Project Design - ***Data Dictionary***

***Airport – page 2***

***Route – page 3***

***Aircraft – page 4***

***Schedule – page 5***

***Flight – page 6***

***Customer – page 7***

***TravelAgent, ServiceItem – page 10***

***FlightService – page 11***

***Booking – page 12***

***Staff – page 13***

***ProfileManager , FlightManager – page 14***

***BookingManager, ServiceManager – page 15***

***Databse – page 16***

***ProfileSystem – page 20***

***ReservationSystem – page 21***

***ReportSystem, ServiceSystem – page 22***

***FlightSystem – page 23***

**Class Name:** Airport

**Superclass:** None

**Attributes:**

**Id :** A number that Uniquely identifies an airport

**Name :** Alphanumeric

**City :** Alphanumeric

**Country :** Alphanumeric

**IATA/FAA :** Alphanumeric, an abbreviation of name.

**Latitude :** A Decimal number representing latitude

**Longitude:** A decimal number representing longitude.

**Altitude:** A number representing altitude in meters.

**Timezone:** A number representing timezone related to GMT time. Example this value is 12 for NZ. This represents GMT+12.

**DST:** A character that represents daylight saving time

**Tz:** Alphanumeric, represents timezone from the tz database.

**Methods:**

**getID():** This method allows access to an airports ID.

**getName():** This method allows access to an airports name.

**getCity():** This method allows access to an airports City.

**getIATA():** This method allow access to an airports IATA code.

**getLat():** This method allows access to an airports Latitude.

**getLong():** This method allows access to an airports Longitude.

**getAlt():** This method allows access to an airports Altitude.

**getTimeZone():** This Method allows access to an airports time zone(GMT).

**getDST():** This method allows access to an airports Daylight saving code.

**getTZ():** This method allows access to an airports TZ database time zone.

**setID(id):** This method allows us to set an airports ID.

**setName(name):** This method allows us to set an airports Name.

**setCity(city):** This method allows us to set an airports City.

**setIATA(IATA code):** This method allows us to set an airports IATA/FAA abbreviation code.

**setLat(latitude):** This method allows us to set an airports latitude.

**setLong(longitude):** This method allows us to set an airports Longitude.

**setAlt(altitude):** This method allows us to set an airports altitude.

**setTimeZone(amt\_hours):** This method allows us to set an airports time zone(GMT)

**setDST(DST\_code):** This method allows us to set an airports daylight saving code.

**setTZ(timezone\_name):**This method allows us to set an airports TZdatabase time zone.

**Class Name: Route**

Superclass: None

Attributes:

ID: A number that uniquely identifies a route.

SourceAirport: An IATA/FAA code that corresponds to an airport

DestinationAirport: An IATA/FAA code that corresponds to an airport

codeshare: A character that can be either ‘NULL’ or ’Y’. Defines whether codesharing occurs on this route.

stops: A number representing the number of stops on this route.

Methods:

getID(): This method allows access to a routes ID.

getSrc(): This method allows access to a routes source airport IATA/FAA code.

getDest(): This method allows access to a routes destination airport IATA/FAA code.

getCodeShare(): This method allows access to a routes codeshare status.

getStops(): This method allows access to the amount of stops on a route.

setID(id): This method allows us to set the ID of a route.

setSrc(IATA\_code): This method allows us to set the source airport IATA/FAA code.

setDest(IATA\_code): This method allows us to set the destination airport IATA/FAA code.

setCodeShare(codeshare\_status):This method allows us to set the codeshare status for a route.

setStops(): This method allows us to set the amount of stops on a route.

**Class Name: Aircraft**

Superclass: None

Attributes:

ID: A number used to uniquely identify an aircraft.

name: Alphanumeric. The name used for the type of aircraft.

inService: A number representing how many of this type of aircraft are in service.

fClass: A number representing how many first class seats are on this aircraft.

bClass: A number representing how many business class seats are on this aircraft.

peClass: A number representing how many premium economy seats are on this aircraft.

eClass: A number representing how many economy seats are on this aircraft.

total: A number representing the total number of seats available on the aircraft.

