- **Name:** Peter Delevoryas
- **Project:** Eat-net
- **Requirements:**
  I am working on UR-4, UR-5, UR-6, which correspond to the use cases UC-04, UC-05, and UC-6. These requirements are “Being able to add food items to a persistent shopping cart”, “Viewing and managing the persistent shopping cart”, and “Making an online payment for the items in the shopping cart”.

Use Case ID:	UC-04
Use Case Name:	Add items to cart
Description:	Consumers can add food items to a persistent, virtual shopping cart.

Actors:	Consumer		
Pre-conditions:	Food items can be viewed by the consumer, and the consumer can navigate to the items they want to purchase.		
Post-conditions:	Virtual shopping cart contains food items selected by consumer.		
Frequency of Use:	One or more times per Eat-net consumer order.		
Flow of Events:		**Actor Action**	**System Response**
	1	Navigates to food item	
	2	Clicks on “add to cart” button	Webpage script adds item ID to browser cookie which constitutes the virtual shopping cart.
	3	Navigates to another food item	
	4	Clicks on “add to cart” button	Browser cookies are persistent across navigation, reloading, leaving site, etc, so cart still contains previous item, and the new item is also added.
Variations:	None		
Exceptions:	Out of memory, invalid cookie state		
Developer Notes:			

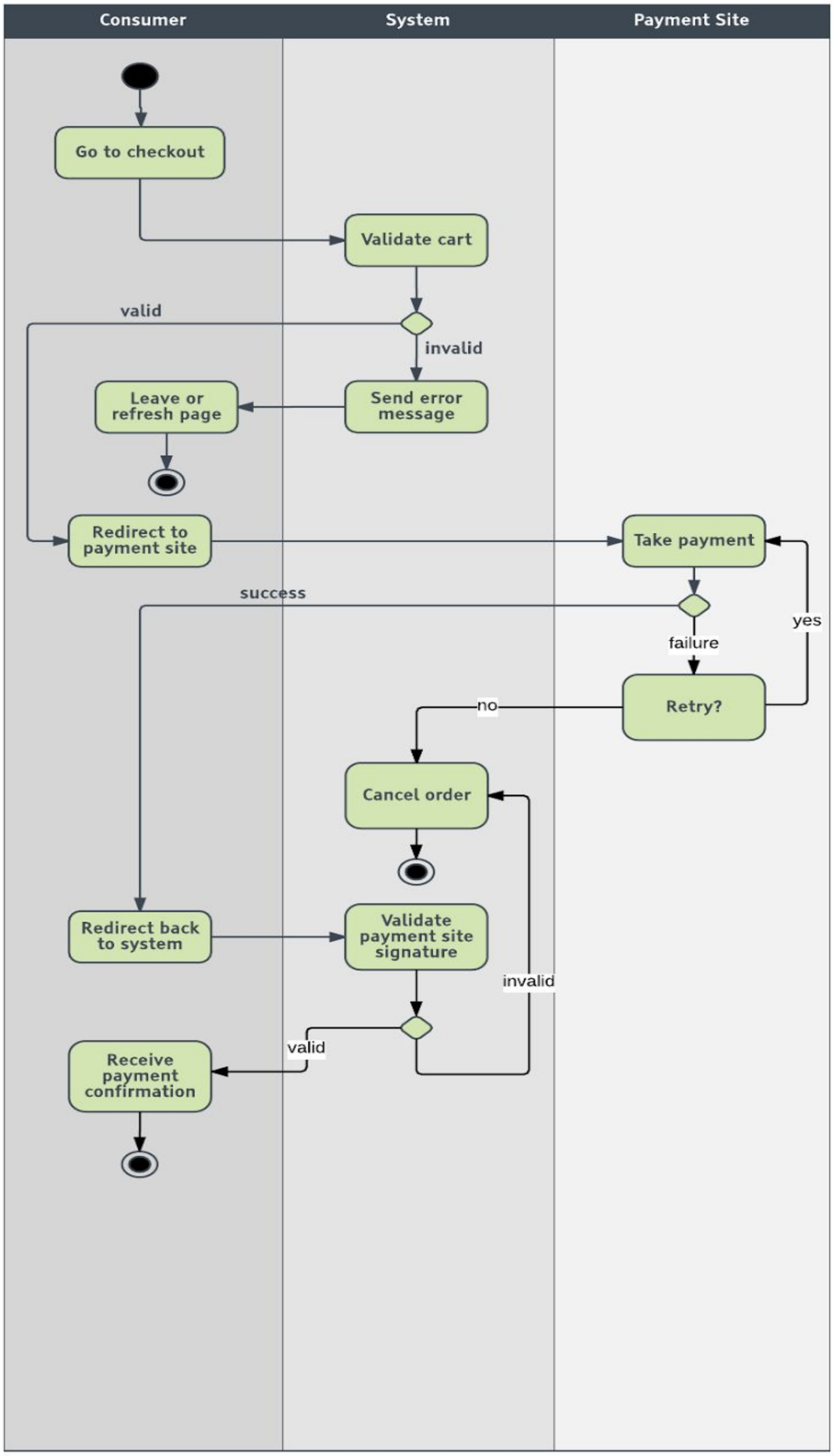
Use Case ID:	UC-05
Use Case Name:	View shopping cart
Description:	Consumers can view the items in their virtual shopping cart, as well as manage the items (change quantity, remove).

