AIRPORT MANAGEMENT SYSTEM
DAMG6210-PROJECT GROUP2

DATABASE DESIGN DOCUMENT

Step 1: Introduction

The **Airport Management System database** is designed and developed to support and manage the critical operations that are involved in the airport activities. Our system addresses the need for effective flight scheduling, resource allocation and passenger management. The basic objective of our database is to ensure that all key areas of the airport's daily operations are properly documented, tracked, and managed.

Our system will handle flight details, bookings, boarding processes, ground support, and resource allocation such as runway and gate assignments. Above all, our system will also manage airport personnel, including airplane crew, ground staff, and security staff assignments, making sure that each part of the airport's activities/operations is/are effectively tracked and coordinated.

Our database will allow to perform the below operation(s):

- **Resource Management:** Managing parking, runways, gates, and ground support equipment.
- **Flight Management:** Managing flights, including runway usage and gate assignments.
- **Passenger Management:** Handling bookings, boarding passes, luggage check-ins and loyalty programs for passengers.
- **Employee Management:** Managing and tracking airport personnel, including ground staff, security staff, and airplane crew.

The whole aim of our system is to create an efficient, centralized platform for all airport management activities, allowing for smooth day-to-day operations, quick decision-making, and effective handling of any emergencies or disruptions.

Step 2: Business Problems Addressed

Our **Airport Management System** addresses various key business problems within the airport's daily operational environment. These problems varies from flight management to passenger services, and resource optimization to security deployment. Each section of the system is customised to solve specific issues that are common in day-to-day airport activities.

1. **Flight Management:**
   - **Problem:** Co-Ordinating flight departures and arrivals, runway usage and gate assignments.
   - **Solution:** Our System supports this by organising flight schedules, tracking them and assigns the runway/gates to ensure the smooth operations.
   - **Entities Involved : FlightDetails, GateInfo, RunwayManagement, Aircraft.**
2. **Passenger Management:**
   - **Problem:** Managing Loyalty Programs, flight bookings, generating/issuing boarding passes and tracking luggage.
   - **Solution:** Our system will effectively manage flight bookings, automates check-ins processes and provides real-time updates of the flight status to the passengers to keep them updated. It also assists with tracking luggage and ensures smooth experience for the passengers
   - **Entities Involved: PassengerInfo, FlightBooking, BoardingPass, Luggage, LoyaltyProgram**
3. **Employee Management:**
   - **Problem:** Assigning and tracking staff assignment of different roles, which includes airplane crew, ground staff, security staff.
   - **Solution:** Our system makes sure that the corresponding personnel are assigned to their dedicated areas such as airplane crew to airplane, ground staff to manage ground vehicle activities, security assignment.
   - **Entity(s) Involved: Employee (AirplaneStaff, GroundStaff, SecurityStaff –** This further subtypes you will see in later part of our project implementation)**.**

4. **Resource Management:**
    - **Problem:** Effectively managing runways, gate assignments and ground equipment, parking slot assignments.
    - **Solution:** Our system tracks and allocates these resources based on flight schedules and passenger needs, ensuring optimal utilization of the airport facilities and equipment.
    - **Entities Involved: GateInfo, RunwayManagement, ParkingInfo, GroundSupportInfo.**
5. **Maintenance Management:**
    - **Problem:** Managing the aircraft and ground equipment support maintenance, ensuring safety and compliance with regulations.
    - **Solution:** Our system tracks all maintenance schedules, types of maintenance that is being carried out whether it is for aircraft or ground equipment. This makes sures that the equipment which are in use are operational and safe to use.
    - **Entities Involved: Aircraft, MaintenanceInfo, GroundSupportInfo.**

The corresponding usage and links of the entities which are listed in the business problems sections are described in the entities and descriptions step.

Step3: Entities and Descriptions

1. **Airport (Further classified to Domestic & International)**
    - **Attributes:** AirportID, AirportName, Type, City, Area
    - **Description:** Represents the details of an airport, such as its name, location, and type (domestic or international). The airport serves as a hub for various operations like managing flights, resources, and employees.

2. **FlightDetails (Further classified into Connecting Flight)**
    - **Attributes:** FlightID, FlightNumber, FlightName, Capacity, DepartureTime, ArrivalTime, FlightSource, FlightDestination, FlightStatus, ActualArrivalTime, ActualDepartureTime, DelayReason
    - **Description:** Manages all information related to each flight, such as scheduling, capacity, status updates, and delays. This entity is central to flight management.

