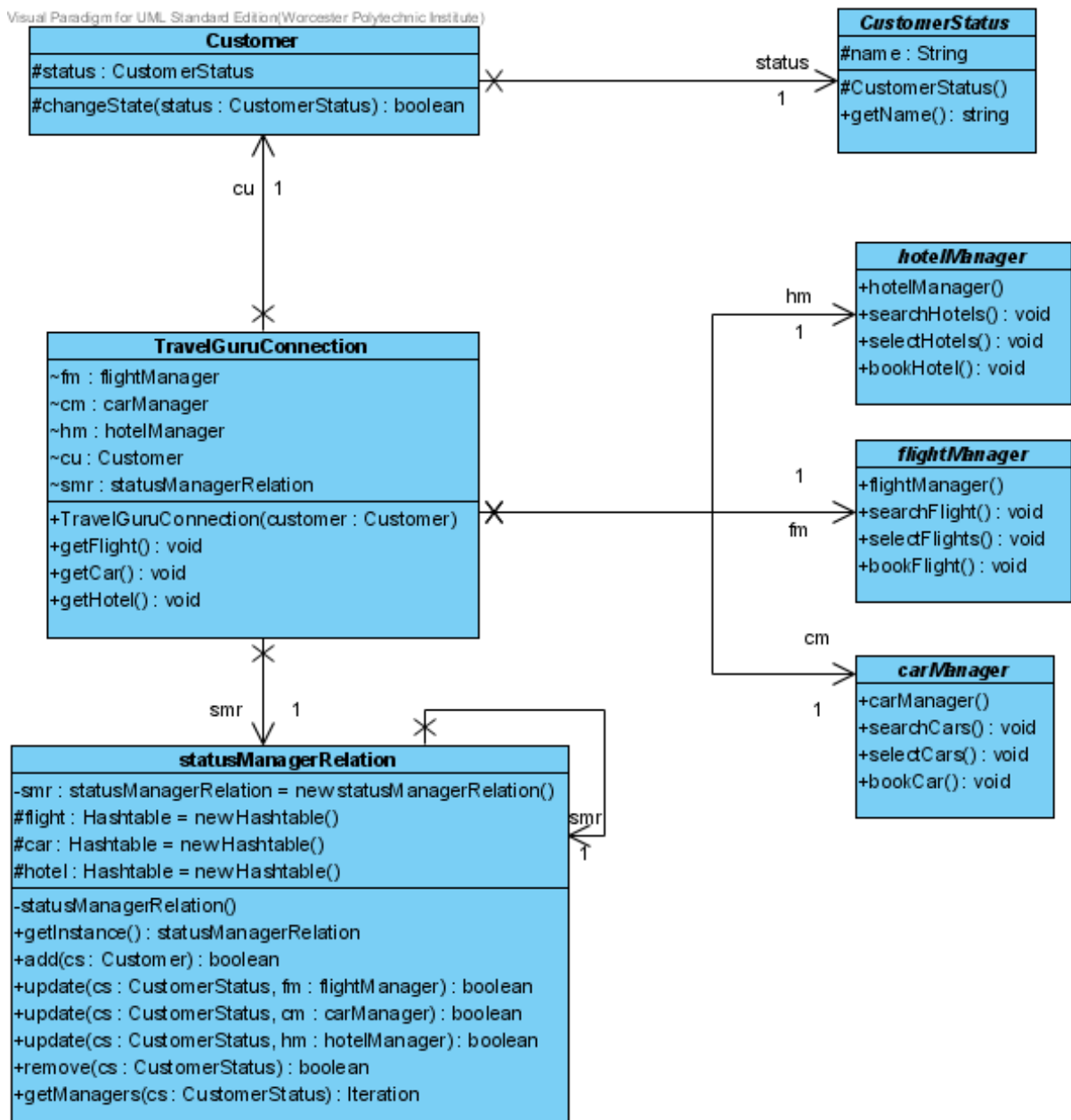









Class Diagram



egreer HW7-A08

Visual Paradigm for UML Standard Edition (Worcester Polytechnic Institute)