Methods:

getID(): This method allows access to an aircrafts ID.

getName(): This method allows access to an aircrafts type.

getInService(): This method allows access to the amount of aircrafts of this type that are in service

getFClass(): This methos allows access to the amount of first class seats on this aircraft.

getBClass(): This method allows access to the amount of business class seats on this aircraft.

getPEClass(): This method allows access to the amount of premium economy class seats on this aircraft.

getEClass(): This method allows access to the amount of economy class seats on this aircraft.

getTotal(): This allows access to the total amount of seats on this aircraft.

setID(id): This method allows us to set an aircrafts ID.

setName(name): This method allows us to set an aircrafts name.

setInService(numOfAircraft): This method allows us to set the amount of aircraft of this type that are in service.

setFClass(fName): This method allows us to set the amount of first class seats on this aircraft

setBClass(lName): This method allows us to set the amount of business class seats on this aircraft.

setPEClass(numOfseats): This method allows us to set the amount of premium economy class seats on this aircraft.

setEClass(numOfSeats): This method allows us to set the amount of economy class seats that are on this aircraft.

setTotal(): This method calculates and sets the total number of seats on this aircraft by adding together the values in fClass,bClass,peClass,and eClass.

**Class: Schedule**

Superclass: None

Attributes:

ID: A number used to uniquely identify a schedule in the database.

flightID: Alphanumeric. A string used to uniquely identify a Flight.

Plane: A number that identifies the type of plane to be used for this schedule. This number refers to a plane in the Aircraft database.

Route: A number that identifies the route to be used in this schedule. This number refers to a route in the route database.

departTime: A string detailing the date,day,time,year and timezone of the departure.

arriveTime: A string detailing the date,day,time,year and timezone of the arrival

Methods:

getID(): This method allows us access to a schedules database ID.

getFlightID(): This method allows us access to the Flight ID.

getPlane(): This method allows us to access the plane type used in a schedule.

getRoute(): This method allows us to access the route used in a schedule.

getDeparture(): This method allows us to access the departure date/time of the schedule.

getArrival(): This method allows us access the arrival date/time of the schedule.

setID(): This method allows us to set a schedules database ID.

setFlightID(): This method allows us to set a schedules flight ID.

setPlane(): This method allows us to set a schedules plane type.

setRoute(): This method allows us to set a schedules route.

setDeparture(): This method allows us to set the schedules departure date/time.

setArrival(): This method allows us to set the schedules arrival date/time

**Class: Flight**

Superclass: None

Attributes:

flightID: Alphanumeric. A string that uniquely identifies a flight. Refers to a schedules flightID.

currentFCap: A number representing how many people are currently booked as first class on a flight. Must be less that aircrafts fClass value.

currentBCap: A number representing how many people are currently booked as Business class on a flight. Must be less that aircrafts BClass value.

currentPECap: A number representing how many people are currently booked as premium economy class on a flight. Must be less that aircrafts PEClass value.

currentECap: A number representing how many people are currently booked as economy class on a flight. Must be less that aircrafts EClass value.

serviceOnFlight: A list of FlightService objects that are allocated to be on this flight.

Methods:

getFlightID(): This method allows us access to a schedules database ID.

getFCap(): This method allows us to access the current amount of first class customers booked on this flight.

getBCap(): This method allows us to access the current amount of business class customers booked on this flight.

getPECap(): This method allows us to access the current amount of premium economy customers booked on this flight.

getECap(): This method allows us to access the current amount of economy customers booked on this flight.

setFlightID(): This method allows us to set the FlightID of a flight.

setFCap(): This method allows us to set the current first class capacity of a flight.

setBCap(): This method allows us to set the current business class capacity of a flight

setPECap(): This method allows us to set the current premium economy capacity of a flight.

setECap(): This method allows us to set the current economy capacity of a flight.

**Class Name: Customer**

Superclass: None

Attributes:

ID: A number used to uniquely define a person

Title: Alphanumeric.

fName: Alphanumeric. Represents a persons first name.

lName: Alphanumeric. Represents a persons last name.

gender: A character that represents a persons gender. Can be either ‘m’ or ‘f’.

