

# UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2022/2023



Wholly owned by UTAR Education Foundation  
(Co. No. 578227-M)  
DU012(A)

## **UCCD 1004 PROGRAMMING CONCEPTS AND PRACTICES**

### **ASSIGNMENT 2 - Cinema Ticketing System**

Group XX			
Name	ID	Programme	UTAR Email
Joseph Lau Yi Zhe	20ACB03834	CS	josephlyzh213@1utar.my
Tharini A/P Vijesh Kumar	220ACB6802	CS	tharini2206802@1utar.my
Fan William	220ACB7661	CS	liamfan@1utar.my

## Task Division

	Name	Modules	Description	*A2 Contribution (Overall, %)
1.	Joseph Lau Yi Zhe	View movies module	Read in movies.txt (in movies.txt, same movies are not necessarily arranged together), arrange movies with same name together, sort ascendingly according to time and display the movie list.	1
		User menu module	Allows user to book tickets, edit booked tickets and perform payment.	2
		Admin update module	Allows admin to change movie slot (showtime/ hall), add movie slot and delete movie slot. The changes will be saved to movies.txt.	3
2.	Tharini A/P Vijesh Kumar	Register user module	Register a user account and store in users.txt. The next compilation of program still able to detect existing users.	1
		Login user module	Verify user login with correct username and password to proceed to user menu interface.	2
		Login admin module	Verify admin login with correct username and password to proceed to admin menu.	3
		Decorations module	Involves the decorations printings and loading bar.	4
3.	Fan William	Check ticket records module	For admin to check ticket records and compute the total income from sold ticket.	1
		Check out module	Update the ticket records.txt every time a payment is made.	2

\* Depends on the evaluation of the markers as well;

The original modules were distributed evenly. The modules are not balanced here because Student 3 just only started his individual modules (User menu module, payment module and checkout module) on 22/4/2023. Due to running out of time, Student 1 finished the rest during Hari Raya which cause imbalanced modules here.

## Report

- Student 1- Test Cases
- Student 2, 3 - Flowchart & Pseudocode

## **Objectives:**

- This is a program of a cinema ticketing system, which could be divided into admin and users.
- The input files are movies.txt, users.txt and ticket records.txt. The changes in the system by user or admin will be updated to the input files.
- The program will require user to register an account at main menu before login to user menu to book movie tickets. At user menu, user can book tickets, cancel their booked tickets before payment and perform payment.
- Admin can perform update on movies slots displayed (change only showtime or hall, add movie slot and delete movie slot), and check ticket records which will calculate the total income from ticket earned.

## **Pseudocode**

```
function main
call: displaymenu function
ask user to assign choice an integer

switch (choice)
case: choice is equal to 1
    display registerUserfunction()
    break

case: choice is equal to 2
if ( is logged in )
    isLoggedIn = loginUser (login_username)
    show showloadingBar (duration,delay)
if LoggedIn
    call: bookMovies function
if ( is not logged in )
    show "You are already logged in"
    show "Redirecting you to the user menu..."
    show showLoadingBar (duration,delay)
    break

case: choice is equal to 3
```

```
if not isLoggedIn
    show "You're not logged in"
    show "Press enter to go back to the main menu"
    call: displaymenu function
    show "Press Y to logout"
    show "Press any other key to go back to main screen"

if logout is assigned to 'Y' or 'y'
    isLoggedIn is equal to false
    show "You have successfully logged out of the system"
    call: displaymenu function

otherwise,
    show "You are still logged in. Press enter to main menu"
    break

case: choice is equal to 4
While attempts less than max_attempts and not login_successful
    Run adminLogin function
while admin_choice more than or equal to 1 && admin_choice less than or equal to 2
if admin_choice is assigned with 1
    run update (movies, moviesOS)
    open moviex.txt
    initialize i equal to Starting_index (movies)
for i less than Size
    post increment i
    once i more than Size, exit loop
    write moviex[i].name, moviex[i].time, moviex[i].location
    close movie.txt
    show "The new movie table"
    show "changes are successfully updated: "
if admin_choice is assigned with 2
```

```
run check_record function
show "press any key to go back to admin menu interface
call: displayadminmenu
show "Enter your choice"
get admin_choice

otherwise,
show "Incorrect login credentials. attempts reamining: "
post increment attempts
max attempts - attempts

if attempts less than 3
show "Press any key to try again... "
else
show you have exceeded the maximum number of login attempts"
call: showLoadingBar (duration, delay)
break

case: choice is equal to 5
show "Goodbye!"
break
return 0

default
show "Invalid choice"
break

While choice not equal to 5
return 0
//end function main

function loadUsers
read "users.txt"
```

```
if "users.txt" opened
    declare username, password as string
    while read username && password from "users.txt"
        assign users[numbers].username with username
        assign users[numbers].password with password
        close "users.txt"
    otherwise,
        show "Error unable to open file"
//end function loadUsers

function displayadminmenu
display admin menu
show "1 Perform Update"
show "2 Check ticket records"
show "Press any other number to exit"
//end function displayadminmenu

function registerUser
display register user banner
if numUser more than or equal to MAX_USERS
    show "Error: maximum number of users reached
otherwise,
    show "Please enter a username"
    get username
    show "Please enter a password"
    get password
    users[numUsers].username = username
    users[numUsers].password = password
    post increment numUsers
    open "users.txt"
    write username and password in "users.txt"
```

```
//end function registerUser

function check_tickets_index
for i less than Size
    post increment i
        if i location for ticket[].name, ticket[].time, and ticket[].location are not assigned
            return i
        exit loop by returning 50
//end function check_tickets_index

function check_exist
for i less than Size
    post increment i
if name is equavalent to i'th of moviesOS[]
return true
    exit loop by returning false
//end function check_exist

function null_index
for i less than Size
    post increment i
if moviesOD[i] not assigned
    post increment pos
otherwise,
    break
exit loop by returning pos
//end function null_index

function set_movieOS
for i less than Size, post increment i
    name = moviesOS[i].name
```

```

if calling check_exist function return false

    index is equal to null_index

        index'th of moviesOS will be assigned to name

//end function set_moviesOS


function check_index

for i less than Size,post increment i

    if ith position of movies[].name, movies[].time, and movies[].location are not assigned

        return i value

    otherwise,

        return 50

//end function check_index


function starting_index

for i less than Size, post increment i

    if ith position of movies[].name, movies[].time, movies[].location are not assigned

        otherwise,

        return 50

//end function starting_index


function movie_details

call set_movieOS

display admin banner

for i less than null_index(moviesOS), post increment i

    for j less than Size, post increment j

        if jth position of movies[].name is equal to ith position of moviesOS[]

            show movies[j].name, show movies[j].time

//end function movie_index


function update

show admin menu banner

```

```
show "choice 1 - 5"
get choice
if choice is more than 4 or less than 1
    show "Invalid input. Please try again..."

while choice is less than 4 or more than 1
    if choice is not equal to 4
        call movie_details function
        while not valid_name, choice not equal to 2, and shoice not equal to 4
            show "Enter the name of movie to perform update/change: "
            get name
            show name

        for i less than Size post increment i
            if ith movies[].name is equal to name
                valid_name is equal to true
                post increment no
                search_index[no] is equal to i
                show no, ith movies[].time, and ith movies[].location

        if valid_name is false
            show "Movie not found in database. Please be careful with the spellings and capital letters

if choice is equavalent to 1
show "Press 'A' to change showtime. Press 'B' for changing hall"
get choice2

if choice2 is equavalent to 'A'
    show "Which slot's showtime do you want to change (Enter number): "
    get no
    index is equal to no'th of search_index[]
```

```
show "changing from:", index'th of movies[].time, "to:"  
get changed_time  
index'th of movies[].time is assigned to changed_time  
call sort function
```

otherwise if choice2 is equivalent to 'B'

```
show "changing from:", index'th of movies[].time, "to: "  
get changed_location  
index'th of movies[].location is assigned to changed_location
```

otherwise

```
show "Invalid choice! Exiting back to Admin Interface..."
```

otherwise if choice is equivalent to 2

```
show "Movie Name: "  
get add_name  
show "Showtime: "  
get add_time  
show "Hall: "  
get add_hall
```

index is assigned with check\_index(movies)

if index is equal to Size

```
show "Error: the movie storage is full. Movie not added. Exiting back to Admin Interface..."
```

else

```
movies[index].name is assigned to add_name  
movies[index].time is assigned to add_time  
movies[index].location is assigned to add_hall  
sort (movies)
```

otherwise if choice is equivalent to 3

```
show "Which slot to delete (Enter number): "
get no
index ia assigned to no'th of search_index
for i less than Size, post increment i
    i'th of movies is assgined with (i+1)'th of movies
    (Size-1)'th of movies is assigned to 0th of movies

while choice is not equal to 4

//end function update

function bookMovies
    display user menu banner
    get option

if option is equavalent to 1
    valid_name is assigned to false
    call movie_details function

    while valid_name is false
        get name
        show name

for i less than Size, post increment i
    if ith of movies[].name is equavalent to name
        valid_name is assigned with true
        post increment no
        no'th of search_index is asigned to i
        show no, i'th position of movies[].time, i'th position of movies[].location
```

```
if valid_name is false
    show "Movie is not found in database. Please be careful with spellings and capital letters"

show "Enter the slot number to book your ticket: "
get no
index is assign to no'th of search_index
ticket_index'th of tickets[].name is assigned to index'th of movies[].name
ticket_index'th of tickets[].time is assigned to index'th of movies[].time
ticket_index'th of tickets[].location is assigned to index'th of movies[].location
post increment ticket_index

otherwise if option is equavalent to 2
index is assigned to check_ticket_index(tickets)

if index is equavalent to 0
    show "You didn't book any ticket yet so you dont have to pay."
otherwise
    show "This is the list of your booked tickets."

for i less than index, post increment i
show i'th position of tickets[].name, tickets[].time, tickets[].location
show "Enter no. of ticket that you want to delete. enter 0 if you dont want to delete"
show "No: "
get num

if num is not equavalent to 0
    for i less than index, post increment i
    (index - 1)'th position of ticket[].name is assigned to nothing
    (index - 1)'th position of ticket[].time is assigned to nothing
    (index - 1)'th position of ticket[].location is assigned to nothing
```

ticket\_index is assigned to index - 1

otherwise if option is equavalent to 3

index is assigned to check\_index(tickets)

if index is equavalent to 0

show ""You didn'd book any ticket yet."

otherwise

show "This is the list of your booked tickets."

for i less than index, post increment i

show i'th position of tickets[].name, tickets[].time, tickets[].location

show "Total payment is RM", index \* 8

show "Enter your credit card number: "

get dummy

show "Payment completed."

open "ticket records.txt"

for i less than index, post increment i

write on i'th position of tickets[].name, tickets[].time, tickets[].location

close "ticket record.txt"

option is assigned to 4

otherwise

show "Your booking is not saved"

show "Press enter to continue...."

while option is not equal to 4

```
//end function bookMovies

function check_record
open inRecord("ticket records.txt")
if inRecord is open
    while inRecord not end of file
        get username, write in "ticket records.txt"
        get ticket write in "ticket records.txt"
        inRecord write profit
        get dummy
        Show username, ticket, profit
        total_profit add with profit is now assigned to profit
        close inRecord

otherwise
Show "Error: Unable to open ticket records.txt!"

//end function check_record

function sort
for i less than Size, post increment i
for j less than Size-1, post increment j

if j'th position of movies[].time is less than movies[j+1].time
temp is assigned to movies [j]
j'th position of movies[] is assigned to movies [j+1]
movies [j+1] is assigned to temp

//end function sort

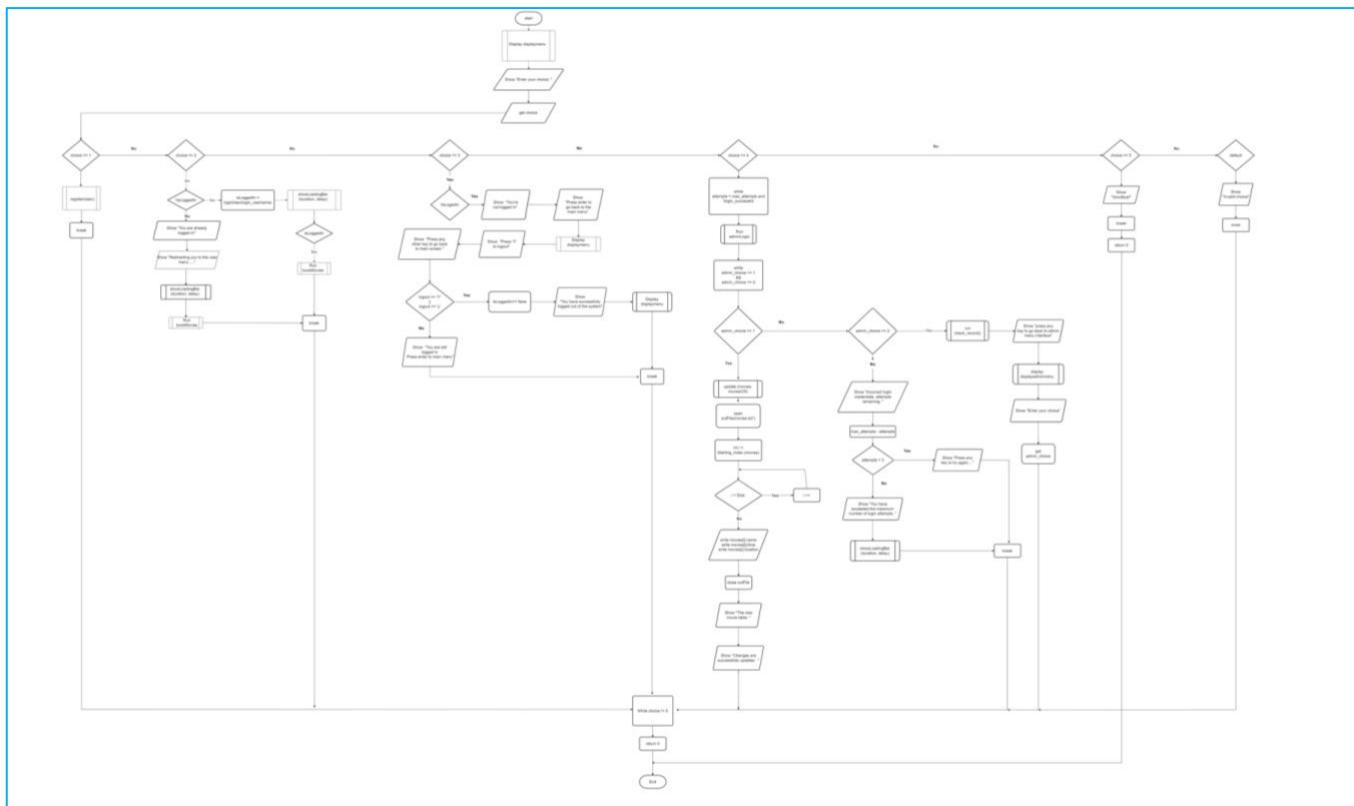
function readMovies
```

```
open inMovies("movies.txt")

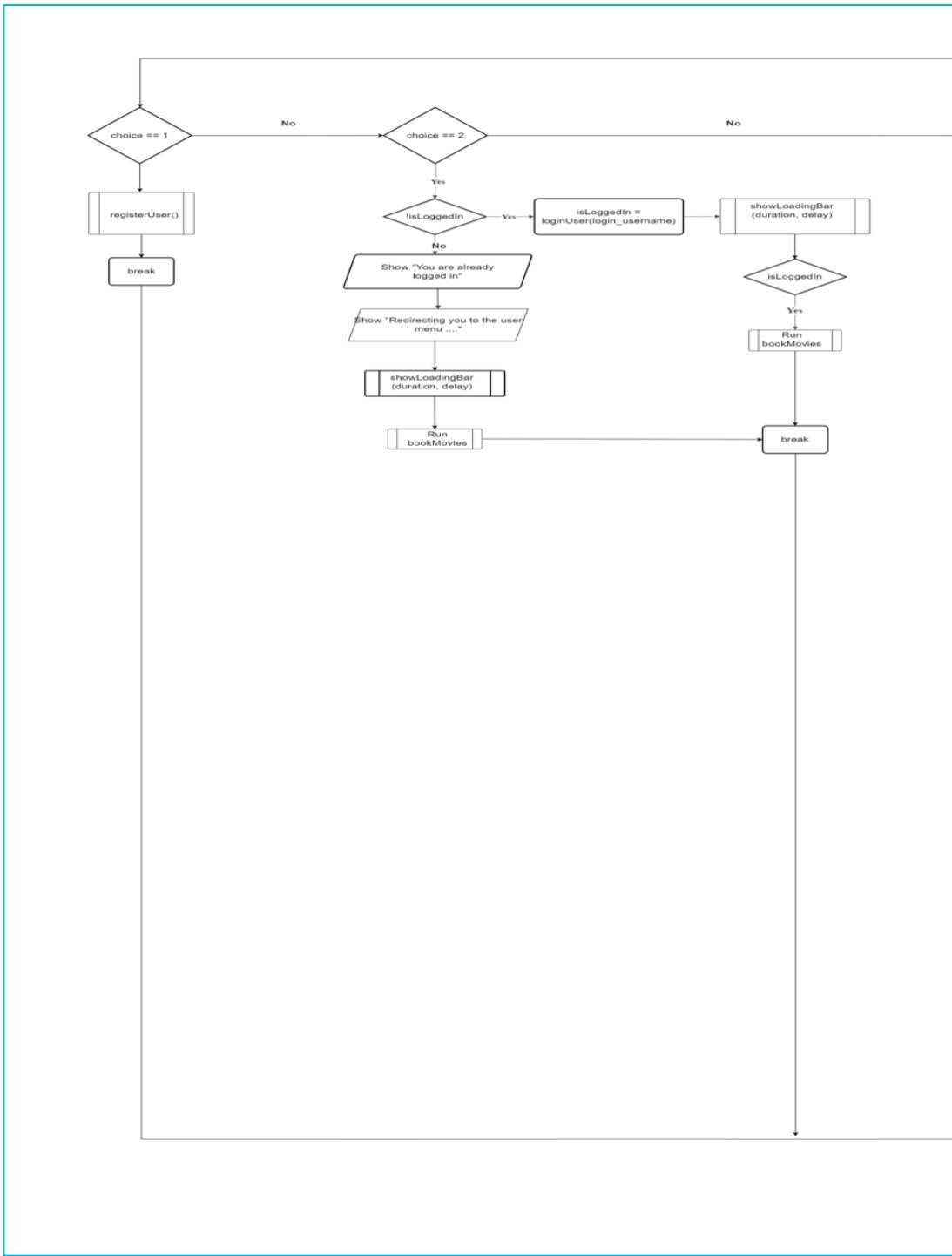
if inMovies is open
    while inMovies not end of file
        read i'th position of movies[].name
        show i'th position of movies[].name
        get i'th position of movies[].location
        post increment i
    close inMovies
otherwise
    show "Unable to open movies.txt"
```

