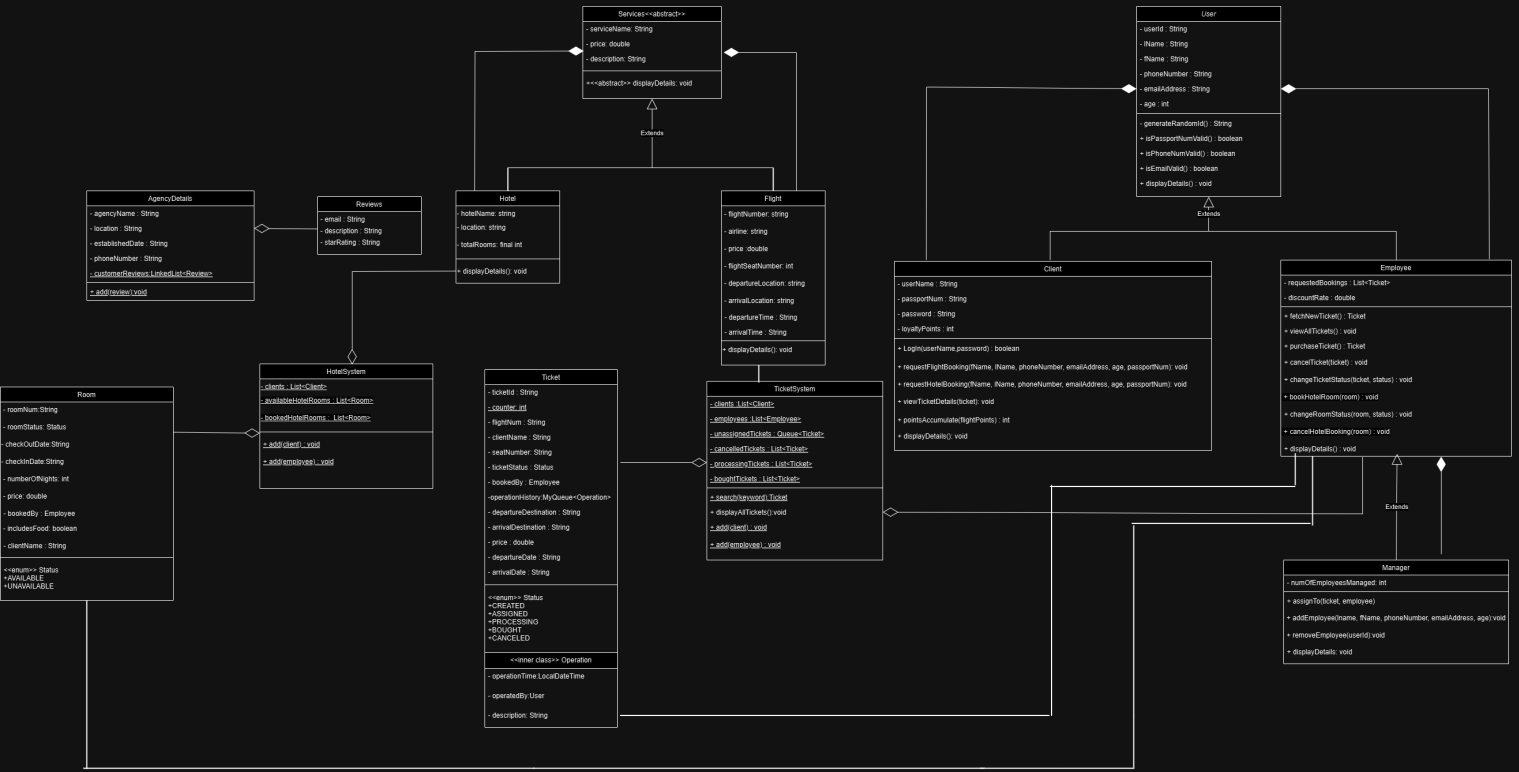
Services<<abstract>>
- serviceName: String
- price: double
- description: String
+<<abstract>> displayDetails: void
Extends
User
- userId : String
- lName : String
- fName : String
- phoneNumber : String
- emailAddress : String
- age : int
- generateRandomId() : String
+ isPassportNumValid() : boolean
+ isPhoneNumValid() : boolean
+ isEmailValid() : boolean
+ displayDetails() : void
Extends
AgencyDetails
- agencyName : String
- location : String
- establishedDate : String
- phoneNumber : String
- customerReviews:LinkedList<Review>
+ add(review):void
Reviews
- email : String
- description : String
- starRating : String
Hotel
- hotelName: string
- location: string
- totalRooms: final int
+ displayDetails(): void
Flight
- flightNumber: string
- airline: string
- price :double
- flightSeatNumber: int
- departureLocation: string
- arrivalLocation: string
- departureTime : String
- arrivalTime : String
+ displayDetails(): void
Client
- userName : String
- passportNum : String
- password : String
- loyaltyPoints : int
+ Login(userName,password) : boolean
+ requestIFlightBooking(fName, lName, phoneNumber, emailAddress, age, passportNum): void
+ requestHotelBooking(fName, lName, phoneNumber, emailAddress, age, passportNum): void
+ viewTicketDetails(ticket): void
+ pointsAccumulate(flightPoints) : int
+ displayDetails(): void
Employee
- requestedBookings : List<Ticket>
- discountRate : double
+ fetchNewTicket() : Ticket
+ viewAllTickets() : void
+ purchaseTicket() : Ticket
+ cancelTicket(ticket) : void
+ changeTicketStatus(ticket, status) : void
+ bookHotelRoom(room) : void
+ changeRoomStatus(room, status) : void
+ cancelHotelBooking(room) : void
+ displayDetails() : void
Extends
HotelSystem
- clients : List<Client>
- availableHotelRooms : List<Room>
- bookedHotelRooms : List<Room>
+ add(client) : void
+ add(employee) : void
Room
- roomNum:String
- roomStatus: Status
- checkOutDate:String
- checkInDate:String
- numberOfNights: int
- price: double
- bookedBy : Employee
- includesFood: boolean
- clientName : String
<<enum>> Status
+AVAILABLE
+UNAVAILABLE
Ticket
- ticketId : String
- counter: int
- flightNum : String
- clientName : String
- seatNumber: String
- ticketStatus : Status
- bookedBy : Employee
-operationHistory:MyQueue<Operation>
- departureDestination : String
- arrivalDestination : String
- price : double
- departureDate : String
- arrivalDate : String
<<enum>> Status
+CREATED
+ASSIGNED
+PROCESSING
+BOUGHT
+CANCELED
<<inner class>> Operation
- operationTime:LocalDateTime
- operatedBy:User
- description: String
TicketSystem
- clients :List<Client>
- employees :List<Employee>
- unassignedTickets : Queue<Ticket>
- cancelledTickets : List<Ticket>
- processingTickets : List<Ticket>
- boughtTickets : List<Ticket>
+ search(keyword):Ticket
+ displayAllTickets():void
+ add(client) : void
+ add(employee) : void
Manager
- numOfEmployeesManaged: int
+ assignTo(ticket, employee)
+ addEmployee(lname, fName, phoneNumber, emailAddress, age):void
+ removeEmployee(userId):void
+ displayDetails: void