DOB: Instantiation of a ‘Date’ class. Represents a persons date of birth.

Phone: Alphanumeric. Represents a persons contact phone number.

Email: String of characters that represent a persons email address.

Address: Alphanumeric. Represents a persons street address.

State: Alphabetic. Represents which state a person resides in.

Country: Alphabetic. Represents which country a person resides.

CardType: Alphanumeric. Represents what type of card the customer has.

CardNum: Numeric. A Number that represents the customers card number.

freqFlierPoints: Numeric. Represents the number of frequent flier points a customer has accredited to them.

Passport: Either ‘TRUE’ or ‘FALSE’. Represents whether a customer has a passport.

Status: Either ‘NULL’, ‘no fly’, or ‘watch’. Represents whether a customer is allowed to book a flight based on previous flight behaviour.

TravelAgent: Alphanumeric. A String that corresponds to the name of a travel agent in the agents database.

Methods:

getID(): This method allows access to a customers ID.

getTitle(): This method allows access to a customers title.

get FName(): This method allows access to a customers first name.

getLName(): This method allows access to a customers last name.

getGender(): This method allows access to a customers gender.

getDOB(): This method allows access to a customers date of birth.

getPhone(): This method allows access to a customers phone number.

getEmail(): This method allows access to a customers email Address.

getAddress(): This method allows access to a customers street address.

getState(): This method allows access to a customers state of residence.

getCountry(): This method allows access to a customers country of residence.

getCardType(): This method allows access to a customers credit card type.

getCardNum(): This method allows access to a customers credit card number.

getfreqFly(): This method allows access to the amount of frequent flier points accredited to the customer.

getPassport(): This method allows access to a customers passport status.

getStatus(): This method allows access to a customers fly status.

getAgent(): This method allows access to the name of a travel agent that the customer used to book a flight.

setID(id): This method allows us to set the ID of a customer.

setTitle(title): This method allows us to set the title of a customer.

setFName(fName): This method allows us to set the first name of a customer.

setLname(lName): This method allows us to set the last name of a customer.

setGender(gender): This method allows us to set the gender of a customer.

setDOB(date): This method allow us to set the date of birth of a customer.

setPhone(ph#): This method allows us to set the phone number of a customer.

setEmail(emailAddress): This method allows us to set the email address of a customer.

setAddress(stAddress): This method allows us to set the street address of a customer.

setState(state): This method allows us to set the state of residence of the customer.

setCountry(country): This method allows us to set the country of residence of the customer.

setCardType(cardType): This method allows us to set the credit card type of the customer.

setCardNum(cardNum): This method allows us to set the credit card number of the customer.

setFreqFly(freqFlyPts): This method allows us to set the frequent flier points for a customer.

setPassport(passportState): This method allows us to set the passport status of the customer.

setStatus(status): This method allows us to set the no fly status of the customer.

setAgent(name): This method allows us to set the agency name that the customer used to book a flight.

**Class Name: Travel Agent**

Superclass: None

Attributes:

ID: A number used to uniquely identifies a Travel Agent

Name: Alphanumeric. Represents the name of the agency.

Phone: Numeric. Represents the travel agents contact phone number.

Email: Alphanumeric. Represents the travel agents contact email address.

Methods:

getID(): This method allows access to a travel agencies ID.

getName(): This method allows access to a travel agencies name.

getPhone(): This method allows access to a travel agencies phone number.

getEmail(): This method allows access to a travel agencies email address.

setID(id): This method allows us to set the ID of an agency.

setName(name): This method allows us to set the name of an agency.

setPhone(phone#): This method allows us to set the phone number of an agency.

setEmail(emailaddress): This method allows us to set the email address of an agency.`

createBooking(): This method allows a travel agent to create a booking for a customer.

**Class Name: ServiceItem**

Superclass: None

Attributes:

ID: A number used to uniquely identify a service item

Item: Alphabetic. Represents a description/name of a service item.

Cost: Numeric. Represents the price of an item.

Availability: Either ‘All’ or ‘International’. Represents what type of flights this item is available on.

