



ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION

CT018-3-1-ICP

INTRODUCTION TO C PROGRAMMING

INDIVIDUAL ASSIGNMENT

STUDENT NAME: CHAN JIA LE (TP049952)

LECTURER: HAIRUL SIVAGURU A/L SUBARMANIYAN

INTAKE CODE: UC1F1809CS(DA)

Contents

1.0 Introduction and Assumption.....	6
1.1 Introduction	6
1.2 Assumption.....	6
2.0 Program Design.....	7
2.1 Pseudocode.....	7
2.2 Flow Chart.....	25
3.0 C Programming Concept.....	41
3.1 Header File	41
3.2 Data Structures	41
3.3 Function.....	42
3.4 Displaying Text	43
3.5 Reading Value	43
3.6 Assigning Value to Variable	43
3.7 Comment	43
3.8 If, Else Statement	44
3.9 Switch, Case	44
3.10 While Loop.....	44
3.11 For Loop	45
3.12 Open / Close File	45
3.13 Declaration Variable.....	45
3.14 Read File.....	46
3.15 Write File.....	46
3.16 Append File	46
3.18 Strcmp Function	47
3.19 Go to	47
3.20 Array.....	47

3.21 System (“CLS”).....	48
3.22 Sleep Function	48
3.23 Strupr Function.....	48
3.24 Remove / Rename File	48
3.25 Fscanf Function	48
4.0 Additional Features	49
4.1 Admin Login	49
4.2 Choosing Room	50
4.3 Change Student Personal Details.....	51
4.4 Show Percentage of Male and Female Hosteler.....	53
5.0 Test Specification Table.....	54
6.0 Sample Output:	58
6.1 Hostel Management System Sample Output:.....	58
6.2 Student Register Sample Output	58
6.2.1 Record Was Found Sample Output	59
6.3 Booking Menu.....	60
6.3.1 Male Booking Menu	60
6.3.2 Female Booking Menu.....	60
6.4 Booking Room	61
6.4.1 Male Block A1 Booking.....	61
6.4.2 Male Block A2 Booking.....	62
6.4.3 Female Block B1 Booking.....	63
6.4.4 Female Block B3 Booking.....	64
6.4.5 Unavailable Block Bookings	65
6.5 Choosing Room Number.....	66
6.5.1 Room Number Chosen is Occupied.....	66
6.5.2 Room Number Chosen is Available	66

6.5.3 Choosing Beds in a room.....	67
6.5.4 Bed Chosen is Occupied.....	67
6.6 Service Subscription Menu	68
6.6.1 Laundry Subscription Menu	68
6.6.2 Gym Subscription Menu.....	68
6.6.3 Week of Stay Given & Successful Booking.....	69
6.7 Student Login	70
6.7.1 No Record of Registration Found.....	71
6.7.2 Record Found & Successful Login.....	71
6.8 Student Login Menu.....	72
6.9 Change/ Cancel Room & Subscribe/ Unsubscribe Laundry or Gym Service.....	72
6.9.1 Change Room	73
6.9.2 Cancel Room.....	75
6.9.3 Change Subscription Service	76
6.10 Check / Change Personal Details	77
6.11 Personal Details Menu.....	78
6.11.1 Change Personal Details	79
6.12 Payment Facility.....	81
6.12.1 Payment Amount More Than Amount Due.....	81
6.12.2 Payment Successfully Made	82
6.13 Exit from Student Login System	82
6.14 Admin Login	83
6.14.1 Admin Successfully Login.....	84
6.15 Admin System	84
6.15.1 Check Overall Details.....	85
6.15.2 Search for Student Details	86
6.15.3 Check Block Room Availability.....	87

6.15.4 Exit from Admin System	88
6.16 Exit Program.....	88
7.0 Conclusion	89
8.0 References.....	90

1.0 | Introduction and Assumption

1.1 | Introduction

Wisdom College is now providing accommodation for its students. There is a total of 8 blocks, 4 blocks for each gender. The Name of the blocks are A1, A2, A3, A4, B1, B2, B3, B4. Block A is for male students and Block B is for female students. Each block contains 100 rooms and the bed quantity is base on the number behind their block. So, Block A1 will be single bed room, block A2 will be double sided bed. Currently only Block A1, A2, B1, B3 is available as the rest are still under construction.

The cost of accommodation is calculated in a weekly basis. The rental for Block A1, B1 will be RM400 per week. Block A2 will be RM200 and Block B3 will be RM150. Students are welcomed to subscribe to additional service as well, the service includes gym service and laundry service, which cost RM10 and RM20 respectively. Students can now use the Hostel Management System to book a hostel room.

1.2 | Assumption

The Hostel Management System is built for students to have an easy way to make a booking for hostel as well as designed to allows staff to get easy access over the information. Students are required to register to make a booking. As for those who already has a booking, they should proceed to login into the system.

For new students that is interested to book a room, they will be required to register. The program will then ask the student for their personal details and identify which block they should stay based on the gender they have chosen; the student will then be able to choose single bed room which is block A1 or double side room block A2 as well as choosing their room. The program will display the available rooms allowing for students to choose the room they desire.

After registration process was done, the program will then ask the user to login using their student ID. Students who login will be able to change, cancel room, check or change personal details, make payment and so on. If students want to change or cancel room, they are required to pay a fine of RM200 or RM300 as a compensation for early cancelation.

Admin menu will be able to be accessed by staffs who know the password only, admin menu allows for easy view of overall total payment receivable and received, list all the student details, search for student and check block available rooms.

2.0 | Program Design

2.1 | Pseudocode

PROGRAM HotelManagementSystem

Create data structure student

```
    string studentID
    string name
    string gender
    integer age
    string phoneNumber
    string blockArea
    integer room number
    integer laundry
    integer gym
    integer weekStay
    integer amountDue
    integer paidAmount
    integer bedNumber
```

Declare member variable apply and check for student structure

Declare array block_a1[100]

Declare array block_a2[100][2]

Declare array block_b1[100]

Declare array block_b3[100][3]

BEGIN

counter = 1

DOWHILE (counter = 1)

```
        call initialize_empty_room()
        call reassign_previous_booking()
```

Print "WISDOM COLLEGE HOSTEL MANAGEMENT SYSTEM"

Print "1) STUDENT REGISTER 2) STUDENT LOGIN"

Print "3) ADMIN LOGIN 4)EXIT"

Print "PLEASE ENTER A CHOICE"

Read choice

IF (choice = 1) THEN

```
            gender = call student_registration()
```

IF (gender = M) THEN

```
                call male_booking_menu()
```

ELSE

```
                call female_booking_menu()
```

ENDIF

ELSE

IF (choice = 2) THEN

```
                call student_login()
```

ENDIF

ELSE

IF (choice = 3) THEN

```
                call admin_login()
```

ENDIF

ELSE

IF (choice = 4) THEN

```
                Print "EXIT PROGRAM "
```

```
                counter = 0
```

ENDIF

ELSE

```

        Print "PLEASE ENTER VALID INPUT"
        continue
    ENDIF
ENDDO
END

FUNCTION initialize_empty_room()
    LOOP count FROM 0 to 99 STEP 1
        block_a1 [count] = 0
        block_a2 [count][0] = 0
        block_a2 [count][1] = 0
        block_b1 [count] = 0
        block_b3 [count][0] = 0
        block_b3 [count][1] = 0
        block_b3 [count][2] = 0
    ENDLOOP
ENDFUNCTION

FUNCTION reassign_previous_booking ()
    OPEN File "Student.txt" for Read
    DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)
        IF (check.blockArea = "A1") THEN
            block_a1[check.roomNumber] = 1
        ELSE
            IF (check.blockArea = "A2") THEN
                block_a2[check.roomNumber][check.bedNumber] = 1
            ENDIF
        ELSE
            IF (check.blockArea = "B1") THEN
                block_b1[check.roomNumber] = 1
            ENDIF
        ELSE
            IF (check.blockArea = "B3") THEN
                block_b3[check.roomNumber][check.bedNumber] = 1
            ENDIF
        ENDIF
    END
    CLOSE File "Student.txt"
END FUNCTION

FUNCTION student_registration()
    OPEN File "Student.txt" for Read
    Print "STUDENT REGISTRATION SYSTEM"
    Print "PLEASE ENTER YOUR STUDENT ID : "
    Read apply.studentID
    DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)
        IF (apply.studentID = check.studentID) THEN
            Print "YOU HAVE REGISTERED, PLEASE LOGIN !"
            RETURN
        ENDIF
    END
    CLOSE File "Student.txt"
    Print "PLEASE ENTER YOUR NAME"
    Read apply.name

```

```

Print "PLEASE ENTER YOUR GENDER (M / F)"
Read apply.gender
Print "PLEASE ENTER YOUR AGE"
Read apply.age
Print "PLEASE ENTER YOUR PHONE NUMBER"
Read apply.phoneNumber
RETURN apply.gender
END FUNCTION

```

```

FUNCTION male_booking_menu()
    OPEN "Student.txt" for append
    a1_available_room = a1_room_available()
    a2_available_room = a2_room_available()
    Print "BOOKING MENU"
    Print "BLOCK A1 AVAILBLE:"
    Print a1_available_room
    Print "BLOCK A2 AVAILABLE"
    Print a2_available_room
    Print "BLOCK A1 = SINGLE ROOM = RM400 WEEKLY"
    Print "BLOCK A2 = DOUBLE ROOM = RM200 WEEKLY"
    Print "PLEASE INSERT A CHOCE ( 1 - 2 )"
    Read block_type
    IF (block_type = 1) THEN
        rental = 400
        apply.blockArea = "A1"
        room_num = call a1room_visual()
        laundry = call laundry_calculation()
        gym = call gym_calculation()
        Print "HOW MANY WEEKS WOULD YOU LIKE TO STAY?"
        Read apply.weekStay
        apply.paidAmount = 0
        total_due = call personal_due(rental, gym, laundry, apply.weekStay,
apply.paidAmount)
            apply.room_num = room_num
            apply.laundry = laundry
            apply.gym = gym
            apply.amountDue = total_due
        Append data structure student all member variable apply details into "Student.txt"
        CLOSE File "Student.txt"
        Print "REGISTER SUCCESSFUL"
        Print "PLEASE LOGIN !"
    ELSE
        rental = 200
        apply.blockArea = "A2"
        room_num = call a2room_visual()
        laundry = call laundry_calculation()
        gym = call gym_calculation()
        Print "HOW MANY WEEKS WOULD YOU LIKE TO STAY?"
        Read apply.weekStay
        apply.paidAmount = 0
        total_due = call personal_due(rental, gym, laundry, apply.weekStay,
apply.paidAmount)
            apply.room_num = room_num
            apply.laundry = laundry
            apply.gym = gym

```

```

        apply.amountDue = total_due
Append data structure student all member variable apply details into "Student.txt"
CLOSE File "Student.txt"
Print "REGISTER SUCCESSFUL"
Print "PLEASE LOGIN !"
ENDIF
END FUNCTION

```

```

FUNCTION female_booking_menu()
OPEN "Student.txt" for append
b1_available_room = b1_room_available()
b3_available_room = b3_room_available()
Print "BOOKING MENU"
Print "BLOCK B1 AVAILABLE:"
Print b1_available_room
Print "BLOCK B3 AVAILABLE"
Print b3_available_room
Print "BLOCK B1 = SINGLE ROOM = RM400 WEEKLY"
Print "BLOCK B3 = TRIPLE ROOM = RM150 WEEKLY"
Print "PLEASE INSERT A CHOCE ( 1 - 3 )"
Read block_type
IF (block_type = 1) THEN
    rental = 400
    apply.blockArea = "B1"
    room_num = call b1room_visual()
    laundry = call laundry_calculation()
    gym = call gym_calculation()
    Print "HOW MANY WEEKS WOULD YOU LIKE TO STAY?"
    Read apply.weekStay
    apply.paidAmount = 0
    total_due = call personal_due(rental, gym, laundry, apply.weekStay,
apply.paidAmount)
        apply.room_num = room_num
        apply.laundry = laundry
        apply.gym = gym
        apply.amountDue
Append data structure student all member variable apply details into "Student.txt"
CLOSE File "Student.txt"
Print "REGISTER SUCCESSFUL"
Print "PLEASE LOGIN !"
ELSE
    rental = 150
    apply.blockArea = "B3"
    room_num = call b3room_visual()
    laundry = call laundry_calculation()
    gym = call gym_calculation()
    Print "HOW MANY WEEKS WOULD YOU LIKE TO STAY?"
    Read apply.weekStay
    apply.paidAmount = 0
    total_due = call personal_due(rental, gym, laundry, apply.weekStay,
apply.paidAmount)
        apply.room_num = room_num
        apply.laundry = laundry
        apply.gym = gym

```

```

        apply.amountDue
Append data structure student all member variable apply details into “Student.txt”
CLOSE File “Student.txt”
Print “REGISTER SUCCESSFUL”
Print “PLEASE LOGIN !”
ENDIF
END FUNCTION

```

```

FUNCTION a1room_visuals()
z = 0
x = 0
Print (“BLOCK A1”)
LOOP count FROM 0 TO 9 STEP 1
    Print “ : ”
    LOOP counter FROM 0 to 9 STEP 1
        Print “A”, z++
        Print block_a1[x++]
    ENDLOOP
    Print “”
ENDLOOP
Print “0 MEANS AVAILABLE      1 MEANS OCCUPIED”
Print “ PLEASE CHOOSE THE ROOM FROM ( 0 – 99 )”
Read room_num
IF (block_a1[room_num] = 0) THEN
    block_a1[room_num] = 1
ELSE
    Print “SORRY THIS ROOM IS OCCUPIED”
ENDIF
RETURN room_num
END FUNCTION

```

```

FUNCTION a2room_visuals()
z = 0
x = 0
Print (“BLOCK A2”)
LOOP count FROM 0 TO 9 STEP 1
    Print “ : ”
    LOOP counter FROM 0 to 9 STEP 1
        Print “A”, z++
        Print block_a2[x][0], block_a2[x][1]
        x++
    ENDLOOP
    Print “”
ENDLOOP
Print “0 MEANS AVAILABLE      1 MEANS OCCUPIED”
Print “ PLEASE CHOOSE THE ROOM FROM ( 0 – 99 )”
Read room_num
IF (block_a2[room_num][0] AND block_a2[room_num][1] = 1) THEN
    Print “SORRY PLEASE CHOOSE ANOTHER ROOM”
ENDIF
Print “PLEASE CHOOSE A BED (L – R)
Read bed_choice
IF (bed_choice = “L” OR bed_choice = “l” ) THEN
    apply.bedNumber = 0
ELSE

```

```

IF (bed_choice = "R" OR bed_choice = "r" ) THEN
    apply.bedNumber = 1
ENDIF
ENDIF
IF (block_a2[room_num][apply.bedNumber] = 1 ) THEN
    Print "SORRY THIS BED IS OCCUPIED"
ELSE
    IF (block_a2[room_num][0] OR block_a2[room_num][1] = 0)
        block_a2[room_num][apply.bedNumber] = 1
    ENDIF
ENDIF
RETURN room_num
END FUNCTION

```

```

FUNCTION b1room_visuals()
z = 0
x = 0
Print ("BLOCK B1")
LOOP count FROM 0 TO 9 STEP 1
    Print " : "
    LOOP counter FROM 0 to 9 STEP 1
        Print "B", z++
        Print block_b1[x++]
    ENDLOOP
    Print ""
ENDLOOP
Print "0 MEANS AVAILABLE      1 MEANS OCCUPIED"
Print " PLEASE CHOOSE THE ROOM FROM ( 0 - 99 )"
Read room_num
IF (block_b1[room_num] = 0) THEN
    block_b1[room_num] = 1
ELSE
    Print "SORRY THIS ROOM IS OCCUPIED"
ENDIF
RETURN room_num
END FUNCTION

```

```

FUNCTION b3room_visuals()
    z = 0
    x = 0
    Print ("BLOCK B3")
    LOOP count FROM 0 TO 9 STEP 1
        Print " : "
        LOOP count FROM 0 to 9 STEP 1
            Print "B", z++
            Print block_b3[x][0], block_b3[x][1],block_b3[x][2]
            x++
        ENDLOOP
        Print ""
    ENDLOOP
    Print "0 MEANS AVAILABLE      1 MEANS OCCUPIED"
    Print " PLEASE CHOOSE THE ROOM FROM ( 0 – 99 )"
    Read room_num
    IF (block_b3[room_num][0] AND block_b3[room_num][1] = 1 AND
block_b3[room_num][2] = 1) THEN
        Print "SORRY PLEASE CHOOSE ANOTHER ROOM"
    ENDIF
    Print "PLEASE CHOOSE A BED (L – M – R)"
    Read bed_choice
    IF (bed_choice = "L" OR bed_choice = "l" ) THEN
        apply.bedNumber = 0
    ELSE
        IF (bed_choice = "M" OR bed_choice = "m" ) THEN
            apply.bedNumber = 1
        ENDIF
    ELSE
        IF (bed_choice = "R" OR bed_choice = "r" ) THEN
            apply.bedNumber = 2
        ENDIF
    ENDIF
    IF (block_a2[room_num][apply.bedNumber] = 1 ) THEN
        Print "SORRY THIS BED IS OCCUPIED"
    ELSE
        IF (block_a2[room_num][0] OR block_a2[room_num][1] = 0) THEN
            block_a2[room_num][apply.bedNumber] = 1
        ENDIF
    ENDIF
    RETURN room_num
END FUNCTION

FUNCTION a1_room_available()
    i = 0
    occupied_room = 0
    DOWHILE ( i < 100)
        IF (block_a1[i] = 1)
            occupied_room++
        ENDIF
        i ++
    ENDDO
    available_room = 100 – occupied_room
    RETURN available_room
END FUNCTION

```

```

FUNCTION a2_room_available()
    i = 0
    occupied_room = 0
    DOWHILE ( i < 100)
        IF ((block_a2[i][0] AND block_a2[i][1]) = 1) THEN
            occupied_room++
        ENDIF
        i ++
    ENDDO
    available_room = 100 - occupied_room
    RETURN available_room
END FUNCTION

```

```

FUNCTION b1_room_available()
    i = 0
    occupied_room = 0
    DOWHILE ( i < 100)
        IF (block_b1[i] = 1) THEN
            occupied_room++
        ENDIF
        i ++
    ENDDO
    available_room = 100 - occupied_room
    RETURN available_room
END FUNCTION

```

```

FUNCTION b3_room_available()
    i = 0
    occupied_room = 0
    DOWHILE ( i < 100)
        IF ((block_b3[i][0] AND block_b3[i][1] AND block_b3[i][2]) = 1) THEN
            occupied_room++
        ENDIF
        i ++
    ENDDO
    available_room = 100 - occupied_room
    RETURN available_room
END FUNCTION

```

```

FUNCTION laundry_calculation()
    Print "LAUNDRY = RM20"
    Print "DO YOU WANT TO ADD LAUNDRY SERVICE"
    Print "1. YES      2.NO"
    Print "PLEASE ENTER A CHOICE (1 - 2)"
    Read laundry_choice
    IF (laundry_choice = 1) THEN
        laundry = 20
        RETURN laundry
    ELSE
        laundry = 0
        RETURN laundry
    ENDIF
END FUNCTION

```

```
FUNCTION gym_calculation()
    Print " GYM  RM10"
    Print "DO YOU WANT TO ADD GYM SERVICE"
    Print "1. YES      2.NO"
    Print "PLEASE ENTER A CHOICE (1 - 2)"
    Read gym_choice
    IF (gym_choice = 1) THEN
        gym = 10
        RETURN gym
    ELSE
        gym = 0
        RETURN gym
    ENDIF
END FUNCTION
```

```

FUNCTION personal_due( rental, gym, laundry, week_stay, paid_amount)
    total_personal_amount = rental + gym + laundry
    total_personal_amount = (total_personal_amount * week_stay) – paid_amount
    RETURN total_personal_amount
END FUNCTION

FUNCTION student_login()
    Print “STUDENT LOGIN SYSTEM”
    Print “PLEASE ENTER YOUR STUDENT ID”
    Read apply.studentID
    OPEN File “Student.txt” for Read
    DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)
        IF (apply.studentID = check.studentID) THEN
            Print “SUCCESSFULLY LOGIN WITH STUDENT ID”
            check_login = 1
            BREAK
        ELSE
            check_login = 0
        ENDIF
    ENDDO
    IF (check_login = 1) THEN
        Print “WELCOME TO STUDENT LOGIN SYSTEM”
        Print “WHAT WOULD YOU LIKE TO DO”
        Print “1. CHANGE OR CANCEL ROOM 2. CHECK PERSONAL DETAIL”
        Print “3. MAKE PAYMENT 4. EXIT TO MAIN MENU”
        Read choice
        IF (choice = 1) THEN
            Print “WOULD YOU LIKE TO 1. CHANGE 2.CANCEL ROOM”
            Print “PLEASE ENTER A CHOICE”
            Read change_choice
            IF (change_choice = 1) THEN
                call change_room()
            ELSE
                IF (change_choice = 2) THEN
                    call cancel_room()
                ENDIF
            ENDIF
        ENDIF
        ELSE
            IF (choice = 2) THEN
                call student_details()
            ENDIF
        ELSE
            IF (choice = 3) THEN
                call payment_facility()
            ENDIF
        ELSE
            IF (choice = 4) THEN
                RETURN
            ENDIF
        ENDIF
    ELSE
        Print “RETURNING TO MAIN MENU”
        RETURN
    ENDIF
ENDIF

```

```

CLOSE File "Student.txt"
END FUNCTION

FUNCTION change_room()
    OPEN File "Student.txt" for Read
    DOWHILE ((Read File and retrieve data structure student at member variable check) != End
of File)
        IF ((apply.studentID = check.studentID) AND check.gender = "M") THEN
            CLOSE File "Student.txt"
            a1_available_room = a1_room_available ()
            a2_available_room = a2_room_available ()
            Print "BOOKING MENU"
            Print "BLOCK A1 AVAILBLE:"
            Print a1_available_room
            Print "BLOCK A2 AVAILABLE"
            Print a2_available_room
            Print "BLOCK A1 = SINGLE ROOM = RM400 WEEKLY"
            Print "BLOCK A2 = DOUBLE ROOM = RM200 WEEKLY"
            Print "PLEASE INSERT A CHOCE ( 1 - 2 )"
            Read block_type
            IF (block_type = 1) THEN
                new_rental = 400
                check.blockArea = "A1"
                new_room_num = call a1room_visual()
                new_laundry = call laundry_calculation()
                new_gym = call gym_calculation()
                Print "HOW MANY WEEKS WOULD YOU LIKE TO STAY?"
                Read new_weekStay
                new_paidAmount = 0
                new_total_due = call personal_due(new_rental, new_gym,
new_laundry, new_weekStay, new_paidAmount)
                OPEN File "Student.txt" for Read
                OPEN File "NewStudent.txt" for Write
                DOWHILE ((Read File "Student.txt" and retrieve data structure
student at member variable check) != End of File)
                    IF (apply.studentID = check.studentID)
                        Write new data inside text file "NewStudent.txt"
                    ELSE
                        Write previous data inside text file "NewStudent.txt"
                    ENDIF
                ENDDO
                CLOSE File "Student.txt"
                CLOSE File "NewStudent.txt"
                REMOVE File "Student.txt"
                RENAME File "NewStudent.txt" to "Student.txt"
                Print " CHANGE ROOM SUCCESS, PLEASE LOGIN AGAIN"
            ELSE
                new_rental = 200
                check.blockArea = "A2"
                new_room_num = call a2room_visual()
                new_laundry = call laundry_calculation()
                new_gym = call gym_calculation()
                Print "HOW MANY WEEKS WOULD YOU LIKE TO STAY?"
                Read new_weekStay
                new_paidAmount = 0
            ENDIF
        ENDIF
    ENDWHILE
ENDFUNCTION