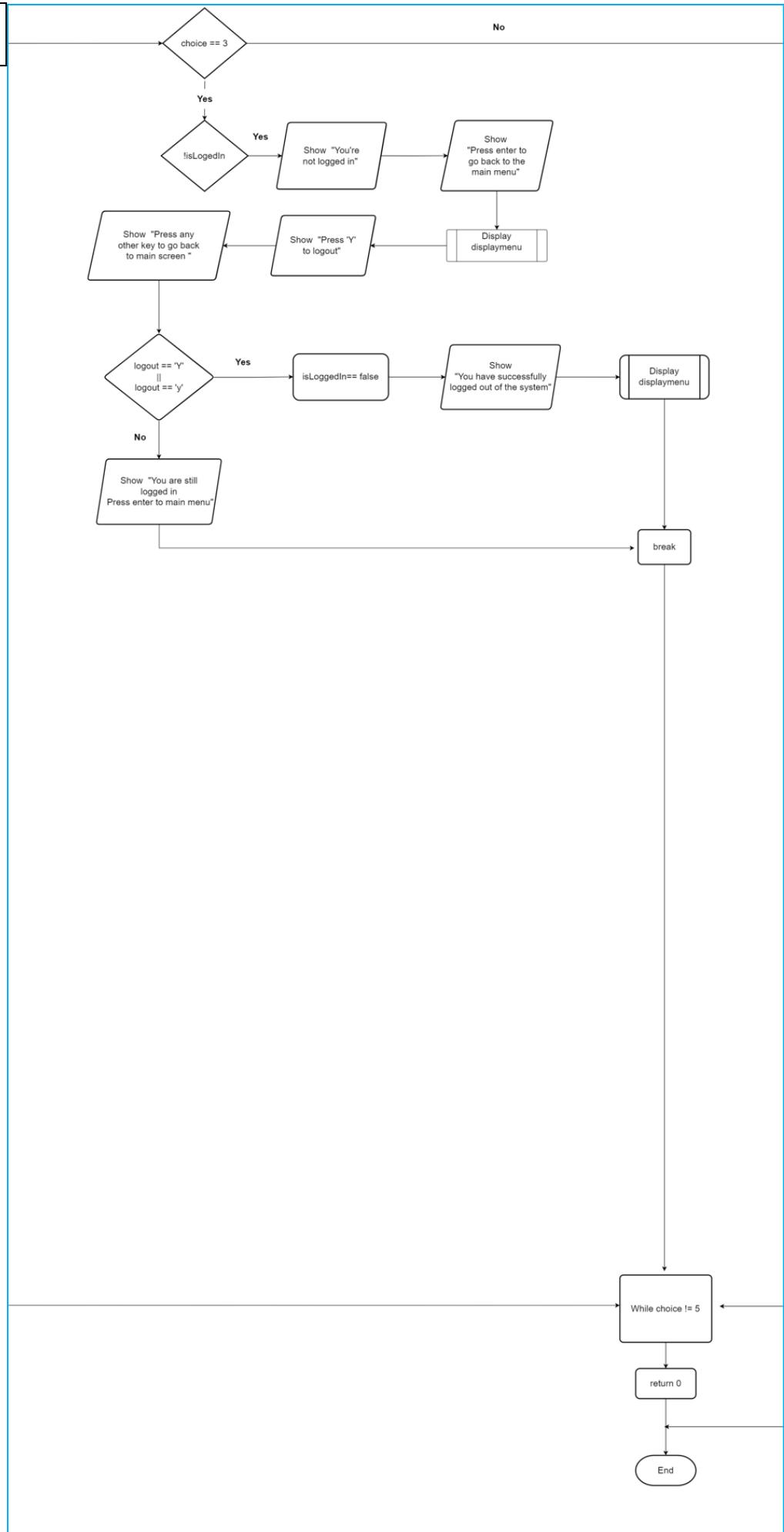
## Flowchart

Flowchart: int main

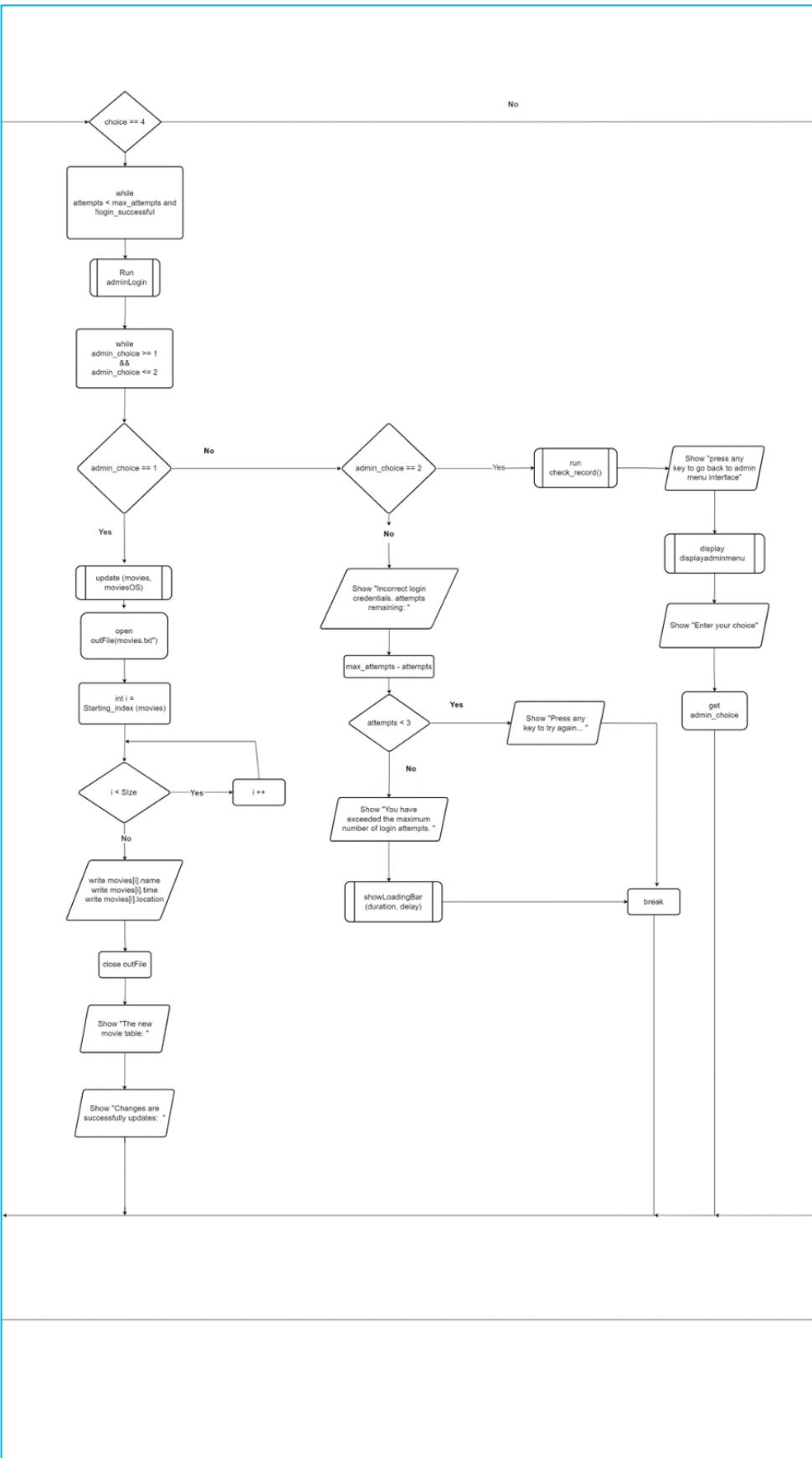


### Flowchart: case 1, case 2

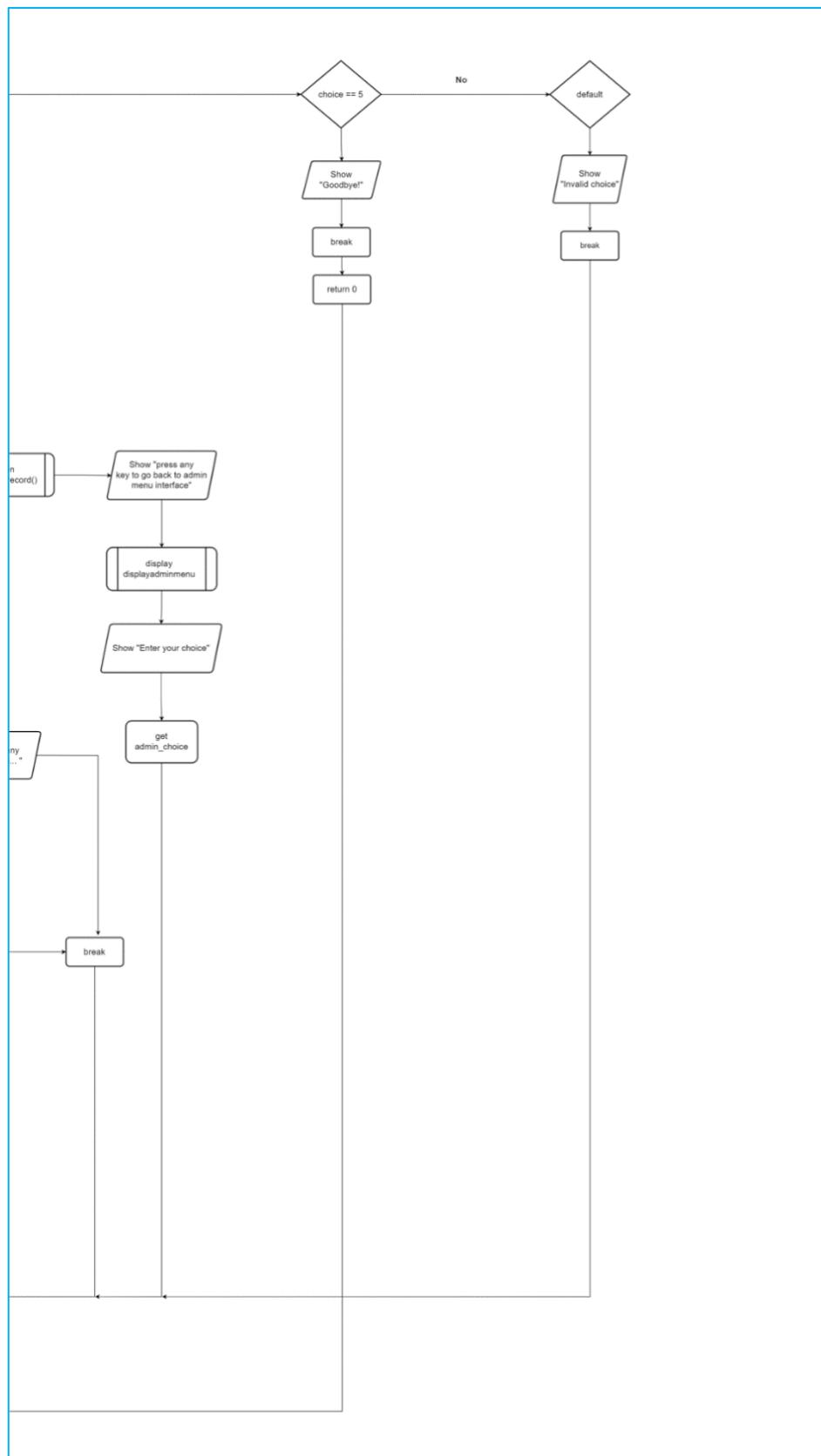


**Flowchart: case 3**

## Flowchart: case 4

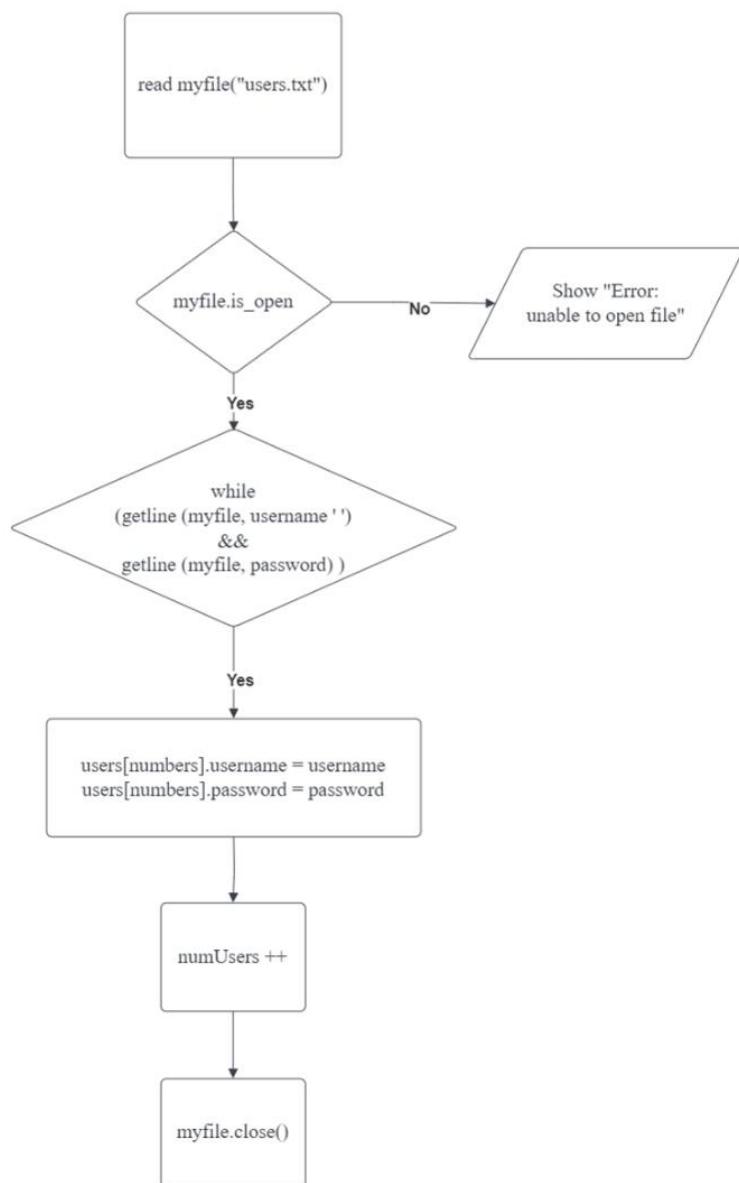


**Flowchart: case 5, default**

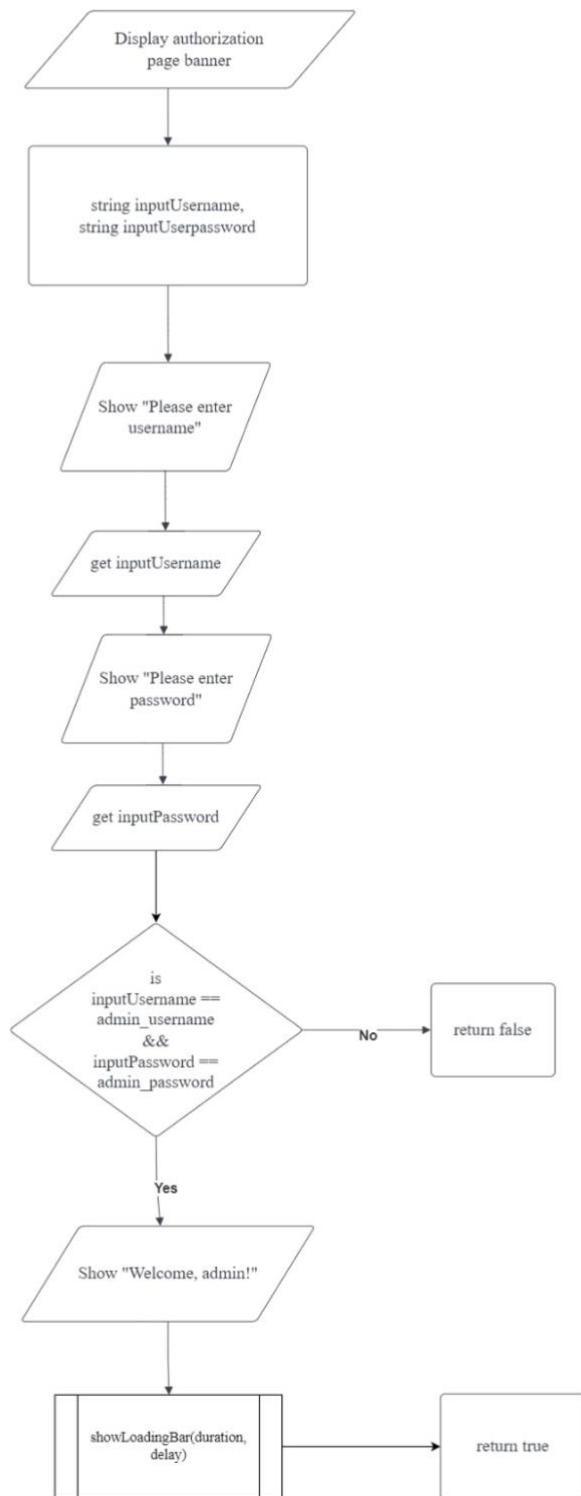


**Flowchart: functions**

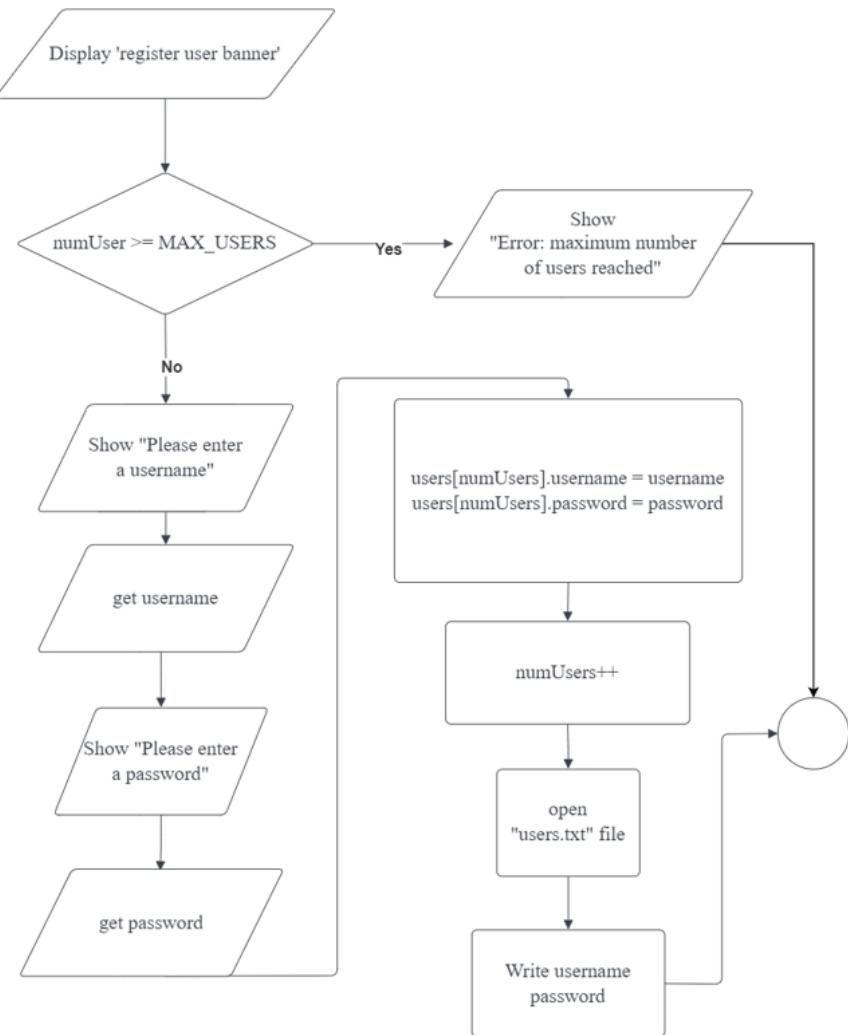
## void loadUsers



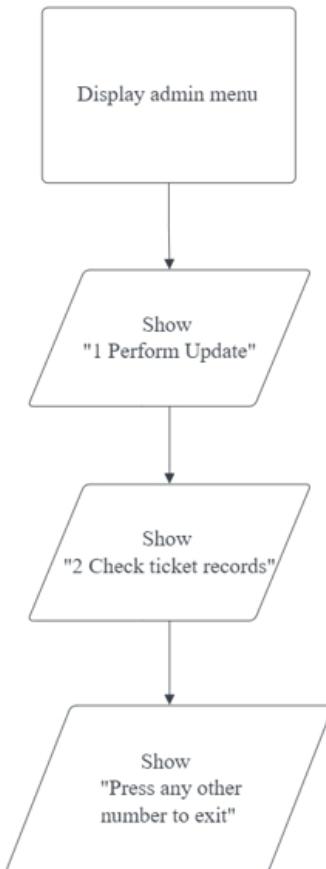
## bool adminLogin



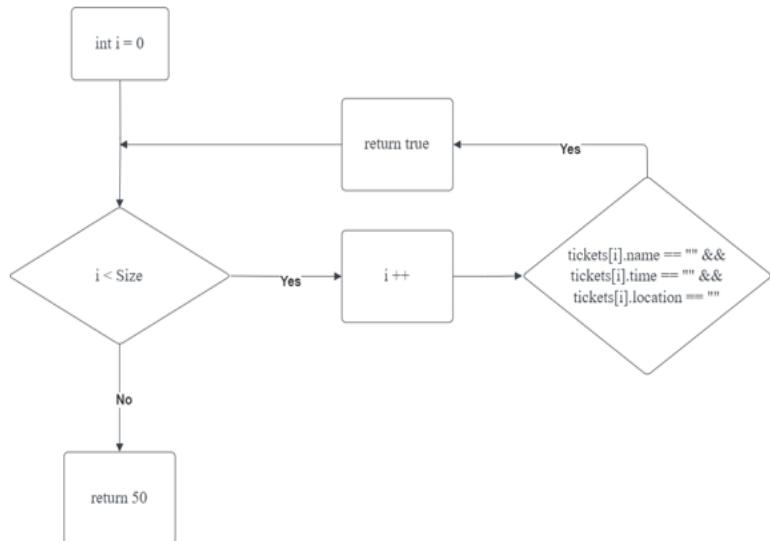
