Concordia University
Engineering and Computer Science

COEN 6312
Model Driven Software Engineering
Winter 2016

Deliverable 4
(ONLINE BOOK STORE)

(Team-Singh)

Guneet Singh
Rattandeep Singh
Satinder pal Singh

Table of Contents

1. Class: Shopping Cart

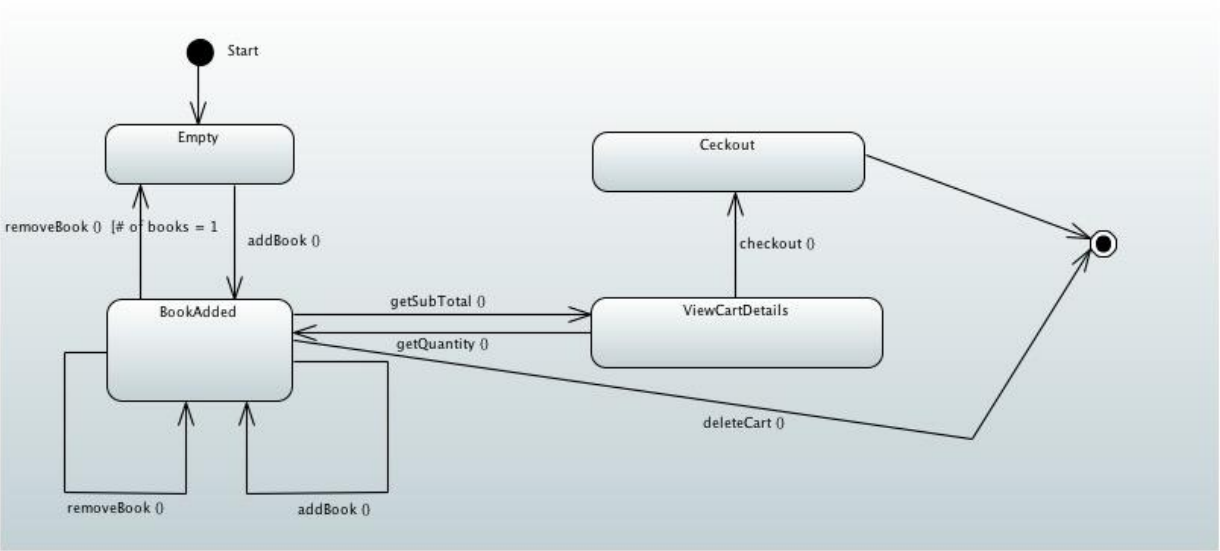

Figure 1: Shopping Cart State Machine Diagram

Action Specification:

```
public class ShoppingCart {
        public int getCartID() {
                return cartID; }
        public void setCartID(int cartID) {
                this.cartID = cartID; }
        public int getQuantity() {
                return quantity; }
        public void setQuantity(int quantity) {
                this.quantity = quantity; }
        public String getBookId() {
                return bookId; }
        public void setBookId(String bookId) {
                this.bookId = bookId; }
        public int getOrderId() {
                return orderId; }
        public void setOrderId(int orderId) {
                this.orderId = orderId; }
        public void getQuantity() {
```

```
		// TODO Auto-generated method
	}
	public void removeBook() {
		// TODO Auto-generated method
	}
	public void viewCartDetails() {
		// TODO Auto-generated method
	}
	public void checkout() {
		// TODO Auto-generated method
	}
	public void getSubTotal() {
		// TODO Auto-generated method
	}
	public void deleteCart() {
		// TODO Auto-generated method
	}
	public void addBook() {
		// TODO Auto-generated method
	}
```

```
:addBook()
If This.add_book==1 and book_added==0
Then book_added == 1
and If this.remove_book == 1
Then book_added == 0
endid;

:getSubTotal(Int book_price, Int total)
If This.bookadded != 0
return (totalPrice)
endif;

:getQuantity(Int quantity)
If This.bookadded > 0
This.viewQuantity()
endif;
```