```

```

        new_total_due = call personal_due(new_rental, new_gym,
new_laundry, new_weekStay, new_paidAmount)
        OPEN File "Student.txt" for Read
        OPEN File "NewStudent.txt" for Write
        DOWHILE ((Read File "Student.txt" and retrieve data structure
student at member variable    check) != End of File)
            IF (apply.studentID = check.studentID) THEN
                Write new data inside text file "NewStudent.txt"
            ELSE
                Write previous data inside text file "NewStudent.txt"
            ENDIF
        ENDDO
        CLOSE File "Student.txt"
        CLOSE File "NewStudent.txt"
        REMOVE File "Student.txt"
        RENAME File "NewStudent.txt" to "Student.txt"
        Print " CHANGE ROOM SUCCESS, PLEASE LOGIN AGAIN"
    ENDIF
    ELSE
        IF ((apply.studentID = check.studentID) AND check.gender = "F") THEN
            CLOSE File "Student.txt"
            b1_available_room = b1_room_available()
            b3_available_room = b3_room_available()
            Print "BOOKING MENU"
            Print "BLOCK B1 AVAILABLE:"
            Print a1_available_room
            Print "BLOCK B3 AVAILABLE"
            Print b3_available_room
            Print "BLOCK B1 = SINGLE ROOM = RM400 WEEKLY"
            Print "BLOCK B3 = TRIPLE ROOM = RM150 WEEKLY"
            Print "PLEASE INSERT A CHOCE ( 1 - 2 )"
            Read block_type
            IF (block_type = 1) THEN
                new_rental = 400
                check.blockArea = "A1"
                new_room_num = call a1room_visual()
                new_laundry = call laundry_calculation()
                new_gym = call gym_calculation()
                Print "HOW MANY WEEKS WOULD YOU LIKE TO
STAY?"
                Read new_weekStay
                new_paidAmount = 0
                new_total_due = call personal_due(new_rental, new_gym,
new_laundry, new_weekStay, new_paidAmount)
                OPEN File "Student.txt" for Read
                OPEN File "NewStudent.txt" for Write
                DOWHILE ((Read File "Student.txt" and retrieve data
structure student at member variable    check) != End of File)
                    IF (apply.studentID = check.studentID) THEN
                        Write new data inside text file
                    ELSE
                        Write previous data inside text file
                    ENDIF
                "NewStudent.txt"
                "NewStudent.txt"
            ENDIF
        ENDIF
    ENDIF

```

```

        ENDDO
        CLOSE File "Student.txt"
        CLOSE File "NewStudent.txt"
        REMOVE File "Student.txt"
        RENAME File "NewStudent.txt" to "Student.txt"
        Print " CHANGE ROOM SUCCESS, PLEASE LOGIN

AGAIN"
        ELSE
            new_rental = 150
            check.blockArea = "B3"
            new_room_num = call b3room_visual()
            new_laundry = call laundry_calculation()
            new_gym = call gym_calculation()
            Print "HOW MANY WEEKS WOULD YOU LIKE TO
STAY?"
            Read new_weekStay
            new_paidAmount = 0
            new_total_due = call personal_due(new_rental, new_gym,
new_laundry, new_weekStay, new_paidAmount)
            OPEN File "Student.txt" for Read
            OPEN File "NewStudent.txt" for Write
            DOWHILE ((Read File "Student.txt" and retrieve data
check) != End of File)
                IF (apply.studentID = check.studentID) THEN
                    Write new data inside text file
                "NewStudent.txt"
                ELSE
                    Write previous data inside text file
                "NewStudent.txt"
                ENDIF
            ENDIF
            CLOSE File "Student.txt"
            CLOSE File "NewStudent.txt"
            REMOVE File "Student.txt"
            RENAME File "NewStudent.txt" to "Student.txt"
            Print " CHANGE ROOM SUCCESS, PLEASE LOGIN

AGAIN"
        ENDIF
    ENDIF
    ENDDO
END FUNCTION

```

```

FUNCTION cancel_room()
Print "ARE YOU SURE YOU WANT TO CANCEL ROOM? Y OR N"
Read cancel_choice
IF (cancel_choice = "Y") THEN
    OPEN File "Student.txt" for Read
    OPEN File "NewStudent.txt" for Write
    DOWHILE ((Read File and retrieve data structure student at member variable
check) != End of File)
        IF (apply.studentID = check.studentID) THEN
            CONTINUE
        ELSE
            Write previous data inside text file "NewStudent.txt"
        ENDIF
    ENDDO
    Print "CANCEL ROOM SUCCESSFUL"
    CLOSE File "Student.txt"
    CLOSE File "NewStudent.txt"
    REMOVE File "Student.txt"
    RENAME File "NewStudent.txt" to "Student.txt"
ELSE
    IF (cancel_choice = "N") THEN
        RETURN
    ENDIF
ENDIF
END FUNCTION

```

```

FUNCTION student_details()
OPEN File "Student.txt" for Read
DOWHILE ((Read File and retrieve data structure student at member variable check) != End
of File)
    IF (apply.studentID = check.studentID) THEN
        Print "STUDENT ID: ", check.studentID
        Print "NAME:", check.name
        Print "GENDER: ", check.gender
        Print "AGE: ", check.age
        Print "PHONE :", check.phoneNumber
        Print "BLOCK :", check.blockArea
        Print "ROOM NUMBER", check.room_number
        Print "WEEK OF STAY: ", check.weekStay
        Print "AMOUNT DUE:", check.amountDue
        Print "PAID AMOUNT:", check.paidAmount
        Print "HERE ARE YOUR DETAILS, ENTER ANYTHING TO EXIT :"
        Read choice
    ENDIF
ENDDO
CLOSE File "Student.txt"
END FUNCTION

```

```

FUNCTION payment_facility()
    Print "PAYMENT FACILITY"
    OPEN File "Student.txt" for Read
    DOWHILE ((Read File and retrieve data structure student at member variable check) != End
    of File)
        IF (apply.studentID = check.studentID) THEN
            payment_due = check.amountDue
        ENDIF
    ENDDO
    CLOSE File "Student.txt"
    Print "HI , YOUR PAYMENT DURE IS ", payment_due
    Print "HOW MUCH WOULD YOU LIKE TO PAY?"
    Read payment_made
    OPEN File "Student.txt" for Read
    OPEN File "NewStudent.txt" for Write
    DOWHILE ((Read File and retrieve data structure student at member variable check) != End
    of File)
        IF (apply.studentID = check.studentID) THEN
            new_payment_due = payment_due - payment_made
            new_payment_made = check.paidAmount + payment_made
            Write new_payment_due and new_payment_made the rest of student data to
            text file "NewStudent.txt"
        ELSE
            Write previous data into text file "NewStudent.txt"
        ENDIF
    ENDDO
    Print "PAYMENT IS SUCCESSFUL"
    CLOSE File "Student.txt"
    CLOSE File "NewStudent.txt"
    REMOVE File "Student.txt"
    RENAME File "NewStudent.txt" to "Student.txt"
END FUNCTION

```

```

FUNCTION admin_login()
    password = 1234
    Print "ADMIN LOGIN SYSTEM"
    Print "PLEASE ENTER PASSWORD"
    Read check_password
    IF (check_password = 1234) THEN
        call admin_system()
    ELSE
        Print "WRONG PASSWORD PLEASE TRY AGAIN"
        RETURN
    ENDIF
END FUNCTION

```

```

FUNCTION admin_system()
Print "WELCOME TO ADMIN SYSTEM"
Print "1) CHECK OVERALL DETAILS 2) SEARCH FOR STUDENT DETAILS"
Read admin_choice
IF (admin_choice = 1) THEN
    call check_r_available_and_m_receivables()
ELSE
    IF (admin_choice = 2) THEN
        call search_student()
    ENDIF
ELSE
    RETURN
ENDIF
END FUNCTION

FUNCTION check_r_available_and_m_receivables()
OPEN File "Student.txt" for Read
a1_room_count = call a1_room_available()
a2_room_count = call a2_room_available(),
b1_room_count = call b1_room_available(),
b3_room_count = call b3_room_available(),
DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)
    IF (check.blockArea = "A1") THEN
        a1_amount_receivable = a1_amount_receivable + check.amountDue
        a1_payment_received = a1_payment_received + check.paidAmount
        a1_student_amount++
    ELSE
        IF (check.blockArea = "A2") THEN
            a2_amount_receivable = a2_amount_receivable + check.amountDue
            a2_payment_received = a2_payment_received + check.paidAmount
            a2_student_amount++
        ENDIF
    ELSE
        IF (check.blockArea = "B1") THEN
            b1_amount_receivable = b1_amount_receivable + check.amountDue
            b1_payment_received = b1_payment_received + check.paidAmount
            b1_student_amount++
        ENDIF
    ELSE
        b3_amount_receivable = b3_amount_receivable + check.amountDue
        b3_payment_received = b3_payment_received + check.paidAmount
        b3_student_amount++
    ENDIF
ENDDO
total_amount_receivable = a1_amount_receivable + a2_amount_receivable +
b1_amount_receivable + b3_amount_receivable
total_room_available = a1_room_count + a2_room_count + b1_room_count +
b3_room_count
total_amount_payment = a1_payment_received + a2_payment_received +
b1_payment_received + b3_payment_received
total_student_amount = a1_student_amount + a2_student_amount + b1_student_amount +
b3_student_amount
Print "Block A1 Available Room :", a1_room_count
Print "Block A2 Available Room :", a2_room_count

```

```

Print "Block B1 Available Room :", b1_room_count
Print "Block B3 Available Room :", b3_room_count
Print "Total Room Available :", total_room_available
Print "Total Amount Receivable :", total_amount_receivable
Print "Total Payment Received:", total_amount_payment
Print "If you want to check all the student details including amount receivable input 1"
Read check_choice
IF (check_choice = 1) THEN
    call check_block_student()
ELSE
    Print "INVALID INPUT"
    RETURN
ENDIF
CLOSE File "Student.txt"
ENDFUNCTION

```

```

FUNCTION check_block_student()
OPEN File "Student.txt" for Read
Print "PLEASE CHOOSE THE BLOCK YOU WANT TO VIEW (A1, A2, B1,B3)"
Read check_block
IF (check_block = "A1") THEN
    DOWHILE ((Read File and retrieve data structure student at member variable
check) != End of File)
        IF (check.blockArea = "A1")
            Print "STUDENT ID:",check.studentID
            Print "NAME:",check.name
            Print "AMOUNT RECEIVABLE: ", check.amountDue
        ENDIF
    ENDDO
ELSE
    IF (check_block = "A2") THEN
        DOWHILE ((Read File and retrieve data structure student at member
variable check) != End of File)
            IF (check.blockArea = "A2")
                Print "STUDENT ID:",check.studentID
                Print "NAME:",check.name
                Print "AMOUNT RECEIVABLE: ", check.amountDue
            ENDIF
        ENDDO
    ENDIF
ELSE
    IF (check_block = "B1") THEN
        DOWHILE ((Read File and retrieve data structure student at member
variable check) != End of File)
            IF (check.blockArea = "B1")
                Print "STUDENT ID:",check.studentID
                Print "NAME:",check.name
                Print "AMOUNT RECEIVABLE: ", check.amountDue
            ENDIF
        ENDDO
    ENDIF
ELSE
    IF (check_block = "B3") THEN
        DOWHILE ((Read File and retrieve data structure student at member
variable check) != End of File)
    ENDIF
ENDIF

```

```

        IF (check.blockArea = "B3")
            Print "STUDENT ID:",check.studentID
            Print "NAME:",check.name
            Print "AMOUNT RECEIVABLE: ", check.amountDue
        ENDIF
    ENDDO
ENDIF
CLOSE File "Student.txt"
Print "PRESS ANYTHING TO EXIT"
Read exit_choice
RETURN
END FUNCTION

FUNCTION search_students()
    Print "STUDENT SEARCH SYSTEM"
    Print "PLEASE INSERT THE STUDENT ID TO SEARCH FOR STUDENT DETAILS :"
    Read search_studentID
    OPEN File "Student.txt" for Read
    DOWHILE ((Read File and retrieve data structure student at member variable check) != End of File)
        IF (apply.studentID = check.studentID) THEN
            check_login = 1
            BREAK
        ELSE
            check_login = 0
        ENDIF
    ENDDO
    IF (check_login = 0) THEN
        Print ("NO RECORD WAS FOUND")
        RETURN
    ELSE
        Print "BOOKING DETAILS OF SEARCH STUDENT"
        Print "STUDENT ID: ", check.studentID
        Print "NAME: ", check.name
        Print "GENDER: ", check.gender
        Print "AGE: ", check.age
        Print "PHONE :", check.phoneNumber
        Print "BLOCK :", check.blockArea
        Print "ROOM NUMBER", check.room_number
        Print "WEEK OF STAY: ", check.weekStay
        Print "AMOUNT DUE:", check.amountDue
        Print "PAID AMOUNT:", check.paidAmount
        Print "HERE ARE YOUR DETAILS, ENTER ANYTHING TO EXIT :"
        Read choice
        RETURN
    ENDIF
END FUNCTION

