



ASSIGNMENT

CT108-3-1-ICP

STUDENT NAME & TP NO.: Aidan Chang Wai Yue (TP063411)

INTAKE CODE: 112022-YGM

MODULE CODE: CT108-3-1-ICP

MODULE NAME: Introduction to C Programming

LECTURER NAME: Yogeswari Muthusamy

HAND-OUT DATE: 31st December 2022

HAND-IN DATE: 28th February 2023

Contents

1.0 Introduction.....	4
2.0 Assumptions.....	5
3.0 Design of the Program (Pseudocode)	6
3.1 Importing Headers	6
3.2 Declaring Structs.....	6
3.3 Registration Functions	7
3.3.1 Incrementing Staff ID	7
3.3.2 Incrementing Approver ID	7
3.3.3 Registering Staff	8
3.3.4 Registering Approver.....	10
3.4 Login Functions	12
3.4.1 Logging in as a Staff.....	12
3.4.2 Logging in as an Approver	13
3.4.3 Logging in as an Administrator	14
3.5 Calander Function.....	15
3.6 Staff Homepage Functions.....	17
3.6.1 Applying Leaves	17
3.6.2 Viewing and Cancelling Leaves	30
3.6.3 Deleting Leaves	36
3.7 Admin Homepage Functions	40
3.7.1 Updating Leaves	40
3.7.2 Searching Staff Leaves	43
3.7.3 Generating Monthly Report	44
3.8 Approver Functions	47
3.8.1 Approving Leaves.....	47
3.8.2 Rejecting Leaves.....	51
3.8.3 Searching Leave Dates.....	54
3.9 All Homepages.....	57
3.9.1 Staff Homepage	57
3.9.2 Admin Homepage	59
3.9.3 Approver Homepage.....	62
3.9.4 Menu	63
3.10 Main Function.....	65
4.0 Additional Features of the Program.....	66

4.1 Feedback Form.....	66
4.2 Terms & Conditions.....	67
4.3 Delete Leaves Function.....	68
5.0 Sample Inputs and Outputs	70
5.1 Menu	70
5.1.1 Feedback Form.....	71
5.1.2 Terms & Conditions.....	72
5.1.3 Exit Function.....	72
5.2 Administrator/Super Administrator	73
5.2.1 Logging in as an Admin.....	73
5.2.2 Adding New Staffs/Approvers.....	75
5.2.2.1 New Staff	75
5.2.2.2 New Approver.....	77
5.2.3 Updating Leave Balance	78
5.2.4 Searching Staff Leave Details.....	80
5.2.5 Generating Monthly Report	81
5.2.6 Logout.....	83
5.3 Staff.....	83
5.3.1 Registering as a Staff	83
5.3.2 Logging in as a Staff	85
5.3.3 Applying Leaves	86
5.3.4 Checking Leave Balance.....	91
5.3.5 Viewing Leave Summary	91
5.3.6 Cancelling Leaves.....	92
5.3.7 Deleting Leaves	96
5.3.8 Logout.....	100
5.4 Approver	101
5.4.1 Logging in as an Approver	101
5.4.2 Approving Leaves.....	102
5.4.3 Rejecting Leaves.....	105
5.4.4 Searching Leave Dates.....	107
5.4.5 Logout.....	109
6.0 Conclusion	110
7.0 References.....	111

1.0 Introduction

The case study of this assignment is centred around creating a leave management system for Asia Pacific University (APU), a well-known private university situated in Bukit Jalil, Kuala Lumpur. With 500 staff and various different departments and schools of which the staff are working under, the university has decided to ask programmers to develop a new and more efficient leave application and management system. The system will be able to house three types of users: administrators or also known as super administrators, superiors or also known as leave approvers, and staff. All of these roles will have their own functions that they should be able to carry out in this system, with the super administrator being able to add new staffs and approvers, update the leave balances for staffs, search for any staff's leave information using their staff ID and generating a monthly report in the form of a text file which details the leave statistics of each school in the university. The staff should be able to apply for their leaves, cancel their leaves, check all of their leave statuses and check the summary of their applied leaves. Finally, the superior or the leave approver should be able to approve or reject leave applications made by staff members and search for a specific date to see the number of staff under them who applied for leave on that day and assist them to decide whether or not to approve or reject certain leave applications.

With all of this in mind, the system will be designed with a simple menu-driven UI so that users can easily navigate throughout the system and understand each function presented in the system. The ultimate objective of this assignment is to not just create a simple system that users can easily interpret and use, but also to make it function efficiently with as little problems as possible. The full program will be coded in the C Programming language and using the CodeBlocks application. This report will first cover the introduction and assumptions of the program, the design of the program which will be the program's pseudocode, followed by the additional features of the program, then the sample inputs and outputs of the code, and finally the conclusion and its references.

2.0 Assumptions

First and foremost, Asia Pacific University is a university which allows its students to enrol in one of its six main schools: the School of Business, the School of Computing, the School of Arts, the School of Engineering, the School of Psychology and the School of Accounting (Asia Pacific University, n.d.). All of these schools will be present in the monthly leave statistics report which the super administrator can generate, however, the program will also add an additional department for its monthly leave statistics report which is its admin department as this report pertains to the whole school, and not just the staff and lecturers working under these departments and schools which students are studying under, so this includes the people who are running and maintaining the internals of the university behind the scenes. So, the monthly leave statistics report will cover all of the schools and the administrative department. It is also to be assumed that the leaves can be applied from the current year up to 2025.

Next, the current program is assumed to only be catering to a single school out of all the six schools, thus the school chosen for this assignment would be the School of Computing, and the departments chosen for this assignment would be the Human Resources Department and the Marketing Department. These two departments will share the exact same leave management and application program as logically, there isn't any changes to be made to the system for these two departments.

And finally, the monthly leave statistics will only have an upper bound of 3 per school excluding the school of computing but including its administrative department as the average absenteeism rate for the general workforce was shown be 3.2%, which in total should make the average number of employees who applied for leave every month in this university to be 16 as 0.032×500 would be 16 (Blakely-Gray, 2022). The School of Computing was excluded in this case as it was chosen as the primary school of focus for this assignment, hence its numbers must reflect the real leave applications from the text file. However, splitting up 16 into 6 different categories of schools would not make it a whole number, thus the upper bound would be made to be 3, making a maximum number of 18 staffs to apply for leave every month for each school excluding the School of Computing which will be shown in the leave statistics report. The superior is also assumed to have around 5 people working under them. When they search for a specific date to view the amount of leave applications, the numbers produced will also be reflective of the real leave applications in the text file, similar to the admin's monthly leave statistics function which was mentioned earlier.

3.0 Design of the Program (Pseudocode)

3.1 Importing Headers

1. IMPORT <stdio.h>
2. IMPORT <stdlib.h>
3. IMPORT <string.h>
4. IMPORT <time.h>

3.2 Declaring Structs

1. DEFINE STRUCT staff:
 - a. DECLARE CHAR staff_username[50]
 - b. DECLARE CHAR staff_password[20]
 - c. DECLARE CHAR staff_type[50]
 - d. DECLARE INT staff_ID
 - e. DECLARE INT annualLeave
 - f. DECLARE INT compLeave
 - g. DECLARE INT emergLeave
 - h. DECLARE INT matLeave
 - i. DECLARE INT medLeave
2. ENDDEFINE staff
3. DEFINE STRUCT approver:
 - a. DECLARE CHAR approver_username[50]
 - b. DECLARE CHAR approver_password[20]
 - c. DECLARE INT app_ID
4. ENDDEFINE approver
5. DEFINE STRUCT leaves:
 - a. DECLARE CHAR typeLeave[50]
 - b. DECLARE CHAR statusLeave[20]
 - c. DECLARE INT leaveStaff_ID
 - d. DECLARE INT day
 - e. DECLARE INT month
 - f. DECLARE INT year
 - g. DECLARE INT duration
6. ENDDEFINE leaves

3.3 Registration Functions

3.3.1 Incrementing Staff ID

1. FUNCTION incrementOFStaffID()
 - a. SET StaffID = 100
 - b. DECLARE STRUCT staff as st
 - c. OPENFILE “staffDetails.txt” FOR READ
 - d. IF “staffDetails.txt” == NULL THEN
 - i. DISPLAY “File not found! \n”
 - ii. RETURN
 - e. ENDIF
 - f. WHILE NOT EOF “staffDetails.txt” DO
 - i. READFILE “staffDetails.txt”
 - ii. SET StaffID = st.staff_ID
 - g. ENDWHILE
 - h. CLOSEFILE “staffDetails.txt”
 - i. RETURN StaffID + 1
2. ENDFUNCTION

3.3.2 Incrementing Approver ID

1. FUNCTION incrementOFApproverID()
 - a. SET ApproverID = 200
 - b. DECLARE STRUCT approver as app
 - c. OPENFILE “approverDetails.txt” FOR READ
 - d. IF “approverDetails.txt” == NULL THEN
 - i. DISPLAY “File not found! \n”
 - ii. RETURN
 - e. ENDIF
 - f. WHILE NOT EOF “approverDetails.txt” DO
 - i. READFILE “approverDetails.txt”
 - ii. SET ApproverID = app.app_ID
 - g. ENDWHILE
 - h. CLOSEFILE “approverDetails.txt”
 - i. RETURN ApproverID + 1
2. ENDFUNCTION

3.3.3 Registering Staff

1. FUNCTION registerStaff()
 - a. DECLARE CHAR dummy_password[50]
 - b. DECLARE INT type_staffCh
 - c. DECLARE STRUCT staff as st
 - d. OPENFILE "staffDetails.txt" FOR APPEND+
 - e. IF "staffDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - f. ENDIF
 - g. CLEAR INPUT BUFFER
 - h. DISPLAY "Which of the following staff are you?\n"
 - i. DISPLAY "1. Academic\n"
 - j. DISPLAY "2. Management\n"
 - k. DISPLAY "3. Technical\n"
 - l. DISPLAY "4. Administrative\n"
 - m. GET type_staffCh
 - n. IF type_staffCh == 1 THEN
 - i. COPY "Academic" to st.staff_type
 - o. ELSEIF type_staffCh == 2 THEN
 - i. COPY " Management" to st.staff_type
 - p. ELSEIF type_staffCh == 3 THEN
 - i. COPY " Technical" to st.staff_type
 - q. ELSEIF type_staffCh == 4 THEN
 - i. COPY " Administrative" to st.staff_type
 - r. ELSE
 - i. DISPLAY "Invalid option.\n"
 - ii. RETURN
 - s. ENDIF
 - t. CLEAR INPUT BUFFER
 - u. DISPLAY "Enter a username: \n"
 - v. GET st.staff_username
 - w. REWINDFILE "staffDetails.txt"


```

x. DECLARE STRUCT staff as currSt
y. WHILE NOT EOF "staffDetails.txt" DO
    i. READFILE "staffDetails.txt"
    ii. IF currSt.staff_username == st.staff_username
        1. DISPLAY "Username already exists, try a different one.\n"
        2. RETURN
    iii. ENDIF
z. ENDWHILE
aa. WHILE TRUE DO
    i. CLEAR INPUT BUFFER
    ii. DISPLAY "Enter a password: \n"
    iii. GET dummy_password
    iv. IF length of dummy_password < 8 OR length of dummy_password > 20
        OR dummy_password has spaces THEN
        1. DISPLAY "Invalid password, must be between 8 and 20
            characters and cannot contain spaces.\n"
        2. CONTINUE
    v. ELSE
        1. CLEAR INPUT BUFFER
        2. DISPLAY "Confirm your password: \n"
        3. GET st.staff_password
        4. IF st.staff_password == dummy_password THEN
            a. SET st.staff_ID = incrementOFStaffID()
            b. SET st.annualLeave = 16
            c. st.compLeave = 7
            d. SET st.emergLeave = 3
            e. SET st.matLeave = 98
            f. SET st.medLeave = 22
            g. WRITEFILE "staffDetails.txt",
                "%d, %s, %s, %s, %d, %d, %d, %d, %d\n", st.staff_ID,
                st.staff_username, st.staff_password, st.staff_type,
                st.annualLeave, st.compLeave, st.emergLeave,
                st.matLeave, st.medLeave
            h. REWINDFILE "staffDetails.txt"

```

- i. DISPLAY "Staff registered successfully with ID %d.\n",
st.staff_ID
 - j. CLOSEFILE "staffDetails.txt"
 - k. BREAK
 - 5. ELSE
 - a. DISPLAY "Passwords do not match!\n"
 - b. CONTINUE
 - 6. ENDIF
 - vi. ENDIF
- bb. ENDWHILE
- cc. CALL menu()
- 2. ENDFUNCTION

3.3.4 Registering Approver

- 1. FUNCTION registerStaff()
 - a. DECLARE CHAR dummy_password[50]
 - b. DECLARE STRUCT approver as app
 - c. OPENFILE "approverDetails.txt" FOR APPEND+
 - d. IF "approverDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - e. ENDIF
 - f. CLEAR INPUT BUFFER
 - g. DISPLAY "Enter a username: \n"
 - h. GET app.approver_username
 - i. REWINDFILE "approverDetails.txt"
 - j. DECLARE STRUCT approver as currApp
 - k. WHILE NOT EOF "approverDetails.txt" DO
 - i. READFILE "approverDetails.txt"
 - ii. IF currApp.approver_username == app.approver_username
 - 1. DISPLAY "Username already exists, try a different one.\n"
 - 2. RETURN
 - iii. ENDIF
- l. ENDWHILE

```

m. WHILE TRUE DO
    i. CLEAR INPUT BUFFER
    ii. DISPLAY "Enter a password: \n"
    iii. GET dummy_password
    iv. IF length of dummy_password < 8 OR length of dummy_password > 20
        OR dummy_password has spaces THEN
        1. DISPLAY "Invalid password, must be between 8 and 20
            characters and cannot contain spaces.\n"
        2. CONTINUE
    v. ELSE
        1. CLEAR INPUT BUFFER
        2. DISPLAY "Confirm your password: \n"
        3. GET app.approver_password
        4. IF app.approver_password == dummy_password THEN
            a. SET app.app_ID = incrementOFApproverID()
            b. WRITEFILE "approverDetails.txt", "%d, %s, %s\n",
                app.app_ID, app.approver_username,
                app.approver_password
            c. REWINDFILE "approverDetails.txt"
            d. DISPLAY "Approver registered successfully with
                ID %d.\n", app.app_ID
            e. CLOSEFILE "approverDetails.txt"
            f. BREAK
        5. ELSE
            a. DISPLAY "Passwords do not match!\n"
            b. CONTINUE
        6. ENDIF
    vi. ENDIF
n. ENDWHILE
o. CALL menu()
2. ENDFUNCTION

```

3.4 Login Functions

3.4.1 Logging in as a Staff

1. FUNCTION staffLogin()
 - a. DECLARE CHAR s_username[50]
 - b. DECLARE CHAR s_password[20]
 - c. DECLARE INT tries = 3
 - d. DECLARE INT s_id = 0
 - e. DECLARE STRUCT staff as st
 - f. OPENFILE “staffDetails.txt” FOR READ
 - g. IF “staffDetails.txt” == NULL THEN
 - i. DISPLAY “File not found!\n”
 - ii. RETURN
 - h. ENDIF
 - i. WHILE tries > 0 DO
 - i. CLEAR INPUT BUFFER
 - ii. DISPLAY "Enter your Staff ID: \n"
 - iii. GET s_id
 - iv. CLEAR INPUT BUFFER
 - v. DISPLAY "Enter username: \n"
 - vi. GET s_username
 - vii. CLEAR INPUT BUFFER
 - viii. DISPLAY "Enter password: \n"
 - ix. GET s_password
 - x. REWINDFILE “staffDetails.txt”
 - xi. WHILE NOT EOF “staffDetails.txt” DO
 1. READFILE “staffDetails.txt”
 2. IF s_id == st.staff_ID AND s_username == st.staff_username AND s_password == st.staff_password THEN
 - a. DISPLAY "Login successful! Redirecting to the staff homepage...\n"
 - b. DECLARE POINTER INT *s_id_ptr = st.staff_ID
 - c. CLOSEFILE “staffDetails.txt”
 - d. CALL staffHomepage(s_id_ptr)

3. ENDIF

xii. ENDWHILE

xiii. DISPLAY "Login failed. %d attempts left.\n"

xiv. SUBTRACT tries by 1

j. ENDWHILE

k. DISPLAY "No attempts left. Returning to main menu...\n"

l. CLOSEFILE "staffDetails.txt"

m. RETURN 0

2. ENDFUNCTION

3.4.2 Logging in as an Approver

1. FUNCTION approverLogin ()

a. DECLARE CHAR a_username [50]

b. DECLARE CHAR a_password[20]

c. DECLARE INT tries = 3

d. DECLARE INT a_id = 0

e. DECLARE STRUCT approver as app

f. OPENFILE "approverDetails.txt" FOR READ

g. IF "approverDetails.txt" == NULL THEN

i. DISPLAY "File not found!\n"

ii. RETURN

h. ENDIF

i. WHILE tries > 0 DO

i. CLEAR INPUT BUFFER

ii. DISPLAY "Enter your Approver ID: \n"

iii. GET a_id

iv. CLEAR INPUT BUFFER

v. DISPLAY "Enter username: \n"

vi. GET a_username

vii. CLEAR INPUT BUFFER

viii. DISPLAY "Enter password: \n"

ix. GET a_password

x. REWINDFILE "approverDetails.txt"

xi. WHILE NOT EOF "approverDetails.txt" DO

1. READFILE “approverDetails.txt”
2. IF a_id == app.approver_ID AND a_username == app.approver_username AND a_password == app.approver_password THEN
 - a. DISPLAY "Login successful! Redirecting to the approver homepage...\n"
 - b. CLOSEFILE “approverDetails.txt”
 - c. CALL approverHomepage()
3. ENDIF
- xii. ENDWHILE
- xiii. DISPLAY "Login failed. %d attempts left.\n"
- xiv. SUBTRACT tries by 1
- j. ENDWHILE
- k. DISPLAY "No attempts left. Returning to main menu...\n"
- l. CLOSEFILE “approverDetails.txt”
- m. RETURN 0
2. ENDFUNCTION

3.4.3 Logging in as an Administrator

1. FUNCTION adminLogin ()
 - a. DECLARE CHAR adm_username [50]
 - b. DECLARE CHAR adm_password[20]
 - c. DECLARE INT tries = 3
 - d. WHILE tries > 0 DO
 - i. CLEAR INPUT BUFFER
 - ii. DISPLAY "Enter username: \n"
 - iii. GET adm_username
 - iv. CLEAR INPUT BUFFER
 - v. DISPLAY "Enter password: \n"
 - vi. GET adm_password
 - vii. IF adm_username == “admin” AND adm_password == “password” THEN
 1. DISPLAY "Login successful! Redirecting to the Admin homepage...\n"