Summary

Name	Documentation
 CustomerStatus	<p>The CustomerStatus is a class that identifies the customer's status with TravelWeenie.</p> <p>CustomerStatus implements the state pattern which allows for changing individual states without affecting others. It is an abstract class instead of an interface so that any default behaviour can be implemented.</p> <p>Other Statuses SubClass this class and create their own methods that would override these defaults like: Member, Previous_Customer, Has_Profile</p> <p>CustomerStatus probably has other methods that are currently not part of the system to change state and store other information.</p>
 Customer	<p>Customer remains basically the same just has a new variable called status for the currentCustomerStatus.</p> <p>Customer has other methods relating to the customer but are not important to the structure of this system.</p>
 hotelManager	<p>If a customer wants to search for and reserve hotels, the hotelManager will do this by the following process:</p> <ol style="list-style-type: none">1. Search for hotels.2. Select hotels.3. Book hotels. <p>The hotelManager is an abstract class because it contains default methods that can be overridden by subclasses.</p>
 TravelGuruConnection	<p>When a customer connects to the TravelWeenie Web site, a connection is created. This connection handles all transactions for the customer for the current session. Whenever the customer wants to find flights, cars, or hotels for a trip, this connection manages each of the tasks.</p> <p>TravelGuruConnection implements the Strategy Pattern. It goes through the algorithm that doesn't change and relies on subclasses to implement the behaviour. The subclasses are all managers</p>
 flightManager	<p>If a customer wants to search for and reserve flights, the flightManager will do this by the following process:</p> <ol style="list-style-type: none">1. Search for flights.2. Select flights.3. Book flights. <p>The flightManager is an abstract class because it contains default methods that can be overridden by subclasses.</p>

 carManager	<p>If a customer wants to search for and reserve cars, the carManager will do this by the following process:</p> <ol style="list-style-type: none"> 1. Search for cars. 2. Select cars. 3. Book cars. <p>The carManager is an abstract class because it contains default methods that can be overridden by subclasses.</p>
 statusManagerRelation	<p>statusManagerRelation contains tables that connect customerStatus to the Managers for each type of search.</p> <p>This helps to eliminate the abundance of dependencies between each status and its relationships.</p> <p>This implements the singleton pattern to make sure that there is only one set of tables.</p>

Details

CustomerStatus

Name	Value
Active	false
Business Model	false
Visibility	public
Leaf	false
Root	false
Documentation	<p>The CustomerStatus is a class that identifies the customer's status with TravelWeenie.</p> <p>CustomerStatus implements the state pattern which allows for changing individual states without affecting others It is an abstract class instead of an interface so that any default behaviour can be implemented.</p> <p>Other Statuses SubClass this class and create their own methods that would override these defaults like: Member, Previous_Customer, Has_Profile</p> <p>CustomerStatus probably has other methods that are currently not part of the system to change state and store other information.</p>

Attributes


protected name : String			
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		

Operations

protected CustomerStatus ()	
Static	false
Leaf	false
Documentation	Constructor for creating a default CustomerStatus
Ordered	false
Unique	true
Query	false

public getName () : string	
Static	false
Leaf	false
Documentation	Returns the name of the status.
Ordered	false
Unique	true
Query	false

Relationships

Unnamed Association		
From	Name	Value
	End Model Element	 Customer
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	private
	Aggregation Kind	None
	Navigable	false
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

References

C:\Users\egreer\Documents\hw7\CustomerStatus.java	
Description	Source
Type	File




Customer

Name	Value
Active	false
Business Model	false
Visibility	public
Leaf	false
Root	false
Documentation	<p>Customer remains basically the same just has a new variable called status for the currentCustomerStatus.</p> <p>Customer has other methods relating to the customer but are not important to the structure of this system.</p>


Attributes


protected status : CustomerStatus			
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		

Operations

protected changeState (status : CustomerStatus) : boolean		
Parameters	status	
	Multiplicity	Unspecified
	Type	 CustomerStatus
	Direction	inout
Static	false	
Leaf	false	
Documentation	Changes the current Status to a new CustomerStatus if needed.	
Ordered	false	
Unique	true	
Query	false	

Relationships

Unnamed Association		
To (status)	Name	Value
	End Model Element	 CustomerStatus
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	1
	Visibility	protected
	Aggregation Kind	None
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

Unnamed Association		
From	Name	Value
	End Model Element	 TravelGuruConnection
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	private
	Aggregation Kind	None
	Navigable	false
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

References

C:\Users\legreer\Documents\hw7\Customer.java	
Description	Source
Type	File

hotelManager

Name	Value
Active	false
Business Model	false
Visibility	public
Leaf	false
Root	false
Documentation	<p>If a customer wants to search for and reserve hotels, the hotelManager will do this by the following process:</p> <ol style="list-style-type: none">1. Search for hotels.2. Select hotels.3. Book hotels. <p>The hotelManager is an abstract class because it contains default methods that can be overridden by subclasses.</p>

Operations


public hotelManager ()	
Static	false
Leaf	false
Documentation	Default constructor for the hotelManager
Ordered	false
Unique	true
Query	false

public searchHotels () : void	
Static	false
Leaf	false
Documentation	Default Method for searching hotels
Ordered	false
Unique	true
Query	false

public selectHotels () : void	
Static	false
Leaf	false
Documentation	Default Method for selecting hotels
Ordered	false
Unique	true
Query	false

public bookHotel () : void	
Static	false
Leaf	false
Documentation	Default Method for booking hotels
Ordered	false
Unique	true
Query	false