```

2.2 | Flow Chart

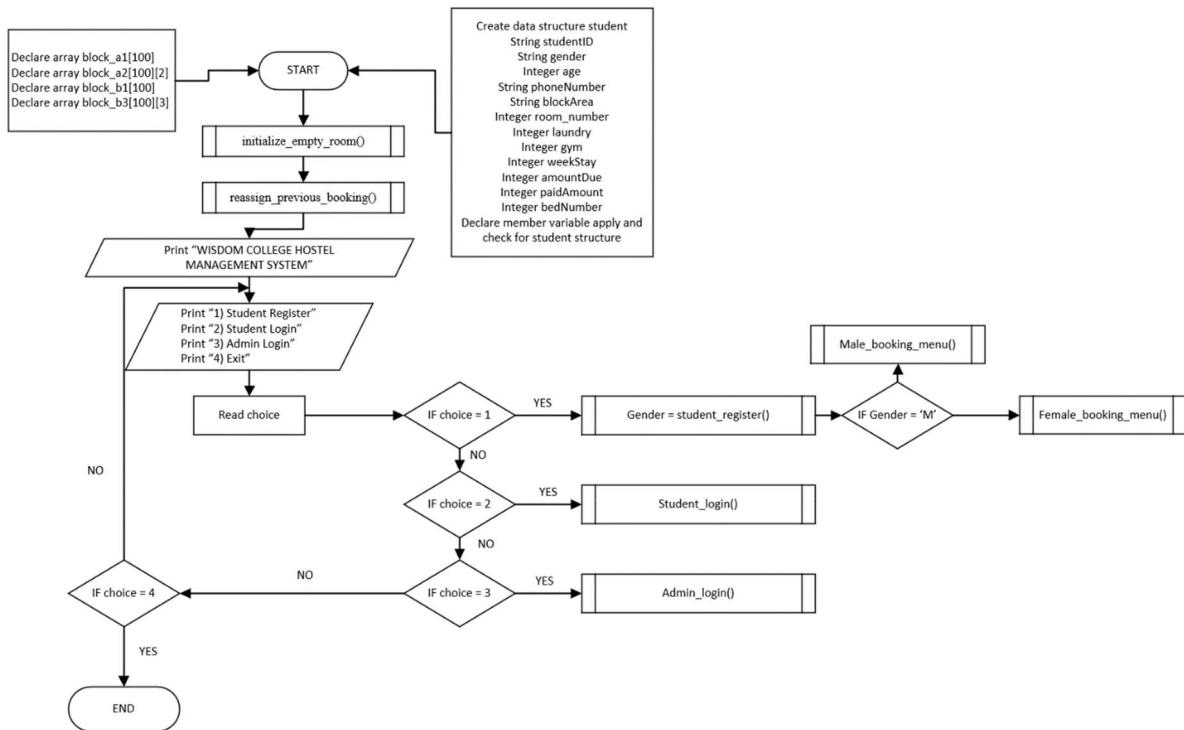


Figure 2.2.1 Hostel Management Main Menu

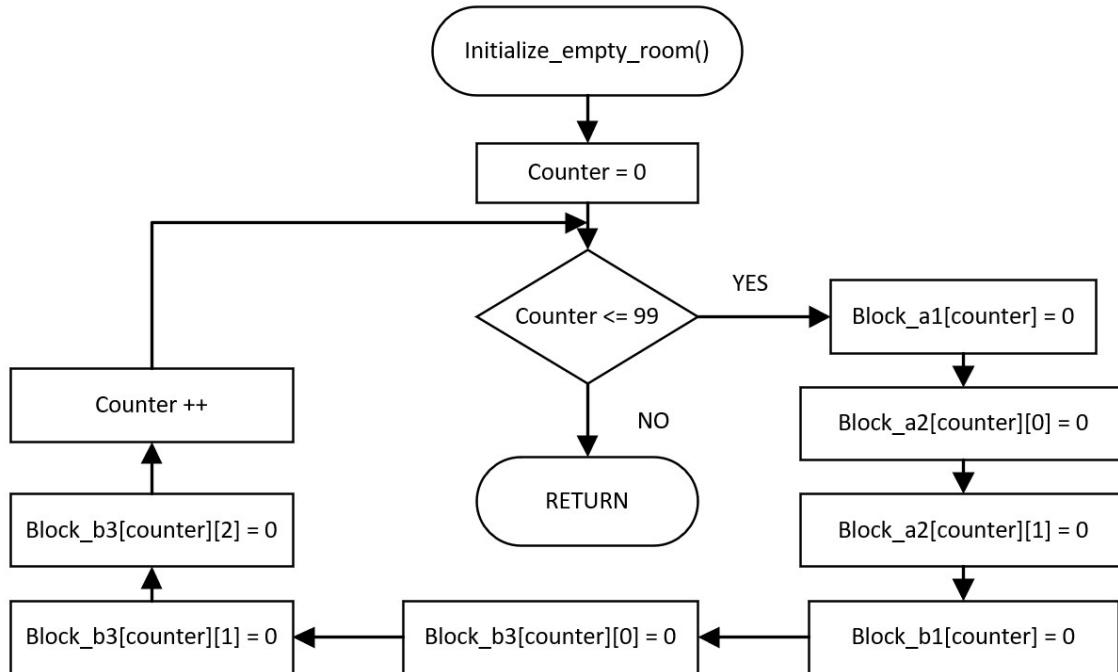


Figure 2.2.2 Initialize Empty Room Function

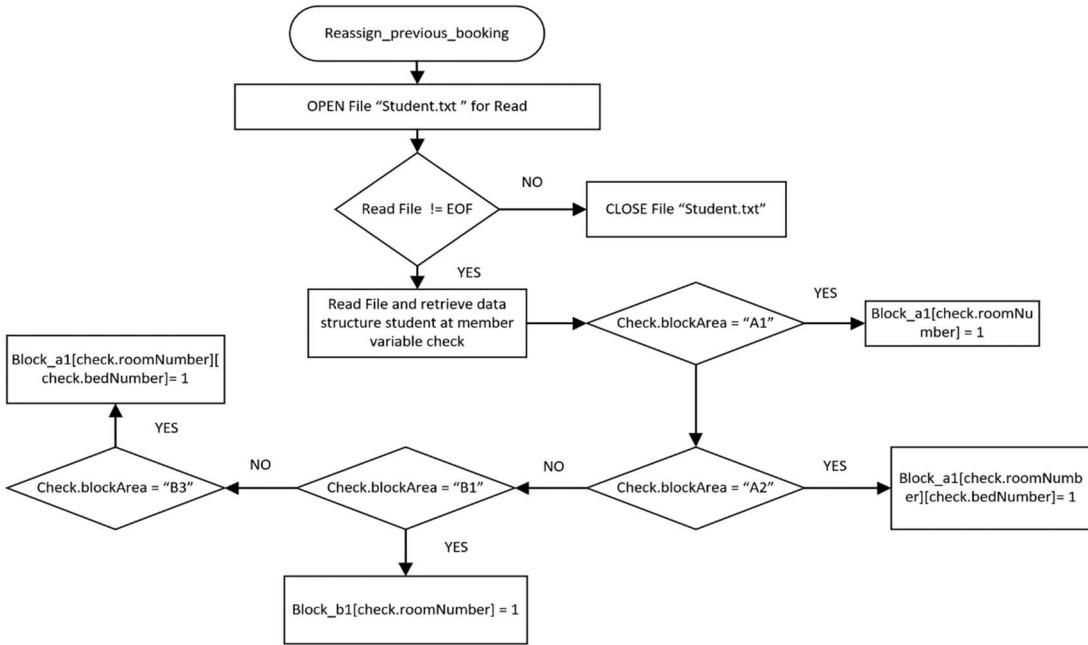


Figure 2.2.3 Reassign Previous Booking Function

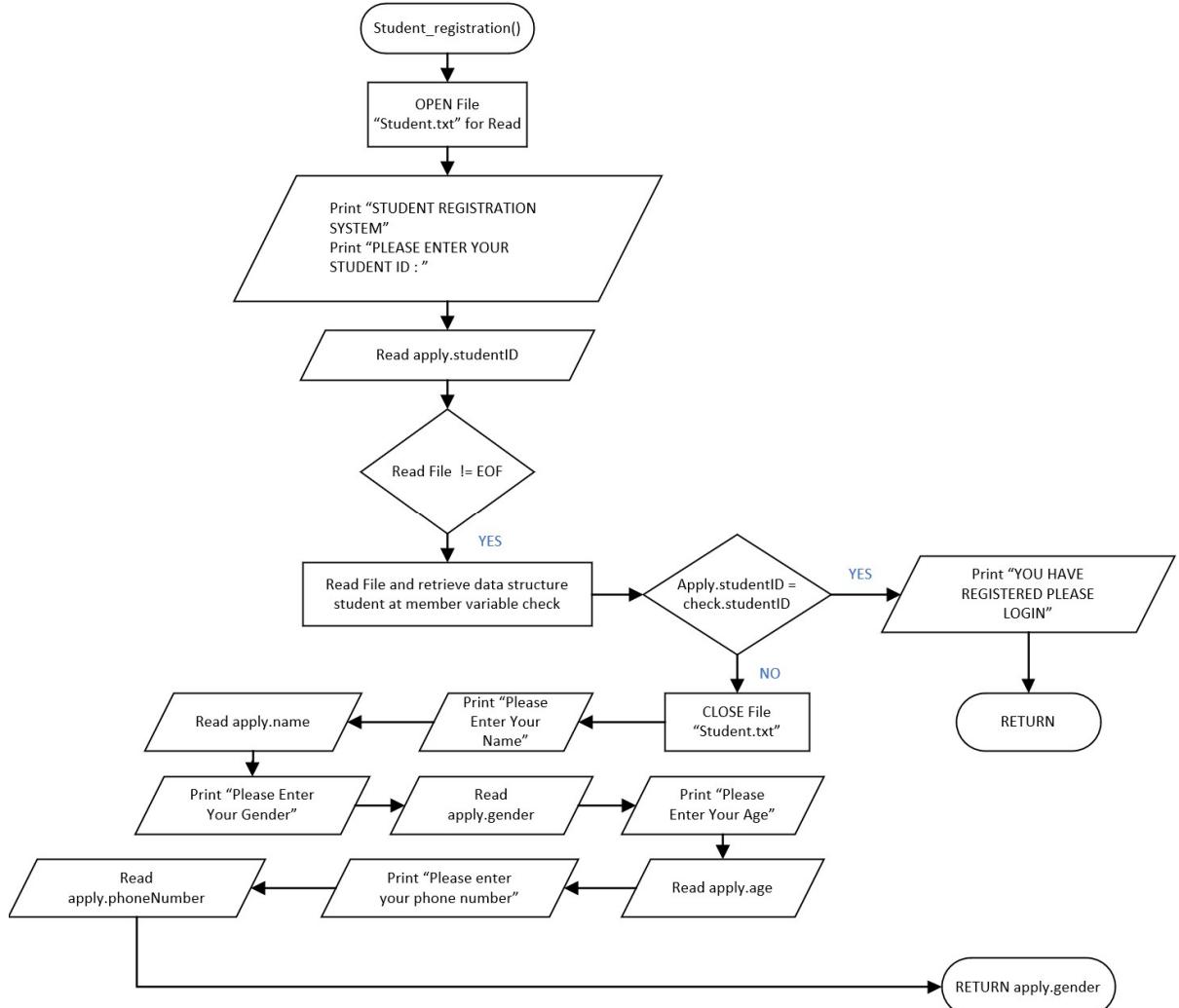


Figure 2.2.4 Student Registration Function

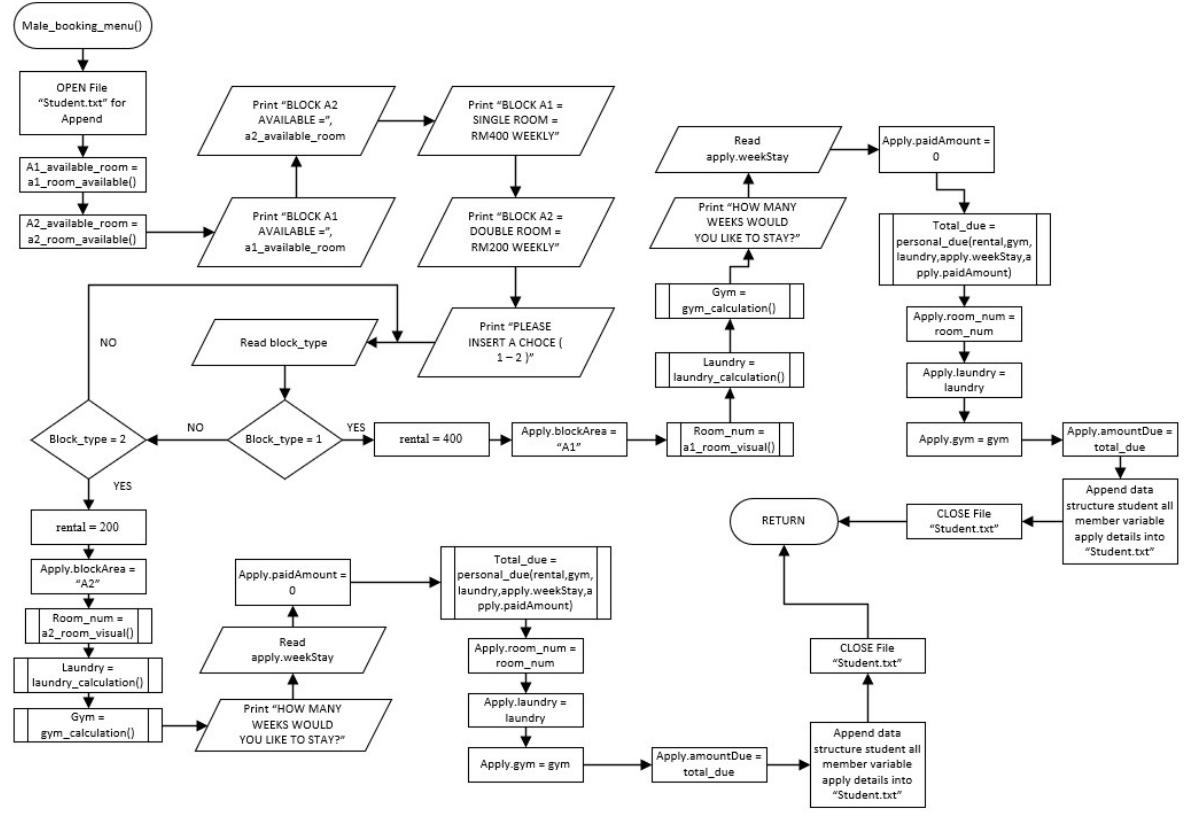


Figure 2.2.5 Male Booking Menu Function

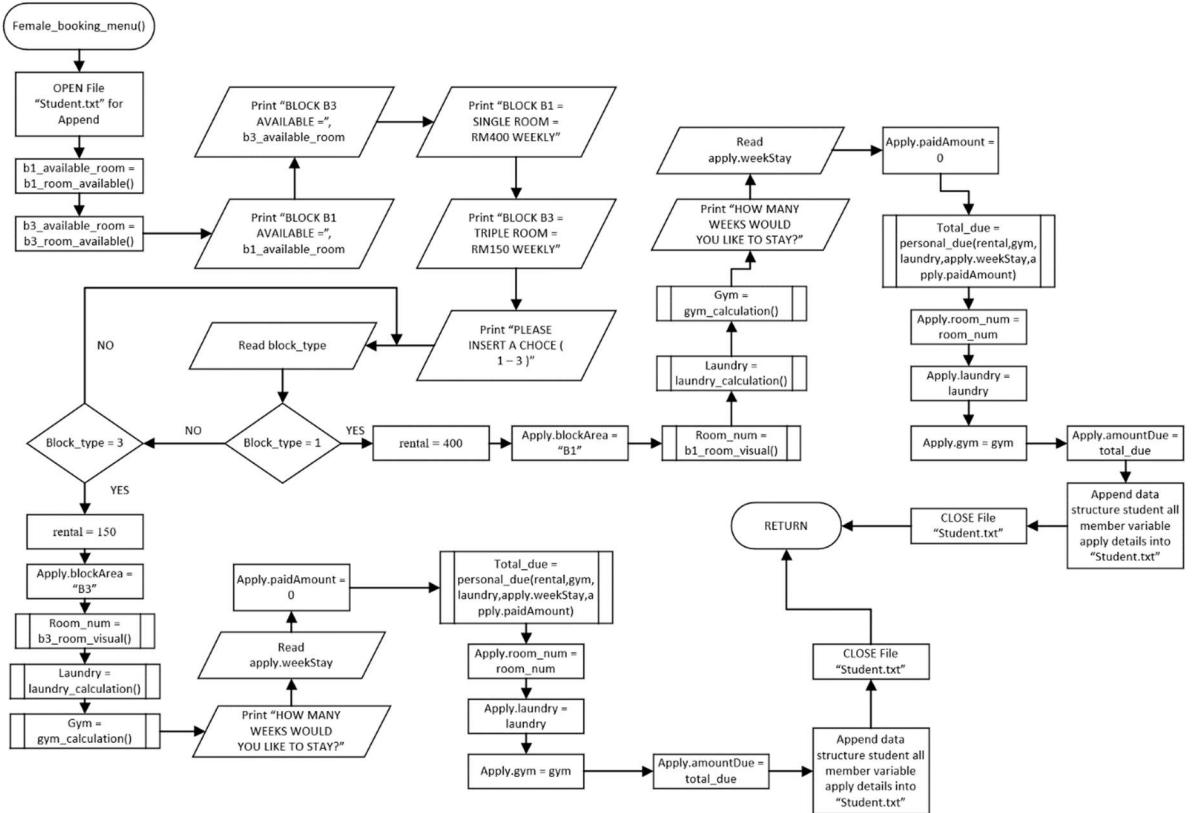


Figure 2.2.6 Female Booking Menu Function

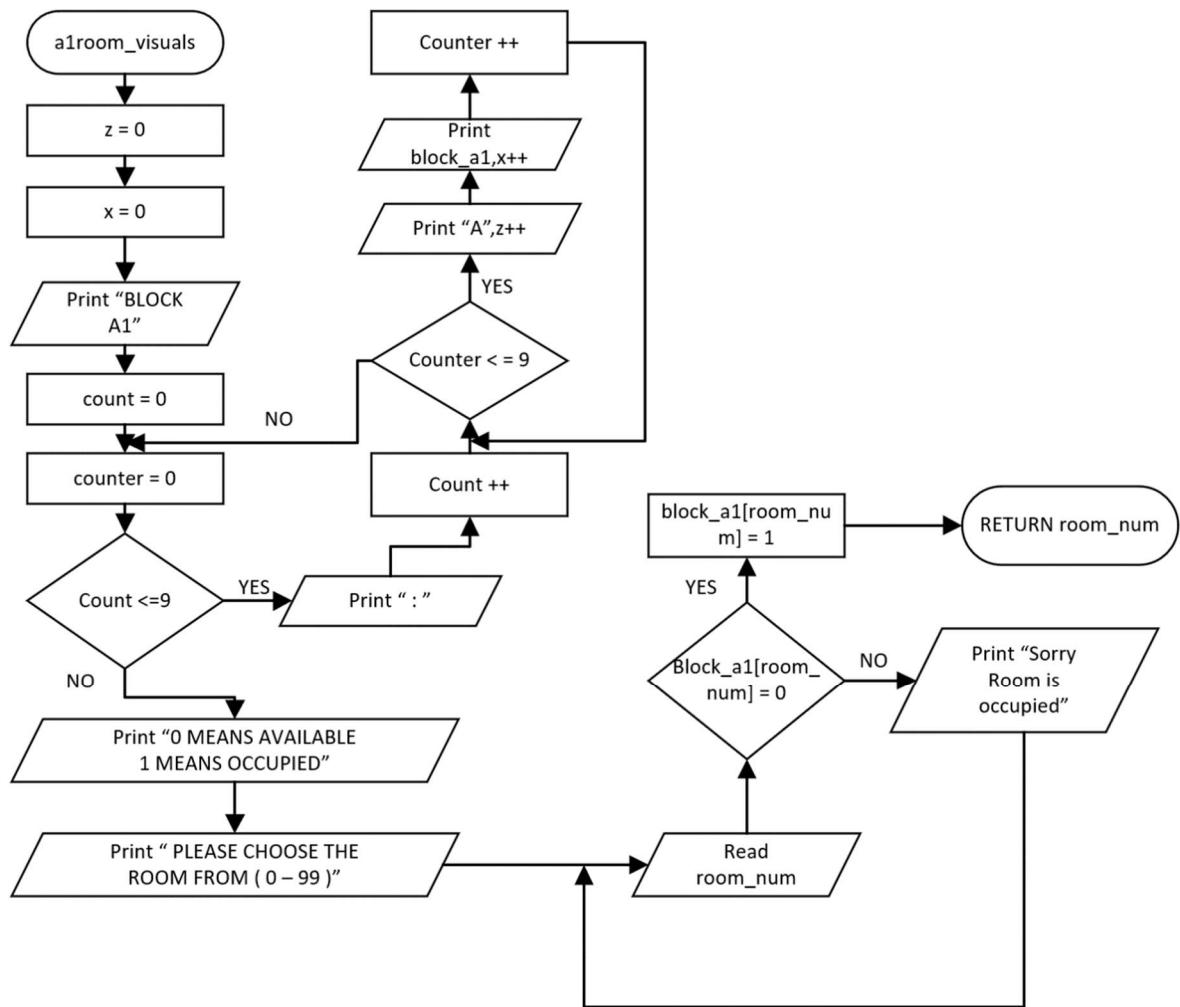


Figure 2.2.7 A1 Room Visual Function

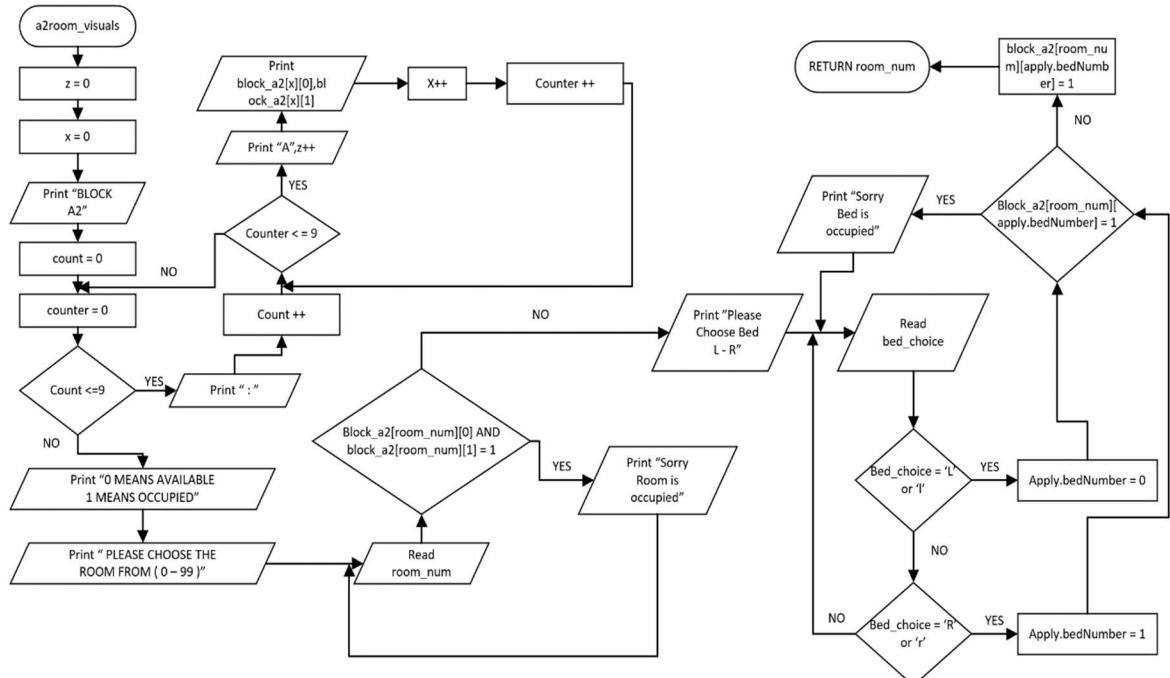


Figure 2.2.8 A2 Room Visual Function

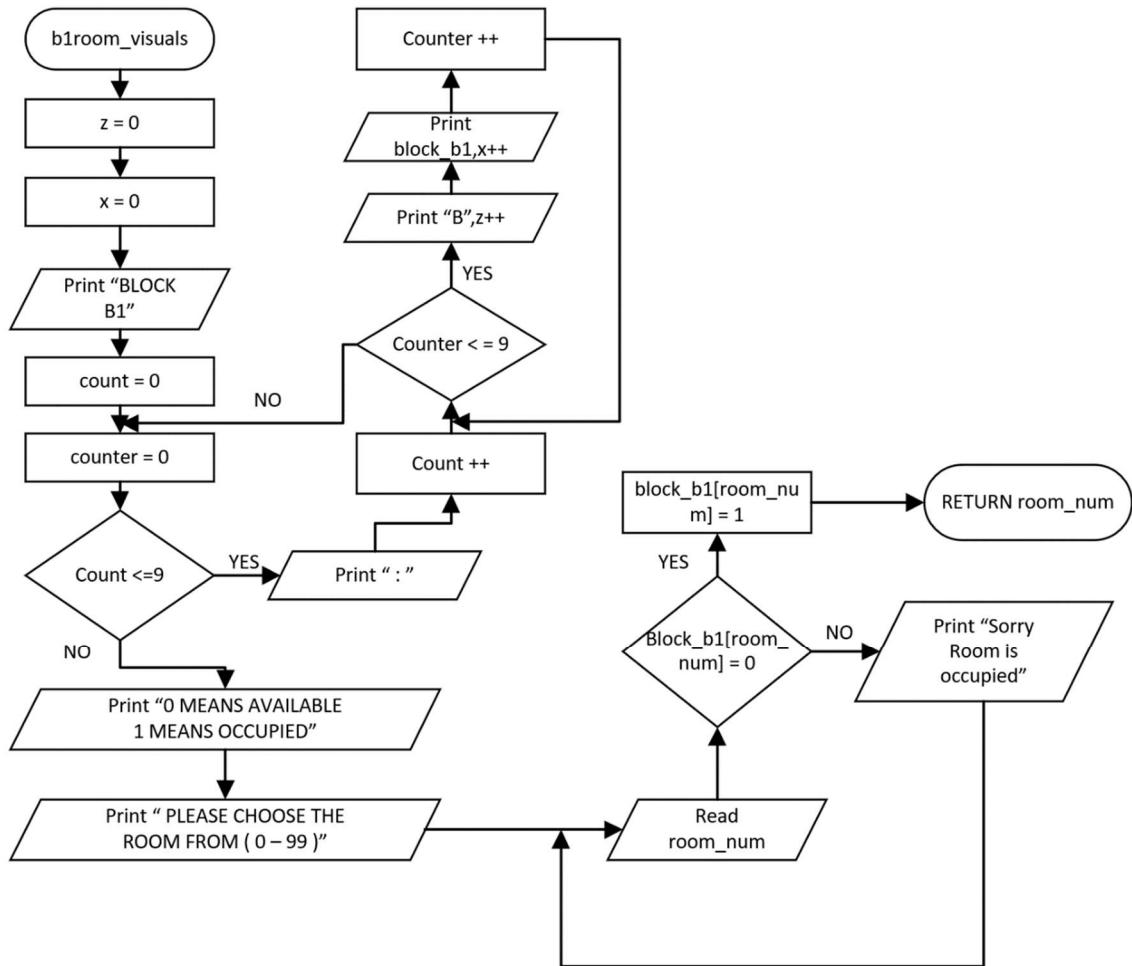


Figure 2.2.9 B1 Room Visuals

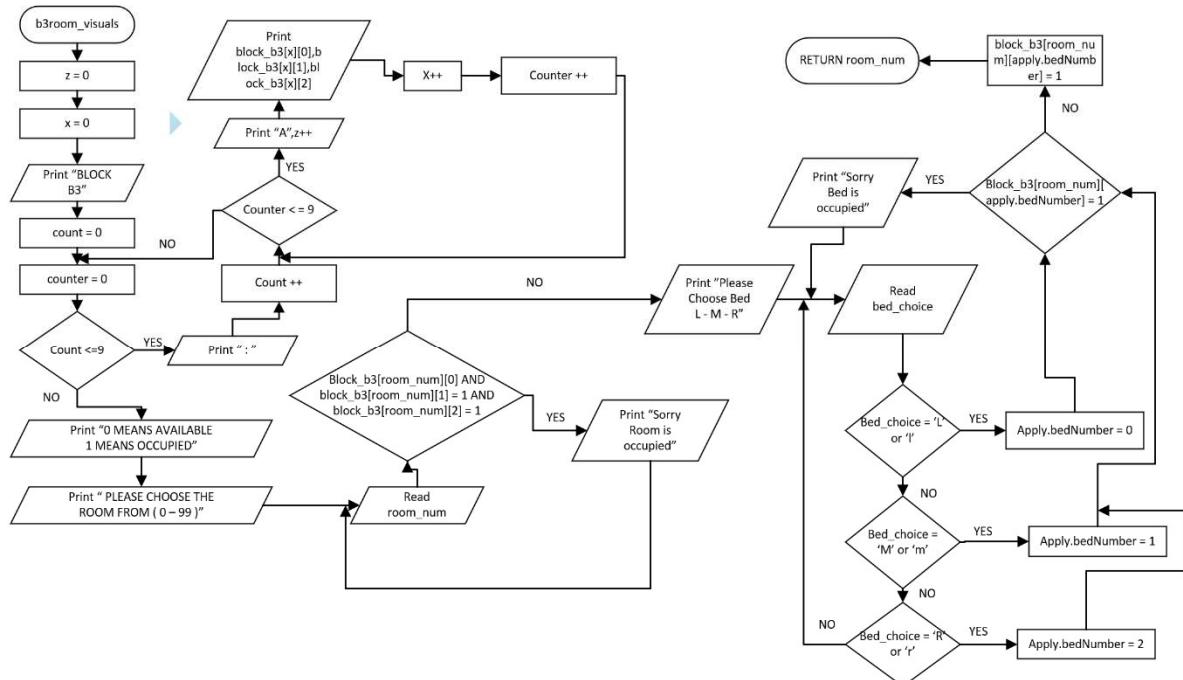


Figure 2.2.10 B3 Room Visuals

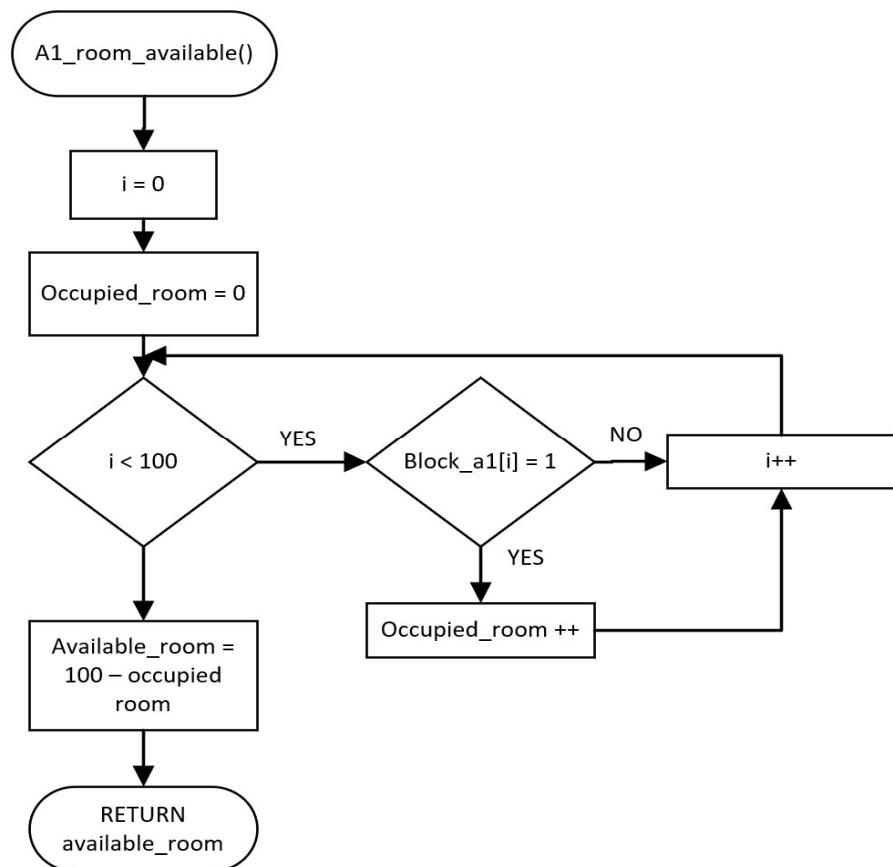


Figure 2.2.11 A1 Room Available Function

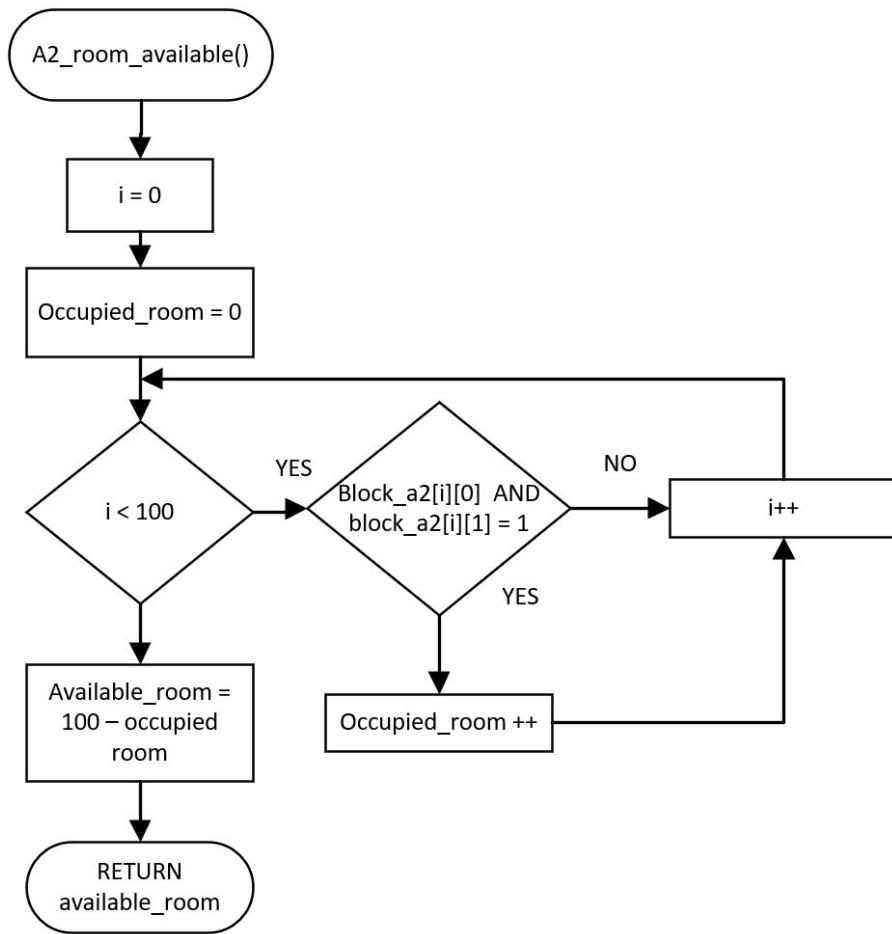


Figure 2.2.12 A2 Room Available Function

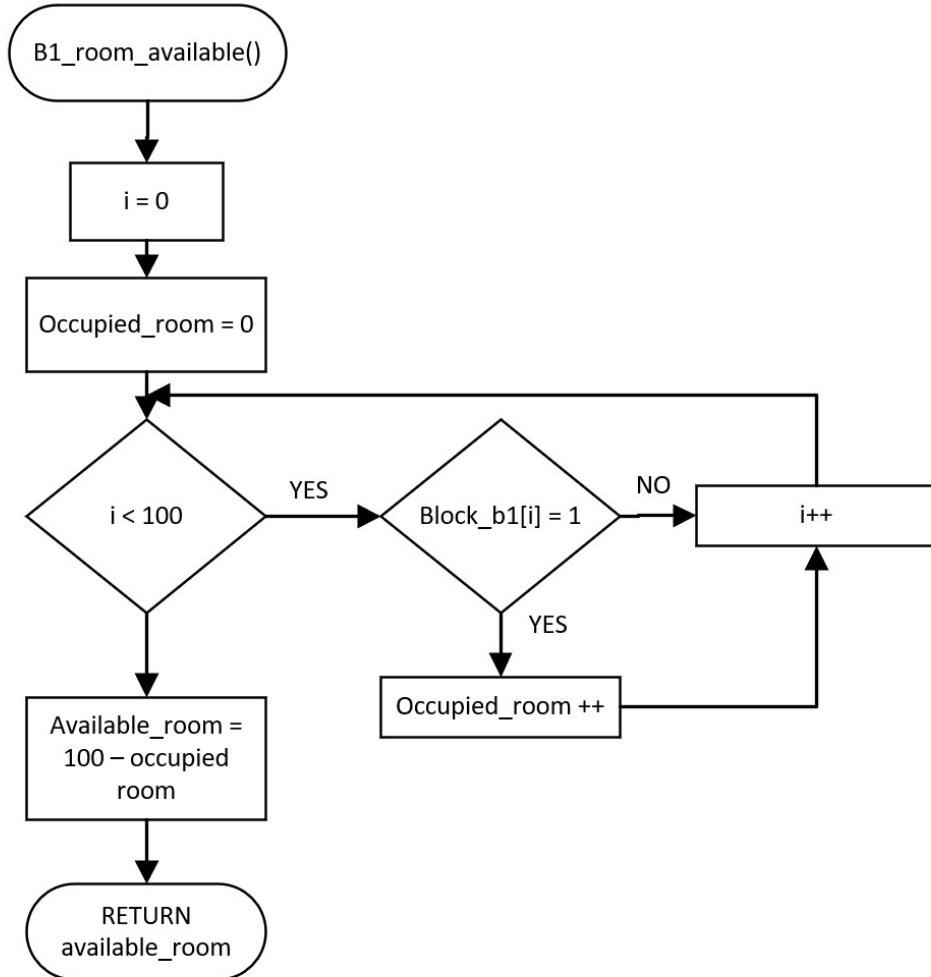


Figure 2.2.13 B1 Room Available Function

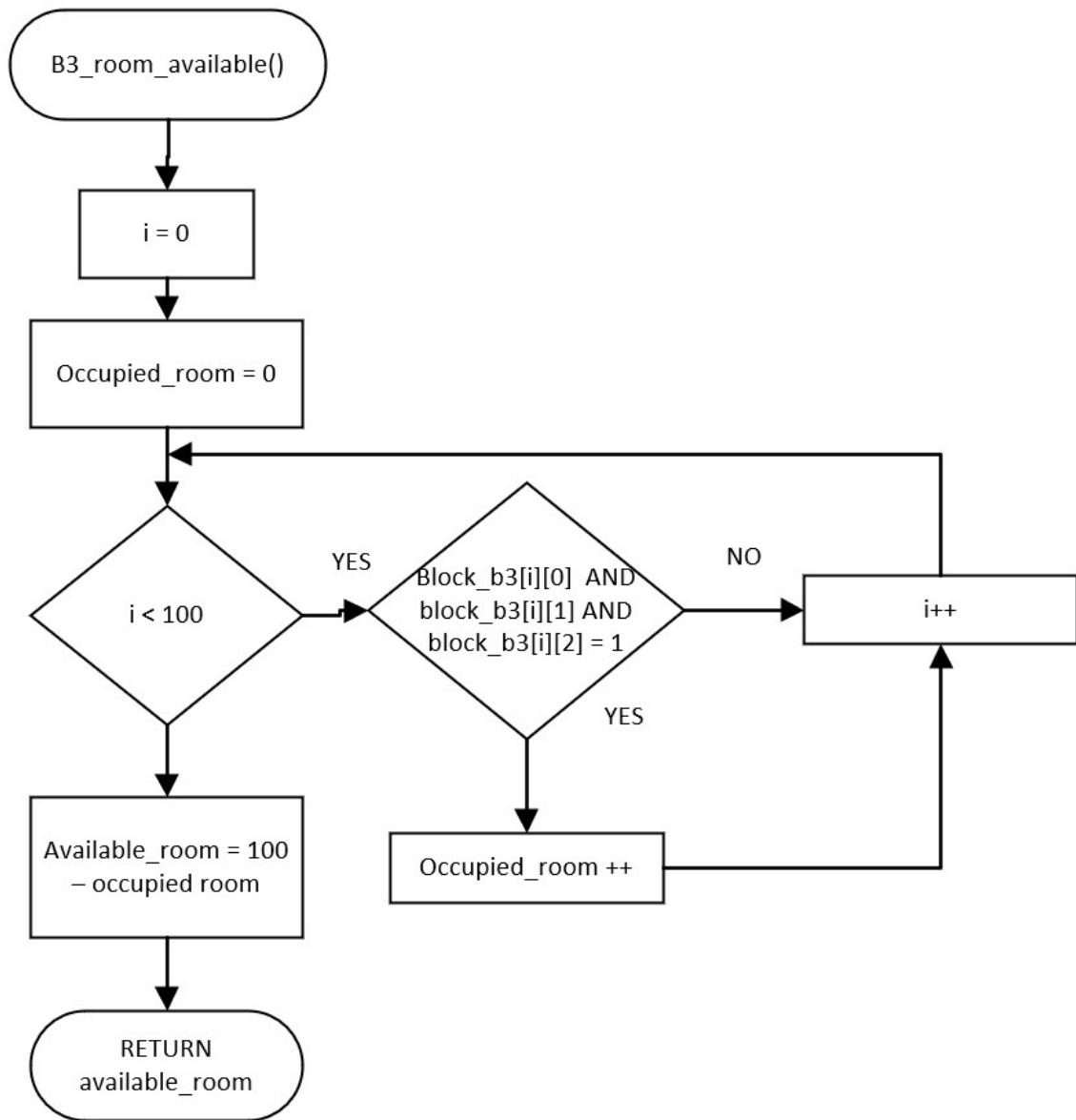


Figure 2.2.14 B3 Room Available Function

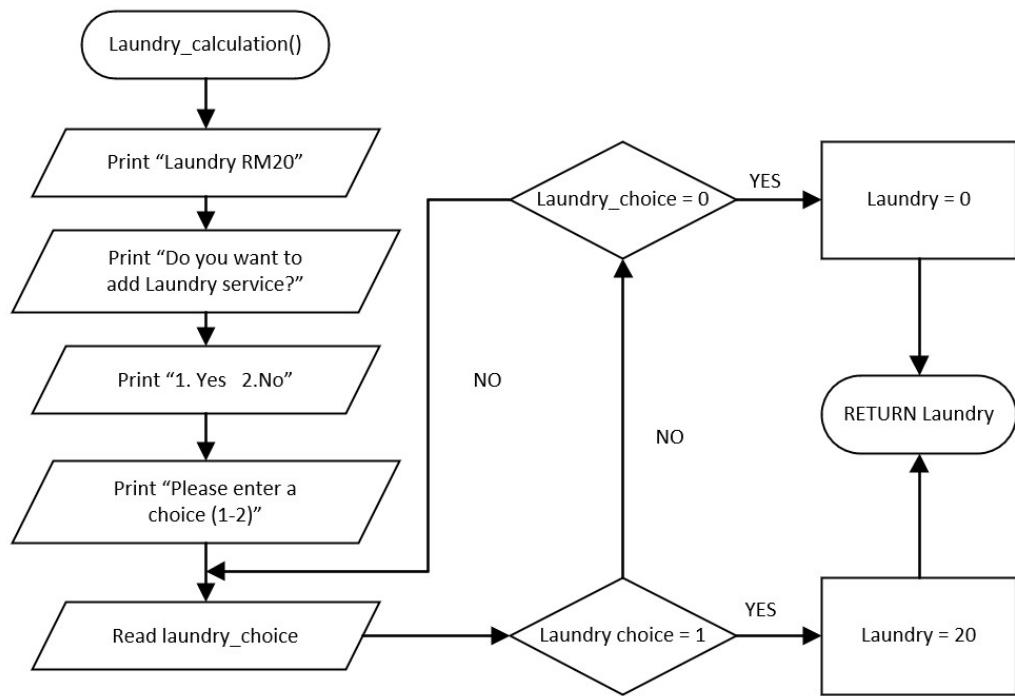


Figure 2.2.15 Laundry Calculation Function

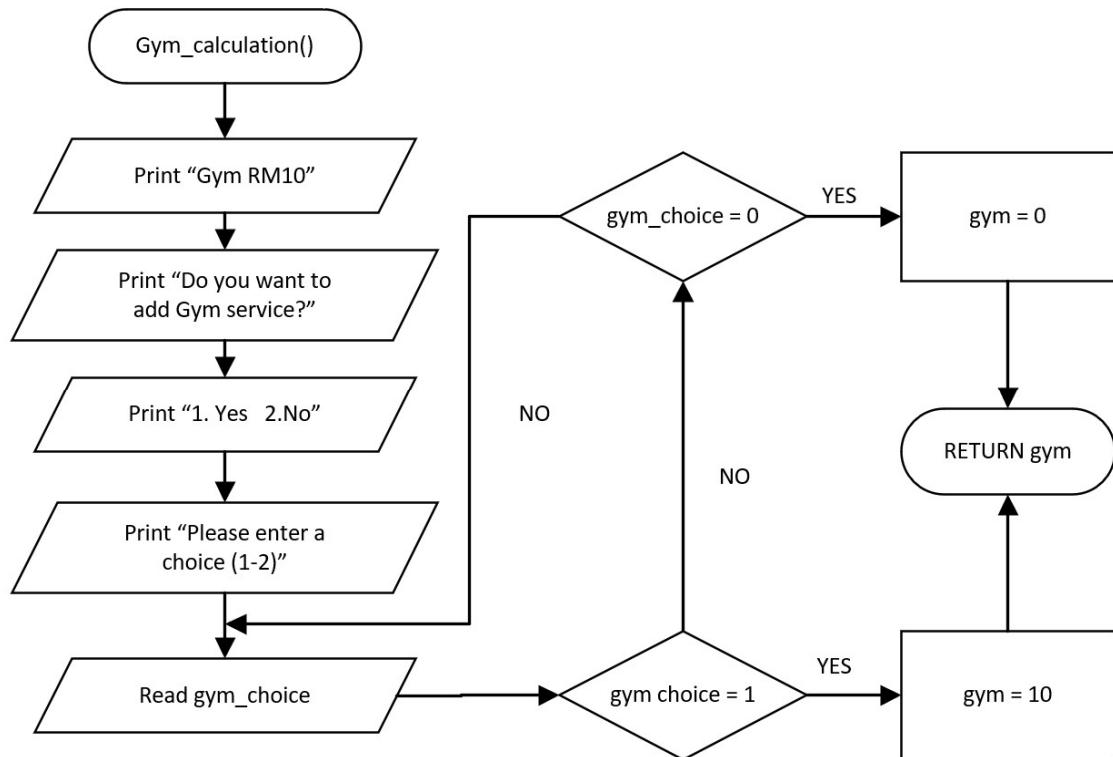


Figure 2.2.16 Gym Calculation Function

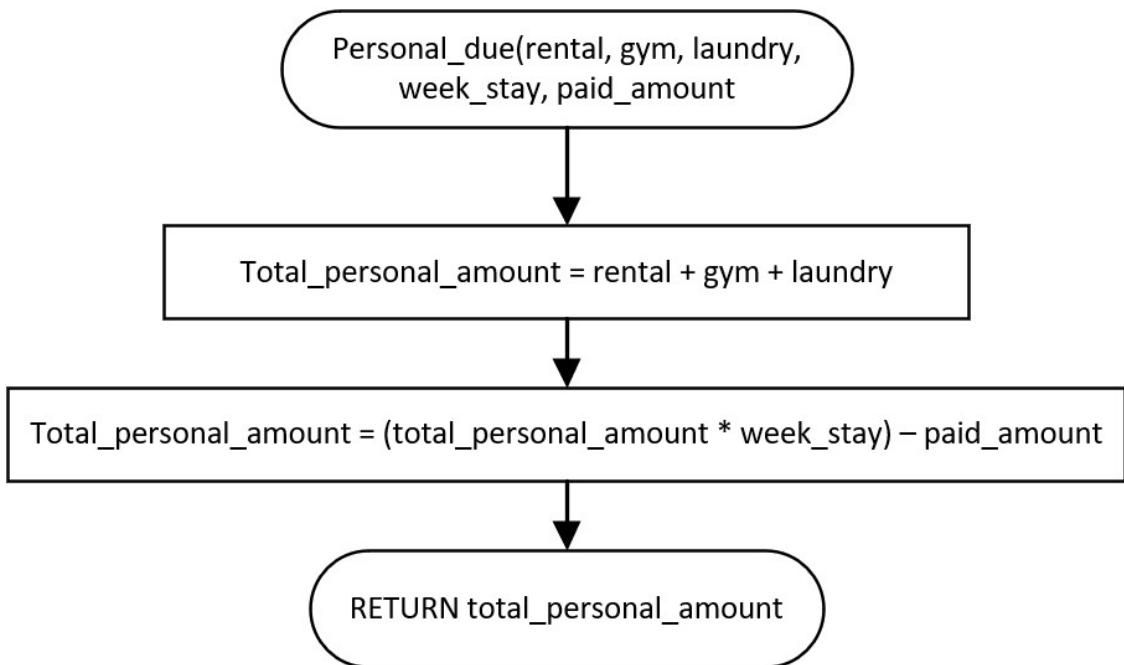


Figure 2.2.17 Personal Due Function

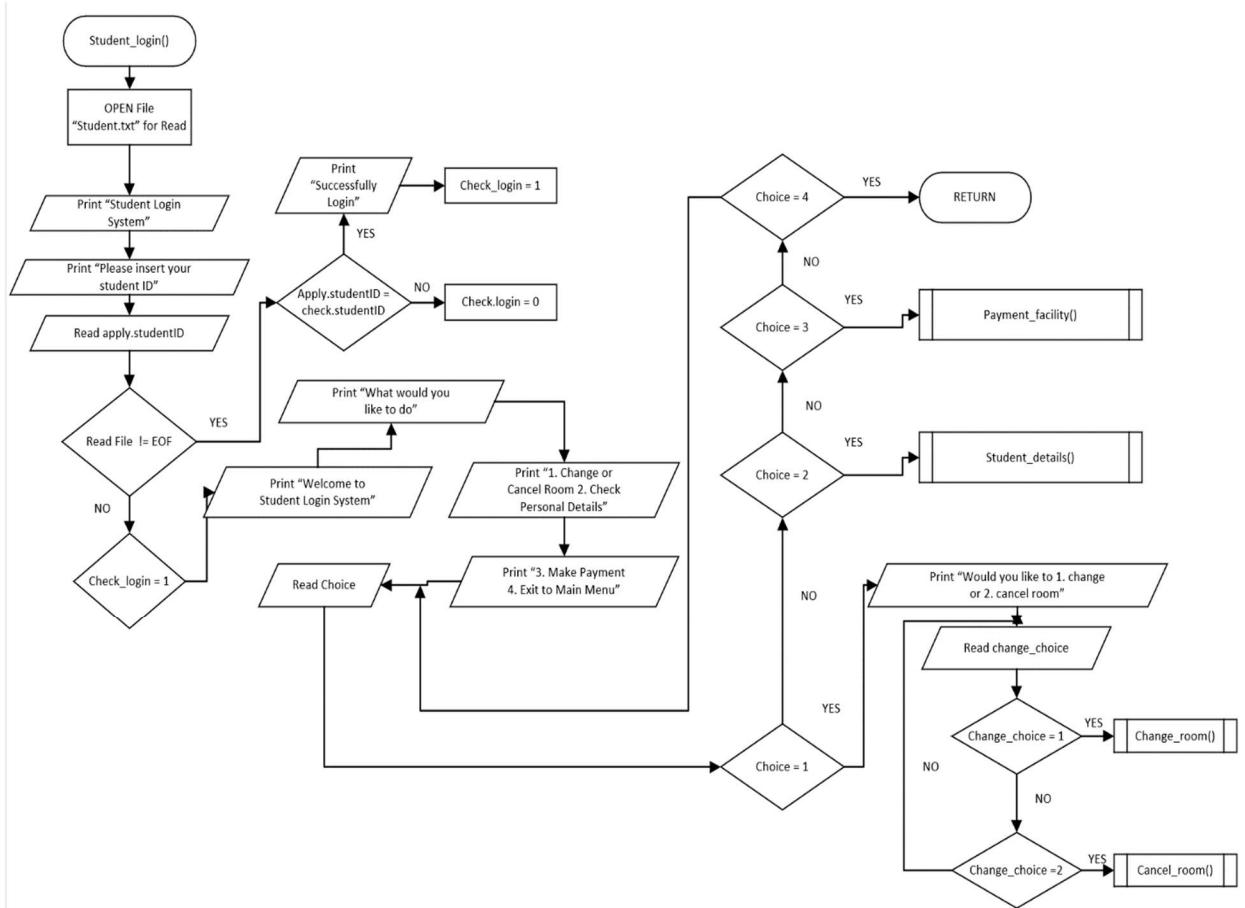


Figure 2.2.18 Student Login Function

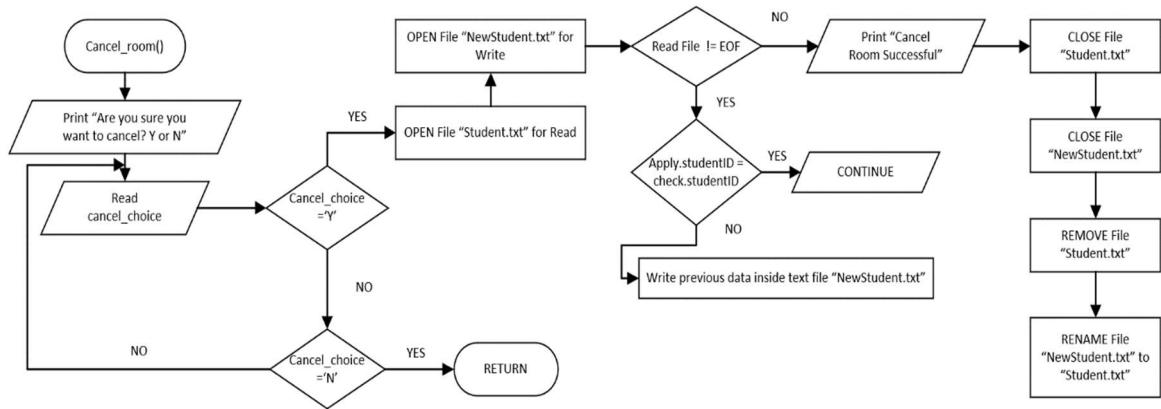


Figure 2.2.19 Cancel Room Function

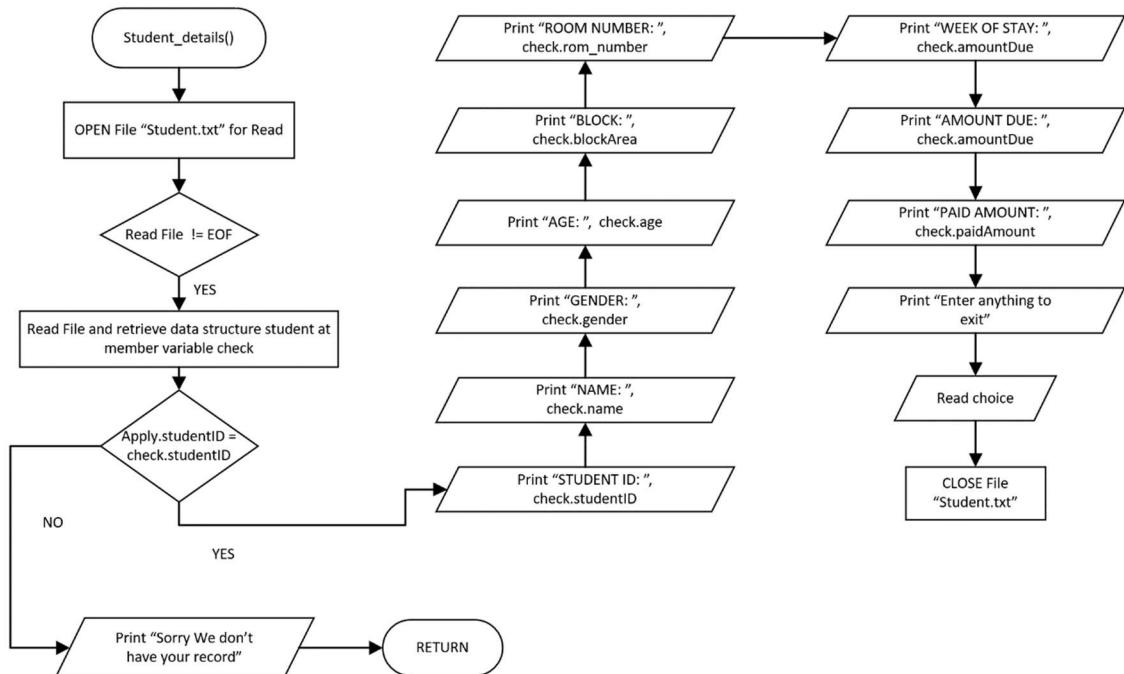


Figure 2.2.20 Student Details Function

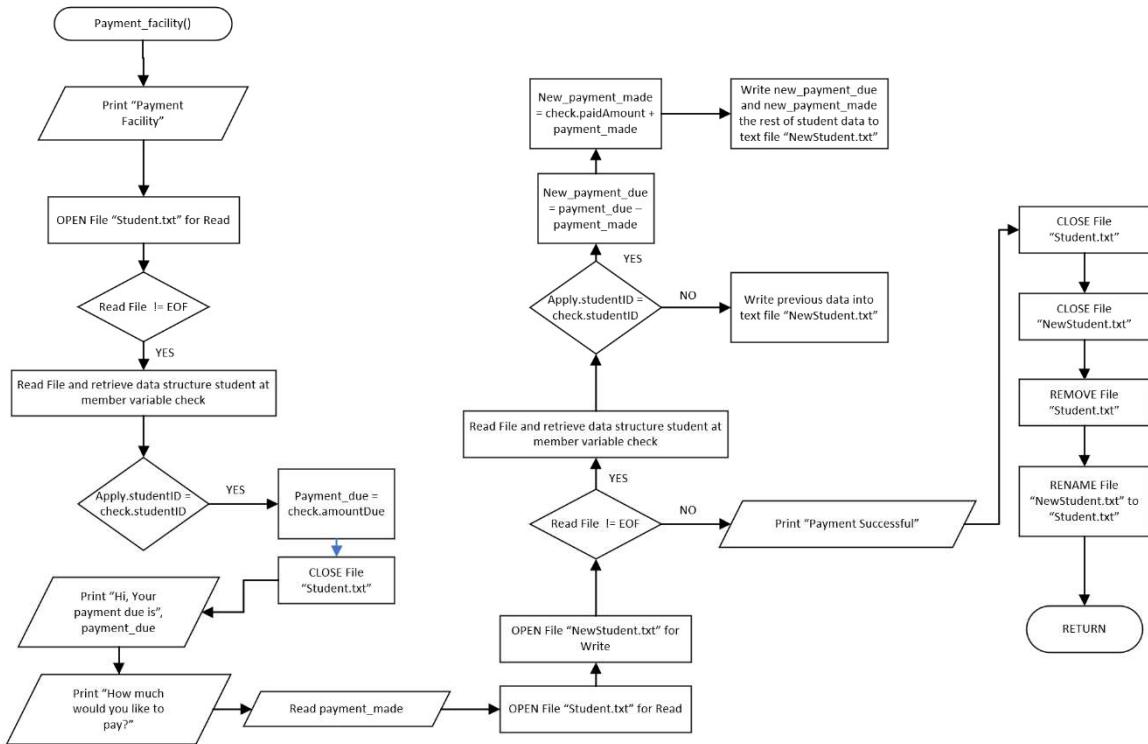


Figure 2.2.21 Payment Facility Function

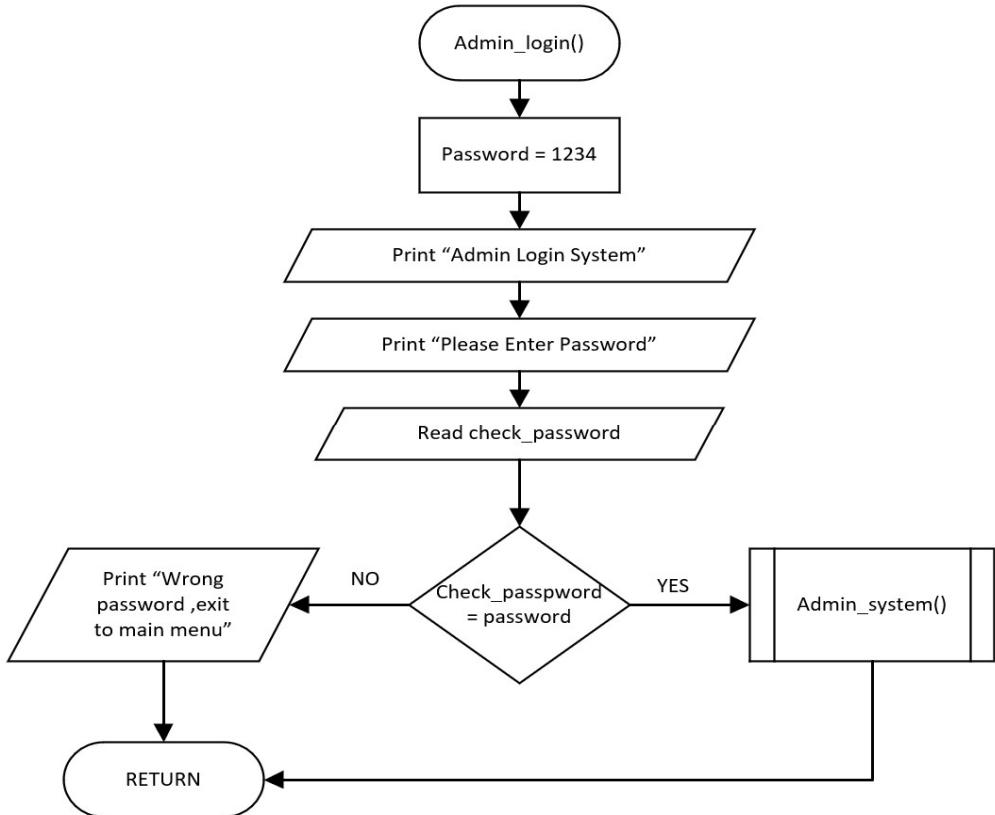


Figure 2.2.22 Admin Login Function

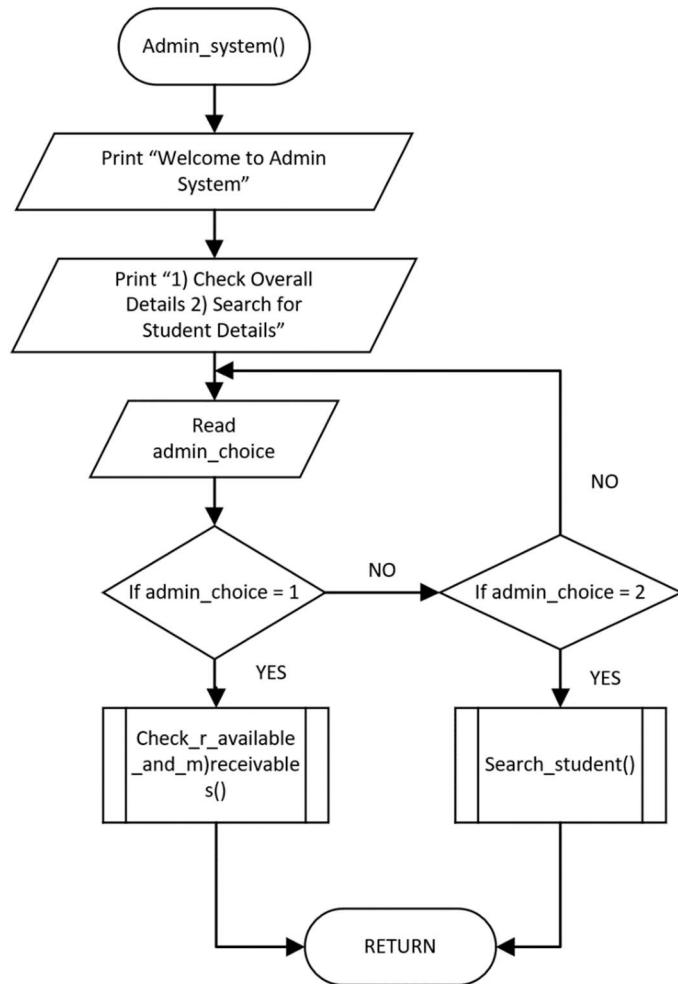


Figure 2.2.23 Admin System Function

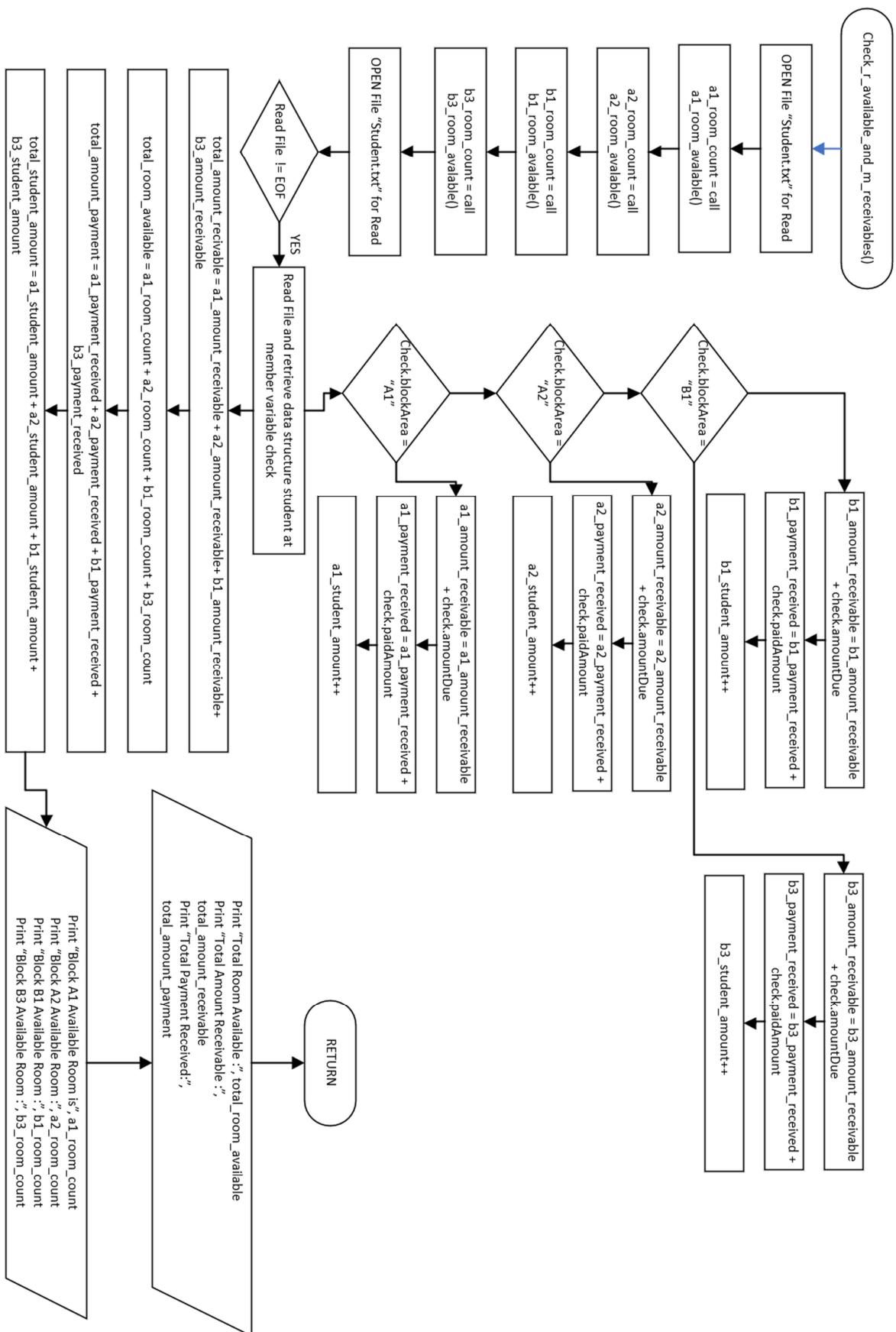


Figure 2.2.24 Check Room Availability and Money Receivable Function

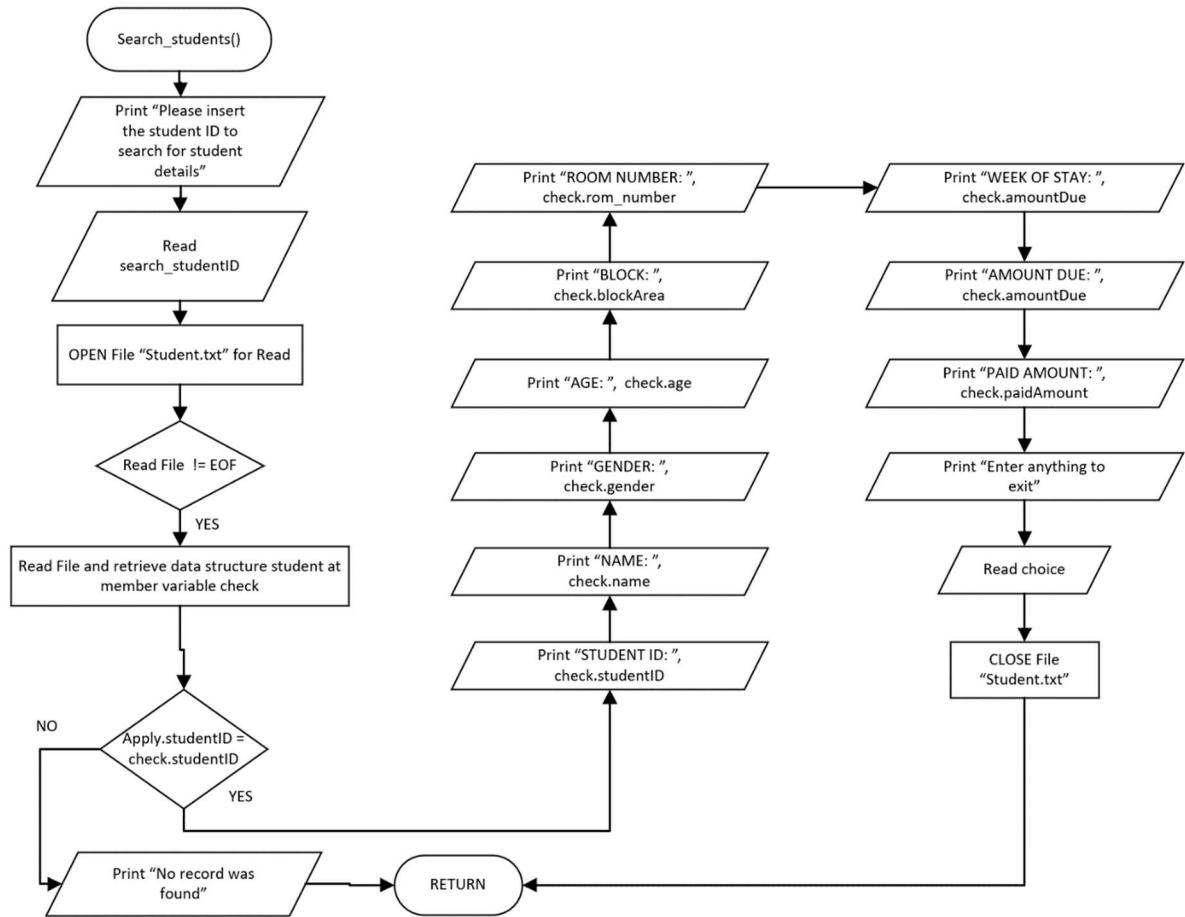


Figure 2.2.25 Search Student Function

3.0 | C Programming Concept

3.1 | Header File

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "total_function.h"
```

Figure 3.1 Header File

A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source file. There are two types of header files, the file that the programmer writes and the files that comes with the compiler. The header file that I use is “#include <stdio.h>”, “#include <string.h>”, “#include <stdlib.h>”.

3.2 | Data Structures

```
struct student {
    char studentID[10];
    char firstName[30];
    char lastName[30];
    char gender[2];
    char identification[15];
    int age;
    char phone_num[20];
    char email[50];
    char block_area[3];
    int room_num;
    int laundry_service;
    int gym_service;
    int week_stay;
    int amount_due;
    int paid_amount;
    int bed_num;
}apply, check;
```

Figure 3.2 Data Structure

Data structure that I applied inside the assignment is for student details, which is studentID, firstName, lastName, gender, identification, age, phone_num, email, block_area, room_num, laundry service, gym_service, week_stay, amount_due, paid_amount, bed_num. There are 2 data structure that I applied, one for apply, one for check.

3.3 | Function

```
/* CREATING FILE WHEN THERE IS NO INITIAL FILE*/
void check_file_created();
/* INITIALIZING ARRAY OF ROOM AND REASSIGN PREVIOUS BOOKING INTO ARRAY*/
void initialize_block();
void reassign_previous_booking();
/* INITIAL REGISTRATION AND BOOKING / CHOOSING ROOM */
int student_registration();
void male_booking_menu();
void female_booking_menu();
int gym_calculation();
int laundry_calculation();
int a1room_visuals();
int a2room_visuals();
int b1room_visuals();
int b3room_visuals();
/* STUDENT LOGIN TO CHANGE OR CANCEL ROOM, ADD SERVICE , CHECK DETAILS, MAKE PAYMENT AND EXIT TO MAIN MENU*/
int student_login();
void change_room();
void cancel_room();
void add_service();
int student_details();
void payment_facility();
/* ADMIN LOGIN, CHECK ROOM AVAILABLE, CHECK AMOUNT RECEIVABLE AND RELAVANT DETAILS*/
void admin_login();
void admin_system();
void check_r_available_and_m_receivables();
void check_block_student();
void search_student();
void check_room_availability();
/* COUNT ROOM AVAILABLE */
int a1_room_available();
int a2_room_available();
int b1_room_available();
int b3_room_available();
/* CALCULATE STUDENT'S PERSONAL DUE AMOUNT */