- 2. CALL adminHomepage()
- viii. ELSE
 - 1. DISPLAY "Login failed. %d attempts left.\n"
 - 2. SUBTRACT tries by 1
- ix. ENDIF
- e. ENDWHILE
- f. DISPLAY "No attempts left. Returning to main menu...\n"
- g. RETURN 0
- 2. ENDFUNCTION

3.5 Calander Function

- 1. FUNCTION leaveCalander(int *day, int *month, int *year)
 - a. SET now = current time
 - b. DECLARE STRUCT local = localtime(current time)
 - c. WHILE TRUE DO
 - i. CLEAR INPUT BUFFER
 - ii. DISPLAY "\nPlease enter your starting day in DD format: \n"
 - iii. GET day
 - iv. CLEAR INPUT BUFFER
 - v. DISPLAY "Please enter your starting month in MM format: \n"
 - vi. GET month
 - vii. CLEAR INPUT BUFFER
 - viii. DISPLAY "Please enter your starting year in YYYY format: \n"
 - ix. GET year
 - x. IF *year > local->tm_year+1900 OR (*year == local->tm_year+1900 AND (*month > local->tm_mon+1 OR (*month == local->tm_mon+1 && *day > local->tm_mday))) THEN
 - 1. IF *year >= 2023 AND *year <= 2025 THEN
 - a. IF 1 <= *month OR *month <= 12 THEN
 - i. IF (1 <= *day AND *day <= 31) AND (*month == 1 OR *month == 3 OR *month == 5 OR *month == 7 OR *month == 8 OR *month == 10 OR *month == 12) THEN
 - 1. DISPLAY "Valid date!\n"

```

2. BREAK
ii. ELSEIF (1 <= *day AND *day <= 30) AND
(*month == 4 OR *month == 6 OR *month == 9
OR *month == 11) THEN
    1. DISPLAY "Valid date!\n"
    2. BREAK
iii. ELSEIF (1 <= *day AND *day <= 28) AND
(*month == 2) THEN
    1. DISPLAY "Valid date!\n"
    2. BREAK
iv. ELSEIF (1 <= *day AND *day <= 29) AND
(*year%4 == 0) THEN
    1. DISPLAY "Valid date!\n"
    2. BREAK
v. ELSE
    1. DISPLAY "Invalid date!\n"
    2. CONTINUE
vi. ENDIF
b. ELSE
    i. DISPLAY "Invalid date!\n"
    ii. CONTINUE
c. ENDIF
2. ELSE
    a. DISPLAY "Invalid date!\n"
    b. CONTINUE
3. ENDIF
xi. ELSE
    1. DISPLAY "Invalid date! Please enter a future date!\n"
    2. CONTINUE
xii. ENDIF
d. ENDWHILE
2. ENDFUNCTION

```


3.6 Staff Homepage Functions

The function for the staff to view their leave balances will be under the staff homepage function, which will be discussed later. Because some Microsoft Word has a limit when it comes to the levels of indentations, when I reach the limit of those indentations I will write the continuation of the pseudocode in a notepad, take a screenshot of it and paste it into the report.

3.6.1 Applying Leaves

1. FUNCTION applyLeave(int *s_id_ptr)
 - a. DECLARE INT applyLeaveChoice
 - b. SET INT day = 0
 - c. SET INT month = 0
 - d. SET INT year = 0
 - e. DECLARE CHAR buff[10]
 - f. DECLARE STRUCT staff as st
 - g. DECLARE STRUCT leaves as lv
 - h. WHILE TRUE DO
 - i. DISPLAY "\n-----\n"
 - ii. DISPLAY "***WELCOME TO THE APU LEAVE APPLICATION
SYSTEM***\n"
 - iii. DISPLAY "-----\n"
 - iv. DISPLAY "1. Annual Leave\n"
 - v. DISPLAY "2. Compassionate Leave\n"
 - vi. DISPLAY "3. Emergency Leave\n"
 - vii. DISPLAY "4. Maternity Leave\n"
 - viii. DISPLAY "5. Medical Leave\n"
 - ix. DISPLAY "6. Back\n"
 - x. DISPLAY "Please enter your choice: \n"
 - xi. CLEAR INPUT BUFFER
 - xii. GET buff
 - xiii. SET applyLeaveChoice = buff converted to INT
 - xiv. SWITCH applyLeaveChoice
 1. CASE 1
 - a. CALL leaveCalander(&day, &month, &year)
 - b. OPENFILE "leaveDetails.txt" FOR READ

- c. IF “leaveDetails.txt” == NULL THEN
 - i. DISPLAY “File not found!\n”
 - ii. CALL staffHomepage(s_id_ptr)
- d. ENDIF
- e. WHILE NOT EOF “leaveDetails.txt” DO
 - i. READFILE “leaveDetails.txt”
 - ii. IF lv.leaveStaff_ID == *s_id_ptr AND lv.day == day AND lv.month == month AND lv.year == year AND lv.statusLeave != “Cancelled” THEN
 - 1. DISPLAY "Sorry, this is a duplicate leave.\n"
 - 2. CLOSEFILE “leaveDetails.txt”
 - 3. CALL staffHomepage(s_id_ptr)
 - iii. ENDIF
- f. ENDWHILE
- g. CLOSEFILE “leaveDetails.txt”
- h. CLEAR INPUT BUFFER
- i. DISPLAY "\nPlease enter the amount of days you would like to take leave for: \n"
- j. GET lv.duration
- k. OPENFILE “staffDetails.txt” FOR READ
- l. IF “staffDetails.txt” == NULL THEN
 - i. DISPLAY “File not found!\n”
 - ii. RETURN
- m. ENDIF
- n. OPENFILE “temp.txt” FOR WRITE
- o. IF “temp.txt” == NULL THEN
 - i. DISPLAY “File not found!\n”
 - ii. RETURN
- p. ENDIF
- q. WHILE NOT EOF “staffDetails.txt” DO
 - i. IF *s_id_ptr == st.staff_ID THEN
 - 1. IF st.annualLeave >= lv.duration THEN

- a. COMPUTE st.annualLeave -=
lv.duration
 - b. SET lv.leaveStaff_ID =
st.staff_ID
 - c. SET lv.typeLeave = "Annual
Leave"
 - d. SET lv.statusLeave = "Pending"
 - e. SET lv.day = day
 - f. SET lv.month = month
 - g. SET lv.year = year
 - h. OPENFILE "leaveDetails.txt"
FOR APPEND
 - i. IF "leaveDetails.txt" == NULL
THEN
 - i. DISPLAY "File not
found!\n"
 - ii. RETURN
 - j. ENDIF
 - k. WRITEFILE "leaveDetails.txt",
"%d, %s, %s, %d-%d-%d, %d
days\n", lv.leaveStaff_ID,
lv.typeLeave, lv.statusLeave,
lv.day, lv.month, lv.year,
lv.duration
 - l. CLOSEFILE "leaveDetails.txt"
 - m. DISPLAY "Leave applied!\n"
2. ELSE
- a. DISPLAY "Leave duration
exceeds balance!\n"
 - b. CLOSEFILE "leaveDetails.txt"
 - c. CLOSEFILE "staffDetails.txt"
 - d. CLOSEFILE "temp.txt"
 - e. REMOVE "temp.txt"
 - f. CALL staffHomepage(s_id_ptr)

- 3. ENDIF
- ii. ENDIF
- iii. WRITEFILE "temp.txt",
"%d, %s, %s, %s, %d, %d, %d, %d, %d\n",
st.staff_ID, st.staff_username, st.staff_password,
st.staff_type, st.annualLeave, st.compLeave,
st.emergLeave, st.matLeave, st.medLeave
- r. ENDWHILE
- s. CLOSEFILE "staffDetails.txt"
- t. CLOSEFILE "temp.txt"
- u. REMOVE "staffDetails.txt"
- v. RENAME "temp.txt" TO "staffDetails.txt"
- w. BREAK
- 2. ENDCASE
- 3. CASE 2
 - a. CALL leaveCalander(&day, &month, &year)
 - b. OPENFILE "leaveDetails.txt" FOR READ
 - c. IF "leaveDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. CALL staffHomepage(s_id_ptr)
 - d. ENDIF
 - e. WHILE NOT EOF "leaveDetails.txt" DO
 - i. READFILE "leaveDetails.txt"
 - ii. IF lv.leaveStaff_ID == *s_id_ptr AND lv.day ==
day AND lv.month == month AND lv.year ==
year AND lv.statusLeave != "Cancelled" THEN
 - 1. DISPLAY "Sorry, this is a duplicate
leave.\n"
 - 2. CLOSEFILE "leaveDetails.txt"
 - 3. CALL staffHomepage(s_id_ptr)
 - iii. ENDIF
 - f. ENDWHILE
 - g. CLOSEFILE "leaveDetails.txt"
 - h. CLEAR INPUT BUFFER

- i. DISPLAY "\nPlease enter the amount of days you would like to take leave for: \n"
- j. GET lv.duration
- k. OPENFILE "staffDetails.txt" FOR READ
- l. IF "staffDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
- m. ENDIF
- n. OPENFILE "temp.txt" FOR WRITE
- o. IF "temp.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
- p. ENDIF
- q. WHILE NOT EOF "staffDetails.txt" DO
 - i. IF *s_id_ptr == st.staff_ID THEN
 - 1. IF st.complLeave >= lv.duration THEN
 - a. COMPUTE st.complLeave -= lv.duration
 - b. SET lv.leaveStaff_ID = st.staff_ID
 - c. SET lv.typeLeave = "Compassionate Leave"
 - d. SET lv.statusLeave = "Pending"
 - e. SET lv.day = day
 - f. SET lv.month = month
 - g. SET lv.year = year
 - h. OPENFILE "leaveDetails.txt" FOR APPEND
 - i. IF "leaveDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - j. ENDIF


```

        i. DISPLAY "File not found!\n"
        ii. CALL staffHomepage(s_id_ptr)
d. ENDIF
e. WHILE NOT EOF "leaveDetails.txt" DO
    i. READFILE "leaveDetails.txt"
    ii. IF lv.leaveStaff_ID == *s_id_ptr AND lv.day ==
        day AND lv.month == month AND lv.year ==
        year AND lv.statusLeave != "Cancelled" THEN
        1. DISPLAY "Sorry, this is a duplicate
            leave.\n"
        2. CLOSEFILE "leaveDetails.txt"
        3. CALL staffHomepage(s_id_ptr)
    iii. ENDIF
f. ENDWHILE
g. CLOSEFILE "leaveDetails.txt"
h. CLEAR INPUT BUFFER
i. DISPLAY "\nPlease enter the amount of days you would
    like to take leave for: \n"
j. GET lv.duration
k. OPENFILE "staffDetails.txt" FOR READ
l. IF "staffDetails.txt" == NULL THEN
    i. DISPLAY "File not found!\n"
    ii. RETURN
m. ENDIF
n. OPENFILE "temp.txt" FOR WRITE
o. IF "temp.txt" == NULL THEN
    i. DISPLAY "File not found!\n"
    ii. RETURN
p. ENDIF
q. WHILE NOT EOF "staffDetails.txt" DO
    i. IF *s_id_ptr == st.staff_ID THEN
        1. IF st.emergLeave >= lv.duration THEN
            a. COMPUTE st.emergLeave -=
                lv.duration

```

- b. SET lv.leaveStaff_ID = st.staff_ID
- c. SET lv.typeLeave = "Emergency Leave"
- d. SET lv.statusLeave = "Pending"
- e. SET lv.day = day
- f. SET lv.month = month
- g. SET lv.year = year
- h. OPENFILE "leaveDetails.txt" FOR APPEND
- i. IF "leaveDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
- j. ENDIF
- k. WRITEFILE "leaveDetails.txt", "%d, %s, %s, %d-%d-%d, %d days\n", lv.leaveStaff_ID, lv.typeLeave, lv.statusLeave, lv.day, lv.month, lv.year, lv.duration
- l. CLOSEFILE "leaveDetails.txt"
- m. DISPLAY "Leave applied!\n"
- 2. ELSE
 - a. DISPLAY "Leave duration exceeds balance!\n"
 - b. CLOSEFILE "leaveDetails.txt"
 - c. CLOSEFILE "staffDetails.txt"
 - d. CLOSEFILE "temp.txt"
 - e. REMOVE "temp.txt"
 - f. CALL staffHomepage(s_id_ptr)
- 3. ENDIF
- ii. ENDIF

- iii. WRITEFILE "temp.txt",
"%d, %s, %s, %s, %d, %d, %d, %d, %d\n",
st.staff_ID, st.staff_username, st.staff_password,
st.staff_type, st.annualLeave, st.compLeave,
st.emergLeave, st.matLeave, st.medLeave
 - r. ENDWHILE
 - s. CLOSEFILE "staffDetails.txt"
 - t. CLOSEFILE "temp.txt"
 - u. REMOVE "staffDetails.txt"
 - v. RENAME "temp.txt" TO "staffDetails.txt"
 - w. BREAK
- 6. ENDCASE
- 7. CASE 4
 - a. CALL leaveCalander(&day, &month, &year)
 - b. OPENFILE "leaveDetails.txt" FOR READ
 - c. IF "leaveDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. CALL staffHomepage(s_id_ptr)
 - d. ENDIF
 - e. WHILE NOT EOF "leaveDetails.txt" DO
 - i. READFILE "leaveDetails.txt"
 - ii. IF lv.leaveStaff_ID == *s_id_ptr AND lv.day ==
day AND lv.month == month AND lv.year ==
year AND lv.statusLeave != "Cancelled" THEN
 - 1. DISPLAY "Sorry, this is a duplicate
leave.\n"
 - 2. CLOSEFILE "leaveDetails.txt"
 - 3. CALL staffHomepage(s_id_ptr)
 - iii. ENDIF
 - f. ENDWHILE
 - g. CLOSEFILE "leaveDetails.txt"
 - h. CLEAR INPUT BUFFER
 - i. DISPLAY "\nPlease enter the amount of days you would
like to take leave for: \n"

```

j.  GET lv.duration
k.  OPENFILE "staffDetails.txt" FOR READ
l.  IF "staffDetails.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
m.  ENDIF
n.  OPENFILE "temp.txt" FOR WRITE
o.  IF "temp.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
p.  ENDIF
q.  WHILE NOT EOF "staffDetails.txt" DO
    i.  IF *s_id_ptr == st.staff_ID THEN
        1. IF st.matLeave >= lv.duration THEN
            a. COMPUTE    st.matLeave    -=
                lv.duration
            b. SET      lv.leaveStaff_ID    =
                st.staff_ID
            c. SET lv.typeLeave = "Maternity
                Leave"
            d. SET lv.statusLeave = "Pending"
            e. SET lv.day = day
            f. SET lv.month = month
            g. SET lv.year = year
            h. OPENFILE    "leaveDetails.txt"
                FOR APPEND
            i. IF "leaveDetails.txt" == NULL
                THEN
                i.  DISPLAY    "File    not
                    found!\n"
                ii. RETURN
            j.  ENDIF
            k.  WRITEFILE    "leaveDetails.txt",
                "%d, %s, %s, %d-%d-%d, %d

```

```

                                days\n",          lv.leaveStaff_ID,
                                lv.typeLeave,       lv.statusLeave,
                                lv.day,           lv.month,       lv.year,
                                lv.duration
1.  CLOSEFILE "leaveDetails.txt"
m.  DISPLAY "Leave applied!\n"
2.  ELSE
    a.  DISPLAY "Leave duration
        exceeds balance!\n"
    b.  CLOSEFILE "leaveDetails.txt"
    c.  CLOSEFILE "staffDetails.txt"
    d.  CLOSEFILE "temp.txt"
    e.  REMOVE "temp.txt"
    f.  CALL staffHomepage(s_id_ptr)
3.  ENDIF
ii.  ENDIF
iii.  WRITEFILE "temp.txt",
        "%d, %s, %s, %s, %d, %d, %d, %d, %d\n",
        st.staff_ID, st.staff_username, st.staff_password,
        st.staff_type, st.annualLeave, st.compLeave,
        st.emergLeave, st.matLeave, st.medLeave
r.  ENDWHILE
s.  CLOSEFILE "staffDetails.txt"
t.  CLOSEFILE "temp.txt"
u.  REMOVE "staffDetails.txt"
v.  RENAME "temp.txt" TO "staffDetails.txt"
w.  BREAK
8.  ENDCASE
9.  CASE 5
    a.  CALL leaveCalander(&day, &month, &year)
    b.  OPENFILE "leaveDetails.txt" FOR READ
    c.  IF "leaveDetails.txt" == NULL THEN
        i.  DISPLAY "File not found!\n"
        ii. CALL staffHomepage(s_id_ptr)

```

```

d. ENDIF
e. WHILE NOT EOF "leaveDetails.txt" DO
    i. READFILE "leaveDetails.txt"
    ii. IF lv.leaveStaff_ID == *s_id_ptr AND lv.day ==
        day AND lv.month == month AND lv.year ==
        year AND lv.statusLeave != "Cancelled" THEN
        1. DISPLAY "Sorry, this is a duplicate
            leave.\n"
        2. CLOSEFILE "leaveDetails.txt"
        3. CALL staffHomepage(s_id_ptr)
    iii. ENDIF
f. ENDWHILE
g. CLOSEFILE "leaveDetails.txt"
h. CLEAR INPUT BUFFER
i. DISPLAY "\nPlease enter the amount of days you would
    like to take leave for: \n"
j. GET lv.duration
k. OPENFILE "staffDetails.txt" FOR READ
l. IF "staffDetails.txt" == NULL THEN
    i. DISPLAY "File not found!\n"
    ii. RETURN
m. ENDIF
n. OPENFILE "temp.txt" FOR WRITE
o. IF "temp.txt" == NULL THEN
    i. DISPLAY "File not found!\n"
    ii. RETURN
p. ENDIF
q. WHILE NOT EOF "staffDetails.txt" DO
    i. IF *s_id_ptr == st.staff_ID THEN
        1. IF st.medLeave >= lv.duration THEN
            a. COMPUTE    st.medLeave    -=
                lv.duration
            b. SET        lv.leaveStaff_ID    =
                st.staff_ID

```

```

c. SET lv.typeLeave = "Medical
   Leave"
d. SET lv.statusLeave = "Pending"
e. SET lv.day = day
f. SET lv.month = month
g. SET lv.year = year
h. OPENFILE    "leaveDetails.txt"
   FOR APPEND
i. IF "leaveDetails.txt" == NULL
   THEN
      i. DISPLAY    "File    not
                    found!\n"
      ii. RETURN
j. ENDIF
k. WRITEFILE    "leaveDetails.txt",
   "%d, %s, %s, %d-%d-%d, %d
   days\n",      lv.leaveStaff_ID,
   lv.typeLeave,   lv.statusLeave,
   lv.day,       lv.month,     lv.year,
   lv.duration
l. CLOSEFILE "leaveDetails.txt"
m. DISPLAY "Leave applied!\n"
2. ELSE
   a. DISPLAY    "Leave    duration
                 exceeds balance!\n"
   b. CLOSEFILE "leaveDetails.txt"
   c. CLOSEFILE "staffDetails.txt"
   d. CLOSEFILE "temp.txt"
   e. REMOVE "temp.txt"
   f. CALL staffHomepage(s_id_ptr)
3. ENDIF
ii. ENDIF
iii. WRITEFILE                                "temp.txt",
"%d, %s, %s, %s, %d, %d, %d, %d, %d\n",

```

```

st.staff_ID, st.staff_username, st.staff_password,
st.staff_type, st.annualLeave, st.compLeave,
st.emergLeave, st.matLeave, st.medLeave

```

- r. ENDFUNCTION
- s. CLOSEFILE "staffDetails.txt"
- t. CLOSEFILE "temp.txt"
- u. REMOVE "staffDetails.txt"
- v. RENAME "temp.txt" TO "staffDetails.txt"
- w. BREAK

10. ENDCASE

xv. ENDSWITCH

i. ENDFUNCTION

2. ENDFUNCTION

3.6.2 Viewing and Cancelling Leaves

1. FUNCTION summaryLeave (int *s_id_ptr)
 - a. SET INT matchFound = 0
 - b. SET INT i = 0
 - c. SET INT cancelled = 0
 - d. DECLARE STRUCT leaves as lv[sizeof(STRUCT leaves)]
 - e. DECLARE STRUCT staff as st
 - f. OPENFILE "leaveDetails.txt" FOR READ
 - g. IF "leaveDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. CALL staffHomepage(s_id_ptr)
 - h. ENDIF
 - i. REWINDFILE "leaveDetails.txt"
 - j. WHILE NOT EOF "leaveDetails.txt" DO
 - i. READ "leaveDetails.txt"
 - ii. IF *s_id_ptr == lv[i].leaveStaff_ID THEN
 1. SET matchFound = 1
 2. I++
 - iii. ENDIF
 - k. ENDFUNCTION

- l. CLOSEFILE "leaveDetails.txt"
- m. IF matchFound = 1 THEN
 - i. DECLARE INT leaveIndex
 - ii. DECLARE INT cancelSelect
 - iii. DECLARE INT numberLeave = 1
 - iv. DISPLAY "\n-----
- \n"
 - v. DISPLAY "NUMBER ID TYPE OF LEAVE LEAVE STATUS
DATE DURATION"
 - vi. DISPLAY "\n-----
_"
 - vii. FOR INT j = 0, j < i, j++ DO
 1. DISPLAY "\n%2d. %10d %18s %12s %5d-%d-%d %5d days\n",
numberLeave, lv[j].leaveStaff_ID, lv[j].typeLeave,
lv[j].statusLeave, lv[j].day, lv[j].month, lv[j].year, lv[j].duration
 2. numberLeave++
 - viii. ENDFOR
 - ix. CLEAR INPUT BUFFER
 - x. DISPLAY "\nWould you like to cancel a leave?\n"
 - xi. DISPLAY "1. Yes\n"
 - xii. DISPLAY "Press anything to go back\n"
 - xiii. GET cancelSelect
 - xiv. IF cancelSelect == 1 THEN
 1. CLEAR INPUT BUFFER
 2. DISPLAY "Enter the number you wish to cancel: \n"
 3. GET leaveIndex
 4. SET leaveIndex = leaveIndex – 1
 5. OPENFILE "staffDetails.txt" FOR READ
 6. IF "staffDetails.txt" == NULL THEN
 - a. DISPLAY "File not found!\n"
 - b. RETURN
 7. ENDIF
 8. OPENFILE "temp.txt" FOR WRITE
 9. IF "temp.txt" == NULL THEN

```

    a. DISPLAY "File not found!\n"
    b. RETURN
10. ENDIF
11. OPENFILE "leaveDetails.txt" FOR READ
12. IF "leaveDetails.txt" == NULL THEN
    a. DISPLAY "File not found!\n"
    b. RETURN
13. ENDIF
14. OPENFILE "templeave.txt" FOR WRITE
15. IF "templeave.txt" == NULL THEN
    a. DISPLAY "File not found!\n"
    b. RETURN
16. ENDIF
17. IF leaveIndex >= 0 AND leaveIndex < i THEN
    a. SET t = current time
    b. DECLARE STRUCT currentTime = localtime(current
        time)
    c. IF currentTime.tm_year + 1900 > lv[leaveIndex].year
        OR (currentTime.tm_year + 1900 ==
        lv[leaveIndex].year AND currentTime.tm_mon + 1 >
        lv[leaveIndex].month) OR (currentTime.tm_year + 1900
        == lv[leaveIndex].year AND currentTime.tm_mon + 1
        == lv[leaveIndex].month AND currentTime.tm_mday >
        lv[leaveIndex].day) THEN
        i. DISPLAY "Leave cannot be canceled because it
            has already passed.\n"
        ii. CLOSEFILE "staffDetails.txt"
        iii. CLOSEFILE "leaveDetails.txt"
        iv. REMOVE "temp.txt"
        v. REMOVE "templeave.txt"
        vi. CALL staffHomepage(s_id_ptr)
    d. ENDIF
    e. WHILE NOT EOF "staffDetails.txt" DO
        i. READ "staffDetails.txt"

```



```

ii. IF *s_id_ptr == st.staff_ID THEN
    1. IF lv[leaveIndex].statusLeave = "Pending"
        OR    lv[leaveIndex].statusLeave    =
            "Approved" THEN
        a. IF  lv[leaveIndex].typeLeave  =
            "Annual Leave" THEN
            i. COMPUTE
                st.annualLeave    +=
                lv[leaveIndex].duration
        b. ELSEIF
            lv[leaveIndex].typeLeave    =
            "Compassionate Leave" THEN
            i. COMPUTE st.compLeave
                +=
                lv[leaveIndex].duration
        c. ELSEIF
            lv[leaveIndex].typeLeave    =
            "Emergency Leave" THEN
            i. COMPUTE
                st.emergLeave    +=
                lv[leaveIndex].duration
        d. ELSEIF
            lv[leaveIndex].typeLeave    =
            "Maternity Leave" THEN
            i. COMPUTE  st.matLeave
                +=
                lv[leaveIndex].duration
        e. ELSEIF
            lv[leaveIndex].typeLeave    =
            "Medical Leave" THEN
            i. COMPUTE  st.medLeave
                +=
                lv[leaveIndex].duration

```

f. ENDIF

```

WHILE NOT EOF "leaveDetails.txt" DO
  DECLARE INT flag = 0
  FOR INT j = 0, j < i, INCREMENT j by 1 DO

    IF j == leaveIndex THEN

      IF lv[j].leaveStaff_ID == *s_id_ptr AND lv[j].statusLeave = "Pending"
      AND lv[j].typeLeave = lv[leaveIndex].typeLeave AND lv[i].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID
      AND lv[i].day == lv[leaveIndex].day AND lv[i].month == lv[leaveIndex].month AND lv[i].year == lv[leaveIndex].year
      AND lv[i].duration == lv[leaveIndex].duration THEN

        SET lv[j].statusLeave = "Cancelled"
        WRITEFILE "templeave.txt" "%d, %s, %s, %d-%d-%d, %d days\n", lv[j].leaveStaff_ID, lv[j].typeLeave, lv[j].statusLeave, lv[j].day, lv[j].month, lv[j].year, lv[j].duration
        CLEAR FILE BUFFER
        SET flag = 1

      ELSEIF lv[j].leaveStaff_ID == *s_id_ptr AND lv[j].statusLeave = "Approved"
      AND lv[j].typeLeave = lv[leaveIndex].typeLeave AND lv[i].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID
      AND lv[i].day == lv[leaveIndex].day AND lv[i].month == lv[leaveIndex].month AND lv[i].year == lv[leaveIndex].year
      AND lv[i].duration == lv[leaveIndex].duration THEN

        SET lv[j].statusLeave = "Cancelled"
        WRITEFILE "templeave.txt" "%d, %s, %s, %d-%d-%d, %d days\n", lv[j].leaveStaff_ID, lv[j].typeLeave, lv[j].statusLeave, lv[j].day, lv[j].month, lv[j].year, lv[j].duration
        CLEAR FILE BUFFER
        SET flag = 1

      ENDIF
    ENDIF
  ENDFOR
ENDWHILE

SET cancelled = 1

```

Figure 1: Continuation of pseudocode due to indentation limit.

2. ELSE

- a. DISPLAY "Cannot cancel leave
as leave is %s!\n",
lv[leaveIndex].statusLeave
- b. CLOSEFILE "leaveDetails.txt"
- c. CLOSEFILE "staffDetails.txt"
- d. REMOVE "temp.txt"
- e. REMOVE "templeave.txt"
- f. CALL staffHomepage(s_id_ptr)

3. ENDIF

iii. ENDIF

- iv. WRITEFILE "temp.txt"
"%d, %s, %s, %s, %d, %d, %d, %d, %d\n",
st.staff_ID, st.staff_username, st.staff_password,
st.staff_type, st.annualLeave, st.compLeave,
st.emergLeave, st.matLeave, st.medLeave

f. ENDWHILE

18. ELSE

- a. CLOSEFILE "leaveDetails.txt"
- b. CLOSEFILE "templeave.txt"
- c. CLOSEFILE "staffDetails.txt"
- d. CLOSEFILE "temp.txt"
- e. REMOVE "temp.txt"

```

        f. REMOVE "templeave.txt"
        g. DISPLAY "Invalid index.\n"
        h. CALL staffHomepage(s_id_ptr)
19. ENDIF
20. IF cancelled THEN
    a. CLOSEFILE "leaveDetails.txt"
    b. CLOSEFILE "templeave.txt"
    c. REMOVE "leaveDetails.txt"
    d. RENAME "templeave.txt" TO "leaveDetails.txt"
    e. DISPLAY "Leave successfully canceled!\n"
    f. CLOSEFILE "staffDetails.txt"
    g. CLOSEFILE "temp.txt"
    h. REMOVE "staffDetails.txt"
    i. RENAME "temp.txt" TO "staffDetails.txt"
    j. CALL staffHomepage(s_id_ptr)
21. ENDIF
xv. ELSE
    1. CLOSEFILE "leaveDetails.txt"
    2. CALL staffHomepage(s_id_ptr)
xvi. ENDIF
n. ELSE
    i. DISPLAY "No leaves applied!\n"
    ii. CLOSEFILE "leaveDetails.txt"
    iii. CALL staffHomepage(s_id_ptr)
o. ENDIF
2. ENDFUNCTION

```

3.6.3 Deleting Leaves

```

1. FUNCTION deleteLeave(int *s_id_ptr)
    a. SET INT matchFound = 0
    b. SET INT i = 0
    c. SET INT deleted = 0
    d. DECLARE STRUCT leaves as lv[sizeof(STRUCT leaves)]
    e. DECLARE STRUCT staff as st
    f. OPENFILE "leaveDetails.txt" FOR READ
    g. IF "leaveDetails.txt" == NULL THEN
        i. DISPLAY "File not found!\n"
        ii. CALL staffHomepage(s_id_ptr)
    h. ENDIF
    i. REWINDFILE "leaveDetails.txt"
    j. WHILE NOT EOF "leaveDetails.txt" DO
        i. READFILE "leaveDetails.txt"
        ii. IF *s_id_ptr == lv[i].leaveStaff_ID THEN
            1. SET matchFound = 1
            2. I++
        iii. ENDIF
    k. ENDWHILE
    l. CLOSEFILE "leaveDetails.txt"
    m. IF matchFound = 1 THEN
        i. DECLARE INT leaveIndex
        ii. DECLARE INT cancelSelect
        iii. SET INT numberLeave = 1
        iv. DISPLAY "\n-----
        ----\n"
        v. DISPLAY "NUMBER ID TYPE OF LEAVE LEAVE STATUS
        DATE DURATION"
        vi. DISPLAY "\n-----
        ----"
        vii. FOR INT j = 0, j < i, j++ DO

```

```

1. DISPLAY "\n%2d. %10d %18s %12s %5d-%d-%d %5d days\n",
   numberLeave,      lv[j].leaveStaff_ID,      lv[j].typeLeave,
   lv[j].statusLeave, lv[j].day, lv[j].month, lv[j].year, lv[j].duration
2. numberLeave++
viii. ENDFOR
ix.   CLEAR INPUT BUFFER
x.    DISPLAY "Enter the number you wish to delete, enter any character
       to go back to the homepage: \n"
xi.   GET leaveIndex
xii.  COMPUTE leaveIndex = leaveIndex - 1
xiii. OPENFILE "staffDetails.txt" FOR READ
xiv.  IF "staffDetails.txt" == NULL THEN
       1. DISPLAY "File not found!\n"
       2. RETURN
xv.   ENDIF
xvi.  OPENFILE "temp.txt" FOR WRITE+
xvii. IF "temp.txt" = NULL THEN
       1. DISPLAY "File not found!\n"
       2. RETURN
xviii. ENDFIF
xix.  OPENFILE "leaveDetails.txt" FOR READ
xx.   IF "leaveDetails.txt" == NULL THEN
       1. DISPLAY "File not found!\n"
       2. RETURN
xxi.  ENDFIF
xxii. OPENFILE "templeave.txt" FOR WRITE+
xxiii. IF "templeave.txt" == NULL THEN
       1. DISPLAY "File not found!\n"
       2. RETURN
xxiv. ENDFIF
xxv.  IF leaveIndex >= 0 AND leaveIndex < i THEN

```

```

WHILE NOT EOF "staffDetails.txt" DO
  IF *s_id_ptr == st.staff_ID THEN
    IF lv[leaveIndex].statusLeave = "Canceled" OR lv[leaveIndex].statusLeave = "Rejected"
      SET INT flag = 0
      WHILE NOT EOF "leaveDetails.txt" DO
        IF lv[i].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID AND lv[i].typeLeave = lv[leaveIndex].typeLeave AND
          lv[i].statusLeave = lv[leaveIndex].statusLeave AND lv[i].day == lv[leaveIndex].day AND lv[i].month == lv[leaveIndex].month
          AND lv[i].year == lv[leaveIndex].year AND lv[i].duration == lv[leaveIndex].duration THEN
          SET flag = 1
        ELSE
          WRITEFILE "templeave.txt" "%d, %s, %s, %d-%d-%d, %d days\n", lv[i].leaveStaff_ID,
            lv[i].typeLeave, lv[i].statusLeave, lv[i].day, lv[i].month, lv[i].year, lv[i].duration
        ENDIF
      ENDDO
      IF (flag != 1) THEN
        DISPLAY "Leave not found!\n"
      ENDIF
      SET deleted = 1
    ELSE
      DISPLAY "Cannot delete leave as leave is %s!\n", lv[leaveIndex].statusLeave
      CLOSEFILE "leaveDetails.txt"
      CLOSEFILE "staffDetails.txt"
      CLOSEFILE "temp.txt"
      CLOSEFILE "templeave.txt"
      REMOVE "temp.txt"
      REMOVE "templeave.txt"
      CALL staffHomepage(s_id_ptr)
    ENDIF
  ENDIF
  WRITEFILE "temp.txt" "%d, %s, %s, %d, %d, %d, %d, %d\n", st.staff_ID, st.staff_username, st.staff_password, st.staff_type,
    st.annualLeave, st.compleave, st.emergLeave, st.matLeave, st.medLeave
ENDDO

```

Figure 2: Continuation of pseudocode

- xxvi. ELSE
 1. CLOSEFILE "leaveDetails.txt"
 2. CLOSEFILE "staffDetails.txt"
 3. CLOSEFILE "temp.txt"
 4. CLOSEFILE "templeave.txt"
 5. REMOVE "temp.txt"
 6. REMOVE "templeave.txt"
 7. DISPLAY "Invalid index.\n"
- xxvii. ENDIF
- xxviii. IF deleted = 1 THEN
 1. CLOSEFILE "leaveDetails.txt"
 2. CLOSEFILE "templeave.txt"
 3. REMOVE "leaveDetails.txt"
 4. RENAME "templeave.txt" TO "leaveDetails.txt"
 5. DISPLAY "Leave successfully deleted!\n"
 6. CLOSEFILE "temp.txt"
 7. CLOSEFILE "staffDetails.txt"
 8. REMOVE "staffDetails.txt"
 9. RENAME "temp.txt" TO "staffDetails.txt"
 10. CALL staffHomepage(s_id_ptr)
- xxix. ENDIF
- n. ELSE
 - i. DISPLAY "No leaves applied!\n"

- ii. CLOSEFILE "leaveDetails.txt"
 - iii. CALL staffHomepage(s_id_ptr)
 - o. ENDIF
- 2. ENDFUNCTION

3.7 Admin Homepage Functions

3.7.1 Updating Leaves

1. FUNCTION updateLeaves
 - a. DECLARE CHAR buff[10]
 - b. DECLARE INT search_staffID
 - c. DECLARE INT updateLeaveType
 - d. SET INT i = 0
 - e. SET INT matchFound = 0
 - f. DECLARE STRUCT staff as st
 - g. SET st.staff_ID = 0
 - h. OPENFILE "staffDetails.txt" FOR READ
 - i. IF "staffDetailst.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - j. ENDIF
 - k. OPENFILE "temp.txt" FOR WRITE+
 - l. IF "temp.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - m. ENDIF
 - n. REWINDFILE "staffDetails.txt"
 - o. DISPLAY "\nSTAFF ID USERNAME TYPE OF STAFF ANNUAL
 COMPASSIONATE EMERGENCY MATERNITY MEDICAL"
 - p. DISPLAY "\n-----
 -----\n"
 - q. WHILE NOT EOF "%d, %[^,], %[^,], %[^,], %d, %d, %d, %d, %d",
 &st.staff_ID, st.staff_username, st.staff_password, st.staff_type,
 &st.annualLeave, &st.compLeave, &st.emergLeave, &st.matLeave,
 &st.medLeave DO
 - i. DISPLAY "%5d %12s %14s %10d %13d %13d %12d %11d\n",
 st.staff_ID, st.staff_username, st.staff_type, st.annualLeave,
 st.compLeave, st.emergLeave, st.matLeave, st.medLeave
 - r. ENDWHILE

- s. DISPLAY "Enter the Staff ID to update the leave balance, enter 1 to go back:
\n"
- t. GET search_staffID
- u. REWINDFILE "staffDetails.txt"
- v. WHILE NOT EOF "staffDetails.txt" DO
 - i. CLEAR INPUT BUFFER
 - ii. IF search_staffID != 1 THEN
 - 1. IF search_staffID == st.staff_ID THEN
 - a. SET matchFound = 1
 - b. DISPLAY "Please select which type of leave you would
like to update\n"
 - c. DISPLAY "1. Annual Leave\n"
 - d. DISPLAY "2. Compassionate Leave\n"
 - e. DISPLAY "3. Emergency Leave\n"
 - f. DISPLAY "4. Maternity Leave\n"
 - g. DISPLAY "5. Medical Leave\n"
 - h. DISPLAY "6. Back\n"
 - i. GET buff
 - j. SET updateLeaveType = buff converted to INT
 - k. SWITCH updateLeaveType
 - i. CASE 1
 - 1. DISPLAY "Enter the new amount of
leaves you would like to update it to: \n"
 - 2. GET st.annualLeave
 - ii. BREAK
 - iii. ENDCASE
 - iv. CASE 2
 - 1. DISPLAY "Enter the new amount of
leaves you would like to update it to: \n"
 - 2. GET st.compLeave
 - v. BREAK
 - vi. ENDCASE
 - vii. CASE 3

1. DISPLAY "Enter the new amount of leaves you would like to update it to: \n"
2. GET st.emergLeave
- viii. BREAK
- ix. ENDCASE
- x. CASE 4
 1. DISPLAY "Enter the new amount of leaves you would like to update it to: \n"
 2. GET st.matLeave
- xi. BREAK
- xii. ENDCASE
- xiii. CASE 5
 1. DISPLAY "Enter the new amount of leaves you would like to update it to: \n"
 2. GET st.medLeave
- xiv. BREAK
- xv. ENDCASE
- xvi. DEFAULT
 1. DISPLAY "Invalid option.\n"
- xvii. ENDCASE
1. ENDSWITCH
2. ENDIF
- iii. ENDIF
- iv. WRITEFILE "temp.txt" "%d, %s, %s, %s, %d, %d, %d, %d, %d\n",
 st.staff_ID, st.staff_username, st.staff_password, st.staff_type,
 st.annualLeave, st.compLeave, st.emergLeave, st.matLeave,
 st.medLeave
- w. ENDWHILE
- x. IF matchFound = 1 THEN
 - i. CLOSEFILE "staffDetails.txt"
 - ii. CLOSEFILE "temp.txt"
 - iii. REMOVE "staffDetails.txt"
 - iv. RENAME "temp.txt" TO "staffDetails.txt"
 - v. DISPLAY "Leave balance updated successfully!\n"

- y. ELSEIF search_staffID != 1 THEN
 - i. CLOSEFILE “staffDetails.txt”
 - ii. CLOSEFILE “temp.txt”
 - iii. REMOVE “temp.txt”
 - iv. DISPLAY "Staff ID not found!\n"
- z. ELSE
 - i. CLOSEFILE “staffDetails.txt”
 - ii. CLOSEFILE “temp.txt”
 - iii. REMOVE “temp.txt”
- aa. ENDIF
- bb. CALL adminHomepage()
- 2. ENDFUNCTION

3.7.2 Searching Staff Leaves

1. FUNCTION searchStaffLeaves()
 - a. SET INT matchFound = 0
 - b. SET INT i = 0
 - c. DECLARE INT searchInput
 - d. DECLARE STRUCT leaves as lv[100]
 - e. OPENFILE “leaveDetails.txt” FOR READ
 - f. IF “leaveDetails.txt” == NULL THEN
 - i. DISPLAY “File not found!\n”
 - ii. RETURN
 - g. ENDIF
 - h. REWINDFILE “leaveDetails.txt”
 - i. CLEAR INPUT BUFFER
 - j. DISPLAY "Please enter the staff ID to view their leave details: \n"
 - k. GET searchInput
 - l. WHILE NOT EOF “leaveDetails.txt” DO
 - i. READFILE “leaveDetails.txt”
 - ii. IF searchInput == lv[i].leaveStaff_ID THEN
 1. SET matchFound = 1
 2. i++
 - iii. ENDIF

```

m. ENDWHILE
n. IF matchFound = 1 THEN
    i. SET INT j = 0
    ii. DISPLAY "\n-----\n"
    iii. DISPLAY "ID      TYPE OF LEAVE    LEAVE STATUS    DATE
                DURATION"
    iv. DISPLAY "\n-----"
    v. FOR int j = 0, j < i, j++ DO
        1. DISPLAY "\n%2d %18s %12s %5d-%d-%d %5d days\n",
            lv[j].leaveStaff_ID, lv[j].typeLeave, lv[j].statusLeave, lv[j].day,
            lv[j].month, lv[j].year, lv[j].duration
    vi. ENDFOR
o. ELSE
    i. DISPLAY "Staff ID not found or leave not applied!\n"
    ii. RETURN
p. ENDIF
q. CLOSEFILE "leaveDetails.txt"
r. CALL adminHomepage()
2. ENDFUNCTION

```

3.7.3 Generating Monthly Report

```

1. FUNCTION generateReport()
    a. DECLARE srand(time(0))
    b. SET now = current time
    c. DECLARE STRUCT *current_time = localtime(current time)
    d. DECLARE STRUCT leaves as lv
    e. OPENFILE "leaveDetails.txt" FOR READ
    f. IF "leaveDetails.txt" == NULL THEN
        i. DISPLAY "File not found!\n"
        ii. RETURN
    g. ENDIF
    h. OPENFILE "ApuLeaveReport.txt" FOR WRITE+
    i. IF "ApuLeaveReport.txt" == NULL THEN
        i. DISPLAY "File not found!\n"

```

```

        ii. RETURN
    j. ENDIF
    k. DECLARE INT leave_count = 0
    l. WHILE NOT EOF "leaveDetails.txt" DO
        i. READFILE "leaveDetails.txt"
        ii. IF lv.month == current_time->tm_mon + 1 AND lv.year ==
            current_time->tm_year + 1900 THEN
            1. Leave_count++
        iii. ENDIF
    m. ENDWHILE
    n. WRITEFILE "ApuLeaveReport.txt" "*****MONTHLY REPORT
        FOR LEAVE APPLICATIONS*****\n"
    o. WRITEFILE "ApuLeaveReport.txt" "-----
        -----\n"
    p. WRITEFILE "ApuLeaveReport.txt" "Department/Faculty          Leaves
        Applied\n"
    q. WRITEFILE "ApuLeaveReport.txt" "-----
        -----\n"
    r. WRITEFILE "ApuLeaveReport.txt" "    Accounting              %-5d\n",
        rand() % (3 + 1)
    s. WRITEFILE "ApuLeaveReport.txt" "    Admin                  %-5d\n",
        rand() % (3 + 1)
    t. WRITEFILE "ApuLeaveReport.txt" "    Arts                    %-5d\n", rand() %
        (3 + 1)
    u. WRITEFILE "ApuLeaveReport.txt" "    Business                %-5d\n",
        rand() % (3 + 1)
    v. WRITEFILE "ApuLeaveReport.txt" "    Computing              %-5d\n",
        leave_count
    w. WRITEFILE "ApuLeaveReport.txt" "    Engineering            %-5d\n",
        rand() % (3 + 1)
    x. WRITEFILE "ApuLeaveReport.txt" "    Psychology             %-5d\n",
        rand() % (3 + 1)
    y. CLOSEFILE "ApuLeaveReport.txt"
    z. CLOSEFILE "leaveDetails.txt"

```

aa. DISPLAY "Report successfully generated!\n"

bb. CALL adminHomepage()

2. ENDFUNCTION

3.8 Approver Functions

3.8.1 Approving Leaves

1. FUNCTION searchStaffLeaves()
 - a. SET INT matchFound = 0
 - b. SET INT i = 0
 - c. SET INT approved = 0
 - d. SET INT numberLeave
 - e. DECLARE INT leaveIndex
 - f. DECLARE STRUCT leaves as lv[sizeof(STRUCT leaves)]
 - g. DECLARE STRUCT staff as st
 - h. OPENFILE "leaveDetails.txt" FOR READ
 - i. IF "leaveDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - j. ENDIF
 - k. REWINDFILE "leaveDetails.txt"
 - l. DISPLAY "\n-----\n"
 - m. DISPLAY "NUMBER ID TYPE OF LEAVE LEAVE STATUS DATE
DURATION"
 - n. DISPLAY "\n-----"
 - o. CLEAR INPUT BUFFER
 - p. DISPLAY "Please enter the staff ID to view their leave details: \n"
 - q. GET searchInput
 - r. WHILE NOT EOF "leaveDetails.txt" DO
 - i. READFILE "leaveDetails.txt"
 - ii. DISPLAY "\n%2d. %10d %18s %12s %5d-%d-%d %5d days\n",
numberLeave, lv[i].leaveStaff_ID, lv[i].typeLeave, lv[i].statusLeave,
lv[i].day, lv[i].month, lv[i].year, lv[i].duration
 - iii. i++
 - iv. numberLeave++
 - s. ENDWHILE
 - t. CLOSEFILE "leaveDetails.txt"
 - u. CLEAR INPUT BUFFER

```

v.  DISPLAY "\nWhich leave would you like to approve? Enter any character to
    go back to the homepage. \n"
w.  GET leaveIndex
x.  COMPUTE leaveIndex = leaveIndex - 1
y.  OPENFILE "staffDetails.txt" FOR READ
z.  IF "staffDetails.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
aa. ENDIF
bb. OPENFILE "temp.txt" FOR WRITE+
cc. IF "temp.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
dd. ENDIF
ee. OPENFILE "leaveDetails.txt" FOR READ
ff. IF "leaveDetails.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
gg. ENDIF
hh. OPENFILE "templeave.txt" FOR WRITE+
ii. IF "templeave.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
jj. ENDIF
kk. IF leaveIndex >= 0 AND leaveIndex < i THEN
    i.  SET t = current time
    ii. DECLARE STRUCT currentTime = *localtime(current time)
    iii. IF currentTime.tm_year + 1900 > lv[leaveIndex].year OR
        (currentTime.tm_year + 1900 == lv[leaveIndex].year AND
         currentTime.tm_mon + 1 > lv[leaveIndex].month) OR
        (currentTime.tm_year + 1900 == lv[leaveIndex].year AND
         currentTime.tm_mon + 1 == lv[leaveIndex].month AND
         currentTime.tm_mday > lv[leaveIndex].day) THEN

```


1. DISPLAY "Leave cannot be approved because it has already passed.\n"
2. CLOSEFILE "staffDetails.txt"
3. CLOSEFILE "leaveDetails.txt"
4. CLOSEFILE "temp.txt"
5. CLOSEFILE "templeave.txt"
6. REMOVE "temp.txt"
7. REMOVE "templeave.txt"
8. CALL approverHomepage()

iv. ENDIF

```

WHILE NOT EOF "staffDetails.txt" DO
  IF st.staff_ID == lv[leaveIndex].leaveStaff_ID THEN
    IF lv[leaveIndex].statusLeave = "Pending" OR lv[leaveIndex].statusLeave = "Rejected" THEN
      WHILE NOT EOF "leaveDetails.txt" DO
        SET INT duplicate = 0
        FOR int j = 0, j < i, j++ DO
          IF j == leaveIndex THEN
            IF lv[j].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID AND lv[j].statusLeave = "Pending" AND
              lv[j].typeLeave = lv[leaveIndex].typeLeave AND lv[j].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID AND
              lv[j].day == lv[leaveIndex].day AND lv[j].month == lv[leaveIndex].month AND lv[j].year == lv[leaveIndex].year
              AND lv[j].duration == lv[leaveIndex].duration THEN
              SET lv[j].statusLeave = "Approved"
              WRITEFILE templeave.txt "%d %s %s %d-%d-%d, %d days\n", lv[j].leaveStaff_ID, lv[j].typeLeave, lv[j].statusLeave,
                lv[j].day, lv[j].month, lv[j].year, lv[j].duration
              CLEAR FILE BUFFER
              SET duplicate = 1
            ELSEIF lv[j].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID AND lv[j].statusLeave = "Rejected" AND
              lv[j].typeLeave = lv[leaveIndex].typeLeave AND lv[j].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID AND
              lv[j].day == lv[leaveIndex].day AND lv[j].month == lv[leaveIndex].month AND lv[j].year == lv[leaveIndex].year
              AND lv[j].duration == lv[leaveIndex].duration THEN
              SET lv[j].statusLeave = "Approved"
          ELSE
            IF lv[leaveIndex].typeLeave = "Annual Leave" THEN
              IF st.annualLeave >= lv[leaveIndex].duration THEN
                COMPUTE st.annualLeave += lv[leaveIndex].duration
              ELSE
                CLOSEFILE "temp.txt"
                CLOSEFILE "templeave.txt"
                CLOSEFILE "leaveDetails.txt"
                CLOSEFILE "staffDetails.txt"
                REMOVE "temp.txt"
                REMOVE "templeave.txt"
                DISPLAY "Leave balance would be less than 0!\n"
                CALL approverHomepage()
              ENDIF
            ELSEIF lv[leaveIndex].typeLeave = "Compassionate Leave" THEN
              IF st.compLeave >= lv[leaveIndex].duration THEN
                COMPUTE st.compLeave += lv[leaveIndex].duration
              ELSE
                CLOSEFILE "temp.txt"
                CLOSEFILE "templeave.txt"
                CLOSEFILE "leaveDetails.txt"
                CLOSEFILE "staffDetails.txt"
                REMOVE "temp.txt"
                REMOVE "templeave.txt"
                DISPLAY "Leave balance would be less than 0!\n"
                CALL approverHomepage()
              ENDIF
            ELSE
              IF lv[leaveIndex].typeLeave = "Emergency Leave" THEN
                IF st.emergLeave >= lv[leaveIndex].duration THEN
                  COMPUTE st.emergLeave += lv[leaveIndex].duration
                ELSE
                  DISPLAY "Leave balance would be less than 0!\n"
                  CALL approverHomepage()
                ENDIF
              ENDIF
            ENDIF
          ENDIF
        ENDFOR
      ENDWHILE
    ENDIF
  ENDIF
ENDWHILE

```

Figure 3 Part 1: Continuation of pseudocode (outside previous if statement)

```

        IF lv[leaveIndex].typeLeave = "Emergency Leave" THEN
            IF st.emergLeave >= lv[leaveIndex].duration THEN
                COMPUTE st.emergLeave += lv[leaveIndex].duration
            ELSE
                CLOSEFILE "temp.txt"
                CLOSEFILE "templeave.txt"
                CLOSEFILE "leaveDetails.txt"
                CLOSEFILE "staffDetails.txt"
                REMOVE "temp.txt"
                REMOVE "templeave.txt"
                DISPLAY "Leave balance would be less than 0!\n"
                CALL approverHomepage()
            ENDIF
        IF lv[leaveIndex].typeLeave = "Maternity Leave" THEN
            IF st.matLeave >= lv[leaveIndex].duration THEN
                COMPUTE st.matLeave += lv[leaveIndex].duration
            ELSE
                CLOSEFILE "temp.txt"
                CLOSEFILE "templeave.txt"
                CLOSEFILE "leaveDetails.txt"
                CLOSEFILE "staffDetails.txt"
                REMOVE "temp.txt"
                REMOVE "templeave.txt"
                DISPLAY "Leave balance would be less than 0!\n"
                CALL approverHomepage()
            ENDIF
        IF lv[leaveIndex].typeLeave = "Medical Leave" THEN
            IF st.medLeave >= lv[leaveIndex].duration THEN
                COMPUTE st.medLeave += lv[leaveIndex].duration
            ELSE
                CLOSEFILE "temp.txt"
                CLOSEFILE "templeave.txt"
                CLOSEFILE "leaveDetails.txt"
                CLOSEFILE "staffDetails.txt"
                REMOVE "temp.txt"
                REMOVE "templeave.txt"
                DISPLAY "Leave balance would be less than 0!\n"
                CALL approverHomepage()
            ENDIF
        ENDIF
        WRITEFILE templeave.txt "%d, %s, %s, %d-%d-%d, %d days\n", lv[j].leaveStaff_ID, lv[j].typeLeave, lv[j].statusLeave,
        lv[j].day, lv[j].month, lv[j].year, lv[j].duration
        SET duplicate = 1
    ENDIF
ENDFOR
IF duplicate != 1 THEN
    WRITEFILE "temp.txt" "%d, %s, %s, %d-%d-%d, %d days\n", lv[i].leaveStaff_ID, lv[i].typeLeave, lv[i].statusLeave,
    lv[i].day, lv[i].month, lv[i].year, lv[i].duration
    CLEAR FILE BUFFER
ENDIF
ENDIF

```

Figure 4 Part 2: Continuation of pseudocode for approving leaves

```

        ENDIF
    ENDWHILE
    SET approved = 1
ELSE
    CLOSEFILE "temp.txt"
    CLOSEFILE "templeave.txt"
    CLOSEFILE "leaveDetails.txt"
    CLOSEFILE "staffDetails.txt"
    REMOVE "temp.txt"
    REMOVE "templeave.txt"
    DISPLAY "Cannot approve leave as leave is %s!\n", lv[leaveIndex].statusLeave
    CALL approverHomepage()
ENDIF
ENDIF
WRITEFILE "temp.txt" "%d, %s, %s, %d, %d, %d, %d, %d\n", st.staff_ID, st.staff_username,
staff_password, st.staff_type, st.annualLeave, st.compleave, st.emergLeave, st.matLeave, st.medLeave
ENDWHILE
CLOSEFILE "leaveDetails.txt"
CLOSEFILE "templeave.txt"
CLEAR FILE BUFFER
REMOVE "leaveDetails.txt"
RENAME "templeave.txt" TO "leaveDetails.txt"
CLOSEFILE "staffDetails.txt"
CLOSEFILE "temp.txt"
REMOVE "staffDetails.txt"
RENAME "temp.txt" TO "staffDetails.txt"

```

Figure 5 Part 3: Continuation of pseudocode for approving leaves

- ii. ELSE
 - i. CLOSEFILE "temp.txt"
 - ii. CLOSEFILE "templeave.txt"
 - iii. CLOSEFILE "leaveDetails.txt"
 - iv. CLOSEFILE "staffDetails.txt"
 - v. REMOVE "templeave.txt"
 - vi. REMOVE "temp.txt"
 - vii. DISPLAY "Invalid index.\n"
- mm. ENDIF
- nn. IF approved = 1 THEN
 - i. DISPLAY "Leave has been approved!\n"
- oo. ENDIF
- pp. CALL approverHomepage()
2. ENDFUNCTION

