Class Name: Customer

Super-classes: People

Attributes:

* Agency. The agency the customer may have used
* PassportHolder. Boolean, true if they have a passport false if not
* CreditCard. Credit Card details including, name, expiry date, number, cvc

Methods:

Class Name: Staff

Super-classes: People

Attributes:

Methods:

Class Name: NonMember

Super-classes: People

Attributes:

Class Name: Administrator

Super-classes: People

Attributes:

Methods:

Class Name: Manager

Super-classes: People

Sub-classes: Flight Manager, Service Manager

Attributes:

Methods:

Class Name: People

Super-classes: User

Attributes:

* Gender. Male, Female or Other
* ID. 8-digit unique number
* Address. Home address
* DOB. Date of birth, DD-MM-YYYY

Methods:

Class Name: User

Super-classes: None

Attributes:

* Name. String containing users first and last name
* Password. Hash value of user password
* Email. String containing alphanumeric values
* Phone. 8 or 10 digit number
* Authority. Singular number denoting authority level, 1-5 for customer - admin
* ReportFunctionClass. See below
* ProfileFunctionClass. See below
* ReserveFunctionClass. See below
* ServiceFunctionClass. See below

Methods:

Class Name: ReportFunction

Super-classes: None

Attributes:

Methods:

* Customer\_report()
  + Allows us to get the customer report
* Flight\_report()
  + Allows us to get the flight report
* Service\_report()
  + Allows us to get the service report
* Agency\_report()
  + Allows us to get the agency report
* Company\_report()
  + Allows us to get the company report

Class Name: Report

Super-classes: None

Sub-classes: customer\_report, flight\_report(), service\_report(), agency\_report(), company\_report()

Attributes:

Methods:

Class Name: customer\_report

Super-classes: Report

Attributes:

* totalCost. The total amount spent from customers
* costInEachMonth. The total amount spent from customers per month
* costInService. The amount spent on services

Methods:

Class Name: flight\_report

Super-classes: Report

Attributes:

* totalCost. A singular floating point number, the total spendatures for all routes
* costOfEachRoute. A list of floating point numbers, the spendatures of each individual route
* costOfEachClass. A list of floating point numbers, the spendatures of each class of airplane

Methods:

Class Name: customer\_report

Super-classes: Report

Attributes:

* totalCost. A floating point number containing the total cost of services
* costInEachMonth. A list of floating point numbers containing the total costs per month of in-flight services
* costInService. A list of floating point numbers

Methods:

Class Name: agency\_report

Super-classes: Report

Attributes:

* totalCost. A floating point number containing the total cost from agencies
* costInEachMonth. A list of floating point numbers containing per month costings
* customerNum. A integer denoting the amount of customers coming through agencies

Methods:

Class Name: company\_report

Super-classes: Report

Attributes:

* totalEarned. Floating point number for the total profit of the company
* totalSchedule. Floating point number for the total something to do with schedule of the company
* earnedInEachRoute. Floating point number for the total earned from each route

Methods:

Class Name: Profile\_Function

Super-classes: None

Attributes:

Methods:

* customer\_info()
* flight\_info()
* flight\_details()
* agency\_info()
* service\_info()
* company\_info()
* create\_flight()
* edit\_flight()
* delete\_flight()
* add\_airport()
* edit\_airport()
* delete\_airport()
* add\_airplane()
* edit\_airplane()
* delete\_airplane()
* enable\_account()
* disable\_account()
* promote()

Class Name: Reserve\_Function

Super-classes: None

Attributes:

Methods:

* register()
* ticket\_reservation()
* service\_reservation()
* change\_seat()
* change\_service\_reservation()
* switch\_seat()
* change\_flight()

Class Name: Service\_Function

Super-classes: None

Attributes:

Methods:

* add\_service()
* change\_service()
* delete\_service()

Class Name: Airport

Super-classes: none

Attributes:

* name. String containing name of airport
* city. String, City the airport is located in
* country. String, Country airport is located in
* IATA/FAA. 3-letter airport code e.g. (LAX or SYD)
* Latitude. 3 integers, respectively, degrees, minutes, seconds, and 1 letter denoting E or W
* Longitude. 3 integers, respectively, degrees, minutes, seconds and 1 letter denoting N or S
* Altitude. An integer for the closest meter to the average altitude of the airport
* Timezone. Integer between -12 and +14 denoting timezone relative to GMT
* DST. Boolean for daylight savings
* databaseTimezone. The timezone for the database

Methods:

Class Name: Fleet

Super-classes: none

Attributes:

* aircraft. List of strings of each aircraft
* ID. List of integers of each aircraft
* inService. List of possible services
* firstClass. Total first class seats, integer
* businessClass. Total business class seats, integer
* premiumEconomyClass. Total premium economy class seats, integer
* economyClass. Total economy class seats, integer
* total. Total seats

Methods:

Class Name: Route

Super-classes: none

Attributes:

* ID. 8-digit number
* sourceAirport. IATA/FAA of departure city
* destinationAirport. IATA/FAA of destination city
* codeShare. The code of the route
* Stop. Any stop other IATA/FAA tags

Methods:

Class Name: Schedule

Super-classes: none

Attributes:

* ID. Id of route
* flightID. Flight ID
* plane. ID of plane
* route. ID of route
* departTime. 4 digit number representing 24 hour time of departure
* arrivalTime. Same as above but for arrivals

Methods:

Class Name: Ticket

Super-classes: none

Attributes:

* date. Date format DD-MM-YYYY
* seat. Alphanumeric, which seat is assigned
* price. Costs of ticket
* route. The destination and departure city IATA/FAA tags
* service. What services are to be received

Methods:

Class Name: ServiceItem

Super-classes: none

Attributes:

* price. A floating point number for cost of service
* name. String for name of service
* id. 8-digit integer
* availability. Boolean true for available false if not

Methods: