GAZİ UNIVERSITY FACULTY OF ENGINEERING COMPUTER ENGINEERING

CENG 213 - OBJECT ORIENTED PROGRAMMING

SEMESTER PROJECT

181180053
Mihriban MISTAÇOĞLU

191180039
Elif ERDEN

AIRLINE MANAGEMENT SYSTEM

For our Object Oriented Programming class project, we are planning to create an Airline Management System using Core Java. In this project, the classes that we are going to are: Role, User, Employee, Passenger, Permission, Reservation, Airlines and Flight. Each class has their respective variables, methods, getter and setter methods and constructors.

Role class is the parent class of the User class and includes the role title, role id and role description of the users that will use the system and the functions for editing, deleting, searching and assigning roles. The class includes roleId, roleTitle, roleDescription variables and editRole(), deleteRole(), searchRole(), assignRole() functions.

User class indicates the user that is currently using the system. In this class, we have the user's id, role, name, e-mail, date of birth and address information and functions for deleting, editing and searching the user. The User class has an aggregation relationship with the Permission class, which indicates that a user can have permissions. This class includes userId, userName, userEMail, userDOB, userAdress, ***permission*** variables and editUser(), deleteUser(), searchUser() functions. User class has Employee class and Passenger class as its child classes. ***Permission*** class has an aggregation relationship with the User class, which means “a user can have permission”.

Employee class indicates the employees of the airline center. It holds an employee’s id, name, mail, phone number, password, address information and functions for editing, deleting and searching for an employee. It includes employeeId, employeeName, enployeeEMail, employeeMobile, employeeAdress variables and editEmployee(), deleteEmployee(), searchEmployee() funcions.

Passenger class holds the information about the passengers. It includes id, name, phone number, e-mail and address information of the passenger and functions for editing, deleting and searching passengers. There are passengerId, passengerName, passengerMobile, passengerEMail, passengerAdress variables and editPassenger(), deletePassenger() and searchPassenger() functions in this class.

Permission class is to manage the actions of the users. Permissions differ depending on whether the user is an employee or a passenger. This class has a permission’s id, title, and description information and functions for editing, deleting and searching for a permission. It includes permissionId, permissionTitle, permissionDescription, ***airlines, reservation, flight*** variables and editPermission(), deletePermission(), searchPermission() functions. Airlines, Reservation and Flight classes has an aggregation relationship with the Permission class. By doing so, we wanted to make Employee and Passenger classes be able to do operations with/on ***Airlines***, ***Reservation*** and ***Flight*** classes.

Reservation class is for creating a reservation and editing, searching or deleting it. This class has a reservation’s id, description, type and date information and functions for editing, searching and deleting a reservation. It includes reservationId, reservationType, reservationDate variables and editReservation(), deleteReservation(), searchReservation() functions.

Airlines class holds the information of an airline company. These information are id, description, type and schedule date and functions for editing, searching and deleting an airline company. There are airlinesId, airlinesDescription, airlinesType, airlinesScheduleDate variables and editAirLines(), deleteAirLines(), searchAirLines() functions.

Flight class is for the flights of an airline company. It includes id, type, title, description and date information about the flight and functions for editing, searching and deleting a flight. There are flightId, flightType, flightTitle, flightDescription, departDate variables and editFlight(), deleteFlight(), searchFlight() functions in this class. flightType variable shows whether a flight is a connected or direct flight, flightTitle is the code of the flight and flightDescription is for the Aircraft name.