## void registerUser



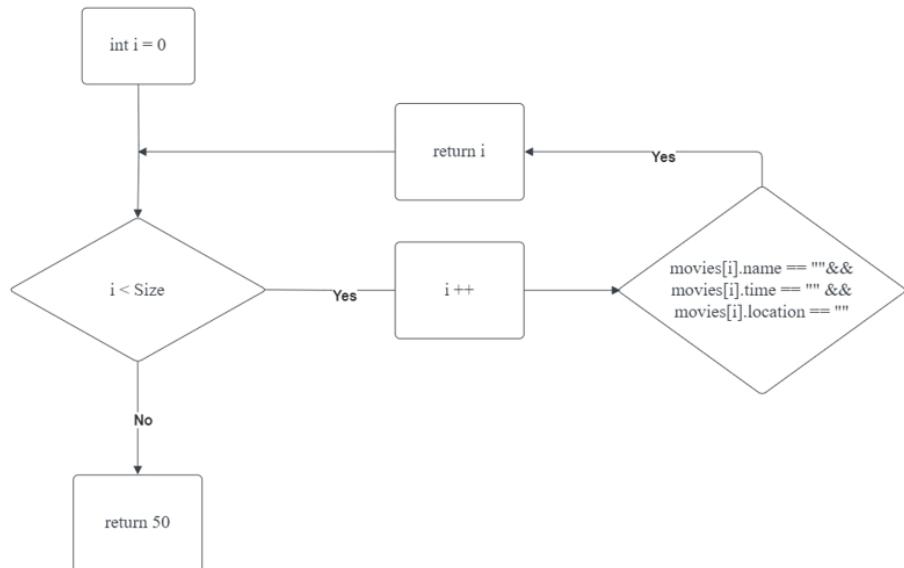
## void displayadminmenu



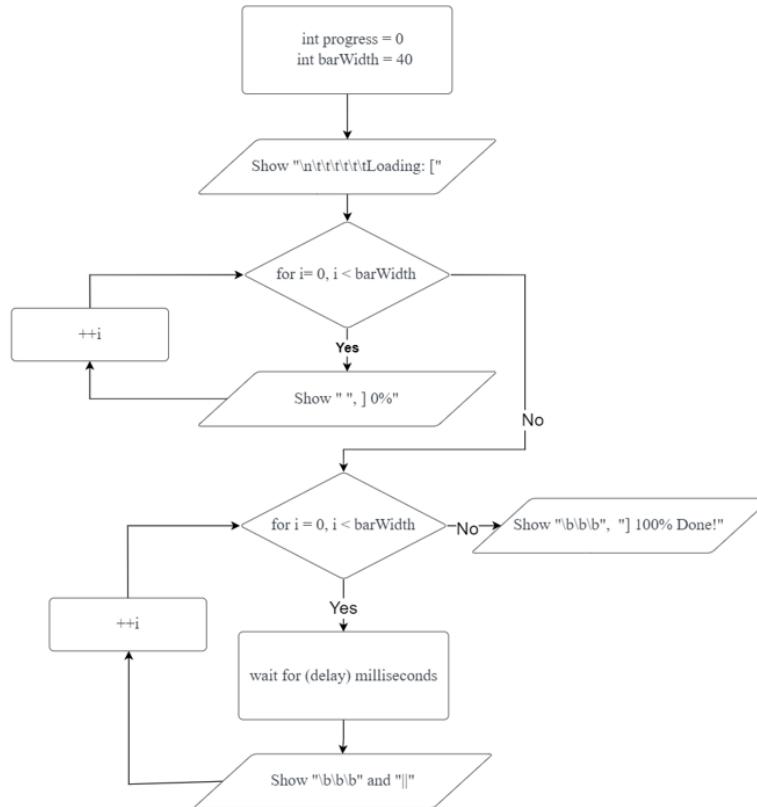
## Flowchart:check tickets index



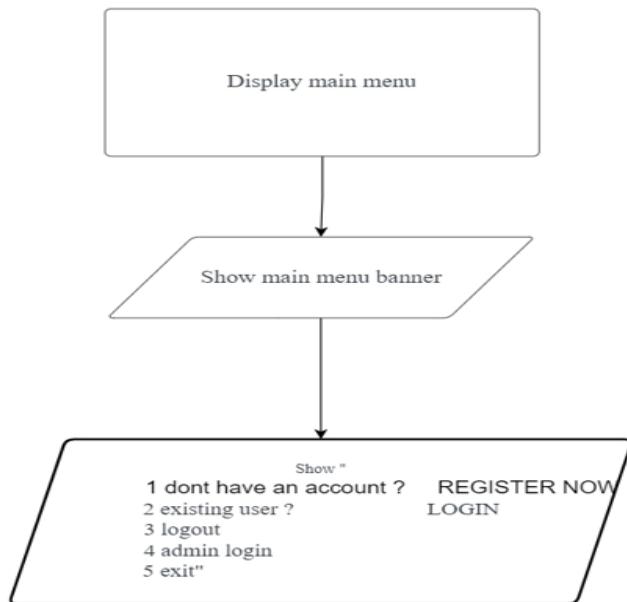
## int check index



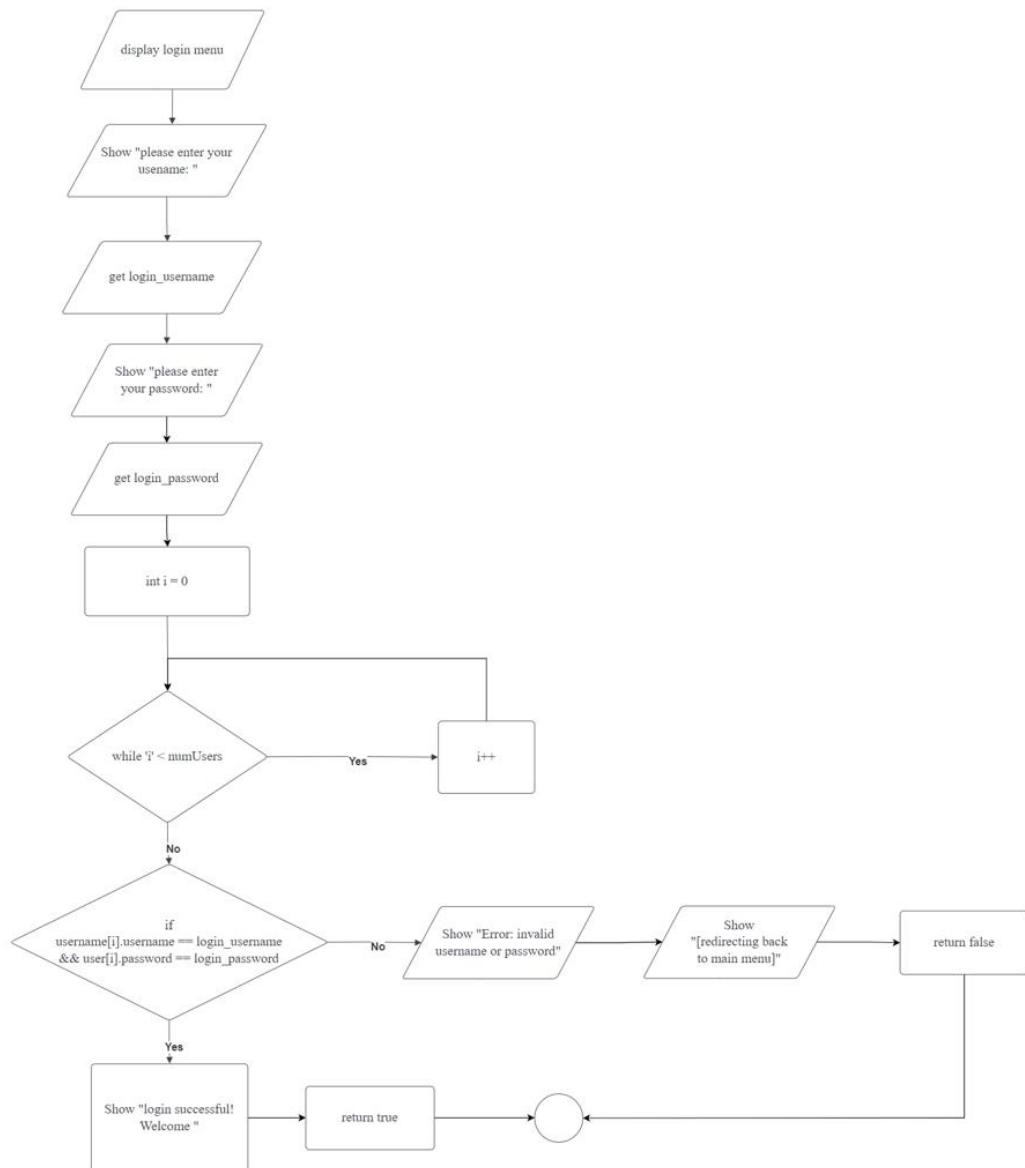
## void loading bar



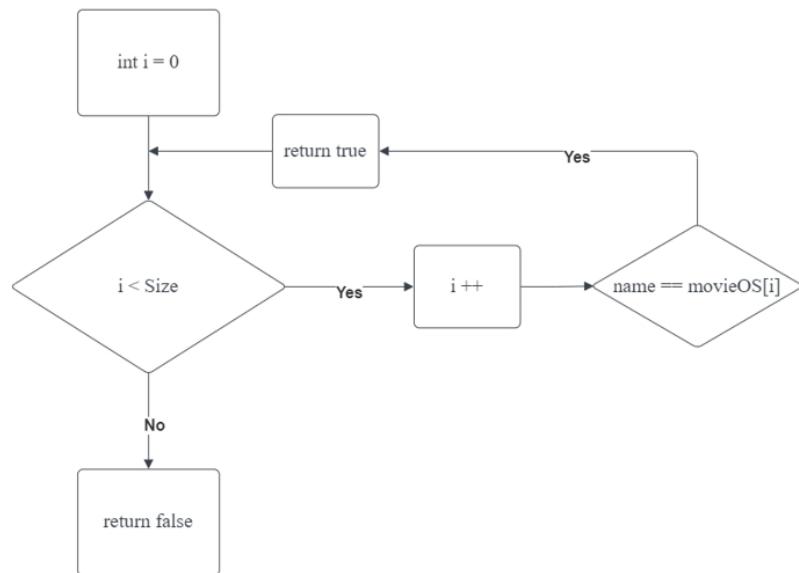
## void displaymenu



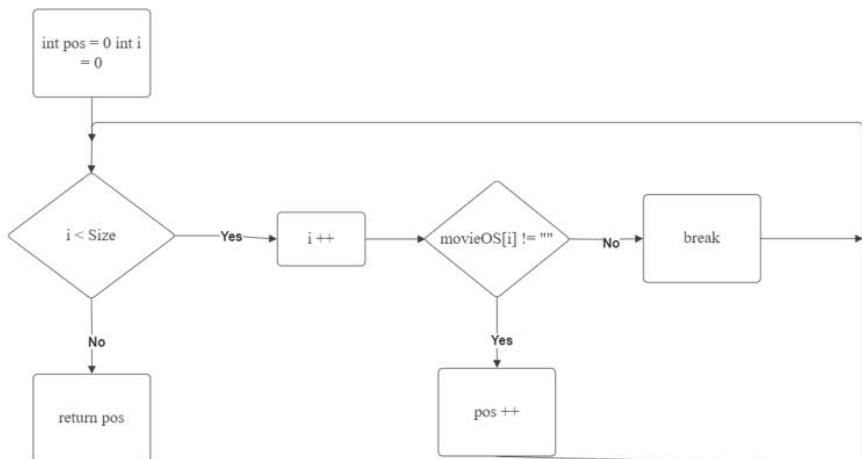
## bool loginUser



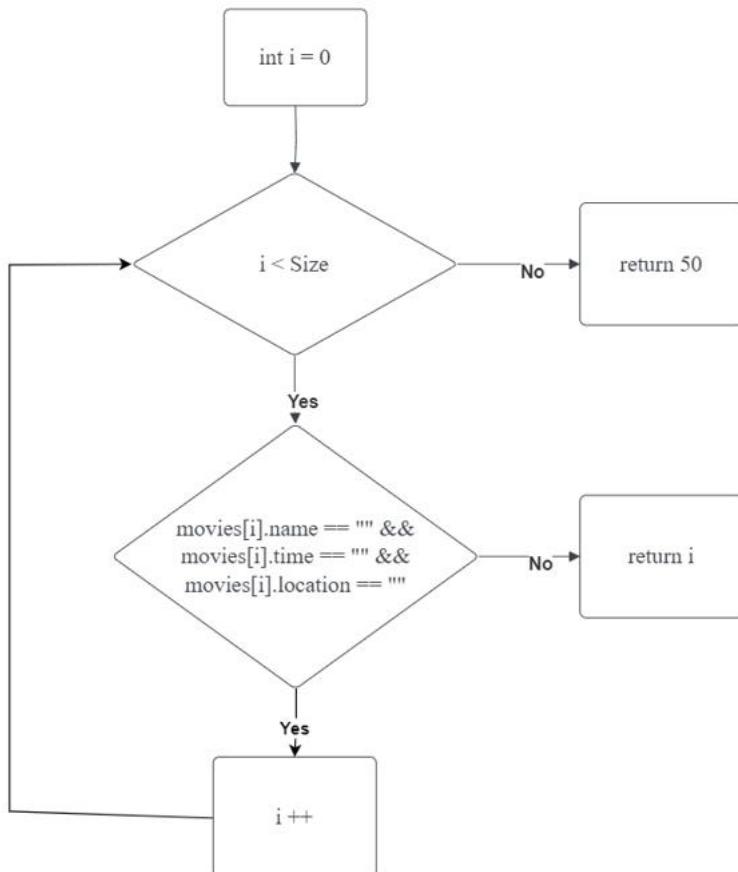
## bool check\_exit



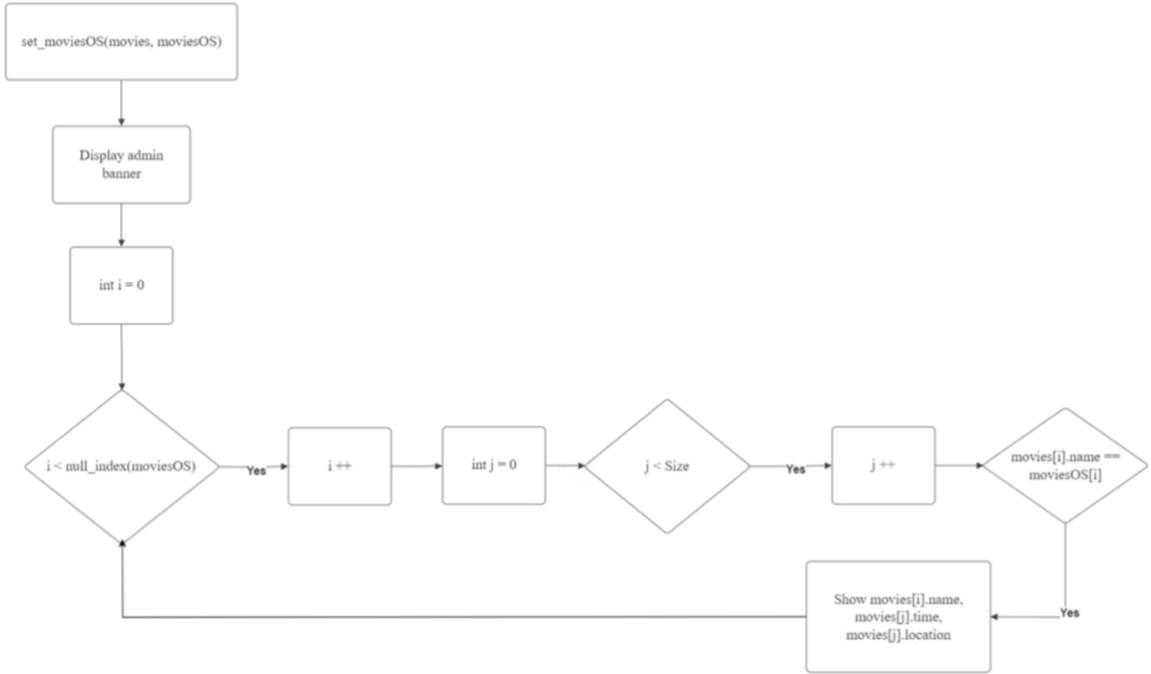
