAmount())));
}
}

System.out.format("\n | %-7s | %-20s | %-8s | %-20s | ", useItem.formatID(),
useItem.getName(), useItem.getQty(), totalAmountStr);

System.out.print("\n +-----+");

useItem = useItemList.getNextData(useItem);
}
}

// view stock in list
public static void stockInViewList(ListInterface<StockIn> useStockInList) {
    StockIn useStockIn = useStockInList.getFirst();

    // display stock in list
    System.out.print("\n
+-----+");
    System.out.format("\n | %-15s | %-30s | %-10s | %-20s | ", "Stock In ID", "Item
Name", "Quantity", "Total Amount (RM)");
    System.out.print("\n
+=====+
=====+");

    for (int i = 0; i < useStockInList.getSize(); i++) {
        String name = String.format("%-20s", useStockIn.getItem().getName());
        int qty = useStockIn.getQty();
        double amount = 0.0;

        if (moneyList.search(useStockIn.getItem()) != null) {
            amount = useStockIn.getAmt();
        } else if (foodBeverageList.search(useStockIn.getItem()) != null) {
            amount = (double) useStockIn.getQty() * useStockIn.getItem().getAmount();
        }

        System.out.format("\n | %-15s | %-30s | %-10d | RM %17.2f |",
useStockIn.formatID(), name, qty, amount);
        System.out.print("\n
+-----+");
    }

    useStockIn = useStockInList.getNextData(useStockIn); // shift to next stockIn
object
}

```

```

    }

}

// view donation list
public static void donationViewList(ListInterface<Donation> useDonationList) {
    Donation useDonation = useDonationList.getFirst();

    // display donation list
    System.out.print("\n
+-----+");
    System.out.format("\n | %-11s | %-40s | %-12s | %-10s |", "Donation ID", "Donor
Name", "Type", "Date");
    System.out.print("\n
=====+=====+=====+=====+
=====+");

    for (int i = 0; i < useDonationList.getSize(); i++) {
        String id = useDonation.formatID();
        String name = String.format("%-40s", useDonation.getDonor().getName());
        String type = String.format("%-12s", useDonation.getDonor().formatType());
        String date = useDonation.formatDonationDate();

        System.out.format("\n | %-11s | %-20s | %-12s | %-10s |", id, name, type, date);
        System.out.print("\n
+-----+-----+-----+-----+");

        useDonation = useDonationList.getNextData(useDonation); // shift to next
donation object
    }
}

/*
 helper tools
 */

// search donation
public static Donation searchDonation(ListInterface<Donation> useDonationList, int
useID) {
    if (useID < 1 || useID > useDonationList.getLast().getId()) { // id out of range
        System.err.println(v.errIdInput);
        return null;
    } else {
        // search for donation detail
        Donation donation = useDonationList.search(new Donation(useID));
    }
}

```

```

        if (donation == null) { // donation not found
            System.err.println("\n Donation ID is invalid. Please try again.");
            return null;
        } else {
            return donation;
        }
    }

// search donor
public static Donor searchDonor(int userID) {
    if (userID < 1 || userID > donorList.getLast().getId()) { // id out of range
        System.err.println(v,errIdInput);
        return null;
    } else {
        // search for donor detail
        Donor donor = donorList.search(new Donor(userID));

        if (donor == null) { // donor not found
            System.err.println("\n Donor ID is invalid. Please try again.");
            return null;
        } else {
            return donor;
        }
    }
}

// search item
public static Item searchItem(ListInterface<Item> useList, int userID) {
    if (userID < 1 || userID > useList.getLast().getId()) { // id out of range
        System.err.println(v,errIdInput);
        return null;
    } else {
        // search for donor detail
        Item item = useList.search(new Item(userID));

        if (item == null) { // donor not found
            System.err.println("\n Donor ID is invalid. Please try again.");
            return null;
        } else {
            return item;
        }
    }
}

// search stock in
public static StockIn searchStockIn(ListInterface<StockIn> useStockInList, int userID) {
}

```

```

if (useID < 1 || useID > useStockInList.getLast().getId()) { // id out of range
    System.out.println("StockIn ID is invalid. Please try again.");
    return null;
} else {
    // search for donor detail
    StockIn stockIn = useStockInList.search(new StockIn(useID));

    if (stockIn == null) { // stockIn not found
        System.out.println("\n StockIn ID is invalid. Please try again.");
        return null;
    } else {
        return stockIn;
    }
}

// donation confirmation
public static String donationConfirmation(Item useItem, String amount) {
    while (optionYN.equals("null")) {
        System.out.print("\n Are you sure to donate " + useItem.getName() + " with
amount " + amount + "? (Y/N): ");
        optionYN = h.scanOptionStr();

        // check whether option input is correct
        optionYN = v.verifyOptionYN(optionYN) ? optionYN : "null";
    }

    return optionYN;
}

// update donation list
public static void donationUpdatePart(Donation updatedDonation) {
    donationList.update(updatedDonation);

    // show result
    donationCard(donationList.search(updatedDonation));
    donorCard(donationList.search(updatedDonation).getDonor());
    System.out.println("\n The donation detail has been updated successfully.");
}

// sorting donation function
public static ListInterface<Donation> donationSort(ListInterface<Donation>
useDonationList, String keyword) {
    ListInterface<Donation> result = new DoublyLinkedList<>();
    Donation currentDonation = new Donation();

    switch (keyword) {

```

```

case "donationTotalAmt":
    currentDonation = useDonationList.getFirst();

    for (int i = 0; i < useDonationList.getSize(); i++) {
        if (result.isEmpty()) { // when the result list is empty
            result.add(currentDonation);
        } else { // there is node inside result list
            Donation temp = result.getFirst();
            boolean isInserted = false;

            for (int j = 0; j < result.getSize(); j++) {
                // check whether currentDonation total amount > result list existing
                donations' total amount
                if (currentDonation.getTotalAmount() > temp.getTotalAmount()) {
                    result.insertBefore(temp, currentDonation);
                    isInserted = true;
                    break;
                }
                temp = result.getNextData(temp); // shift to next donation in result list
            }

            // if currentDonation total amount < all existing donations' total amount in
            result list, add to the last position
            if (!isInserted) result.add(currentDonation);
        }
        currentDonation = useDonationList.getNextData(currentDonation); // shift to
        next donation in useDonationList
    }
    break;
    default:
}
return result;
}

// sorting donated item function
public static ListInterface<Item> donatedItemSort(ListInterface<Item> useItemList,
String keyword) {
    ListInterface<Item> result = new DoublyLinkedList<>();
    Item currentItem = new Item();

    switch (keyword) {
        case "foodBeverageQty":
            currentItem = useItemList.getFirst();

            for (int i = 0; i < useItemList.getSize(); i++) {
                if (result.isEmpty()) { // when the result list is empty
                    result.add(currentItem);
                }
            }
    }
}

```

```

} else { // there is node inside result list
    Item temp = result.getFirst();
    boolean isInserted = false;

    for (int j = 0; j < result.getSize(); j++) {
        if (currentItem.getQty() > temp.getQty()) { // check whether currentItem
total quantity > result list existing donations' total quantity
            result.insertBefore(temp, currentItem);
            isInserted = true;
            break;
        }
        temp = result.getNextData(temp); // shift to next donation in result list
    }

    if (!isInserted) result.add(currentItem); // if currentItem total amount < all
existing donations' total amount in result list, add to the last position
}
currentItem = useItemList.getNextData(currentItem); // shift to next
donation in useItemList
}
break;
case "moneyAmt":
    currentItem = useItemList.getFirst();

    for (int i = 0; i < useItemList.getSize(); i++) {
        if (result.isEmpty()) { // when the result list is empty
            result.add(currentItem);
        } else { // there is node inside result list
            Item temp = result.getFirst();
            boolean isInserted = false;

            for (int j = 0; j < result.getSize(); j++) {
                if (currentItem.getTotalAmt() > temp.getTotalAmt()) { // check whether
currentItem total amount > result list existing donations' total amount
                    result.insertBefore(temp, currentItem);
                    isInserted = true;
                    break;
                }
                temp = result.getNextData(temp); // shift to next donation in result list
            }

            if (!isInserted) result.add(currentItem); // if currentItem total amount < all
existing donations' total amount in result list, add to the last position
        }
        currentItem = useItemList.getNextData(currentItem); // shift to next
donation in useItemList
    }
}

```

```

        break;
    case "thresholdAmt":
        currentItem = useItemList.getFirst();

        for (int i = 0; i < useItemList.getSize(); i++) {
            if (result.isEmpty()) { // when the result list is empty
                result.add(currentItem);
            } else { // there is node inside result list
                Item temp = result.getFirst();
                boolean isInserted = false;

                for (int j = 0; j < result.getSize(); j++) {
                    if (currentItem.getTotalAmt() < temp.getTotalAmt()) { // check whether
currentItem total amount < result list existing donations' total amount
                        result.insertBefore(temp, currentItem);
                        isInserted = true;
                        break;
                    }
                    temp = result.getNextData(temp); // shift to next donation in result list
                }

                if (!isInserted) result.add(currentItem); // if currentItem total amount < all
existing donations' total amount in result list, add to the last position
            }
            currentItem = useItemList.getNextData(currentItem); // shift to next
donation in useItemList
        }
        break;
    case "thresholdQty":
        currentItem = useItemList.getFirst();

        for (int i = 0; i < useItemList.getSize(); i++) {
            if (result.isEmpty()) { // when the result list is empty
                result.add(currentItem);
            } else { // there is node inside result list
                Item temp = result.getFirst();
                boolean isInserted = false;

                for (int j = 0; j < result.getSize(); j++) {
                    if (currentItem.getQty() < temp.getQty()) { // check whether currentItem
total quantity < result list existing donations' total quantity
                        result.insertBefore(temp, currentItem);
                        isInserted = true;
                        break;
                    }
                    temp = result.getNextData(temp); // shift to next donation in result list
                }
            }
        }
    }
}

```

```

        if (!isInserted) result.add(currentItem); // if currentItem total amount < all
existing donations' total amount in result list, add to the last position
    }
    currentItem = useItemList.getNextData(currentItem); // shift to next
donation in useItemList
}
break;
default:
}
return result;
}

// calculate donation total amount
public static ListInterface<Donation> calcDonationAmt(ListInterface<Donation>
useDonationList) {
    Donation useDonation = useDonationList.getFirst();

    for (int i = 0; i < useDonationList.getSize(); i++) {
        double totalAmt = 0.0;
        ListInterface<StockIn> useStockInList = useDonation.getStockIn();
        StockIn useStockIn = useStockInList.getFirst();

        for (int j = 0; j < useStockInList.getSize(); j++) {
            if (moneyList.search(useStockIn.getItem()) != null) {
                totalAmt += useStockIn.getAmount(); // add money item amount
            } else if (foodBeverageList.search(useStockIn.getItem()) != null) {
                totalAmt += (double) useStockIn.getQty() *
useStockIn.getItem().getAmount(); // add food beverage item amount
            }
            useStockIn = useStockInList.getNextData(useStockIn); // shift to next data
        }
        useDonation.setTotalAmount(totalAmt);
        useDonationList.update(useDonation); // update the totalAmt recorded in
donation within useDonationList
        useDonation = useDonationList.getNextData(useDonation); // shift to next data
    }
    return useDonationList;
}

// calculate donated item qty
public static ListInterface<Item> calcDonatedItem(ListInterface<Donation>
useDonationList, int useCategory) {
    ListInterface<Item> resultList = new DoublyLinkedList<>();
    Donation useDonation = useDonationList.getFirst();

    for (int i = 0; i < useDonationList.getSize(); i++) {

```

```

ListInterface<StockIn> useStockInList = useDonation.getStockIn();
StockIn useStockIn = useStockInList.getFirst();

for (int j = 0; j < useStockInList.getSize(); j++) {
    Item currentStockInItem = useStockIn.getItem();
    Item existingItem = resultList.search(currentStockInItem);

    if (existingItem == null) {
        // Create a new Item instance with the relevant data
        Item newItem = new Item();
        newItem.setId(currentStockInItem.getId());
        newItem.setName(currentStockInItem.getName());
        newItem.setQty(useStockIn.getQty());

        if (moneyList.search(currentStockInItem) != null) {
            newItem.setAmount(0);
            newItem.setTotalAmt(useStockIn.getAmt());
        } else {
            newItem.setAmount(currentStockInItem.getAmount());
            newItem.setTotalAmt((double) newItem.getQty() *
newItem.getAmount());
        }
        resultList.add(newItem);
    } else {
        // Update existing item in resultList
        if (moneyList.search(existingItem) != null) {
            existingItem.setTotalAmt(existingItem.getTotalAmt() +
useStockIn.getAmt());
        } else if (foodBeverageList.search(existingItem) != null) {
            existingItem.setQty(existingItem.getQty() + useStockIn.getQty());
            existingItem.setTotalAmt((double) existingItem.getQty() *
existingItem.getAmount());
        }
    }
}

useStockIn = useStockInList.getNextData(useStockIn);
}

useDonation = useDonationList.getNextData(useDonation);
}

// Filter the resultList based on the useCategory
Item currentItem = resultList.getFirst();
ListInterface<Item> removeItemList = new DoublyLinkedList<>();

for (int i = 0; i < resultList.getSize(); i++) {
    if ((useCategory == 1 && moneyList.search(currentItem) == null) || (useCategory

```

```

== 2 && moneyList.search(currentItem) != null)) {
    removeItemList.add(currentItem);
}

currentItem = resultList.getNextData(currentItem);
}
resultList = resultList.minus(removeItemList);
return resultList;
}

// find out stock in list with the same item
public static ListInterface<StockIn> findItemStockIn(ListInterface<Donation>
useDonationList, Item useItem) {
    ListInterface<StockIn> resultList = new DoublyLinkedList<>();
    Donation currentDonation = useDonationList.getFirst();

    for (int i = 0; i < useDonationList.getSize(); i++) {
        ListInterface<StockIn> useStockInList = currentDonation.getStockIn();
        StockIn currentStockIn = useStockInList.getFirst();

        for (int j = 0; j < useStockInList.getSize(); j++) {
            if (currentStockIn.getItem().compareTo(useItem) == 0) {
                resultList.add(currentStockIn);
            }
            currentStockIn = useStockInList.getNextData(currentStockIn);
        }
        currentDonation = useDonationList.getNextData(currentDonation);
    }
    return resultList;
}

// find out stock in list with the same donor
public static ListInterface<StockIn> findDonorStockIn(ListInterface<Donation>
useDonationList, Donor useDonor) {
    ListInterface<StockIn> resultList = new DoublyLinkedList<>();
    Donation currentDonation = useDonationList.getFirst();

    for (int i = 0; i < useDonationList.getSize(); i++) {
        if (currentDonation.getDonor().compareTo(useDonor) == 0) {
            ListInterface<StockIn> useStockInList = currentDonation.getStockIn();
            StockIn currentStockIn = useStockInList.getFirst();

            for (int j = 0; j < useStockInList.getSize(); j++) {
                resultList.add(currentStockIn);
                currentStockIn = useStockInList.getNextData(currentStockIn);
            }
        }
    }
}

```

```
        currentDonation = useDonationList.getNextData(currentDonation);
    }
    return resultList;
}

// sum total stock in quantity
public static int sumStockInQty(ListInterface<StockIn> useStockInList) {
    int total = 0;
    StockIn currentStockIn = useStockInList.getFirst();

    for (int i = 0; i < useStockInList.getSize(); i++) {
        total += currentStockIn.getQty();
        currentStockIn = useStockInList.getNextData(currentStockIn);
    }
    return total;
}

// sum total stock in quantity
public static double sumStockInAmt(ListInterface<StockIn> useStockInList) {
    double total = 0;
    StockIn currentStockIn = useStockInList.getFirst();

    for (int i = 0; i < useStockInList.getSize(); i++) {
        total += currentStockIn.getAmt();
        currentStockIn = useStockInList.getNextData(currentStockIn);
    }
    return total;
}

// find the item where donor contribute the most
public static Item findBestContributorItem(ListInterface<Item> useItemList, int
useCategory) {
    Item bestItem = null;
    Item currentItem = useItemList.getFirst();

    for (int i = 0; i < useItemList.getSize(); i++) {
        if (bestItem == null) {
            bestItem = currentItem;
        } else {
            if (useCategory == 1) {
                currentItem.setSearchBy("contributeAmt");
            } else {
                currentItem.setSearchBy("contributeQty");
            }
            if (bestItem.compareTo(currentItem) == 0) {
                bestItem = currentItem;
            }
        }
    }
}
```

```
    }
    currentItem = useItemList.getNextData(currentItem);
}

return bestItem;
}

}
```

Entity class: Donation.java

```
package entity;

import adt.ListInterface;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import utility.Helper;

public class Donation implements Comparable<Donation> {
    private static int idCounter = 0;
    private int id;
    private Donor donor;
    private ListInterface<StockIn> stockIn;
    private String notes;
    private LocalDate donationDate;
    private String searchBy;
    private LocalDate startDate;
    private LocalDate endDate;
    private double totalAmount;

    public Donation() {
        this.searchBy = "unknown";
        this.startDate = null;
        this.endDate = null;
        this.totalAmount = 0.0;
    }

    public Donation(int id) {
        this.id = id;
        this.searchBy = "unknown";
        this.startDate = null;
        this.endDate = null;
        this.totalAmount = 0.0;
    }
}
```

```
public Donation(Donor donor, ListInterface<StockIn> stockIn, String notes, LocalDate  
donationDate) {  
    this.id = ++idCounter;  
    this.donor = donor;  
    this.stockIn = stockIn;  
    this.notes = notes;  
    this.donationDate = donationDate;  
    this.searchBy = "unknown";  
    this.startDate = null;  
    this.endDate = null;  
    this.totalAmount = 0.0;  
}  
  
public int getId() {  
    return id;  
}  
  
public Donor getDonor() {  
    return donor;  
}  
  
public ListInterface<StockIn> getStockIn() {  
    return stockIn;  
}  
  
public String getNotes() {  
    return notes;  
}  
  
public LocalDate getDonationDate() {  
    return donationDate;  
}  
  
public String getSearchBy() {  
    return searchBy;  
}  
  
public LocalDate getStartDate() {  
    return startDate;  
}  
  
public LocalDate getEndDate() {  
    return endDate;  
}  
  
public double getTotalAmount() {  
    return totalAmount;
```

```
}

public void setId(int id) {
    this.id = id;
}

public void setDonor(Donor donor) {
    this.donor = donor;
}

public void setStockIn(ListInterface<StockIn> stockIn) {
    this.stockIn = stockIn;
}

public void setNotes(String notes) {
    this.notes = notes;
}

public void setDonationDate(LocalDate donationDate) {
    this.donationDate = donationDate;
}

public void setSearchBy(String searchBy) {
    this.searchBy = searchBy;
}

public void setStartDate(LocalDate startDate) {
    this.startDate = startDate;
}

public void setEndDate(LocalDate endDate) {
    this.endDate = endDate;
}

public void setTotalAmount(double totalAmount) {
    this.totalAmount = totalAmount;
}

public String formatID() {
    return String.format("DN%04d", id);
}

public String formatDonationDate() {
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");
    return donationDate.format(formatter);
}
```

```

@Override
public String toString() {
    return "Donation ID: " + id + "\n"
        + "Donor ID: " + donor.getId() + "\n"
        + "Notes: " + notes + "\n"
        + "Donation Date: " + donationDate.toString() + "\n";
}

@Override
public int compareTo(Donation o) {
    switch (o.getSearchBy()) {
        case "donorID":
            return donor.getId() == o.getDonor().getId() ? 0 : -1;
        case "donorName":
            return
donor.getName().toLowerCase().contains(o.getDonor().getName().toLowerCase()) ? 0 : -1;
        case "itemID":
            StockIn currentStockInID = stockIn.getFirst();
            for (int i = 0; i < stockIn.getSize(); i++) {
                if (currentStockInID.getItem().getId() ==
o.getStockIn().getFirst().getItem().getId()) return 0;
                currentStockInID = stockIn.getNextData(currentStockInID);
            }
            return -1;
        case "itemName":
            StockIn currentStockInName = stockIn.getFirst();
            for (int i = 0; i < stockIn.getSize(); i++) {
                if
(currentStockInName.getItem().getName().toLowerCase().contains(o.getStockIn().getFirst(
).getItem().getName().toLowerCase())) return 0;
                currentStockInName = stockIn.getNextData(currentStockInName);
            }
            return -1;
        case "donationDate":
            return Helper.checkDateRange(donationDate, o.getStartDate(), o.getEndDate()) ?
0 : -1;
        default:
            return id == o.getId() ? 0 : -1;
    }
}
}

```

2. Screenshot

Donation Subsystem Menu

| Donation Subsystem | | |
|--------------------------|--------|--|
| Function | Option | |
| Add new donation | 1 | |
| View donation | 2 | |
| Search donation | 3 | |
| Update donation | 4 | |
| Remove donation | 5 | |
| Donation Report | 6 | |
| Create new donation item | 7 | |
| Exit | 0 | |

Please select function (0 - 7):

Figure 1: Menu of donation subsystem

1.1 Add New Donation

| | |
|---|---------------------------------------|
| Please enter donor ID ('1' for 'DR0001') ('0' to exit): 3 | |
| + | -----+ Donor ID: DR0003 +-----+ |
| + | Name: Linda Ho +-----+ |
| + | Type: Government +-----+ |
| Please confirm the donor info (Y/N); Y | |

Figure 1.1.1: Confirmation dialog when selecting donor

```
+-----+  
| Category | Option |  
+-----+-----+  
| Money | 1 |  
+-----+-----+  
| Food and Beverages | 2 |  
+-----+  
Please select the category option of donation item ('0' to back): 1  
  
+-----+-----+-----+  
| Item ID | Item Name | Quantity | Total Amount (RM) |  
+-----+-----+-----+  
| IM0001 | Cash | 0 | RM 200.00 |  
+-----+-----+-----+  
| IM0002 | Bank Transfer | 0 | RM 300.00 |  
+-----+-----+-----+  
| IM0003 | Cheque | 0 | RM 150.00 |  
+-----+-----+-----+  
| IM0004 | Credit Card | 0 | RM 500.00 |  
+-----+-----+-----+  
| IM0005 | PayPal | 0 | RM 250.00 |  
+-----+-----+-----+  
| IM0006 | Mobile Payment | 0 | RM 100.00 |  
+-----+-----+-----+  
| IM0007 | Gift Card | 0 | RM 50.00 |  
+-----+-----+-----+  
| IM0008 | Cryptocurrency | 0 | RM 600.00 |  
+-----+  
Please select the ID of donation item ('1' for 'IM0001') ('0' to exit): 2  
  
Please enter the donation amount (RM) ('0' to exit): 450  
  
Are you sure to donate 'Bank Transfer' with amount RM 450.00? (Y/N): Y
```

```

Do you want to donate other items? (Y/N): y

+-----+
| Item ID | Item Name           | Quantity | Total Amount (RM) |
+-----+
| IM0001  | Cash                 | 0        | RM      200.00   |
+-----+
| IM0003  | Cheque                | 0        | RM      150.00   |
+-----+
| IM0004  | Credit Card           | 0        | RM      500.00   |
+-----+
| IM0005  | PayPal                | 0        | RM      250.00   |
+-----+
| IM0006  | Mobile Payment          | 0        | RM      100.00   |
+-----+
| IM0007  | Gift Card              | 0        | RM      50.00    |
+-----+
| IM0008  | Cryptocurrency          | 0        | RM      600.00   |
+-----+
Please select the ID of donation item ('1' for 'IM0001')('0' to exit): 6

Please enter the donation amount (RM) ('0' to exit): 200

Are you sure to donate 'Mobile Payment' with amount RM 200.00? (Y/N): y

```

Figure 1.1.2 & 1.1.3: Select donation item with quantity or amount

```

Summarize Donation
=====

Any notes you want to write here? ('-' if no comment):
General donation

+-----+
| Donation ID: | DN0022               |
+-----+
| Stock In Item:
| IN0046     Bank Transfer     RM 450.00
| IN0047     Mobile Payment    RM 200.00
+-----+
| Notes:       | General donation         |
+-----+
| Donation date: | 26-08-2024               |
+-----+
You have successfully made a donation. Thank you!

