UNIVERSITI TEKNOLOGI MARA | Cawangan Melaka Kampus Jasin

UNIVERSITI TEKNOLOGI MARA (UiTM)

CAWANGAN MELAKA KAMPUS JASIN

DIPLOMA IN COMPUTER SCIENCE (CDCS110)

PROGRAMMING PARADIGM (CSC305)

GROUP ASSIGNMENT

M3CS1104A

PREPARED BY:

NAME	STUDENT ID
MUHAMMAD HARIZ BIN MOHD ZAKI	2022486466
AHMAD AIMAN ARIF BIN AHMAD ALIAS	2022881236
MUHAMMAD IRFAN SYAFIQ BIN ZAIDI	2022488964
MUHAMMAD SYAHIR RIFIQI BIN NIZAM	2022459769

PREPARED FOR:

WANNORAINI BINTI WAN ABDUL LATIF

SUBMISSION DATE:

JULY 12TH 2024

Table Of Contents

1.0 PROJECT BACKGROUND

Sports Equipment Reservation System

Sports and recreation facility is a popular destination for students in university looking to stay active and healthy. However, managing the rental of sports equipment, such as basketballs, soccer balls, and tennis rackets, can be a tedious and time-consuming task for the staff. To improve efficiency and students' satisfaction, the facility will be implementing a system for managing the rental of sports equipment. This would allow students to easily reserve equipment in advance and return it at their convenience, while also providing the staff with a clear overview of rental activity and inventory management.

Our project, named Sports Equipment Reservation System, is a combination of C language and Java programming intended to allow students to reserve sports equipment at either a recreational center or sports facilities. The program is designed to make the process of reserving equipment more efficient and convenient for both staff and students.

Staff members can handle the program to let the students browse a list of available equipment, select the items they want to reserve, and confirm their reservation. The program also keeps track of the available inventory, so students are only able to reserve equipment that is currently available. The program is intended to be used by staff members at the facility to manage reservations and keep track of inventory.

The project is used to streamline the sports equipment reservation process and reduce the burden on staff members who need to keep track of the reservation and inventory manually. By having this task automated, the C language and Java programming will save time and improve the efficiency of the facility. It also may be able to improve the students and staff experience overall, who now have the ability to easily see and make reservations rather than having to wait in line to reserve particular sports equipment.

2.0 PROBLEM TO BE SOLVED

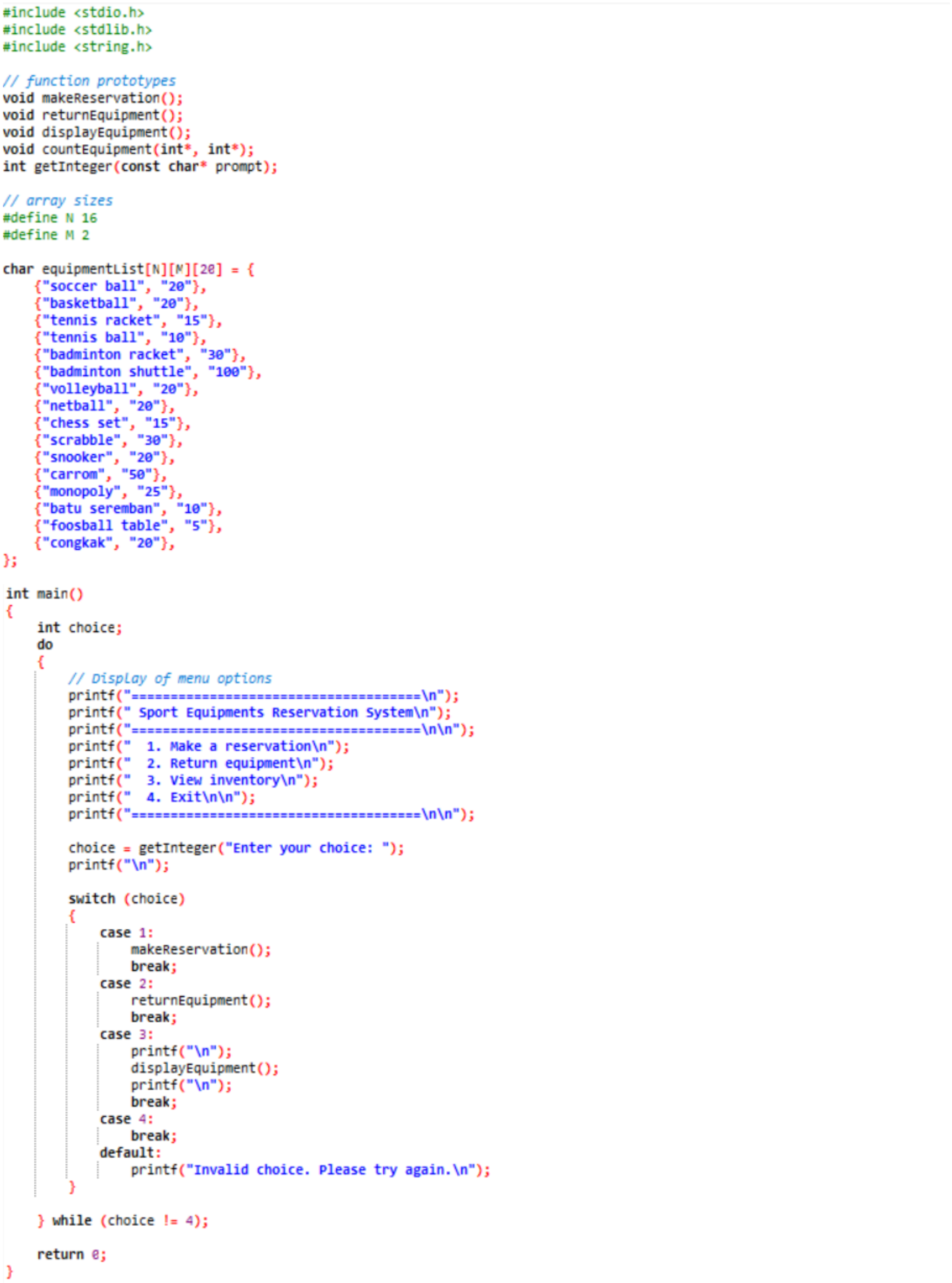
Currently, the facility relies on a manual system where students must visit the front desk to borrow equipment and return it at the end of their visit, and the staff needs to manually check the inventory and keep track of the reservations. This system is inefficient, as it can lead to long lines and confusion over which equipment is available. Additionally, staff may struggle to keep track of which equipment has been rented out and when it is due to be returned, leading to potential errors and lost revenue. The sports equipment store is struggling with inefficient inventory management, leading to stockouts, overstock situations, and difficulties in tracking product movements. Critical issues to be addressed include:

3.0 OBJECTIVE

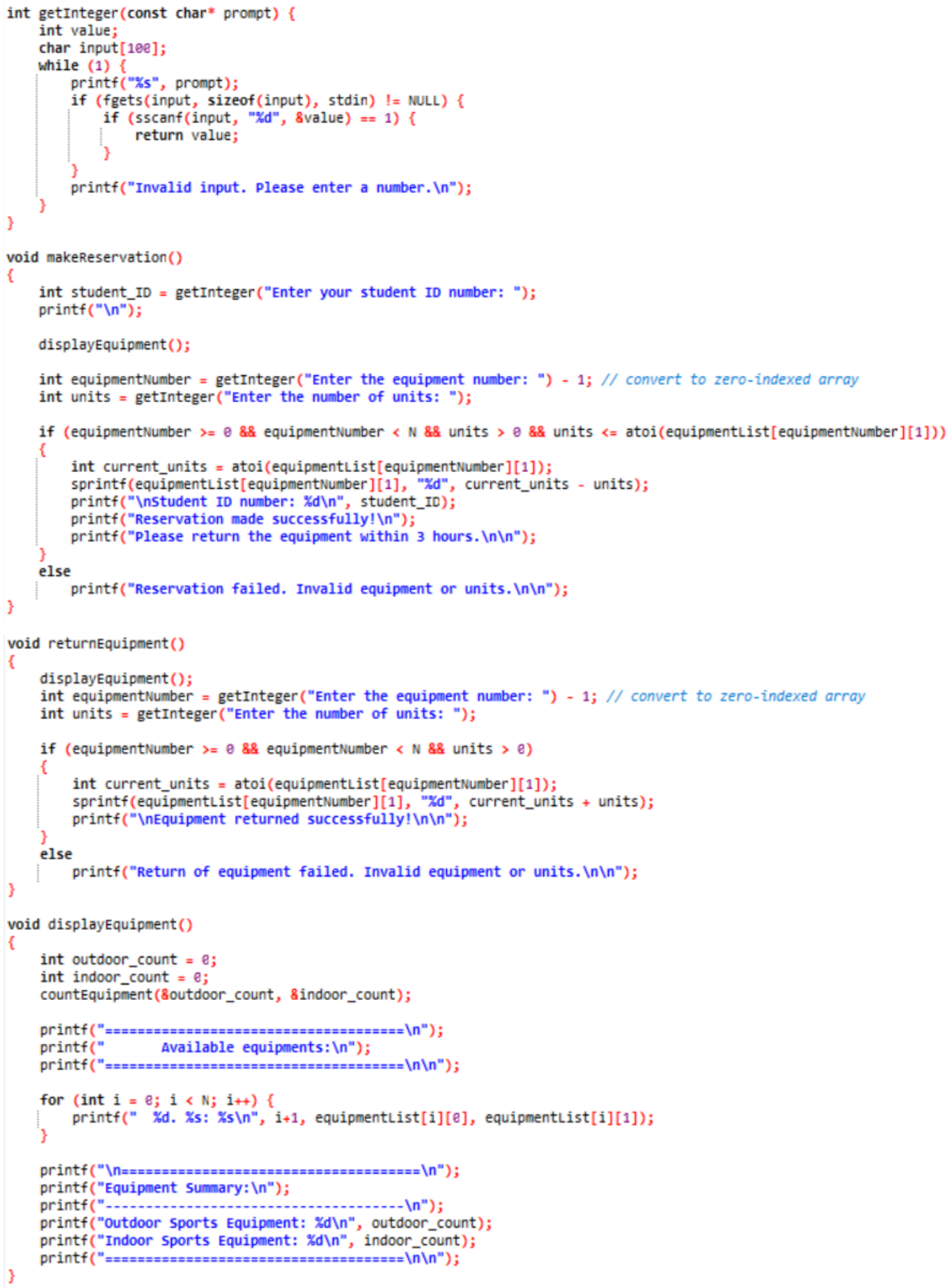
1. To streamline the process of reserving sports equipment.
2. To reduce the burden on staff members by automating the task of managing reservation and tracking inventory.
3. To improve the efficiency of the facility by saving time and reducing the need for manual tasks.
4. To track the available inventory of sports equipment and ensure that users are only able to reserve equipment that is currently available.
5. To improve the user experience for staff by providing a convenient and easy-to-use system for reserving equipment.
6. To avoid both reservation and inventory management mistakes that can be caused by human errors, especially when dealing with large numbers of inventory and reservations.

4.0 Language Of The Project : C and Java

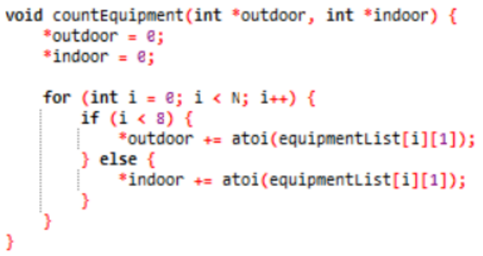
C Language



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// function prototypes
void makeReservation();
void returnEquipment();
void displayEquipment();
void countEquipment(int*, int*);
int getInteger(const char* prompt);

// array sizes
#define N 16
#define M 2

char equipmentList[N][M][20] = {
    {"soccer ball", "20"},
    {"basketball", "20"},
    {"tennis racket", "15"},
    {"tennis ball", "10"},
    {"badminton racket", "30"},
    {"badminton shuttle", "100"},
    {"volleyball", "20"},
    {"netball", "20"},
    {"chess set", "15"},
    {"scrabble", "30"},
    {"snooker", "20"},
    {"carrom", "50"},
    {"monopoly", "25"},
    {"batu seremban", "10"},
    {"foosball table", "5"},
    {"congkak", "20"},
};

int main()
{
    int choice;
    do
    {
        // Display of menu options
        printf("=====================================\n");
        printf(" Sport Equipments Reservation System\n");
        printf("=====================================\n\n");
        printf("  1. Make a reservation\n");
        printf("  2. Return equipment\n");
        printf("  3. View inventory\n");
        printf("  4. Exit\n\n");
        printf("=====================================\n\n");

        choice = getInteger("Enter your choice: ");
        printf("\n");

        switch (choice)
        {
            case 1:
                makeReservation();
                break;
            case 2:
                returnEquipment();
                break;
            case 3:
                printf("\n");
                displayEquipment();
                printf("\n");
                break;
            case 4:
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }

    } while (choice != 4);

    return 0;
}
```



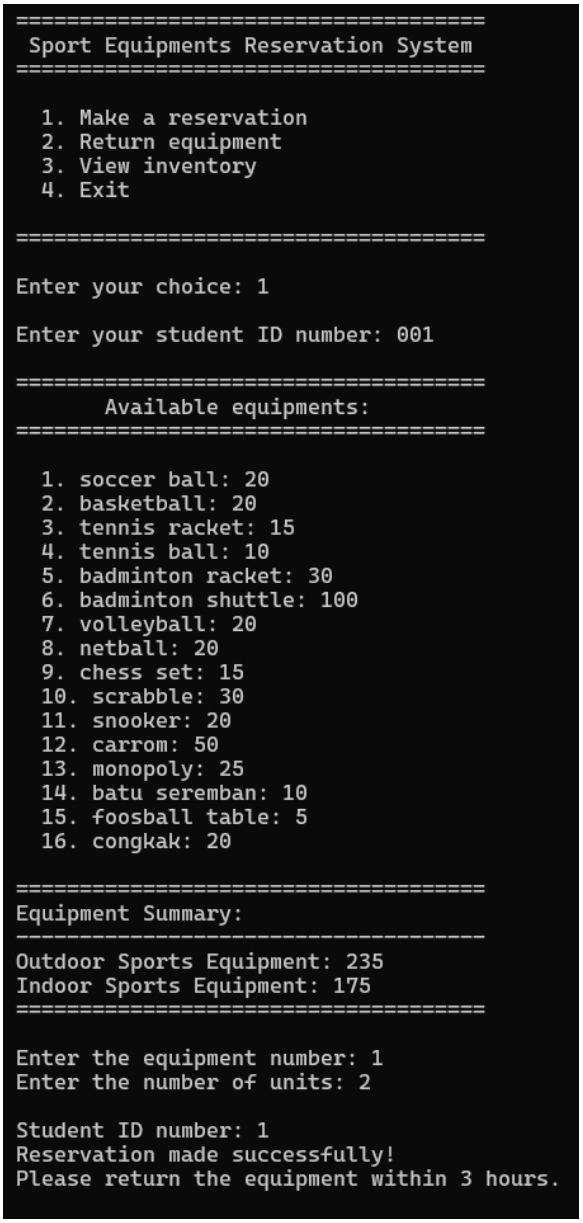
```c
int getInteger(const char* prompt) {
    int value;
    char input[100];
    while (1) {
        printf("%s", prompt);
        if (fgets(input, sizeof(input), stdin) != NULL) {
            if (sscanf(input, "%d", &value) == 1) {
                return value;
            }
        }
        printf("Invalid input. Please enter a number.\n");
    }
}

void makeReservation()
{
    int student_ID = getInteger("Enter your student ID number: ");
    printf("\n");

    displayEquipment();

    int equipmentNumber = getInteger("Enter the equipment number: ") - 1; // convert to zero-indexed array
    int units = getInteger("Enter the number of units: ");

    if (equipmentNumber >= 0 && equipmentNumber < N && units > 0 && units <= atoi(equipmentList[equipmentNumber][1]))
    {
        int current_units = atoi(equipmentList[equipmentNumber][1]);
        sprintf(equipmentList[equipmentNumber][1], "%d", current_units - units);
        printf("\nStudent ID number: %d\n", student_ID);
        printf("Reservation made successfully!\n");
        printf("Please return the equipment within 3 hours.\n\n");
    }
    else
        printf("Reservation failed. Invalid equipment or units.\n\n");
}

void returnEquipment()
{
    displayEquipment();
    int equipmentNumber = getInteger("Enter the equipment number: ") - 1; // convert to zero-indexed array
    int units = getInteger("Enter the number of units: ");

    if (equipmentNumber >= 0 && equipmentNumber < N && units > 0)
    {
        int current_units = atoi(equipmentList[equipmentNumber][1]);
        sprintf(equipmentList[equipmentNumber][1], "%d", current_units + units);
        printf("\nEquipment returned successfully!\n\n");
    }
    else
        printf("Return of equipment failed. Invalid equipment or units.\n\n");
}

void displayEquipment()
{
    int outdoor_count = 0;
    int indoor_count = 0;
    countEquipment(&outdoor_count, &indoor_count);

    printf("======================================\n");
    printf("       Available equipments:\n");
    printf("======================================\n\n");

    for (int i = 0; i < N; i++) {
        printf("  %d. %s: %s\n", i+1, equipmentList[i][0], equipmentList[i][1]);
    }

    printf("\n======================================\n");
    printf("Equipment Summary:\n");
    printf("--------------------------------------\n");
    printf("Outdoor Sports Equipment: %d\n", outdoor_count);
    printf("Indoor Sports Equipment: %d\n", indoor_count);
    printf("======================================\n\n");
}
```



```c
void countEquipment(int *outdoor, int *indoor) {
    *outdoor = 0;
    *indoor = 0;

    for (int i = 0; i < N; i++) {
        if (i < 8) {
            *outdoor += atoi(equipmentList[i][1]);
        } else {
            *indoor += atoi(equipmentList[i][1]);
        }
    }
}
```

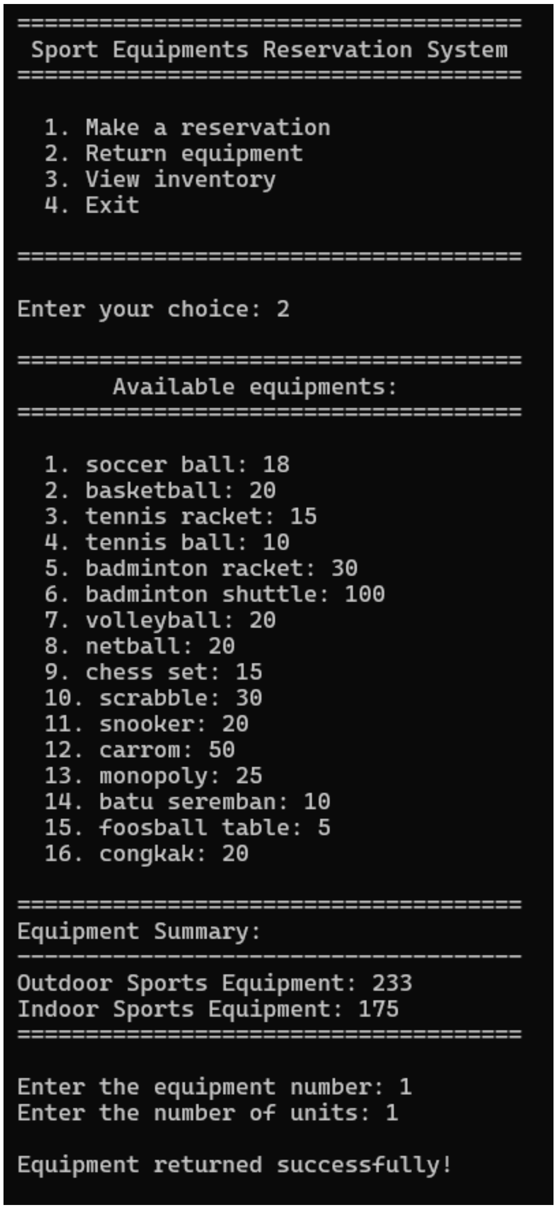
Sample Output

Choice number 1



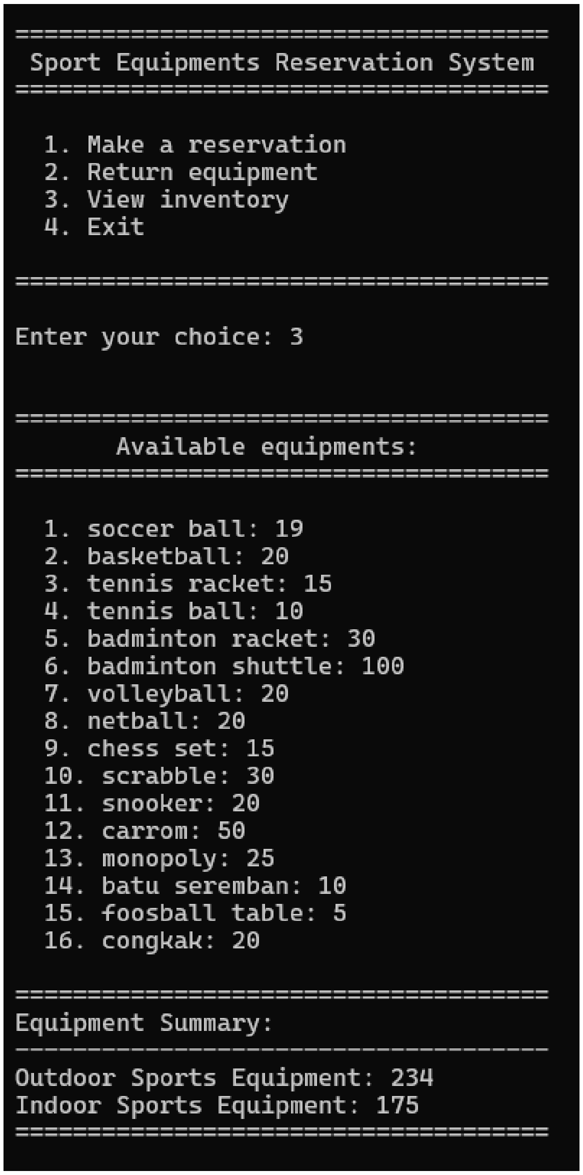
```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: 1