## int null\_index



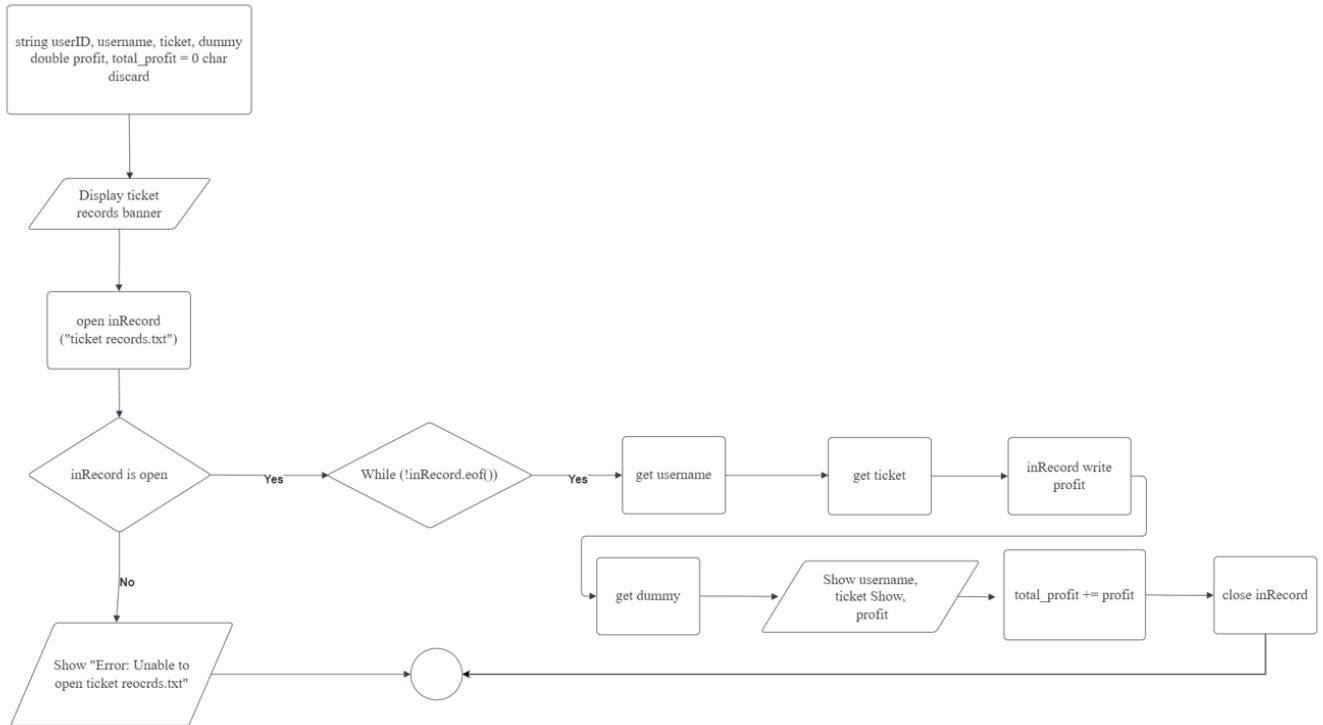
## int starting\_tickets\_index



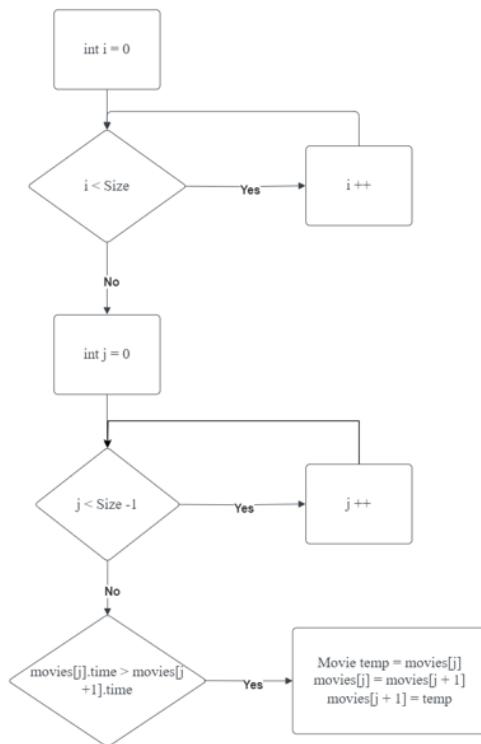
## void movie\_details



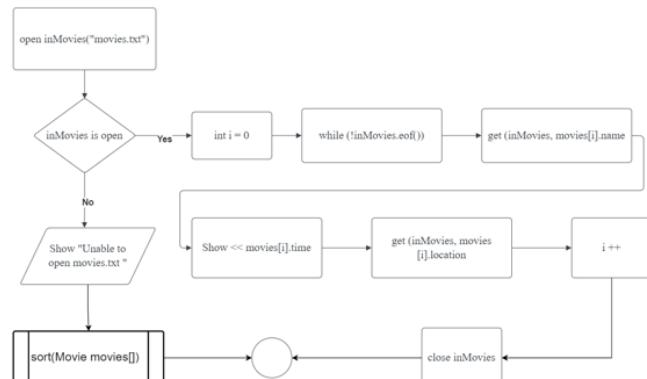
## void check\_record



## void sort(Movie movies[])

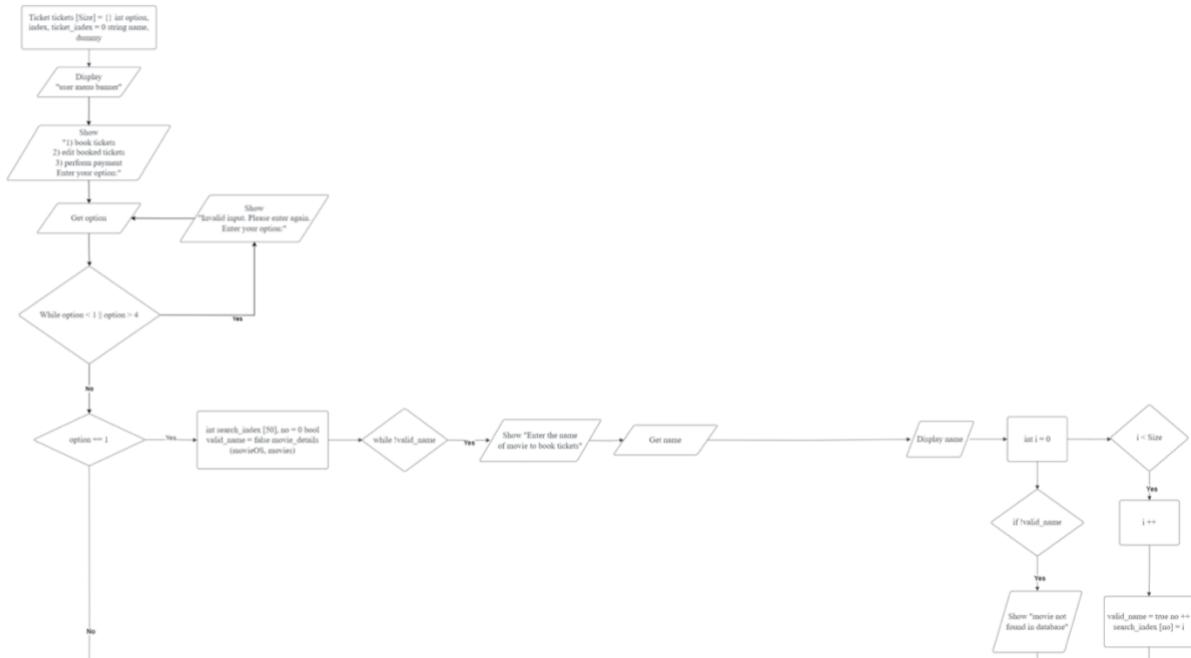


## void readMovies(string moviesOS[], Movie movies [])

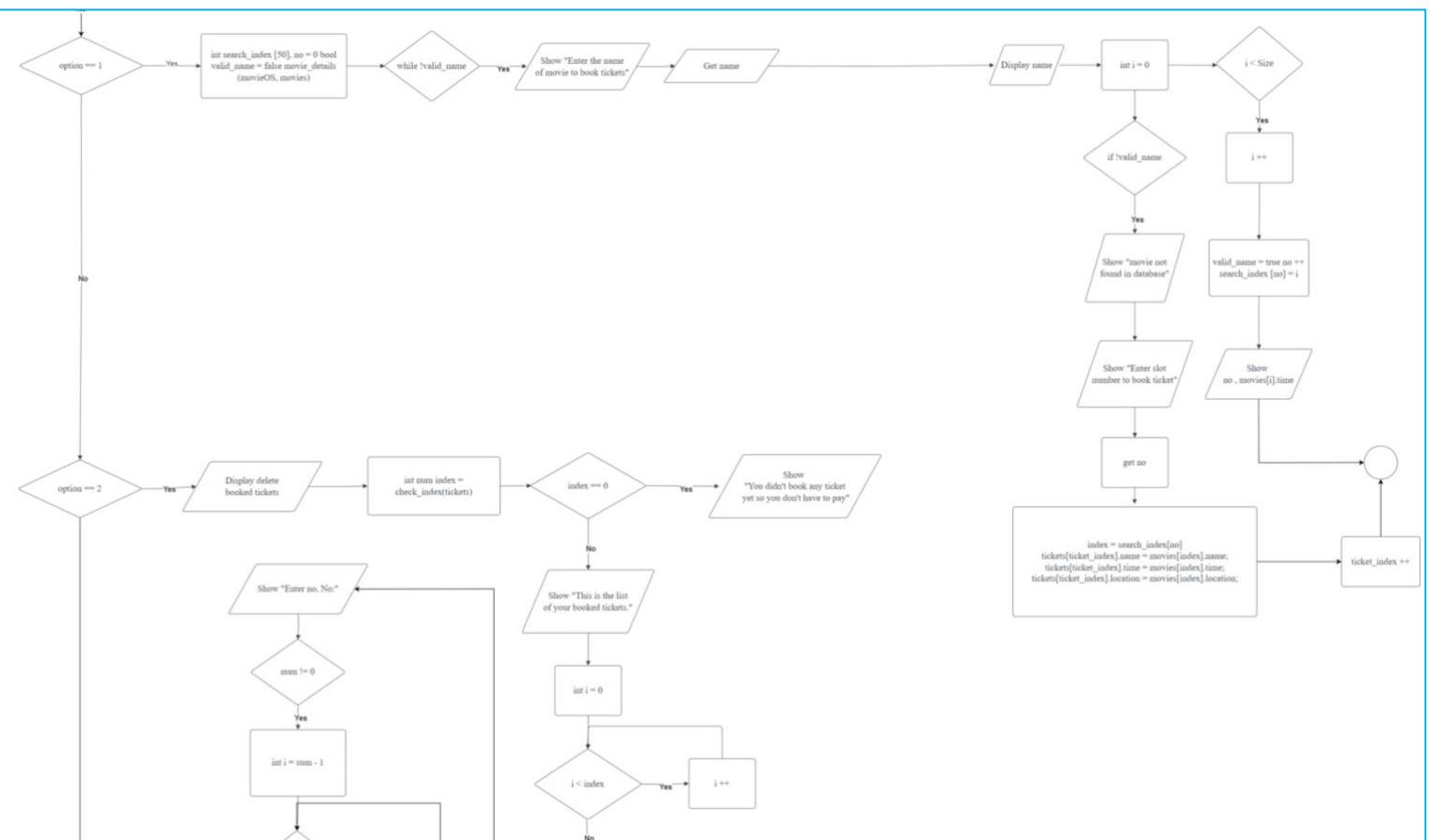


## Void bookMovies (part 1)

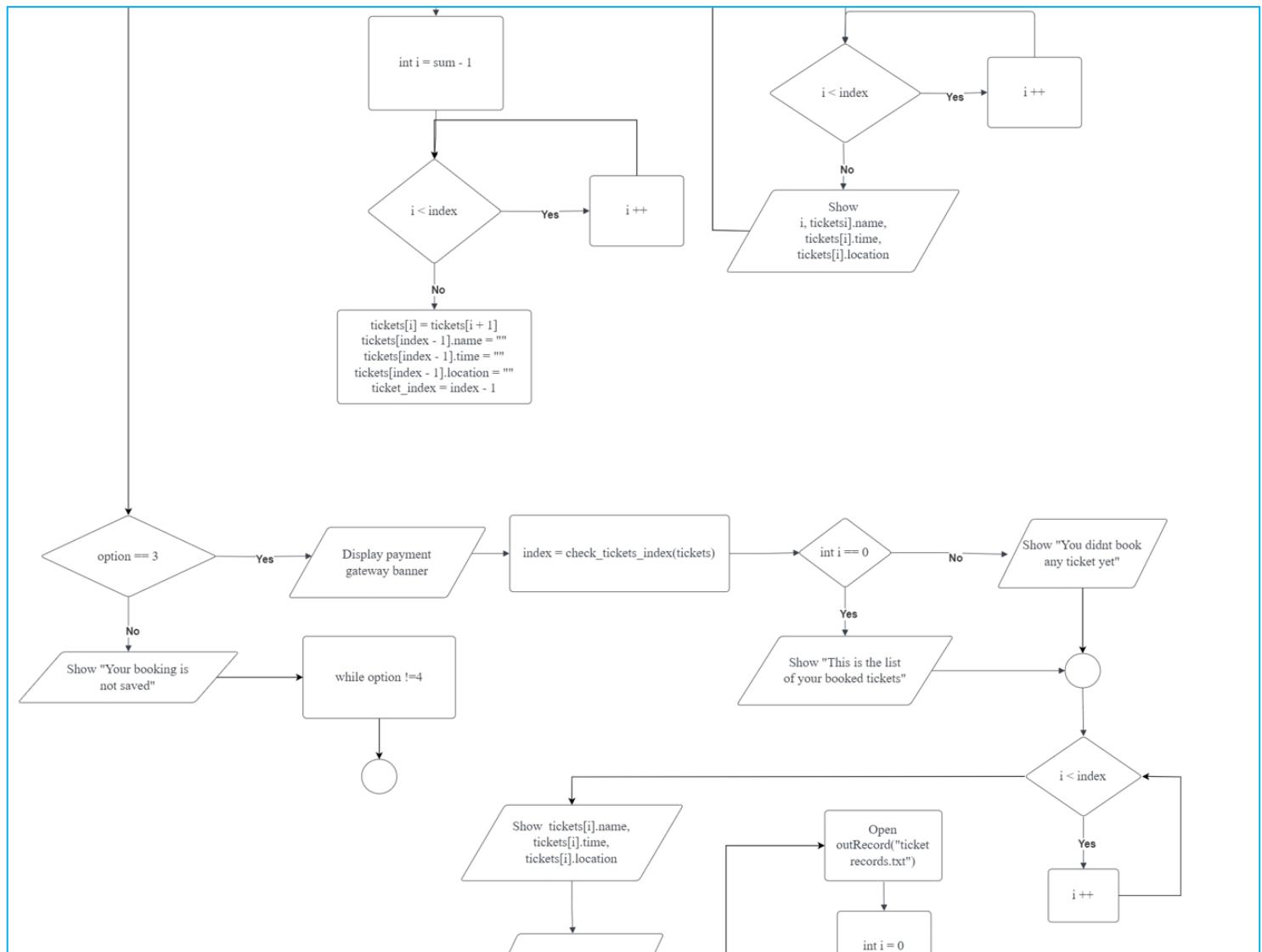
### Flowchart: void bookMovies



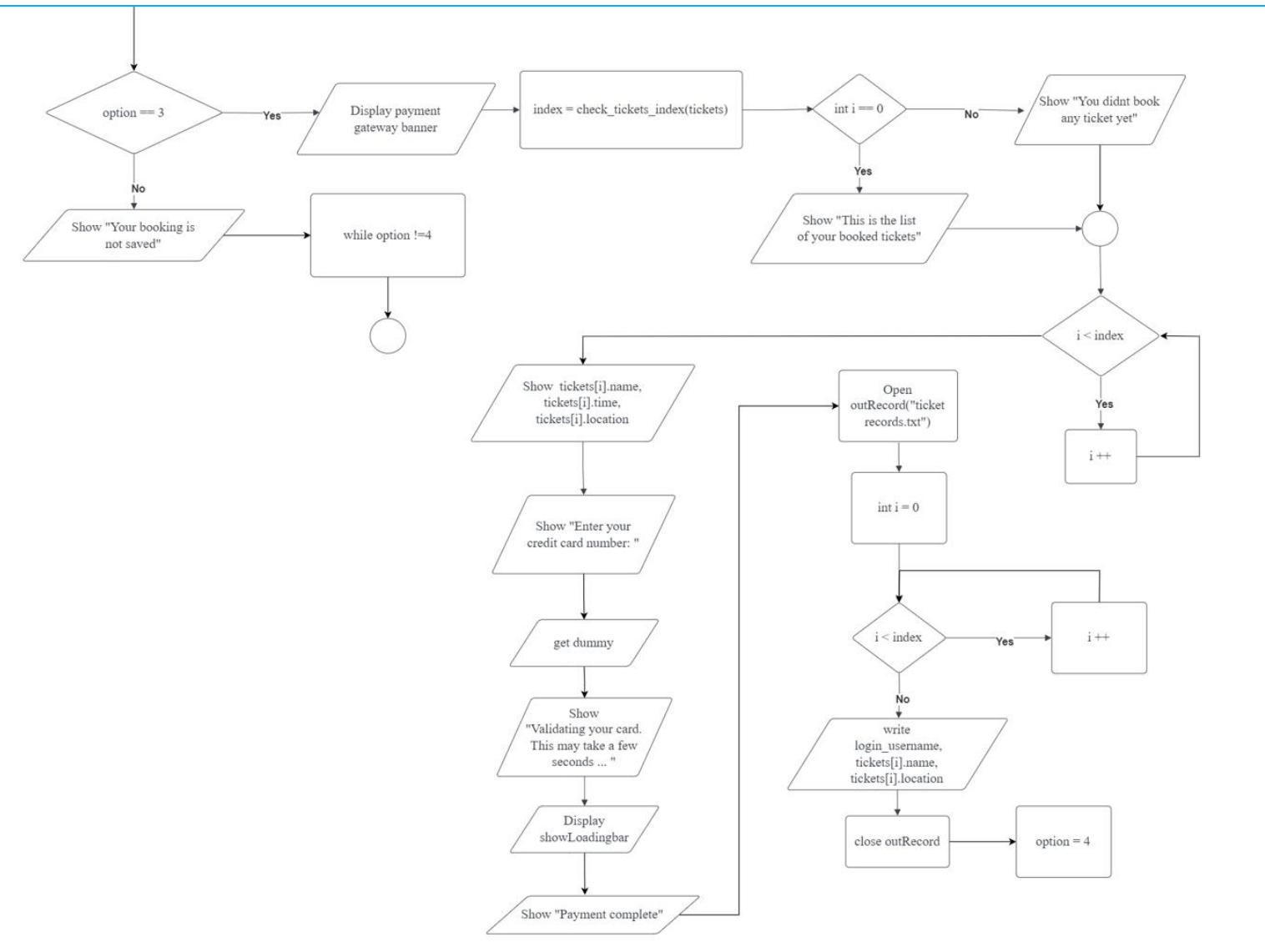
## Void bookMovies (part 2)