3. **FlightBooking**
    - **Attributes:** TicketID, TicketNumber, PurchaseDate, FareClass, BookingStatus, TicketPrice
    - **Description:** Tracks ticket purchases and bookings made by passengers for flights. This entity links passengers to the flights they are booked on.

4. **PassengerInfo**
    - **Attributes:** PassengerID, Name, Age, Nationality, CheckinStatus, CustomerFeedback
    - **Description:** Stores personal and travel information about passengers, including their check-in status and any feedback they provide about their experience.

5. **BoardingPass**
    - **Attributes:** BoardingPassID, SeatNumber, BoardingGroup, IssueTime
    - **Description:** Tracks details related to passengers' boarding passes, including seat assignments, boarding groups, and the time the boarding pass was issued.

6. **Luggage**
    - **Attributes:** LuggageID, NumberOfBags, Weight, Status, BaggageLocation
    - **Description:** Manages details about the luggage checked in by passengers, including the number of bags, their weight, and current location within the airport.

7. **GateInfo**
    - **Attributes:** GateID, GateNumber, Terminal, GateStatus
    - **Description:** Manages the assignment of gates to flights and their status, helping ensure flights are efficiently directed to the appropriate gates.

8. **RunwayManagement**
   - **Attributes:** RunwayID, RunwayNumber, OccupiedStatus
   - **Description:** Tracks the usage of runways, managing availability for take-off and landing to ensure smooth flight operations.
9. **ParkingInfo**
   - **Attributes:** ParkingID, SlotNumber, AvailabilityStatus, ReservedStatus, ParkingFee
   - **Description:** Tracks the parking slots available at the airport for passengers, allowing for reservations and managing real-time availability.
10. **Aircraft**
    - **Attributes:** AircraftID, AircraftType, Capacity, MaintenanceStatus, FuelConsumption, AircraftAge
    - **Description:** Manages information related to the aircraft used by airlines, including their type, capacity, and maintenance records.
11. **Airline**
    - **Attributes:** AirlineID, AirlineName, Country
    - **Description:** Represents the airline companies operating flights out of the airport. Each airline is responsible for managing their own fleet and flights.

12. **LoyaltyProgram**
    - **Attributes:** LoyaltyID, LoyaltyPoints, LoyaltyStatus, EnrollmentDate
    - **Description:** Tracks passengers' loyalty program details, including the number of points accumulated and their status within the program.

13. **Employee (Further classified into AirplaneStaff, Security Staff, Ground Staff)**
    - **Attributes:** EmployeeID, Name, Salary, Role, Department
    - **Description:** Represents all employees working in the airport, from airplane crew to ground support and security staff. Each employee is assigned a role and department.

14. **GroundSupportInfo**
    - **Attributes:** EquipmentID, EquipmentType, GSM_Status, UsageDuration, MaintenanceFrequency
    - **Description:** Tracks the operational status and usage of ground support equipment, such as baggage carts and fuel trucks, ensuring that airport operations are fully supported.

15. **MaintenanceInfo**
    - **Attributes:** MaintenanceID, MaintenanceType, MaintenanceDate, MaintenanceDuration, Maintenance Status, NextScheduledMaintenance.
    - **Description:** Tracks maintenance events for aircraft and ground support equipment, ensuring that all maintenance activities are scheduled, completed, and monitored for compliance. This entity will play an key role in the operational readiness of all the airport resources.

We have mentioned a point in our scope document where we described that we will be displaying the flight status continuously for the convenience of the passengers. To achieve that **rather taking one more Entity called "NoticeBoard"** we prefer to **showcase it as a "VIEW" a concept** in database as we feel there are no unique attributes, we can list to this entity apart from "**NoticeID**" and remaining will be duplicated or fetched from **FlightDetails** entity.

In addition to this we have introduced attributes like **DelayReason, Actual Departure/Arrival Time, LoyaltyID, CustomerFeedback, FareClass, BookingStatus, TicketPrice, Usage Duration, MaintenanceFrequency, ParkingFee, FuelConsumption, AircraftAge** to generate analytics/reports like **Monthly Flight Punctuality Report, Passenger Report, Staff Utilization Report, Gate Utilization Report, Revenue by route, Maintenance Schedule Report, Passenger Feedback and Satisfaction Trends etc. to improve the airport activities**.

Step4: Relationships between Entities

1. **Airport to FlightDetails**:
   - **Relationship**: One-to-many.
   - **Explanation**: An airport can host many flights, but each flight departs from or arrives at one specific airport.

2. **Airport to GateInfo:**
   - **Relationship:** One-to-many.
   - **Explanation:** An airport manages multiple gates, but each gate is associated with one airport.