2. Class:Bank

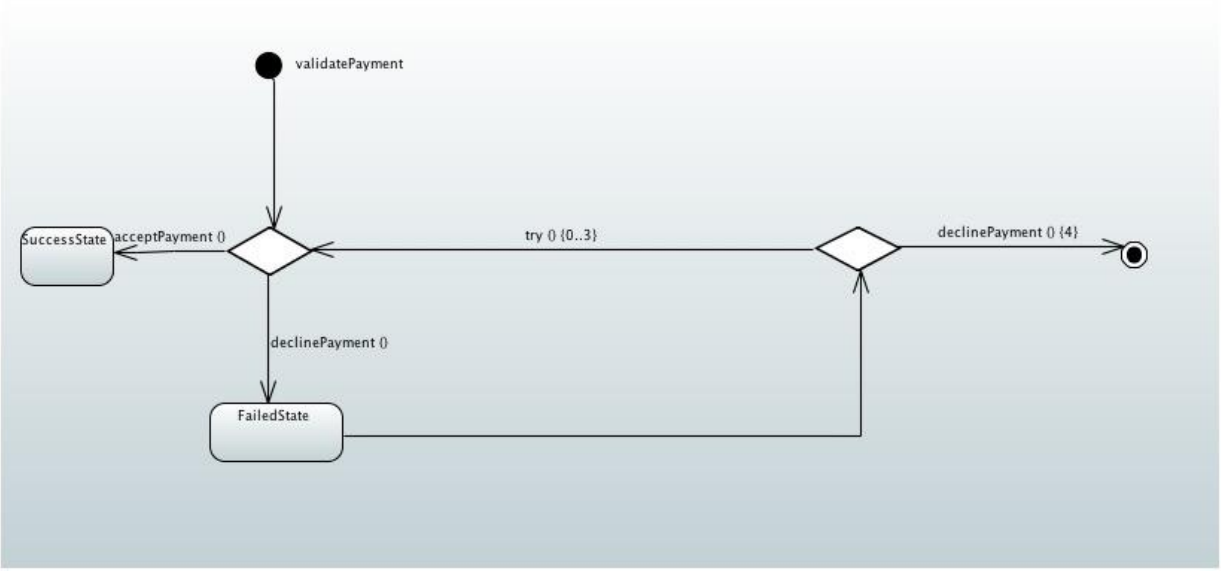

Figure 2: Bank State Machine Diagram

Action Specification:

```
public class Payment
{
        String payment_status;

        public static makePayment(int amount,int card_number, String card_holder_name)
        {
                if (validateDetails(card_number,card_holder_name)==true
                                and sufficientAmount(amount)==true)
                {
                        payment_status=paymentConfirm();//display the status about payment
                        System.out.println(payment_status)
                }
                else generateAlert() // display error message
        }

        public String validatePayment(int card_number, String card_holder_name )
        {
                //validate card information
                return status;// status will be local variable which confirm the card details
```

```
        }

        public int sufficientAmount(int amount )
        {
                //validate sufficient balance
                return status;// status will be local variable which display the sufficient amount
        }
        public String acceptPayment()
        {
                //display message
                return status;// status will be local variable which display the payment status
        }

}
```

```
:validatePayment(amount,payment_status, card_number, card_holder_name)
If this.validateCard (card_number, card_holder_name) == true and this.sufficientAmount == true
Then this.payment_status = This.acceptPayment()
Alert(payment_status)
endif;

:declinePayment()
If this.sufficientAmount(amount)==false
Then
If this.try() <= 3;
this.validateCard (card_number, card_holder_name) == true and this.sufficientAmount == true
Then this.payment_status = This.acceptPayment()
Alert(payment_status)
Else
declinePaymentConfirmation
Alert(declinePayment)
endif;
```

3. Class:Customer

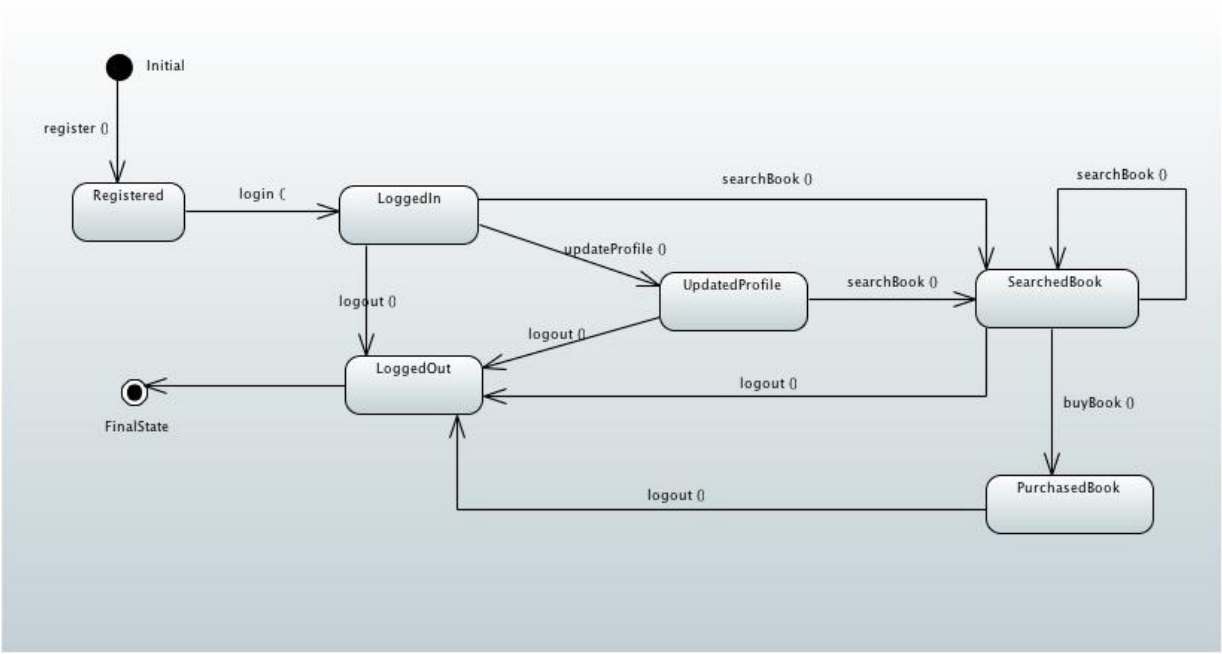

Figure 3: Customer State Machine Diagram

Action Specification:

```
public class Registercustomer {

public static void register(String customer_name, int password,String first_name,String
last_name,//others)
        {
                //code to register customer by using all the parameters
        }
}


public class Book{

        Public static void searchBook(book_id, book_title, book_author, book_edition)
        {
        if(book_id!=null and book_title=null and book_author=null and book_edition!=null)
        {// here we code for search flights
```

```
                // String result is local variable to store search result.
                ViewBookInformation( result); // this is for displaying search result

        }
        else
                generateAlert() // display error message
        }

}
        public static ViewBookInformation(String result)
        { // display search result

                // System.out.print(result)   to display search}

}
        }
```

```
:register(String customer_name, int password,String first_name,String last_name)
If this. customer_name!=null and this.password!=null and this. first_name!=null and this.
last_name!=null Then This.customerSaved = This.saveCustomer ()
Alert(customerSaved);
endif;

:searchBook(String book_id, book_title, book_author, book_edition)

book_id!=null and book_title=null and book_author=null and book_edition!=null

Then This.ViewBookInformation = This.BookInformationViewed()
Alert(ViewBookInformation);
endif;
```