

# **Flight-Company-Database-Description**

**To run the project, you need to run the Main.java file**

## **Case Story**

An airline company includes flights and employees. Each airline company has a sequential company number, the company name, the year of establishment, a flight array, an employee array, and a sales system. In every flight, there is a flight date, a flight number, an array of employees on the plane, an array of passengers who bought tickets, the destination and departure of the flight, the maximum number of suitcases (baggage). Every flight will operate if there is at least 75% occupancy and at least 10 employees registered. Ticket sales (contact department) are identified in the system by flight and passenger and checks if there is a place on the flight. For every passenger and employee, there is a full name, ID, phone number, gender, and home address. For every passenger, in addition to human characteristics, there is also whether disabled (yes/no), number of suitcases, and a message box (vector). The airline company has the option to contact a passenger (contact department) through the identification of the flight number and ID. For every employee, in addition to human characteristics, there is also the security feature in the company. Employee registration for the flight (contact department), identified in the system by flight and employee, the system will register the employee only if the same employee exists in the general employee array of the airline company.

## **Process 1: ticket sale and passenger registration for the flight**

The airline company's system will receive a passenger and a flight that are entered into the system. The system will activate the buyTheTicket operation from the saleTicket class, and afterwards from the Flight class, the RegisterPassanger operation will be activated which checks if there is a place on the specific flight, if there is enough space for his baggage and only then will it return true or false accordingly. A message will be printed on the screen indicating whether the registration for the flight and the purchase of the ticket was successful or not. If the flight date was purchased, it will be updated to the current date

## **Process 2: Daily flight check**

Every day, the airline company's system will run on all existing flights in the system. Flights that are scheduled for the coming week from the date of the check and do not meet the occupancy condition of 75% of the total seats on the plane will be deleted from the system, a message will be printed on the screen and an update message will be sent to all the passengers registered for that flight.

## **An idea for further development of the system**

It is possible to create a class that contains a number of subsidiary companies of the main airline company, these subsidiary companies will be kept in a vector of airline companies and all orders will be performed through that object. For example, it will be possible to compare different airline companies, profits, number of flights that were carried out, collecting data for a large database. Currently, the system is intended for internal use of the company's employees, and in the future it will be possible to add methods in the passenger class, so that the user can enter the airline company's website and register himself for a flight and update details about different flights, this action will be done by using the toString method of the FlightCompany class. Also, it will be possible to export flight details to national registration systems for entry and exit from the country.

# UML - CLASS DIAGRAM