### Void bookMovies (part 3)

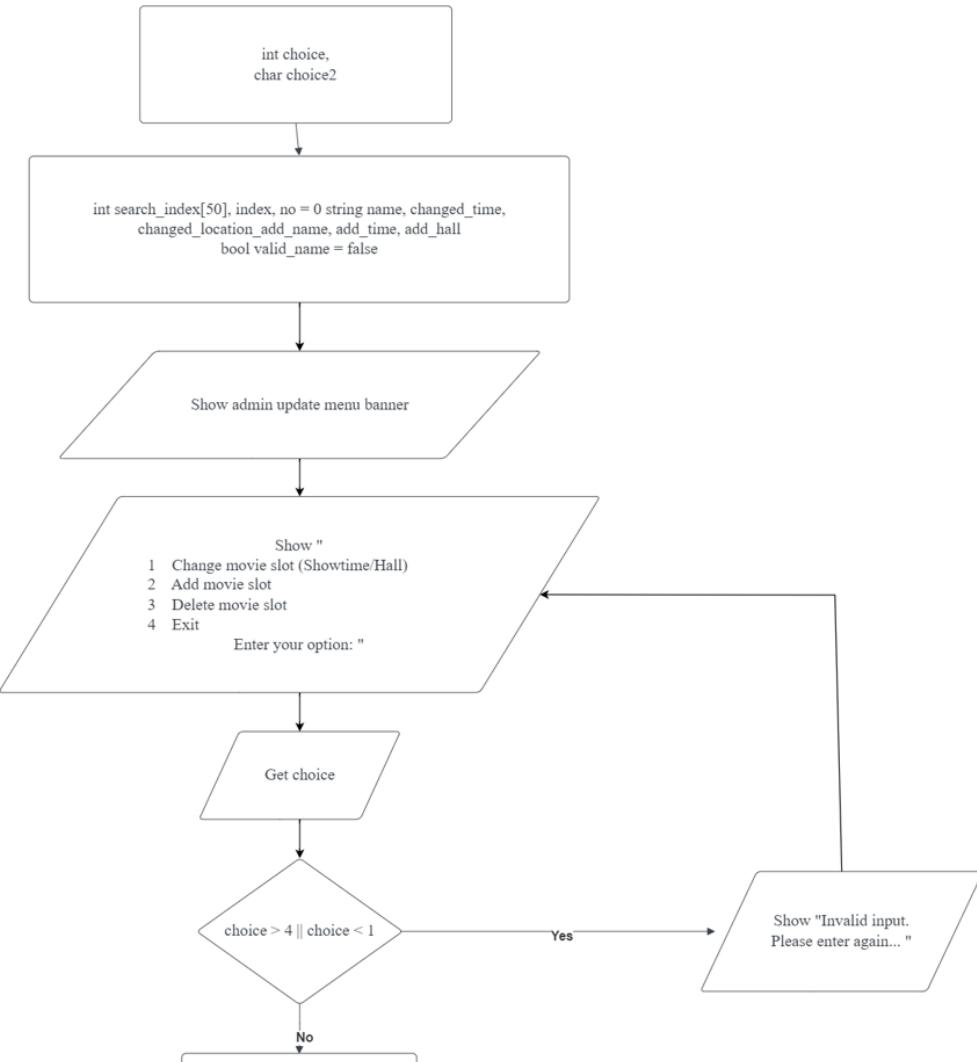


## Void bookMovies (part 4)

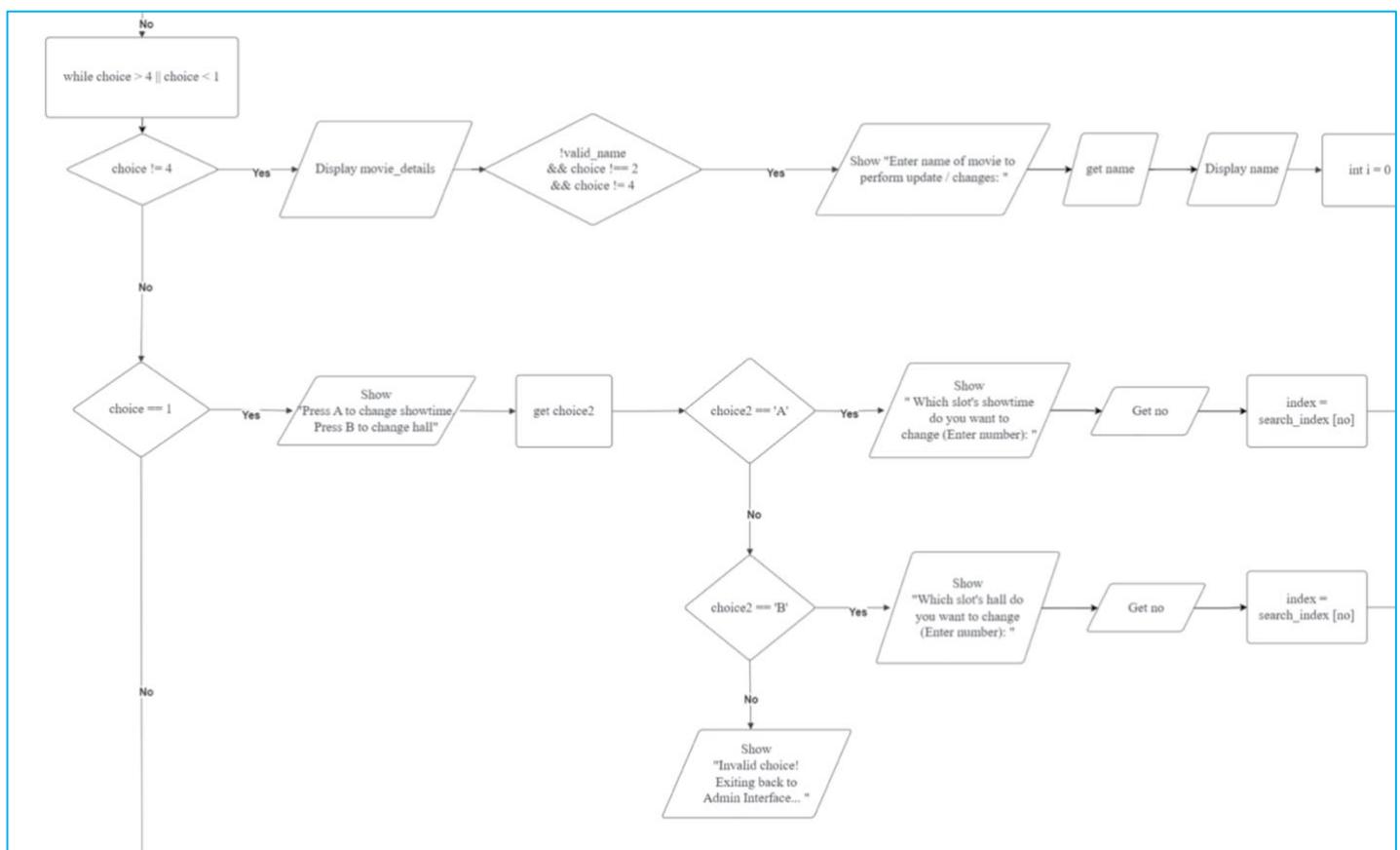


## Void update (part 1.0)

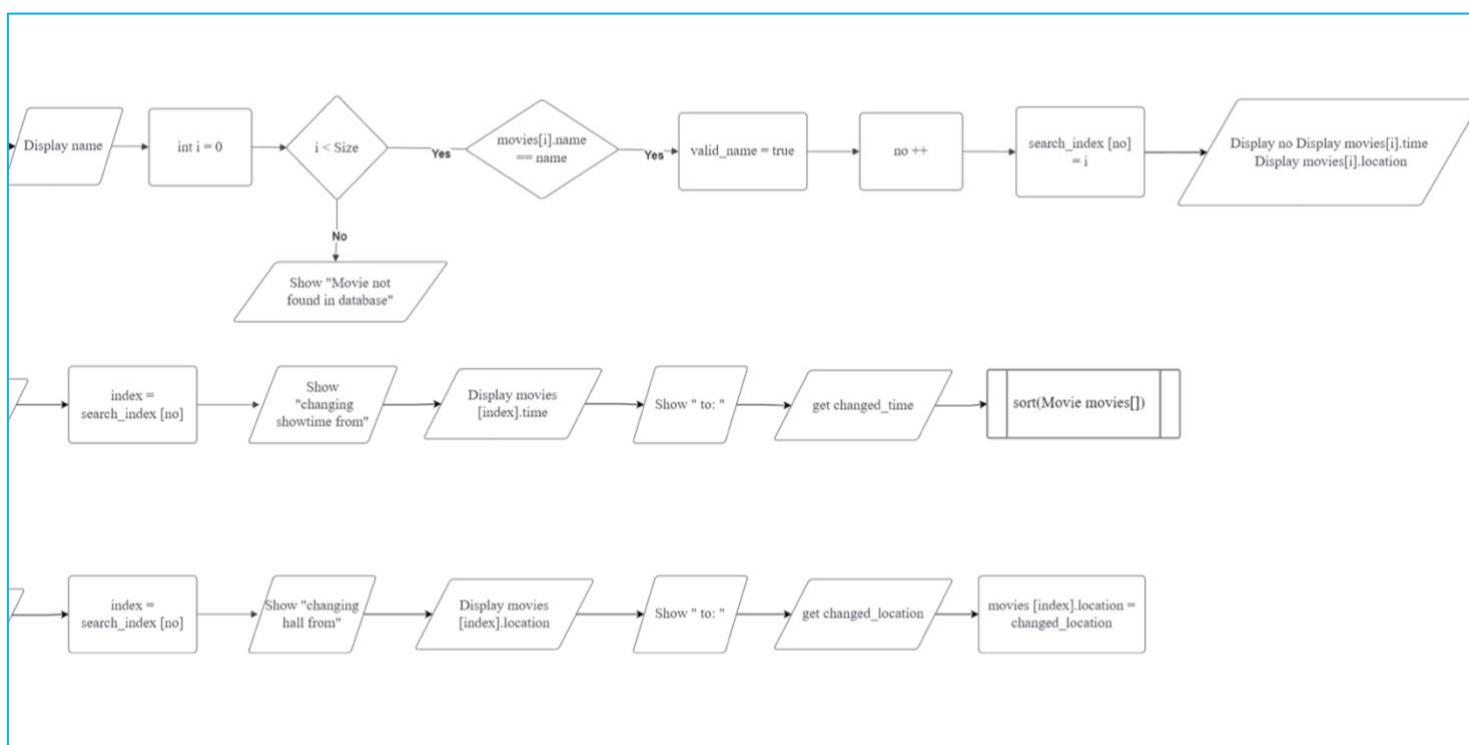
### void update (admin)



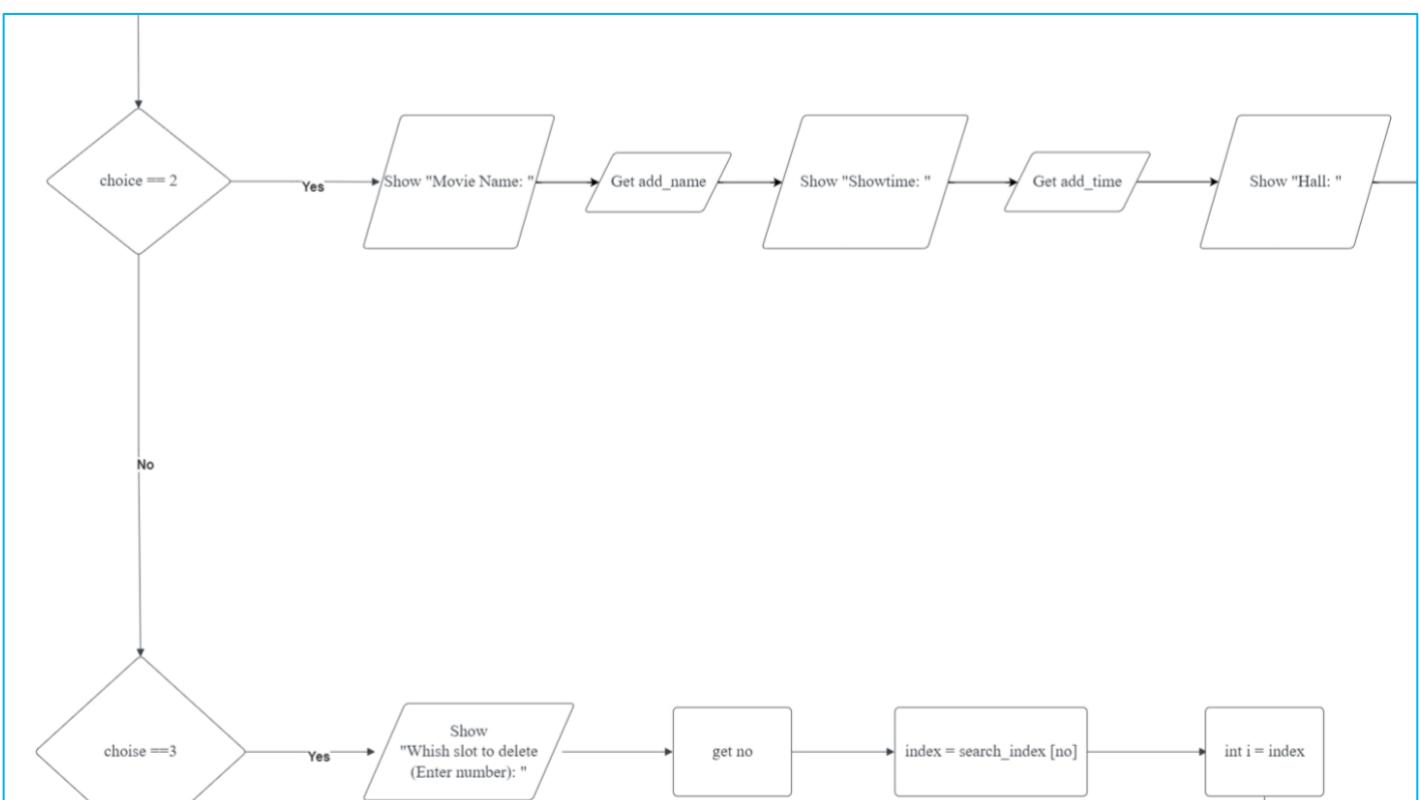
## Void update (part 2.0)



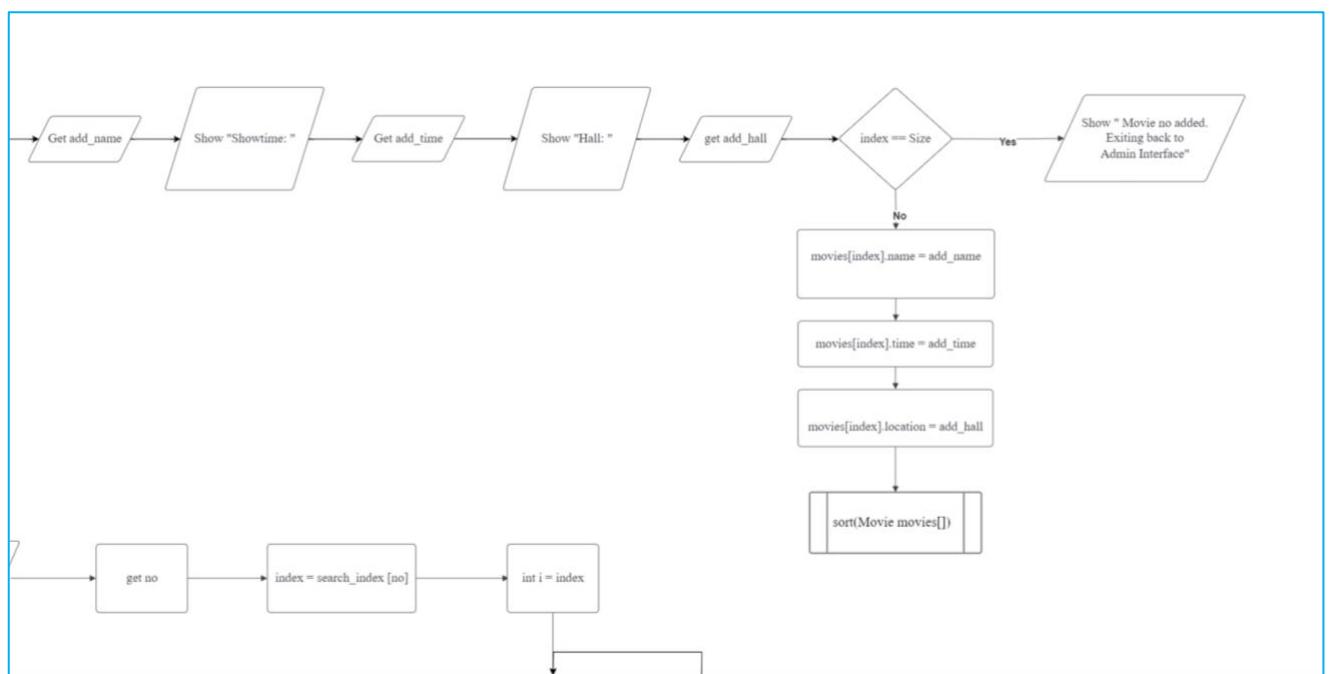
## Void update (part 2.1)



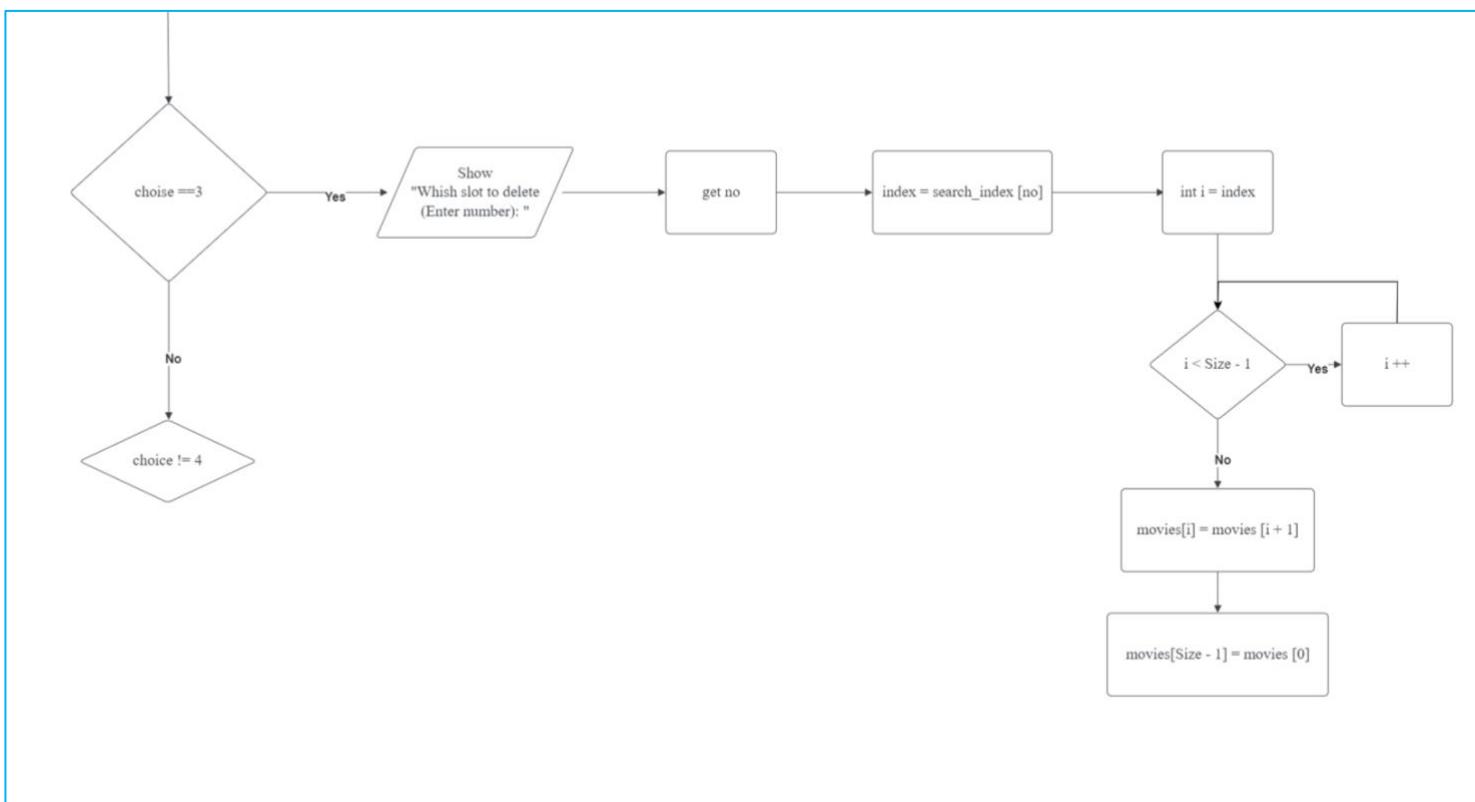
### Void update (part 3.0)



### Void update (part 3.1)



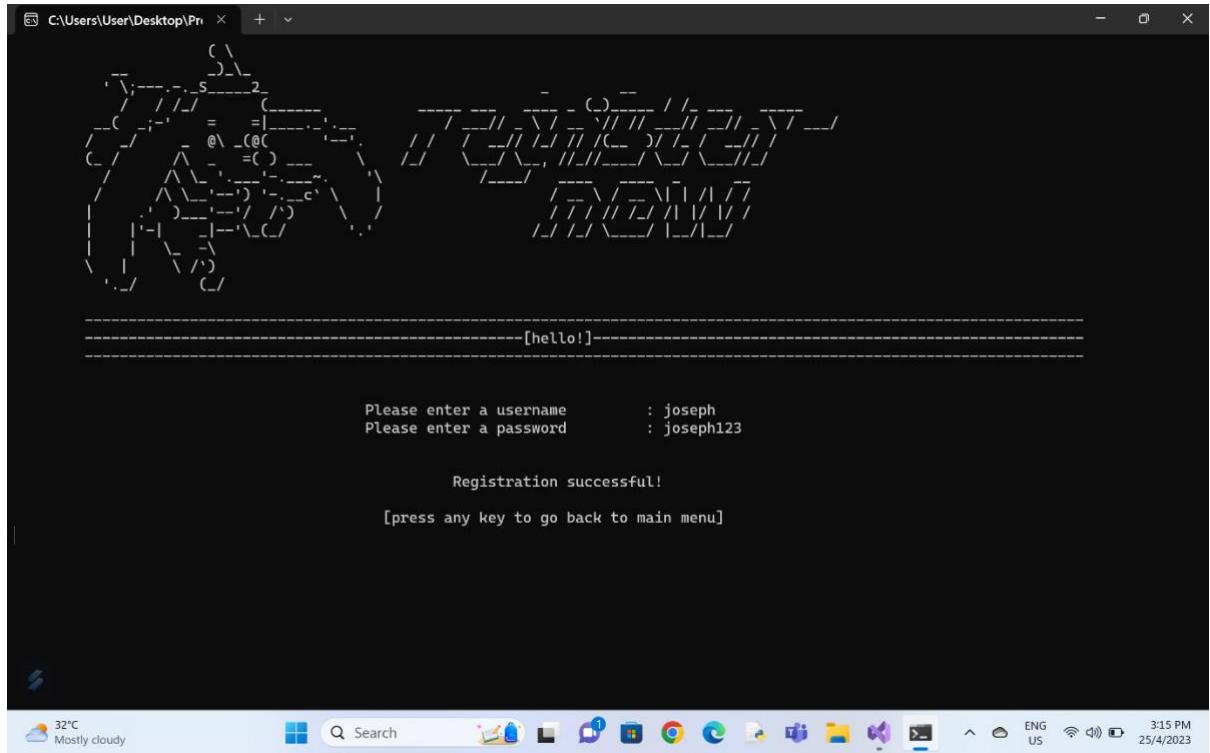
## Void update (part 4.0)



## Test Cases

### 1) User Registration

Two user accounts are registered and stored in users.txt.



