Formal Requirements Specification
Airline Reservation System

Submitted in partial fulfillment of the requirements of the degree of Master of Software Engineering

Kaavya Kuppa
CIS 895 – MSE Project
Department of Computing and Information Sciences
Kansas State University

Committee Members:
1. Dr. Daniel Andresen
2. Dr. Torben Amtoft
3. Dr. Mitchell L. Neilsen
model AirlineReservationSystem

----- CLASSES

class User
attributes
  userid: string
  password: string
  emailid: string
  name: string
  loginstatus : Boolean
operations
  VerifyLogin(email : string , password : string) : Boolean = user.allInstances -> exists ( u: user | u.emailid = emailid and u.password = password)
end

class Customer < User
attributes
  userid : string
  password : string
  customername : string
  emailid : string
  address : string
  phonenumber : integer
  creditcardnumber : integer
  age : integer
operations
  register()
  login() : Boolean
  searchflights()
  bookflights()
searchpackages()
bookpackages()
searchhotels()
bookhotels()
browse()
end

class flight()
attributes
    flightnumber : integer
    noofseats : integer
    source : string
    destination : string
end

class hotel()
attributes
    hotelname : string
    hotelid : integer
    noofrooms: integer
end

class package()
attributes
    packageid : integer
end

class seat()
attributes
    seatnumber : integer
end
---- ASSOCIATIONS

-- This association is being written to indicate that there should be some number of seats on the plane greater than zero.

association planeseats between
Flight[1] role belongsto
Seat[10..300] role has
end

-- A customer can book any no.of seats on the flight, depending upon the capacity of the flight.

Association customerandseats between
Customer[1] role bookedby
Seat[1..*] role books
End

-- A customer should be seated in the flight in only one seat.

Association customerseated between
Customer[1] role holds
Seat[1] role isheld
End

---- CONSTRAINTS

-- This constraint is written to indicate that each customer who registers to the Airline Reservation system website should have a unique id.

Context user
inv uniqueid:
user.allInstances -> forAll( u1,u2 | u1 <> u2 implies u1.userid <> u2.userid)
end
--This constraint is written to indicate that each user will have a unique e-mail id.
Context user
inv uniqueemail:
user.allInstances -> forAll( us1,us2 | us1 <> us2 implies us1.emailid <> us2.emailid)
end

-- This constraint is written to indicate that each customer’s e-mail id, should be unique
Context customer
inv uniqueemail:
customer.allInstances -> forAll(c1,c2| c1<>c2 implies c1.emailid <> c2.emailid)
end

-- This constraint is written to indicate that the number of seats in the flight should be
greater than zero
Context flight
inv noofseats:
flight.allInstances -> forAll ( f1 | f1.nooseats >= 1)
end

--This constraint is written to ensure that no two flights have the same number.
Context flight
Inv uniquename:
flight.allInstances -> forAll ( f1,f2| f1<>f2 implies f1.flightnumber <> f2.flightnumber)
end

--This constraint is written to ensure that a hotel can have multiple rooms.
Context Hotel
Inv hotelrooms:
Hotel.allInstances -> forAll(h1| (h1.noofrooms ) ->size() >=1)
end