Do you want to make donation for another donors? (Y/N): |

```

Figure 1.1.4: Summarize donation

1.2 View Donation

```
Please select function (0 - 7): 2

+-----+
| Type of list | Option |
+=====+=====+
| Donation list | 1 |
+-----+
| Donation list By criteria | 2 |
+-----+
| Donation item list | 3 |
+-----+

Please select list ('0' to exit):
```

Figure 1.2.1: Select type of donation list

| Donation ID | Donor Name | Type | Date |
|-------------|-------------------------------|--------------|------------|
| DN0001 | Hope for All | Organization | 15-08-2024 |
| DN0002 | HappyP Sdn Bhd | Organization | 09-08-2024 |
| DN0003 | Global Charities | Organization | 01-08-2024 |
| DN0004 | Ministry of Health | Organization | 30-06-2024 |
| DN0005 | Emily Goh | Government | 20-07-2024 |
| DN0006 | Vanessa Tan | Organization | 18-08-2024 |
| DN0007 | Care International | Government | 05-08-2024 |
| DN0008 | Sophia Ng | Government | 22-07-2024 |
| DN0009 | Bright Minds Foundation | Organization | 10-07-2024 |
| DN0010 | Chloe Teo | Organization | 01-07-2024 |
| DN0011 | Save the Earth | Government | 25-06-2024 |
| DN0012 | Thomas Ong | Government | 19-05-2024 |
| DN0013 | Benjamin Lau | Government | 03-05-2024 |
| DN0014 | Nicholas Tan | Government | 15-04-2024 |
| DN0015 | Ministry of Finance | Government | 01-04-2024 |
| DN0016 | Youtuber Celine | Organization | 22-03-2024 |
| DN0017 | Ministry of Communications | Organization | 10-03-2024 |
| DN0018 | Empower Communities | Organization | 28-02-2024 |
| DN0019 | Ministry of Foreign Affairs | Organization | 15-02-2024 |
| DN0020 | Ministry of Rural Development | Organization | 30-01-2024 |
| DN0021 | Linda Ho | Government | 26-08-2024 |

You may enter donation ID to view more details of donation ('l' for 'DN0001')('0' to exit):

Figure 1.2.2: Donation list

1.2.1 View donation details (view next and previous donation details)

```
You may enter donation ID to view more details of donation ('l' for 'DN0001') ('0' to exit): 3

+-----+
| Donation ID: | DN0003 |
+-----+
| Stock In Item: |
| IN0005    PayPal      RM 250.00 |
| IN0006    Mobile Payment   RM 100.00 |
+-----+
| Notes:       | For winter season aid |
+-----+
| Donation date: | 01-08-2024 |
+-----+

+-----+
| Donor ID: | DR0016 |
+-----+
| Name:     | Global Charities |
+-----+
| Type:     | Organization |
+-----+
1) View next donation (based on ID)
2) View previous donation (based on ID)
0) Back

Select option:
```

Figure 1.2.2.1: Donation details of selected donation

```
Select option: 1

+-----+
| Donation ID: | DN0004 |
+-----+
| Stock In Item: |
| IN0007      Gift Card          RM 50.00 |
| IN0008      Cryptocurrency    RM 600.00 |
| IN0009      Canned Beans       x 250   |
+-----+
| Notes:        | Annual donation |
+-----+
| Donation date: | 30-06-2024 |
+-----+

+-----+
| Donor ID: | DR0031 |
+-----+
| Name:     | Ministry of Health |
+-----+
| Type:     | Organization |
+-----+
1) View next donation (based on ID)
2) View previous donation (based on ID)
0) Back

Select option: |
```

Figure 1.2.2.2: Next donation details

```
Select option: 2

+---+
| Donation ID: | DN0003 |
+---+
| Stock In Item:
| IN0005    PayPal      RM 250.00
| IN0006    Mobile Payment   RM 100.00
+---+
| Notes:          | For winter season aid
+---+
| Donation date: | 01-08-2024
+---+

+---+
| Donor ID: | DR0016
+---+
| Name:     | Global Charities
+---+
| Type:     | Organization
+---+
1) View next donation (based on ID)
2) View previous donation (based on ID)
0) Back

Select option:
```

Figure 1.2.2.3: Previous donation details

1.2.2 View donation list by criteria

```
+--+
| Type of list          | Option   |
+---+-----+-----+
| Donation list          | 1        |
+---+-----+-----+
| Donation list By criteria | 2        |
+---+-----+-----+
| Donation item list      | 3        |
+---+-----+-----+


Please select list ('0' to exit): 2

+--+
| Filter By           | Option   |
+---+-----+-----+
| Donor ID            | 1        |
+---+-----+-----+
| Donor Name           | 2        |
+---+-----+-----+
| Donated Item ID      | 3        |
+---+-----+-----+
| Donated Item Name     | 4        |
+---+-----+-----+
| Donation Date         | 5        |
+---+-----+-----+


Please select a filter ('0' to exit): |
```

Figure 1.2.3.1: Filter by option

```
Please select a filter ('0' to exit): 1

Please enter donor ID ('1' for 'DR0001')('0' to exit): 2

+-----+
| Donor ID: | DR0002           |
+-----+
| Name:     | HappyP Sdn Bhd       |
+-----+
| Type:     | Organization          |
+-----+


Please confirm the donor info (Y/N); y

+-----+-----+-----+-----+
| Donation ID | Donor Name           | Type      | Date    |
+-----+-----+-----+-----+
| DN0002      | HappyP Sdn Bhd       | Organization | 09-08-2024 |
+-----+-----+-----+-----+


You may enter donation ID to view more details of donation ('1' for 'DN0001')('0' to exit):
```

Figure 1.2.3.2: Filter by donor ID

```
Please select a filter ('0' to exit): 2

Enter donor name ('0' to exit): Thomas Ong

+-----+-----+-----+
| Donation ID | Donor Name           | Type      | Date       |
+-----+-----+-----+
| DN0012     | Thomas Ong             | Government | 19-05-2024 |
+-----+-----+-----+
| DN0017     | Thomas Ong             | Government | 10-03-2024 |
+-----+-----+-----+
You may enter donation ID to view more details of donation ('l' for 'DN0001')('0' to exit): |
```

Figure 1.2.3.3: Filter by donor name

```
Please select a filter ('0' to exit): 3
```

| Item ID Item Name | Quantity | Total Amount (RM) |
|----------------------------|----------|-------------------|
| IM0001 Cash | 0 | RM 200.00 |
| IM0002 Bank Transfer | 0 | RM 300.00 |
| IM0003 Cheque | 0 | RM 150.00 |
| IM0004 Credit Card | 0 | RM 500.00 |
| IM0005 PayPal | 0 | RM 250.00 |
| IM0006 Mobile Payment | 0 | RM 100.00 |
| IM0007 Gift Card | 0 | RM 50.00 |
| IM0008 Cryptocurrency | 0 | RM 600.00 |
| IM0009 Canned Beans | 420 | RM 4200.00 |
| IM0010 Rice Pack | 830 | RM 20750.00 |
| IM0011 Bottled Water | 190 | RM 2280.00 |
| IM0012 Orange Juice | 330 | RM 6600.00 |
| IM0013 Spaghetti | 720 | RM 10800.00 |
| IM0014 Tomato Sauce | 200 | RM 1400.00 |
| IM0015 Olive Oil | 350 | RM 6300.00 |
| IM0016 Corn Flakes | 250 | RM 2125.00 |
| IM0017 Green Tea | 300 | RM 1800.00 |
| IM0018 Instant Coffee | 180 | RM 2250.00 |
| IM0019 Milk | 120 | RM 480.00 |
| IM0020 Chocolate Bar | 400 | RM 4000.00 |
| IM0021 Canned Tuna | 220 | RM 1210.00 |
| IM0022 Peanut Butter | 170 | RM 1360.00 |
| IM0023 Whole Wheat Bread | 280 | RM 980.00 |

| Donation ID | Item Name | Quantity | Type | Value |
|-------------|-------------------|----------|------|---------|
| IM0024 | Frozen Pizza | 60 | RM | 900.00 |
| IM0025 | Cheddar Cheese | 140 | RM | 1260.00 |
| IM0026 | Yogurt | 110 | RM | 715.00 |
| IM0027 | Granola Bars | 300 | RM | 3600.00 |
| IM0028 | Pasta Sauce | 180 | RM | 1800.00 |
| IM0029 | Frozen Vegetables | 90 | RM | 720.00 |
| IM0030 | Butter | 240 | RM | 1200.00 |
| IM0031 | Apple Juice | 130 | RM | 910.00 |
| IM0032 | Honey | 100 | RM | 1500.00 |
| IM0033 | Oats | 210 | RM | 1785.00 |
| IM0034 | Canned Pineapple | 190 | RM | 950.00 |
| IM0035 | Almond Milk | 150 | RM | 975.00 |

Please select the ID of donation item ('l' for 'IM0001')('0' to exit): 11

| Donation ID | Donor Name | Type | Date |
|-------------|-----------------------------|--------------|------------|
| DN0005 | Emily Goh | Government | 20-07-2024 |
| DN0019 | Ministry of Foreign Affairs | Organization | 15-02-2024 |

You may enter donation ID to view more details of donation ('l' for 'DN0001')('0' to exit):

Figure 1.2.3.4: Filter by donation item ID

Please select a filter ('0' to exit): 4

Please enter donated item name ('0' to exit): spaghetti

| Donation ID | Donor Name | Type | Date |
|-------------|-------------------------------|--------------|------------|
| DN0006 | Vanessa Tan | Organization | 18-08-2024 |
| DN0020 | Ministry of Rural Development | Organization | 30-01-2024 |

You may enter donation ID to view more details of donation ('l' for 'DN0001')('0' to exit): |

Figure 1.2.3.5: Filter by donation item name