Methods:

getID(): This method allows access to a Service Items ID.

getItem(): This method allows access to a service items description.

getCost(): This method allows access to a service items price.

getAvail(): This method allows access to a service items availability.

setID(id): This method allows us to set the ID of a service Item.

setItem(desc): This method allows us to set the description of a service item.

setCost(price): This method allows us to set the price of a service item.

setAvail(availability): This method allows us to set the availability of a service item.

**Class: FlightService**

Superclass: none.

Attributes:

ID: A number that uniquely identifies a FlightService in the databse.

FlightID: A string that identifies the flight this service belongs to. Refers to a Flight class’s FlightID.

ServiceItem: A number that identifies a ServiceItem. This refers to a ServiceItem class’s ID.

itemAmount: A number that identifies how many of the ServiceItem will be on the flight.

Method:

getID(): This method allows us to access a FlightServices ID.

getFlightID(): This method allows us access to a FlightService’s Flight ID.

getServiceItem(): This method allows us access to a FlightService’s service item.

getItemAmount(): This method allows us to access to the amount of serviceItem’s that will be on a flight.

setID(): This methos allows us to set the ID of a flight service object.

setFlightID(): This method allows us to set the flightID of the flight this FlightService belongs to.

setServiceItem(): This method allows us to set which item this flightService is for.

setItemAmt(): This method allows us to set the amount of ServiceItems that will be on the flight.

**Class: Booking**

Superclass: none

Attributes:

CustomerID: A number that uniquely identifies the customer that has booked a flight.

FlightID: A string that identifies the flight this Booking belongs to

seatNumber: A String that identifies which seat on the plane this booking is for.

seatClass: A string that identifies what class of seating this booking is for.

Services: A list of FlightService objects that represent what services the customer has requested for this flight.

Price: A number that represents the price of this booking.

Methods:

getCustomer(): This method allows us access to the Customers ID.

getFlight(): This method allows us access to the Flight’s ID.

getSeatNum(): This method allows us access to the seat number this booking is for.

getSeatClass(): This method allows us access to the seat class this booking is for.

getServices(): This method allows us access to the ServiceItems that the customer has requested for this flight.

getPrice(): This method allows us access to the price of this booking.

setCustomer(): This method allows us to set the customer ID for this booking,

setFlight(): This method allows us to set the FlightID for this booking.

setSeatNum(): This method allows us to set the seat number for this booking.

setSeatClass(): This method allows us to set the seat class for this booking.

setServices(): This method allows us to set the customers requested services for this booking.

setPrice(): This method allows us to set the price of this booking.

**Class Name: Staff**

Superclass: None

Attributes:

ID: A number used to uniquely identify a staff member.

Password: A sting of characters representing a staffs login password.

userType: A string that identifies the type of staff member the object represents. This can be either “Staff”,”ProfileManager”,”ServiceManager”,”FlightManager”, ”BookingManager” or “Admin”.

Methods:

getID(): This method allows us to access a staff members ID.

getPassword(): This method allows us to access a staff members password

setID(id): This method allows us to set the id of a staff member.

setPassword(password):This method allows us to set the password of a staff member.

findCustomer(): This method allows the user to retrieve an existing customers details.

createCustAccount(): This method allows the user to create an account for a new customer.

viewCustBookings(): This allows the user to view all the bookings made by an existing customer including service details.

modifyCustDetails(): This allows the user to modify an existing customers personal details.

modifyCustBooking(): This allows the user to modify an existing customers booking details.

**Class: ProfileManager**

Superclass: Staff

Attributes:

ID: A number used to uniquely identify a staff member.

Password: A sting of characters representing a staffs login password.

Methods:

getID(): This method allows us to access a staff members ID.

getPassword(): This method allows us to access a staff members password

setID(id): This method allows us to set the id of a staff member.

setPassword(password):This method allows us to set the password of a staff member.

findCustomer(): This method allows the user to retrieve an existing customers details.

createCustAccount(): This method allows the user to create an account for a new customer.

viewCustBookings(): This allows the user to view all the bookings made by an existing customer including service details.

modifyCustDetails(): This allows the user to modify an existing customers personal details.

modifyFreqFlier(customerID): This method allows the user to modify a customers frequent flier points.

**Class: Flight manager**

Superclass: Staff

Attributes:

ID: A number used to uniquely identify a staff member.

Password: A sting of characters representing a staffs login password.

Methods:

modifyAircrafts(): This method allows the user to add new/edit existing information about aircrafts in the database.

modifySchedules(): This method allows the user to add new/edit existing information about schedules in the database.

modifyAirports(): This method allows the user to add new/edit existing information about Airports in the database.

modifyRoutes(): This method allows the user to add new/edit existing information about Routes in the database.

viewFlightReports(): This method allows the user to see various reports about flights created by the system.

**Class: BookingManager**

Superclass: Staff

Attributes:

ID: A number used to uniquely identify a staff member.

Password: A sting of characters representing a staffs login password.

Methods:

modifyCustBooking(): This allows the user to modify an existing customers booking details.

modifySeating(): This allows the user to modify customer seating even if seats are occupied.

modifyNoFly(): This allows the user to modify the no-fly list in the database.

viewBookingReports(): This method allows the user to see various reports about Bookings created by the system.

**Class: ServiceManager**

Superclass: Staff

Attributes:

ID: A number used to uniquely identify a staff member.

Password: A sting of characters representing a staffs login password.

Methods:

modifyServiceItems(): This method allows the user to modify existing, or add new items to the ServiceItems database.

modifyFlightService(flightID): This method allows the user to modify the service details of a flight.

viewServiceReports(): This method allows the user to view various reports about services generated by the system.

**Class: Database**

Superclass: none

Attributes:

Employees: A list of staff objects.

Airports: A list of Airport objects.

Routes: A list of Route objects.

Aircrafts: A list of Aircraft objects.

Customers: A list of Customer objects.

Agents: A list of TravelAgent objects.

ServiceItems: A list of ServiceItem objects.

FlightServices: A list of FlightService objects.

Flights: A list of Flight objects.

Bookings: A list of Booking objects.

Method:

loadEmployees(): This method loads information from a file into staff objects which are places in the employees list.

loadAirports(): This method loads information from a file into Airport objects which are places in the airports list.

loadRoutes(): This method loads information from a file into Route objects which are places in the routes list.

loadAircrafts (): This method loads information from a file into Aircraft objects which are places in the aircrafts list.

loadCustomers(): This method loads information from a file into Customer objects which are places in the customers list.

loadAgents(): This method loads information from a file into TravelAgent objects which are places in the agents list.

loadServiceItems(): This method loads information from a file into ServiceItem objects which are places in the serviceItems list.

loadFlightServices(): This method loads information from a file(If it already exists) into FlightService objects which are placed in the flightServices list.

loadFlights(): This method loads information from a file into Flight objects which are placed in the flights list.

loadBookings(): This method loads information from a file into Booking objects which are placed in the Bookings list.

Load(): This method loads all data for the database. Utilised functions defined above.

searchStaff(username): This method allows us to search the employees list and access an employee based on their username.

searchAirports(ID): This method allows us to search the airports list and access an airport based on its ID.

searchRoutes(ID): This method allows us to search the routes list and access a route based on its ID.

searchRoutes(src,dest): This method allows us to search the routes list and access a route based on a source airport and a destination airport.

searchAircrafts(ID): This method allows us to search the aircrafts list and access an aircraft based on its ID.

searchCustomers(customer\_email): This method allows us to search the Customers list and access a customer based on their email address.

searchAgents(agent\_name): This method allows us to search the agentss list and access a TravelAgent based on their name.

searchServiceItems(ID): This method allows us to search the serviceItems list and access a ServiceItem based on its ID.

searchFlightServices(flightID): This method allows us to search the flightServices list and access a FlightService based on its flightID.

searchFlightsByID(flightID): This method allows us to search the flights list and access a flight based on its flightID.

searchFlightsByDate(route,date): This method allows us the search the flights list and access flights based on their date and route.

searchSchedule(flightID): This method allows us to search the schedules list and access a schedule based on its flightID.