Actors:	Consumer		
Pre-conditions:	The consumer may or may not have visited the site previously, and the consumer may or may not have added items to the virtual shopping cart.		
Post-conditions:	The consumer's shopping cart has been modified or viewed appropriately.		
Frequency of Use:	Once per consumer checkout, and/or more often while browsing.		
Flow of Events:		**Actor Action**	**System Response**
	1	Sends browser cookies to system over HTTPS.	Validates cookie state, looks up info for items in cart, sends back shopping cart view
	2	User changes quantity for item using text field or button.	Client-side script changes view and updates cookies.
	3	User clicks button removing item from cart.	Client-side script changes view and removes item from cookies.
Variations:	None		
Exceptions:	Out of memory, invalid cookie state, invalid quantity		
Developer Notes:			

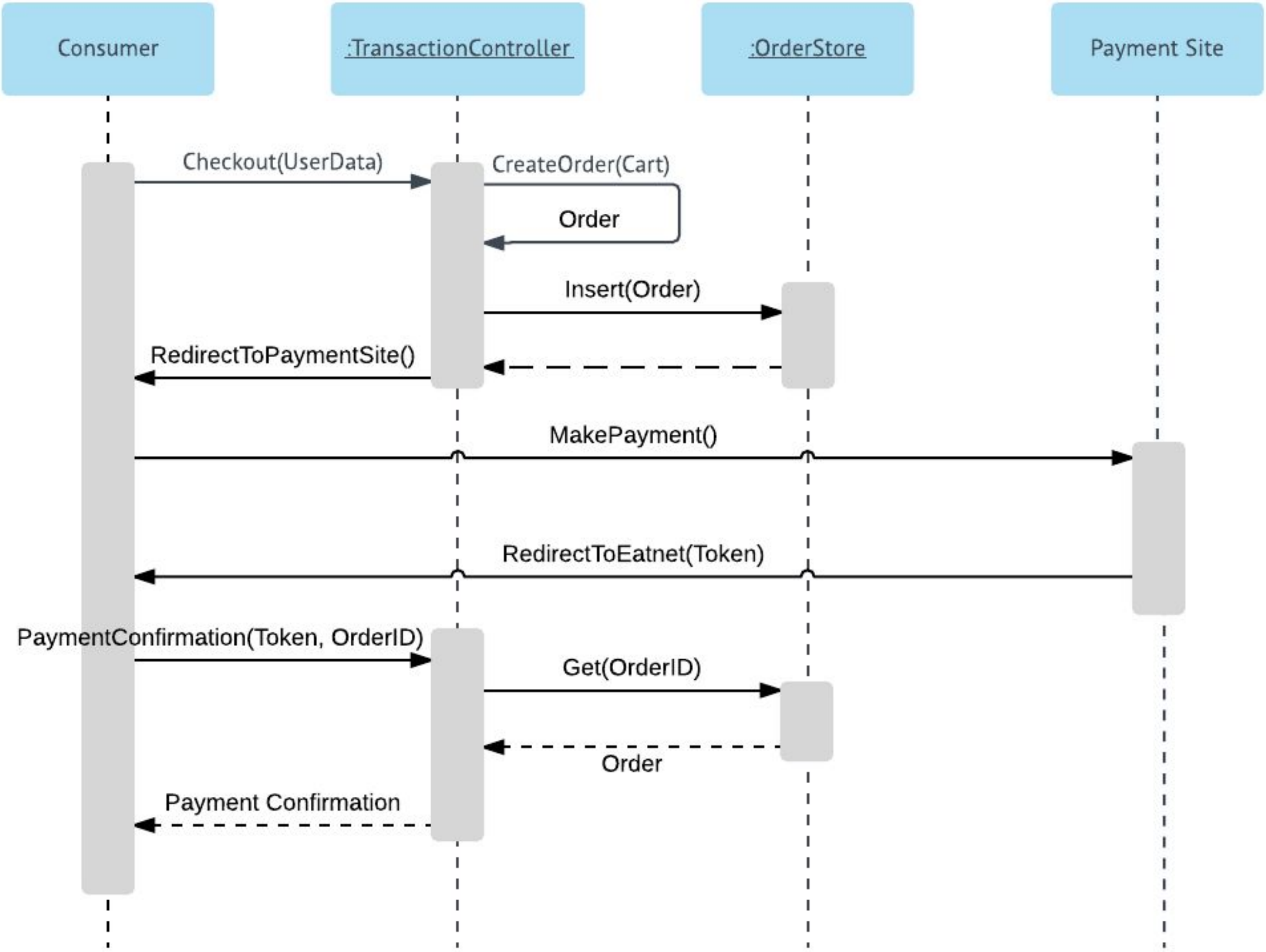
Use Case ID:	UC-06
Use Case Name:	Make Payment
Description:	Consumers can pay for the items in their virtual shopping cart, and payment is distributed to the restaurants corresponding to each item.

Actors:	Consumer, Business		
Pre-conditions:	The consumer has added one or more items to their virtual shopping cart. Business users have added payment information to the Eat-net database. Consumer has valid form of payment.		
Post-conditions:	Consumer is given payment receipt. Payment is distributed to respective business users. Transaction amount is represented correctly.		
Frequency of Use:	Once per consumer checkout.		
Flow of Events:		**Actor Action**	**System Response**
	1	Navigates to virtual checkout, sending current virtual shopping cart to system	Validates shopping cart, calculates price of individual items and total.
	2	User confirms intent to make payment.	Redirect to payment website securely with appropriate transaction info.
	3	User completes transaction on external payment website.	Payment website redirects back to Eat-net
	4	User receives payment confirmation page and other order information.	
Variations:	None		
Exceptions:	Out of memory, invalid cookie state, transaction completion failure		
Developer Notes:			

- **Activity Diagram:**
  For the activity diagram, I have selected UC-06, "Make Payment". It is shown on the next page.



- **User Interactions:**
  This is the sequence diagram of the objects interacting in UC-06, “Make Payment”.


Consumer
:TransactionController
:OrderStore
Payment Site
Checkout(UserData)
CreateOrder(Cart)
Order
Insert(Order)
RedirectToPaymentSite()
MakePayment()
RedirectToEatnet(Token)
PaymentConfirmation(Token, OrderID)
Get(OrderID)
Order
Payment Confirmation