title Display Screening Tickets Sold Capacity

actor Staff

control ": CinemaSystem" as CS

participant ": Cinema" as C

entity Screenings

Staff->CS:display(date)

activate CS

CS->C:getScreenings(date)

activate C

loop once for each screenings

C->Screenings:getDate

activate Screenings

Screenings-->C:return date

deactivate Screenings

end

C-->CS:return screenings

deactivate C

CS->CS:updateDisplay()

CS-->Staff:

deactivate CS

title Schedule Move Screening BCE

actor Staff

control ": CinemaSystem" as CS

participant ": Cinema" as C

entity m：Movie

entity s：Screen

entity ": Screening" as Screening

Staff->CS:scheduleScreening(details)

activate CS

CS->C:createScreening(details)

activate C

C->C:m：getMovie(move\_name)

C->C:s：getScreen(screen)

C->\*Screening:Screening(date, start\_time, ticket\_sold, s, m)

activate Screening

Screening-->C:

deactivate Screening

C-->CS:

deactivate C

CS->CS:updateDisplay()

CS-->Staff:

deactivate CS

title Schedule Move Screening ECE

actor Staff

control ": CinemaSystem" as CS

entity "s: Screening" as S

Staff->CS:scheduleScreening(details)

activate CS

CS->CS:checkOverlapScreening()

activate CS

CS-->CS:return true

deactivate CS

CS->Staff:alertOverlapScreening

activate Staff

Staff-->CS:

deactivate Staff

CS-->Staff:

deactivate CS

title Reschedule Movie Screening BCE

actor Staff

control ": ManagementSystem" as CS

participant ": Cinema" as C

entity current：Screenings

entity m：Movie

entity s：Screen

entity selected：Screening

Staff->CS:selectScreening(details)

activate CS

loop for all screenings on the screen

CS->current：Screenings:time = getStartTime()

CS->current：Screenings:s = getScreen()

end

CS-->Staff:

deactivate CS

Staff->CS:rescheduleScreening(details)

activate CS

CS->C:m := getMovie()

CS->C:s := getScreen()

CS->selected：Screening:setMovie(m)

CS->selected：Screening:setScreen(s)

CS->selected：Screening:setStartTime(start\_time)

CS->CS:updateDisplay()

Staff<--CS:

deactivate CS

title Reschedule Movie Screening ECE(a)

actor Staff

control ": ManagementSystem" as CS

entity current：Screenings

Staff->CS:selectScreening(details)

activate CS

loop for all screenings on the screen

CS->current：Screenings:time = getStartTime()

CS->current：Screenings:s = getScreen()

end

CS-->Staff:

deactivate CS

Staff->CS:rescheduleScreening(details)

activate CS

CS->CS:checkOverlapScreening()

activate CS

CS-->CS:return true

deactivate CS

CS->Staff:alertOverlapScreening()

activate Staff

Staff-->CS:

deactivate Staff

Staff<--CS:

deactivate CS

title Reschedule Movie Screening ECE(b)

actor Staff

control ": CinemaSystem" as CS

entity current：Screenings

entity selected：Screening

Staff->CS:selectScreening(details)

activate CS

loop for all screenings on screen

CS->current：Screenings:time=getStartTime()

CS->current：Screenings:s=getScreen()

end

CS-->Staff:

deactivate CS

Staff->CS:rescheduleScreening(details)

activate CS

CS->selected：Screening:getTicketsSold()

activate selected：Screening

CS<--selected：Screening: return ticketsSold

deactivate selected：Screening

CS->CS:checkHasSoldTickets()

activate CS

CS-->CS:return true

deactivate CS

CS->Staff:alertTicketsSold()

activate Staff

Staff-->CS:

deactivate Staff

Staff<--CS:

deactivate CS

title Cancel Movie Screening BCE

actor Staff

control ": ManagementSystem" as CS

entity current：Screenings

entity selected：Screening

Staff->CS:selectScreenings(details)

activate CS

loop for all screenings on screen

CS->current：Screenings:time=getStartTime()

CS->current：Screenings:s=getScreen()

end

CS-->Staff:

deactivate CS

Staff->CS:cancel()

activate CS

CS->Staff:confirm()

activate Staff

Staff-->CS:return "yes"

deactivate Staff

CS->selected：Screening:<< destroy >>

destroy selected：Screening

CS->CS:updateDisplay()

Staff<--CS:

deactivate CS

title Cancel Movie Screening ECE

actor Staff

control ": ManagementSystem" as CS

entity current：Screenings

entity selected：Screening

Staff->CS:selectScreenings(details)

activate CS

loop for all screenings on screen

CS->current：Screenings:time=getStartTime()

CS->current：Screenings:s=getScreen()

end

deactivate CS

Staff->CS:cancelScreening()

activate CS

CS->selected：Screening:getTicketsSold()

activate selected：Screening

CS<--selected：Screening: return ticketsSold

deactivate selected：Screening

CS->CS:checkTicketsSold()

activate CS

CS-->CS:return ticketsSold

deactivate CS

CS->Staff:alertTicketsSold

activate Staff

Staff-->CS:

deactivate Staff

Staff<--CS:

deactivate CS

title Sell Tickets BCE

actor Staff

control ": ManagementSystem" as CS

entity current：Screenings

entity "s: Screen" as s

entity selected：Screening

Staff->CS:selectScreening(details)

activate CS

loop for all screenings on screen

CS->current：Screenings:time=getStartTime()

CS->current：Screenings:s=getScreen()

end

CS-->Staff:

deactivate CS

Staff->CS:sellTicket()

activate CS

CS->s:getCapacity()

activate s

s-->CS:return capacity

deactivate s

CS->selected：Screening:getTicketsSold

activate selected：Screening

selected：Screening-->CS: return ticketsSold

deactivate selected：Screening

CS->selected：Screening:changeTicketsSold()

activate selected：Screening

selected：Screening-->CS:

deactivate selected：Screening

CS->CS:updateDisplay()

Staff<--CS:

deactivate CS

title Sell Tickets ECE

actor Staff

control ": CinemaSystem" as CS

entity current：Screenings

entity "s: Screen" as s

entity selected：Screening

Staff->CS:selectScreening(details)

activate CS

loop for all screenings on screen

CS->current：Screenings:time=getStartTime()

CS->current：Screenings:s=getScreen()

end

CS-->Staff:

deactivate CS

Staff->CS:sellTicket()

activate CS

CS->s:getCapacity()

activate s

s-->CS:return capacity

deactivate s

CS->selected：Screening:getTicketsSold()

activate selected：Screening

selected：Screening-->CS: return ticketsSold

deactivate selected：Screening

CS->CS:checkInsufficientTickets()

activate CS

CS-->CS:return true

deactivate CS

CS->Staff:alertInsufficientTickets()

activate Staff

Staff-->CS:

deactivate Staff

Staff<--CS:

deactivate CS