int personal_due(int rental, int gym, int laundry, int week_stay, int paid_amount);
/* EXIT PROGRAM */
void exitProgram();
```

Figure 3.1 Total Function

Functions are groups of statements that perform the same programmed task. I've used different function grouped based on their functionality which is student registration and booking, student login for changing details and check details, admin login and so on.

3.4 | Displaying Text

```
printf("\t*****\n");
printf("\t| WELCOME TO WISDOM COLLEGE HOSTEL MANAGEMENT SYSTEM |\n");
printf("\t*****\n\n");
printf("\tPLEASE PROCEED TO FOLLOWING CHOICES :\n");
printf("\t1) STUDENT REGISTER \n\t2) STUDENT LOGIN\n\t3) ADMIN LOGIN \n\t4) EXIT\n\n");
printf("\tPLEASE ENTER A CHOICE (1-4): ");
```

Figure 3.4 Displaying Text

Printf function is used like shown in figure 3.4 to display text. Printf function is included in the header file in compile under “#include <stdio.h>”.

3.5 | Reading Value

```
scanf("%d", &choice);
```

Figure 3.5 Reading Value

Scarf function is used to store a value, as shown in figure 3.5, scarf function will request an input of “%d” which is decimal to the variable choice. Scarf function is included in the header file in compile under “#include <stdio.h>”.

3.6 | Assigning Value to Variable

```
rental = 400;
```

Figure 3.6 Assigning Value to Variable

Assigning variable is used to initialize the value of the variable. Figure 3.6 assigns the value 400 into the variable rental. Before assigning value, the variable must be first declared into either int, float, string, character or so on.

3.7 | Comment

```
/* THIS IS THE MAIN SOURCE FILE AND THE MAIN FUNCTION */
```

Figure 3.7 Comment

Comment are used in programs to let other user understand their code, giving comments of what the function does. Comment are used by simply inserting based on the format /* (text) */. Figure 3.7 shows that this comment is to allow users to know it is the main function.

3.8 | If, Else Statement

```
if (strcmp(strupr(apply.studentID), check.studentID) == 0 && check.laundry_service == 0) {  
    printf("\tSUBSCRIPTION MENU :\n\n");  
    new_laundry = laundry_calculation();  
    if (check.block_area[0] == 'A' && check.block_area[1] == '1')  
        rental = 400;  
    else if (check.block_area[0] == 'A' && check.block_area[1] == '2')  
        rental = 200;  
    else if (check.block_area[0] == 'B' && check.block_area[1] == '1')  
        rental = 400;  
    else  
        rental = 150;
```

Figure 3.8 If, Else Statement

If else statement is used to check conditions, for this instance if condition is ‘A’ and ‘1’ then it proceeds to execute the statement below and else if another condition and else the remaining conditions.

3.9 | Switch, Case

```
switch (add_choice) {  
case 'L':  
case 'l':
```

Figure 3.9 Switch Case Statement

Switch statement is used to test a variable against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case. In this instance, switch is checking the variable add_choice if it is case ‘L’ or ‘l’ if proceeds to execute the statement below.

3.10 | While Loop

```
while (fscanf(block, "%s \t %s \t %s \t %s \t %s \t %d \t %s \t %s \t %s \t %d \t %d \t %d \t %d \t %d \t %d \n",strupr  
| if (strcmp(strupr(apply.studentID), check.studentID) == 0)  
|     fprintf(newBlock, "%s \t %s \t %s \t %s \t %d \t %s \t %s \t %s \t %s \t %d \n", s  
| else  
|     fprintf(newBlock, "%s \t %s \t %s \t %s \t %d \t %s \t %s \t %s \t %s \t %d \t %d \t %d \t %d \t %d \t %d \n", s  
| }
```

Figure 3.10 While Loop

While Loop will repeatedly execute a target statement if the set condition is true. Figure 3.10 has shown a while loop of while scanning file is not end of file its going to continuously execute until it reaches the end of the file.

3.11 | For Loop

```
for (i = 0; i < 10; i++) {  
    printf("\t");  
    for (j = 1; j < 11; j++) {  
        printf("A%.2d ", z++);  
        printf(" %d ", block_a1[x++]);  
    }  
    printf("\n\n");  
}
```

Figure 3.11 For Loop

For Loop is a statement that allows programmers to write a loop that suit the needs of executing a specific number of times. Figure 3.7 will run 100 times, because there's a for loop of 10 and a inside loop of 10. So, it will run 10 times x 10 times which is 100 times.

3.12 | Open / Close File

```
block = fopen("Student.txt", "r");  
fclose(block);
```

Figure 3.12 Open and Close File

