Movie Theater Management System

The movie theater management system (MTMS) is designed to manage the operation of a movie theater. The MTMS will allow efficient management of different entities interacting with the theater, such as customer, staff, movie, screen, payment, showtime, and food stand. Given enough time, the project will also include additional functionalities such as an admin page and a login account form for members.

Classes:

1. Customer: represents customers of the movie theater with the availability to have a guest checkout.

Attributes: name, booking id

Methods: getName, getBookingId, CancelBooking

2. Staff: represents employees working at the theater with regards to the staffing schedule and assignment of duties and assign managers and task them with running/operating movie theater (ie food stand inventory, cash register proceeds at close, costs to operate)

Attributes: staff name, Employee ID, schedule, assignment of duties (role), hourly rate

Methods: getEmployeeName, getEmployeeID, assignSchedule, assignEmployeeRoles, getSalary, viewSchedule

3. Movie: represents the movies being shown at the theater. This class will allow users the option to select the movie they would like to see. This class will store information (attributes) such as Movie ID, Title, Genre, Duration, and Screen ID. Following the user's completion of this class after they pick a movie, information regarding the movie’s showtime and assigned screen will appear. Possible methods for this class will include: getMovieInfo, displayMovieDetails, selectMovie

4. Screen: represents the physical screens in the theater. This class will store information such as Screen ID, Screen Name (Screen1, ect.), Screen Type t(Standard v.s. IMAX), Seating Layout, Seating Capacity, and Seat Status (taken/available). Our system will offer users the option to choose between 4 standard movie screens and 1 IMAX screen. Possible methods for this class will include: getScreenInfo, getAvailiableSeats, getSeatStatus, displaySeatingChart, reserveSeat

5. Payment: represents all payment-related operations with the ability to receive payments from customers, ability to pay vendors, and have a database(excel file) where financial accountability and profits can be accessed by managers, and can generate receipts.

Attributes: payment id, booking session id, payment amount, cashier id,

Methods: processPaymentWithCard, processPaymentWithCash, generateReceipt, refundPayment, get_invoiceData(update financial record)

6. Showtime: represents specific showtime schedules of the movies. This class will store information such as Showtime ID, Movie ID, Screen ID, Start Time, and End Time. Each showtime will be linked to one movie. Only one showtime will be in operation on a screen at a time. After the user selects the showtime for their desired movie they will then be moved to the Screen class where they will be able to select their seats. Possible methods for this class will include: getShowtimeInfo, displayShowtimeDetails, getStartTime, getEndTime, setStartTime, setEndTime

7. Food stand: represents the place that sells food and beverage inside the theater. Each food stand is associated with a specific location (more than 1 cashier), available menu items, and the name of the staff managing it. Whenever there is a purchase, it will handle the receipt and update the inventory of the food stand. It also keeps track of the order history.

Attributes: food stand id, menu item id, menu item names, hours of operation, responsible staff. Possible methods include, but not limited to: addMenuItem, updateMenuItem, removeMenuItem, checkAvailableMenuItems, updateInventory, assignStaff, setHoursOfOperation, viewOrderHistory

8. Analytics: provides insights into the sales and revenue of the theater (can also generate reports). It helps to keep track of sales, revenue, and trends, as well as other insights that can be beneficial for the theater to improve the experience of the customers. It will have methods that can generate reports to show these statistics. It can also pull data from the food stand inventory file and generate a report of current stock of the ingredients and products.

Attributes: timePeriod (for other data, only pull the data from excel/txt file when method is called to avoid storing too much data when running the program. This way, only necessary data is pulled. For example, if an instance of the class only calls for the generateFoodSalesReport in the program then it won't need to waste memory storing data for sales data and waste time to pull a lot of data from the sales data file). Possible methods include, but not limited to: generateSalesReport, analyzeTrends, generateFoodSalesReport, pullInventoryData, exportAnalyticsReport.

9. Inventory: represents the inventory for the food stand. It manages the stock of the ingredients and products for the food stand. It will alert the staff when certain ingredients are low in stock.

Attributes: item ID, name, category, quantity, unit price, recipe(so the program can predict whether the stock is about to run out). Possible methods include, but not limited to:

addInventoryItem, updateStock, checkStock, alertLowStock, reorderMoreItems, trackIngredientUsage.