```

Please select a filter ('0' to exit): 5

Enter donation start date (yyyy-mm-dd) ('0' to exit): 2024-1-1

Enter donation end date (yyyy-mm-dd) ('0' to exit): 2024-6-30

+-----+
| Donation ID | Donor Name           | Type      | Date       |
+-----+-----+-----+-----+
| DN0004     | Ministry of Health        | Organization | 30-06-2024 |
+-----+-----+-----+-----+
| DN0011     | Save the Earth            | Government  | 25-06-2024 |
+-----+-----+-----+-----+
| DN0012     | Thomas Ong                | Government  | 19-05-2024 |
+-----+-----+-----+-----+
| DN0013     | Benjamin Lau              | Government  | 03-05-2024 |
+-----+-----+-----+-----+
| DN0014     | Nicholas Tan               | Government  | 15-04-2024 |
+-----+-----+-----+-----+
| DN0015     | Ministry of Finance       | Government  | 01-04-2024 |
+-----+-----+-----+-----+
| DN0016     | Youtuber Celine           | Organization | 22-03-2024 |
+-----+-----+-----+-----+
| DN0017     | Thomas Ong                | Government  | 10-03-2024 |
+-----+-----+-----+-----+
| DN0018     | Empower Communities       | Organization | 28-02-2024 |
+-----+-----+-----+-----+
| DN0019     | Ministry of Foreign Affairs | Organization | 15-02-2024 |
+-----+-----+-----+-----+
| DN0020     | Ministry of Rural Development | Organization | 30-01-2024 |
+-----+-----+-----+-----+
You may enter donation ID to view more details of donation ('l' for 'DN0001')('0' to exit): |

```

Figure 1.2.3.6: Filter by donation date range

1.2.3 View donation item list

| Please select list ('0' to exit): 3 | | | | |
|-------------------------------------|-------------------|----------|-------------------|--|
| Item ID | Item Name | Quantity | Total Amount (RM) | |
| IM0001 | Cash | 0 | RM 200.00 | |
| IM0002 | Bank Transfer | 0 | RM 300.00 | |
| IM0003 | Cheque | 0 | RM 150.00 | |
| IM0004 | Credit Card | 0 | RM 500.00 | |
| IM0005 | PayPal | 0 | RM 250.00 | |
| IM0006 | Mobile Payment | 0 | RM 100.00 | |
| IM0007 | Gift Card | 0 | RM 50.00 | |
| IM0008 | Cryptocurrency | 0 | RM 600.00 | |
| IM0009 | Canned Beans | 420 | RM 4200.00 | |
| IM0010 | Rice Pack | 830 | RM 20750.00 | |
| IM0011 | Bottled Water | 190 | RM 2280.00 | |
| IM0012 | Orange Juice | 330 | RM 6600.00 | |
| IM0013 | Spaghetti | 720 | RM 10800.00 | |
| IM0014 | Tomato Sauce | 200 | RM 1400.00 | |
| IM0015 | Olive Oil | 350 | RM 6300.00 | |
| IM0016 | Corn Flakes | 250 | RM 2125.00 | |
| IM0017 | Green Tea | 300 | RM 1800.00 | |
| IM0018 | Instant Coffee | 180 | RM 2250.00 | |
| IM0019 | Milk | 120 | RM 480.00 | |
| IM0020 | Chocolate Bar | 400 | RM 4000.00 | |
| IM0021 | Canned Tuna | 220 | RM 1210.00 | |
| IM0022 | Peanut Butter | 170 | RM 1360.00 | |
| IM0023 | Whole Wheat Bread | 280 | RM 980.00 | |
| IM0024 | Frozen Pizza | 60 | RM 900.00 | |

| |
|---|
| IM0025 Cheddar Cheese 140 RM 1260.00 |
| IM0026 Yogurt 110 RM 715.00 |
| IM0027 Granola Bars 300 RM 3600.00 |
| IM0028 Pasta Sauce 180 RM 1800.00 |
| IM0029 Frozen Vegetables 90 RM 720.00 |
| IM0030 Butter 240 RM 1200.00 |
| IM0031 Apple Juice 130 RM 910.00 |
| IM0032 Honey 100 RM 1500.00 |
| IM0033 Oats 210 RM 1785.00 |
| IM0034 Canned Pineapple 190 RM 950.00 |
| IM0035 Almond Milk 150 RM 975.00 |

You may enter item ID to view more details of item ('1' for 'IM0001') ('0' to exit):

Figure 1.2.3.1: Donation item list

| |
|---|
| You may enter item ID to view more details of item ('1' for 'IM0001') ('0' to exit): 18 |
| +-----+ Item ID: IM0018 |
| +-----+ Item Name: Instant Coffee |
| +-----+ Category: Food and Beverages |
| +-----+ Description: A jar of instant coffee, 200g |
| +-----+ Quantity: 180 |
| +-----+ Price Per Item: RM 12.50 |
| +-----+ Total Amount: RM 2250.00 |
| +-----+ |
| 1) View next item (based on ID) |
| 2) View previous item (based on ID) |
| 0) Back |
| Select option: |

Figure 1.2.3.2: Donation item details

1.3 Search Donation

```
Please select function (0 - 7): 3

+-----+
| Search by           | Option   |
+=====+=====+
| Donation ID        | 1         |
+-----+-----+
| Donor ID           | 2         |
+-----+-----+

Please select list ('0' to exit): |
```

Figure 1.3.1: Menu of searching donation

1.3.1 Search by donation ID

```
Please select function (0 - 7): 3

+-----+
| Search by           | Option   |
+=====+=====+
| Donation ID        | 1         |
+-----+-----+
| Donor ID           | 2         |
+-----+-----+

Please select list ('0' to exit): 1

Please enter the donation ID ('1' for 'DN0001') ('0' to exit): 14

+-----+
| Donation ID:    | DN0014          |
+-----+
| Stock In Item:  |
| IN0030     Butter      x 100   |
| IN0031     Apple Juice  x 210   |
| IN0032     Honey       x 190   |
+-----+
| Notes:          | Community support |
+-----+
| Donation date: | 15-04-2024      |
+-----+

Do you want to continue search other donation by donation ID? ('Y' - Yes; 'N' - No): |
```

Figure 1.3.1: Search by donation ID

1.3.2 Search by donor ID

```
+-----+  
| Search by          | Option   |  
+=====+=====+  
| Donation ID       |      1    |  
+-----+-----+  
| Donor ID          |      2    |  
+-----+-----+  
  
Please select list ('0' to exit): 2  
  
Please enter the donor ID ('1' for 'DR0001')('0' to exit): 18  
  
+-----+-----+-----+  
| Donation ID | Donor Name           | Type      | Date     |  
+=====+=====+=====+=====+=====+  
| DN0001      | Hope for All          | Organization | 15-08-2024 |  
+-----+-----+-----+  
You may enter donation ID to view more details of donation ('1' for 'DN0001')('0' to exit):
```

Figure 1.3.2 Search by donor ID

1.4 Update Donation

```
Please select function (0 - 7): 4
```

| Donation ID | Donor Name | Type | Date |
|-------------|-------------------------------|--------------|------------|
| DN0001 | Hope for All | Organization | 15-08-2024 |
| DN0002 | HappyP Sdn Bhd | Organization | 09-08-2024 |
| DN0003 | Global Charities | Organization | 01-08-2024 |
| DN0004 | Ministry of Health | Organization | 30-06-2024 |
| DN0005 | Emily Goh | Government | 20-07-2024 |
| DN0006 | Vanessa Tan | Organization | 18-08-2024 |
| DN0007 | Care International | Government | 05-08-2024 |
| DN0008 | Sophia Ng | Government | 22-07-2024 |
| DN0009 | Bright Minds Foundation | Organization | 10-07-2024 |
| DN0010 | Chloe Teo | Organization | 01-07-2024 |
| DN0011 | Save the Earth | Government | 25-06-2024 |
| DN0012 | Thomas Ong | Government | 19-05-2024 |
| DN0013 | Benjamin Lau | Government | 03-05-2024 |
| DN0014 | Nicholas Tan | Government | 15-04-2024 |
| DN0015 | Ministry of Finance | Government | 01-04-2024 |
| DN0016 | Youtuber Celine | Organization | 22-03-2024 |
| DN0017 | Thomas Ong | Government | 10-03-2024 |
| DN0018 | Empower Communities | Organization | 28-02-2024 |
| DN0019 | Ministry of Foreign Affairs | Organization | 15-02-2024 |
| DN0020 | Ministry of Rural Development | Organization | 30-01-2024 |

```
Please enter the donation ID you want to update ('l' for 'DN0001')('0' to exit):
```

```
Please enter the donation ID you want to update ('l' for 'DN0001')('0' to exit): 7

+-----+
| Donation ID: | DN0007 |
+-----+
| Stock In Item: |
| IN0015      Olive Oil      x 150 |
| IN0016      Corn Flakes    x 240 |
+-----+
| Notes:       | Support for education |
+-----+
| Donation date: | 26-08-2024 |
+-----+
+-----+
| Update           | Option   |
+-----+-----+
| Switch donor     | 1         |
+-----+-----+
| Modify quantity of current item | 2         |
+-----+-----+
| Add new item      | 3         |
+-----+-----+
| Remove item       | 4         |
+-----+-----+
| Update notes      | 5         |
+-----+-----+

Select an option ('0' to exit): |
```

Figure 1.4: Search the donation that is wanted to update

1.4.1 Switch donor

```
Select an option ('0' to exit): 1

Current donor:

+-----+
| Donor ID: | DR0024 |
+-----+
| Name:     | Care International |
+-----+
| Type:     | Government |
+-----+
Enter donor ID to switch the donor ('0' to exit): 8

+-----+
| Donor ID: | DR0008 |
+-----+
| Name:     | Nicholas Tan |
+-----+
| Type:     | Government |
+-----+

Are you sure to switch to this donor (DR0008) for this donation DN0007? ('Y' - Yes; 'N' - No): y
```

Figure 1.4.1.1: Show current donor and confirm the new donor details

```
+-----+
| Donation ID: | DN0007 |
+-----+
| Stock In Item: |
| IN0015      Olive Oil      x 150 |
| IN0016      Corn Flakes    x 240 |
+-----+
| Notes:       | Support for education |
+-----+
| Donation date: | 26-08-2024 |
+-----+
+-----+
| Donor ID: | DR0008 |
+-----+
| Name:     | Nicholas Tan |
+-----+
| Type:     | Government |
+-----+
The donation detail has been updated successfully.
```

Figure 1.4.1.2: Result of successful donation update (donor)

1.4.2 Modify quantity of current item

```
Select an option ('0' to exit): 2

+-----+
| Stock In ID | Item Name           | Quantity | Total Amount (RM) |
+-----+
| IN0015       | Olive Oil            | 150      | RM             2700.00 |
+-----+
| IN0016       | Corn Flakes          | 240      | RM             2040.00 |
+-----+
Select the stock in ID to modify the quantity ('0' to exit):
```

Figure 1.4.2.1: List of updateable item quantity in selected donation

```
Select the stock in ID to modify the quantity ('0' to exit): 16

+-----+
| Stock In ID:      | IN0016          |
+-----+
| Stock In Item Name: | Corn Flakes      |
+-----+
| Stock In Quantity: | 240             |
+-----+
| Total Amount:     | RM 2040.00      |
+-----+
Enter the new quantity ('0' to exit): 300

Are you sure to change the quantity of stock (IN0016) to 300? ('Y' - Yes; 'N' - No): y
```

Figure 1.4.2.2: Select stock in ID and update the quantity of the selected item

```
+-----+
| Donation ID:   | DN0007          |
+-----+
| Stock In Item:
| IN0015    Olive Oil      x 150
| IN0016    Corn Flakes    x 300
+-----+
| Notes:        | Support for education |
+-----+
| Donation date: | 26-08-2024       |
+-----+
+-----+
| Donor ID:    | DR0024          |
+-----+
| Name:        | Care International |
+-----+
| Type:        | Government       |
+-----+
The donation detail has been updated successfully.
```

Figure 1.4.2.3: Result of successful donation update (item quantity)

1.4.3 Add new item

```
Select an option ('0' to exit): 3

+-----+
| Category           | Option  |
+=====+=====+
| Money              | 1      |
+-----+
| Food and Beverages | 2      |
+-----+
Please select the category option of donation item ('0' to back):
```

Figure 1.4.3.1: Selecting the category of donation item

| Item ID | Item Name | Quantity | Total Amount (RM) |
|---------|-------------------|----------|-------------------|
| IM0009 | Canned Beans | 420 | 4200.00 |
| IM0010 | Rice Pack | 830 | 20750.00 |
| IM0011 | Bottled Water | 190 | 2280.00 |
| IM0012 | Orange Juice | 330 | 6600.00 |
| IM0013 | Spaghetti | 720 | 10800.00 |
| IM0014 | Tomato Sauce | 200 | 1400.00 |
| IM0017 | Green Tea | 300 | 1800.00 |
| IM0018 | Instant Coffee | 180 | 2250.00 |
| IM0019 | Milk | 120 | 480.00 |
| IM0020 | Chocolate Bar | 400 | 4000.00 |
| IM0021 | Canned Tuna | 220 | 1210.00 |
| IM0022 | Peanut Butter | 170 | 1360.00 |
| IM0023 | Whole Wheat Bread | 280 | 980.00 |
| IM0024 | Frozen Pizza | 60 | 900.00 |
| IM0025 | Cheddar Cheese | 140 | 1260.00 |
| IM0026 | Yogurt | 110 | 715.00 |
| IM0027 | Granola Bars | 300 | 3600.00 |
| IM0028 | Pasta Sauce | 180 | 1800.00 |
| IM0029 | Frozen Vegetables | 90 | 720.00 |
| IM0030 | Butter | 240 | 1200.00 |
| IM0031 | Apple Juice | 130 | 910.00 |
| IM0032 | Honey | 100 | 1500.00 |
| IM0033 | Oats | 210 | 1785.00 |
| IM0034 | Canned Pineapple | 190 | 950.00 |
| IM0035 | Almond Milk | 150 | 975.00 |