3.8.2 Rejecting Leaves

1. FUNCTION searchStaffLeaves()
 - a. SET INT matchFound = 0
 - b. SET INT i = 0
 - c. SET INT rejected = 0
 - d. SET INT numberLeave
 - e. DECLARE INT leaveIndex
 - f. DECLARE STRUCT leaves as lv[sizeof(STRUCT leaves)]
 - g. DECLARE STRUCT staff as st
 - h. OPENFILE "leaveDetails.txt" FOR READ
 - i. IF "leaveDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - j. ENDIF
 - k. REWINDFILE "leaveDetails.txt"
 - l. DISPLAY "\n-----\n"
 - m. DISPLAY "NUMBER ID TYPE OF LEAVE LEAVE STATUS DATE
DURATION"
 - n. DISPLAY "\n-----"
 - o. CLEAR INPUT BUFFER
 - p. DISPLAY "Please enter the staff ID to view their leave details: \n"
 - q. GET searchInput
 - r. WHILE NOT EOF "leaveDetails.txt" DO
 - i. READFILE "leaveDetails.txt"
 - ii. DISPLAY "\n%2d. %10d %18s %12s %5d-%d-%d %5d days\n",
numberLeave, lv[i].leaveStaff_ID, lv[i].typeLeave, lv[i].statusLeave,
lv[i].day, lv[i].month, lv[i].year, lv[i].duration
 - iii. i++
 - iv. numberLeave++
 - s. ENDWHILE
 - t. CLOSEFILE "leaveDetails.txt"
 - u. CLEAR INPUT BUFFER

```

v.  DISPLAY "\nWhich leave would you like to reject? Enter any character to go
    back to the homepage. \n"
w.  GET leaveIndex
x.  COMPUTE leaveIndex = leaveIndex - 1
y.  OPENFILE "staffDetails.txt" FOR READ
z.  IF "staffDetails.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
aa. ENDIF
bb. OPENFILE "temp.txt" FOR WRITE+
cc. IF "temp.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
dd. ENDIF
ee. OPENFILE "leaveDetails.txt" FOR READ
ff. IF "leaveDetails.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
gg. ENDIF
hh. OPENFILE "templeave.txt" FOR WRITE+
ii. IF "templeave.txt" == NULL THEN
    i.  DISPLAY "File not found!\n"
    ii. RETURN
jj. ENDIF
kk. IF leaveIndex >= 0 AND leaveIndex < i THEN
    i.  SET t = current time
    ii. DECLARE STRUCT currentTime = *localtime(current time)
    iii. IF currentTime.tm_year + 1900 > lv[leaveIndex].year OR
        (currentTime.tm_year + 1900 == lv[leaveIndex].year AND
         currentTime.tm_mon + 1 > lv[leaveIndex].month) OR
        (currentTime.tm_year + 1900 == lv[leaveIndex].year AND
         currentTime.tm_mon + 1 == lv[leaveIndex].month AND
         currentTime.tm_mday > lv[leaveIndex].day) THEN

```

1. DISPLAY "Leave cannot be rejected because it has already passed.\n"
2. CLOSEFILE "staffDetails.txt"
3. CLOSEFILE "leaveDetails.txt"
4. CLOSEFILE "temp.txt"
5. CLOSEFILE "templeave.txt"
6. REMOVE "temp.txt"
7. REMOVE "templeave.txt"
8. CALL approverHomepage()

iv. ENDIF

```

WHILE NOT EOF "staffDetails.txt" DO
  READ "staffDetails.txt"
  IF st.staff_ID == lv[leaveIndex].leaveStaff_ID THEN
    IF lv[leaveIndex].leaveStatus = "Pending" OR lv[leaveIndex].leaveStatus = "Approved" THEN
      IF lv[leaveIndex].typeLeave = "Annual Leave" THEN
        COMPUTE st.annualLeave += lv[leaveIndex].duration
      IF lv[leaveIndex].typeLeave = "Compassionate Leave" THEN
        COMPUTE st.compleave += lv[leaveIndex].duration
      IF lv[leaveIndex].typeLeave = "Emergency Leave" THEN
        COMPUTE st.emergLeave += lv[leaveIndex].duration
      IF lv[leaveIndex].typeLeave = "Maternity Leave" THEN
        COMPUTE st.matLeave += lv[leaveIndex].duration
      IF lv[leaveIndex].typeLeave = "Medical Leave" THEN
        COMPUTE st.medLeave += lv[leaveIndex].duration

    WHILE NOT EOF "leaveDetails.txt" DO
      READ "leaveDetails.txt"
      DECLARE INT duplicate = 0
      FOR INT j = 0, j < i, INCREMENT j by 1 DO

        IF j == leaveIndex THEN

          IF lv[j].leaveStaff_ID == *s_id_ptr AND lv[j].statusLeave = "Pending"
          AND lv[j].typeLeave = lv[leaveIndex].typeLeave AND lv[j].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID
          AND lv[j].day == lv[leaveIndex].day AND lv[j].month == lv[leaveIndex].month AND lv[j].year == lv[leaveIndex].year
          AND lv[j].duration == lv[leaveIndex].duration THEN

            SET lv[j].statusLeave = "Rejected"
            WRITEFILE "templeave.txt" "%d, %s, %s, %d-%d-%d, %d days\n", lv[j].leaveStaff_ID, lv[j].typeLeave, lv[j].statusLeave,
            lv[j].day, lv[j].month, lv[j].year, lv[j].duration
            CLEAR FILE BUFFER
            SET flag = 1

          ELSEIF lv[j].leaveStaff_ID == *s_id_ptr AND lv[j].statusLeave = "Approved"
          AND lv[j].typeLeave = lv[leaveIndex].typeLeave AND lv[j].leaveStaff_ID == lv[leaveIndex].leaveStaff_ID
          AND lv[j].day == lv[leaveIndex].day AND lv[j].month == lv[leaveIndex].month AND lv[j].year == lv[leaveIndex].year
          AND lv[j].duration == lv[leaveIndex].duration THEN

            SET lv[j].statusLeave = "Rejected"
            WRITEFILE "templeave.txt" "%d, %s, %s, %d-%d-%d, %d days\n", lv[j].leaveStaff_ID, lv[j].typeLeave, lv[j].statusLeave,
            lv[j].day, lv[j].month, lv[j].year, lv[j].duration
            CLEAR FILE BUFFER
            SET flag = 1

        ENDIF
      ENDIF
    ENDIF
  ENDIF

```

Figure 6 Part 1: Continuation of pseudocode for rejecting leaves.

```

      ENDIF
    ENDIF
  ENDIF
  IF duplicate != 1 THEN
    WRITEFILE "%d, %s, %s, %d-%d-%d, %d days\n", lv[i].leaveStaff_ID, lv[i].typeLeave, lv[i].statusLeave, lv[i].day,
    lv[i].month, lv[i].year, lv[i].duration
    CLEAR FILE BUFFER
  ENDIF
ENDWHILE
SET rejected = 1
ELSE
  CLOSEFILE "temp.txt"
  CLOSEFILE "templeave.txt"
  CLOSEFILE "leaveDetails.txt"
  CLOSEFILE "staffDetails.txt"
  REMOVE "templeave.txt"
  REMOVE "temp.txt"
  DISPLAY "Cannot reject leave as leave is %s!\n", lv[leaveIndex].statusLeave
  CALL approverHomepage()
ENDIF
ENDIF
WRITEFILE "temp.txt" "%d, %s, %s, %s, %d, %d, %d, %d, %d\n", st.staff_ID, st.staff_username, st.staff_password, st.staff_type, st.annualLeave,
st.compleave, st.emergLeave, st.matLeave, st.medLeave
ENDWHILE
CLOSEFILE "leaveDetails.txt"
CLOSEFILE "templeave.txt"
REMOVE "leaveDetails.txt"
RENAME "templeave.txt" TO "leaveDetails.txt"

CLOSEFILE "staffDetails.txt"
CLOSEFILE "temp.txt"
REMOVE "staffDetails.txt"
RENAME "temp.txt" TO "staffDetails.txt"

```

Figure 7 Part 2: Continuation of pseudocode for rejecting leaves.

II. ELSE

i. CLOSEFILE "temp.txt"

- ii. CLOSEFILE "templeave.txt"
- iii. CLOSEFILE "leaveDetails.txt"
- iv. CLOSEFILE "staffDetails.txt"
- v. REMOVE "templeave.txt"
- vi. REMOVE "temp.txt"
- vii. DISPLAY "Invalid index.\n"
- mm. ENDIF
- nn. IF approved = 1 THEN
 - i. DISPLAY "Leave has been rejected!\n"
- oo. ENDIF
- pp. CALL approverHomepage()
- 2. ENDFUNCTION

3.8.3 Searching Leave Dates

1. FUNCTION searchData()
 - a. DECLARE INT day
 - b. DECLARE INT month
 - c. DECLARE INT year
 - d. SET INT matchFound = 0
 - e. SET INT i = 0
 - f. DECLARE INT searchInput
 - g. DECLARE STRUCT leaves as lv[100]
 - h. WHILE TRUE DO
 - i. CLEAR INPUT BUFFER
 - ii. DISPLAY "\nPlease enter your starting day in DD format: \n"
 - iii. GET day
 - iv. CLEAR INPUT BUFFER
 - v. DISPLAY "Please enter your starting month in MM format: \n"
 - vi. GET month
 - vii. CLEAR INPUT BUFFER
 - viii. DISPLAY "Please enter your starting year in YYYY format: \n"
 - ix. GET year
 - x. IF *year >= 2023 AND *year <= 2025 THEN
 1. IF 1 <= *month OR *month <= 12 THEN
 - a. IF (1 <= *day AND *day <= 31) AND (*month == 1 OR *month == 3 OR *month == 5 OR *month == 7 OR

```

*month == 8 OR *month == 10 OR *month == 12 )
THEN
    i. DISPLAY "Valid date!\n"
    ii. BREAK
b. ELSEIF (1 <= *day AND *day <= 30) AND (*month ==
4 OR *month == 6 OR *month == 9 OR *month == 11)
THEN
    i. DISPLAY "Valid date!\n"
    ii. BREAK
c. ELSEIF (1 <= *day AND *day <= 28) AND (*month ==
2) THEN
    i. DISPLAY "Valid date!\n"
    ii. BREAK
d. ELSEIF (1 <= *day AND *day <= 29) AND (*year%4
== 0) THEN
    i. DISPLAY "Valid date!\n"
    ii. BREAK
e. ELSE
    i. DISPLAY "Invalid date!\n"
    ii. CONTINUE
f. ENDIF
2. ELSE
    a. DISPLAY "Invalid date!\n"
    b. CONTINUE
3. ENDIF
xi. ELSE
    1. DISPLAY "Invalid date!\n"
    2. CONTINUE
xii. ENDIF
i. ENDWHILE
j. OPENFILE "leaveDetails.txt" FOR READ
k. IF "leaveDetails.txt" == NULL THEN
    i. DISPLAY "File not found!\n"
l. ENDIF

```

```

m. REWIND "leaveDetails.txt"
n. CLEAR INPUT BUFFER
o. WHILE NOT EOF "leaveDetails.txt"
    i. READ "leaveDetails.txt"
    ii. IF lv[i].day == day AND lv[i].month == month AND lv[i].year == year
        THEN
            1. SET matchFound = 1
            2. I++
    iii. ENDIF
p. ENDWHILE
q. IF matchFound = 1 THEN
    i. SET INT j = 0
    ii. SET INT numberLeaveDate = 0
    iii. FOR j = 0, j < I, j++ DO
        1. numberLeaveDate++
    iv. ENDFOR
    v. DISPLAY "%d number of staff applied on that day.", numberLeaveDate
r. ELSE
    i. DISPLAY "No leaves applied on that day!\n"
    ii. CLOSEFILE "leaveDetails.txt"
    iii. RETURN
s. ENDIF
t. CLOSEFILE "leaveDetails.txt"
u. CALL approverHomepage()
2. ENDFUNCTION

```


3.9 All Homepages

3.9.1 Staff Homepage

1. FUNCTION staffHomepage(INT *s_id_ptr)
 - a. DECLARE INT sHomepage_choice
 - b. DECLARE CHAR buff[10]
 - c. DECLARE STRUCT leaves as lv
 - d. DECLARE STRUCT staff as st
 - e. WHILE TRUE DO
 - i. DISPLAY "\n-----\n"
 - ii. DISPLAY "***WELCOME TO THE STAFF HOMEPAGE***\n"
 - iii. DISPLAY "-----\n"
 - iv. DISPLAY "1. Apply Leaves\n"
 - v. DISPLAY "2. View No. of Leaves Available\n"
 - vi. DISPLAY "3. View/Cancel Applied Leaves\n"
 - vii. DISPLAY "4. Delete Leaves\n"
 - viii. DISPLAY "5. Logout\n"
 - ix. DISPLAY "Please enter your choice: \n"
 - x. CLEAR INPUT BUFFER
 - xi. GET buff
 - xii. SET sHomepage_choice = buff converted to INT
 - xiii. SWITCH sHomepage_choice
 1. CASE 1
 - a. CALL applyLeave(s_id_ptr)
 2. BREAK
 3. ENDCASE
 4. CASE 2
 - a. OPENFILE "staffDetails.txt" FOR READ
 - b. IF "staffDetails.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - c. ENDIF
 - d. WHILE NOT EOF "staffDetails.txt" DO
 - i. IF *s_id_ptr == st.staff_ID THEN

1. DISPLAY "\nAnnual Leave: %d\n",
st.annualLeave
2. DISPLAY "Compassionate Leave: %d\n",
st.compLeave
3. DISPLAY "Emergency Leave: %d\n",
st.emergLeave
4. DISPLAY "Maternity Leave: %d\n",
st.matLeave
5. DISPLAY "Medical Leave: %d\n",
st.medLeave
6. CONTINUE
- ii. ENDIF
- e. ENDWHILE
- f. CLOSEFILE "staffDetails.txt"
5. BREAK
6. ENDCASE
7. CASE 3
 - a. CALL summaryLeave(s_id_ptr)
8. BREAK
9. ENDCASE
10. CASE 4
 - a. CALL deleteLeave(s_id_ptr)
11. BREAK
12. ENDCASE
13. CASE 5
 - a. CALL menu()
14. BREAK
15. DEFAULT
 - a. DISPLAY "Invalid choice.\n"
 - b. CONTINUE
16. ENDCASE
- xiv. ENDSWITCH
- f. ENDWHILE
2. ENDFUNCTION

3.9.2 Admin Homepage

1. FUNCTION adminHomepage()
 - a. DECLARE INT admHomepage_choice
 - b. DECLARE INT staffORapp
 - c. DECLARE CHAR buff[10]
 - d. DECLARE CHAR dummy_password[20]
 - e. DECLARE INT type_staffCh
 - f. DECLARE STRUCT leaves as lv
 - g. DECLARE STRUCT staff as st
 - h. WHILE TRUE
 - i. DISPLAY "\n-----\n"
 - ii. DISPLAY "***WELCOME TO THE ADMIN HOMEPAGE***\n"
 - iii. DISPLAY "-----\n"
 - iv. DISPLAY "1. Add New Staff/Approver\n"
 - v. DISPLAY "2. Update Leave Balance\n"
 - vi. DISPLAY "3. Search Staff\n"
 - vii. DISPLAY "4. Generate Monthly Report\n"
 - viii. DISPLAY "5. Logout\n"
 - ix. DISPLAY "Please enter your choice: \n"
 - x. CLEAR INPUT BUFFER
 - xi. GET buff
 - xii. SET admHomepage_choice = buff converted to INT
 - xiii. SWITCH admHomepage_choice
 1. CASE 1
 - a. DISPLAY "Would you like to add a staff or an approver?\n"
 - b. WHILE TRUE DO
 - i. DISPLAY "1. Staff\n"
 - ii. DISPLAY "2. Approver\n"
 - iii. DISPLAY "3. Back\n"
 - iv. CLEAR INPUT BUFFER
 - v. GET buff
 - vi. SET staffORapp = buff converted to INT

vii. SWITCH staffORapp

```

CASE 1:
    SET fptr_staff = OPENFILE "staffDetails.txt" FOR APPEND+
    IF fptr_staff == NULL THEN
        DISPLAY "File not found!\n"
        RETURN
    ENDIF

    CLEAR INPUT BUFFER
    DISPLAY "which of the following staff are you?\n"
    DISPLAY "1. Academic\n"
    DISPLAY "2. Management\n"
    DISPLAY "3. Technical\n"
    DISPLAY "4. Administrative\n"
    GET type_staffCh

    IF type_staffCh == 1 THEN
        SET st.staff_type = "Academic"
    ELSEIF type_staffCh == 2 THEN
        SET st.staff_type = "Management"
    ELSEIF type_staffCh == 3 THEN
        SET st.staff_type = "Technical"
    ELSEIF type_staffCh == 4 THEN
        SET st.staff_type = "Administrative"
    ELSE
        DISPLAY "Invalid option.\n"
        RETURN
    ENDIF

    CLEAR INPUT BUFFER
    DISPLAY "Enter a username: \n"
    GET st.staff_username

    REWINDFILE "staffDetails.txt"
    DECLARE STRUCT staff as currSt
    WHILE NOT EOF "staffDetails.txt" DO
        READ "staffDetails.txt"
        IF currSt.staff_username = st.staff_username THEN
            DISPLAY "Username already exists, try a different one.\n"
            CALL adminHomepage()
        ENDIF
    ENDWHILE

    WHILE TRUE DO
        CLEAR INPUT BUFFER
        DISPLAY "Enter a password: \n"
        GET dummy_password

```

Figure 8 Part 1: Continuation of pseudocode.

```

    IF length of dummy_password < 8 OR length of dummy_password > 20 OR dummy_password has spaces THEN
        DISPLAY "Invalid password, must be between 8 and 20 characters and cannot contain spaces.\n"
        CONTINUE
    ELSE
        CLEAR INPUT BUFFER
        DISPLAY "Confirm your password: \n"
        GET st.staff_password
        IF st.staff_password = dummy_password THEN
            SET st.staff_ID = incrementOfStaffID()
            SET st.annualLeave = 16
            SET st.compleave = 7
            SET st.emergLeave = 3
            SET st.matLeave = 98
            SET st.medLeave = 22
            WRITEFILE "staffDetails.txt" "%d, %s, %s, %s, %d, %d, %d, %d\n", st.staff_ID, st.staff_username,
            st.staff_password, st.staff_type, st.annualLeave, st.compleave, st.emergLeave, st.matLeave, st.medLeave
            REWINDFILE "staffDetails.txt"
            DISPLAY "Staff registered successfully with ID %d.\n", st.staff_ID
            SET INT s_id_ptr = st.staff_ID
            CLOSEFILE "staffDetails.txt"
            DECLARE INT admStApplyLeave
            DISPLAY "would you like to apply for leave?\n"
            DISPLAY "1. Yes\n"
            DISPLAY "Anything else to go back to homepage\n"
            GET admStApplyLeave
            IF admStApplyLeave == 1 THEN
                CALL applyLeave(s_id_ptr)
            ELSE
                CALL adminHomepage()
            ENDIF
            BREAK
        ELSE
            DISPLAY "Passwords do not match!\n"
            CONTINUE
        ENDIF
    ENDIF
ENDWHILE
BREAK
ENDCASE
CASE 2:
    CALL registerApprover()
BREAK
ENDCASE
DEFAULT
    DISPLAY "Invalid choice.\n"
    CONTINUE
ENDCASE

```

Figure 9 Part 2: Continuation of pseudocode.

viii. ENDSWITCH

- c. ENDWHILE
- 2. BREAK
- 3. ENDCASE
- 4. CASE 2:
 - a. CALL updateLeaves()
- 5. BREAK
- 6. ENDCASE
- 7. CASE 3
 - a. CALL searchStaffDates()
- 8. BREAK
- 9. ENDCASE
- 10. CASE 4
 - a. CALL generateReport()
- 11. BREAK
- 12. ENDCASE
- 13. CASE 5
 - a. CALL menu()
- 14. BREAK
- 15. ENDCASE
- 16. DEFAULT
 - a. DISPLAY "Invalid choice.\n"
- 17. CONTINUE
- 18. ENDCASE
- xiv. ENDSWITCH
- i. ENDWHILE
- 2. ENDFUNCTION

3.9.3 Approver Homepage

1. FUNCTION approverHomepage()
 - a. DECLARE INT appHomepage_choice
 - b. DECLARE CHAR buff[10]
 - c. WHILE TRUE DO
 - i. DISPLAY "\n-----\n"
 - ii. DISPLAY "***WELCOME TO THE APPROVER
HOMEPAGE***\n"
 - iii. DISPLAY "-----\n"
 - iv. DISPLAY "1. Approve Leaves\n"
 - v. DISPLAY "2. Reject Leaves\n"
 - vi. DISPLAY "3. Search Leave Dates\n"
 - vii. DISPLAY "4. Logout\n"
 - viii. DISPLAY "Please enter your choice: \n"
 - ix. CLEAR INPUT BUFFER
 - x. GET buff
 - xi. SET appHomepage_choice = buff converted to INT
 - xii. SWITCH appHomepage_choice
 1. CASE 1
 - a. CALL approveLeaves()
 2. BREAK
 3. ENDCASE
 4. CASE 2
 - a. CALL rejectLeaves()
 5. BREAK
 6. ENDCASE
 7. CASE 3
 - a. CALL searchDate()
 8. BREAK
 9. ENDCASE
 10. CASE 4
 - a. CALL menu()
 11. BREAK

- 12. ENDCASE
- 13. DEFAULT
 - a. DISPLAY "Invalid choice.\n"
- 14. CONTINUE
- 15. ENDCASE
- xiii. ENDSWITCH
- d. ENDWHILE
- 2. ENDFUNCTION

3.9.4 Menu

- 1. FUNCTION menu()
 - a. DECLARE INT menu_choice
 - b. DECLARE CHAR buff[10]
 - c. WHILE TRUE DO
 - i. DISPLAY "\n-----\n"
 - ii. DISPLAY "****WELCOME TO THE APU LEAVE MANAGEMENT
SYSTEM****\n"
 - iii. DISPLAY "-----\n"
 - iv. DISPLAY "1. Admin Login\n"
 - v. DISPLAY "2. Staff Login\n"
 - vi. DISPLAY "3. Staff Registration\n"
 - vii. DISPLAY "4. Approver Login\n"
 - viii. DISPLAY "5. Feedback\n"
 - ix. DISPLAY "6. Terms & Conditions\n"
 - x. DISPLAY "7. Exit\n"
 - xi. DISPLAY "Please enter your choice: \n"
 - xii. CLEAR INPUT BUFFER
 - xiii. GET buff
 - xiv. SET menu_choice = buff converted to INT
 - xv. SWITCH menu_choice
 - 1. CASE 1
 - a. CALL adminLogin()
 - 2. BREAK
 - 3. ENDCASE

4. CASE 2
 - a. CALL staffLogin()
5. BREAK
6. ENDCASE
7. CASE 3
 - a. CALL registerStaff()
8. BREAK
9. ENDCASE
10. CASE 4
 - a. CALL approverLogin()
11. BREAK
12. ENDCASE
13. CASE 5
 - a. CALL SYSTEM OPEN BROWSER LINK
https://docs.google.com/forms/d/e/1FAIpQLSfuXAv-LaBdEyGC7hbACPg-wiv7yBt1cTk0svqLSBQcHI-ygw/viewform?usp=sf_link
14. CONTINUE
15. ENDCASE
16. CASE 6
 - a. DECLARE CHAR buffer[255]
 - b. OPENFILE "readme.txt" FOR READ
 - c. IF "readme.txt" == NULL THEN
 - i. DISPLAY "File not found!\n"
 - ii. RETURN
 - d. ENDIF
 - e. WHILE NOT EOF "readme.txt" DO
 - i. DISPLAY "\n%s", buffer
 - f. ENDWHILE
 - g. CLOSEFILE "readme.txt"
17. CONTINUE
18. ENDCASE
19. CASE 7
 - a. EXIT PROGRAM

20. DEFAULT

a. DISPLAY "Invalid choice.\n"

21. CONTINUE

22. ENDCASE

xvi. ENDSWITCH

d. ENDWHILE

2. ENDFUNCTION

3.10 Main Function

1. FUNCTION main()

a. CALL menu()

b. RETURN 0

2. ENDFUNCTION

4.0 Additional Features of the Program

4.1 Feedback Form


The first additional feature of the program would be its feedback system. This system was designed for users to fill in and submit their feedback on how the program can be improved even further. The feedback form includes 4 questions which all pertain to how the user thinks the system can be further improved in the future. To access this function, the user just needs to start the program as the function is located in the main menu. Then, the user has to enter '5', and by doing so, the system will redirect the user to the feedback form in their main browser.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
5|
```

Figure 10: Accessing the feedback function.

APU Leave System Feedback

Please enter your feedback here. User responses will be anonymous.

 [Switch accounts](#)

*Required

How easy is it to navigate around the system? *

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

How efficient is the system when it comes to managing leaves? *

1 2 3 4 5

☐ ☐ ☐ ☐ ☐

Figure 11: The feedback form.

4.2 Terms & Conditions

The second additional function of the program would be its terms and conditions function. This function serves to provide the guidelines and the policies of the program so that the user has more information and is educated on how the website works and what can and cannot be done in this system. It provides grounded rules too such as warning the user against falsifying their leave approvals and informing the user that they cannot cancel their leaves when the applied date has already past or is the current date. To access this function, users just need to start the program and enter '6'.

```

-----
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
6|

```

Figure 12: Accessing the Terms & Conditions.

```

1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
6

1. Eligibility: The leave management system is only available to faculty and staff members of the university who are authorized to take leave. Students are not eligible to use this system.

2. Leave Types: The system supports various types of leaves, including annual leave, compassionate leave, emergency leave, maternity leave, and medical leave types as per the university policy. If you would like to apply for a paternity leave, please visit the admin or email the admin via admin@apu.edu.my.

3. Approval Process: All leave requests must be approved by the appropriate authority. The system will route the request to the relevant person for approval based on the type of leave and the length of absence.

4. Accrual and Balance: The system will automatically calculate the balance and accrual of leaves based on the type of leave. Faculty and staff members can view their leave balances and history in the system.

5. Leave Cancellation: Faculty and staff members can cancel their leave requests in the system before the leave start date. However, cancellation of leave requests after the start date will not be allowed.

6. Record Keeping: The system maintains an accurate record of all leave transactions, including leave requests, approvals, rejections, and cancellations.

7. Misuse of the System: Misuse of the leave management system, including falsifying leave records or approving one's leave request, may result in disciplinary action.

```

Figure 13: The Terms & Conditions being displayed to the user.

4.3 Delete Leaves Function

The final additional function is exclusive to the staff only, and that is the delete leaves function. With this function, the staffs can clear their cancelled and/or rejected leaves if they choose to do so in order to make their leave history and summary look neater. To access this function, the user needs to login as a staff into the system by entering '2' in the main menu and entering the correct credentials for their corresponding staff.

```

-----
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
2
Enter your Staff ID:
|

```

Figure 14: Logging in as a staff from the main menu.

Once logged in, the user has to input '4' from the staff homepage. If the staff has applied for leaves, they can enter the function, however, if they have not applied for leaves, the system will inform them that they have no leaves applied before redirecting them back to the staff homepage.

```

-----
***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Delete Leaves
5. Logout
Please enter your choice:
4
No leaves applied!

-----
***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Delete Leaves
5. Logout
Please enter your choice:
|

```

Figure 15: If there are no leaves that were applied, the system will redirect the staff back to their homepage.

However, if they have applied for leaves, the system will allow them to delete them provided if the leave status is either 'Cancelled' or 'Rejected' only. Otherwise, the system will tell the staff that the leave cannot be deleted because by logic, the staff has no right to delete a leave that has been approved already and a leave that is currently pending.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	5-3-2023	10 days
2.	101	Compassionate Leave	Approved	4-3-2023	10 days

Enter the number you wish to delete, enter any character to go back to the homepage:
1
Cannot delete leave as leave is Pending!

Figure 16: The staff cannot delete the leave if it is either cancelled or rejected.

If the leave's status is 'Cancelled' or 'Rejected', the leave will be deleted from their history.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	5-3-2023	10 days
2.	101	Medical Leave	Canceled	3-2-2023	10 days
3.	101	Compassionate Leave	Approved	4-3-2023	10 days

Enter the number you wish to delete, enter any character to go back to the homepage:
2
Leave successfully deleted!

WELCOME TO THE STAFF HOMEPAGE

1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Delete Leaves
5. Logout
Please enter your choice:
4

The leave has been deleted when viewing all the leaves

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	5-3-2023	10 days
2.	101	Compassionate Leave	Approved	4-3-2023	10 days

Enter the number you wish to delete, enter any character to go back to the homepage:
|

Figure 17: The leave has been deleted.

5.0 Sample Inputs and Outputs

The following segment will cover the inputs and outputs of the system, including its error validations. It will be split up into segments and the different users in order so as to make it neater.

5.1 Menu

The first page the user sees when they run the system would be the main menu, where the user can choose whether to login as an admin, staff or approver, register as a staff, read the terms and conditions, submit a feedback or exit the system.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
|
```

Figure 18: The menu.

The menu and all the other menus in the system are coded using a switch statement, which is a more efficient form of an if-else statement specifically catered towards menu-based systems as its syntax is easier to read and write compared to the if-else statements (Programiz, n.d.). To enter a page, the user just has to enter the number under the “Please enter your choice” statement which corresponds to the page. For example, if the user wishes to login as a staff, they just need to enter ‘2’. However, if the user tries to enter anything other than the numbers presented on the screen, the program will issue its default statement from the switch statement. The switch statement will then repeat itself so that the program does not end then and there. This logic will follow throughout the entire program as the majority of the system is menu-based.

```

***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
8
Invalid choice.

```

Figure 19: The default statement will be executed if the user enters an invalid number/data type.

The first 4 functions will be discussed in the proceeding segments, and even though the feedback and terms and conditions functions were discussed earlier in the Additional Functions segment of the report, I will still discuss it here.

5.1.1 Feedback Form

To access the feedback form, the user has to enter '5' in the main menu. The system will then redirect the user to their main browser to fill up a form regarding the feedback of the system. There, users can answer what can be improved upon and give suggestions of what other functions can be added into the system.

Figure 20: The feedback form. This only shows 2 of 4 questions in the form.

5.1.2 Terms & Conditions

If the user wishes to view the terms and conditions as well as the policies and guidelines of the system, they may do so in the main menu. All they have to do is enter '6' to view it.

```

1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
6

1. Eligibility: The leave management system is only available to faculty and staff members of the university who are authorized to take leave. Students are not eligible to use this system.

2. Leave Types: The system supports various types of leaves, including annual leave, compassionate leave, emergency leave, maternity leave, and medical leave types as per the university policy. If you would like to apply for a paternity leave, please visit the admin or email the admin via admin@apu.edu.my.

3. Approval Process: All leave requests must be approved by the appropriate authority. The system will route the request to the relevant person for approval based on the type of leave and the length of absence.

4. Accrual and Balance: The system will automatically calculate the balance and accrual of leaves based on the type of leave. Faculty and staff members can view their leave balances and history in the system.

5. Leave Cancellation: Faculty and staff members can cancel their leave requests in the system before the leave start date. However, cancellation of leave requests after the start date will not be allowed.

6. Record Keeping: The system maintains an accurate record of all leave transactions, including leave requests, approvals, rejections, and cancellations.

7. Misuse of the System: Misuse of the leave management system, including falsifying leave records or approving one's leave request, may result in disciplinary action.

```

Figure 21: The Terms & Conditions.

5.1.3 Exit Function

If the user wishes to exit the system, they may enter '7' to end the process.

```

-----
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----

1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
7

Process returned 0 (0x0)   execution time : 712.330 s
Press any key to continue.

```

Figure 22: Exiting the system.

5.2 Administrator/Super Administrator

5.2.1 Logging in as an Admin

If the user is a system administrator, the user has to enter '1' from the main menu in order to login as one. There, the user has to enter their username and password. If the credentials are correct, the user can login to the homepage. However, if the credentials do not match, the system will ask the user to retry upwards to 2 more times as this is a security procedure.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
1  
Enter username:  
not the correct username  
Enter password:  
this isn't the correct password either  
Login failed. 2 attempts left.
```

Figure 23: Inputting the incorrect credentials when logging in. If the user fails to input the correct credentials, they will have up to 2 more attempts left before they are booted back to the main menu.

If the user inputs the incorrect credentials 3 times, the system will kick the user back to the main menu.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
1  
Enter username:  
what username did i have again  
Enter password:  
what do you mean password??  
Login failed. 2 attempts left.  
Enter username:  
there's an attempt system?  
Enter password:  
oh no... i'm not an admin so i don't know any of these credentials...  
Login failed. 1 attempts left.  
Enter username:  
oh god, i hope i get this right  
Enter password:  
it's my last try after all  
Login failed. 0 attempts left.  
No attempts left. Returning to main menu...
```

Figure 24: The user being booted back to the main menu after using all of their attempts.

However, if the user inputs the correct credentials, they can enter the homepage.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
1  
Enter username:  
admin  
Enter password:  
password  
Login successful! Redirecting to the Admin homepage...
```

Figure 25: By inputting the correct login credentials, the user can access the homepage.

In the admin homepage, the admin has a number of functions presented to them in a menu-based style.

```

-----
***WELCOME TO THE ADMIN HOMEPAGE***
-----
1. Add New Staff/Approver
2. Update Leave Balance
3. Search Staff
4. Generate Monthly Report
5. Logout
Please enter your choice:
|

```

Figure 26: The admin homepage.

5.2.2 Adding New Staffs/Approvers

To add a new staff or approver, the admin has to input '1'. There, they have the choice of choosing whether it's a staff or approver which they want to add. They may also input '3' there to go back to the admin homepage.

```

-----
***WELCOME TO THE ADMIN HOMEPAGE***
-----
1. Add New Staff/Approver
2. Update Leave Balance
3. Search Staff
4. Generate Monthly Report
5. Logout
Please enter your choice:
1
Would you like to add a staff or an approver?
1. Staff
2. Approver
3. Back

```

Figure 27: Adding a new staff or approver.

5.2.2.1 New Staff

By inputting '1' to add a new staff, the administrator must first choose which role they would like to assign to for the staff. Then, the administrator has to enter a name, password and finally confirm the password. By fulfilling all of the above requirements, the new staff will be added with their staff ID, assigned name, password, role and of course, all of their leave balances. The staff ID which is assigned to the staff starts from 100 and will be incremented by 1 every time a new staff is added to the system, making each staff ID unique for each staff. After this,

the system will ask the administrator whether they would like to apply for leave for the staff. They can enter '1' to apply for leave and anything else to go back to the admin homepage. The leave application process will be discussed in the upcoming staff section.

```
Which of the following staff are you?
1. Academic
2. Management
3. Technical
4. Administrative
1
Enter a username:
tom
Enter a password:
password
Confirm your password:
password
Staff registered successfully with ID 103.
Would you like to apply for leave?
1. Yes
Anything else to go back to homepage
|
```

Figure 28: The administrator successfully adding a new staff to the system.

However, if the inputted name already exists in the database, the system will inform the administrator that the name has already existed and they will have to repeat the whole process again.

```
Which of the following staff are you?
1. Academic
2. Management
3. Technical
4. Administrative
1
Enter a username:
staff
Username already exists, try a different one.
```

Figure 29: If a username already exists, the system will inform the administrator about it and they will have to redo the whole registration process again.

Similarly, if the system detects any of the scenarios listed below:

1. If the password does not match with each other when confirming the password.
2. If the password is lesser than 8 characters long or has more than 20 characters.
3. If the password contains any spaces.

Then the system will ask the administrator to re-enter a password and confirm it.

```
Which of the following staff are you?
1. Academic
2. Management
3. Technical
4. Administrative
1
Enter a username:
tom
Enter a password:
password
Confirm your password:
notpassword
Passwords do not match!
Enter a password:
|
```

Figure 30: In this case, the passwords do not match when confirming them, so the system will ask the administrator to repeat the password adding process.

These validations and processes will be followed in the staff and approver registration process too, the only difference between registering a staff as an admin and anyone but an admin is that by registering for a staff as an administrator, the system will automatically ask if you would like to apply for a leave. Hence under any other circumstances, the system will not present the user that function.

5.2.2.2 New Approver

Registering a new approver is the same as registering for a new staff. The administrator has to enter a name and a password before confirming the password. If the name already exists in the database, which is the text file, the system will reject the name and will ask the administrator to repeat the registration process. Similarly, if the system detects any of the scenarios listed below:

1. If the password does not match with each other when confirming the password.
2. If the password is lesser than 8 characters long or has more than 20 characters.
3. If the password contains any spaces.

Then the system will ask the administrator to re-enter a password and confirm it.