To read, write and append information or data inside the file, file must be open and initialize for the usage either read, write or append. After opening the file, we must close the file after we finish using the file for read, write or append.

3.13 | Declaration Variable

```
int choice, check_register, check_login;
```

Figure 3.13 Declaration Variable

Variable can be declared as int, float, char and so on. Variable should be declared in a name that is easily understandable.

3.14 | Read File

```
block = fopen("Student.txt", "r");
```

Figure 3.14 Read File

When files are opened, we will need to specify the usage. In figure 3.14, “Student.txt” is opened for reading, so it will read the information or data inside the text file and no modification of the information is allowed while reading file.

3.15 | Write File

```
newBlock = fopen("NewStudent.txt", "w");
```

Figure 3.15 Write File

If file is specified to write, which is what is shown on figure 3.15. The program will overwrite or write on the “NewStudent.txt” file. If there is stored info inside “NewStudent.txt”, it is going to overwrite the whole file.

3.16 | Append File

```
block = fopen("Student.txt", "a+");
```

Figure 3.16 Append File

If file is specified to append, which is what is shown on figure 3.16. The program will append and add information inside “Student.txt”, while not overwriting the old info that is inside the file. So, the new information will be added inside the file and the old info will remain.

3.17 | Strcpy Function

```
strcpy(apply.block_area, "B3");
```

Figure 3.17 Strcpy Function

The function strcpy is a built-in function included in the header file “#include <string.h>” that is included in the compiler. Strcpy is a function that copies an array of character which is also string inside the variable. In figure 3.17 the string or character array of “B3” will be copied inside the variable apply. block_area. (Programiz, n.d.)

3.18 | Strcmp Function

```
if (strcmp(strupr(apply.studentID), check.studentID) == 0)
```

Figure 3.18 Strcmp Function

The function strcmp is a built-in function included in the header file “#include <string.h>” that is included in the compiler. Strcmp is a function that compares 2 string variables. In figure 3.18, the program compares string apply. studentID with check. studentID. If there is no difference, the output will be 0. (Tutorialspoint, n.d.)

3.19 | Go to

```
cancel_menu:  
goto cancel_menu;
```

Figure 3.19 Go to Function

The function goto is function that allows program to go back a certain point of the program. For this instance, in figure 3.19, cancel_menu: is created as a location, and if in a certain point I want to go back to the location and run the program from the location again, goto function will be used as shown in figure 3.19. (Singh, 2015)

3.20 | Array

```
int block_a1[100];  
int block_a2[100][2];  
int block_b1[100];  
int block_b3[100][3];
```

Figure 3.20 Array

Array is a collection of elements that could store multiple data. As shown in figure 3.20. block_a1 and block_b1 is going to have 100 elements, and block_a2, block_b3 are having 2 and 3 elements inside the 100 elements as represent to beds in the room.

3.21 | System (“CLS”)

```
system("CLS");
```

Figure 3.21 System ("CLS")

System("CLS") function is used to clear the printed text when running the program.

3.22 | Sleep Function

```
_sleep(2000);
```

Figure 3.22 Sleep Function

Sleep Function is called to make a delay to execute the next line of program. As shown above on figure 3.22, the syntax is `_sleep()` and the number inside the bracket represents milliseconds. So, Figure 3.22 will delay 2000 millisecond which is also 2 second.

3.23 | Strupr Function

```
strupr(check.studentID)
```

Figure 3.23 Strupr Function

Strupr function is used to make all string into upper case. Figure 3.23 converts all string inside check. studentID to upper case letters.

3.24 | Remove / Rename File

```
remove("Student.txt");  
rename("NewStudent.txt", "Student.txt");
```

Figure 3.24 Remove / Rename File

Remove function is used to delete a text file and rename function is used to rename file. As shown above in figure 3.24, the program will delete “Student.txt”, rename “NewStudent.txt” to “Student.txt”.

3.25 | Fscanf Function

Figure 23.25 Fscanf Function

Function `fscanf` reads formatted input from a string and will then assign the scanned value into the variable that is desired. (Tutorialspoint, n.d.)

4.0 | Additional Features

4.1 | Admin Login

```
void admin_login() {
    int admin_choice;
    char password[10] = "AbcD1234", check_password[10];
    admin_login_menu:
    system("CLS");
    printf("\t*****\n");
    printf("\t| ADMIN LOGIN SYSTEM |\n");
    printf("\t*****\n\n");
    printf("\tWOULD YOU LIKE TO: 1) PROCEED LOGIN 2) EXIT TO MAIN MENU\n");
    printf("\tWHAT WOULD YOU LIKE TO DO (1-2) : ");
    scanf("%d", &admin_choice);
    switch (admin_choice) {
        case 1:
            printf("\tPLEASE KEY IN THE PASSWORD : ");
            scanf("%s", check_password);
            if (strcmp(password, check_password) == 0) {
                printf("\n\n\t*** SUCCESSFULLY LOGIN INTO ADMIN SYSTEM ***");
                _sleep(2000);
                printf("\n\t*** REDIRECTING TO ADMIN SYSTEM! ***");
                _sleep(2000);
                system("CLS");
                admin_system();
            }
            else {
                printf("\t* SORRY WRONG PASSWORD, PLEASE ENTER CORRECT PASSWORD! *\n");
                _sleep(3000);
                printf("\n\t*** REDIRECTING TO ADMIN LOGIN SYSTEM ! ***");
                _sleep(2000);
                system("CLS");
                goto admin_login_menu;
            }
        break;
    }
}
```

Figure 4.1.1 Admin Login Source Code

```
*****  
| ADMIN LOGIN SYSTEM |  
*****  
  
WOULD YOU LIKE TO: 1) PROCEED LOGIN 2) EXIT TO MAIN MENU  
WHAT WOULD YOU LIKE TO DO (1-2) : 1  
PLEASE KEY IN THE PASSWORD : AbcD1234  
  
*** SUCCESSFULLY LOGIN INTO ADMIN SYSTEM ***
```

Figure 4.1.2 | Admin Login Sample Output

Figure 4.1.1 shows the source code of the password which is “AbcD1234”, if the password is correct, the program will proceed to redirect users to the admin system as shown in Figure 4.1.2.

4.2 | Choosing Room

```

int a1room_visuals() {
    room_selection:
    int i, j, z = 0, x = 0, room_num;
    system("CLS");
    printf("\t*****\n");
    printf("\t|          BLOCK A1          |\n");
    printf("\t*****\n");
    for (i = 0; i < 10; i++) {
        printf("\t");
        for (j = 1; j < 11; j++) {
            printf("A%.2d ", z++);
            printf(" %d ", block_a1[x++]);
        }
        printf("\n\n");
    }
    printf("\t*****\n");
    printf("\t|      0 MEANS AVAILABLE          1 MEANS OCCUPIED      |\n");
    printf("\t*****\n");
    printf("\tYOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): ");
    scanf("%d", &room_num);
    if (block_a1[room_num] == 1) {
        printf("\tSORRY, THIS ROOM HAS BEEN OCCUPIED, PLEASE CHOOSE ANOTHER ROOM");
        _sleep(2000);
        goto room_selection;
    }
    else if (block_a1[room_num] == 0)
        block_a1[room_num] = 1;
    else {
        printf("PLEASE ENTER A VALID OPTION !");
        goto room_selection;
    }
    return room_num;
}

```

Figure 4.2.1 Choose Room Source Code

```

*****
|          BLOCK A1          |
*****
A00  0  A01  0  A02  0  A03  0  A04  0  A05  0  A06  0  A07  0  A08  0  A09  0
A10  0  A11  0  A12  0  A13  0  A14  0  A15  0  A16  0  A17  0  A18  0  A19  0
A20  0  A21  0  A22  0  A23  0  A24  0  A25  0  A26  0  A27  0  A28  0  A29  0
A30  0  A31  0  A32  0  A33  0  A34  0  A35  0  A36  0  A37  0  A38  0  A39  0
A40  0  A41  0  A42  0  A43  0  A44  0  A45  0  A46  0  A47  0  A48  0  A49  0
A50  0  A51  0  A52  0  A53  0  A54  0  A55  0  A56  0  A57  0  A58  0  A59  0
A60  0  A61  0  A62  0  A63  0  A64  0  A65  0  A66  0  A67  0  A68  0  A69  0
A70  0  A71  0  A72  0  A73  0  A74  0  A75  0  A76  0  A77  0  A78  0  A79  0
A80  0  A81  0  A82  0  A83  0  A84  0  A85  0  A86  0  A87  0  A88  0  A89  0
A90  0  A91  0  A92  0  A93  0  A94  0  A95  0  A96  1  A97  0  A98  0  A99  0
*****
|      0 MEANS AVAILABLE          1 MEANS OCCUPIED      |
*****
YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): -

```

Figure 4.2.2 Choose Room Sample Output

Figure 4.2.1 shows the source code of choosing room. As shown in figure 4.2.2, the program will display all the available room as 0 and chosen room will be shown as 1. User can choose their desire room number if the room is available.

4.3 | Change Student Personal Details

Figure 4.3.1 Change Student Personal Details Source Code

***** HERE IS YOUR BOOKING DETAILS *****

STUDENT ID : TP041523
YOUR FIRST NAME : PEI
YOUR LAST NAME : XIAO
YOUR GENDER : F
YOUR IDENTIFICATION: 990415077524
YOUR AGE : 20
YOUR PHONE : 0124061252
YOUR EMAIL : pei_xiao99@gmail.com
YOUR BLOCK : B1
YOUR ROOM : 1
YOUR PAYMENT DUE : 860
YOUR PAYMENT MADE : 0

WOULD YOU LIKE TO MAKE ANY CHANGES TO YOUR DETAILS ? YES(Y) OR NO(N) :y
WHAT DETAILS WOULD YOU LIKE TO CHANGE? (1-7) :
1) STUDENT ID
2) FIRST NAME
3) LAST NAME
4) AGE
5) IDENTIFICATION
6) PHONE NUMBER
7) EMAIL
1
PLEASE INSERT THE CORRECT STUDENT ID (TP000000) :TP049952_

Figure 4.3.2 Change Personal Details Sample Output

TP041523	Pei	Xiaox	F	990415077524	20	0124061252	pei_xiaox99@gmail.com	B1	1	20	10
TP204357	Robert		Downey	M	AK729302	40	0174338694	iron_man99@gmail.com	A1	96	
TP038343	Teh	Peng	M	980213947204	21	0172843029	tehpeng_99@gmail.com	A2	0	20	10
TP038243	Thean	Tan	F	780216076784	41	0124168988	tanloothen_78@gmail.com	B3	0	0	0

Figure 4.3.3 File Information Before Change of Details

TP049952	Pei	Xiaox	F	990415077524	20	0124061252	pei_xiaox99@gmail.com	B1	1	20	10
TP204357	Robert		Downey	M	AK729302	40	0174338694	iron_man99@gmail.com	A1	96	
TP038343	Teh	Peng	M	980213947204	21	0172843029	tehpeng_99@gmail.com	A2	0	20	10
TP038243	Thean	Tan	F	780216076784	41	0124168988	tanloothen_78@gmail.com	B3	0	0	0

Figure 4.3.4 File Information After Change of Details

Figure 4.3.1 shows the source code of allowing students to change their personal details if they found that details are incorrect. As shown in Figure 4.3.2, once you enter y for making changes, the program will again request for another input from the user asking which details they would want to change from 1) student id to 7) email. For this instance, user input 1 which leads to changing their student ID, after input TP049952, as shown above, figure 4.3.3 is the one before the change. After the input, the program will then write the change inside the text file as shown in figure 4.3.4. So, the Student ID will change from TP041523 to TP049952 and it will be written inside the text file for future use.

4.4 | Show Percentage of Male and Female Hosteler

```

total_room_available = a1_room_count + a2_room_count + b1_room_count + b3_room_count;
total_amount_receivable = a1_amount_receivable + a2_amount_receivable + b1_amount_receivable + b3_amount_receivable;
total_amount_payment = a1_payment_received + a2_payment_received + b1_payment_received + b3_payment_received;
total_hosteler = a1_student_amount + a2_student_amount + b1_student_amount + b3_student_amount;
total_male_student = (a1_student_amount + a2_student_amount)/ (float)1;
total_female_student = (b1_student_amount + b3_student_amount) / (float)1;
male_percentage = 100 * (total_male_student / (float)total_hosteler);
female_percentage = 100 * (total_female_student / (float)total_hosteler);
check_menu:
printf("\t*****\n\t| CHECK MENU | \n\t*****\n");
printf("\t-----\n");
printf("\t| BLOCK | GENDER | AVAILABILITY | TYPE | AMOUNT RECEIVABLE | PAYMENT RECEIVED | STUDENT BOOKED AMOUNT | \n");
printf("\t-----\n");
printf("\t| A 1 | MALE | %3d | SINGLE | %5d | %5d | %3d | \n", a1_room_count,
printf("\t| A 2 | MALE | %3d | 2-BEDS | %5d | %5d | %3d | \n", a2_room_count,
printf("\t| B 1 | FEMALE | %3d | SINGLE | %5d | %5d | %3d | \n", b1_room_count,
printf("\t| B 3 | FEMALE | %3d | 3-BEDS | %5d | %5d | %3d | \n", b3_room_count,
printf("\t-----\n");
printf("\tTOTAL ROOM AVAILAIBLE: %d TOTAL AMOUNT RECEIVABLE: %d TOTAL PAYMENT RECEIVED: %d TOTAL HOSTELER: %d\n\n",total_room_available,
printf("\tPERCENTAGE OF MALE STUDENT: %.2f%% PERCENTAGE OF FEMALE STUDENTS: %.2f%% \n",male_percentage,female_percentage);

```

Figure 4.4.1 Calculation & Showing of Male & Female Hosteler Percentage Source Code

```

*****\n| CHECK MENU | \n*****\n-----\n| BLOCK | GENDER | AVAILABILITY | TYPE | AMOUNT RECEIVABLE | PAYMENT RECEIVED | STUDENT BOOKED AMOUNT |\n-----\n| A 1 | MALE | 94 | SINGLE | 9068 | 4102 | 6 |\n| A 2 | MALE | 97 | 2-BEDS | 5666 | 1054 | 7 |\n| B 1 | FEMALE | 96 | SINGLE | 9677 | 2133 | 4 |\n| B 3 | FEMALE | 99 | 3-BEDS | 2449 | 1011 | 4 |\n-----\nTOTAL ROOM AVAILAIBLE: 386 TOTAL AMOUNT RECEIVABLE: 26860 TOTAL PAYMENT RECEIVED: 8300 TOTAL HOSTELER: 21\nPERCENTAGE OF MALE STUDENT: 61.90% PERCENTAGE OF FEMALE STUDENTS: 38.10%\n1) CHECK ALL STUDENTS DETAILS BY BLOCK\n2) EXIT TO ADMIN SYSTEM MENU\nPLEASE ENTER A CHOICE (1-2) :\n
```

Figure 4.4.2 Percentage of Male & Female Hosteler Sample Output

Figure 4.4.1 shows the calculation and the source code of the check menu and as shown in figure 4.4.2, the program is going to show all the necessary details like room availability, total payment receivable, payment received and especially the percentage of male and female students as shown in figure 4.4.2, there is 61.9 % male students and 38.1 % female students.

5.0 | Test Specification Table

Function	Input	Expected Output	Given Input	Output	Results
Main Manu	1	Enter to Student Registration Menu	1	Enter Registration Menu	Pass
	2	Enter to Student Login Menu	2	Enter Student Login Menu	Pass
	3	Enter Admin Login Menu	3	Enter Admin Login Menu	Pass
			3	Redirect to Main Menu	Fail
	4	Exit Program	4	Exit Program	Pass
			4	Redirect to Main Menu	Fail
Student Registration Menu	Registered Student ID	Redirect Back to Main Menu	Registered Student ID	Proceeds to Register	Fail
		Proceeds to Register	Registered Student ID	Redirect to Main Menu	Pass
	New Student ID	Proceeds to Register	New Student ID	Redirect to Main Menu	Fail
			New Student ID	Proceeds to Register	Pass
		Registered Student ID	Registered Student ID	Proceeds to Register	Fail
Student Login Menu	Registered Student ID	Proceeds to Student Login System	Registered Student ID	Proceeds to Student Login System	Pass
			Registered Student ID	Redirect to Main Menu	Fail
		Redirect to Main Menu	New Student ID	Redirect to Main Menu	Pass

	New Student ID		New Student ID	Proceeds to Student Login System	Fail
Student Login System	1	Enter choice Change or Cancel Room	1	Returned to Student Login Menu	Fail
			1	Enter choice Change or Cancel Room	Pass
			1	Shows Booking Menu	Fail
			1	Redirect to Main Menu	Fail
	2	Shows Personal Details and ask for user if they want to change	2	Shows Personal Details and ask for user if they want to change	Pass
			2	Redirect to Main Menu	Fail
			2	Redirect to Admin Login Menu	Fail
	3	Direct to Payment Facility to make payment	3	Direct to Payment Facility to make payment	Pass
			3	Remains in Student Login Menu	Fail
	4	Exit to Main Menu	4	Exit Program	Fail
			4	Exit to Main Menu	Pass
			4	Redirects to Student Login Menu	Fail
Admin Login Menu	1	Ask for password Input	1	Ask for password Input	Pass
			1	Redirects to Admin System	Fail
			1	Exits to Main Menu	Fail
	Password	Redirects to Admin System	Password	Login into Admin System	Pass

			Password	States wrong password and request for password again	Fail
	2	Exit to Main Menu	2	Remain in Admin Login Menu	Fail
			2	Exit to Main Menu	Pass
Admin System	1	Shows overall details like total amount receivable and total students	1	No record found, and only displays 0	Fail
			1	Redirect to Student Search Engine	Fail
			1	Shows overall details like total amount receivable and total students	Pass
	2	Redirects to Student Search Engine	2	Shows overall details, total amount receivable and total amount received	Fail
			2	Redirects to Student Search Engine	Pass
	3	Ask user to choose block to view availability	3	Shows total percentage of male and female student	Fail
			3	Redirect to Main Menu	Fail
			3	Exit Program	Fail
			3	Ask user to choose block to view availability	Pass
	4	Exits to Main Menu	4	Exit to Main Menu	Pass
			4	Exit Program	Fail

Student Registration System	New Student ID	Proceed to Enter personal Details	New Student ID	Redirects to Main Menu because record was found	Fail
			New Student ID	Proceed to Enter personal Details	Pass
	Registered Student ID	Redirects to Main Menu	Registered Student ID	Redirects to Main Menu	Pass
			Registered Student ID	Proceeds to Enter personal details	Fail
Male / Female Booking Menu Choose Room	1 to 99	Booking Successful if room is available, unsuccessful booking if room is booked by other students	7	Successful Booking, and room is available	Pass
			8	Successful Booking, and room is available	Pass
			9	Successful Booking, and room is available	Pass
			11	Unsuccessful Booking because room is booked by other student	Pass
Male / Female Booking Menu Choose Bed	L, l	Left Bed is available successfully booked bed	L	Booking successful although it is booked	Fail
			I	Successfully booked available bed	Pass
	M, m	Middle Bed is available successfully booked bed	M	Booking fail although bed is available	Fail
			m	Successfully booked available bed	Pass
	R, r	Right Bed is available and successfully booked bed	R	Successfully booked available bed	Pass

6.0 | Sample Output:

6.1 | Hostel Management System Sample Output:

```
*****
| WELCOME TO WISDOM COLLEGE HOSTEL MANAGEMENT SYSTEM |
*****  
PLEASE PROCEED TO FOLLOWING CHOICES :  
1) STUDENT REGISTER  
2) STUDENT LOGIN  
3) ADMIN LOGIN  
4) EXIT  
PLEASE ENTER A CHOICE (1-4): -
```

Figure 6.1: Hostel Management System Main Menu

When program is first executed, the program will display the hostel management system main menu with 4 choices which is 1. Student register, 2. Student login, 3. Admin login, 4. Exit.

6.2 | Student Register Sample Output

```
*****
| WELCOME TO WISDOM COLLEGE HOSTEL MANAGEMENT SYSTEM |
*****  
PLEASE PROCEED TO FOLLOWING CHOICES :  
1) STUDENT REGISTER  
2) STUDENT LOGIN  
3) ADMIN LOGIN  
4) EXIT  
PLEASE ENTER A CHOICE (1-4): 1
```

Figure 6.2.1 If Enter Choice 1

```
*****
| STUDENT REGISTRATION SYSTEM |  
*****  
PLEASE ENTER YOUR STUDENT ID (TP000000) : -
```

Figure 6.2.2: If Enter Choice 1

After the input 1, the program will then direct students to the student registration system, for them to proceed register to book a hostel room. The program will then ask the user to input the student ID with the format TP000000.

6.2.1 | Record Was Found Sample Output

```
*****
|          STUDENT REGISTRATION SYSTEM          |
*****  
  
PLEASE ENTER YOUR STUDENT ID (TP000000) : TP049952  
WE HAVE FOUND A RECORD THAT YOU HAVE REGISTERED, PLEASE LOGIN !  
  
*****          RETURNING BACK TO MAIN MENU          *****
```

Figure 3.2.1.1 Found a record of register

After input the student ID, if the student ID was found registered, the program will then output “WE HAVE FOUND A RECORD THAT YOU HAVE REGISTERED, PLEASE LOGIN!”, and the program will redirect the users back to the Hostel Management Main Menu.

6.2.2 | Successful Register Sample Output

```
*****
|          STUDENT REGISTRATION SYSTEM          |
*****  
  
PLEASE ENTER YOUR STUDENT ID (TP000000) : TP049952  
PLEASE ENTER YOUR FIRST NAME : Jared  
PLEASE ENTER YOUR LAST NAME : Chan  
PLEASE ENTER YOUR GENDER MALE (M) OR FEMALE (F) : M  
PLEASE ENTER YOUR IC OR PASSPORT (990515729931) or (AK518233) :  
990423076521  
PLEASE ENTER YOUR AGE : 20  
PLEASE ENTER YOUR PHONE NUMBER : 0134020303  
PLEASE ENTER YOUR EMAIL ADDRESS : jaredchan0074@gmail.com  
  