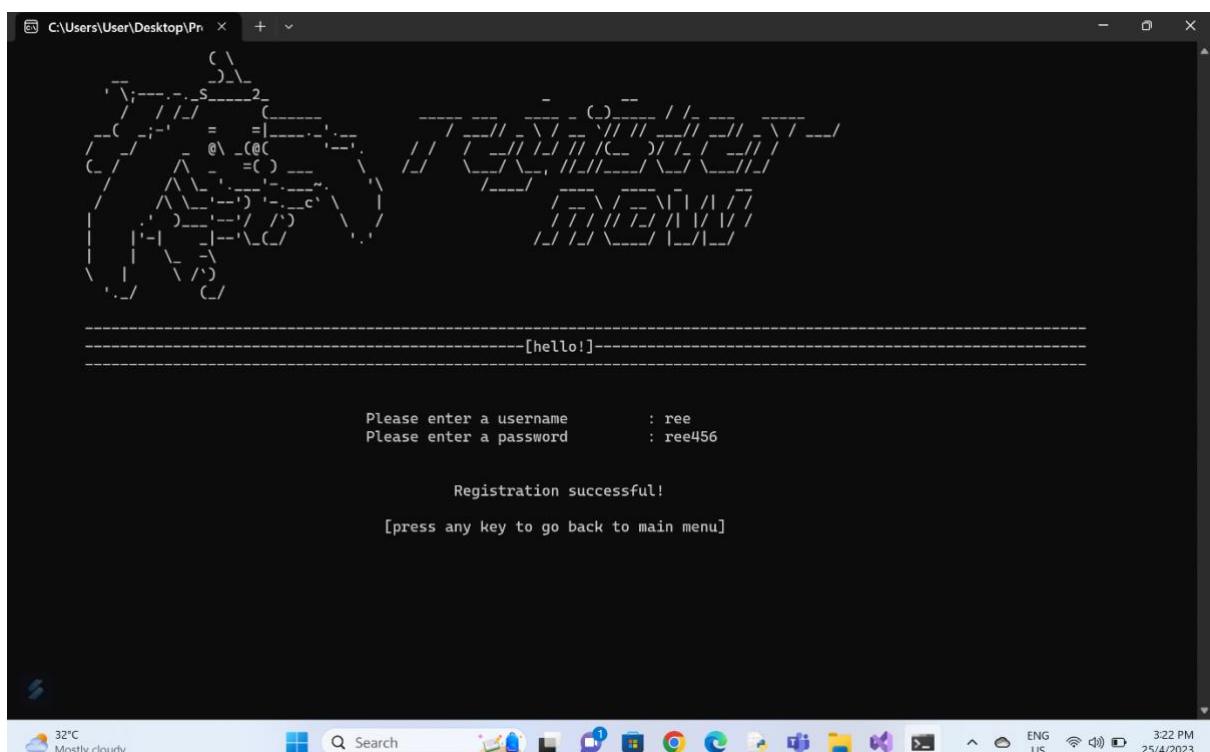
```
C:\Users\User\Desktop\Prn > + ^

[hello!]

Please enter a username      : joseph
Please enter a password      : joseph123

Registration successful!
[press any key to go back to main menu]
```

The screenshot shows a Windows command-line interface (CMD) window titled 'C:\Users\User\Desktop\Prn'. It displays a registration process for a user named 'joseph'. The user is prompted to enter a username ('joseph') and a password ('joseph123'). After the input, a success message 'Registration successful!' is displayed, followed by a prompt to press any key to return to the main menu. The taskbar at the bottom shows the date and time as 3:15 PM on 25/4/2023.



```
C:\Users\User\Desktop\Prn > + ^

[hello!]

Please enter a username      : ree
Please enter a password      : ree456

Registration successful!
[press any key to go back to main menu]
```

The screenshot shows a second instance of a Windows command-line interface (CMD) window titled 'C:\Users\User\Desktop\Prn'. It displays the registration process for another user named 'ree'. The user is prompted to enter a username ('ree') and a password ('ree456'). After the input, a success message 'Registration successful!' is displayed, followed by a prompt to press any key to return to the main menu. The taskbar at the bottom shows the date and time as 3:22 PM on 25/4/2023.

## 2) User Login

Below is the case for wrong password during user login.

The screenshot shows a terminal window titled 'C:\Users\User\Desktop\Prn'. The window contains a decorative ASCII art banner at the top. Below it, a prompt '[login now]' is visible. The terminal then asks for a username ('please enter your username: joseph') and a password ('please enter your password: joseph234'). An error message follows: 'Error: invalid username or password' and 'you'll be redirected back to the main menu'. A progress bar labeled 'Loading: [|||||...|%]' is shown at the bottom. The taskbar at the bottom of the screen displays the Windows Start button, a search bar, and various pinned application icons. The system tray shows the date and time as '25/4/2023' and '3:16 PM'.

Below are the cases of matching password and username.

The screenshot shows a terminal window titled 'C:\Users\User\Desktop\Prn'. The window contains a decorative ASCII art banner at the top. Below it, a prompt '[login now]' is visible. The terminal then asks for a username ('please enter your username: joseph') and a password ('please enter your password: joseph123'). A success message follows: 'login successful! Welcome joseph' and 'press any key to continue...'. The taskbar at the bottom of the screen displays the Windows Start button, a search bar, and various pinned application icons. The system tray shows the date and time as '25/4/2023' and '3:16 PM'.

```
C:\Users\User\Desktop\Prn <--> [login now]-----  
please enter your username: ree  
please enter your password: ree456  
  
login successful! Welcome ree  
press any key to continue...  
  
32°C  
Mostly cloudy
```

Below is the case when the user had already login but haven't log out, enter choice 2 to login.

```
C:\Users\User\Desktop\Pr +   
 \^`.;`l`'`l  
 |@`';`];`[;`-`j  
 {`.;`'`L`';`;  
 -`i`-`  
 -`u`QJ,  
 -`}`--`---  
 ||`T`-`---`  
 ;`c`u-`  
 :`||;`j`  
 .`j`u-?  
 -`--  
 i`  
  
-----[main menu]-----  
  
 1 dont have an account? REGISTER NOW  
 2 existing user? Login  
 3 log out  
 4 admin login  
 5 exit  
  
 Enter your choice: 2  
  
 You are already logged in  
 Redirecting you to the user menu...  
  
 Loading: [|||||] 100%  
  
 32°C ENG US 3:18 PM  
 Mostly cloudy
```

### **3) User Logout**

These are the cases where users want to log out their account.

```
C:\Users\User\Desktop\Pr... + ▾
[main menu]
1 dont have an account? REGISTER NOW
2 existing user? Login
3 log out
4 admin login
5 exit

Enter your choice: 3
Log out?
Press 'Y' to logout
Press any other key to go back to the main screen
:n
You're still logged in
Press enter to go back to the main menu
32°C
Mostly cloudy
Search
Clouds
File
Edit
View
Insert
Format
Tools
Help
ENG
US
3:21 PM
25/4/2023
```

```
C:\Users\User\Desktop\Pr... + ▾
[main menu]
1 dont have an account? REGISTER NOW
2 existing user? Login
3 log out
4 admin login
5 exit

Enter your choice: 3
Log out?
Press 'Y' to logout
Press any other key to go back to the main screen
:y
You have successfully logged out of the system
Press enter to go back to the main menu
32°C
Mostly cloudy
Search
Clouds
File
Edit
View
Insert
Format
Tools
Help
ENG
US
3:21 PM
25/4/2023
```

### 3) Book ticket

This is one of the core processes of this program, to book tickets. After typing in the movie name, it displays the available slots for the movie. If movie name does not exist in the memory, it will also return error messages.

```
C:\Users\User\Desktop\Pr + - _ X
-----[movies and showtimes]-----
Movies | Showtime | Halls
-----[minions]
Minions | 08:00 | Hall A
Minions | 10:00 | Hall B
Minions | 12:00 | Hall C
-----[angry bird]
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
-----[Harry Potter]
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
-----[megamind]
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A

Enter the name of movie to book tickets: Minons

Minons | Showtime| Halls
Movie not found in database.
Please be careful with the spellings and capital letters.

Enter the name of movie to book tickets: |

32°C Mostly cloudy Search
ENG US 3:17 PM 25/4/2023

C:\Users\User\Desktop\Pr + - _ X
-----[movies and showtimes]-----
Movies | Showtime | Halls
-----[minions]
Minions | 10:00 | Hall B
Minions | 12:00 | Hall C
-----[angry bird]
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
-----[Harry Potter]
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
-----[megamind]
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A

Enter the name of movie to book tickets: Minons

Minons | Showtime| Halls
Movie not found in database.
Please be careful with the spellings and capital letters.

Enter the name of movie to book tickets: Minions

Minions | Showtime| Halls
(1) | 08:00 | Hall A
(2) | 10:00 | Hall B
(3) | 12:00 | Hall C

Enter the slot number to book your ticket: 2

Press enter to continue...

32°C Mostly cloudy Search
ENG US 3:17 PM 25/4/2023
```

```
C:\Users\User\Desktop\Prn x + - ○ X
-----
[movies and showtimes]
-----
Movies | Showtime | Halls
Minions | 08:00 | Hall A
Minions | 10:00 | Hall B
Minions | 12:00 | Hall C
-----
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
-----
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
-----
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A
-----
Enter the name of movie to book tickets: Minions
-----
Movies | Showtime | Halls
(1) | 08:00 | Hall A
(2) | 10:00 | Hall B
(3) | 12:00 | Hall C
-----
Enter the slot number to book your ticket: 3
Press enter to continue...
| ⚡ 32°C Mostly cloudy 3:19 PM 25/4/2023
```

```
C:\Users\User\Desktop\Prn x + - ○ X
/____/
-----
[movies and showtimes]
-----
Movies | Showtime | Halls
Minions | 08:00 | Hall A
Minions | 10:00 | Hall B
Minions | 12:00 | Hall C
-----
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
-----
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
-----
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A
-----
Enter the name of movie to book tickets: Angry Bird
-----
Movies | Showtime | Halls
(1) | 08:00 | Hall B
(2) | 10:00 | Hall C
-----
Enter the slot number to book your ticket: 1
Press enter to continue...
| ⚡ 32°C Mostly cloudy 3:23 PM 25/4/2023
```

```
C:\Users\User\Desktop\Prn + - x [movies and showtimes]
-----[movies and showtimes]-----
Movies | Showtime | Halls
Minions | 08:00 | Hall A
Minions | 10:00 | Hall B
Minions | 12:00 | Hall C
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A
Enter the name of movie to book tickets: Megamind
-----[Megamind]-----
Megamind | Showtime| Halls
(1) | 15:00| Hall C
(2) | 17:00| Hall C
(3) | 20:00| Hall A
Enter the slot number to book your ticket: 1
Press enter to continue...
| 32°C Mostly cloudy | Search | Edge | File Explorer | Task View | Start | Taskbar | Network | ENG US | 3:27 PM | 25/4/2023
```

```
C:\Users\User\Desktop\Prn + - x [movies and showtimes]
-----[movies and showtimes]-----
Minions | 10:00 | Hall B
Minions | 12:00 | Hall C
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A
Enter the name of movie to book tickets: Harry Petter
-----[Harry Petter]-----
Harry Petter | Showtime| Halls
Movie not found in database.
Please be careful with the spellings and capital letters.
Enter the name of movie to book tickets: Harry Potter
-----[Harry Potter]-----
Harry Potter | Showtime| Halls
(1) | 10:00| Hall C
(2) | 18:00| Hall B
(3) | 22:00| Hall C
Enter the slot number to book your ticket: 3
Press enter to continue...
| 32°C Mostly cloudy | Search | Edge | File Explorer | Task View | Start | Taskbar | Network | ENG US | 3:23 PM | 25/4/2023
```

```

C:\Users\User\Desktop\Prn x + -
-----[movies and showtimes]-----
Movies | Showtime | Halls
Minions | 08:00 | Hall A
Minions | 10:00 | Hall B
Minions | 12:00 | Hall C
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A

Enter the name of movie to book tickets: Harry Potter

Harry Potter | Showtime| Halls
(1) | 10:00| Hall C
(2) | 18:00| Hall B
(3) | 22:00| Hall C

Enter the slot number to book your ticket: 2
Press enter to continue...

```

32°C Mostly cloudy    Search    ⌂    3:28 PM 25/4/2023

Below is the ticket booking for newly added movie slot by the admin, Replicas. It shows that whatever changes admin updated, the movie slots table will also updated simultaneously.

```

C:\Users\User\Desktop\Prn x + -
-----[movies and showtimes]-----
Movies | Showtime | Halls
Minions | 08:00 | Hall A
Minions | 15:00 | Hall B
Minions | 17:00 | Hall C
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
Angry Bird | 23:00 | Hall A
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
Megamind | 15:00 | Hall C
Megamind | 17:00 | Hall C
Megamind | 20:00 | Hall A
Replicas | 15:00 | Hall D

Enter the name of movie to book tickets: Replicas

Replicas | Showtime| Halls
(1) | 15:00| Hall D

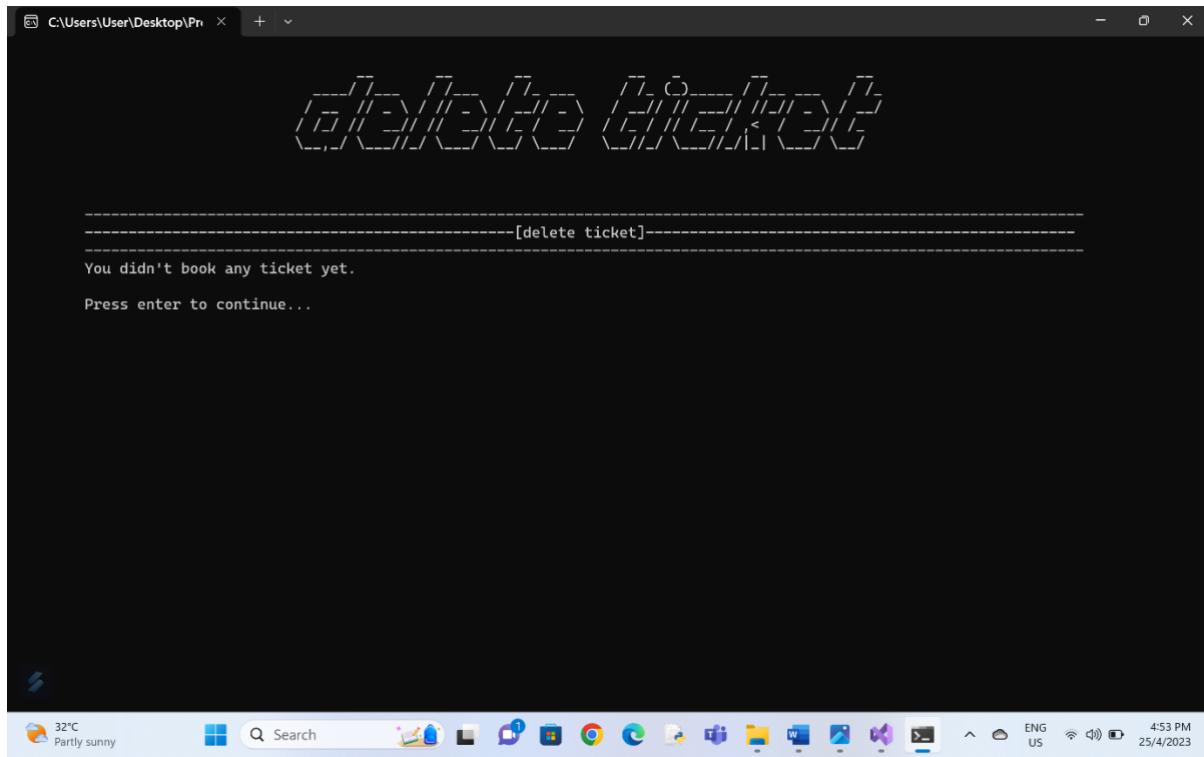
Enter the slot number to book your ticket: 1
Press enter to continue...

```

33°C Mostly cloudy    Search    ⌂    3:37 PM 25/4/2023

#### **4) Edit Booked tickets**

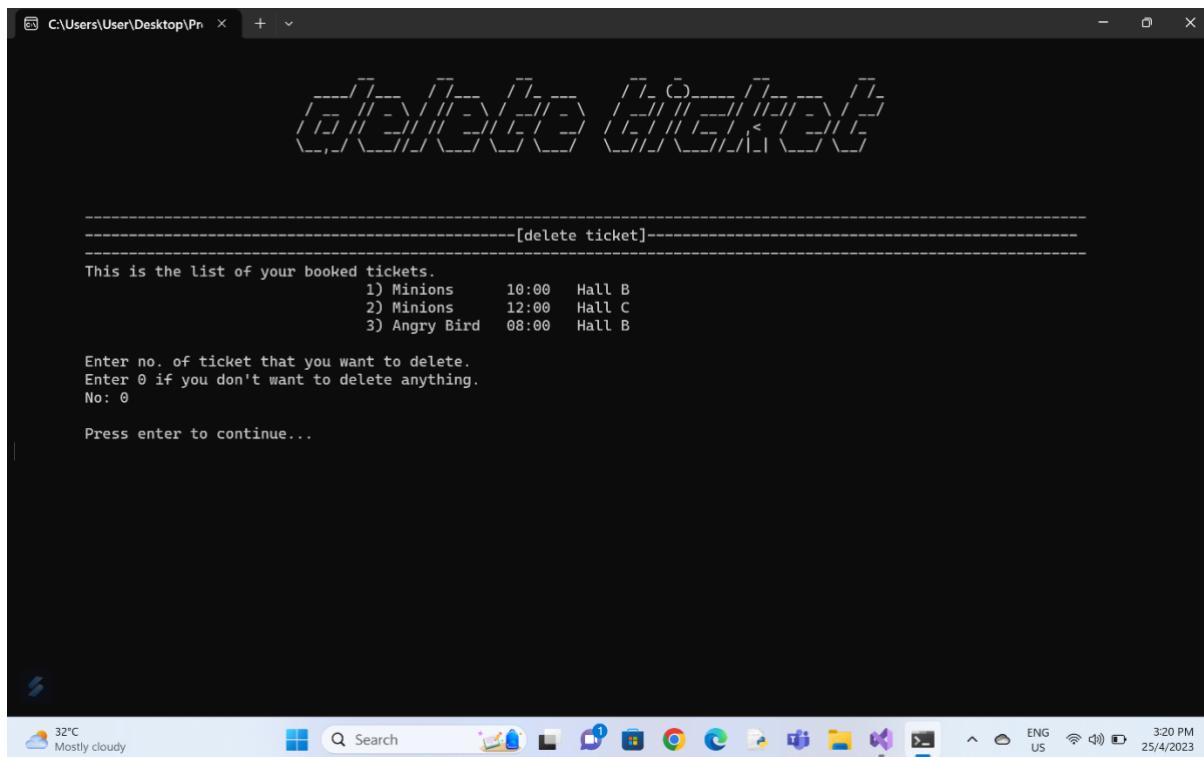
Below shows the case when no ticket has been booked yet when user choose to edit booked tickets.



```
C:\Users\User\Desktop\Prn + - X
-----
[delete ticket]
-----
You didn't book any ticket yet.
Press enter to continue...
-----
```

This screenshot shows a terminal window on a Windows operating system. The title bar indicates the path 'C:\Users\User\Desktop\Prn'. The window contains a single line of text: 'You didn't book any ticket yet.' followed by a prompt 'Press enter to continue...'. The desktop background is visible at the bottom, showing the taskbar with various icons and the system tray.