Relationships

Unnamed Association		
From	Name	Value
	End Model Element	 TravelGuruConnection
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	private
	Aggregation Kind	None
	Navigable	false
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

References

C:\Users\egreer\Documents\hw7\hotelManager.java	
Description	Source
Type	File

TravelGuruConnection

Name	Value
Active	false
Business Model	false
Visibility	public
Leaf	false
Root	false
Documentation	<p>When a customer connects to the TravelWeenie Web site, a connection is created. This connection handles all transactions for the customer for the current session. Whenever the customer wants to find flights, cars, or hotels for a trip, this connection manages each of the tasks.</p> <p>TravelGuruConnection implements the Strategy Pattern. It goes through the algorithm that doesn't change and relies on subclasses to implement the behaviour. The subclasses are all managers</p>

Attributes

package fm : flightManager			
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		


package cm : carManager			
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		

package hm : hotelManager			
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		

package cu : Customer			
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		

package smr : statusManagerRelation			
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		

Operations


public TravelGuruConnection (customer : Customer)		
Parameters	customer	
	Multiplicity	Unspecified
	Type	 Customer
	Direction	inout
Static	false	
Leaf	false	
Documentation	Constructor for the TravelGuruConnection	
Ordered	false	
Unique	true	
Query	false	


public getFlight () : void	
Static	false
Leaf	false
Documentation	Implements the algorithm of searching through flights, selecting flights, and then booking them.
Ordered	false
Unique	true
Query	false


public getCar () : void	
Static	false
Leaf	false
Documentation	Implements the algorithm of searching through cars, selecting cars, and then booking them.
Ordered	false
Unique	true
Query	false


public getHotel () : void	
Static	false
Leaf	false
Documentation	Implements the algorithm of searching through hotels, selecting hotels, and then booking them.
Ordered	false
Unique	true
Query	false


Relationships

Unnamed Association		
To (fm)	Name	Value
	End Model Element	 flightManager
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	1
	Visibility	package
	Aggregation Kind	None
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

Unnamed Association		
To (cm)	Name	Value
	End Model Element	 carManager
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	1
	Visibility	package
	Aggregation Kind	None
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

Unnamed Association		
To (hm)	Name	Value
	End Model Element	 hotelManager
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	1
	Visibility	package
	Aggregation Kind	None
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

Unnamed Association		
To (cu)	Name	Value
	End Model Element	 Customer
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	1
	Visibility	package
	Aggregation Kind	None
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

Unnamed Association		
To (smr)	Name	Value
	End Model Element	 statusManagerRelation
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	1
	Visibility	package
	Aggregation Kind	None
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

References

C:\Users\egreen\Documents\hw7\TravelGuruConnection.java	
Description	Source
Type	File

flightManager

Name	Value
Active	false
Business Model	false
Visibility	public
Leaf	false
Root	false
Documentation	<p>If a customer wants to search for and reserve flights, the flightManager will do this by the following process:</p> <ol style="list-style-type: none"> 1. Search for flights. 2. Select flights. 3. Book flights. <p>The flightManager is an abstract class because it contains default methods that can be overridden by subclasses.</p>

Operations


public flightManager ()	
Static	false
Leaf	false
Documentation	Default Constructor for the flightManager
Ordered	false
Unique	true
Query	false

public searchFlight () : void	
Static	false
Leaf	false
Documentation	Default Method for searching flights
Ordered	false
Unique	true
Query	false

public selectFlights () : void	
Static	false
Leaf	false
Documentation	Default Method for selecting flights
Ordered	false
Unique	true
Query	false

public bookFlight () : void	
Static	false
Leaf	false
Documentation	Default Method for booking flights
Ordered	false
Unique	true
Query	false

Relationships

Unnamed Association		
From	Name	Value
	End Model Element	 TravelGuruConnection
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	private
	Aggregation Kind	None
	Navigable	false
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

References

C:\Users\egreen\Documents\hw7\flightManager.java	
Description	Source
Type	File

carManager

Name	Value
Active	false
Business Model	false
Visibility	public
Leaf	false
Root	false
Documentation	<p>If a customer wants to search for and reserve cars, the carManager will do this by the following process:</p> <ol style="list-style-type: none"> 1. Search for cars. 2. Select cars. 3. Book cars. <p>The carManager is an abstract class because it contains default methods that can be overridden by subclasses.</p>