Enter your student ID number: 001

=====================================
       Available equipments:
=====================================

  1. soccer ball: 20
  2. basketball: 20
  3. tennis racket: 15
  4. tennis ball: 10
  5. badminton racket: 30
  6. badminton shuttle: 100
  7. volleyball: 20
  8. netball: 20
  9. chess set: 15
  10. scrabble: 30
  11. snooker: 20
  12. carrom: 50
  13. monopoly: 25
  14. batu seremban: 10
  15. foosball table: 5
  16. congkak: 20

=====================================
Equipment Summary:
-------------------------------------
Outdoor Sports Equipment: 235
Indoor Sports Equipment: 175
=====================================

Enter the equipment number: 1
Enter the number of units: 2

Student ID number: 1
Reservation made successfully!
Please return the equipment within 3 hours.
```

Choice number 2



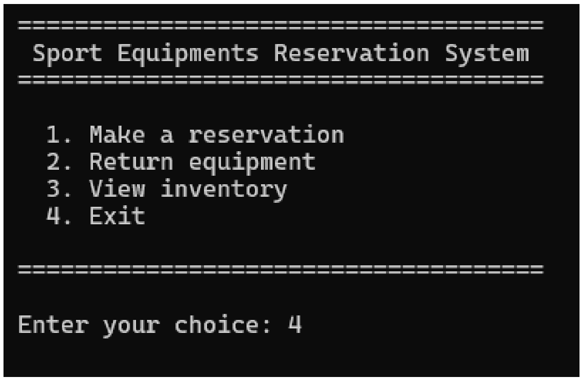
```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: 2

=====================================
       Available equipments:
=====================================

  1. soccer ball: 18
  2. basketball: 20
  3. tennis racket: 15
  4. tennis ball: 10
  5. badminton racket: 30
  6. badminton shuttle: 100
  7. volleyball: 20
  8. netball: 20
  9. chess set: 15
  10. scrabble: 30
  11. snooker: 20
  12. carrom: 50
  13. monopoly: 25
  14. batu seremban: 10
  15. foosball table: 5
  16. congkak: 20

=====================================
Equipment Summary:
-------------------------------------
Outdoor Sports Equipment: 233
Indoor Sports Equipment: 175
=====================================

Enter the equipment number: 1
Enter the number of units: 1

Equipment returned successfully!
```

Choice number 3



```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: 3


=====================================
       Available equipments:
=====================================

  1. soccer ball: 19
  2. basketball: 20
  3. tennis racket: 15
  4. tennis ball: 10
  5. badminton racket: 30
  6. badminton shuttle: 100
  7. volleyball: 20
  8. netball: 20
  9. chess set: 15
  10. scrabble: 30
  11. snooker: 20
  12. carrom: 50
  13. monopoly: 25
  14. batu seremban: 10
  15. foosball table: 5
  16. congkak: 20

=====================================
Equipment Summary:
-------------------------------------
Outdoor Sports Equipment: 234
Indoor Sports Equipment: 175
=====================================
```

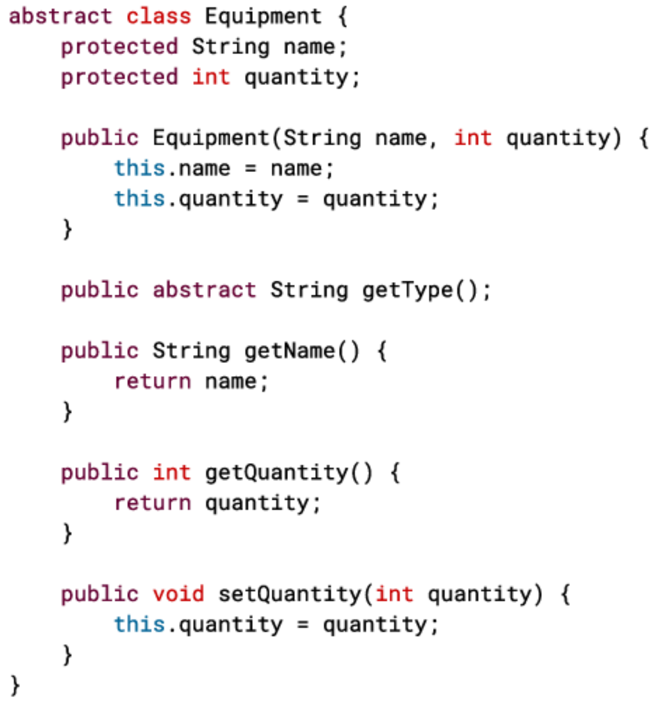
Choice number 4



