**Department of Data Science**



**Airport Management System**

**Submitted By: Submitted To:**

Ijaz Ul Hassan : 221980030 Ather Ikhlaq

Hassan Rashid : 221980028

Sulman Gill : 221980022

Abu Sayyam : 221980033

**Abstract:**

The Airport Management System aims to design and implement a user-friendly interface for both passengers and administrators. It encompasses functionalities such as passenger management, reservation handling, flight details management, and emergency services, all integrated into a Graphical User Interface (GUI).

**Flight Management Class**

**Data Structure**: LinkedHashSet

**Flight Attributes**:

flightNumber: Unique identifier for the flight.

source: Departure location.

destination: Arrival location.

departureTime: Scheduled departure time.

status: Enum representing flight status (e.g., ON\_TIME, DELAYED, CANCELLED).

delayInMinutes: Delay time in case of delay.

**Methods:**

addFlight(): Adds a new flight.

updateFlight(): Updates details of an existing flight.

deleteFlight(): Deletes a flight.

displayFlightDetails(): Displays details of a specific flight.

displayAllFlights(): Displays details of all flights.

delayFlight(): Delays a flight.

getNumberOfFlights(): Returns total flights count.

getFlights(): Returns iterable of all flights.

**Ticket Management**

**Data Structure**: HashMap, undoStack (Stack)

**Attributes**:

ticketNumber

passengerName

seatNumber

flightNumber

**Methods**:

cancelTicket(int ticketNumber)

getTicketsByFlightNumber(String flightNumber)

modifyTicket(int ticketNumber, String text, String text1, String text2)

bookTicket(int ticketNumber, String passengerName, String seatNumber, String flightNumber)

findTicket(int ticketNumber)

displayTicketDetails(int ticketNumber)

displayAllTickets()

getTicketsForFlight(String flightNumber)

searchByFlightNumber(String flightNumber)

getNumberOfTickets()

undoLastBooking()

**Passenger Management**

**Data Structure**:

- **Passengers**:LinkedList

-**UndoStack**: Stack

- **ProcessingQueue**: Queue.

**Attributes:**

passengerName

age

address

gender

Data Structures:

passengers: LinkedList

undoStack: Stack

processingQueue: Queue

**Methods:**

updatePassenger()

addPassenger()

deletePassenger()

undoLastDelete()

displayAllPassengers()

displayPassengerDetails()

**Aircraft Management System**

**Data Structure**:

- **Aircrafts**:ArrayList

**Attributes:**

registrationNumber

aircraftType

seatingCapacity

assignedCrewMember

Data Structures:

aircrafts: ArrayList

highestCapacityAircraft: AircraftManagement

**Methods:**

addAircraft()

updateAircraft()

deleteAircraft()

displayAircraftDetails()

displayAllAircrafts()

assignCrewMembersToAircraft()

searchByRegistrationNumber()

getAircraftWithHighestCapacity()

**Emergency Services**

**Data Structure**:

- **Emergency Services**:Stack

**Attributes:**

emergencyStack: Stack

**Methods:**

EmergencyServices()

addEmergencyRequest(String request)

handleEmergencyRequest()

clearEmergencyRequests()

getEmergencyRequestsCount()

isEmergencyStackEmpty()

viewTopEmergencyRequest()

**Lost and Found**

**Data Structure**:

- **Lost and Found** :TreeMap

**Attributes:**

lostItems: TreeMap

**Methods**:

LostAndFound()

addLostItem()

findLostItem()

addMultipleLostItems()

isLostItemExists()

getTotalLostItemsCount()

getLostItems()

searchLostItemsByCategory()

removeFoundItem()

getLostItemsByCategory()

**Security Management**

**Data Structure**:

**Security Management :** PriorityQueue

**Attributes:**

securityQueue: PriorityQueue

completedTasks: List

**Methods:**

SecurityManagement()

addSecurityTask()

getNextSecurityTask()

processSecurityTasks()

addSecurityTasks()

getSecurityTasksCount()

clearSecurityTasks()

isHighPriorityTaskAvailable()

isLowPriorityTaskAvailable()

getCompletedTasks()

markTaskAsCompleted()

getAllTasksInQueue()

isTaskInQueue()

getSecurityTasks()

sortSecurityTasks()

**Staff Management**

**Data Structure**:

Top of Form **StaffManagement :** HashMap

**Attributes:**

staffMap: HashMap

**Methods:**

StaffManagement()

addStaff()

removeStaff()

updateStaffDetails()

getStaffDetails()

isStaffExist()

getTotalStaffCount()

clearAllStaff()

isStaffListEmpty()

updateStaffRole()

searchStaffByName()

getAllStaffDetailsAsString()

**BHP (Bag Handling System)**

**Data Structure**:

**BHP :** Queue implemented using a LinkedList

**Attributes:**

GUI Components

**Methods:**

BHP()

AddButtonListener()

RemoveButtonListener()

UpdateButtonListener()

SearchButtonListener()

ShowAllButtonListener()

isNumeric()

clearFields()

Data Structure:

bagInfoQueue: Queue (LinkedList)

**Cargo Class**

**Data Structure**:

**BHP :** Queue implemented using a LinkedList

**Attributes:**

name (String)

weight (double)

**Constructor:**

Cargo(String name, double weight)

**Methods:**

getName()

getWeight()

toString()

**MangeCargo Class**

**Data Structure**:

**MangeCargo :** List

**Attributes:**

cargoList (List<Cargo>)

**Constructor:**

MangoCaro()

Methods:

addCargo()

removeCargo()

getTotalWeight()

getCargoCount()

getCargoList()

**ManageFoodAndRetailServices**

**Data Structure**:

**ManageFoodAndRetailServices :** List

**Attributes:**

foodItems (List<FoodItem>)

retailItems (List<RetailItem>)

foodTextArea (Component)

retailTextArea (Component)

**Constructors:**

ManageFoodAndRetailServices()

**Methods:**

updateFoodTextArea()

updateRetailTextArea()

**Action Listeners:**

addFoodButton.addActionListener()

addRetailButton.addActionListener()