3. **Airport to RunwayManagement:**
   - **Relationship**: One-to-many.
   - **Explanation**: An airport has multiple runways, but each runway is assigned to one airport.

4. **Airport to ParkingInfo:**
   - **Relationship**: One-to-many.
   - **Explanation**: An airport offers many parking slots, but each parking slot belongs to one airport.

5. **FlightDetails to FlightBooking:**
   - **Relationship**: One-to-many.
   - **Explanation**: A flight can have many bookings, but each booking is for one specific flight.

6. **FlightDetails to GateInfo:**
   - **Relationship**: Many-to-one.
   - **Explanation**: Multiple flights can use the same gate, but each flight has one assigned gate.

7. **FlightDetails to RunwayManagement:**
   - **Relationship**: Many-to-one.
   - **Explanation**: Multiple flights can use the same runway for take-off or landing, but each flight is assigned to one runway.

8. **FlightDetails to Aircraft:**
   - **Relationship**: Many-to-one.
   - **Explanation**: Multiple flights can use the same aircraft, but each flight is assigned to one aircraft.

9. **FlightDetails to ConnectingFlight(Future Enhancement):**
   - **Relationship**: One-to-one (subtype).
   - **Explanation**: A flight can be either a direct flight or a connecting flight. The ConnectingFlight entity is a subtype of FlightDetails.

10. **FlightBooking to PassengerInfo:**
    - **Relationship:** Many-to-one.
    - **Explanation:** A passenger can make multiple flight bookings, but each booking is linked to one passenger.

11. **FlightBooking to BoardingPass:**
    - **Relationship:** One-to-one.
    - **Explanation:** Each flight booking generates one boarding pass, and each boarding pass is tied to a flight booking.

12. **BoardingPass to GateInfo:**
    - **Relationship:** Many-to-one.
    - **Explanation:** Multiple boarding passes can reference the same gate, but each boarding pass refers to one gate.

13. **PassengerInfo to Luggage:**
    - **Relationship:** One-to-many.
    - **Explanation:** A passenger can check in multiple pieces of luggage, but each piece of luggage belongs to one passenger.

14. **Luggage to FlightBooking:**
    - **Relationship:** Many-to-one.
    - **Explanation:** Multiple pieces of luggage can be checked in for the same booking, but each luggage item is tied to one booking.

15. **PassengerInfo to LoyaltyProgram:**

- **Relationship:** One-to-one.
- **Explanation:** A passenger can enroll in one loyalty program, and each loyalty program account is linked to one passenger.

16. **Employee to Airport:**
    - **Relationship:** Many-to-one.
    - **Explanation:** Multiple employees can work at the same airport, but each employee is assigned to one airport.

17. **Employee to GroundSupportInfo:**
    - **Relationship:** One-to-many.
    - **Explanation:** Ground support staff members are responsible for operating or maintaining ground support equipment.

18. **GroundSupportInfo to Airport:**
    - **Relationship:** Many-to-one.
    - **Explanation:** Multiple ground support equipment items are assigned to one airport.

19. **Aircraft to MaintenanceInfo**
    - **Relationship:** One-to-many.
    - **Explanation:** An aircraft can have multiple maintenance events, but each maintenance event is related to one specific aircraft.

20. **GroundSupportInfo to MaintenanceInfo**
    - **Relationship:** One-to-many.
    - **Explanation:** Ground support equipment can have multiple maintenance events, but each maintenance event is tied to one piece of equipment.

Going forward we can see few more subtypes like Domestic, International Airport(s), AirplaneCrew, GroundStaff, SecurityStaff to make to it more clear and operations specific.

Step5: Key Database Decision(s):

1. **Maintaining Flight and Ticket Price Separately**
    - **Design Decision:** The price of the flight (in FlightDetails) is tracked separately from the ticket price (in FlightBooking).
    - **Reason:** This approach allows for dynamic pricing models, where ticket prices vary based on fare class, time of booking, and other factors. Keeping these prices separate enables more accurate financial reporting and data analysis.

2. **LoyaltyProgram as a Separate Entity**
    - **Design Decision:** We opted to keep LoyaltyProgram as a separate entity linked to PassengerInfo, rather than merging the data.
    - **Reason:** This decision allows passengers to enroll in a loyalty program independently from their booking details, which keeps the system flexible. It also helps avoid storing unnecessary loyalty data for passengers not enrolled in the program.