```
Please select the ID of donation item ('l' for 'IM0001') ('0' to exit): 28
Please enter the donation quantity ('0' to exit): 120
Are you sure to donate 'Pasta Sauce' with amount 120? (Y/N): Y
```

Figure 1.4.3.2: Selecting the donation item and donation item quantity to be donated

```
Do you want to add more item? ('Y' - Yes; 'N' - No): n

+-----+
| Donation ID: | DN0007 |
+-----+
| Stock In Item: |
| IN0015     Olive Oil      x 150 |
| IN0016     Corn Flakes    x 300 |
| IN0046     Pasta Sauce    x 120 |
+-----+
| Notes:       | Support for education |
+-----+
| Donation date: | 26-08-2024 |
+-----+

+-----+
| Donor ID: | DR0024 |
+-----+
| Name:     | Care International |
+-----+
| Type:     | Government |
+-----+
The donation detail has been updated successfully.
```

Figure 1.4.3.3: Result of successful donation update (add new item)

1.4.4 Remove Item

```
Select an option ('0' to exit): 4

+-----+
| Stock In ID | Item Name           | Quantity | Total Amount (RM) |
+=====+
| IN0015      | Olive Oil            | 150      | RM          2700.00 |
+-----+
| IN0016      | Corn Flakes          | 300      | RM          2550.00 |
+-----+
| IN0046      | Pasta Sauce          | 120      | RM          1200.00 |
+-----+
Enter the stock in ID that you want to remove ('l' for 'IN0001') ('0' to exit):
```

Figure 1.4.4.1: List of donation item which can be deleted

```
Enter the stock in ID that you want to remove ('l' for 'IN0001')('0' to exit): 15

+-----+
| Stock In ID: | IN0015 |
+-----+
| Stock In Item Name: | Olive Oil |
+-----+
| Stock In Quantity: | 150 |
+-----+
| Total Amount: | RM 2700.00 |
+-----+

Are you sure to remove this item (IN0015) from this donation? ('Y' - Yes; 'N' - No): y

The item has been removed successfully.
```

Figure 1.4.4.2: Select the donated item to be deleted

1.4.5 Update Notes

```
Select an option ('0' to exit): 5

Current Notes:
Support for education

Enter new notes ('0' to exit):
Education supply
```

Figure 1.4.5.1: Update donation notes

```
+-----+
| Donation ID: | DN0007 |
+-----+
| Stock In Item:
| IN0015     Olive Oil      x 150 |
| IN0016     Corn Flakes    x 240 |
+-----+
| Notes:       | Education supply |
+-----+
| Donation date: | 26-08-2024 |
+-----+

+-----+
| Donor ID: | DR0024 |
+-----+
| Name:     | Care International |
+-----+
| Type:     | Government |
+-----+
The donation detail has been updated successfully.
```

Figure 1.4.5.2: Result of successful donation update (update notes)

1.5 Remove Donation

| Please select function (0 - 7): 5 | | | |
|-----------------------------------|-------------------------------|--------------|------------|
| Donation ID | Donor Name | Type | Date |
| DN0001 | Hope for All | Organization | 15-08-2024 |
| DN0002 | HappyP Sdn Bhd | Organization | 09-08-2024 |
| DN0003 | Global Charities | Organization | 01-08-2024 |
| DN0004 | Ministry of Health | Organization | 30-06-2024 |
| DN0005 | Emily Goh | Government | 20-07-2024 |
| DN0006 | Vanessa Tan | Organization | 18-08-2024 |
| DN0007 | Care International | Government | 26-08-2024 |
| DN0008 | Sophia Ng | Government | 22-07-2024 |
| DN0009 | Bright Minds Foundation | Organization | 10-07-2024 |
| DN0010 | Chloe Teo | Organization | 01-07-2024 |
| DN0011 | Save the Earth | Government | 25-06-2024 |
| DN0012 | Thomas Ong | Government | 19-05-2024 |
| DN0013 | Benjamin Lau | Government | 03-05-2024 |
| DN0014 | Nicholas Tan | Government | 15-04-2024 |
| DN0015 | Ministry of Finance | Government | 01-04-2024 |
| DN0016 | Youtuber Celine | Organization | 22-03-2024 |
| DN0017 | Thomas Ong | Government | 10-03-2024 |
| DN0018 | Empower Communities | Organization | 28-02-2024 |
| DN0019 | Ministry of Foreign Affairs | Organization | 15-02-2024 |
| DN0020 | Ministry of Rural Development | Organization | 30-01-2024 |

Please enter the donation ID you want to select ('1' for 'DN0001') ('0' to exit): 3

Figure 1.5.1: Selecting donation to be removed

```
Please enter the donation ID you want to select ('l' for 'DN0001') ('0' to exit): 3

+-----+
| Donation ID: | DN0003 |
+-----+
| Stock In Item: |
| IN0005    PayPal      RM 250.00 |
| IN0006    Mobile Payment RM 100.00 |
+-----+
| Notes:       | For winter season aid |
+-----+
| Donation date: | 01-08-2024 |
+-----+

Are you sure to remove this donation (DN0003) ?: ('Y' - Yes; 'N' - No): y|
```

Figure 1.5.2: Confirmation to remove the donation

| Donation ID | Donor Name | Type | Date |
|-------------|-------------------------------|--------------|------------|
| DN0001 | Hope for All | Organization | 15-08-2024 |
| DN0002 | HappyP Sdn Bhd | Organization | 09-08-2024 |
| DN0004 | Ministry of Health | Organization | 30-06-2024 |
| DN0005 | Emily Goh | Government | 20-07-2024 |
| DN0006 | Vanessa Tan | Organization | 18-08-2024 |
| DN0007 | Care International | Government | 26-08-2024 |
| DN0008 | Sophia Ng | Government | 22-07-2024 |
| DN0009 | Bright Minds Foundation | Organization | 10-07-2024 |
| DN0010 | Chloe Teo | Organization | 01-07-2024 |
| DN0011 | Save the Earth | Government | 25-06-2024 |
| DN0012 | Thomas Ong | Government | 19-05-2024 |
| DN0013 | Benjamin Lau | Government | 03-05-2024 |
| DN0014 | Nicholas Tan | Government | 15-04-2024 |
| DN0015 | Ministry of Finance | Government | 01-04-2024 |
| DN0016 | Youtuber Celine | Organization | 22-03-2024 |
| DN0017 | Thomas Ong | Government | 10-03-2024 |
| DN0018 | Empower Communities | Organization | 28-02-2024 |
| DN0019 | Ministry of Foreign Affairs | Organization | 15-02-2024 |
| DN0020 | Ministry of Rural Development | Organization | 30-01-2024 |

Donation has been deleted successfully.

Figure 1.5.3: Result of successful removing donation

1.6 Donation Report

```
Please select function (0 - 7): 6

+---+
| Report | Option |
+=====+=====+
| Top 10 Major Donation Report | 1 |
+-----+-----+
| Top 10 Donated Items Report | 2 |
+-----+-----+
| Top 10 Critical Shortages Report | 3 |
+-----+-----+

Please select a report ('0' to exit): |
```

Figure 1.6: Select type of donation report to review

1.6.1 Top 10 Major Donation Report

```
Please select a report ('0' to exit): 1

Enter donation start date (yyyy-mm-dd) ('0' to exit): 2024-1-30

Enter donation end date (yyyy-mm-dd) ('0' to exit): 2024-8-15
```

Figure 1.6.1.1: Prompt for donation date range

| Top 10 Major Donation Report | | | | | |
|------------------------------|----------|-------------------------------|--------------|---------------|--|
| Donation ID | Donor ID | Donor Name | Total Amount | Donation Date | |
| DN0005 | DR0007 | Emily Goh | RM 13060.00 | 20-07-2024 | |
| DN0017 | DR0006 | Thomas Ong | RM 10300.00 | 10-03-2024 | |
| DN0020 | DR0042 | Ministry of Rural Development | RM 10190.00 | 30-01-2024 | |
| DN0014 | DR0008 | Nicholas Tan | RM 4820.00 | 15-04-2024 | |
| DN0011 | DR0019 | Save the Earth | RM 4350.00 | 25-06-2024 | |
| DN0013 | DR0012 | Benjamin Lau | RM 3440.00 | 03-05-2024 | |
| DN0016 | DR0023 | Youtuber Celine | RM 3415.00 | 22-03-2024 | |
| DN0004 | DR0031 | Ministry of Health | RM 3150.00 | 30-06-2024 | |
| DN0019 | DR0040 | Ministry of Foreign Affairs | RM 2640.00 | 15-02-2024 | |
| DN0009 | DR0030 | Bright Minds Foundation | RM 2580.00 | 10-07-2024 | |


```
Do you want to generate other report? ('Y' - Yes; 'N' - No):
```

Figure 1.6.1.2: Top 10 Major Donated Report generated based on the specified donation date range

1.6.2 Top 10 Donated Items Report

```
Please select a report ('0' to exit): 2

+---+
| Category | Option |
+---+---+
| Money | 1 |
+---+---+
| Food and Beverages | 2 |
+---+---+
Please select the category option of donation item ('0' to back):
```

Figure 1.6.2: Selecting category of donated items report

1.6.2.1 Money Category

```
Please select the category option of donation item ('0' to back): 1

Enter donation start date (yyyy-mm-dd) ('0' to exit): 2024-2-15

Enter donation end date (yyyy-mm-dd) ('0' to exit): 2024-8-15
```

Figure 1.6.2.1.1: Prompt for donation date range

| Top 10 Donated Item Report (Money) | | | | |
|------------------------------------|----------------|--------------|---------------------|--|
| Item ID | Item Name | Total Amount | Major Contributor | |
| IM0002 | Bank Transfer | RM 950.00 | Hope for All | |
| IM0008 | Cryptocurrency | RM 600.00 | Ministry of Health | |
| IM0001 | Cash | RM 500.00 | Thomas Ong | |
| IM0003 | Cheque | RM 500.00 | Empower Communities | |
| IM0004 | Credit Card | RM 500.00 | HappyP Sdn Bhd | |
| IM0005 | PayPal | RM 250.00 | Global Charities | |
| IM0006 | Mobile Payment | RM 100.00 | Global Charities | |
| IM0007 | Gift Card | RM 50.00 | Ministry of Health | |

```
Do you want to generate other report? ('Y' - Yes; 'N' - No): |
```

Figure 1.6.2.1.2: Top 10 Donated Item Report (Money) based on the specified donation date range

1.6.2.2 Food and Beverage Category

```
Please select the category option of donation item ('0' to back): 2
Enter donation start date (yyyy-mm-dd) ('0' to exit): 2024-2-5
Enter donation end date (yyyy-mm-dd) ('0' to exit): 2024-7-30
```

Figure 1.6.2.2.1: Prompt for donation date range

| Top 10 Donated Item Report (Food & Beverage) | | | | | |
|--|------------------|----------|--------------|-------------------------|--|
| Item ID | Item Name | Quantity | Total Amount | Major Contributor | |
| IM0010 | Rice Pack | 820 | RM 20500.00 | Emily Goh | |
| IM0009 | Canned Beans | 520 | RM 5200.00 | Youtuber Celine | |
| IM0025 | Cheddar Cheese | 300 | RM 2700.00 | Save the Earth | |
| IM0021 | Canned Tuna | 280 | RM 1540.00 | Chloe Teo | |
| IM0011 | Bottled Water | 250 | RM 3000.00 | Emily Goh | |
| IM0028 | Pasta Sauce | 240 | RM 2400.00 | Benjamin Lau | |
| IM0034 | Canned Pineapple | 230 | RM 1150.00 | Ministry of Finance | |
| IM0019 | Milk | 220 | RM 880.00 | Bright Minds Foundation | |
| IM0031 | Apple Juice | 210 | RM 1470.00 | Nicholas Tan | |
| IM0032 | Honey | 190 | RM 2850.00 | Nicholas Tan | |

Do you want to generate other report? ('Y' - Yes; 'N' - No):

Figure 1.6.2.2.2: Top 10 Donated Item Report (Food & Beverage) based on the specified donation date range

1.6.3 Top 10 Critical Shortages Report