```

Would you like to add a staff or an approver?
1. Staff
2. Approver
3. Back
2
Enter a username:
michael
Enter a password:
password
Confirm your password:
password
Approver registered successfully with ID 203.

```

Figure 31: The administrator successfully adding an approver to the system.

However, the unique thing about the approver registration function is that only the administrator can add the approver, the staff or even the approver themselves cannot add an approver to the system as the university is wary of users registering themselves as approvers to approve their own leaves and/or reject other staff's leaves, effectively abusing the approver's abilities. The other unique feature which differs the staffs from the approvers is the fact that the staff's ID starts from 100 whereas the approver's ID starts from 200. The final unique feature which differs the approvers from the staffs is the fact that the approvers do not have leaves themselves as they only serve to approve or reject leaves, not apply for them.

5.2.3 Updating Leave Balance

The next function that the administrator has access to is the update leave balance function, which corresponds to '2' in the admin homepage. This is where the administrator can select the particular staff and select which type of leave that they would like to update to. For example, if the administrator wishes to update the staff with the staff ID of 101, the administrator can simply just enter '101' into the column. If they wish to go back to the admin homepage however, they can enter '1'.

STAFF ID	USERNAME	TYPE OF STAFF	ANNUAL	COMPASSIONATE	EMERGENCY	MATERNITY	MEDICAL
101	staff	Academic	50	10010	3	98	22
102	staff 2	Technical	1000	7	3	98	22
103	tom	Academic	16	7	3	98	22

Enter the Staff ID to update the leave balance, enter 1 to go back:
101|

Figure 32: If the administrator wishes to update the leave balance of a staff, they will have to enter the staff's corresponding ID.

Once they enter the staff ID, the system will ask the administrator to enter which leave that they would like to update via a menu-based system. Finally, the administrator will enter the amount of leaves that they would like to update that staff's particular leave balance to.

STAFF ID	USERNAME	TYPE OF STAFF	ANNUAL	COMPASSIONATE	EMERGENCY	MATERNITY	MEDICAL
101	staff	Academic	50	10010	3	98	22
102	staff 2	Technical	1000	7	3	98	22
103	tom	Academic	16	7	3	98	22

Enter the Staff ID to update the leave balance, enter 1 to go back:
101
Please select which type of leave you would like to update
1. Annual Leave
2. Compassionate Leave
3. Emergency Leave
4. Maternity Leave
5. Medical Leave
6. Back
1
Enter the new amount of leaves you would like to update it to:
100|

Figure 33: Updating staff ID 101's annual leave balance to 100.

The leave balance for that staff's corresponding leave type will then be updated in the text file and subsequently in the program as well.

Enter the new amount of leaves you would like to update it to:
100
Leave balance updated successfully!

WELCOME TO THE ADMIN HOMEPAGE

1. Add New Staff/Approver
2. Update Leave Balance
3. Search Staff
4. Generate Monthly Report
5. Logout
Please enter your choice:
2

STAFF ID	USERNAME	TYPE OF STAFF	ANNUAL	COMPASSIONATE	EMERGENCY	MATERNITY	MEDICAL
101	staff	Academic	100	10010	3	98	22
102	staff 2	Technical	1000	7	3	98	22
103	tom	Academic	16	7	3	98	22

Enter the Staff ID to update the leave balance, enter 1 to go back:
|

Figure 34: The updated leave balance is reflected and shown in real time.

However, if the administrator enters in a staff ID which does not exist, the system will say that the staff ID does not exist and will redirect the administrator to the admin homepage.

STAFF ID	USERNAME	TYPE OF STAFF	ANNUAL	COMPASSIONATE	EMERGENCY	MATERNITY	MEDICAL
101	staff	Academic	30	10010	3	98	22
102	staff 2	Technical	1000	670	3	98	22
103	tom	Academic	16	7	3	98	22

Enter the Staff ID to update the leave balance, enter 1 to go back:
 asdouabso
 Staff ID not found!

 WELCOME TO THE ADMIN HOMEPAGE

 1. Add New Staff/Approver
 2. Update Leave Balance
 3. Search Staff
 4. Generate Monthly Report
 5. Logout
 Please enter your choice:
 |

Figure 35: The system redirects the administrator to the admin homepage if the staff ID is not found.

5.2.4 Searching Staff Leave Details

The third function that the administrator has exclusive access to is that they can view the staff's leaves just by searching their staff ID. This function lies in the "Search Staff" function, which is initiated by entering '3' in the admin homepage. When the administrator enters the function, they will need to key in the staff ID to see the staff's corresponding leave.

 WELCOME TO THE ADMIN HOMEPAGE

 1. Add New Staff/Approver
 2. Update Leave Balance
 3. Search Staff
 4. Generate Monthly Report
 5. Logout
 Please enter your choice:
 3
 Please enter the staff ID to view their leave details:
 101

ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
101	Annual Leave	Pending	15-2-2023	10 days
101	Annual Leave	Rejected	30-4-2023	10 days
101	Compassionate Leave	Canceled	5-6-2023	10 days
101	Annual Leave	Canceled	4-5-2023	10 days

Figure 36: The Search Staff function in action.

However, if the staff ID does not exist or the staff ID's staff has not applied for any leave, the system will say that the staff ID is not found or there is no leave applied before redirecting the administrator to the admin homepage. In this case, we are going to test using the staff that we

just added into the system earlier with the staff ID of 103. They have not applied for any leave yet.

```

-----
***WELCOME TO THE ADMIN HOMEPAGE***
-----
1. Add New Staff/Approver
2. Update Leave Balance
3. Search Staff
4. Generate Monthly Report
5. Logout
Please enter your choice:
3
Please enter the staff ID to view their leave details:
103
Staff ID not found or leave not applied!

-----
***WELCOME TO THE ADMIN HOMEPAGE***
-----
1. Add New Staff/Approver
2. Update Leave Balance
3. Search Staff
4. Generate Monthly Report
5. Logout
Please enter your choice:
|

```

Figure 37: What would happen if the staff ID has not applied for any leave or there is no such staff ID that exists.

5.2.5 Generating Monthly Report

The last function that the admin has exclusive access to is that they can generate the monthly leave statistics in a text file. As mentioned earlier in the Assumptions chapter, the upper bound for all the schools excluding the School of Computing will be 3 with its lowest bound being 0. However because this system's example focuses on the School of Computing, the School of Computing's leave statistics will be reflective of the applied leaves. For instance, the current month is February and the current year is 2023 at the time of writing this. So, only the leaves that were applied in February will be displayed in the monthly leave report for the School of Computing.

To access this function, the administrator has to enter '4' in the admin homepage. By doing so, the system will write the report and inform the administrator that the report has been generated.

```

-----
***WELCOME TO THE ADMIN HOMEPAGE***
-----
1. Add New Staff/Approver
2. Update Leave Balance
3. Search Staff
4. Generate Monthly Report
5. Logout
Please enter your choice:
4
Report successfully generated!

```

Figure 38: Generating the report.

For context, there are only 2 leaves that are applied in the current month and year, as shown in the text file which acts as our leave database.

```

leaveDetails - Notepad
File Edit View

101, Annual Leave, Pending, 15-2-2023, 10 days
101, Annual Leave, Rejected, 30-4-2023, 10 days
102, Maternity Leave, Approved, 20-2-2023, 10 days
101, Compassionate Leave, Canceled, 5-6-2023, 10 days
101, Annual Leave, Canceled, 4-5-2023, 10 days

```

Figure 39: The leaves that are applied in February are underlined with red ink.

So, the monthly leave report should only display 2 under the School of Computing.

*****MONTHLY REPORT FOR LEAVE APPLICATIONS*****	
Department/Faculty	Leaves Applied
Accounting	0
Admin	0
Arts	3
Business	0
Computing	2
Engineering	2
Psychology	2

Figure 40: The monthly leave report reflects the leave applications for the School of Computing in real time.

5.2.6 Logout

The final function which the administrator can do is to logout. Once they enter '5' in the admin homepage, the administrator will be redirected to the main menu.

```

-----
***WELCOME TO THE ADMIN HOMEPAGE***
-----
1. Add New Staff/Approver
2. Update Leave Balance
3. Search Staff
4. Generate Monthly Report
5. Logout
Please enter your choice:
5

-----
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
|

```

Figure 41: The administrator logs out.

5.3 Staff

5.3.1 Registering as a Staff

To register as a staff, the user has to enter '3' in the main menu of the program. There, the user can assign their role, add their name, password and confirm their password.

```

-----
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
3
Which of the following staff are you?
1. Academic
2. Management
3. Technical
4. Administrative
2
Enter a username:
jan
Enter a password:
password
Confirm your password:
password
Staff registered successfully with ID 104.

```

Figure 42: Registering a new staff.

However, if the staff tries to register with a username which has already been registered, the system will reject it and it will redirect the user to the main menu.

```

*****WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM*****
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
3
Which of the following staff are you?
1. Academic
2. Management
3. Technical
4. Administrative
1
Enter a username:
staff
Username already exists, try a different one.

```

Figure 43: The system will inform the user that the username already exists.

Similarly, if the system detects any of the scenarios below:

1. If the password does not match with each other when confirming the password.
2. If the password is lesser than 8 characters long or has more than 20 characters.
3. If the password contains any spaces.

Then the system will ask the user to re-enter a password and confirm it.

```

*****WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM*****
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
3
Which of the following staff are you?
1. Academic
2. Management
3. Technical
4. Administrative
2
Enter a username:
another staff
Enter a password:
password
Confirm your password:
notpassword
Passwords do not match!
Enter a password:
|

```

Figure 44: In this case, the user entered a password which does not match when they confirm the password, so the system prompts the user to re-enter the password.

5.3.2 Logging in as a Staff

To login as a staff, the user has to input '2' in the main menu. There, the user is prompted to enter the valid credentials for the staff. If the user enters the correct credentials, they will be redirected to the staff homepage.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
2  
Enter your Staff ID:  
101  
Enter username:  
staff  
Enter password:  
password  
Login successful! Redirecting to the staff homepage...  
  
-----  
***WELCOME TO THE STAFF HOMEPAGE***  
-----  
1. Apply Leaves  
2. View No. of Leaves Available  
3. View/Cancel Applied Leaves  
4. Logout  
Please enter your choice:  
|
```

Figure 45: If the user enters the correct credentials for that staff, the user will be redirected to the staff homepage.

However, the login system has an attempt system; meaning that the user is given 3 chances to input the correct credentials. If they fail to input the valid credentials 3 times, the system will boot the user back to the main menu as a security feature.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
2  
Enter your Staff ID:  
oasidn  
Enter username:  
asodin  
Enter password:  
asdojn  
Login failed. 2 attempts left.  
Enter your Staff ID:  
asodin  
Enter username:  
asdipn  
Enter password:  
asdpin  
Login failed. 1 attempts left.  
Enter your Staff ID:  
asoidn  
Enter username:  
alsidn  
Enter password:  
asdpin  
Login failed. 0 attempts left.  
No attempts left. Returning to main menu...
```

Figure 46: Failing to login 3 times will initiate the system to kick the user back to the main menu.

5.3.3 Applying Leaves

To apply for leaves, the staff has to input '1' in the staff homepage. And by doing so, the staff is greeted with a menu-based system where they can choose which type of leave they would like to apply for.

```
-----  
***WELCOME TO THE STAFF HOMEPAGE***  
-----  
1. Apply Leaves  
2. View No. of Leaves Available  
3. View/Cancel Applied Leaves  
4. Logout  
Please enter your choice:  
1  
  
-----  
***WELCOME TO THE APU LEAVE APPLICATION SYSTEM***  
-----  
1. Annual Leave  
2. Compassionate Leave  
3. Emergency Leave  
4. Maternity Leave  
5. Medical Leave  
6. Back  
Please enter your choice:  
|
```

Figure 47: The leave application system.

Applying for all types of leaves are the same, so for this report, I will only show how the application process will be like for only one type of leave, which is the Annual Leave. To apply for the leaves, the user has to enter the number which corresponds to the type of leave that they would like to apply for, in this case since it's the Annual Leave, it will be '1'.

In here, the system will prompt the user to enter a day in DD format, a month in MM format and a year in YYYY format. If it is a valid date, the system will ask the user to enter the amount of days that they would like to take leave for. Finally, after the user inputs the amount of days, the system will issue them a notice that their leave has been applied for and the staff will be redirected back to the staff homepage. The leave application is the same for the admin after they successfully register a staff as they follow the same logic and code.

```

*****WELCOME TO THE APU LEAVE APPLICATION SYSTEM*****
*****
1. Annual Leave
2. Compassionate Leave
3. Emergency Leave
4. Maternity Leave
5. Medical Leave
6. Back
Please enter your choice:
1

Please enter your starting day in DD format:
5
Please enter your starting month in MM format:
9
Please enter your starting year in YYYY format:
2023
Valid date!

Please enter the amount of days you would like to take leave for:
10
Leave applied!

```

Figure 48: Successfully applying for a leave.

However, if the system detects one of the following scenarios:

1. If the leave year is lesser than 2023 or more than 2025.
2. If it is an invalid leave date or if the date does not exist, such as 31st April or 32nd December, or if the year is not a leap year and the date applied is 29th February.
3. If the current date has already past the leave's date.
4. If the data input for the leave date is incorrect, such as strings, floats, doubles or chars and negative numbers or zeroes.

Then the system will prompt the user to re-enter a valid date.

```

*****WELCOME TO THE APU LEAVE APPLICATION SYSTEM*****
*****
1. Annual Leave
2. Compassionate Leave
3. Emergency Leave
4. Maternity Leave
5. Medical Leave
6. Back
Please enter your choice:
1

Please enter your starting day in DD format:
30
Please enter your starting month in MM format:
2
Please enter your starting year in YYYY format:
2023
Invalid date!

Please enter your starting day in DD format:
|

```

Figure 49: In the scenario above, the staff attempted to enter a date which does not exist, so the system prompts the user to re-enter a valid date.

Similarly, if the system detects one of the following scenarios:

1. If the leave date already exists and the staff ID of the leave is the same as the current staff and the user attempts to apply for the same date.
2. If the staff's leave balance is not enough to apply for the leave, for example applying for a 10-day leave when you only have 7 days left in your leave balance.

Then the system will redirect the user to the staff homepage.

```

Please enter your starting day in DD format:
5
Please enter your starting month in MM format:
9
Please enter your starting year in YYYY format:
2023
Valid date!

Please enter the amount of days you would like to take leave for:
10
Leave applied!    1st Application (5-9-2023)

-----
***WELCOME TO THE APU LEAVE APPLICATION SYSTEM***
-----
1. Annual Leave
2. Compassionate Leave
3. Emergency Leave
4. Maternity Leave
5. Medical Leave
6. Back
Please enter your choice:
1
                2nd Application (5-9-2023)
Please enter your starting day in DD format:
5
Please enter your starting month in MM format:
9
Please enter your starting year in YYYY format:
2023
Valid date!
Sorry, this is a duplicate leave.

```

Figure 50: In this scenario, the staff tries to enter a duplicate date, so the system redirects the user to the staff homepage.

However, do keep in mind that different staff IDs can apply for the same leave date.

```

***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
2
Enter your Staff ID:
102
Enter username:
staff 2
Enter password:
password
Login successful! Redirecting to the staff homepage...

-----
***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
1

-----
***WELCOME TO THE APU LEAVE APPLICATION SYSTEM***
-----
1. Annual Leave
2. Compassionate Leave
3. Emergency Leave
4. Maternity Leave
5. Medical Leave
6. Back
Please enter your choice:
1
Please enter your starting day in DD format:
5
Please enter your starting month in MM format:
9
Please enter your starting year in YYYY format:
2023
Valid date!

Please enter the amount of days you would like to take leave for:

```

Staff ID 102, it was 101 earlier

Applying for leave on 5-9-2023 too

Figure 51: As long as the staff IDs are different, other staffs can apply for an existing leave date.

5.3.4 Checking Leave Balance

To check for the leave balance, the staff can enter '2' in the staff homepage. The system will then display all of the available leaves for that staff, which is updated in real time.

```

*****
***WELCOME TO THE STAFF HOMEPAGE***
*****
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
2

Annual Leave: 30
Compassionate Leave: 10010
Emergency Leave: 3
Maternity Leave: 98
Medical Leave: 22

```

Figure 52: Viewing the leave balance.

5.3.5 Viewing Leave Summary

The leave summary is combined with the cancel leave function since they can be executed in order, but I will discuss them separately in this report. To view the applied leaves for that staff, they can enter '3' in the staff homepage. There, the system will display the leaves history for that staff which includes the type of leave applied, leave status, date of leave and duration of leave. Exiting this function will be discussed in the next segment which is cancelling the leaves as they are tied together.

```

*****
***WELCOME TO THE STAFF HOMEPAGE***
*****
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
3

-----
NUMBER      ID      TYPE OF LEAVE      LEAVE STATUS      DATE      DURATION
-----
1.          101      Annual Leave      Pending          15-2-2023      10 days
2.          101      Annual Leave      Rejected         30-4-2023      10 days
3.          101      Compassionate Leave      Canceled         5-6-2023      10 days
4.          101      Annual Leave      Canceled         4-5-2023      10 days
5.          101      Annual Leave      Pending          5-9-2023      10 days

```

Figure 53: Viewing the summary of leaves applied.

5.3.6 Cancelling Leaves

To cancel a leave, the staff has to input '3' in the staff homepage. There, the system will display all of the leaves applied by that staff. The staff has a choice of choosing whether they would like to cancel the leave or not, if they would like to cancel the leave, they would just need to input '1', otherwise, the system will redirect the user to the staff homepage.

```

-----
***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
3
-----

```

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Rejected	30-4-2023	10 days
3.	101	Compassionate Leave	Canceled	5-6-2023	10 days
4.	101	Annual Leave	Canceled	4-5-2023	10 days
5.	101	Annual Leave	Pending	5-9-2023	10 days

```

Would you like to cancel a leave?
1. Yes
Press anything to go back
|

```

Figure 54: The user can choose whether to cancel a leave or not.

By inputting '1', the system will ask the staff which leave they would like to cancel. All the staff has to do is to input the number which corresponds to the leave. If the input is valid, the system will cancel the leave.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Rejected	30-4-2023	10 days
3.	101	Compassionate Leave	Canceled	5-6-2023	10 days
4.	101	Annual Leave	Canceled	4-5-2023	10 days
5.	101	Annual Leave	Pending	5-9-2023	10 days

Would you like to cancel a leave?
 1. Yes
 Press anything to go back
 1
 Enter the number you wish to cancel:
 5
 Leave successfully canceled!

Figure 55, Part 1: Cancelling a leave.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Rejected	30-4-2023	10 days
3.	101	Compassionate Leave	Canceled	5-6-2023	10 days
4.	101	Annual Leave	Canceled	4-5-2023	10 days
5.	101	Annual Leave	Canceled	5-9-2023	10 days

Would you like to cancel a leave?
 1. Yes
 Press anything to go back
 |

Figure 56 Part 2: The leave has successfully been cancelled.

After cancelling the leave, the staff's leave balance for that type of leave will be refunded back based on the duration of the cancelled leave.

```

***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
2

Annual Leave: 30
Compassionate Leave: 10010
Emergency Leave: 3
Maternity Leave: 98
Medical Leave: 22
  
```

```

***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
2

Annual Leave: 40
Compassionate Leave: 10010
Emergency Leave: 3
Maternity Leave: 98
Medical Leave: 22
  
```

Figure 57: Before and after cancelling the leave. The cancelled leave's duration was 10 days and it was an annual leave, hence why the annual leave has 10 added to it in the staff's leaves balance.

Additionally after cancelling that leave, that same staff can apply for a leave on the same date as the cancelled leave again. But keep in mind that this can only happen if the leave has been cancelled first.

```

***WELCOME TO THE APU LEAVE APPLICATION SYSTEM***
-----
1. Annual Leave
2. Compassionate Leave
3. Emergency Leave
4. Maternity Leave
5. Medical Leave
6. Back
Please enter your choice:
1

Please enter your starting day in DD format:
5
Please enter your starting month in MM format:
9
Please enter your starting year in YYYY format:
2023
Valid date!

