

```

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

class FrontDesk {
    private String username;
    private String password;
    private Theater theater;

    public FrontDesk(String username, String password, Theater
theater) {
        this.username = username;
        this.password = password;
        this.theater = theater;
    }

    public boolean login(String enteredUsername, String
enteredPassword) {
        return username.equals(enteredUsername) &&
password.equals(enteredPassword);
    }

    public void updatePassword(String newPassword) {
        password = newPassword;
        System.out.println("Password updated successfully.");
    }

    public void viewSeatingArrangement(String date, String showTime) {
        theater.displaySeatingArrangement(date, showTime);
    }

    public void bookTicket(String date, String showTime, int
seatNumber) {
        theater.bookTicket(date, showTime, seatNumber);
    }

    public void viewBookingStatus() {
        theater.displayBookingStatus();
    }
}

class Theater {
    private Map<String, Map<String, boolean[]>> seatingArrangement;
    private Map<String, Integer> ticketPrices;

    public Theater() {

```

```

        seatingArrangement = new HashMap<>();
        ticketPrices = new HashMap<>();
    }

    public void addShow(String date, String showTime, int totalSeats,
int ticketPrice) {
        seatingArrangement.putIfAbsent(date, new HashMap<>());
        seatingArrangement.get(date).put(showTime, new
boolean[totalSeats]);
        ticketPrices.put(date + showTime, ticketPrice);
    }

    public void displaySeatingArrangement(String date, String
showTime) {
        if (seatingArrangement.containsKey(date) &&
seatingArrangement.get(date).containsKey(showTime)) {
            boolean[] seats =
seatingArrangement.get(date).get(showTime);
            System.out.println("Seating Arrangement for " + date + " "
+ showTime + ":");
            for (int i = 0; i < seats.length; i++) {
                System.out.print(seats[i] ? "X " : i + 1 + " ");
            }
            System.out.println();
        } else {
            System.out.println("Invalid date or show time.");
        }
    }

    public void bookTicket(String date, String showTime, int
seatNumber) {
        if (seatingArrangement.containsKey(date) &&
seatingArrangement.get(date).containsKey(showTime)) {
            boolean[] seats =
seatingArrangement.get(date).get(showTime);
            if (seatNumber > 0 && seatNumber <= seats.length &&
!seats[seatNumber - 1]) {
                seats[seatNumber - 1] = true;
                System.out.println("Ticket booked successfully.
Amount: " + ticketPrices.get(date + showTime));
            } else {
                System.out.println("Invalid seat number or seat
already booked.");
            }
        } else {
            System.out.println("Invalid date or show time.");
        }
    }

```

```

    }
}

    public void displayBookingStatus() {
        System.out.println("Booking Status:");
        for (Map.Entry<String, Map<String, boolean[]>> dateEntry :
seatingArrangement.entrySet()) {
            String date = dateEntry.getKey();
            for (Map.Entry<String, boolean[]> showEntry :
dateEntry.getValue().entrySet()) {
                String showTime = showEntry.getKey();
                boolean[] seats = showEntry.getValue();
                int bookedSeats = 0;
                for (boolean seat : seats) {
                    if (seat) {
                        bookedSeats++;
                    }
                }
                System.out.println(date + " " + showTime + ": Booked
Seats: " + bookedSeats);
            }
        }
    }
}

```

```

public class MovieTicketBooking {
    public static void main(String[] args) {
        Theater theater = new Theater();
        theater.addShow("2023-12-10", "15:00", 20, 600);
        theater.addShow("2023-12-10", "18:00", 15, 700);

        FrontDesk frontDesk = new FrontDesk("admin", "password",
theater);

        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\n1. Login\n2. Update Password\n3.
View Seating Arrangement\n" +
                "4. Book Ticket\n5. View Booking Status\n6.
Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {

```

```

        case 1:
            System.out.print("Enter username: ");
            String enteredUsername = scanner.nextLine();
            System.out.print("Enter password: ");
            String enteredPassword = scanner.nextLine();
            if (frontDesk.login(enteredUsername,
enteredPassword)) {
                System.out.println("Login successful.");
            } else {
                System.out.println("Login failed. Invalid
username or password.");
            }
            break;
        case 2:
            if (frontDesk.login("admin", "password")) {
                System.out.print("Enter new password: ");
                String newPassword = scanner.nextLine();
                frontDesk.updatePassword(newPassword);
            } else {
                System.out.println("Authentication failed.
Please login first.");
            }
            break;
        case 3:
            System.out.print("Enter date (YYYY-MM-DD): ");
            String date = scanner.nextLine();
            System.out.print("Enter show time: ");
            String showTime = scanner.nextLine();
            frontDesk.viewSeatingArrangement(date, showTime);
            break;
        case 4:
            System.out.print("Enter date (YYYY-MM-DD): ");
            String bookDate = scanner.nextLine();
            System.out.print("Enter show time: ");
            String bookShowTime = scanner.nextLine();
            System.out.print("Enter seat number: ");
            int seatNumber = scanner.nextInt();
            scanner.nextLine();
            frontDesk.bookTicket(bookDate, bookShowTime,
seatNumber);

            break;
        case 5:
            frontDesk.viewBookingStatus();
            break;
        case 6:

```

```
        System.out.println("Exiting the program. Thank  
you!");  
        System.exit(0);  
    default:  
        System.out.println("Invalid choice. Please enter a  
valid option.");  
    }  
}  
}
```