```
Please select a report ('0' to exit): 3

+-----+
| Category | Option |
+=====+=====+
| Money    | 1      |
+-----+
| Food and Beverages | 2      |
+-----+
Please select the category option of donation item ('0' to back):
```

Figure 1.6.3: Selecting category of donated items report

1.6.3.1 Money Category

```
Please select the category option of donation item ('0' to back): 1
Enter the threshold amount of donation item (RM) ('0' to exit): 300
```

Figure 1.6.3.1.1: Prompt for threshold amount of donation item

| Top 10 Critical Shortage Report (Money) | | | | | |
|---|----------------|----|----------------|----|-----------------|
| Threshold Amount (RM) : 300.00 | | | | | |
| Item ID | Item Name | | Current Amount | | Amount Required |
| IM0001 | Cash | RM | 200.00 | RM | 100.00 |
| IM0003 | Cheque | RM | 150.00 | RM | 150.00 |
| IM0005 | PayPal | RM | 250.00 | RM | 50.00 |
| IM0006 | Mobile Payment | RM | 100.00 | RM | 200.00 |
| IM0007 | Gift Card | RM | 50.00 | RM | 250.00 |

Do you want to generate other report? ('Y' - Yes; 'N' - No):

Figure 1.6.3.1.2: Top 10 Critical Shortages Report (Money) based on the specified threshold amount

1.6.3.2 Food and Beverage Category

```
Please select the category option of donation item ('0' to back): 2
Enter the threshold quantity of donation item ('0' to exit): 500
```

Figure 1.6.3.2.1: Prompt for threshold quantity of donation item

| Top 10 Critical Shortage Report (Food & Beverage) | | | | | |
|---|-------------------|--|------------------|--|-------------------|
| Threshold Quantity: 500 | | | | | |
| Item ID | Item Name | | Current Quantity | | Quantity Required |
| IM0024 | Frozen Pizza | | 60 | | 440 |
| IM0029 | Frozen Vegetables | | 90 | | 410 |
| IM0032 | Honey | | 100 | | 400 |
| IM0026 | Yogurt | | 110 | | 390 |
| IM0019 | Milk | | 120 | | 380 |
| IM0031 | Apple Juice | | 130 | | 370 |
| IM0025 | Cheddar Cheese | | 140 | | 360 |
| IM0035 | Almond Milk | | 150 | | 350 |
| IM0022 | Peanut Butter | | 170 | | 330 |
| IM0018 | Instant Coffee | | 180 | | 320 |

Do you want to generate other report? ('Y' - Yes; 'N' - No): |

Figure 1.6.3.2.2: Top 10 Critical Shortages Report (Food &Beverage) based on the specified threshold quantity

1.7 Create new donation item

```
Please select function (0 - 7): 7

Enter the name of the item ('0' to exit): Granola Bars

+-----+
| Category | Option |
+-----+-----+
| Money | 1 |
+-----+-----+
| Food and Beverages | 2 |
+-----+
Please select the category option of donation item ('0' to back): 2

Enter the description of the item ('0' to exit session): Box of 24 granola bars

Enter the initial quantity of the item ('0' to exit session): 150

Enter the amount per item (RM) ('0' to exit session): 6
```

Figure 1.7.1: Prompt for new donation item info

```
+--+
| Item ID: | IM0036 |
+--+
| Item Name: | Granola Bars |
+--+
| Category: | Food and Beverages |
+--+
| Description: | Box of 24 granola bars |
+--+
| Quantity: | 150 |
+--+
| Price Per Item: | RM 6.00 |
+--+
| Total Amount: | RM 900.00 |
+--+
The donation item has been created successfully.

Do you want to continue create more item? ('Y' - Yes; 'N' - No):
```

Figure 1.7.2: Result of successful creation of new donation item

<INSTRUCTIONS> (please delete all instructions before submitting your assignment report)

<remove instructions>

- i. Ensure that the **author name** of each class is stated in the comment section of each classes,
- ii. **Copy & Paste** of your source code listing in NetBeans(source codes) that include the use of collection ADTs.
- iii. For Source Code listing, use **font color blue** to all lines of code that involve the **declaration or creation of collection ADT objects (control class, Dao)**, and the **invocation of collection ADT methods**.

</INSTRUCTIONS>

| Name | Student ID | Prog / Tut.Grp | Signature |
|------------|------------|----------------|---|
| Ng Ian Kai | 24WMR09090 | RSD1 G5 |  |

Subsystem : Donation Distribution Subsystem

3. Source codes Listing.
 - a. Controller: **DonationDistributionControl.java**

```
package control;

import adt.DoublyLinkedList;
import adt.ListInterface;
import utility.Node;
import dao.Initializer;
import static boundary.CLONCharity.h;
import static boundary.CLONCharity.v;
import static control.DoneeControl.updateTotalDonationDistribution;
import entity.DonationDistribution;
import entity.DonationDistribution.StatusHistory;
import entity.Donee;
import entity.InventoryMovement;
import entity.Item;
import java.time.LocalDate;
```

```

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.time.format.DateTimeParseException;
import java.util.Comparator;

/**
 *
 * @author Ng Ian Kai
 */
public class DonationDistributionControl {

    private static ListInterface<Donee> doneeList =
        Initializer.doneeListP.merge(Initializer.doneeListO.merge(Initializer.doneeListF));
    private static ListInterface<Item> foodBeverageList = Initializer.foodBeverageList;
    private static ListInterface<Item> moneyList = Initializer.moneyList;
    private static ListInterface<Item> mergedList = moneyList.merge(foodBeverageList);
    private static ListInterface<DonationDistribution> distributionList = Initializer.distributionList;

    public static void distributionMenu() {
        int optionInt;

        do {
            System.out.print("\n      Donation Distribution Subsystem\n"
                + " +-----+\\n"
                + " | Function       | Option |\\n"
                + " +-----+\\n"
                + " | Add new distribution   | 1  |\\n"
                + " +-----+-----+\\n"
                + " | View distributions     | 2  |\\n"
                + " +-----+-----+\\n"
                + " | Update distribution details | 3  |\\n"
                + " +-----+-----+\\n"
                + " | Remove distribution     | 4  |\\n"
                + " +-----+-----+\\n"
                + " | Track distribution status | 5  |\\n"
                + " +-----+-----+\\n"
                + " | Generate distribution report | 6  |\\n"
                + " +-----+-----+\\n"
                + " | Back to main menu       | 0  |\\n"
                + " +-----+-----+\\n"
                + " Please select function (0 - 6): ");

            optionInt = h.scanInt();

            switch (optionInt) {
                case 0:
                    System.out.println("\n Back to main menu page...\\n");
            }
        } while (optionInt != 0);
    }
}

```

```
    h.sleep();
    break;

case 1:
    addDistribution();
    break;

case 2:
    viewDistributionFunction();
    break;

case 3:
    updateDistribution();
    break;

case 4:
    removeDistribution();
    break;

case 5:
    trackDistribution();
    break;

case 6:
    generateReport();
    break;

default:
    System.out.println(v.errOptionInt);
}
} while (optionInt != 0);

return;
}

private static void addDistribution() {
    //Select Donee to distribute
    displayDonees();
    System.out.print("Select a Donee (0 to back) > ");
    int doneePosition = h.nextInt();
    if (doneePosition <= 0 || doneePosition > doneeList.getSize()) {
        if (doneePosition != 0) {
            System.out.println(v.errOptionInt);
        }
        return;
    }
}
```

```

Donee selectedDonee = selectDonee(doneePosition);

//Select Items for the distribution
ListInterface<InventoryMovement> movementItems = new DoublyLinkedList<>();

while (true) {
    System.out.println("\n      Select item(s) to distribute: ");
    System.out.println("-----");
    displayItems();
    System.out.println("-----");
    System.out.print("Select item to distribute (enter 0 to finish) > ");
    int itemPosition = h.scanInt();
    if (itemPosition <= 0 || itemPosition > mergedList.getSize()) {
        if (itemPosition != 0) {
            System.out.println(v.errOptionInt);
            return;
        }
        break;
    }
    Item selectedItem = selectItem(itemPosition);
    if (movementItems.contains(new InventoryMovement(selectedItem, 0, 0))) {
        System.out.println("Cannot add the same item again!");
        break;
    }

    int quantity = 0, isValid = 1;
    double amount = 0.0;
    do {
        if (selectedItem.getName().equals("Cash") || selectedItem.getName().equals("Bank Transfer")) {
            if (selectedItem.getAmount() <= 0) {
                System.out.println("No more amount!");
                break;
            }

            System.out.print("Enter Amount to distribute > ");
            amount = h.scanInt();

            if (amount <= 0 || amount > selectedItem.getAmount()) {
                System.out.println("Please enter a valid amount!");
                isValid = 0;
            } else {
                isValid = 1;
            }
        } else {
            if (selectedItem.getQty() <= 0) {
                System.out.println("No more quantity!");
                break;
            }
        }
    } while (!isValid);
}

```

```

}

System.out.print("Enter quantity to distribute > ");
quantity = h.scanInt();

if (quantity <= 0 || quantity > selectedItem.getQty()) {
    System.out.println("Please enter a valid quantity!");
    isValid = 0;
} else {
    isValid = 1;
}

amount = selectedItem.getAmount();
}

} while (isValid == 0);

if (selectedItem.getQty() != 0) {
    if (quantity > 0) {
        movementItems.add(new InventoryMovement(selectedItem, quantity, amount));
    }
} else {
    if (amount > 0) {
        movementItems.add(new InventoryMovement(selectedItem, quantity, amount));
    }
}
}

if (movementItems.getSize() > 0) {
    DonationDistribution distribution = new DonationDistribution(selectedDonee, movementItems);
    distributionList.add(distribution);

    Node<InventoryMovement> currentMovementNode = movementItems.searchNode(1);
    while (currentMovementNode != null) {
        Item currentItem = mergedList.search(currentMovementNode.getData().getItem());
        InventoryMovement currentMovementItem = currentMovementNode.getData();
        if (currentMovementNode.getData().getItemQuantity() != 0) {
            currentItem.setQty(currentItem.getQty() - currentMovementItem.getItemQuantity());
        } else {
            currentItem.setAmount(currentItem.getAmount() - currentMovementItem.getMoneyAmount());
        }

        currentMovementNode = currentMovementNode.getNext();
    }
}

System.out.println("Donation Distribution added successfully!");
h.pressAnyKeyToContinue();

```

```

    }

private static void viewDistributionFunction() {
    StringBuilder listOfDistributions = new StringBuilder();
    listOfDistributions.append("\n          List of Donation Distributions \n");

    Node<DonationDistribution> currentNode = distributionList.searchNode(1);
    while (currentNode != null) {
        DonationDistribution distribution = currentNode.getData();
        ListInterface<InventoryMovement> movementItems = distribution.getMovementItems();

        listOfDistributions.append("|----|-----|-----|-----|-----|\n");
        listOfDistributions.append("| ID | Distribution Date | Donee Name | Tracking Status | DONATIONS
COUNT |\n");
        listOfDistributions.append("|----|-----|-----|-----|-----|\n");
        listOfDistributions.append(String.format("| %02d | %13s | %7s | %10s | %5d items |%\n",
            distribution.getId(),
            distribution.getDate(),
            distribution.getDonee().getName(),
            distribution.getCurrentTrackingStatus(),
            movementItems.getSize()
        ));
    }

    listOfDistributions.append("|\n      Items: |\n" + "\n|Items:\n|\n");
}

Node<InventoryMovement> movementNode = movementItems.searchNode(1);
int index = 1;
while (movementNode != null) {
    InventoryMovement movement = movementNode.getData();
    Item item = movement.getItem();
    String itemDetails;

    if (item.getName().equals("Cash") || item.getName().equals("Bank Transfer")) {
        itemDetails = String.format("| %02d. %-20s AMT: RM%-10.2f\n", index,
        item.getName(), movement.getMoneyAmount());
    } else {
        itemDetails = String.format("| %02d. %-20s Qty: %-10d\n", index,
        item.getName(), movement.getItemQuantity());
    }
    listOfDistributions.append(itemDetails);
    movementNode = movementNode.getNext();
    index++;
}

listOfDistributions.append("|\n-----|\n\n");

```

```

        currentNode = currentNode.getNext();
    }

    System.out.println(listOfDistributions.toString());

    h.pressAnyKeyToContinue();
}

private static void viewDistributions() {
    int optionInt;

    do {
        Node<DonationDistribution> currentNode = distributionList.searchNode(1);
        Node<InventoryMovement> currentMovementNode;

