Movie Theater Software Design

Prepared By: Ahmed Hageb, Aaron Maryfield, Michael Zalameda, Trenton Taylor, and James Yang

March 1, 2023
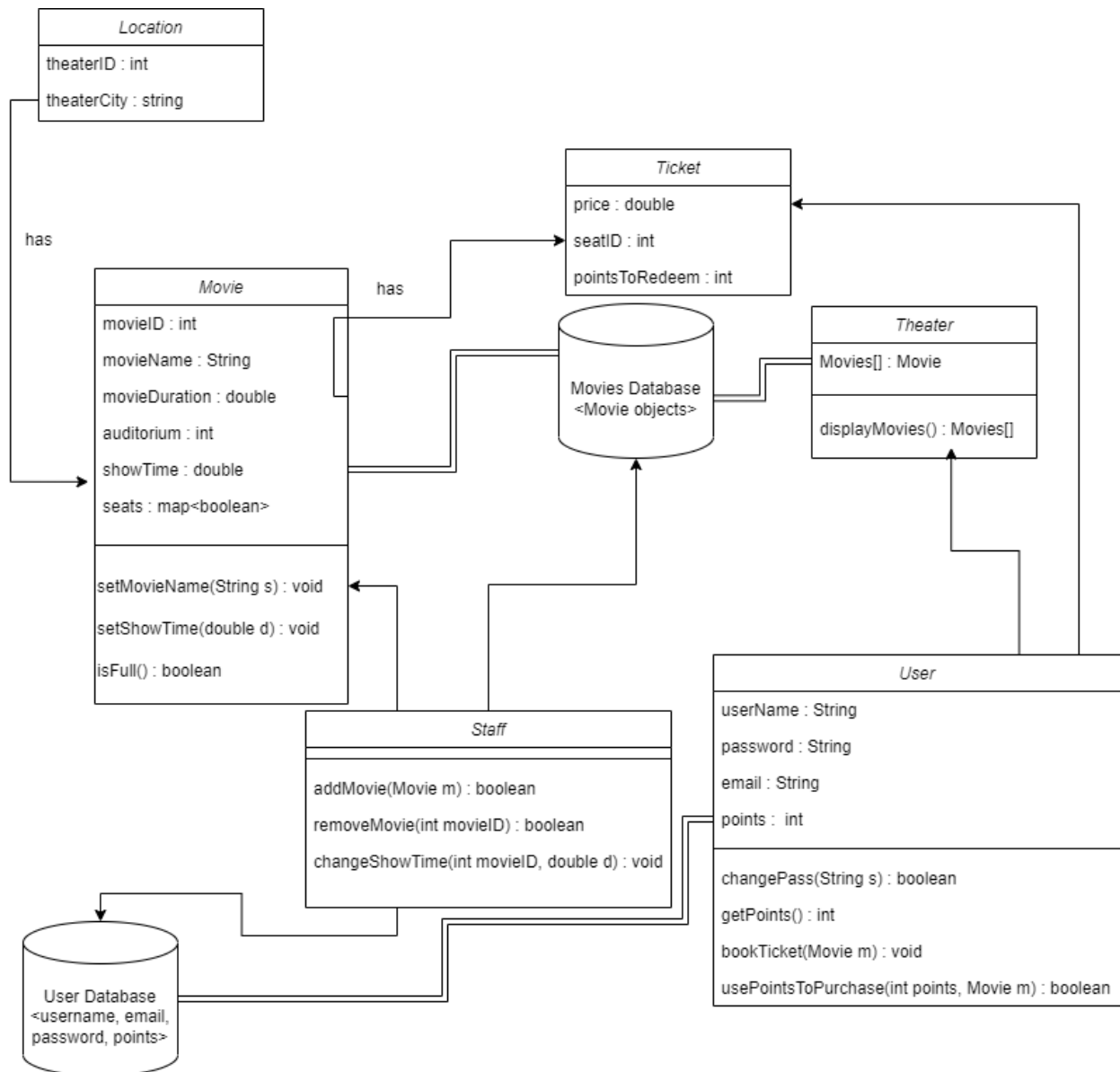
System Description

The movie theater software system is a fully integrated platform created to enhance the functioning of the theater. This system facilitates an effective and hassle-free process for customers to access different services. These services include:

1. Account: Customers can create accounts in order to have an easier experience when using the software as their information such as name, email, and credit card will be saved to their accounts.
2. Ticketing: Customers can purchase tickets online or at the box office using the software.
3. Reservations: Customers can choose to reserve seats for specific showtimes. The reservation system will make sure those seats are held until purchase.
4. Loyalty program: Users that join the loyalty program will have access to special offers and redeemable points on purchases.

Software Architecture Overview

- Architectural diagram of all major components
- UML Class Diagram
- Description of classes
  1. Theater
  2. Movie
  3. Ticket
  4. Staff
  5. User
- Description of attributes
  -
- Description of operations

Location

theaterID : int

theaterCity : string

has

Movie

movieID : int

movieName : String

movieDuration : double

auditorium : int

showTime : double

seats : map<boolean>

setMovieName(String s) : void

setShowTime(double d) : void

isFull() : boolean

has

Ticket

price : double

seatID : int

pointsToRedeem : int

Movies Database
<Movie objects>

Theater

Movies[] : Movie

displayMovies() : Movies[]

Staff

addMovie(Movie m) : boolean

removeMovie(int movieID) : boolean

changeShowTime(int movieID, double d) : void

User

userName : String

password : String

email : String

points : int

changePass(String s) : boolean

getPoints() : int

bookTicket(Movie m) : void

usePointsToPurchase(int points, Movie m) : boolean

User Database
<username, email,
password, points>

Data Dictionary

User:
userName : used to log in
password : used to log in
email
points : used for redeeming for movie tickets
changePass(String s) : method for changing the password
getPoints() : return the number of points you account has
bookTicket(Movie m) : method used to book a ticket for a specific Movie Object
usePointsToPurchase(points, Movie) : return true if your points allow you to get a ticket for the Movie
Staff:
addMovie(Movie m) : allows the staff to add a movie object to the Movie Database
removeMovie(int movieID) : allows the staff to remove a movie from the Movie database
changeShowTime(int movieID, double d) : allows the staff to change the showtime of a movie in the database

Location:
theaterID : used to identify theater by exclusive ID
theaterCity : used to identify city address of theater
Ticket:
price : used to display cost per ticket in USD
seatID : used to identify theater seat number for assigned seating
pointsToRedeem : used to display additional points received for future purchases
Theater:
Movies[] : array list with available movies stored as elements
displayMovies() : returns all elements of Movies[] in most recent to oldest order

* descriptions should be detailed and specify data types, function interfaces, parameters, etc...

Development plan and timeline

- Partitioning of tasks
- Team member responsibilities

Notes:

- to receive credit for the assignment, each group member must push at least one commit to the GitHub repository.