```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: 4
```

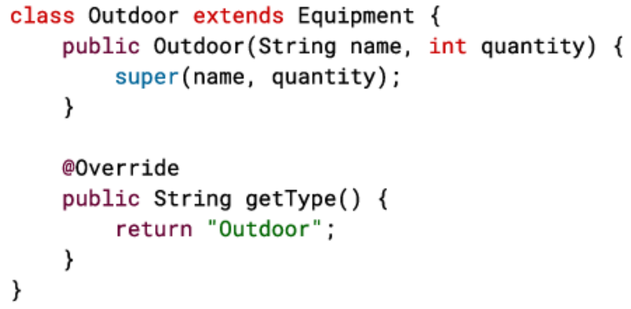
Java Language

Equipment Class



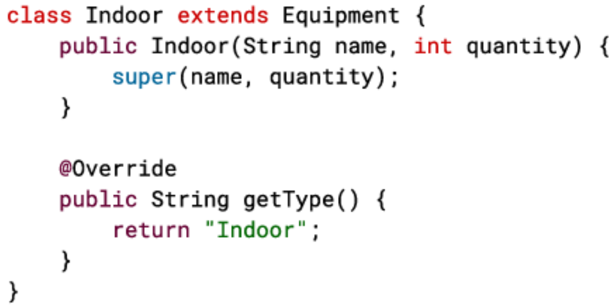
```java
abstract class Equipment {
    protected String name;
    protected int quantity;

    public Equipment(String name, int quantity) {
        this.name = name;
        this.quantity = quantity;
    }

    public abstract String getType();

    public String getName() {
        return name;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
}
```

Outdoor Class



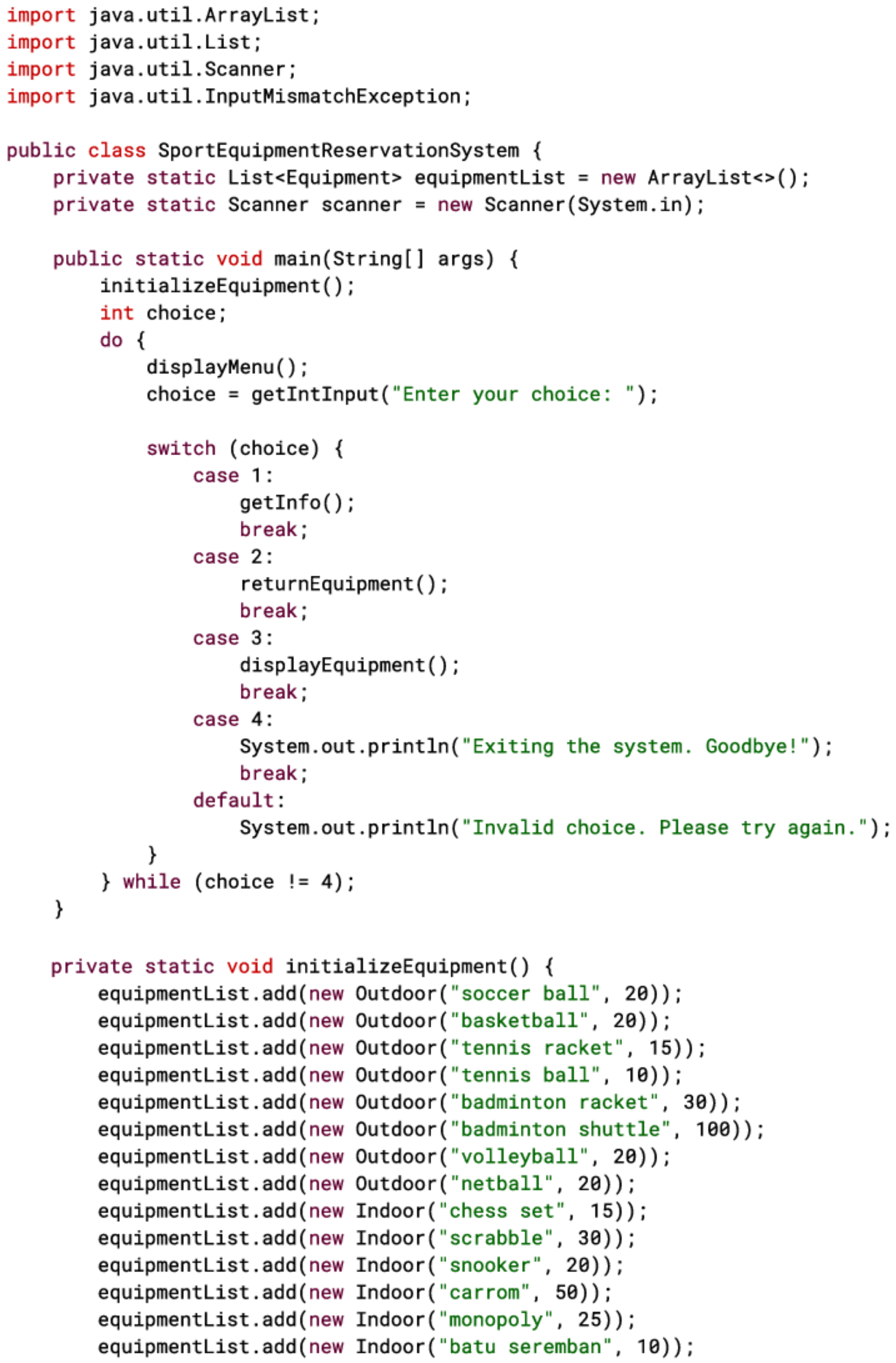
```java
class Outdoor extends Equipment {
    public Outdoor(String name, int quantity) {
        super(name, quantity);
    }

    @Override
    public String getType() {
        return "Outdoor";
    }
}
```

Indoor Class



```java
class Indoor extends Equipment {
    public Indoor(String name, int quantity) {
        super(name, quantity);
    }

    @Override
    public String getType() {
        return "Indoor";
    }
}
```

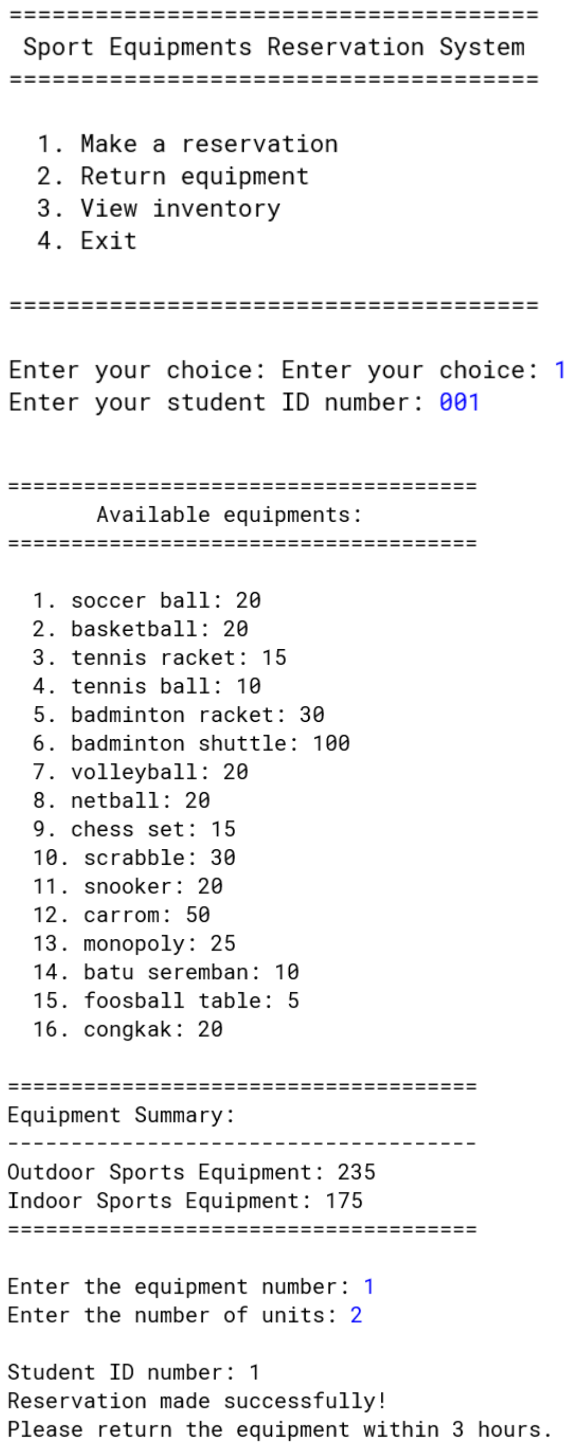
SportsEquipmentReservationSystem Class