*****          PLEASE WAIT FOR 4 SECONDS          *****  
  
*****          SUCCESS          *****  
          REDIRECTING TO BOOKING MENU          -
```

Figure 6.2.2.1 Successful Register

If the student ID was newly registered, the program will then ask for user to input their relevant details for this instance first name, last name, gender, identification, age, phone number and email address. The program will then redirect users to the booking menu.

6.3 | Booking Menu

6.3.1 | Male Booking Menu

***** MENU *****							
BLOCK	GENDER	AVAILABILITY	TYPE	RENTALS	LAUNDRY	GYM	
A 1	MALE	99	SINGLE	RM400	RM20	RM10	
A 2	MALE	100	2-BEDS	RM200	RM20	RM10	
A 3	MALE	UNAVAILABLE	3-BEDS	RM150	RM20	RM10	
A 4	MALE	UNAVAILABLE	4-BEDS	RM100	RM20	RM10	

1) BLOCK A1 , SINGLE BEDDED ROOM , RM400 WEEKLY
2) BLOCK A2 , DOUBLE BEDDED ROOM , RM200 WEEKLY
3) BLOCK A3 , THREE BEDDED ROOM , RM150 WEEKLY
4) BLOCK A4 , FOUR BEDDED ROOM , RM100 WEEKLY

PLEASE CHOOSE THE BLOCK YOU DESIRE (1 - 4):

Figure 6.3.1.1: Male Booking Menu

After successfully registered, if student is a male, program will redirect to the male booking menu as shown in the figure above, in a clear glance the room type, block, rentals, services and room availability. Program will then request an input of 1 to 4 from user.

6.3.2 | Female Booking Menu

***** MENU *****							
BLOCK	GENDER	AVAILABILITY	TYPE	RENTALS	LAUNDRY	GYM	
B 1	FEMALE	99	SINGLE	RM400	RM20	RM10	
B 2	FEMALE	UNAVAILABLE	2-BEDS	RM200	RM20	RM10	
B 3	FEMALE	100	3-BEDS	RM150	RM20	RM10	
B 4	FEMALE	UNAVAILABLE	4-BEDS	RM100	RM20	RM10	

1) BLOCK B1 , SINGLE BEDDED ROOM , RM400 WEEKLY
2) BLOCK B2 , DOUBLE BEDDED ROOM , RM200 WEEKLY
3) BLOCK B3 , THREE BEDDED ROOM , RM150 WEEKLY
4) BLOCK B4 , FOUR BEDDED ROOM , RM100 WEEKLY

PLEASE CHOOSE THE BLOCK YOU DESIRE (1 - 4): ■

Figure 6.3.2.1 Female Booking Menu

After successfully registered, if student is a female, program will redirect to the female booking menu as shown in the figure above, in a clear glance the room type, block, rentals, services and room availability. Program will then request an input of 1 to 4 from user.

6.4 | Booking Room

6.4.1 | Male Block A1 Booking

BLOCK A1																													
A00	0	A01	0	A02	0	A03	0	A04	0	A05	0	A06	0	A07	0	A08	0	A09	0										
A10	0	A11	0	A12	0	A13	0	A14	0	A15	0	A16	0	A17	0	A18	0	A19	0										
A20	0	A21	0	A22	0	A23	0	A24	0	A25	0	A26	0	A27	0	A28	0	A29	0										
A30	0	A31	0	A32	0	A33	0	A34	0	A35	0	A36	0	A37	0	A38	0	A39	0										
A40	0	A41	0	A42	0	A43	0	A44	0	A45	0	A46	0	A47	0	A48	0	A49	0										
A50	0	A51	0	A52	0	A53	0	A54	0	A55	0	A56	0	A57	0	A58	0	A59	0										
A60	0	A61	0	A62	0	A63	0	A64	0	A65	0	A66	0	A67	0	A68	0	A69	0										
A70	0	A71	0	A72	0	A73	0	A74	0	A75	0	A76	0	A77	0	A78	0	A79	0										
A80	0	A81	0	A82	0	A83	0	A84	0	A85	0	A86	0	A87	0	A88	0	A89	0										
A90	0	A91	0	A92	0	A93	0	A94	0	A95	0	A96	1	A97	0	A98	0	A99	0										

0 MEANS AVAILABLE										1 MEANS OCCUPIED																			

YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99):																													

Figure 6.4.1.1: Block A1 Booking

If user chooses Block A1 by input 1, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it's occupied by another student. The program will then request for an input for the room they choose.

6.4.2 | Male Block A2 Booking

BLOCK A2									
A00	(0,0)	A01	(0,0)	A02	(0,0)	A03	(0,0)	A04	(0,0)
A05	(0,0)	A06	(0,0)	A07	(0,0)	A08	(0,0)	A09	(0,0)
A10	(0,0)	A11	(0,0)	A12	(0,0)	A13	(0,0)	A14	(0,0)
A15	(0,0)	A16	(0,0)	A17	(0,0)	A18	(0,0)	A19	(0,0)
A20	(0,0)	A21	(0,0)	A22	(0,0)	A23	(0,0)	A24	(0,0)
A25	(0,0)	A26	(0,0)	A27	(0,0)	A28	(0,0)	A29	(0,0)
A30	(0,0)	A31	(0,0)	A32	(0,0)	A33	(0,0)	A34	(0,0)
A35	(0,0)	A36	(0,0)	A37	(0,0)	A38	(0,0)	A39	(0,0)
A40	(0,0)	A41	(0,0)	A42	(0,0)	A43	(0,0)	A44	(0,0)
A45	(0,0)	A46	(0,0)	A47	(0,0)	A48	(0,0)	A49	(0,0)
A50	(0,0)	A51	(0,0)	A52	(0,0)	A53	(0,0)	A54	(0,0)
A55	(0,0)	A56	(0,0)	A57	(0,0)	A58	(0,0)	A59	(0,0)
A60	(0,0)	A61	(0,0)	A62	(0,0)	A63	(0,0)	A64	(0,0)
A65	(0,0)	A66	(0,0)	A67	(0,0)	A68	(0,0)	A69	(0,0)
A70	(0,0)	A71	(0,0)	A72	(0,0)	A73	(0,0)	A74	(0,0)
A75	(0,0)	A76	(0,0)	A77	(0,0)	A78	(0,0)	A79	(0,0)
A80	(0,0)	A81	(0,0)	A82	(0,0)	A83	(0,0)	A84	(0,0)
A85	(0,0)	A86	(0,0)	A87	(0,0)	A88	(0,0)	A89	(0,0)
A90	(0,0)	A91	(0,0)	A92	(0,0)	A93	(0,0)	A94	(0,0)
A95	(0,0)	A96	(0,0)	A97	(0,0)	A98	(0,0)	A99	(0,0)

Figure 6.4.2.1 Block A2 Booking

If user chooses Block A2 by input 2, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it's occupied by another student. The program will then request for an input for the room they choose. As shown in the male booking menu, A2 is a double bedroom. So as figure shown above there is two 0s for users to choose. The user will be asked for choosing a room number first and later program will request user to pick the left bed or right bed.

6.4.3 | Female Block B1 Booking

BLOCK B1											

B00	1	B01	0	B02	0	B03	0	B04	0	B05	0
B06	0	B07	0	B08	0	B09	0	B10	0	B11	0
B12	0	B13	0	B14	0	B15	0	B16	0	B17	0
B18	0	B19	0	B20	0	B21	0	B22	0	B23	0
B24	0	B25	0	B26	0	B27	0	B28	0	B29	0
B30	0	B31	0	B32	0	B33	0	B34	0	B35	0
B36	0	B37	0	B38	0	B39	0	B40	0	B41	0
B42	0	B43	0	B44	0	B45	0	B46	0	B47	0
B48	0	B49	0	B50	0	B51	0	B52	0	B53	0
B54	0	B55	0	B56	0	B57	0	B58	0	B59	0
B60	0	B61	0	B62	0	B63	0	B64	0	B65	0
B66	0	B67	0	B68	0	B69	0	B70	0	B71	0
B72	0	B73	0	B74	0	B75	0	B76	0	B77	0
B78	0	B79	0	B80	0	B81	0	B82	0	B83	0
B84	0	B85	0	B86	0	B87	0	B88	0	B89	0
B90	0	B91	0	B92	0	B93	0	B94	0	B95	0
B96	0	B97	0	B98	0	B99	0	*****			
0 MEANS AVAILABLE			1 MEANS OCCUPIED						*****		
YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99):											

Figure 6.4.3.1 Block B1 Booking

If user chooses Block B1 by input 1, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it's occupied by another student. The program will then request for an input for the room they choose.

6.4.4 | Female Block B3 Booking

Figure 6.4.4.1 Block B3 Booking

If user chooses Block B3 by input 3, program will then show all the rooms that are available as shown above. 0 meaning is available and 1 means it's occupied by another student. The program will then request for an input for the room they choose. As shown in the male booking menu, A2 is a double bedroom. So as figure shown above there is two 0s for users to choose. The user will be asked for choosing a room number first and later program will request user to pick the left bed or right bed.

6.4.5 | Unavailable Block Bookings

***** MENU *****							
BLOCK	GENDER	AVAILABILITY	TYPE	RENTALS	LAUNDRY	GYM	
A 1	MALE	98	SINGLE	RM400	RM20	RM10	
A 2	MALE	99	2-BEDS	RM200	RM20	RM10	
A 3	MALE	UNAVAILABLE	3-BEDS	RM150	RM20	RM10	
A 4	MALE	UNAVAILABLE	4-BEDS	RM100	RM20	RM10	

1) BLOCK A1 , SINGLE BEDDED ROOM , RM400 WEEKLY
 2) BLOCK A2 , DOUBLE BEDDED ROOM , RM200 WEEKLY
 3) BLOCK A3 , THREE BEDDED ROOM , RM150 WEEKLY
 4) BLOCK A4 , FOUR BEDDED ROOM , RM100 WEEKLY

PLEASE CHOOSE THE BLOCK YOU WANT TO CHANGE (1 - 4): 3
 SORRY TO INFORM YOU THAT, BLOCK A3 AND A4 IS TEMPORARILY UNAVAILABLE, PLEASE CHOOSE ANOTHER BLOCK

Figure 6.4.5.1 Unavailable Block in Male Booking Menu

***** MENU *****							
BLOCK	GENDER	AVAILABILITY	TYPE	RENTALS	LAUNDRY	GYM	
B 1	FEMALE	99	SINGLE	RM400	RM20	RM10	
B 2	FEMALE	UNAVAILABLE	2-BEDS	RM200	RM20	RM10	
B 3	FEMALE	100	3-BEDS	RM150	RM20	RM10	
B 4	FEMALE	UNAVAILABLE	4-BEDS	RM100	RM20	RM10	

1) BLOCK B1 , SINGLE BEDDED ROOM , RM400 WEEKLY
 2) BLOCK B2 , DOUBLE BEDDED ROOM , RM200 WEEKLY
 3) BLOCK B3 , THREE BEDDED ROOM , RM150 WEEKLY
 4) BLOCK B4 , FOUR BEDDED ROOM , RM100 WEEKLY

PLEASE CHOOSE THE BLOCK YOU DESIRE (1 - 4): 2
 SORRY TO INFORM YOU THAT, BLOCK B2 AND B4 IS TEMPORARILY UNAVAILABLE, PLEASE CHOOSE ANOTHER BLOCK

Figure 6.4.5.2 Unavailable Block in Female Booking Menu

Currently, Block A3 & Block A4 in Male's booking menu is now under construction. Block B2 & B4 in Female's booking menu is now under construction. If user inputs either A3, A4, B2 or B4. The program will state that the block is now unavailable and will request another valid input from the user to repick their blocks again.

6.5 | Choosing Room Number

6.5.1 | Room Number Chosen is Occupied

BLOCK A1																													
A00	0	A01	0	A02	0	A03	0	A04	0	A05	0	A06	0	A07	0	A08	0	A09	0										
A10	0	A11	0	A12	0	A13	0	A14	0	A15	0	A16	0	A17	0	A18	0	A19	0										
A20	0	A21	0	A22	0	A23	0	A24	0	A25	0	A26	0	A27	0	A28	0	A29	0										
A30	0	A31	0	A32	0	A33	0	A34	0	A35	0	A36	0	A37	0	A38	0	A39	0										
A40	0	A41	0	A42	0	A43	0	A44	0	A45	0	A46	0	A47	0	A48	0	A49	0										
A50	0	A51	0	A52	0	A53	0	A54	0	A55	0	A56	0	A57	0	A58	0	A59	0										
A60	0	A61	0	A62	0	A63	0	A64	0	A65	0	A66	0	A67	0	A68	0	A69	0										
A70	0	A71	0	A72	0	A73	0	A74	0	A75	0	A76	0	A77	0	A78	0	A79	0										
A80	0	A81	0	A82	0	A83	0	A84	0	A85	0	A86	0	A87	0	A88	0	A89	0										
A90	0	A91	0	A92	0	A93	0	A94	0	A95	0	A96	1	A97	0	A98	0	A99	0										

0 MEANS AVAILABLE										1 MEANS OCCUPIED																			

YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): 96																													
SORRY, THIS ROOM HAS BEEN OCCUPIED, PLEASE CHOOSE ANOTHER ROOM.																													

Figure 6.5.1.1 Room is Occupied

If user inputs a room number that is occupied, in this instance is A96, the program will that output “SORRY, THIS ROOM HAS BEEN OCCUPIED, PLEASE CHOOSE ANOTHER ROOM.” and request the input again from the user.

6.5.2 | Room Number Chosen is Available

BLOCK A1																													
A00	0	A01	0	A02	0	A03	0	A04	0	A05	0	A06	0	A07	0	A08	0	A09	0										
A10	0	A11	0	A12	0	A13	0	A14	0	A15	0	A16	0	A17	0	A18	0	A19	0										
A20	0	A21	0	A22	0	A23	0	A24	0	A25	0	A26	0	A27	0	A28	0	A29	0										
A30	0	A31	0	A32	0	A33	0	A34	0	A35	0	A36	0	A37	0	A38	0	A39	0										
A40	0	A41	0	A42	0	A43	0	A44	0	A45	0	A46	0	A47	0	A48	0	A49	0										
A50	0	A51	0	A52	0	A53	0	A54	0	A55	0	A56	0	A57	0	A58	0	A59	0										
A60	0	A61	0	A62	0	A63	0	A64	0	A65	0	A66	0	A67	0	A68	0	A69	0										
A70	0	A71	0	A72	0	A73	0	A74	0	A75	0	A76	0	A77	0	A78	0	A79	0										
A80	0	A81	0	A82	0	A83	0	A84	0	A85	0	A86	0	A87	0	A88	0	A89	0										
A90	0	A91	0	A92	0	A93	0	A94	0	A95	0	A96	1	A97	0	A98	0	A99	0										

0 MEANS AVAILABLE										1 MEANS OCCUPIED																			

YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): 2																													

Figure 6.5.2.1 Room is Available

If user inputs a room number that is available, the program will then redirect the user to the laundry subscription menu.

6.5.3 | Choosing Beds in a room

```
A90  (0,0)  A91  (0,0)  A92  (0,0)  A93  (0,0)  A94  (0,0)  
A95  (0,0)  A96  (0,0)  A97  (0,0)  A98  (0,0)  A99  (0,0)  
*****  
| 0 MEANS AVAILABLE 1 MEANS OCCUPIED |  
*****  
YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): 0  
PLEASE CHOOSE A BED LEFT(L) OR RIGHT(R) ?: L
```

Figure 6.5.3.1 Choosing Bed in Block A2

```
B85  (0,0,0)  B86  (0,0,0)  B87  (0,0,0)  B88  (0,0,0)  B89  (0,0,0)  
B90  (0,0,0)  B91  (0,0,0)  B92  (0,0,0)  B93  (0,0,0)  B94  (0,0,0)  
B95  (0,0,0)  B96  (0,0,0)  B97  (0,0,0)  B98  (0,0,0)  B99  (0,0,0)  
*****  
| 0 MEANS AVAILABLE 1 MEANS OCCUPIED |  
*****  
YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): 0  
PLEASE CHOOSE A BED LEFT(L) OR MIDDLE(M) OR RIGHT(R) ?: M
```

Figure 6.5.3.2 Choosing bed in Block B3

Choosing bed after choosing room is for both Block A2 and B3, as block A2 is a double bedroom and B3 is 3-bedded room, after choosing the room, the program will then request the user to input either L, M or R, which represents Left, Middle and Right. In block A2 is L and R, in block B3 is L, M and R. After user chooses their bed, program will then redirect users to the laundry subscription menu.

6.5.4 | Bed Chosen is Occupied

```
B85  (0,0,0)  B86  (0,0,0)  B87  (0,0,0)  B88  (0,0,0)  B89  (0,0,0)  
B90  (0,0,0)  B91  (0,0,0)  B92  (0,0,0)  B93  (0,0,0)  B94  (0,0,0)  
B95  (0,0,0)  B96  (0,0,0)  B97  (0,0,0)  B98  (0,0,0)  B99  (0,0,0)  
*****  
| 0 MEANS AVAILABLE 1 MEANS OCCUPIED |  
*****  
YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): 0  
PLEASE CHOOSE A BED LEFT(L) OR MIDDLE(M) OR RIGHT(R) ?: L  
SORRY, THIS BED HAS BEEN OCCUPIED, PLEASE CHOOSE ANOTHER BED  
PLEASE CHOOSE A BED LEFT(L) OR MIDDLE(M) OR RIGHT(R) ?:
```

Figure 6.5.4.1 Bed Chosen is Occupied

If the bed that has been chosen by the user is occupied. The program will output “SORRY, THIS BED HAS BEEN OCCUPIED, PLEASE CHOOSE ANOTHER BED”, and proceeds to request another input from the user.

6.6 | Service Subscription Menu

6.6.1 | Laundry Subscription Menu

```
*****  
|       LAUNDRY    RM20    |  
*****  
  
WOULD YOU LIKE TO SUBSCRIBE TO LAUNDRY EXPENSES ?  
1) YES, I WOULD LIKE TO  
2) NO , I WOULD NOT  
PLEASE ENTER YOUR CHOICE (1 - 2):
```

Figure 6.6.1.1 Laundry Subscription

After being redirected from choosing the room number. The program will ask user if they want to add on laundry subscription and will request for an input of either 1. Yes or 2. No. After user input, the program will then show gym subscription menu.

6.6.2 | Gym Subscription Menu

```
*****  
|       LAUNDRY    RM20    |  
*****  
  
WOULD YOU LIKE TO SUBSCRIBE TO LAUNDRY EXPENSES ?  
1) YES, I WOULD LIKE TO  
2) NO , I WOULD NOT  
PLEASE ENTER YOUR CHOICE (1 - 2): 1  
*****  
|       GYM       RM10    |  
*****  
  
WOULD YOU LIKE TO SUBSCRIBE TO GYM SERVICE ?  
1) YES, I WOULD LIKE TO  
2) NO , I WOULD NOT  
PLEASE ENTER YOUR CHOICE (1 - 2): 2
```

Figure 6.6.2.1: Gym Subscription

After being redirected from laundry subscription. The program will ask user if they want to add on gym subscription and will request for an input of either 1. Yes or 2. No.

6.6.3 | Week of Stay Given & Successful Booking

```
*****
|       LAUNDRY    RM20    |
*****  
  
WOULD YOU LIKE TO SUBSCRIBE TO LAUNDRY EXPENSES ?  
1) YES, I WOULD LIKE TO  
2) NO , I WOULD NOT  
PLEASE ENTER YOUR CHOICE (1 - 2): 1  
*****  
|       GYM      RM10    |
*****  
  
WOULD YOU LIKE TO SUBSCRIBE TO GYM SERVICE ?  
1) YES, I WOULD LIKE TO  
2) NO , I WOULD NOT  
PLEASE ENTER YOUR CHOICE (1 - 2): 2  
HOW MANY WEEKS WOULD YOU LIKE TO STAY?5  
  
*****      CONGRATULATIONS YOU HAVE SUCCESSFULLY REGISTERED      *****  
*****      PLEASE PROCEED TO LOGIN      *****
```

Figure 6.6.3.1 Given Week of Stay & Successful Booking

After gym subscription, the program will then ask the user to input the number of weeks they want to stay at the hostel. As shown above, after input the number of weeks user wanted to stay the program will then congratulate the user for successful booking and register thus output “PLEASE PROCEED TO LOGIN” to guide user to proceed to student login on the hostel management main menu and the program will redirect the user to the hostel management main menu.

6.7 | Student Login

```
*****
| WELCOME TO WISDOM COLLEGE HOSTEL MANAGEMENT SYSTEM |
*****  
  
PLEASE PROCEED TO FOLLOWING CHOICES :  
1) STUDENT REGISTER  
2) STUDENT LOGIN  
3) ADMIN LOGIN  
4) EXIT  
  
PLEASE ENTER A CHOICE (1-4): 2
```

Figure 6.7.1 If Enter Choice 2

```
*****
| STUDENT LOGIN SYSTEM |
*****  
  
PLEASE INSERT YOUR STUDENT ID (TP000000) : ■
```

Figure 6.7.2 If Enter Choice 2

After the input 2, the program will then direct students to the student login system, for them to proceed login to check and change details as well as make payment. The program will then ask the user to input the student ID with the format TP000000.

6.7.1 | No Record of Registration Found

```
*****
|           STUDENT LOGIN SYSTEM           |
*****
PLEASE INSERT YOUR STUDENT ID (TP000000) : 2

*
* SORRY WE DON'T HAVE YOUR RECORD, PLEASE REGISTER! *
****          RETURNING BACK TO MAIN MENU          ****
```

Figure 6.7.1.1 If No Record of Registration Found

As shown in figure 6.7.1.1, If student inputs a student ID that is has not been registered, the system will then output a line of words, requesting unregistered user to register, and then the program will redirect back to hotel management system main menu.

6.7.2 | Record Found & Successful Login

```
*****
|           STUDENT LOGIN SYSTEM           |
*****
PLEASE INSERT YOUR STUDENT ID (TP000000) : TP049952

HI, JARED WELCOME BACK! WE WILL REDIRECT YOU TO THE LOGIN SYSTEM !

REDIRECTING TO LOGIN SYSTEM
```

Figure 6.7.2.1 Record Found & Successful Login

Figure 6.7.2.1 shows that if a record was found, the program will then say hi to the student and the program will then redirect the user to the login menu.

6.8 | Student Login Menu

```
*****  
|           WELCOME TO STUDENT LOGIN SYSTEM           |  
*****  
  
WHAT WOULD YOU LIKE TO DO?  
  
1) CHANGE / CANCEL ROOM  
2) CHECK / CHANGE PERSONAL DETAILS  
3) MAKE PAYMENT  
4) EXIT TO HOSTEL MANAGEMENT SYSTEM  
  
PLEASE ENTER A CHOICE (1-4):
```

Figure 6.8.1 Student Login Menu

As shown in figure 6.8.1, after being redirected from Student Login System, the program will then request another input from the user. Input 1 for change or cancel room, 2. Check and change personal details, 3. Make payment, 4 will be exit to the hostel management main menu.

6.9 | Change/ Cancel Room & Subscribe/ Unsubscribe Laundry or Gym Service

```
*****  
|           WELCOME TO STUDENT LOGIN SYSTEM           |  
*****  
  
WHAT WOULD YOU LIKE TO DO?  
  
1) CHANGE / CANCEL ROOM  
2) CHECK / CHANGE PERSONAL DETAILS  
3) MAKE PAYMENT  
4) EXIT TO HOSTEL MANAGEMENT SYSTEM  
  
PLEASE ENTER A CHOICE (1-4): 1  
WOULD YOU LIKE TO :  
1) CHANGE ROOM  
2) CANCEL ROOM  
3) CHANGE LAUNDRY OR GYM SERVICE : -
```

Figure 6.9.1 If Choice 1 Change / Cancel Room

If choice 1 is the given input, then the program will ask for another user input for either 1 change room 2 cancel room 3 change laundry or gym service as shown in figure 6.9.1.

6.9.1 | Change Room

```
*****
|           WELCOME TO STUDENT LOGIN SYSTEM           |
*****
WHAT WOULD YOU LIKE TO DO?

1) CHANGE / CANCEL ROOM
2) CHECK / CHANGE PERSONAL DETAILS
3) MAKE PAYMENT
4) EXIT TO HOSTEL MANAGEMENT SYSTEM

PLEASE ENTER A CHOICE (1-4): 1
WOULD YOU LIKE TO :
1) CHANGE ROOM
2) CANCEL ROOM
3) CHANGE LAUNDRY OR GYM SERVICE : 1
```

Figure 6.9.1.1 Choice 1 Change Room

```
*****
|           MENU           |
*****
-----|-----|-----|-----|-----|-----|-----|-----|-----|
| BLOCK | GENDER | AVAILABILITY | TYPE | RENTALS | LAUNDRY | GYM | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| A 1  | MALE   | 94           | SINGLE | RM400   | RM20   | RM10 |
| A 2  | MALE   | 97           | 2-BEDS | RM200   | RM20   | RM10 |
| A 3  | MALE   | UNAVAILABLE | 3-BEDS | RM150   | RM20   | RM10 |
| A 4  | MALE   | UNAVAILABLE | 4-BEDS | RM100   | RM20   | RM10 |
-----|-----|-----|-----|-----|-----|-----|-----|-----|
1) BLOCK A1 , SINGLE BEDDED ROOM , RM400 WEEKLY
2) BLOCK A2 , DOUBLE BEDDED ROOM , RM200 WEEKLY
3) BLOCK A3 , THREE BEDDED ROOM , RM150 WEEKLY
4) BLOCK A4 , FOUR BEDDED ROOM , RM100 WEEKLY

PLEASE CHOOSE THE BLOCK YOU WANT TO CHANGE (1 - 4):
```

Figure 6.9.1.2 After Insert 1 Redirect to Booking Menu

After choosing choice 1 which is change room, the program will then redirect user to the booking menu like when they have successfully registered to change another room or change block. The menu showed in figure 6.9.1.2 is the male booking menu, because when student login into the system, the program read the information from text file and realize student is a male, so the program will redirect this student to male booking menu.

BLOCK A1																													
A00	0	A01	1	A02	0	A03	0	A04	1	A05	0	A06	0	A07	0	A08	0	A09	0										
A10	0	A11	0	A12	0	A13	0	A14	0	A15	0	A16	0	A17	0	A18	0	A19	0										
A20	0	A21	0	A22	1	A23	0	A24	0	A25	0	A26	0	A27	0	A28	0	A29	0										
A30	0	A31	0	A32	0	A33	0	A34	0	A35	0	A36	0	A37	0	A38	0	A39	0										
A40	0	A41	0	A42	0	A43	0	A44	0	A45	0	A46	0	A47	0	A48	0	A49	0										
A50	0	A51	0	A52	0	A53	0	A54	0	A55	0	A56	1	A57	0	A58	0	A59	0										
A60	0	A61	0	A62	0	A63	0	A64	0	A65	0	A66	0	A67	0	A68	0	A69	0										
A70	0	A71	0	A72	0	A73	0	A74	0	A75	0	A76	0	A77	0	A78	0	A79	0										
A80	0	A81	0	A82	0	A83	0	A84	1	A85	0	A86	0	A87	0	A88	0	A89	0										
A90	0	A91	0	A92	0	A93	0	A94	0	A95	0	A96	1	A97	0	A98	0	A99	0										

0 MEANS AVAILABLE										1 MEANS OCCUPIED																			

YOU CAN NOW CHOOSE AVAILABLE ROOMS (0-99): 3																													

Figure 6.9.1.3 Choose Room to Change

```
*****  
| LAUNDRY RM20 |  
*****  
  
WOULD YOU LIKE TO SUBSCRIBE TO LAUNDRY EXPENSES ?  
1) YES, I WOULD LIKE TO  
2) NO , I WOULD NOT  
PLEASE ENTER YOUR CHOICE (1 - 2): 1  
*****  
| GYM RM10 |  
*****  
  
WOULD YOU LIKE TO SUBSCRIBE TO GYM SERVICE ?  
1) YES, I WOULD LIKE TO  
2) NO , I WOULD NOT  
PLEASE ENTER YOUR CHOICE (1 - 2): 2  
HOW MANY WEEKS WOULD YOU LIKE TO STAY?5  
  
***** CONGRATULATIONS YOU HAVE SUCCESSFULLY CHANGED *****  
***** PLEASE LOGIN AGAIN *****  
PLEASE NOTE THAT A FINE OF RM 200 WILL BE CHARGED, PLEASE PAY THE ADMIN !
```

Figure 4.9.1.4 Booking New Room

As shown in figure 6.9.1.3 and figure 3.9.1.4, the process of booking a new room is the same as normal booking, but after everything is successfully, student will be asked to pay a fine of RM200 to the admin.

6.9.2 | Cancel Room

```
*****
|           WELCOME TO STUDENT LOGIN SYSTEM           |
*****
WHAT WOULD YOU LIKE TO DO?

1) CHANGE / CANCEL ROOM
2) CHECK / CHANGE PERSONAL DETAILS
3) MAKE PAYMENT
4) EXIT TO HOSTEL MANAGEMENT SYSTEM

PLEASE ENTER A CHOICE (1-4): 1
WOULD YOU LIKE TO :
1) CHANGE ROOM
2) CANCEL ROOM
3) CHANGE LAUNDRY OR GYM SERVICE : 2
ARE YOU SURE YOU WANTED TO CANCEL THE ROOM? YES(Y) OR NO(N):y
```

Figure 6.9.2.1 Cancel Room

```
****      WE HAVE SUCCESSFULLY CANCELLED YOUR ROOM!      ****
**      HOPE TO SEE YOU AGAIN SOON IN THE FUTURE      **
```

Figure 6.9.2.2 Text displayed After Successfully Cancelled

```
*****
|           STUDENT LOGIN SYSTEM           |
*****
PLEASE INSERT YOUR STUDENT ID (TP000000) : TP049952

*
* SORRY WE DON'T HAVE YOUR RECORD, PLEASE REGISTER! *
```

Figure 6.9.2.3 Information Deleted after cancel room

After input choice 2 for cancel room as shown in figure 6.9.2.1, the program will then ask if the user really meant to cancel the room. If user inputs Y, the program will then print text that shows user has successfully cancelled the room as shown in figure 6.9.2.2. Afterwards, if student tries to login, there won't be any record of the student again as shown in figure 6.9.2.3.

6.9.3 | Change Subscription Service

```
*****
|           WELCOME TO STUDENT LOGIN SYSTEM           |
*****
WHAT WOULD YOU LIKE TO DO?