3. **Tracking Operational Status with GSM_Status and MaintenanceStatus**
    - **Design Decision:** We kept GSM_Status in GroundSupportInfo to reflect the real-time operational state of the equipment and MaintenanceStatus in MaintenanceInfo to track the steps of the maintenance process.
    - **Reason:** This separation ensures clarity in tracking both the current availability of equipment and its maintenance progress, making it easier to manage airport resources.

4. **Detailed Attributes for FlightDetails for Tracking Delays**
    - **Design Decision:** FlightDetails includes specific attributes such as ActualArrivalTime, ActualDepartureTime, and DelayReason to accurately track any delays in flight schedules.
    - **Reason:** These attributes enable the creation of reports that monitor flight punctuality, analyse common delay reasons, and help airlines and airports optimize flight schedules to minimize delays.

5. **Detailed Maintenance Tracking for Both Aircraft and Ground Equipment**
    - **Design Decision:** The MaintenanceInfo entity tracks Maintenance Date, Maintenance Type, Maintenance Duration, and NextScheduledMaintenance for both aircraft and ground support equipment.
    - **Reason:** This allows for generating maintenance reports, identifying patterns in equipment downtime, and scheduling preventative maintenance to avoid delays. Such reports are essential for safety compliance and operational efficiency.

6. **Parking Management through ParkingInfo**
    - **Design Decision:** ParkingInfo includes AvailabilityStatus, ReservedStatus, and ParkingFee, providing detailed data about parking operations at the airport.
    - **Reason:** These attributes enable reports on parking slot availability, usage patterns, and revenue from parking fees, helping to optimize parking space utilization and predict future demand.

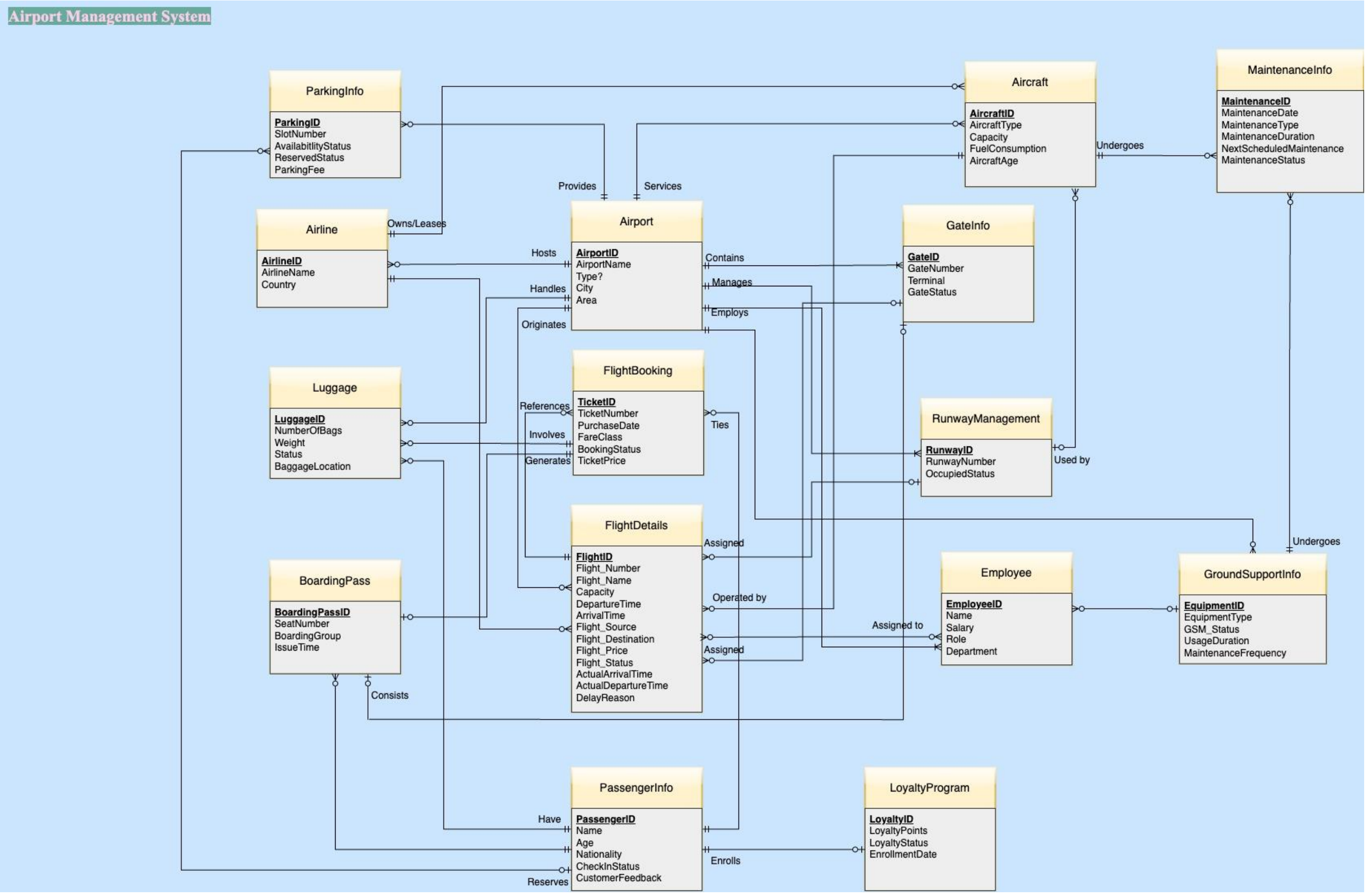
7. **Tracking Loyalty Program Points and Status**
    - **Design Decision:** The LoyaltyProgram entity includes LoyaltyPoints and LoyaltyStatus, which can track passenger participation in loyalty programs.
    - **Reason:** These attributes enable the generation of reports on passenger loyalty, point accumulation, and redemption patterns. These reports can help the airport and airlines develop strategies to increase customer retention and reward frequent travellers.