```java
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.InputMismatchException;

public class SportEquipmentReservationSystem {
    private static List<Equipment> equipmentList = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        initializeEquipment();
        int choice;
        do {
            displayMenu();
            choice = getIntInput("Enter your choice: ");

            switch (choice) {
                case 1:
                    getInfo();
                    break;
                case 2:
                    returnEquipment();
                    break;
                case 3:
                    displayEquipment();
                    break;
                case 4:
                    System.out.println("Exiting the system. Goodbye!");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 4);
    }

    private static void initializeEquipment() {
        equipmentList.add(new Outdoor("soccer ball", 20));
        equipmentList.add(new Outdoor("basketball", 20));
        equipmentList.add(new Outdoor("tennis racket", 15));
        equipmentList.add(new Outdoor("tennis ball", 10));
        equipmentList.add(new Outdoor("badminton racket", 30));
        equipmentList.add(new Outdoor("badminton shuttle", 100));
        equipmentList.add(new Outdoor("volleyball", 20));
        equipmentList.add(new Outdoor("netball", 20));
        equipmentList.add(new Indoor("chess set", 15));
        equipmentList.add(new Indoor("scrabble", 30));
        equipmentList.add(new Indoor("snooker", 20));
        equipmentList.add(new Indoor("carrom", 50));
        equipmentList.add(new Indoor("monopoly", 25));
        equipmentList.add(new Indoor("batu seremban", 10));
        equipmentList.add(new Indoor("foosball table", 5));
        equipmentList.add(new Indoor("congkak", 20));
    }

    private static void displayMenu() {
        System.out.println("=====================================");
        System.out.println(" Sport Equipments Reservation System");
        System.out.println("=====================================\n");
        System.out.println("  1. Make a reservation");
        System.out.println("  2. Return equipment");
        System.out.println("  3. View inventory");
        System.out.println("  4. Exit\n");
        System.out.println("=====================================\n");
        System.out.print("Enter your choice: ");
    }
    private static void getInfo() {
        int studentID = getIntInput("Enter your student ID number: ");
        System.out.println();
        makeReservation(studentID);
    }

    private static void makeReservation(int studentID) {
        displayEquipment();
        int equipmentNumber = getIntInput("Enter the equipment number: ") - 1; //
convert to zero-indexed
        int units = getIntInput("Enter the number of units: ");

        if (equipmentNumber >= 0 && equipmentNumber < equipmentList.size() && units >
0 && units <= equipmentList.get(equipmentNumber).getQuantity()) {
            Equipment equipment = equipmentList.get(equipmentNumber);
            equipment.setQuantity(equipment.getQuantity() - units);
            System.out.println("\nStudent ID number: " + studentID);
            System.out.println("Reservation made successfully!");
            System.out.println("Please return the equipment within 3 hours.\n");
        } else {
            System.out.println("Reservation failed. Invalid equipment or units.\n");
        }
    }

    private static void returnEquipment() {
        displayEquipment();
        int equipmentNumber = getIntInput("Enter the equipment number: ") - 1; //
convert to zero-indexed
        int units = getIntInput("Enter the number of units: ");

        if (equipmentNumber >= 0 && equipmentNumber < equipmentList.size() && units >
0) {
            Equipment equipment = equipmentList.get(equipmentNumber);
            equipment.setQuantity(equipment.getQuantity() + units);
            System.out.println("\nEquipment returned successfully!\n");
        } else {
            System.out.println("Return of equipment failed. Invalid equipment or
units.\n");
        }
    }

    private static void displayEquipment() {
        System.out.println("=====================================");
        System.out.println("       Available equipments:");
        System.out.println("=====================================\n");

        for (int i = 0; i < equipmentList.size(); i++) {
            Equipment equipment = equipmentList.get(i);
            System.out.printf("  %d. %s: %d\n", i + 1, equipment.getName(),
equipment.getQuantity());
        }

        int[] counts = countEquipment();
        System.out.println("\n=====================================");
        System.out.println("Equipment Summary:");
        System.out.println("-------------------------------------");
        System.out.println("Outdoor Sports Equipment: " + counts[0]);
        System.out.println("Indoor Sports Equipment: " + counts[1]);
        System.out.println("=====================================\n");
    }

    private static int[] countEquipment() {
        int outdoorCount = 0;
        int indoorCount = 0;

        for (Equipment equipment : equipmentList) {
            if (equipment instanceof Outdoor) {
                outdoorCount += equipment.getQuantity();
            } else if (equipment instanceof Indoor) {
                indoorCount += equipment.getQuantity();
            }
        }

        return new int[]{outdoorCount, indoorCount};
    }

    private static int getIntInput(String prompt) {
        while (true) {
            try {
                System.out.print(prompt);
                return scanner.nextInt();
            } catch (InputMismatchException e) {
                System.out.println("Invalid input. Please enter a number.");
                scanner.nextLine(); // Clear the invalid input
            }
        }