1) CHANGE / CANCEL ROOM
2) CHECK / CHANGE PERSONAL DETAILS
3) MAKE PAYMENT
4) EXIT TO HOSTEL MANAGEMENT SYSTEM

PLEASE ENTER A CHOICE (1-4): 1
WOULD YOU LIKE TO :
1) CHANGE ROOM
2) CANCEL ROOM
3) CHANGE LAUNDRY OR GYM SERVICE : 3
WHICH SERVICE WOULD YOU LIKE TO ADD ON OR CANCEL ? LAUNDRY(L) OR GYM(G) : L
```

Figure 6.9.3.1 Change Laundry or Gym Service

If choice 3 is the input, the program will ask which subscription service the user wants to add as shown in figure 6.9.3.1. After the input, the program will then ask the user which service they like to add or cancel. Either laundry or gym.

```
UNSUBSCRIPTION MENU :

*****
|           LAUNDRY      RM20           |
*****
WOULD YOU LIKE TO SUBSCRIBE TO LAUNDRY EXPENSES ?
1) YES, I WOULD LIKE TO
2) NO , I WOULD NOT
PLEASE ENTER YOUR CHOICE (1 - 2): 2

***      SUCCESSFULLY UNSUBSCRIBE TO THE LAUNDRY SERVICE      ***
***      REDIRECTING TO THE MAIN MENU, PLEASE HOLD !      ***
```

Figure 6.9.3.2 Unsubscribe Laundry Service

If L was chosen, the program will then redirect users to the subscription or unsubscribe menu based on their previous choosing. As shown above in figure 6.9.3.2, the user has previously subscribed to laundry service, so he is redirected to unsubscribe menu. If they chose 2 not to subscribe the service again. The program will output that he had successfully unsubscribe to laundry service and then will be redirected to the main menu.

```

SUBSCRIPTION MENU :

*****
|          GYM      RM10      |
*****
WOULD YOU LIKE TO SUBSCRIBE TO GYM SERVICE ?
1) YES, I WOULD LIKE TO
2) NO , I WOULD NOT
PLEASE ENTER YOUR CHOICE (1 - 2): 1

***          SUCCESSFULLY SUBSCRIBE TO THE GYM SERVICE          ***
***          REDIRECTING TO THE MAIN MENU, PLEASE HOLD !          ***

```

Figure 6.9.3.3 Subscribe to Gym Service

If G was chosen, the program will then redirect users to the subscription or unsubscribe menu based on their previous choosing. As shown above in figure 6.9.3.3, the user has previously not subscribed to gym service, so he is redirected to subscribe menu. If they chose 1 to subscribe the service. The program will output that he had successfully subscribe to gym service and then will be redirected to the main menu.

6.10 | Check / Change Personal Details

```

*****
|          WELCOME TO STUDENT LOGIN SYSTEM      |
*****
WHAT WOULD YOU LIKE TO DO?

1) CHANGE / CANCEL ROOM
2) CHECK / CHANGE PERSONAL DETAILS
3) MAKE PAYMENT
4) EXIT TO HOSTEL MANAGEMENT SYSTEM

PLEASE ENTER A CHOICE (1-4): 2

```

Figure 6.10.1 Choice 2 Check / Change Details

If user input is 2 as shown in figure 6.10.1, the program will then redirect the user to the personal detail menu.

6.11 | Personal Details Menu

```
***** HERE IS YOUR BOOKING DETAILS *****  
  
STUDENT ID : TP049952  
YOUR FIRST NAME : JARED  
YOUR LAST NAME : CHAN  
YOUR GENDER : M  
YOUR IDENTIFICATION: 990423076521  
YOUR AGE : 20  
YOUR PHONE : 0134020303  
YOUR EMAIL : jaredchan0074@gmail.com  
YOUR BLOCK : A1  
YOUR ROOM : 1  
YOUR WEEK OF STAY : 5  
YOUR PAYMENT DUE : 2050  
YOUR PAYMENT MADE : 0  
LAUNDRY : NO  
GYM : YES  
  
WOULD YOU LIKE TO MAKE ANY CHANGES TO YOUR DETAILS ? YES(Y) OR NO(N) :■
```

Figure 6.11.1 Personal Details Menu

After being redirected from the student login menu, the personal details will show all the details of the student who login just now in the program. The information includes student ID, first name, last name, gender, identification and so on as shown in figure 6.11.1. The program will then request for an input asking if the user wants to change their personal details.

6.11.1 | Change Personal Details

```
***** HERE IS YOUR BOOKING DETAILS *****

STUDENT ID : TP049952
YOUR FIRST NAME : JARED
YOUR LAST NAME : CHAN
YOUR GENDER : M
YOUR IDENTIFICATION: 990423076521
YOUR AGE : 20
YOUR PHONE : 0134020303
YOUR EMAIL : jaredchan0074@gmail.com
YOUR BLOCK : A1
YOUR ROOM : 1
YOUR WEEK OF STAY : 5
YOUR PAYMENT DUE : 2050
YOUR PAYMENT MADE : 0
LAUNDRY : NO
GYM : YES

WOULD YOU LIKE TO MAKE ANY CHANGES TO YOUR DETAILS ? YES(Y) OR NO(N) :y
WHAT DETAILS WOULD YOU LIKE TO CHANGE? (1-7) :
1) STUDENT ID
2) FIRST NAME
3) LAST NAME
4) AGE
5) IDENTIFICATION
6) PHONE NUMBER
7) EMAIL
4
```

Figure 6.11.1.1 Change Personal Details

If user inputs Y, the program will then ask user which information to change. As shown in figure 6.11.1.1, the user wants to change the age.

```
WOULD YOU LIKE TO MAKE ANY CHANGES TO YOUR DETAILS ? YES(Y) OR NO(N) :y
WHAT DETAILS WOULD YOU LIKE TO CHANGE? (1-7) :
1) STUDENT ID
2) FIRST NAME
3) LAST NAME
4) AGE
5) IDENTIFICATION
6) PHONE NUMBER
7) EMAIL
4
PLEASE INSERT THE CORRECT AGE : 21
```

Figure 6.11.1.2 Changing Age

After choosing 4, the program will then request user to input the correct age as shown in figure 6.11.1.2. User input 21 as shown in figure 6.11.1.2.

```
*****      HERE IS YOUR BOOKING DETAILS      *****
STUDENT ID      : TP049952
YOUR FIRST NAME : JARED
YOUR LAST NAME  : CHAN
YOUR GENDER      : M
YOUR IDENTIFICATION: 990423076521
YOUR AGE         : 21
YOUR PHONE        : 0134020303
YOUR EMAIL        : jaredchan0074@gmail.com
YOUR BLOCK        : A1
YOUR ROOM         : 1
YOUR WEEK OF STAY : 5
YOUR PAYMENT DUE  : 2050
YOUR PAYMENT MADE : 0
LAUNDRY           : NO
GYM               : YES

WOULD YOU LIKE TO MAKE ANY CHANGES TO YOUR DETAILS ? YES(Y) OR NO(N) :-
```

Figure 6.11.1.3 Personal Detail Age Changed

After user input the correct age, the program will then change the age from 20 to 21 as shown in figure 6.11.1.3. The program can change the student ID, first name, last name, age, identification, phone number and email.

```
*****      HERE IS YOUR BOOKING DETAILS      *****
STUDENT ID      : TP049952
YOUR FIRST NAME : JARED
YOUR LAST NAME  : CHAN
YOUR GENDER      : M
YOUR IDENTIFICATION: 990423076521
YOUR AGE         : 21
YOUR PHONE        : 0134020303
YOUR EMAIL        : jaredchan0074@gmail.com
YOUR BLOCK        : A1
YOUR ROOM         : 1
YOUR WEEK OF STAY : 5
YOUR PAYMENT DUE  : 2050
YOUR PAYMENT MADE : 0
LAUNDRY           : NO
GYM               : YES

WOULD YOU LIKE TO MAKE ANY CHANGES TO YOUR DETAILS ? YES(Y) OR NO(N) :N
REDIRECTING TO THE LOGIN MENU, PLEASE HOLD !-
```

Figure 6.11.1.4 If User Chose Not to Change

If user inputs N in the personal detail menu, the program will redirect the user to the login menu as shown in figure 6.11.1.4.

6.12 | Payment Facility

```
*****
|           WELCOME TO STUDENT LOGIN SYSTEM           |
*****
WHAT WOULD YOU LIKE TO DO?

1) CHANGE / CANCEL ROOM
2) CHECK / CHANGE PERSONAL DETAILS
3) MAKE PAYMENT
4) EXIT TO HOSTEL MANAGEMENT SYSTEM

PLEASE ENTER A CHOICE (1-4): 3
```

Figure 6.12.1 If Choice 3

```
*****
|           PAYMENT FACILITY           |
*****
HI JARED, YOUR PAYMENT DUE IS 2050, HOW MUCH WOULD YOU LIKE TO PAY? :
```

Figure 6.12.2 If Choice 3

If user inputs 3 when in the student login system, the program will then redirect user to the payment facility. The payment facility will show the amount due by the student and ask for them to make their payment.

6.12.1 | Payment Amount More Than Amount Due

```
*****
|           PAYMENT FACILITY           |
*****
HI JARED, YOUR PAYMENT DUE IS 2050, HOW MUCH WOULD YOU LIKE TO PAY? : 2100
SORRY, PLEASE MAKE A PAYMENT LESSER THAN YOUR PAYMENT DUE !
```

Figure 6.12.1.1 If Payment is more than amount due

If users chose to pay more than the amount due, the program will request the user to make a payment lesser than their amount due and will again request another amount from the user as shown in figure 6.12.1.1.

6.12.2 | Payment Successfully Made

```
*****
|          PAYMENT FACILITY          |
*****
HI JARED, YOUR PAYMENT DUE IS 2050, HOW MUCH WOULD YOU LIKE TO PAY? : 200
***          PAYMENT SUCCESSFULLY MADE !          ***
***          REDIRECTING TO LOGIN MENU          ***
***
```

Figure 6.12.2.1 Payment Successfully Made

```
YOUR PAYMENT DUE      : 1850
YOUR PAYMENT MADE    : 200
```

Figure 6.12.2.2 Student Details changed after Payment successfully made

After payment successfully made, the program will then update students' details, and redirect the student to the login menu as shown in figure 6.12.2.1. After student detail has been updated the payment made will be added and the payment due will be updated based on payment made as shown in figure 6.12.2.2.

6.13 | Exit from Student Login System

```
*****
|          WELCOME TO STUDENT LOGIN SYSTEM          |
*****
WHAT WOULD YOU LIKE TO DO?
1) CHANGE / CANCEL ROOM
2) CHECK / CHANGE PERSONAL DETAILS
3) MAKE PAYMENT
4) EXIT TO HOSTEL MANAGEMENT SYSTEM
PLEASE ENTER A CHOICE (1-4): 4
REDIRECTING TO THE HOSTEL MANAGEMENT SYSTEM, PLEASE HOLD ! -
```

Figure 6.13.1 Exit from Student Login System

If user inputs 4, the program will then redirect users back to the hotel management system main menu, as shown in the figure 6.13.1.

6.14 | Admin Login

```
*****  
| WELCOME TO WISDOM COLLEGE HOSTEL MANAGEMENT SYSTEM |  
*****  
  
PLEASE PROCEED TO FOLLOWING CHOICES :  
1) STUDENT REGISTER  
2) STUDENT LOGIN  
3) ADMIN LOGIN  
4) EXIT  
  
PLEASE ENTER A CHOICE (1-4): 3
```

Figure 6.14.1 If Choice 3 Admin Login

```
*****  
| ADMIN LOGIN SYSTEM |  
*****  
  
WOULD YOU LIKE TO: 1) PROCEED LOGIN 2) EXIT TO MAIN MENU  
WHAT WOULD YOU LIKE TO DO (1-2) :
```

Figure 6.14.2 Admin Login System

If user chooses 3 as input, the program will then redirect the user to the admin login system as shown in figure 6.14.1 and figure 6.14.2.

6.14.1 | Admin Successfully Login

```
*****
|           ADMIN LOGIN SYSTEM           |
*****
WOULD YOU LIKE TO: 1) PROCEED LOGIN 2) EXIT TO MAIN MENU
WHAT WOULD YOU LIKE TO DO (1-2) : 1
PLEASE KEY IN THE PASSWORD : AbcD1234

***      SUCCESSFULLY LOGIN INTO ADMIN SYSTEM      ***
***      REDIRECTING TO ADMIN SYSTEM!      ***
-
```

Figure 6.14.1.1 Successful Admin Login

As shown in figure 6.14.1.1, After successful login by entering password AbcD1234, the program will then redirect the user to the admin system.

6.15 | Admin System

```
*****
|           WELCOME TO ADMIN SYSTEM           |
*****
1) CHECK OVERALL DETAILS
2) SEARCH FOR STUDENT DETAILS
3) CHECK BLOCK ROOM AVAILABILITY
4) EXIT TO MAIN MENU
WHAT WOULD YOU LIKE TO DO ? (1-4): -
```

Figure 6.15.1 Admin System

Figure 6.15.1 shows the admin system. Within admin system there are 4 choices which is 1 check overall details, 2. search for student details, 3. Check block room availability and 4. Exit to main menu.

6.15.1 | Check Overall Details

***** CHECK MENU *****								
BLOCK	GENDER	AVAILABILITY	TYPE	AMOUNT RECEIVABLE	PAYMENT RECEIVED	STUDENT BOOKED	AMOUNT	
A 1	MALE	94	SINGLE	9198	4302	6		
A 2	MALE	97	2-BEDS	5666	1054	7		
B 1	FEMALE	96	SINGLE	9677	2133	4		
B 3	FEMALE	99	3-BEDS	2449	1011	4		

TOTAL ROOM AVAIBLE: 386 TOTAL AMOUNT RECEIVABLE: 26990 TOTAL PAYMENT RECEIVED: 8500 TOTAL HOSTELER: 21

PERCENTAGE OF MALE STUDENT: 61.90% PERCENTAGE OF FEMALE STUDENTS: 38.10%

1) CHECK ALL STUDENTS DETAILS BY BLOCK
2) EXIT TO ADMIN SYSTEM MENU
PLEASE ENTER A CHOICE (1-2) :

Figure 6.15.1.1 Check Overall Details

After input 1 and redirect to the check overall menu, there will be a table stating all the necessary details, available room, amount receivable, payment received, student booked amount, total room available, total amount receivable, total payment received and total hosteler as well as the percentage of male and female students. The program will then request an input from user to either 1 list all the details of hosteler in the block or 2, exit to admin system menu.

***** CHECK MENU *****								
BLOCK	GENDER	AVAILABILITY	TYPE	AMOUNT RECEIVABLE	PAYMENT RECEIVED	STUDENT BOOKED	AMOUNT	
A 1	MALE	94	SINGLE	9198	4302	6		
A 2	MALE	97	2-BEDS	5666	1054	7		
B 1	FEMALE	96	SINGLE	9677	2133	4		
B 3	FEMALE	99	3-BEDS	2449	1011	4		

TOTAL ROOM AVAIBLE: 386 TOTAL AMOUNT RECEIVABLE: 26990 TOTAL PAYMENT RECEIVED: 8500 TOTAL HOSTELER: 21

PERCENTAGE OF MALE STUDENT: 61.90% PERCENTAGE OF FEMALE STUDENTS: 38.10%

1) CHECK ALL STUDENTS DETAILS BY BLOCK
2) EXIT TO ADMIN SYSTEM MENU
PLEASE ENTER A CHOICE (1-2) : 1
WHICH BLOCK WOULD YOU LIKE TO SEARCH ? (A1 , A2 , B1 , B3)
PLEASE ENTER YOUR CHOICE : a1
STUDENT ID: TP204357 FIRST NAME: ROBERT GENDER: M BLOCK: A1 ROOM: 96 IDENTIFICATION: AK729302 PHONE: 0174338694 EMAIL: iron_man99@gmail.com AMOUNT RECEIVABLE: 2150
STUDENT ID: TP028375 FIRST NAME: JUN GENDER: M BLOCK: A1 ROOM: 4 IDENTIFICATION: 990523029482 PHONE: 0150385739 EMAIL: junhong_99@gmail.com AMOUNT RECEIVABLE: 518
STUDENT ID: TP029483 FIRST NAME: BEN GENDER: M BLOCK: A1 ROOM: 84 IDENTIFICATION: 991213082732 PHONE: 019283029 AMEMAIL: ben_thong_22@yahoo.com AMOUNT RECEIVABLE: 2100
STUDENT ID: TP059271 FIRST NAME: HENG GENDER: M BLOCK: A1 ROOM: 22 IDENTIFICATION: 990912018372 PHONE: 0168273822 EMAIL: lee_heng@yahoo.com AMOUNT RECEIVABLE: 1290
STUDENT ID: TP058472 FIRST NAME: DING GENDER: M BLOCK: A1 ROOM: 56 IDENTIFICATION: 92038420382 PHONE: 0125839204 EMAIL: TP058472@mail.apu.edu.my AMOUNT RECEIVABLE: 1290
STUDENT ID: TP049952 FIRST NAME: JARED GENDER: M BLOCK: A1 ROOM: 1 IDENTIFICATION: 990423076521 PHONE: 0134020303 EMAIL: jaredchan0074@gmail.com AMOUNT RECEIVABLE: 1850

PRESS ANYTHING TO EXIT :

Figure 6.15.1.2 Checking all hosteler details

After input one, the program will request the block that the admin wants to check. Figure 6.15.1.2 inputs a1, so the program will list all the details of the hosteler staying in block a1, including name, student ID as well as amount receivable and then ask for any input for user to exit to the admin system.

REDIRECTING TO THE ADMIN SYSTEM, PLEASE HOLD !

Figure 6.15.1.2 Exit from Check Menu

If user inputs 2 in the check menu, the program will redirect the user to the admin system as shown in figure 6.15.1.2.

6.15.2 | Search for Student Details

```
*****| STUDENT SEARCH SYSTEM |*****  
*****|  
PLEASE INSERT THE STUDENT ID TO SEARCH THE STUDENT DETAILS : TP049952_
```

Figure 6.15.2.1 Student Search System

After user inputs 2 in Admin System, the program will redirect admin to the student search system as shown in figure 6.15.2.1. The program will request for student ID to search.

```
***** HERE IS THE BOOKING DETAILS OF THE STUDENT *****  
  
STUDENT ID      : TP049952  
FIRST NAME     : JARED  
LAST NAME      : CHAN  
GENDER          : M  
IDENTIFICATION: 990423076521  
AGE             : 21  
PHONE           : 0134020303  
EMAIL           : jaredchan0074@gmail.com  
BLOCK           : A1  
ROOM            : 1  
PAYMENT DUE     : 1850  
PAYMENT MADE    : 200  
  
ENTER ANYTHING TO EXIT : _
```

Figure 6.15.2.2 Searched student details

After input student ID, if there is a record, the program will fetch the information from the text file and display all the relevant details as shown in figure 6.15.2.2 for the admin. After that, the program will request any input to exit to the admin system.

6.15.3 | Check Block Room Availability

```
*****
|           WELCOME TO ADMIN SYSTEM           |
*****
1) CHECK OVERALL DETAILS
2) SEARCH FOR STUDENT DETAILS
3) CHECK BLOCK ROOM AVAILABILITY
4) EXIT TO MAIN MENU
WHAT WOULD YOU LIKE TO DO ? (1-4): 3
WHICH BLOCK WOULD YOU LIKE TO CHECK? (A1 , A2 , B1 , B3) ? : a1
```

Figure 6.15.3.1 Check Block Room Availability

```
*****
|           BLOCK A1           |
*****
A00  0  A01  1  A02  0  A03  0  A04  1  A05  0  A06  0  A07  0  A08  0  A09  0
A10  0  A11  0  A12  0  A13  0  A14  0  A15  0  A16  0  A17  0  A18  0  A19  0
A20  0  A21  0  A22  1  A23  0  A24  0  A25  0  A26  0  A27  0  A28  0  A29  0
A30  0  A31  0  A32  0  A33  0  A34  0  A35  0  A36  0  A37  0  A38  0  A39  0
A40  0  A41  0  A42  0  A43  0  A44  0  A45  0  A46  0  A47  0  A48  0  A49  0
A50  0  A51  0  A52  0  A53  0  A54  0  A55  0  A56  1  A57  0  A58  0  A59  0
A60  0  A61  0  A62  0  A63  0  A64  0  A65  0  A66  0  A67  0  A68  0  A69  0
A70  0  A71  0  A72  0  A73  0  A74  0  A75  0  A76  0  A77  0  A78  0  A79  0
A80  0  A81  0  A82  0  A83  0  A84  1  A85  0  A86  0  A87  0  A88  0  A89  0
A90  0  A91  0  A92  0  A93  0  A94  0  A95  0  A96  1  A97  0  A98  0  A99  0
*****
|           0 MEANS AVAILABLE           1 MEANS OCCUPIED           |
*****
ENTER ANYTHING TO EXIT!
```

Figure 6.15.3.2 Check Room Availability Visuals

After students input 3 for choice check block availability, the program will request an input from user to choose which block admin wants to view. Figure 6.15.3.1 shows that user input a1, then the program will display the whole block a1 to check available rooms as shown in figure 6.15.3.2. The program will then request any input to exit to the admin system.

6.15.4 | Exit from Admin System

```
*****
|          WELCOME TO ADMIN SYSTEM          |
*****
1) CHECK OVERALL DETAILS
2) SEARCH FOR STUDENT DETAILS
3) CHECK BLOCK ROOM AVAILABILITY
4) EXIT TO MAIN MENU
WHAT WOULD YOU LIKE TO DO ? (1-4): 4

REDIRECTING TO THE HOSTEL MANAGEMENT SYSTEM, PLEASE HOLD !
```

Figure 6.15.4.1 Exit from Admin System

After user input 4, the program will redirect user to hotel management system as shown in figure 6.15.4.1.

6.16 | Exit Program

```
-----THANKS FOR USING THE HOSTEL MANAGEMENT SYSTEM-----
-----HOPE TO SEE YOU SOON !!-----
```

Figure 6.16.1 Exit from Program

After input 4 in the hostel management system, the program will exit the main menu and proceed to terminate the system.

7.0 | Conclusion

The Wisdom Hostel Management system was made by more than 20 functions each having its own unique features and functionality. The system contains 22 source file and one header file that contains all the function, thus 1 file for storing information of the students. The functionality of the system is ready to be utilized and it is ready to be presented to Wisdom College.

Process of developing this system has opened my eyes as if how system is developed based on simple yet hard steps like designing the program in pseudocode and flowchart, having to debug one of the most difficult logic errors. Overall, I've really learnt a lot from our Lecturer Mr. Sivaguru, as he had not only taught us how to program, but most of the time having detail explanations as if what skill should be needed and giving us a short glimpse of how the working environment works.

8.0 | References

Programiz, n.d. *C strcpy()*. [Online]

Available at: <https://www.programiz.com/c-programming/library-function/string.h/strcpy>
[Accessed 10 June 2019].

Singh, C., 2015. *C – goto statement with example*. [Online]

Available at: <https://beginnersbook.com/2014/01/c-goto-statement/>
[Accessed 10 June 2019].

Tutorialspoint, n.d. *C library function - fscanf()*. [Online]

Available at: https://www.tutorialspoint.com/c_standard_library/c_function_fscanf.htm
[Accessed 12 June 2019].

Tutorialspoint, n.d. *C library function - strcmp()*. [Online]

Available at: https://www.tutorialspoint.com/c_standard_library/c_function_strcmp.htm
[Accessed 10 June 2019].