        System.out.println("\n                  Donation Distribution List          ");
        System.out.println(" |---|-----|-----|-----|-----|");
        System.out.println(" | ID | DATE OF DISTRIBUTION | DONEE NAME | TRACKING STATUS | DONATIONS
COUNT |");
        System.out.println(" |---|-----|-----|-----|-----|");
        while (currentNode != null) {
            System.out.printf(" | %02d | %15s   | %7s   | %11s   | %4d item(s)   |\n",
                currentNode.getData().getId(),
                currentNode.getData().getDate(),
                currentNode.getData().getDonee().getName(),
                currentNode.getData().getCurrentTrackingStatus(),
                currentNode.getData().getMovementItems().getSize());
            currentNode = currentNode.getNext();
        }

        System.out.println(" |---|-----|-----|-----|-----|\n");

        System.out.print("Select the index of the distribution to view the distribution details [1.." +
distributionList.getSize() + "] (enter anything else to skip) > ");
        optionInt = h.scanInt();

        if (optionInt <= 0 || optionInt > distributionList.getSize()) {
            break;
        } else {
            currentNode = distributionList.searchNode(optionInt);
            System.out.println("\n                  Distribution #" + optionInt);
            System.out.println("=====");
            System.out.println("1. Date of Distribution: " + currentNode.getData().getDate());
            System.out.println("2. Tracking Status: " + currentNode.getData().getCurrentTrackingStatus());
            System.out.println("3. Donee Name: " + currentNode.getData().getDonee().getName());
            System.out.println("4. Distributed Items: ");
        }
    }
}

```

```

currentMovementNode = currentNode.getData().getMovementItems().searchNode(1);
int index = 1;
while (currentMovementNode != null) {
    System.out.println(
        "\tItem " + index + " = " + currentMovementNode.getData().getItem().getName()
        + " (QTY: " + currentMovementNode.getData().getItemQuantity() + ")"
        + " (AMT: " + currentMovementNode.getData().getMoneyAmount() + ")"
    );
    index++;
    currentMovementNode = currentMovementNode.getNext();
}
System.out.println("=====");

} while (optionInt != 0);
}

private static void updateDistribution() {
    viewDistributions();
    System.out.print("Select the index of the distribution you wish to update [1.." + distributionList.getSize() +
"] (0 to back) >");
    int optionInt = h.scanInt();

    if (optionInt <= 0 || optionInt > distributionList.getSize()) {
        return;
    } else {
        Node<DonationDistribution> currentNode = distributionList.searchNode(optionInt);

        if (currentNode != null) {
            System.out.print("Select the index (REFER TO THE DISTRIBUTION DETAILS) of the detail you wish to
update [1..4] (0 to finish) >");
            optionInt = h.scanInt();

            switch (optionInt) {
                case 0:
                    break;

                case 1:
                    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
                    String dateInput;
                    LocalDate newDate = null;

                    do {
                        System.out.print("Update distribution date to (yyyy-MM-dd) >");
                        dateInput = h.scanStr();

```

```

try {
    newDate = LocalDate.parse(dateInput, formatter); // Try to parse the input date
} catch (DateTimeParseException e) {
    System.out.println("Invalid format! Please use the format yyyy-MM-dd.");
    newDate = null; // Reset newDate to null if parsing fails
}
} while (newDate == null);

currentNode.getData(). setDate(newDate);
System.out.println("Updated Successfully!");
break;

case 2:
String newStatus = null;
LocalDateTime dateTime = LocalDateTime.now();

System.out.println("\nUpdate Tracking Status To");
System.out.println("=====");
System.out.println("1. Pending");
System.out.println("2. In Transit");
System.out.println("3. Delivered");
System.out.println("=====");
System.out.print("Select the index of the tracking status you wish to update to [1..3] (0 to cancel)
>");

switch (h.nextInt()) {
    case 0:
        break;

    case 1:
        newStatus = "Pending";
        break;

    case 2:
        newStatus = "In Transit";
        break;

    case 3:
        newStatus = "Delivered";
        break;

    default:
        System.out.println(v.errOptionInt);
}

if (newStatus != null) {
    currentNode.getData().setCurrentTrackingStatus(newStatus);
}

```

```

System.out.println("Updated Successfully!");

// Add the status update to the history
DonationDistribution.StatusHistory statusHistory = currentNode.getData().new
StatusHistory(dateTime, newStatus);
    currentNode.getData().getStatusHistory().add(statusHistory);
}
break;

case 3:
    displayDonees();

    System.out.print("Select the index of the donee to update to [1.." + doneeList.getSize() + "] (0 to
cancel) > ");
    optionInt = h.scanInt();

    if (optionInt <= 0 || optionInt > doneeList.getSize()) {
        if (optionInt != 0) {
            System.out.println(v.errOptionInt);
        }
        break;
    }

    currentNode.getData().setDonee(doneeList.searchNode(optionInt).getData());
    System.out.println("Updated Successfully!");
    break;

case 4:
    System.out.println("\n      Update Distribution Items");
    System.out.println("=====");
    System.out.println("1. Add new distribution item");
    System.out.println("2. Update existing distribution item");
    System.out.println("3. Remove existing distribution item");
    System.out.println("=====");
    System.out.print("Select the index of the update you want to perform [1..3] (0 to cancel) > ");

    optionInt = h.scanInt();
    switch (optionInt) {
        case 0:
            break;

        case 1:
            System.out.println("\n  Items available to add into the distribution");
            System.out.println("-----");
            displayItems();
            System.out.println("-----");
            System.out.print("Select index of an item to add to this distribution (0 to cancel) > ");
    }
}

```

```

optionInt = h.scanInt();
if (optionInt <= 0 || optionInt > mergedList.getSize()) {
    if (optionInt != 0) {
        System.out.println(v.errOptionInt);
    }
    break;
}
Item selectedItem = selectItem(optionInt);

int quantity = 0,
isValid = 1;
double amount = 0.0;
do {
    if (selectedItem.getName().equals("Cash") || selectedItem.getName().equals("Bank
Transfer")) {
        System.out.print("Enter Amount to distribute > ");
        amount = h.scanInt();

        if (amount <= 0 || amount > selectedItem.getAmount()) {
            System.out.println("Please enter a valid amount!");
            isValid = 0;
        } else {
            isValid = 1;
        }
    } else {
        System.out.print("Enter quantity to distribute > ");
        quantity = h.scanInt();

        if (quantity <= 0 || quantity > selectedItem.getQty()) {
            System.out.println("Please enter a valid quantity!");
            isValid = 0;
        } else {
            isValid = 1;
        }
        amount = selectedItem.getAmount();
    }
} while (isValid == 0);

InventoryMovement movementItem = new InventoryMovement(selectedItem, quantity,
amount);
if (currentNode.getData().getMovementItems().contains(new
InventoryMovement(selectedItem, 0, 0))) {
    System.out.println("Cannot add the same item again!");
    break;
}

```

```

currentNode.getData().addMovementItem(movementItem);

Item currentItem = mergedList.search(movementItem.getItem());
if (movementItem.getItemQuantity() != 0) {
    currentItem.setQty(currentItem.getQty() - movementItem.getItemQuantity());
} else {
    currentItem.setAmount(currentItem.getAmount() - movementItem.getMoneyAmount());
}

System.out.println("Added Successfully!");
break;

case 2:
    Node<InventoryMovement> currentMovementNode =
currentNode.getData().getMovementItems().searchNode(1);

    System.out.println("\n      Items in current distribution");

System.out.println("=====");

    int index = 1;
    while (currentMovementNode != null) {

        System.out.println(
            "\tItem " + index + " = " + currentMovementNode.getData().getItem().getName()
            + " (QTY: " + currentMovementNode.getData().getItemQuantity() + ")"
            + " (AMT: " + currentMovementNode.getData().getMoneyAmount() + ")"
        );

        index++;
        currentMovementNode = currentMovementNode.getNext();
    }

System.out.println("=====");
    System.out.print("Select the index of the item you want to update [1.." +
currentNode.getData().getMovementItems().getSize() + "] > ");
    optionInt = h.scanInt();
    if (optionInt <= 0 || optionInt > currentNode.getData().getMovementItems().getSize()) {
        if (optionInt != 0) {
            System.out.println(v.errOptionInt);
        }
        break;
    }

    currentMovementNode =

```

```

currentNode.getData().getMovementItems().searchNode(optionInt);
    Item selectedMovementItem = null;
    Node<Item> currentItemNode = mergedList.searchNode(1);

    while (currentItemNode != null) {
        if
(currentMovementNode.getData().getItem().getName().equals(currentItemNode.getData().getName())) {
            selectedMovementItem = currentItemNode.getData();
            break;
        }

        currentItemNode = currentItemNode.getNext();
    }

    int currentItemQuantity = currentMovementNode.getData().getItemQuantity();
    double currentMoneyAmount = currentMovementNode.getData().getMoneyAmount();

    if (currentMovementNode.getData().getItem().getName().equals("Cash") || currentMovementNode.getData().getItem().getName().equals("Bank Transfer")) {
        do {
            System.out.print("Enter the updated money amount > ");
            optionInt = h.scanInt();

            if (optionInt <= 0 || optionInt > selectedMovementItem.getAmount() + currentMovementNode.getData().getMoneyAmount()) {
                System.out.println("Please enter a valid amount!");
            } else {
                currentMovementNode.getData().setMoneyAmount(optionInt);

            }
        } while (optionInt <= 0 || optionInt > selectedMovementItem.getAmount() + currentMovementNode.getData().getMoneyAmount());
    } else {
        do {
            System.out.print("Enter the updated item quantity > ");
            optionInt = h.scanInt();

            if (optionInt <= 0 || optionInt > selectedMovementItem.getQty() + currentMovementNode.getData().getItemQuantity()) {
                System.out.println("Please enter a valid quantity!");
            } else {
                currentMovementNode.getData().setItemQuantity(optionInt);
            }
        } while (optionInt <= 0 || optionInt > selectedMovementItem.getQty() + currentMovementNode.getData().getItemQuantity());
    }
}

```

```

currentItem = mergedList.search(currentMovementNode.getData().getItem());
if (currentItem.getQty() != 0) {
    currentItem.setQty(currentItem.getQty() - optionInt + currentItemQuantity);
} else {
    currentItem.setAmount(currentItem.getAmount() - optionInt + currentMoneyAmount);
}
System.out.println("Updated Successfully!");
break;

case 3:
    currentMovementNode = currentNode.getData().getMovementItems().searchNode(1);

    System.out.println("\n      Items in current distribution");

System.out.println("=====");
index = 1;
while (currentMovementNode != null) {

    System.out.println(
        "\tItem " + index + " = " + currentMovementNode.getData().getItem().getName()
        + " (QTY: " + currentMovementNode.getData().getItemQuantity() + ")"
        + " (AMT: " + currentMovementNode.getData().getMoneyAmount() + ")"
    );
    index++;
    currentMovementNode = currentMovementNode.getNext();
}

System.out.println("=====");
System.out.print("Select the index of the item you want to delete [1.." +
currentNode.getData().getMovementItems().getSize() + "] (0 to cancel) > ");
optionInt = h.scanInt();
if (optionInt <= 0 || optionInt > currentNode.getData().getMovementItems().getSize()) {
    if (optionInt != 0) {
        System.out.println(v.errOptionInt);
    }
    break;
}
currentMovementNode =
currentNode.getData().getMovementItems().searchNode(optionInt);
currentNode.getData().removeMovementItem(currentMovementNode.getData());
System.out.println("Deleted Successfully!");
break;

default:

```

```

        System.out.println(v.errOptionInt);
    }

    break;

default:
    System.out.println(v.errOptionInt);
}
h.pressAnyKeyToContinue();
}

}

private static void removeDistribution() {
    viewDistributions();
    System.out.print("Select the index of the distribution you wish to remove [1.." + distributionList.getSize()
+ "] (0 to back) > ");
    int optionInt = h.scanInt();

    if (optionInt <= 0 || optionInt > distributionList.getSize()) {
        if (optionInt != 0) {
            System.out.println(v.errOptionInt);
        }
        return;
    } else {
        Node<DonationDistribution> currentNode = distributionList.searchNode(optionInt);
        Node<DonationDistribution> nextNode = currentNode.getNext();
        distributionList.remove(currentNode.getData());

        int currentLastId = DonationDistribution.getLastID();
        DonationDistribution.setLastID(currentLastId - 1);

        while (nextNode != null) {
            nextNode.getData().setId(nextNode.getData().getId() - 1);

            nextNode = nextNode.getNext();
        } // Decrement the ID of subsequent entries after the removed entry by 1