Operations


public carManager ()	
Static	false
Leaf	false
Documentation	Default constructor for the carManager
Ordered	false
Unique	true
Query	false

public searchCars () : void	
Static	false
Leaf	false
Documentation	Default Method for searching cars
Ordered	false
Unique	true
Query	false

public selectCars () : void	
Static	false
Leaf	false
Documentation	Default Method for selecting cars
Ordered	false
Unique	true
Query	false

public bookCar () : void	
Static	false
Leaf	false
Documentation	Default Method for booking Cars
Ordered	false
Unique	true
Query	false

Relationships

Unnamed Association		
From	Name	Value
	End Model Element	 TravelGuruConnection
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	private
	Aggregation Kind	None
	Navigable	false
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

References

C:\Users\egreen\Documents\hw7\carManager.java	
Description	Source
Type	File

statusManagerRelation

Name	Value
Active	false
Business Model	false
Visibility	public
Leaf	false
Root	false
Documentation	<p>statusManagerRelation contains tables that connect customerStatus to the Managers for each type of search.</p> <p>This helps to eliminate the abundance of dependencies between each status and its relationships.</p> <p>This implements the singleton pattern to make sure that there is only one set of tables.</p>

Attributes

private smr : statusManagerRelation			
Initial Value	new statusManagerRelation()		
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		

protected flight : Hashtable			
Initial Value	new Hashtable()		
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		


protected car : Hashtable			
Initial Value	new Hashtable()		
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		



protected hotel : Hashtable			
Initial Value	new Hashtable()		
Getter	false	Setter	false
Derived	false		
Multiplicity	Unspecified		
Aggregation	None		



Operations



private statusManagerRelation ()	
Static	false
Leaf	false
Documentation	protected constructor to implement the singleton pattern
Ordered	false
Unique	true
Query	false


public getInstance () : statusManagerRelation	
Static	false
Leaf	false
Documentation	Returns the SMR to implements the singleton pattern.
Ordered	false
Unique	true
Query	false


public add (cs : Customer) : boolean		
Parameters	cs	
	Multiplicity	Unspecified
	Type	 Customer
	Direction	inout
Static	false	
Leaf	false	
Documentation	Adds a new customerStatus to the tables with default managers	
Ordered	false	
Unique	true	
Query	false	

public update (cs : CustomerStatus, fm : flightManager) : boolean		
Parameters	cs	
	Multiplicity	Unspecified
	Type	 CustomerStatus
	Direction	inout
	fm	
	Multiplicity	Unspecified
	Type	 flightManager
	Direction	inout
Static	false	
Leaf	false	
Documentation	Adds a new customerStatus to the flight table with given manager	
Ordered	false	
Unique	true	
Query	false	


public update (cs : CustomerStatus, cm : carManager) : boolean		
Parameters	cs	
	Multiplicity	Unspecified
	Type	 CustomerStatus
	Direction	inout
	cm	
	Multiplicity	Unspecified
	Type	 carManager
	Direction	inout
Static	false	
Leaf	false	
Documentation	Adds a new customerStatus to the car table with given manager	
Ordered	false	
Unique	true	
Query	false	


public update (cs : CustomerStatus, hm : hotelManager) : boolean		
Parameters	cs	
	Multiplicity	Unspecified
	Type	 CustomerStatus
	Direction	inout
	hm	
	Multiplicity	Unspecified
	Type	 hotelManager
	Direction	inout
Static	false	
Leaf	false	
Documentation	Adds a new customerStatus to the hotel table with given manager	
Ordered	false	
Unique	true	
Query	false	


public remove (cs : CustomerStatus) : boolean		
Parameters	cs	
	Multiplicity	Unspecified
	Type	 CustomerStatus
	Direction	inout
Static	false	
Leaf	false	
Documentation	Removes the CustomerStatus from all tables	
Ordered	false	
Unique	true	
Query	false	

public getManagers (cs : CustomerStatus) : Iteration		
Parameters	cs	
	Multiplicity	Unspecified
	Type	 CustomerStatus
	Direction	inout
Static	false	
Leaf	false	
Documentation	Returns the managers for that Status	
Ordered	false	
Unique	true	
Query	false	

Relationships

Unnamed Association		
To (smr)	Name	Value
	End Model Element	 statusManagerRelation
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	1
	Visibility	private
	Aggregation Kind	None
	Navigable	true
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

Unnamed Association		
From	Name	Value
	End Model Element	 statusManagerRelation
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	private
	Aggregation Kind	None
	Navigable	false
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

Unnamed Association		
From	Name	Value
	End Model Element	 TravelGuruConnection
	Provide Property Getter Method	false
	Provide Property Setter Method	false
	Multiplicity	Unspecified
	Visibility	private
	Aggregation Kind	None
	Navigable	false
Abstract	false	
Leaf	false	
Visibility	Unspecified	
Derived	false	

References

C:\Users\egreer\Documents\hw7\statusManagerRelation.java	
Description	Source
Type	File