



COEN 6312

Model Driven Software Engineering

Deliverable-3

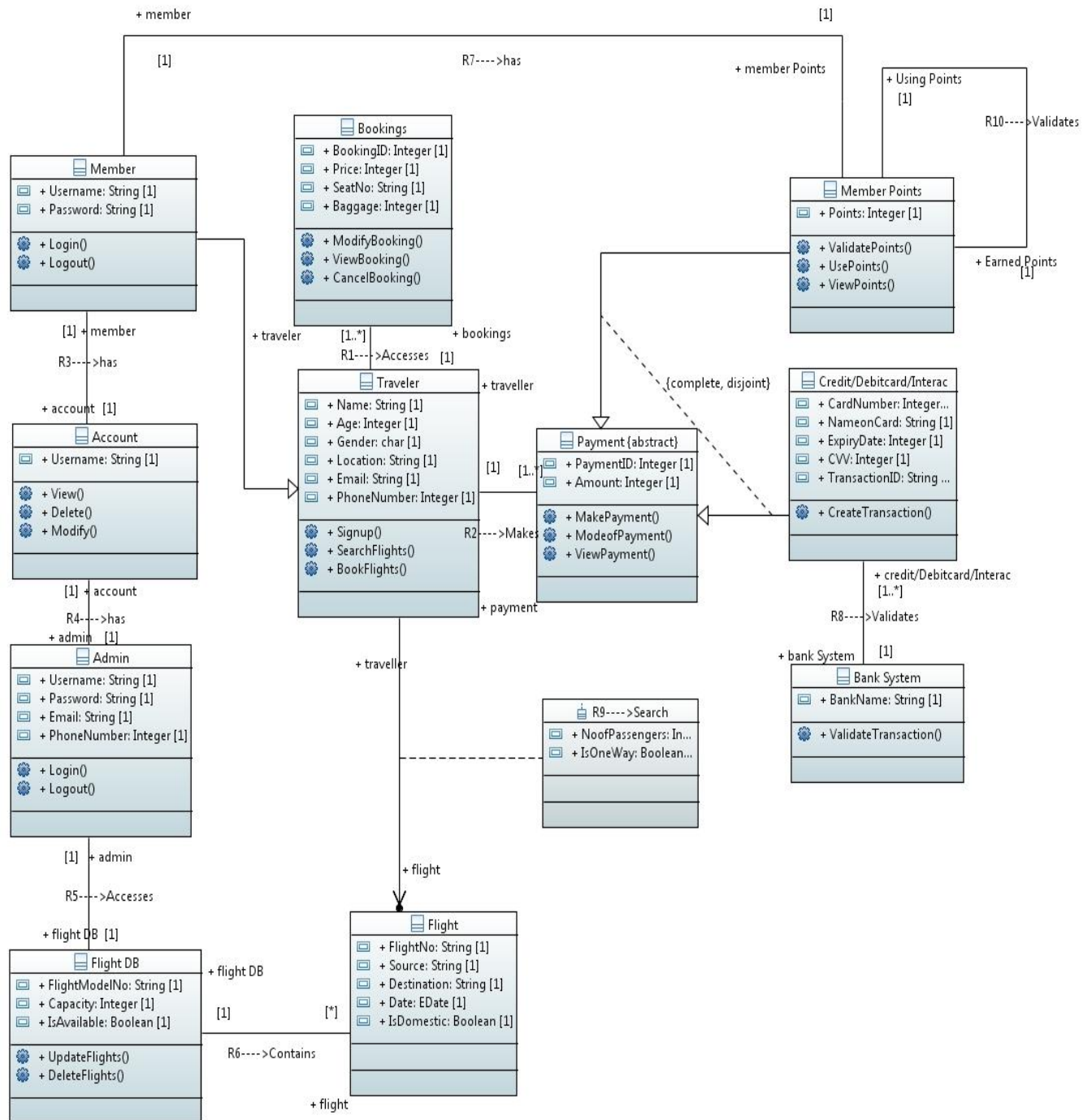
Project-Team

<u>Name</u>	<u>ID No.</u>
Ganesh Santhar	40010625
BhanuPrakashRamineni	27107838
Jithin James	27420854
RambabuKunchala	27262957
RakhiUbriani	27396333

Index

1. Class Diagram for FlyinTravel (Flight Reservation System)	2
2. Class Description	3
3. Object Constraint Language over Class Diagram.....	5

1. Class Diagram for FlyinTravel (Flight Reservation System)



2. Class Description

1. Traveler

The Traveler class is the base class for Member class. Traveler class abstracts all the objects of the traveler who is willing to book a flight using the system. This class contains the personal information of the traveler that is needed for booking. The attributes of the class will be Name, Age, Gender, Location, Email and PhoneNumber. The SearchFlights method is meant to retrieve the available flights of the flight class and the BookFlights method will book one of the retrieved flights. There is also a Signup method for creating an account in the system.

2. Member

The Member class has more properties and methods in addition to that of a Traveler class. They are represented with attributes and methods such as Username and Password, login and logout respectively. Member will accumulate points on bookings made. Each member will be identified uniquely by the attribute Username.

3. Account

The Account class contains accounts of Members and Administrator. Each Account is uniquely identified with the attribute Username. Its behavior is described by the methods such as View, Delete and Modify the account.

4. Admin

The Admin class will act as Administrator for the system. The Administrator can only update the flight database. This class is represented by properties such as Username, Password, Email and Phone Number. The behavior of the Admin is represented by methods such as login and logout.

5. FlightDB

The FlightDB class contains information of all the Flights. It has properties such as FlightModelNo, Capacity and IsAvailable. It Provides methods such as UpdateFlights and DeleteFlights to the Admin class. IsAvailable attribute gives the status of the flight, false indicates a cancelled flight.

6. Flight

The Flight Class shows available Flights to the Traveler that satisfies the search criteria. It is represented by attributes such as FlightNo, Source, Destination, Date, IsDomestic. A false in IsDomestic indicates an International flight.

7. Search

The Search Class is an association class for Traveler and Flight classes. It consists of common properties i.e. NumberofPassengers and IsOneWay that relates to both the classes. The false for the attribute IsOneWay indicates a RoundTrip search.

8. Bookings

The Bookings class comprises of all the bookings made by the traveler. It contains the details of each booking namely BookingID, Price of the booking, SeatNo and Baggage details. BookingID will uniquely identifies each Booking. The class contains methods such as ModifyBooking, ViewBooking and CancelBooking to describe its behavior.

9. Payment

The Payment class is the base class for MemberPoints and Credit/Debitcard/Interac classes. It contains of all the payments that are linked with the Traveler class. The properties of each payment is represented by the attributes such as PaymentID and Amount. Its behavior is represented by methods such as MakePayment, ModeofPayment and ViewPayment. Each payment will be uniquely identified by a PaymentID.

10.MemberPoints

The MemberPoints class plays a dual role i.e. containing the member points and validating the points for payment. It also displays the points earned by the member. It is identified by the attribute Points. It can be accessed by methods such as ValidatePoints, UsePoints and ViewPoints.

11.BankSystem

The BankSystem class represents the network of banks that connects with Flight Reservation System, FlyinTravel. It validates the transactions made via Credit/Debitcard/Interac payment. This is achieved by the attribute BankName and method ValidateTransactions.

12.Credit/Debitcard/Interac

The Credit/ Debitcard/ Interac class is a payment mode that holds the attributes CardNumber, NameonCard, ExpiryDate, CVV, TransactionID and a transaction that is initiated by CreateTransaction method.

3. Object Constraint Language over Class Diagram

1. The minimum age of traveler to access the system is 18 years.

Context Traveler
Inv: self.Age() \geq 18

2. An Account should be owned by a Member or Admin.

Context Account
Inv: self.R3->NotEmpty()
Or
self.R4->NotEmpty()

3. Booking should be successful only if the payment is made.

Context Traveler
Inv: self.R1.BookingID()->NotEmpty()
Implies
self.R2.PaymentID()->NotEmpty()

4. Admin must login to the Account to access the Flight database.

Context Admin
Inv: self.R5->NotEmpty()
Implies
self.R4->NotEmpty()

5. Calling then Signup method in Traveler creates an Account in the system.

Context Traveler::Signup(M)
pre: self.R3->excludes(M)
post: self.R3->includes(M)

6. Calling Delete method in Account should remove the Member Account in the system.

Context: Account::Delete(M)
pre: self.R3->includes(M)
post: self.R3->excludes(M)

7. Username of the accounts must be unique in the Account.

Context Account

Inv: $\text{self.allInstances()} \rightarrow \text{ForAll}(u1, u2 \mid u1 \neq u2 \text{ implies } u1.\text{Username} \neq u2.\text{Username})$

8. No two Booking IDs should be same.

Context Bookings

Inv: $\text{self.allInstances()} \rightarrow \text{ForAll}(C1, C2 \mid C1 \neq C2 \text{ implies } C1.\text{BookingID} \neq C2.\text{BookingID})$