Please enter the amount of days you would like to take leave for:
|
  
```

Figure 58: The staff can apply for a leave on the 5th of September 2023 after cancelling a leave of the same date.

However, if the system detects that the leave's applied date has already passed, then the staff will not be able to cancel the leave. For the scenario below, I will try cancelling the leave with

a date of 15th of February 2023. For context, the current date at the time of writing this is 17th February 2023.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Rejected	30-4-2023	10 days
3.	101	Compassionate Leave	Canceled	5-6-2023	10 days
4.	101	Annual Leave	Canceled	4-5-2023	10 days
5.	101	Annual Leave	Canceled	5-9-2023	10 days

Would you like to cancel a leave?
 1. Yes
 Press anything to go back
 1
 Enter the number you wish to cancel:
 1
 Leave cannot be canceled because it has already passed.

Figure 59: The system does not allow the leave to be cancelled as the date has already passed.

Similarly, the system will detect if the staff tries to cancel a leave that has already been cancelled or rejected. if they try to do so, the system will not accept that and will redirect the staff to the staff homepage.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Rejected	30-4-2023	10 days
3.	101	Compassionate Leave	Canceled	5-6-2023	10 days
4.	101	Annual Leave	Canceled	4-5-2023	10 days
5.	101	Annual Leave	Canceled	5-9-2023	10 days

Would you like to cancel a leave?
 1. Yes
 Press anything to go back
 1
 Enter the number you wish to cancel:
 5
 Cannot cancel leave as leave is Canceled!

Figure 60: If the leave status is either cancelled or rejected, the system will redirect the user to the staff homepage.

5.3.7 Deleting Leaves

The final exclusive function the staff has access to is the delete leaves function. With this function, the staffs can clear their cancelled and/or rejected leaves if they choose to do so in order to make their leave history and summary look neater. To access this function, the staff has to input '4' from the staff homepage.

WELCOME TO THE STAFF HOMEPAGE					
1. Apply Leaves					
2. View No. of Leaves Available					
3. View/Cancel Applied Leaves					
4. Delete Leaves					
5. Logout					
Please enter your choice:					
4					
NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	5-3-2023	10 days
2.	101	Compassionate Leave	Approved	4-3-2023	10 days
Enter the number you wish to delete, enter any character to go back to the homepage:					

Figure 61: Accessing the delete leaves function from the staff homepage.

If the staff has applied for leaves, they can enter the function, however, if they have not applied for leaves, the system will inform them that they have no leaves applied before redirecting them back to the staff homepage.

```
-----  
***WELCOME TO THE STAFF HOMEPAGE***  
-----  
1. Apply Leaves  
2. View No. of Leaves Available  
3. View/Cancel Applied Leaves  
4. Delete Leaves  
5. Logout  
Please enter your choice:  
4  
No leaves applied!  
  
-----  
***WELCOME TO THE STAFF HOMEPAGE***  
-----  
1. Apply Leaves  
2. View No. of Leaves Available  
3. View/Cancel Applied Leaves  
4. Delete Leaves  
5. Logout  
Please enter your choice:  
|
```

Figure 62: If there are no leaves that were applied, the system will redirect the staff back to their homepage.

To delete a leave, the staff has to input the number corresponding to the leave that is displayed which they wish to delete, just like the cancel leave function. If the staff wishes to exit this function, they can input any character, the system will then redirect the staff back to the staff homepage. Likewise, if the staff enters a number which does not correspond to any leave which is currently being displayed on screen, the system will recognize it as being an invalid index and will also redirect the staff back to the homepage.

```

-----
NUMBER      ID      TYPE OF LEAVE      LEAVE STATUS      DATE      DURATION
-----
1.          101      Annual Leave      Pending          5-3-2023      10 days
2.          101      Compassionate Leave      Approved          4-3-2023      10 days
Enter the number you wish to delete, enter any character to go back to the homepage:
3
Invalid index.

-----
***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Delete Leaves
5. Logout
Please enter your choice:
|

```

Figure 63: Entering an invalid number/data type will make the system redirect the staff back to the staff homepage.

If the staff has applied for leaves and inputs a valid number, the system will allow them to delete them provided if the leave status is either 'Cancelled' or 'Rejected' only. Otherwise, the system will tell the staff that the leave cannot be deleted because by logic, the staff has no right to delete a leave that has been approved already and a leave that is currently pending.

```

-----
NUMBER      ID      TYPE OF LEAVE      LEAVE STATUS      DATE      DURATION
-----
1.          101      Annual Leave      Pending          5-3-2023      10 days
2.          101      Compassionate Leave      Approved          4-3-2023      10 days
Enter the number you wish to delete, enter any character to go back to the homepage:
1
Cannot delete leave as leave is Pending!

```

Figure 64: The staff cannot delete the leave if it is either cancelled or rejected.

If the leave's status is 'Cancelled' or 'Rejected', the leave will be deleted from their history.

```

-----
NUMBER      ID      TYPE OF LEAVE      LEAVE STATUS      DATE      DURATION
-----
1.          101      Annual Leave      Pending          5-3-2023      10 days
2.          101      Medical Leave      Canceled          3-2-2023      10 days
3.          101      Compassionate Leave      Approved          4-3-2023      10 days
Enter the number you wish to delete, enter any character to go back to the homepage:
2
Leave successfully deleted!

-----
***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Delete Leaves
5. Logout
Please enter your choice:
4

-----
NUMBER      ID      TYPE OF LEAVE      LEAVE STATUS      DATE      DURATION
-----
1.          101      Annual Leave      Pending          5-3-2023      10 days
2.          101      Compassionate Leave      Approved          4-3-2023      10 days
Enter the number you wish to delete, enter any character to go back to the homepage:
|

```

The leave has been deleted when viewing all the leaves

Figure 65: The leave has been deleted.

5.3.8 Logout

To logout as a staff, the staff needs to input '4' in the staff homepage. They will then be redirected to the main menu.

```
-----  
***WELCOME TO THE STAFF HOMEPAGE***  
-----  
1. Apply Leaves  
2. View No. of Leaves Available  
3. View/Cancel Applied Leaves  
4. Logout  
Please enter your choice:  
4  
  
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
|
```

Figure 66: Logging out as a staff.

5.4 Approver

5.4.1 Logging in as an Approver

To login as an approver, the user has to enter '4' in the main menu. There, the user will be prompted to enter the correct credentials which include the approver ID, corresponding name and corresponding password. If all the credentials match each other, the approver will be redirected to the approver homepage.

```
-----  
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***  
-----  
1. Admin Login  
2. Staff Login  
3. Staff Registration  
4. Approver Login  
5. Feedback  
6. Terms & Conditions  
7. Exit  
Please enter your choice:  
4  
Enter your Approver ID:  
201  
Enter username:  
approver  
Enter username:  
password  
Login successful! Redirecting to the approver homepage...  
  
-----  
***WELCOME TO THE APPROVER HOMEPAGE***  
-----  
1. Approve Leaves  
2. Reject Leaves  
3. Search Leave Dates  
4. Logout  
Please enter your choice:  
|
```

Figure 67: Successfully logging in as an approver.

However, if the credentials do not match to the corresponding approver, the system will deduct an attempt. The user will then have 2 more attempts to login with the valid credentials. If they fail to do so after 3 attempts, the system will boot the user back to the main menu as a security feature.

```

*****WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM*****
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
4
Enter your Approver ID:
asodin
Enter username:
asdina
Enter username:
sdkub
Login failed. 2 attempts left.
Enter your Approver ID:
saoifn
Enter username:
sfaliaba
Enter username:
askub
Login failed. 1 attempts left.
Enter your Approver ID:
saodibn
Enter username:
fkuab
Enter username:
sub
Login failed. 0 attempts left.
No attempts left. Returning to main menu...

```

Figure 68: If there are no more attempts left, the user will be redirected to the main menu.

5.4.2 Approving Leaves

To approve leaves, the approver has to enter '1' in the approver homepage. There, the full list of leaves that were applied by all users including its details will be displayed for the approver to approve. To approve a leave, the approver has to enter the number which corresponds to the dedicated leave in the list. After inputting a valid number, the system will update the leave status of that leave, display a message saying that the leave has successfully been approved before redirecting the approver back to the approver homepage. In the scenario below, I will approve the second leave.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Rejected	30-4-2023	10 days
3.	102	Maternity Leave	Approved	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to approve? Enter any character to go back to the homepage.
 2 ←
 Leave has been approved!

Figure 69 Part 1: Approving the second leave.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Approved	30-4-2023	10 days
3.	102	Maternity Leave	Approved	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to approve? Enter any character to go back to the homepage.
 |

Figure 70 Part 2: The leave has been approved.

The leave can only be approved if the leave status is either pending or rejected. If it is rejected, the leave duration will be deducted from the staff's leave type's balance because since it was rejected in the first place, the duration of the leave would be refunded back to the staff's leave balance. However, since the leave has been approved now, the leave balance would be deducted again.

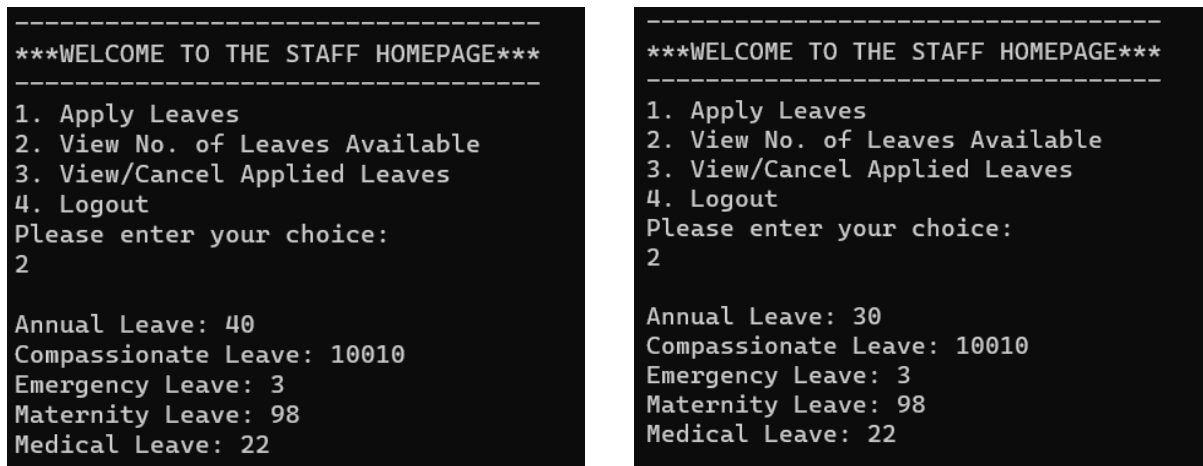


Figure 71: Before and after the leave that was initially rejected got approved. The annual leave balance was deducted as a result.

However, the leave cannot be approved if the leave status is either Approved or Cancelled, or if the date of the leave has already past. In the scenario below, the approver is not able to approve the first leave as the date has already passed since the current date as of writing this is the 17th of February 2023. By doing any of the previously mentioned scenarios, the approver will be redirected back to the approver homepage.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Rejected	30-4-2023	10 days
3.	102	Maternity Leave	Approved	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to approve? Enter any character to go back to the homepage.
1
Leave cannot be approved because it has already passed.

Figure 72: The approver is unable to approve the first leave as its date has already passed, despite its status being Pending.

Similarly, the approver can enter any character to go back to the homepage, and that includes other data types. Moreover, if the approver inputs an invalid number, the system will redirect the approver back to the approver homepage as well.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Approved	30-4-2023	10 days
3.	102	Maternity Leave	Approved	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to approve? Enter any character to go back to the homepage.
7
Invalid index.

Figure 73: By inputting an invalid number or data type, the system will redirect the approver back to the approver homepage.

5.4.3 Rejecting Leaves

To reject leaves, the approver has to enter '2' in the approver homepage. There, the full list of leaves that were applied by all users including its details will be displayed for the approver to reject. To reject a leave, the approver has to enter the number which corresponds to the dedicated leave in the list. After inputting a valid number, the system will update the leave status of that leave, display a message saying that the leave has successfully been rejected before redirecting the approver back to the approver homepage. In the scenario below, I will reject the third leave.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Approved	30-4-2023	10 days
3.	102	Maternity Leave	Approved	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to reject? Enter any character to go back to the homepage.
3 ←
Leave has been rejected!

Figure 74 Part 1: Rejecting the third leave of the list.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Approved	30-4-2023	10 days
3.	102	Maternity Leave	Rejected	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to reject? Enter any character to go back to the homepage.
|

Figure 75 Part 2: The third leave has been rejected.

The leave can only be approved if the leave status is either Pending or Approved. If it is approved, the leave duration will be refunded back to the staff's leave type's balance because since it was approved in the first place, the duration of the leave would be deducted from the staff's leave balance. However, since the leave has been rejected now, the leave balance would be refunded again, just like the cancel leave function.

```

***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
2

Annual Leave: 1000
Compassionate Leave: 670
Emergency Leave: 3
Maternity Leave: 98
Medical Leave: 22

```

```

***WELCOME TO THE STAFF HOMEPAGE***
-----
1. Apply Leaves
2. View No. of Leaves Available
3. View/Cancel Applied Leaves
4. Logout
Please enter your choice:
2

Annual Leave: 1000
Compassionate Leave: 670
Emergency Leave: 3
Maternity Leave: 108
Medical Leave: 22

```

Figure 76: Before and after rejecting the leave. The maternity leave has been refunded back to the staff's leave balance.

However, the leave cannot be rejected if the leave status is either Rejected or Cancelled, or if the date of the leave has already past. In the scenario below, the approver is not able to reject the first leave as the date has already passed since the current date as of writing this is the 17th of February 2023. By doing any of the previously mentioned scenarios, the approver will be redirected back to the approver homepage.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Approved	30-4-2023	10 days
3.	102	Maternity Leave	Rejected	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to reject? Enter any character to go back to the homepage.
1
Leave cannot be rejected because it has already passed.

Figure 77: The first leave of the list cannot be rejected as the date has already passed.

Similarly, the approver can enter any character to go back to the homepage, and that includes other data types. Moreover, if the approver inputs an invalid number, the system will redirect the approver back to the approver homepage as well.

NUMBER	ID	TYPE OF LEAVE	LEAVE STATUS	DATE	DURATION
1.	101	Annual Leave	Pending	15-2-2023	10 days
2.	101	Annual Leave	Approved	30-4-2023	10 days
3.	102	Maternity Leave	Rejected	20-2-2023	10 days
4.	101	Compassionate Leave	Canceled	5-6-2023	10 days
5.	101	Annual Leave	Canceled	4-5-2023	10 days
6.	101	Annual Leave	Canceled	5-9-2023	10 days

Which leave would you like to reject? Enter any character to go back to the homepage.
7
Invalid index.

Figure 78: Since the leave index is invalid, the approver will be redirected to the approver homepage.

5.4.4 Searching Leave Dates

The approver has the function to search specific dates to see the number of staff who applied for leave on that date. To access this function, the approver needs to input '3' in the approver homepage. There, the approver has to enter a date in DD, MM and YYYY format. If the date has a leave, then the system will return the number of staff who applied for leave on that date.

For example, if the approver inputs a date like the 30th of March 2023, which is a valid date and has a leave application (you may refer to Figure 59), the system will return the number of staff who applied for leave on that date.

```

*****
***WELCOME TO THE APPROVER HOMEPAGE***
*****
1. Approve Leaves
2. Reject Leaves
3. Search Leave Dates
4. Logout
Please enter your choice:
3

Please enter your starting day in DD format:
30
Please enter your starting month in MM format:
4
Please enter your starting year in YYYY format:
2023
Valid date!
1 number of staff applied on that day.

```

Figure 79: The system returns the number of staff who applied for leave on a valid date.

However, if no leaves were applied on a date, then the system will print a message saying that no leaves were applied on that day.

```

*****
***WELCOME TO THE APPROVER HOMEPAGE***
*****
1. Approve Leaves
2. Reject Leaves
3. Search Leave Dates
4. Logout
Please enter your choice:
3

Please enter your starting day in DD format:
1
Please enter your starting month in MM format:
2
Please enter your starting year in YYYY format:
2023
Valid date!
No leaves applied on that day!

```

Figure 80: The system says that there were no leaves applied on this date.

However, if the system detects one of the following scenarios:

1. If the leave year is lesser than 2023 or more than 2025.
2. If it is an invalid date or if the leave date does not exist, such as 31st April or 32nd December, or if the year is not a leap year and the date applied is 29th February.
3. If the data input for the leave date is incorrect, such as strings, floats, doubles or chars and negative numbers or zeroes.

Then the system will prompt the approver to re-enter a valid date.

```

-----
***WELCOME TO THE APPROVER HOMEPAGE***
-----
1. Approve Leaves
2. Reject Leaves
3. Search Leave Dates
4. Logout
Please enter your choice:
3

Please enter your starting day in DD format:
31
Please enter your starting month in MM format:
february
Please enter your starting year in YYYY format:
2023
Invalid date!

```

Figure 81: If there is an invalid date, the system will return a statement which says that it is an invalid date before prompting the approver to re-enter a valid date.

5.4.5 Logout

To logout as an approver, the approver needs to input '4' in the approver homepage. There, the approver will then be redirected back to the main menu.

```

-----
***WELCOME TO THE APPROVER HOMEPAGE***
-----
1. Approve Leaves
2. Reject Leaves
3. Search Leave Dates
4. Logout
Please enter your choice:
4

-----
***WELCOME TO THE APU LEAVE MANAGEMENT SYSTEM***
-----
1. Admin Login
2. Staff Login
3. Staff Registration
4. Approver Login
5. Feedback
6. Terms & Conditions
7. Exit
Please enter your choice:
|

```

Figure 82: Logging out as an approver.

6.0 Conclusion

In conclusion, this assignment was a great way for students to apply what they have learned in their C Programming lectures and tutorials into their assignment. In addition to honing their skills, C Programming has been frequently brought up in online forums and is credited to helping a lot of developers today learn about memory allocation and how to manage their resources better, providing a strong framework and foundation for their subsequent coding endeavours (Singh, 2021). Due to this, I have developed a greater understanding of the language and the programming process through the creation and debugging of code. The whole module was a joyride, with a lot of ups and downs when doing the assignment, but it was still fun, nonetheless.

7.0 References

Asia Pacific University. (n.d.). *Undergraduate Studies* . Retrieved from Asia Pacific University:
<https://www.apu.edu.my/our-courses/undergraduate-studies>

Blakely-Gray, R. (29 August, 2022). *Learn How to Calculate Your Absenteeism Rate—and Use It—Today*. Retrieved from Patriot Software:
<https://www.patriotsoftware.com/blog/payroll/how-to-calculate-absenteeism-rate/>

Programiz. (n.d.). *C switch Statement*. Retrieved from Programiz:
<https://www.programiz.com/c-programming/c-switch-case-statement>

Singh, A. P. (18 October, 2021). *5 Reasons Why You Should Learn C As Your First Language*. Retrieved from Level Up Coding: <https://levelup.gitconnected.com/5-reasons-why-a-you-should-learn-c-as-your-first-language-7db054620bc0>