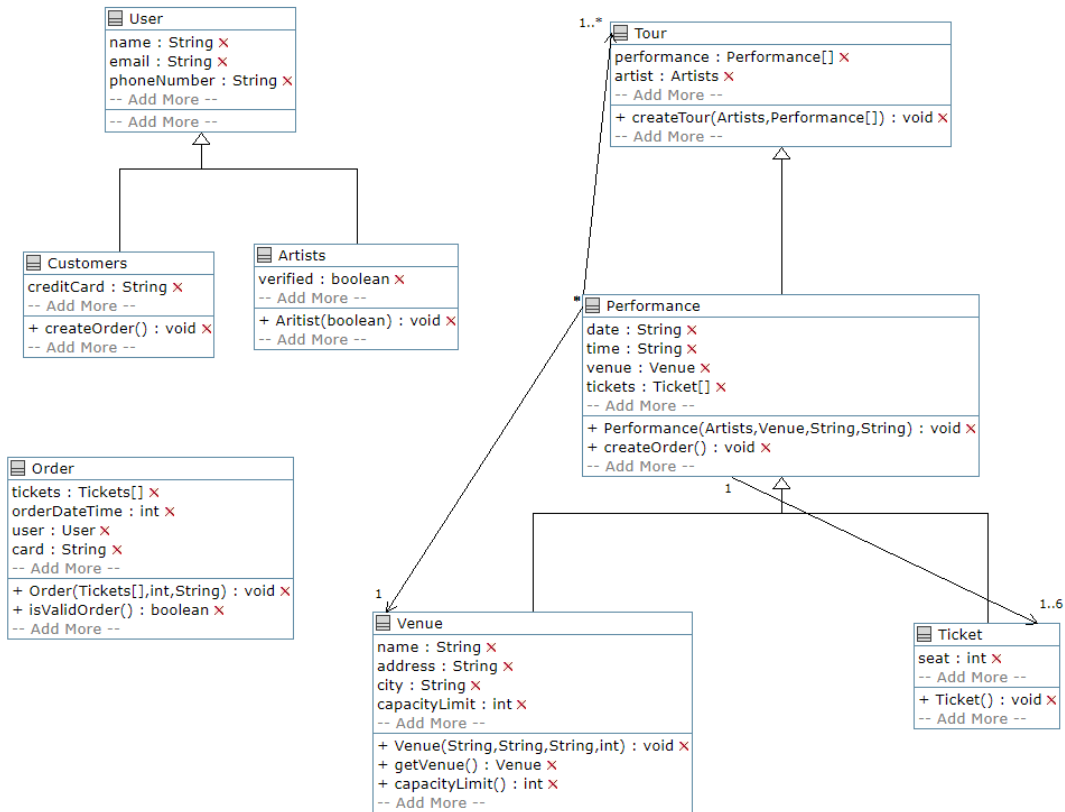


Lab 3 - Uml - Group 11

Question 1



namespace System.core;

```
class User {  
    String name;  
    String email;  
    String phoneNumber;  
  
}
```

```
class Tour{  
  
    Performance[] performance;  
    Artists artist;
```

```

public createTour(Artists artist, Performance[] performance){
    if(artist.getVerified()){
        this.performance = performance;
        this.artist = artist;
    }else{
        return;
    }
}

1..* <- * Performance;

}

class Performance{
    isA Tour;

    String date;
    String time;
    Artists artist;
    Venue venue;

    Ticket[] tickets;

    public Performance(Artists artist, Venue venue, String date, String time){
        if(artist.getVerified()){
            this.venue = venue;
            this.artist = artist;
            this.date = date;
            this.time = time;
            this.tickets = venue.getCapacityLimit();
        }else{
            return;
        }
    }

    public createOrder(){
        Order tempOrder = new Order(tickets, orderDateTime, creditCard);

```

```

    if(tempOrder.isVaildOrder()){

    }
}

}

class Venue {
    isA Performance;
    String name;
    String address;
    String city;
    int capacityLimit;

    public Venue(String name, String address, String city, int capacityLimit){
        this.name = name;
        this.address = address;
        this.city = city;
        this.capacityLimit = capacityLimit;
    }

    public Venue getVenue(){
        return this;
    }

    public int capacityLimit(){
        return this;
    }

    1 <- * Performance;

}

class Order {

    Tickets[] tickets;
    int orderDateTime;
    User user;
    String card;

```

```

public Order(Tickets[] ticket, int orderDateTime, String creditCard){
    this.card = creditCard;
    System.out.println(ticket);
}

public boolean isValidOrder(){
    return this.card.length() == 16;
}

}

class Ticket{
    isA Performance;
    int seat;

    public Ticket(){}

    1..6 <- 1 Performance;
}

class Artists {
    boolean verified;
    isA User;

    public Aritist(boolean verified){
        this.verified = verified;
    }

    public boolean getVerified(){
        return verified;
    }

}

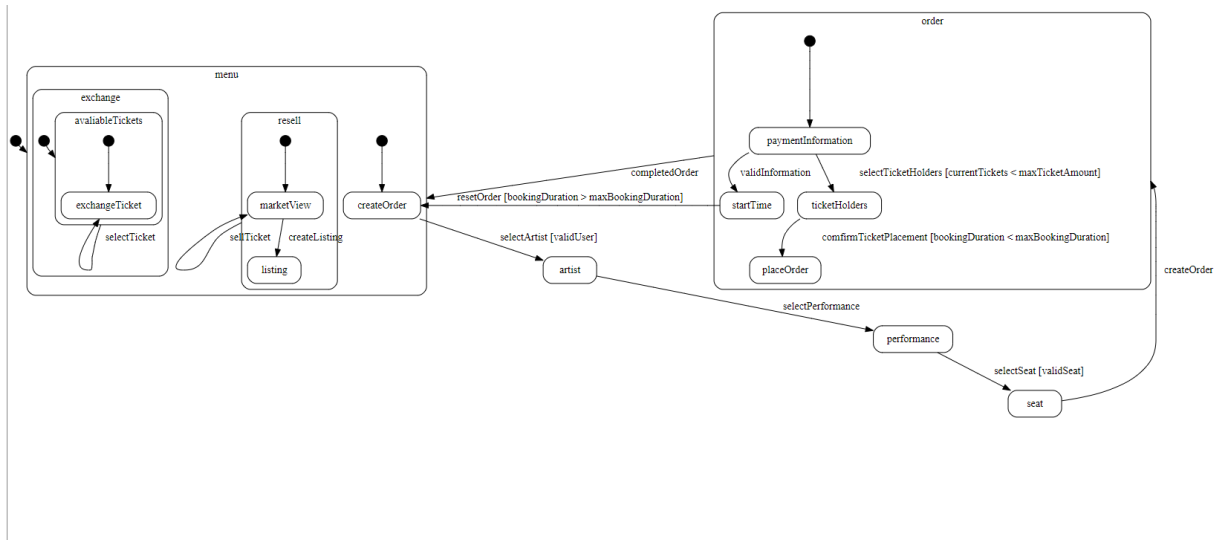
class Customers {
    isA User;
    String creditCard;

    public createOrder(){};

```

}

Question 2



```

class BookingMenu{

    int maxTicketAmount = 6;
    int currentTickets = 0;
    Boolean bookingDuration;
    Boolean maxBookingDuration;
    Boolean validUser;
    Boolean validSeat;

    state {

        menu {

            createOrder{
                selectArtist[validUser] -> artist;
            }

            resell {
                sellTicket -> marketView;
            }
        }
    }
}
  
```

```

    marketView {
      createListing -> listing;
    }

    listing{

    }

  }

  exchange {

    availableTickets{

      selectTicket -> exchangeTicket;

      exchangeTicket {

      }

    }

  }

}

artist{
  selectPerformance -> performance;
}
performance{
  selectSeat[validSeat] -> seat;
}
seat{
  createOrder -> order;
}

order{

```

```
paymentInformation{
  validInformation -> startTime;
  selectTicketHolders[currentTickets < maxTicketAmount ] -> ticketHolders;
}

startTime {
  resetOrder[bookingDuration > maxBookingDuration] -> createOrder;
}

ticketHolders{
  confirmTicketPlacement[bookingDuration < maxBookingDuration] -> placeOrder;
}

placeOrder{

}

completedOrder -> menu;
}

}

}
```