        System.out.println("Removed Successfully!");
        h.pressAnyKeyToContinue();
    }
}

private static void trackDistribution() {
    int optionInt;

```

```

System.out.println("\n=====");
System.out.println("1. Filter by Status");
System.out.println("2. View Tracking History");
System.out.println("=====");
System.out.print("Select the index of the function you want to choose [1..2] (0 to back) > ");
optionInt = h.scanInt();

switch (optionInt) {
    case 1:
        System.out.print("Enter status to filter by (Pending/In Transit/Delivered) > ");
        String status = h.scanStr();

        ListInterface<DonationDistribution> filteredList = filterByStatus(distributionList, status);
        System.out.println("\n  Filtered List:");
        System.out.println(filteredList.toString());
        break;

    case 2:
        viewDistributions();
        System.out.print("Enter the index of the distribution you wish to view its tracking history (0 to cancel)
> ");
        optionInt = h.scanInt();
        if (optionInt <= 0 || optionInt > distributionList.getSize()) {
            if (optionInt != 0) {
                System.out.println(v.errOptionInt);

                break;
            }
        }

        DonationDistribution distribution = distributionList.searchNode(optionInt).getData();
        if (distribution != null) {
            System.out.println("\n          Tracking History");

System.out.println("=====");
=====");

        ListInterface<StatusHistory> history = distribution.getStatusHistory();
        for (int i = 1; i <= history.getSize(); i++) {
            System.out.println(history.searchNode(i).getData().toString());
        }
    } else {
        System.out.println("Distribution not found.");
    }

System.out.println("=====");

```

```

=====
break;

default:
    if (optionInt != 0) {
        System.out.println("Invalid Option!");
    }
}

if (optionInt != 0) {
    h.pressAnyKeyToContinue();
}
}

private static void generateReport() {
    updateTotalDonationDistribution(null, null);

    ListInterface<DonationDistribution> sortedList = DonationDistributionControl.distributionList;

    System.out.println("\n      Types of Summary Report");
    System.out.println("=====-----");
    System.out.println("1. Top 10 distributions with highest value");
    System.out.println("2. Top 10 distributions with most donation items");
    System.out.println("=====-----");
    System.out.print("Please select the index of the report you want to generate [1..2] (0 to back) > ");
    int optionInt = h.scanInt();

    switch (optionInt) {
        case 1:
            sortedList.bubbleSort(false, new Comparator<DonationDistribution>() {
                @Override
                public int compare(DonationDistribution d1, DonationDistribution d2) {
                    return Double.compare(d1.calculateDistributionValue(), d2.calculateDistributionValue());
                }
            }); //Sort by Descending (Highest value to lowest)

            System.out.println("\n      Top 10 Distributions with highest value");
            System.out.println("|-----|-----|-----|-----|-----|-----|");
            System.out.println("| Rank | Distribution ID | Distribution Date | Donee Name | Distribution Value |");
            System.out.println("Tracking Status |");
            System.out.println("|-----|-----|-----|-----|-----|");

            Node<DonationDistribution> currentNode = sortedList.searchNode(1);
            for (int i = 1; i <= 10 && i <= sortedList.getSize(); i++) {
                DonationDistribution distribution = currentNode.getData();
                System.out.printf(" | %d | %6s | %14s | %6s | RM%-10.2f | %12s | \n",
                    i,

```

```

        distribution.getIdStr(),
        distribution.getDate(),
        distribution.getDonee().getName(),
        distribution.calculateDistributionValue(),
        distribution.getCurrentTrackingStatus());

    currentNode = currentNode.getNext();
}
System.out.println("-----|");

break;

case 2:
// Sort by Descending order of item count (highest to lowest)
sortedList.bubbleSort(false, new Comparator<DonationDistribution>() {
    @Override
    public int compare(DonationDistribution d1, DonationDistribution d2) {
        return Integer.compare(d1.getMovementItems().getSize(), d2.getMovementItems().getSize());
    }
});

System.out.println("\n          Top 10 Distributions with Most Donation Items");
System.out.println("-----|-----|-----|-----|-----|-----|-----|");
System.out.println(" | Rank | Distribution ID | Distribution Date | Donee Name | Donation Counts |"
Tracking Status |");
System.out.println(" |-----|-----|-----|-----|-----|-----|-----|");

currentNode = sortedList.searchNode(1);
for (int i = 1; i <= 10 && i <= sortedList.getSize(); i++) {
    DonationDistribution distribution = currentNode.getData();
    System.out.printf(" | %d | %6s | %14s | %6s | %6d | %12s | \n",
        i,
        distribution.getIdStr(),
        distribution.getDate(),
        distribution.getDonee().getName(),
        distribution.getMovementItems().getSize(),
        distribution.getCurrentTrackingStatus());

    currentNode = currentNode.getNext();
}
System.out.println("-----|");
break;

default:
if (optionInt != 0) {
    System.out.println("Invalid Option!");
}
}

```

```

}

if (optionInt != 0) {
    h.pressAnyKeyToContinue();
}
}

private static void displayDonees() {
    Node<Donee> currentNode = doneeList.searchNode(1);

    System.out.println("\n  Donee List");
    System.out.println("-=====");
    int index = 1;
    while (currentNode != null) {
        System.out.println(index + ". " + currentNode.getData().getName());
        currentNode = currentNode.getNext();
        index++;
    }
    System.out.println("-=====");
}

private static Donee selectDonee(int position) {
    Node<Donee> currentNode = doneeList.searchNode(position);

    return ((currentNode != null) ? currentNode.data : null);
}

private static void displayItems() {
    Node<Item> currentNode;

    currentNode = mergedList.searchNode(1);
    int index = 1;
    while (currentNode != null) {
        if (currentNode.getData().getName().equals("Cash") || currentNode.getData().getName().equals("Bank Transfer")) {
            System.out.println(index + ". " + currentNode.getData().getName() + " (AMT: " +
currentNode.getData().getAmount() + ")");
        } else {
            System.out.println(index + ". " + currentNode.getData().getName() + " (QTY: " +
currentNode.getData().getQty() + ")");
        }
        currentNode = currentNode.getNext();
        index++;
    }
}

private static Item selectItem(int itemIndex) {
}

```

```

Node<Item> currentNode = mergedList.searchNode(itemIndex);

return ((currentNode != null) ? currentNode.data : null);
}

private static ListInterface<DonationDistribution> filterByStatus(ListInterface<DonationDistribution>
distributionList, String status) {
    ListInterface<DonationDistribution> filteredList = new DoublyLinkedList<>();

    Node<DonationDistribution> currentNode = distributionList.searchNode(1);
    while (currentNode != null) {
        if (currentNode.getData().getCurrentTrackingStatus().equalsIgnoreCase(status)) {
            filteredList.add(currentNode.getData());
        }

        currentNode = currentNode.getNext();
    }

    return filteredList;
}
}

```

b. Entity: DonationDistribution.java

```

package entity;

import adt.DoublyLinkedList;
import java.time.LocalDate;
import adt.ListInterface;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

/**
 *
 * @author ianka
 */
public class DonationDistribution implements Comparable<DonationDistribution> {

    private static int lastID = 0; // Static variable to keep track of the latest assigned id

    private int id;

```

```
private LocalDate date;
private String currentTrackingStatus;
private Donee donee;
private ListInterface<InventoryMovement> movementItems;
private ListInterface<StatusHistory> statusHistory;

public DonationDistribution() {
    this.id = 0;
    this.date = null;
    this.currentTrackingStatus = null;
    this.donee = null;
    this.movementItems = null;
    this.statusHistory = new DoublyLinkedList<>();
}

public DonationDistribution(Donee donee, ListInterface<InventoryMovement> movementItems) {
    this.id = ++lastID;
    this.date = LocalDate.now();
    this.currentTrackingStatus = "Pending";
    this.donee = donee;
    this.movementItems = movementItems;
    this.statusHistory = new DoublyLinkedList<>();
}

public static int getLastID() {
    return lastID;
}

public static void setLastID(int lastID) {
    DonationDistribution.lastID = lastID;
}

public int getId() {
    return id;
}

public String getIdStr() {
    return String.format("DI%04d", id);
}

public void setId(int id) {
    this.id = id;
}

public LocalDate getDate() {
    return date;
}
```

```
public void setDate(LocalDate date) {
    this.date = date;
}

public String getCurrentTrackingStatus() {
    return currentTrackingStatus;
}

public void setCurrentTrackingStatus(String currentTrackingStatus) {
    this.currentTrackingStatus = currentTrackingStatus;
}

public Donee getDonee() {
    return donee;
}

public void setDonee(Donee donee) {
    this.donee = donee;
}

public ListInterface<InventoryMovement> getMovementItems() {
    return movementItems;
}

public void setMovementItems(ListInterface<InventoryMovement> movementItems) {
    this.movementItems = movementItems;
}

public void addMovementItem(InventoryMovement movementItem) {
    this.movementItems.add(movementItem);
}

public void removeMovementItem(InventoryMovement movementItem) {
    this.movementItems.remove(movementItem);
}

public ListInterface<StatusHistory> getStatusHistory() {
    return statusHistory;
}

public void setStatusHistory(ListInterface<StatusHistory> statusHistory) {
    this.statusHistory = statusHistory;
}

public double calculateDistributionValue() {
    return donee.getTotalItemValueReceived() + donee.getTotalMoneyReceived();
```

```

}

@Override
public int compareTo(DonationDistribution o) {
    return id == o.getId() ? 0 : -1;
}

@Override
public String toString() {
    return String.format(
        " +-----+-----+\n"
        "+ " | %-20s: | %30s | \n"
        "+ +-----+-----+\n"
        "+ " | %-20s: | %30s | \n"
        "+ +-----+-----+\n"
        "+ " | %-54s | \n"
        "+ " |           | \n"
        "+ "%s"
        "+ " |           -----+\n"
        "+ " | %23s %-17s: %11.2f | \n"
        "+ " |           -----+\n"
        "+ " |           | \n"
        "+ +-----+-----+\n"
        "+ " | %-20s: | %30s | \n"
        "+ +-----+-----+\n"
        "+ " | %-20s: | %30s | \n"
        "+ +-----+-----+\n\n",
        "Distribution ID", getIdStr(),
        "Donee", donee.getName() + " - " + donee.getIdStr(),
        "Stock Out Item:",
        movementItems.toString(),
        "", "Total value (RM)", calculateDistributionValue(),
        "Distribution Date", date.toString(),
        "Status", currentTrackingStatus);
}

public class StatusHistory implements Comparable<StatusHistory> {

    private LocalDateTime dateTime;
    private String statusChange;

    public StatusHistory(LocalDateTime dateTime, String statusChange) {
        this.dateTime = dateTime;
        this.statusChange = statusChange;
    }

    public LocalDateTime getDateTime() {
}

```

```
    return dateTIme;
}

public void setDateTIme(LocalDateTime dateTIme) {
    this.dateTIme = dateTIme;
}

public String getStatusChange() {
    return statusChange;
}

public void setStatusChange(String statusChange) {
    this.statusChange = statusChange;
}

@Override
public String toString() {
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
    return "{ DateTime of status change: " + dateTIme.format(formatter)
        + ", Status changed to: " + statusChange + " }";
}

@Override
public int compareTo(StatusHistory o) {
    return this.dateTIme.compareTo(o.dateTIme);
}
}
```

4. Screenshots