searchBookings(customerID): This method allows us to search the bookings list and access a booking based on its customerID.

addEmployees(staff): This method allows us to add a staff member to the employees list.

addAirports(airport): This method allows us to add a airport to the airports list.

addRoutes(route): This method allows us to add a route to the routes list.

addAircrafts(aircraft): This method allows us to add a aircraft to the aircrafts list.

addCustomers(customer): This method allows us to add a customer to the customers list.

addAgents(TravelAgent): This method allows us to add a TravelAgent to the agents list

addServiceItems(ServiceItem): This method allows us to add a ServiceItem to the serviceItems list

addSchedule(Schedule): This method allows us to add a Schedule to the schedules list.

addFlightServices(FlightService): This method allows us to add a FlightService to the flightServices list.

addFlights(Flight): This method allows us to add a Flight to the flights list

addBooking(Booking): This method allows us to add a booking to the bookings list. returns booking number. All updates to the database also occur(Example add seating details etc).

removeEmployee(Employee): This method allows us to remove an employee from the employees list.

removeAirport(Airport): This method allows us to remove an Airport from the airports list.

removeRoute(Route): This method allows us to remove a route from the routes list.

removeAircraft(Aircraft): This method allows us to remove an aircraft from the aircrafts list.

removeCustomer(Customer): This method allows us to remove an customer from the customers list.

removeAgent(TravelAgent): This method allows us to remove a Travel agent from the agents list.

removeServiceItem(ServiceItem): This method allows us to remove a ServiceItem from the serviceItems list.

removeSchedule(Schedule): This method allows us to remove a schedule from the schedules list.

removeFlightService(FlightService): This method allows us to remove a FlightService from the flightServices list.

removeFlight(Flight): This method allows us to remove a Flight from the flights list.

removeBooking(Booking): This method allows us to remove an booking from the bookings list.

modifyEmployee(username): This method allows us to modify an employee in the employees list.

modifyAirport(ID): This method allows us to modify an airport in the airports list.

modifyRoute(ID): This method allows us to modify a route in the routes list.

modifyAircraft(ID): This method allows us to modify an aircraft in the aircrafts list.

modifyCustomer(customer\_email): This method allows us to modify a customer in the customers list.

modifyAgent(agent\_name): This method allows us to modify a travel agent in the agents list.

modifyServiceItem(ID): This method allows us to modify a serviceItem in the serviceItems list.

modifyFlightService(FlightService): This method allows us to modify a flightService in the flightServices list.

modifyFlight(flightID): This method allows us to modify a flight in the flights list.

modifySchedule(flightServiceID): This method allows us to modify an schedule in the schedules list.

modifyBooking(customerID): This method allows us to modify a booking in the bookings list.

saveEmployee(): This method saves the contents of the employees list to a file.

saveAirport(): This method saves the contents of the airports list to a file.

saveRoute(): This method saves the contents of the routes list to a file.

saveAircraft(): This method saves the contents of the aircrafts list to a file.

saveCustomer(): This method saves the contents of the customers list to a file.

saveAgent(): This method saves the contents of the agents list to a file.

saveServiceItem(): This method saves the contents of the serviceItems list to a file.

saveFlightService(): This method saves the contents of the flightServices list to a file.

saveSchedule(): This method saves the contents of the schedules list to a file.

saveFlight(): This method saves the contents of the flights list to a file.

saveBooking(): This method saves the contents of the bookings list to a file.

Save(): This method saves all data from the database. Utilises methods defined above.

**Class: ProfileSystem**

Superclass: none

Attributes:

Database: A reference to the database.

Methods:

findCustomer(email): This method allows the user to retrieve an existing customers details.

modifyNoFly(): This allows the user to modify the no-fly list in the database.

modifyCustomer(email): This method allows the user to modify any customer information.

findStaff(username): This method will find a staff member in the databse via their username

modifyStaff(username): This method will allow the user to modify any staff information.

findTravelAgent(agentName): This method will find a TravelAgent in the database via their agency name.

modifyTravelAgent(agentName): This method allows the user to modify any TravelAgent information.

customerLogin(email,password): This method allows a customer to log into the system.

staffLogin(username,password): This method allows a staff member to log into the system.