Below shows the interfaces when there are tickets booked already when user choose to edit booked tickets.



```
C:\Users\User\Desktop\Prn + - X
-----
[delete ticket]
-----
This is the list of your booked tickets.
1) Minions    10:00   Hall B
2) Minions    12:00   Hall C
3) Angry Bird 08:00   Hall B

Enter no. of ticket that you want to delete.
Enter 0 if you don't want to delete anything.
No: 0

Press enter to continue...
-----
```

This screenshot shows a terminal window displaying a list of booked tickets. The list includes three entries: 'Minions' at 10:00 in Hall B, 'Minions' at 12:00 in Hall C, and 'Angry Bird' at 08:00 in Hall B. Below the list, there is a prompt asking the user to enter the number of the ticket they want to delete or '0' if they do not want to delete anything. The user has entered '0'. The terminal also includes a standard Windows-style taskbar at the bottom.

```
C:\Users\User\Desktop\Prn + - X
-----[delete ticket]-----
This is the list of your booked tickets.
1) Angry Bird 08:00 Hall B

Enter no. of ticket that you want to delete.
Enter 0 if you don't want to delete anything.
No: 1

Press enter to continue...  
  
32°C Mostly cloudy  Search  ⌂  3:23 PM 25/4/2023
```

```
C:\Users\User\Desktop\Prn + - X
-----[delete ticket]-----
This is the list of your booked tickets.
1) Megamind 15:00 Hall C
2) Harry Potter 22:00 Hall C
3) Harry Potter 18:00 Hall B

Enter no. of ticket that you want to delete.
Enter 0 if you don't want to delete anything.
No: 2

Press enter to continue...  
  
32°C Mostly cloudy  Search  ⌂  3:28 PM 25/4/2023
```

## 5) Payment

The design here only simulates a credit card payment process. Therefore, any input for credit card number is accepted. Also, noted that after payment, the ticket records are saved to ticket records.txt. As such, all the payment here are able to be seen when admin view ticket records in Test Case 11: Check Ticket Records.

```
C:\Users\User\Desktop\Pr < + X
-----[payment gateway]-----
This is the list of your booked tickets.
1) Minions    10:00    Hall B
2) Minions    12:00    Hall C
3) Angry Bird 08:00    Hall B
Total payment is RM24
Enter your credit card number: 281391
[ Validating your card. This may take a few seconds ... ]
Loading: [ ] 100% Done!||||||||||||||||||||||%
Payment completed.

32°C
Mostly cloudy
Search
ENG US
3:20 PM
25/4/2023
```

```
C:\Users\User\Desktop\Pr < + X
-----[payment gateway]-----
This is the list of your booked tickets.
1) Harry Potter 22:00    Hall C
Total payment is RM8
Enter your credit card number: 911182
[ Validating your card. This may take a few seconds ... ]
Loading: [ ] 100% Done!||||||||||||||||||%
Payment completed.

Press enter to continue...
32°C
Mostly cloudy
Search
ENG US
3:25 PM
25/4/2023
```

```
C:\Users\User\Desktop\Prn + - X
-----[payment gateway]-----
This is the list of your booked tickets.
1) Megamind    15:00    Hall C
2) Harry Potter 18:00    Hall B
-----
Total payment is RM16
-----
Enter your credit card number: 281391
[ Validating your card. This may take a few seconds ... ]
Loading: [ ] 100% Done!||||||||||||||||||||||%
Payment completed.

Press enter to continue...

```



```
C:\Users\User\Desktop\Prn + - X
-----[payment gateway]-----
This is the list of your booked tickets.
1) Replicas    15:00    Hall D
-----
Total payment is RM8
-----
Enter your credit card number: 281391
[ Validating your card. This may take a few seconds ... ]
Loading: [ ] 100% Done!||||||||||||||||||%
Payment completed.

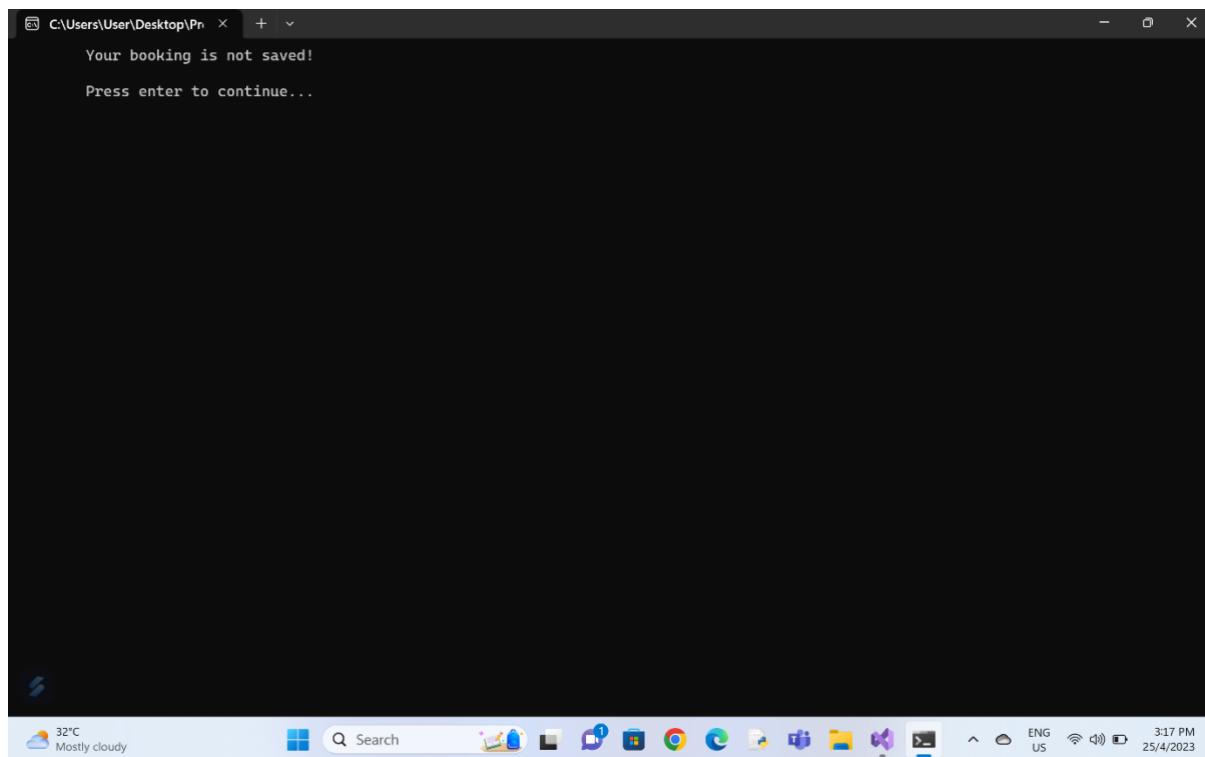
Press enter to continue...

```



## **6) Exit**

This is the exit in user menu, the ticket bookings will not be saved if users choose to exit instead of payment.



## **7) Admin Login**

Below is the case where admin username and password is correctly matched.

The screenshot shows a terminal window titled 'C:\Users\User\Desktop\Pr' with a dark background. At the top, there are several decorative symbols resembling stylized trees or brackets. Below them, a dashed line separates the header from the main content. The text '[authorization page]' is displayed. The terminal then prompts for a username and password:

```
Please enter your username: admin
Please enter your password: password
```

After the password is entered, it displays a welcome message and a loading bar:

```
Welcome, admin!
redirecting to admin menu
Loading: [|||||.....|||||.....|||||.....|||||.....%]
```

The taskbar at the bottom of the screen shows the date and time as '25/4/2023' and '3:29 PM'. It also displays icons for weather (32°C, mostly cloudy), search, and various system and application icons.

Below are the cases where admin username and password are wrongly matched.

The screenshot shows a terminal window titled 'C:\Users\User\Desktop\Pr' with a dark background. At the top, there are several decorative symbols resembling stylized trees or brackets. Below them, a dashed line separates the header from the main content. The text '[authorization page]' is displayed. The terminal then prompts for a username and password:

```
Please enter your username: admin
Please enter your password: dontknow
```

After the password is entered, it displays an error message and a message telling the user to press a key:

```
Incorrect login credentials. Attempts remaining: 2
Press any key to try again...
```

The taskbar at the bottom of the screen shows the date and time as '25/4/2023' and '3:41 PM'. It also displays icons for weather (33°C, mostly cloudy), search, and various system and application icons.

```
C:\Users\User\Desktop\Pr < + >

[authorization page]

Please enter your username: admin
Please enter your password: stilldontknow

Incorrect login credentials. Attempts remaining: 1
Press any key to try again...

33°C
Mostly cloudy
Search
Windows Start
File Explorer
OneDrive
Photoshop
PowerShell
Windows Terminal
Google Chrome
Microsoft Edge
FileZilla
Xbox
File Manager
PowerShell ISE
Visual Studio
Mail
Cloud
ENG
US
Wi-Fi
3:41 PM
25/4/2023
```

```
C:\Users\User\Desktop\Pr < + >

[authorization page]

Please enter your username: admin
Please enter your password: stillstillstilldontknow

Incorrect login credentials. Attempts remaining: 0
You have exceeded the maximum number of login attempts.
redirecting to the main page....
Loading: [██████████] 100%
```

## **8) Change Movie Slot (Showtime/ Hall)**

Below shows the case where admin perform update on changing movie slots.

```
C:\Users\User\Desktop\Pr... + ... ----- [movies and showtimes] -----

| Movies       | Showtime | Halls  |
|--------------|----------|--------|
| Minions      | 08:00    | Hall A |
| Minions      | 10:00    | Hall B |
| Minions      | 12:00    | Hall C |
| Minions      | 15:00    | Hall B |
| Angry Bird   | 08:00    | Hall B |
| Angry Bird   | 10:00    | Hall C |
| Angry Bird   | 23:00    | Hall A |
| Harry Potter | 10:00    | Hall C |
| Harry Potter | 18:00    | Hall B |
| Harry Potter | 22:00    | Hall C |
| Megamind     | 15:00    | Hall C |
| Megamind     | 17:00    | Hall C |
| Megamind     | 20:00    | Hall A |
| Replicas     | 15:00    | Hall A |



Enter the name of movie to perform update/changes: Minionss



| Minionss                                                                                  | Showtime | Halls |
|-------------------------------------------------------------------------------------------|----------|-------|
| Movie not found in database.<br>Please be careful with the spellings and capital letters. |          |       |



Enter the name of movie to perform update/changes:



32°C Mostly cloudy    Search    ⌂    3:33 PM 25/4/2023


```
C:\Users\User\Desktop\Pr... + ... -----

Megamind	20:00	Hall A
Replicas	15:00	Hall A



Enter the name of movie to perform update/changes: Minionss



Minionss	Showtime	Halls
Movie not found in database. Please be careful with the spellings and capital letters.		



Enter the name of movie to perform update/changes: Minions



Minions	Showtime	Halls
(1)	08:00	Hall A
(2)	10:00	Hall B
(3)	12:00	Hall C
(4)	15:00	Hall B



Press 'A' for changing showtime.  
Press 'B' for changing hall.



Enter: A



Which slot's showtime do you want to change (Enter number): 3  
Changing showtime from 12:00 to: 17:00



Loading: [ ] 100% Done!||||||||||||||||||||||%



[ successfully added the changes ]



Press enter to go back to admin update menu...



33°C Mostly cloudy    Search    ⌂    3:34 PM 25/4/2023


```


```

```
C:\Users\User\Desktop\Pr... + - _ x

Minions | 17:00 | Hall C
-----
Angry Bird | 08:00 | Hall B
Angry Bird | 10:00 | Hall C
Angry Bird | 23:00 | Hall A
-----
Harry Potter | 10:00 | Hall C
Harry Potter | 18:00 | Hall B
Harry Potter | 22:00 | Hall C
-----
Meganind | 15:00 | Hall C
Meganind | 17:00 | Hall C
Meganind | 20:00 | Hall A
-----
Replicas | 15:00 | Hall A

Enter the name of movie to perform update/changes: Replicas

Replicas | Showtime| Halls
-----
(1) | 15:00| Hall A

Press 'A' for changing showtime.
Press 'B' for changing hall.

Enter: B

Which slot's hall do you want to change (Enter number): 1
Changing hall from Hall A to: Hall D

Loading: [ ] 100% Done!||||||||||||||||||%
[ successfully added the changes ]
Press enter to go back to admin update menu...

33°C
Mostly cloudy
Search
ENG US
3:35 PM
25/4/2023
```

### **9) Add Movie Slot**

Below shows the case where admin perform update on adding movie slots, he can either add a new movie or existing movie. When adding a new slot for existed movie, the new added movie slot will be grouped together with the same movies instead of added at the back of the table.

```
C:\Users\User\Desktop\Prn < + >
/_/ /_ \___/ |__|__/ /____//_ / / \___/ |__|__//_/_/ / / \___/ /____/
```

---

[movies and showtimes]

Movies	Showtime	Halls
Minions	08:00	Hall A
Minions	10:00	Hall B
Minions	12:00	Hall C
Angry Bird	08:00	Hall B
Angry Bird	10:00	Hall C
Harry Potter	10:00	Hall C
Harry Potter	18:00	Hall B
Harry Potter	22:00	Hall C
Megamind	15:00	Hall C
Megamind	17:00	Hall C
Megamind	20:00	Hall A
Replicas	15:00	Hall A

Movie Name: Minions  
Showtime: 15:00  
Hall: Hall B

Loading: [ ] 100% Done!||||||||||||||||||||||%

[ successfully added the changes ]

Press enter to go back to admin update menu...



32°C Mostly cloudy 3:32 PM 25/4/2023 ENG US

```
C:\Users\User\Desktop\Prn < + >
/_/
```

---

[movies and showtimes]

Movies	Showtime	Halls
Minions	08:00	Hall A
Minions	10:00	Hall B
Minions	12:00	Hall C
Minions	15:00	Hall B
Angry Bird	08:00	Hall B
Angry Bird	10:00	Hall C
Harry Potter	10:00	Hall C
Harry Potter	18:00	Hall B
Harry Potter	22:00	Hall C
Megamind	15:00	Hall C
Megamind	17:00	Hall C
Megamind	20:00	Hall A
Replicas	15:00	Hall A

Movie Name: Angry Bird  
Showtime: 23:00  
Hall: Hall A

Loading: [ ] 100% Done!||||||||||||||||||%

[ successfully added the changes ]

Press enter to go back to admin update menu...



32°C Mostly cloudy 3:33 PM 25/4/2023 ENG US

### **10) Delete movie slot**

Below shows the case where admin perform update on delete movie slot. An updated new movie is displayed after admin finish performing updates.

```
C:\Users\User\Desktop\Prn x + - [movies and showtimes]-----  


| Movies       | Showtime | Halls  |
|--------------|----------|--------|
| Minions      | 08:00    | Hall A |
| Minions      | 10:00    | Hall B |
| Minions      | 15:00    | Hall B |
| Minions      | 17:00    | Hall C |
| Angry Bird   | 08:00    | Hall B |
| Angry Bird   | 10:00    | Hall C |
| Angry Bird   | 23:00    | Hall A |
| Harry Potter | 10:00    | Hall C |
| Harry Potter | 18:00    | Hall B |
| Harry Potter | 22:00    | Hall C |
| Megamind     | 15:00    | Hall C |
| Megamind     | 17:00    | Hall C |
| Megamind     | 20:00    | Hall A |
| Replicas     | 15:00    | Hall D |

  
Enter the name of movie to perform update/changes: Minions  
  


| Movies | Showtime | Halls  |
|--------|----------|--------|
| (1)    | 08:00    | Hall A |
| (2)    | 10:00    | Hall B |
| (3)    | 15:00    | Hall B |
| (4)    | 17:00    | Hall C |

  
Which slot to delete (Enter number): 2
```

```
C:\Users\User\Desktop\Pr x + - 33°C ENG
Mostly cloudy US 3:36 PM
25/4/2023

[[{"Movie": "Minions", "Hall": "Hall A", "Showtime": "08:00"}, {"Movie": "Minions", "Hall": "Hall B", "Showtime": "15:00"}, {"Movie": "Minions", "Hall": "Hall C", "Showtime": "17:00"}, {"Movie": "Angry Bird", "Hall": "Hall B", "Showtime": "08:00"}, {"Movie": "Angry Bird", "Hall": "Hall C", "Showtime": "10:00"}, {"Movie": "Angry Bird", "Hall": "Hall A", "Showtime": "23:00"}, {"Movie": "Harry Potter", "Hall": "Hall C", "Showtime": "10:00"}, {"Movie": "Harry Potter", "Hall": "Hall B", "Showtime": "18:00"}, {"Movie": "Harry Potter", "Hall": "Hall C", "Showtime": "22:00"}, {"Movie": "Megamind", "Hall": "Hall C", "Showtime": "15:00"}, {"Movie": "Megamind", "Hall": "Hall C", "Showtime": "17:00"}, {"Movie": "Megamind", "Hall": "Hall A", "Showtime": "20:00"}, {"Movie": "Replicas", "Hall": "Hall D", "Showtime": "15:00"}]]-----[movies and showtimes]-----

| Movies       | Showtime | Halls  |
|--------------|----------|--------|
| Minions      | 08:00    | Hall A |
| Minions      | 15:00    | Hall B |
| Minions      | 17:00    | Hall C |
| Angry Bird   | 08:00    | Hall B |
| Angry Bird   | 10:00    | Hall C |
| Angry Bird   | 23:00    | Hall A |
| Harry Potter | 10:00    | Hall C |
| Harry Potter | 18:00    | Hall B |
| Harry Potter | 22:00    | Hall C |
| Megamind     | 15:00    | Hall C |
| Megamind     | 17:00    | Hall C |
| Megamind     | 20:00    | Hall A |
| Replicas     | 15:00    | Hall D |



Changes are successfully updated.



press any key to go back to admin menu interface...


```

### **11) Check ticket records**

Below shows the case where admin check ticket records, all the records here is obtained from the payment at Test Case 5: Payment.

```
C:\Users\User\Desktop\Prn + [ticket records]
-----
UserID: joseph
Minions 10:00 Hall B
Price: 8

UserID: joseph
Minions 12:00 Hall C
Price: 8

UserID: joseph
Angry Bird 08:00 Hall B
Price: 8

UserID: ree
Harry Potter 22:00 Hall C
Price: 8

UserID: joseph
Megamind 15:00 Hall C
Price: 8

UserID: joseph
Harry Potter 18:00 Hall B
Price: 8

UserID: joseph
Replicas 15:00 Hall D
Price: 8

Current total income: 56

press any key to go back to admin menu interface...
-----
```

## **12) Some extra input validations that was not shown yet**

This is the section to demonstrate all the other input validation available in the program that was not able to be shown in the previous 11 test cases, starting at main menu.

Below is the validation case at user menu interface.

```
C:\Users\User\Desktop\Pr > + <

[user menu]
1) Book tickets
2) Edit booked ticket
3) Perform payment
4) Exit
Enter your option: 6

Invalid input. Please enter again.
Enter your option: 9

Invalid input. Please enter again.
Enter your option: 2

33°C
Mostly cloudy
Search
Windows Start
File Explorer
OneDrive
Edge
Google Chrome
FileZilla
File Manager
PowerShell
Task View
System
3:41 PM
ENG US
25/4/2023
```

Below is the validation case at admin menu interface.

```
C:\Users\User\Desktop\Pr > + <

[admin update menu]
1      Change movie slot : Showtime / Hall
2      Add movie slot
3      Delete movie slot
4      Exit

Enter your option: 5

Invalid input. Please press enter to continue...
33°C
Mostly cloudy
Search
Windows Start
File Explorer
OneDrive
Edge
Google Chrome
FileZilla
File Manager
PowerShell
Task View
System
3:42 PM
ENG US
25/4/2023
```

## **Source Code**

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>

//loading bar animation for decoration
#include <chrono>
#include <thread>
using namespace std;

//admin username and password
const string admin_username = "admin";
const string admin_password = "password";

//user count
const int MAX_USERS = 100; // maximum number of registered users
int numUsers = 0; // number of registered users

//used for registration and login
struct User {
    string username;
    string password;
} users[MAX_USERS];

//loading bar
const int duration = 5; // total duration of the loading bar in seconds
const int delay = 100; // delay between progress updates in milliseconds

//maximum 50 movies slot
const int Size = 50;

//movies slots
struct Movie {
    string name, time, location;
};

//tickets booked
struct Ticket {
```

```

        string name, time, location;
    };

//functions prototypes
//by tharini
void registerUser();
void showLoadingBar(int duration, int delay);
bool loginUser(string& login_username);
bool adminLogin();
void displayadminmenu();
void displaymenu();
void loadUsers();

//by joseph
void readMovies(string moviesOS[], Movie movies[]);
void sort(Movie movies[]);
bool check_exist(string name, string moviesOS[]);
int null_index(string moviesOS[]);
void set_movieOS(Movie movies[], string moviesOS[]);
int check_index(Movie movies[]);
int starting_index(Movie movies[]);
void movie_details(string moviesOS[], Movie movies[]);
void update(Movie movies[], string moviesOS[]);
int check_tickets_index(Ticket tickets[]);
void bookMovies(string moviesOS[], Movie movies[], string login_username);

//by william
void check_record();
void check_out(string login_username, Ticket tickets[], int index);

int main() {

    //read and set up movie list memory
    Movie movies[Size] = { };
    string moviesOS[Size] = { }; //moviesOS is movies on show
    readMovies(moviesOS, movies);
}

```

```

char choice;
int admin_choice;
string login_username;

//loadUsersFromFile();
bool isLoggedIn = false;
bool validInput = false;

do {
    system("CLS");
    displaymenu();
    cout << "\n\t\t\t\tEnter your choice: ";
    cin >> choice;

    switch (choice) {
        //register user
        case '1':
            system("CLS");
            registerUser();
            cin.ignore();
            break;
        //user login
        case '2':
            if (!isLoggedIn) {
                system("CLS");
                isLoggedIn = loginUser(login_username);
                showLoadingBar(duration, delay);
                if (isLoggedIn)
                    bookMovies(moviesOS, movies, login_username); //go to user menu
            }
            else {
                cout << "\n\tYou are already logged in" << endl;
                cout << "\tRedirecting you to the user menu...\n" << endl;

                showLoadingBar(duration, delay);

                cin.ignore();

                system("CLS");
                bookMovies(moviesOS, movies, login_username); //go to user menu
            }
    }
}

```

```

    }

    break;

//user log out
case '3':
{
    char logout;

    if (!isLoggedIn) {
        cout << "\n\tYou're not logged in\n\tPress enter to go back to the main menu" << endl;
        cin.ignore();
        cin.ignore();
        displaymenu();
        break;
    }

    cout << "|tLog out?" << endl;
    cout << "\n\tPress 'Y' to logout" << endl;
    cout << "\n\tPress any other key to go back to the main screen" << endl;
    cout << "\t:";

    cin >> logout;

    if (logout == 'Y' || logout == 'y') {
        isLoggedIn = false;
        cout << "\n\tYou have successfully logged out of the system\n\tPress enter to go back to the
main menu" << endl;
        cin.ignore();
        cin.ignore();
        displaymenu();

        showLoadingBar(duration, delay);
    }
    else {
        cout << "\n\tYou're still logged in\n\tPress enter to go back to the main menu" << endl;
        cin.ignore();
        cin.ignore();
    }
    break;
}

//admin login
case '4':

```

```

{

int max_attempts = 3;
int attempts = 0;
bool login_successful = false;

while (attempts < max_attempts && !login_successful) {
    system("CLS");
    if (adminLogin()) {
        login_successful = true;
        displayadminmenu();

        cout << "\n\t\t\t\t\tEnter your choice: ";
        cin >> admin_choice;
        cout << endl << endl;

        while (admin_choice >= 1 && admin_choice <= 2) {

            if (admin_choice == 1) {

                //start of admin update menu
                update(movies, moviesOS);

                //update movies.txt
                ofstream outFile("movies.txt", ios::out);

                for (int i = starting_index(movies); i < Size; i++) {
                    outFile << movies[i].name << endl;
                    outFile << movies[i].time << " " << movies[i].location << endl << endl;
                }

                outFile.close();

                //Display updated movie table
                cout << "The new movie table:\n\n";
                movie_details(moviesOS, movies);
                cout << "Changes are successfully updated." << endl;
            }
        }
    }
}
else if (admin_choice == 2)
}

```



```
        system("Pause");
        return 0;
    }

//function definitions
//add previously added users to the struct
void loadUsers() {
    ifstream myfile("users.txt");
    if (myfile.is_open()) {
        string username, password;
        while (getline(myfile, username, ' ') && getline(myfile, password)) {
            users[numUsers].username = username;
            users[numUsers].password = password;
            numUsers++;
        }
        myfile.close();
    } else {
        cout << "Error: unable to open file" << endl;
    }
}

//user registration
void registerUser() {
    //register user banner
    cout << "\t      ( \\" << endl;
    cout << "\t _ _)_\\_ " << endl;
    cout << "\t ' \\\---.-_S____2_ _ _ _ " << endl;
    cout << "\t / ///_ (____ _ _ _ _ _ _ _ _ _ _ /_ _ _ _ " << endl;
    cout << "\t __( _;- = =|_____.'_ _ / ____//_ \\ / _ `// / _ // _ \\ / _ /" << endl;
    cout << "\t _/_ @\\_(@(_ '--. // / _ // / /(_ )//_ / _ // /" << endl;
    cout << "\t(_/ _ \\\_ _ =() _ \\\_ /_ \\\_ / \\\_, // _ / _ / \\\_ / _ /" << endl;
    cout << "\t / _ \\\_ '._-._~. \\\_ /_ _ _ _ _ _ " << endl;
    cout << "\t / _ \\\_ '--' '-.c` \\\_ / _ \\\_ / \\\_ / _ /" << endl;
    cout << "\t| . )__'--' /) _ / _ / _ / / /" << endl;
    cout << "\t| |- _|-"\_(_/ _ !' / _ / _ / \\\_ / _ /" << endl;
```

```

cout << "\t| | \_ -|| " << endl;
cout << "\t\| | \|/" " << endl;
cout << "\t'./_ (/_ " << endl;
cout << "\n\t-----";
----- " << endl;
cout << "\t-----[hello!]-";
----- " << endl;
cout << "\t-----";
----- " << endl;
cin.ignore();

if (numUsers >= MAX_USERS) {
    cout << "Error: maximum number of users reached" << endl;
    return;
}

string username, password;
cout << "\n\n\t\t\tPlease enter a username\t\t: ";
getline(cin, username);

cout << "\t\t\tPlease enter a password\t\t: ";
getline(cin, password);

users[numUsers].username = username;
users[numUsers].password = password;
numUsers++;

// Save user to users.txt file
ofstream myfile;
myfile.open("users.txt", ios_base::app);
myfile << username << " " << password << endl;
myfile.close();

cout << "\n\t\t\tRegistration successful!" << endl;
cout << "\n\t\t\t[press any key to go back to main menu]" << endl;
}

//user login verification
bool loginUser(string& login_username) {
    loadUsers();
}

```

```

//login user banner
cout << "\t _ , - " << endl;
cout << "\t.' ` .___.>','- .. - . " << endl;
cout << "\t`-. ; .--' . - ._.@; " << endl;
cout << "\t ; !_.----_.!' / .[_.@^'. " << endl;
cout << "\t ; / : ! ; :..._ ` . _ _ _ _ " << endl;
cout << "\t : ; ;[ _T-; `!-.`-. // _ _ _ _ _ _ _ _ _ " <<
endl;
cout << "\t \\. -: ;`~-=_88p. __.}.-' / _ \\\/_\\////_\\ \\\_//_\\/_\\/_/" <<
endl;
cout << "\t `~-_.-' \\\_ /L._ Y', P$T888; "" // _// _\\////_\\/_//_\\/_//_\\/_/" <<
endl;
cout << "\t .'-_.-' / ;$$$$$$]8P; _// _/_\\/_//_/_\\/_/_/_/_/_/_/_/_/_/_/" <<
endl;
cout << "\t \\/_ // 'Y$$P' ^,' " << endl;
cout << "\t ;\\_ `.\_ " << endl;
cout << "\t ]_\\ \\_;" << endl;
cout << "\n\n\t-----"
----- " << endl;
cout << "\t-----[login now]-----"
----- " << endl;
cout << "\t-----"
----- " << endl;

string login_password;
cin.ignore();
cout << "\n\n\t|t|t|t|t|t|please enter your username: ";
getline(cin, login_username);
cout << endl;
cout << "\t|t|t|t|t|t|please enter your password: ";
getline(cin, login_password);

for (int i = 0; i < numUsers; i++) {
    if (users[i].username == login_username && users[i].password == login_password) {
        cout << "\n\n\t login successful! Welcome " << users[i].username << endl;
        cout << "\t press any key to continue..." << endl;
        cin.ignore();
        system("CLS");
        return true;
    }
}
cout << "\n\n\tError: invalid username or password" << endl;

```







```

cout << "\t  ,: ||j ;   / _\\// / / / _\\/_\\/_`_\\/_`_\\/_`_\\/_`_/" << endl;
cout << "\t .j  u- ;: / _\\// / / / _\\/_\\/_\\/_\\/_\\/_\\(_ ) " << endl;
cout << "\t ' ~ -- ;: \\_\_// / \_\_, / \\_\_// / / / / \\_\_// / / / / \_\_, //___/ " <<
endl;
cout << "\t .;      j     / ____/ ,;   " << endl;
cout << "\t i       {           ;   " << endl;

cout << "\n\n\t-----  
----- " << endl;
cout << "\t-----[main menu]-----  
----- " << endl;
cout << "\t-----  
----- " << endl;

cout << "\n\t\t\t\tdont have an account? REGISTER NOW" << endl;
cout << "\t\t\t\texisting user?      Login" << endl;
cout << "\t\t\t\t3\tdlog out" << endl;
cout << "\t\t\t\t4\tdadmin login" << endl;
cout << "\t\t\t\t5\textit" << endl;

}

//find the starting index for the null structs in array
int check_tickets_index(Ticket tickets[]) {
    for (int i = 0; i < Size; i++) {
        if (tickets[i].name == "" && tickets[i].time == "" && tickets[i].location == "")
            return i;
    }
    return 50;
}

//check for existence of same movie in moviesOS array
bool check_exist(string name, string moviesOS[]) {
    for (int i = 0; i < Size; i++) {
        if (name == moviesOS[i])
            return true;
    }
    return false;
}

//check for first position without null string
int null_index(string moviesOS[]) {

```

```

int pos = 0;
for (int i = 0; i < Size; i++) {
    if (moviesOS[i] != "") {
        pos++;
    } else
        break;
}

return pos;
}

//find out movies on show without repeating movies
void set_movieOS(Movie movies[], string moviesOS[]) {
    for (int i = 0; i < Size; i++) {
        string name = movies[i].name;

        if (!check_exist(name, moviesOS)) {
            int index = null_index(moviesOS);
            moviesOS[index] = name;
        }
    }
}

//find the starting index for the null structs in array
int check_index(Movie movies[]) {
    for (int i = 0; i < Size; i++) {
        if (movies[i].name == "" && movies[i].time == "" && movies[i].location == "")
            return i;
    }

    return 50;
}

//find starting index that contains movie slot in movies array
int starting_index(Movie movies[]) {
    for (int i = 0; i < Size; i++) {
        if (movies[i].name == "" && movies[i].time == "" && movies[i].location == "");
        else
            return i;
    }

    return 50;
}

```



```

cout << "\t\t\t\t / __ `__ // __ `__ \\\\/__ \\\/_ / __ `__ \\\/_ \\\/_ " << endl;
cout << "\t\t\t\t /_ / / /_ / / / / / / / / / / / / _ / / / / / / " << endl;
cout << "\t\t\t\t\_\_\_,/\_\_,_// / / / / / / / / / / / / / \_\_\_// / / / \_\_,/\_ " << endl;
cout << endl;
cout << endl;
cout << "\n\n\t-----\n-----" << endl;
cout << "\t-----[admin update menu]\n-----" << endl;
cout << "\t-----\n-----" << endl;

cout << "\n\t\t\t\tChange movie slot : Showtime / Hall" << endl;
cout << "\t\t\t\tAdd movie slot" << endl;
cout << "\t\t\t\tDelete movie slot" << endl;
cout << "\t\t\t\tExit" << endl;

cout << "\n\tEnter your option: ";
cin >> choice;
cin.ignore();
cout << endl;

if (choice > 4 || choice < 1) {
    cout << "\n\tInvalid input. Please press enter to continue..." << endl;
    cin.ignore();
}
}

} while (choice > 4 || choice < 1);

system("CLS");

//display movie list as reference when not exit
if (choice != 4)
    movie_details(moviesOS, movies);

//check for valid movie name, display specific movie details if valid
while (!valid_name && choice != 2 && choice != 4) {
    cout << "\n\tEnter the name of movie to perform update/changes: ";
    getline(cin, name);
    cout << endl << endl;
    cout << "\t\t\t\t" << left << setw(30) << name << " | Showtime" << setw(15) << " | Halls"
<< endl;
    cout << "\t\t\t\t-----" << endl;

//Display specific movie details for admin as reference
for (int i = 0; i < Size; i++) {

```

```

if (movies[i].name == name) {
    valid_name = true;
    no++;
    search_index[no] = i; //collect index of specific movie
    cout << "\t\t\t\t(" << no << ")" << right << setw(29) << "| " << setw(10) <<
movies[i].time << "| " << movies[i].location << endl;
}
}

if (!valid_name)
    cout << "\t\t\t\tMovie not found in database.\n\t\t\t\tPlease be careful with the spellings
and capital letters.\n\t\t\t-----" << endl;
}

//change showtime or hall
if (choice == 1) {

    cout << "\n\t\t\t\tPress 'A' for changing showtime.\n\t\t\t\tPress 'B' for changing
hall.\n\tEnter: ";
    cin >> choice2;
    cout << endl;

    //change showtime
    if (choice2 == 'A') {
        cout << "\tWhich slot's showtime do you want to change (Enter number): ";
        cin >> no;
        index = search_index[no];
        cout << "\tChanging showtime from " << movies[index].time << " to: ";
        cin.ignore();
        getline(cin, changed_time);
        movies[index].time = changed_time;
        sort(movies);

        showLoadingBar(duration, delay);
        cout << "\n\t\t\t\t[ successfully added the changes ]";
        cout << "\n\tPress enter to go back to admin update menu..." << endl;
        cin.ignore();
    }
}

//change hall
else if (choice2 == 'B') {
    cout << "\tWhich slot's hall do you want to change (Enter number): ";
    cin >> no;
    index = search_index[no];
    cout << "\tChanging hall from " << movies[index].location << " to: ";
    cin.ignore();
    getline(cin, changed_location);
    movies[index].location = changed_location;
}

```







```

cout << "\t\t\t\tMovie not found in database.\n\t\t\t\tPlease be careful with the spellings
and capital letters.\n\t\t\t-----" << endl;
}

cout << "\tEnter the slot number to book your ticket: ";
cin >> no;
index = search_index[no];
tickets[ticket_index].name = movies[index].name;
tickets[ticket_index].time = movies[index].time;
tickets[ticket_index].location = movies[index].location;
ticket_index++;
}
//delete booked tickets
else if (option == 2) {

    cout << "\n\t\t\t-----" << endl;
    cout << "\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t\t\t\t\t-----" << endl;
    cout << "\t\t\t\t\t\t\t\t\t\t-----" << endl;
    cout << endl;

    cout << "\n\n\t-----"
        "-----" << endl;
    cout << "\t-----[delete ticket]"
        "-----" << endl;
    cout << "\t-----"
        "-----" << endl;

    int num;
    index = check_tickets_index(tickets);

    if (index == 0)
        cout << "\tYou didn't book any ticket yet." << endl;
    else {
        cout << "\tThis is the list of your booked tickets." << endl;

        for (int i = 0; i < index; i++) {
            cout << "\t\t\t\t" << i + 1 << " ) " << tickets[i].name << "\t" << tickets[i].time << "\t" <<
            tickets[i].location << endl;
        }

        cout << endl;
        cout << "\tEnter no. of ticket that you want to delete.\n\tEnter 0 if you don't want to delete
anything." << endl;
        cout << "\tNo: ";
        cin >> num;
        if (num != 0) {

```



```

showLoadingBar(duration, delay);
cout << "\tPayment completed." << endl;

//update ticket records
check_out(login_username, tickets, index);

option = 4;
}
}

//exit
else
    cout << "\tYour booking is not saved!" << endl;

cin.ignore();
cout << "\n\tPress enter to continue..." << endl;
cin.ignore();

} while (option != 4);
}

//admin: check ticket records
void check_record() {

system("CLS");
string userID, username, ticket, dummy;
double profit, total_profit = 0;
char discard;

cout << "\n\t\t\t\t _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ " << endl;
cout << "\t\t\t\t / / ( ) _ _ / / _ _ _ _ / / _ _ _ _ _ _ _ _ _ _ _ _ _ _ / / _ _ " <<
endl;
cout << "\t\t\t\t / _ / / _ / / / _ \\\\ / _ / / _ / _ \\\\ / _ / / _ \\\\ / _ / / _ / _ / " << endl;
cout << "\t\t\t\t / _ / / / _ / , < / _ / / _ / / / _ / / / / / ( _ ) " << endl;
cout << "\t\t\t\t \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / _ \\\\ / " <<
endl;
cout << endl;
cout << "\n\n\t-----";
----- " << endl;
cout << "\t-----[ticket records]-----";
----- " << endl;
cout << "\t-----";
----- " << endl;

ifstream inRecord("ticket records.txt");

//read in ticket records.txt
if (inRecord.is_open()) {

```

```

while (!inRecord.eof()) {
    inRecord >> userID;
    getline(inRecord, username);
    getline(inRecord, ticket);
    inRecord >> profit;
    getline(inRecord, dummy);
    inRecord.get(discard);

    cout << "\tUserID: " << username << endl;
    cout << "\t" << ticket << endl;
    cout << "\tPrice: " << profit << endl << endl;
    total_profit += profit;

}
inRecord.close();

cout << "\n\tCurrent total income: " << total_profit << endl;
}

else
    cout << "\tError: Unable to open ticket records.txt!";

cin.ignore();
}

```

```

//sort movies in ascending order according to time
void sort(Movie movies[]) {
    for (int i = 0; i < Size - 1; i++) {
        for (int j = 0; j < Size - 1; j++) {
            if (movies[j].time > movies[j + 1].time) {
                Movie temp = movies[j];
                movies[j] = movies[j + 1];
                movies[j + 1] = temp;
            }
        }
    }
}

```

```

//read in movies.txt
void readMovies(string moviesOS[], Movie movies[]) {
    ifstream inMovies("movies.txt");

    if (inMovies.is_open()) {
        int i = 0;

        while (!inMovies.eof()) {

```

```

getline(inMovies, movies[i].name);
inMovies >> movies[i].time;
inMovies.ignore();
getline(inMovies, movies[i].location);
inMovies.ignore();
i++;
}
inMovies.close();
}
else
cout << "\tUnable to open movies.txt!";

//sort the movies array ascendingly according to time
sort(movies);
}

//update ticket record.txt by each purchase
void check_out(string login_username, Ticket tickets[], int index) {
ofstream outRecord("ticket records.txt", ios::app);

for (int i = 0; i < index; i++) {
    outRecord << "User: " << login_username << endl;
    outRecord << tickets[i].name << " " << tickets[i].time << " " << tickets[i].location << endl;
    outRecord << "8" << endl;
}
outRecord.close();
}

```