    }
}
```

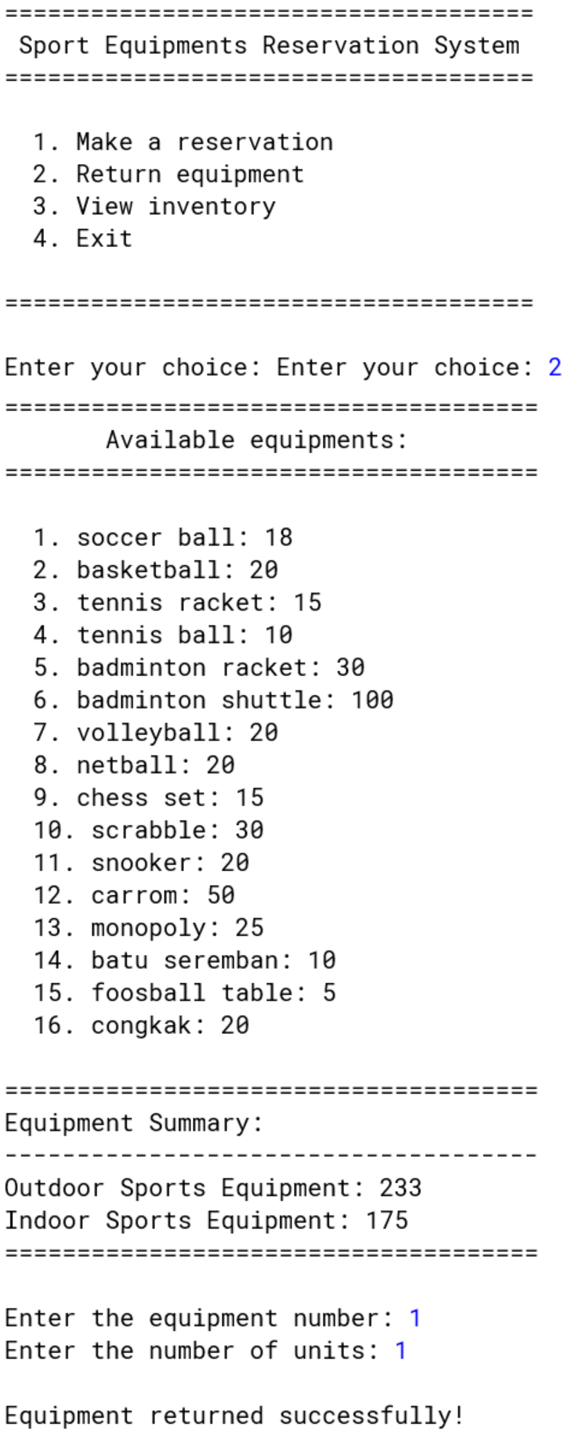
Sample Output

Choice number 1



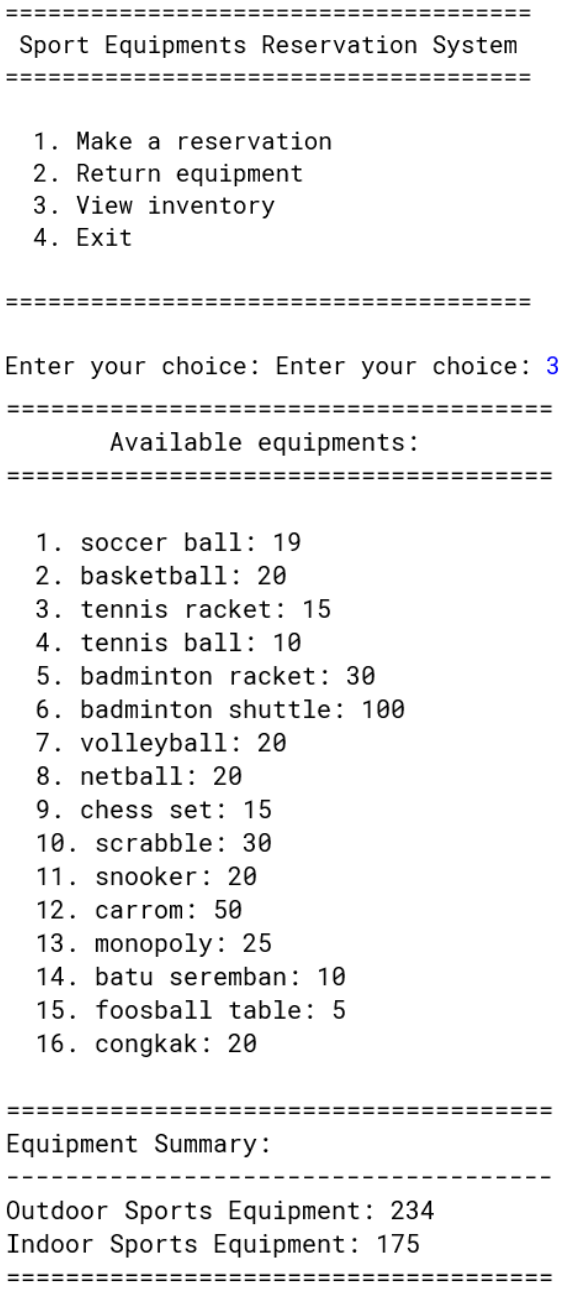
```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: Enter your choice: 1
Enter your student ID number: 001


=====================================
       Available equipments:
=====================================

  1. soccer ball: 20
  2. basketball: 20
  3. tennis racket: 15
  4. tennis ball: 10
  5. badminton racket: 30
  6. badminton shuttle: 100
  7. volleyball: 20
  8. netball: 20
  9. chess set: 15
  10. scrabble: 30
  11. snooker: 20
  12. carrom: 50
  13. monopoly: 25
  14. batu seremban: 10
  15. foosball table: 5
  16. congkak: 20

=====================================
Equipment Summary:
-------------------------------------
Outdoor Sports Equipment: 235
Indoor Sports Equipment: 175
=====================================

Enter the equipment number: 1
Enter the number of units: 2

Student ID number: 1
Reservation made successfully!
Please return the equipment within 3 hours.
```

Choice number 2



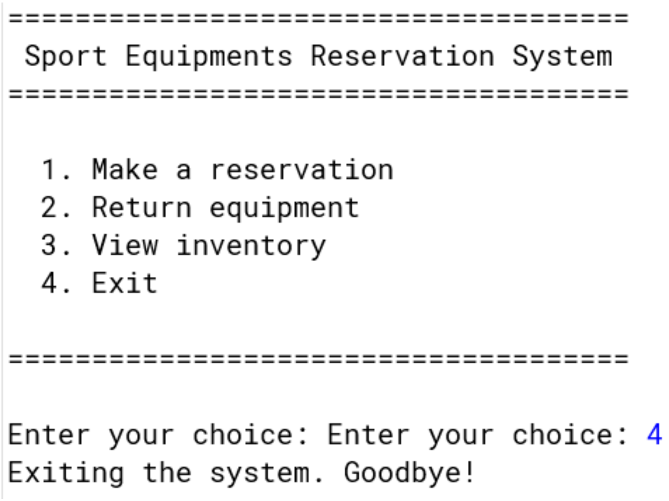
```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: Enter your choice: 2
=====================================
       Available equipments:
=====================================

  1. soccer ball: 18
  2. basketball: 20
  3. tennis racket: 15
  4. tennis ball: 10
  5. badminton racket: 30
  6. badminton shuttle: 100
  7. volleyball: 20
  8. netball: 20
  9. chess set: 15
  10. scrabble: 30
  11. snooker: 20
  12. carrom: 50
  13. monopoly: 25
  14. batu seremban: 10
  15. foosball table: 5
  16. congkak: 20

=====================================
Equipment Summary:
-------------------------------------
Outdoor Sports Equipment: 233
Indoor Sports Equipment: 175
=====================================