Validate(customer,pwd): This method validates the login details of a customer against a password. Method used in customerLogin() function.

Validate(staff,pwd): This method validates the login details of a staff object against a password. Method used in staffLogin() function.

createCustAccount(): This method allows the user to create an account for a new customer.

viewCustBookings(): This allows the user to view all the bookings made by an existing customer including service details.

modifyCustDetails(email): This allows the user to modify an existing customers personal details.

removeCustomer(email): This method allows the user to remove a customers details from the system.

removeStaff(username): This method allows the user to remove a staff member from the system.

**Class: ReservationSystem**

Superclass: none

Attributes:

Database: A reference to the database.

Methods:

createBooking(customerID): This method allows the user to create a new booking.

modifyCustBooking(): This allows the user to modify an existing customers booking details.

modifySeating(): This allows the user to modify customer seating even if seats are occupied.

viewBookingReports(): This method allows the user to see various reports about Bookings created by the system.

getFlightSeating(flightID): This method returns information about seating occupancy for a flight.

AddCustomerToFlight(booking,flightID): This method modifies a flight objects information when a customer has booked seats on a flight

addBooking(flightID): This method adds a new booking into the bookings list

getTicketPrice(flightID): This method returns the ticket price for a flight.

searchFlights(IATA,IATA,date): This method returns a list of flights.Obtained by searching the databasewith a route object(created with 2 IATA codes), and a date.

selectAirport(airport name): This method allows the user to select an airport during the booking process.

custCancelBooking(customerID,booking): This method allows a customers booking to be removed from the system.

agentBooking(agentID): This method allows a booking agent to book a flight for a customer.

displayAirports(): This method returns a list of airports sorted alphabetically and displays them to the user.

Confirm(booking): This method confirms booking details for a customers booking. addBooking(booking) is called to update the database.

getSeating(FlightID): This method retrieves seating details about a flight.

**Class: ReportSystem**

Superclass: none

Attributes:

Database: A reference to the database.

Methods:

generateReservationReport(): This method allows the user to see all reports about reservations from the system.

generateServiceReport(): This method allows the user to see all reports about services from the system.

generateFlightReport(): This method allows the user to see all reports about flights from the system.

generateProfileReport(): This method allows the user to see all reports about profiles from the system.

**Class: ServiceSystem**

Superclass: none

Attributes:

Database: A reference to the database.

Methods:

getFlightService(FlightID): This method returns a list of available service items on a particular flight.

modifyServiceItem(ID): This method allows the user to modify any information about a serviceItem.

modifyFlightService(ID): This method allows the user to modify any information about a FlightService.

addServiceItem(serviceItem): This method allows the user to add a serviceItem to the database.

removeServiceItem(ID): This method allows the user to remove a serviceItem from the database.

addFlightService(ID): This method allows the user to add a flightService to the database.

removeFlightService(ID): This method allows the user to remove a flightService from the database.

searchFlightServices(flightID): This method returns a list of flightService information about a flight.

**Class: FlightSystem**

Attributes:

Database: A reference to the database.

Methods:

modifyAircrafts(): This method allows the user to add new/edit existing information about aircrafts in the database.

modifySchedules(): This method allows the user to add new/edit existing information about schedules in the database.

modifyAirports(): This method allows the user to add new/edit existing information about Airports in the database.

modifyRoutes(): This method allows the user to add new/edit existing information about Routes in the database.

removeAircraft(aircraft): This method allows us to remove an aircraft from the database. This will also alert the user if the airport to be deleted is referenced elsewhere in the system.

removeRoute(route): This method allows us to remove a route from the database. This will also alert the user if the route or its attributes are referenced elsewhere in the system.

removeSchedule(schedule): This method allows us to remove a schedule from the database. This will also alert the user if the schedule or its attributes are referenced elsewhere in the system.

removeAirport(airport): This method allows us to remove a airport from the database. This will also alert the user if the airport is referenced elsewhere in the system.