Uml class Diagram::Main

MovieManager

-movieList: vector<Movie>
-deletedMovies: vector<Movie>
-movieSlots: vector<MovieSlot>

+displayMovieList(): void
+viewAllMovies(): void
+registerMovie(): void
+modifyMovie(movieName: string): void
+deleteMovie(movieName: string): void
+restoreMovie(movieName: string): void
+sortMoviesBy(choice: int): void
+searchMovieByName(movieName: string): void
+displayMovieListWithSlots(): void
+addMovieAndSlot(movieName: string, MovieSlot: string): void

Movie

-ID: string
-name: string
-registeredName: string
-registeredID: string
-genre: string
-price: double
-entryDate: Date
-exitDate: Date
-ratingPg: int
-registerTime: time_t
-modifiedTime: time_t
-deletedTime: time_t
-restoredTime: time_t
-slotNumber: int

+Movie()
+Date()
+MovieSlot()

Employee

-ID: string
-name: string
-department: string
-position: string
-shift: string
-joinDate: Date
-exitDate: Date
-salary: double
-attendanceStatus: char
-registeredName: string
-registeredID: string
-registerTime: time_t
-modifiedTime: time_t
-deletedTime: time_t
-restoredTime: time_t
-attendenceTime: time_t
-individualCounts: map

+updateAttendanceStatus(status: char): void
+getAttendanceStatus(): char

Reservation

-CustomerList(): vector<Customer>
-deletedCustomer(): vector<Customer>
+reservedSeats(): unordered_set<string>
+isSeatReserved(const string& seat): bool
+reserveSeat(const string& seat): void
+displayCustomerList(): void
+registerCustomer(...): void
+searchCustomerByName(...): void
+deleteCustomer(...): void
+cancelReservation(): void

Cinema Complex

-movies: List<Movie>
-employees: List<Employee>
-reservations: List<Reservation>
-tickets: List<Ticket>

+displayMovieManagement(): void
+displayEmployeeManagement(): void
+displayReservationSystem(): void
+displayTicketManagement(): void
+login(): void
+loadingBar(): void

EmployeeManager

-employeeList: vector <Employee>
-deletedEmployees: vector <Employee>

+displayEmployeeList()
+registerEmployee()
+modifyEmployee(empName: string)
+deleteEmployee(empName: string)
+restoreEmployee(empName: string)
+searchEmployeeByName(empName: string)
+markAttendanceByName(empName: string)
+displayIndividualAttendance(empName: string)
+displayAttendance()
+modifyAttendance(empName: string, newStatus: char)
+saveAttendanceToFile()

Customer

+ID: string
+name: string
+age: int
+seat: string
+hall: string
+price: double
+entryDate: Date
+arriveDate: Date
+exitDate: Date
+registerTime: time_t

Ticket

-name: string
-movie: string
-hall: string
-seat: string
-ticketId: int
-price: float

+isSeatReserved(): bool
+get(): string
+writeToStream(ofstream&): void

TicketGenerator

-reservedSeats: unordered_set<string>
-tickets: vector<Ticket>

+generateTicket(): Ticket
+checkSeatAvailability(string): bool
+displayAllTickets(): void
+deleteTicket(int): bool

1   *   1   1   *   1   *   1   *