Enter the equipment number: 1
Enter the number of units: 1

Equipment returned successfully!
```

Choice number 3



```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: Enter your choice: 3
=====================================
       Available equipments:
=====================================

  1. soccer ball: 19
  2. basketball: 20
  3. tennis racket: 15
  4. tennis ball: 10
  5. badminton racket: 30
  6. badminton shuttle: 100
  7. volleyball: 20
  8. netball: 20
  9. chess set: 15
  10. scrabble: 30
  11. snooker: 20
  12. carrom: 50
  13. monopoly: 25
  14. batu seremban: 10
  15. foosball table: 5
  16. congkak: 20

=====================================
Equipment Summary:
-------------------------------------
Outdoor Sports Equipment: 234
Indoor Sports Equipment: 175
=====================================
```

Choice number 4



```
=====================================
 Sport Equipments Reservation System
=====================================

  1. Make a reservation
  2. Return equipment
  3. View inventory
  4. Exit

=====================================

Enter your choice: Enter your choice: 4
Exiting the system. Goodbye!
```

5.0 Assessment

CSC305 Group Assessment's Rubric

Project Title:		#	Group Members Name	UiTM Number
		1		
Group:		2		
Submission Date:		3		
Presentation Date:		4		

A) Imperative Programming Using C

#	Score Criteria			None ***0 point***	2992901 Poor 1 Poor ***1 point***	2992902 Fair 2 Fair ***2 points***	2992903 Good 3 Good ***3 points***	Weightage	Subtotal ***Score x w***
1	**Array**	a.	**Declaration, Definition, Manipulation**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*minor error/mistake* 2992906 Program w	*correct*	*2*	
		e.	**Array**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*4*	

2	**Subprogram**	a.	**With return value**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*2*	
		b.	**Without return value**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*2*	
		c.	**Function call**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*2*	
3	**Basic Input/ Output**	a.	**Add Data**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*7*	
		~~b.~~	~~**Search & Update Data**~~	~~*Not exist in the product*~~	~~*exists but incorrect*~~ 2992905 Program wo	~~*exists with minor error/mistake*~~ 2992906 Program w	~~*exists and works well in all area*~~	~~*3*~~	
		~~c.~~	~~**Delete Data**~~	~~*Not exist in the product*~~	~~*exists but incorrect*~~ 2992905 Program wo	~~*exists with minor error/mistake*~~ 2992906 Program w	~~*exists and works well in all area*~~	~~*2*~~	

4	**Program**	**User friendliness**	Not exist in the product	Flow between process is **unaccaptable**	Flow between process is **accaptable**	Flow between process **well organized**	3	
		Good Programming Style	Not exist in the product	Program code is **not organized** according to the G.P.Style 2992905 Program wo	Program code is **neat** however no remarks (comments) written in the codes. 2992906 Program w	Program code is **well organized**, with **remarks (comments)** according to the G.P.Style	3	
Total Score for C Program **:** **(Maximum of 75 points)**								

B) Object-Oriented Programming Using JAVA

#	Score Criteria			*None* ***0 point***	2992901 Poor 1 *Poor* ***1 point***	2992902 Fair 2 *Fair* ***2 points***	2992903 Good 3 *Good* ***3 points***	*Weightage*	*Subtotal* ***Score x w***
1	**Inheritance**	a.	**Superclass**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
		b.	**Subclass**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
2	**Encapsulation**	a.	**All data members (fields) of a class are declared private**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
		b.	**Accessibility of data member, which are wrapped in the class**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
3		a.	**abstract class**	*Not exist in*	*exists but incorrect*	*exists with minor*	*exists and works*	*3*	

	Abstraction			*the product*	*2992905* Program wo	*error/mistake* 2992906 Program w	*well in all area*		
		b.	**abstract method definition**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
		c.	**action if the abstraction method being called**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
4	**Polymorphism**	**a.**	**Overloading**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
		b.	**Overriding**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*3*	
		b.	**Without return value**	*Not exist in the product*	*exists but incorrect* 2992905 Program wo	*exists with minor error/mistake* 2992906 Program w	*exists and works well in all area*	*2*	
		d.	**Function call**	*Not exist in*	*exists but incorrect*	*exists with minor*	*exists and works*	*2*	

				the product	2992905 Program wo	error/mistake 2992906 Program w	well in all area		
6	**Basic Input/ Output**	**a.**	**Add Data**	Not exist in the product	exists but incorrect 2992905 Program wo	exists with minor error/mistake 2992906 Program w	exists and works well in all area	7	
		b.	~~**Search & Update Data**~~	~~Not exist in the product~~	~~exists but incorrect~~ 2992905 Program wo	~~exists with minor error/mistake~~ 2992906 Program w	~~exists and works well in all area~~	~~3~~	
		c.	~~**Delete Data**~~	~~Not exist in the product~~	~~exists but incorrect~~ 2992905 Program wo	~~exists with minor error/mistake~~ 2992906 Program w	~~exists and works well in all area~~	~~2~~	
7	**Program**	**User friendliness**		Not exist in the product	Flow between process is **unaccaptable**	Flow between process is **accaptable**	Flow between process **well organized**	3	
		Good Programming Style		Not exist in the product	Program code is **not organized** according to the G.P.Style 2992905 Program wo	Program code is **neat** however no remarks (comments) written in the codes.	Program code is **well organized**, with **remarks (comments)** according to the G.P.Style	3	

					2992906 Program w			
Total Score for JAVA Program : ***(Maximum of 132 points)***								

C) **Other Criteria**

#	Score Criteria	*None* ***0 point***	2992901 Poor 1 *Poor* ***1 point***	2992902 Fair 2 *Fair* ***2 points***	2992903 Good 3 *Good* ***3 points***	*Weightage*	*Subtotal* ***Score x w***
1	**Critical Thinking on Problem Solving**	*Entire problem/ case study **not been solved.***	*Less of **50%** of the problem/ case study **solved** and not all criteria fulfilled.*	***70%** of the problem/ case study **solved** and not all the criteria fulfilled.*	*The problem/ case study **solved** and all the criteria fulfilled.*	*3*	
2	**Presentation**	*Does not present*	*Presented however unable to provide satisfying answer to questions asked*	*Presented and satisfying answer all question asked*	*Well-presented and able to answer well all question asked*	*3*	
3	**Documentation** a. **Project Proposal**	*No documentation presented*	*Not organized, **less than 50%***	*70% of items supplied*	*All items supplied Well documented*	*2*	

					items supplied				
		b.	**Final Report**	*No documentation presented*	*Not organized,* ***less than 50%*** *items supplied*	*70% of items supplied*	*All items supplied Well documented*	*3*	
							Total Score for Other Criteria **:** ***(Maximum of 33 points)***		

OVERALL SCORE

	C *(max:75 points)*	***JAVA*** *(max:132 points)*	***Other Criteria*** *(max:33 points)*	***OVERALL SCORE:*** *(maximum of 240 points)*	*Percentage: (10%)*
SUB-SCORE					