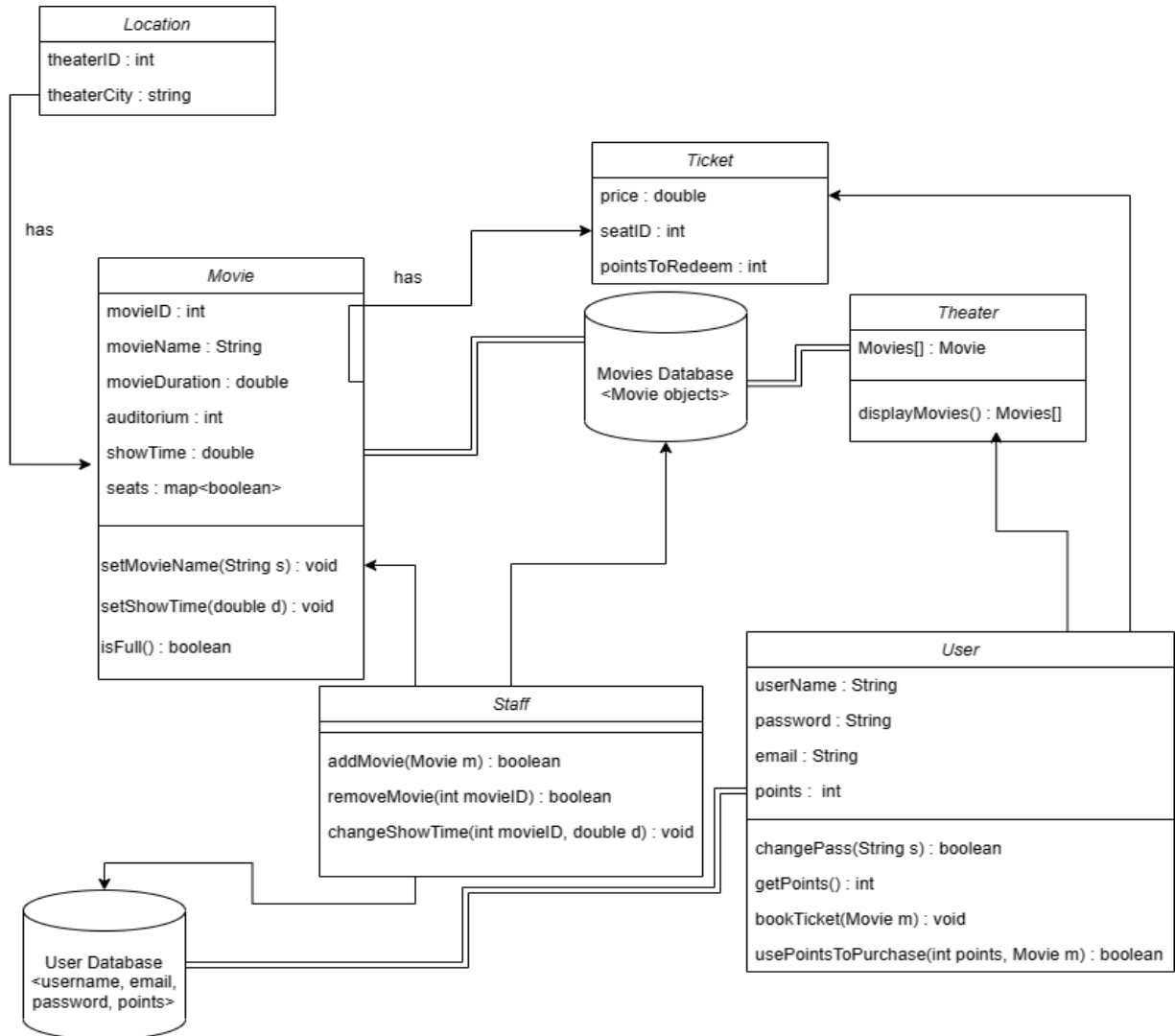


Movie Theater Test Plan

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

March 22, 2023

1. **Software Design Specification:** This is an up-to-date version of your Software Architecture Overview (UML Class Diagram + description) from Assignment 2. If you feel that your design could use further improvements, feel free to make additional changes.



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 the number of points needed to redeem a ticket

Theater:

Movies[] : array list with available movies stored as elements

displayMovies() : returns all elements of Movies[] in most recent to oldest order

Movie:

movieID : used to identify the movie's with its unique ID

movieName : used to identify the movie's name

movieDuration : used to identify the duration of the movie

auditorium : used to display the auditorium number where the movie is held in

showTime : used to display the movie's showtime

seats : used to get the availability of the movie's seats from the Movies database

setMovieName(String s) : allow the staff to set the movie's name

setShowTime(Double d) : allow the staff to set the movie's showtime

isFull() : returns true if the seats for the movies are full, else returns false

2. **Verification Test Plan:** This section lays out test plans for verification. You are required to include at least two test sets (Reminder: a test set consists of one or more tests that targets a feature) for **all** three granularities discussed in class (**unit, functional, system**). In addition to the test itself, you will need to clearly explain your test. Be as detailed as possible, identifying what features of your design you are testing, what the test sets/vectors are, and how your selected test(s) cover the targeted feature(s). (Hint: use or modify your design diagram to indicate the target and scope of the tests, what kind of failures you are covering, etc.). Feel free to use any methods discussed in class.
- 3.

Requirements:

- Create at least 2 test cases for each test set
- Once finished, explain what **feature/integration/function** you tested in **great detail**
- If you think there could be an addition of data members or functions in any class in the UML Diagram, discuss it with 2 other people and then add it to the UML Diagram with a description of its functionality/purpose in the DATA Dictionary

Unit Testing: (White Box)

- Test Set 1: (Trenton)
 -
- Test Set 2: (Aaron)
 -

Functional Testing: (Black Box)

- Test Set 1: (James)
 -
- Test Set 2:
 -

System Testing: (White Box)

- Test Set 1: (Ahmed)
 -
- Test Set 2: (Michael)
 -

Note: to receive credit for the assignment, each group member must push atleast one commit to the github repository.