8. **Baggage Tracking through the Luggage Entity**
    - **Design Decision:** Luggage includes attributes such as NumberOfBags, Weight, and BaggageLocation to track luggage throughout the airport.
    - **Reason:** These attributes allow the generation of reports that analyse luggage handling efficiency, lost baggage incidents, and baggage load distribution across flights. This can improve the luggage handling process and enhance the passenger experience.

9. **Inclusion of ConnectingFlight as a Subtype of FlightDetails (During implementation)**
    - **Design Decision:** We chose to implement ConnectingFlight as a subtype of FlightDetails to distinguish between direct flights and flights with layovers. This avoids redundancy and makes the database easier to manage by only extending the FlightDetails entity when necessary.
    - **Reason:** This allows the generation of reports that differentiate between direct flights and connecting flights, which can be useful for analysing passenger preferences, flight patterns, and airport congestion related to layovers.

10. **Employee Subtypes: AirplaneStaff, GroundStaff, and SecurityStaff (During implementation)**
    - **Design Decision:** Instead of maintaining a large Employee table with roles, we opted to create subtypes: AirplaneStaff, GroundStaff, and SecurityStaff.
    - **Reason:** This decision allows us to track employees in more detail, ensuring that the system is flexible enough to handle role-specific attributes (e.g., FlightID for airplane crew and ShiftInfo for ground and security staff). This prevents the Employee table from becoming overloaded with irrelevant information for certain roles.

Conceptual ERD


Airport Management System
ParkingInfo
ParkingID
SlotNumber
AvailabitlityStatus
ReservedStatus
ParkingFee
Aircraft
AircraftID
AircraftType
Capacity
FuelConsumption
AircraftAge
MaintenanceInfo
MaintenanceID
MaintenanceDate
MaintenanceType
MaintenanceDuration
NextScheduledMaintenance
MaintenanceStatus
Undergoes
Provides
Services
Airline
AirlineID
AirlineName
Country
Owns/Leases
Airport
AirportID
AirportName
Type?
City
Area
Hosts
Handles
Originates
Contains
Manages
Employs
GateInfo
GateID
GateNumber
Terminal
GateStatus
Luggage
LuggageID
NumberOfBags
Weight
Status
BaggageLocation
FlightBooking
TicketID
TicketNumber
PurchaseDate
FareClass
BookingStatus
TicketPrice
References
Involves
Generates
Ties
RunwayManagement
RunwayID
RunwayNumber
OccupiedStatus
Used by
FlightDetails
FlightID
Flight_Number
Flight_Name
Capacity
DepartureTime
ArrivalTime
Flight_Source
Flight_Destination
Flight_Price
Flight_Status
ActualArrivalTime
ActualDepartureTime
DelayReason
Assigned
Operated by
Assigned to
Assigned
Undergoes
BoardingPass
BoardingPassID
SeatNumber
BoardingGroup
IssueTime
Consists
Employee
EmployeeID
Name
Salary
Role
Department
GroundSupportInfo
EquipmentID
EquipmentType
GSM_Status
UsageDuration
MaintenanceFrequency
PassengerInfo
PassengerID
Name
Age
Nationality
CheckInStatus
CustomerFeedback
Have
Reserves
Enrolls
LoyaltyProgram
LoyaltyID
LoyaltyPoints
LoyaltyStatus
EnrollmentDate