

Chapter 5 – Extending the Requirements Models

Solutions to End-of-Chapter Problems

Review Questions

1. What are the models that describe use cases in more detail?

Fully developed use case description, activity diagrams, and system sequence diagrams.

2. What two UML diagrams are used to model domain classes?

Problem domain class diagram and state machine diagram.

3. Which part of a use case description can also be modeled by using an activity diagram?

The “flow of activities” section.

4. Explain the difference between a use case and a scenario. Give a specific example of a use case with a few possible scenarios.

A use case is the entire function or user goal or event. A scenario is one specific version or instance of that use case.

From RMO we have *Create customer account* as a use case. But we might have *Create online Customer account* and *Create instore customer account* and even *Create phone customer account* as different scenarios.

5. List the parts or compartments of a fully developed use case description.

- Use case name
- Scenario
- Triggering event
- Brief description
- Actors
- Related use cases
- Stakeholders
- Preconditions
- Postconditions
- Flow of activities
- Exception conditions

6. Compare/contrast precondition and postcondition.

A precondition describes the “states” of data and the system that must exist before the use case can begin. For example to add an item to a shopping cart, the item must exist in the database.

A postcondition describes the states of data and the system that must exist after the use case completes. For example after *Create a customer account* use case, a customer account object (record) must exist.

7. Compare/contrast postcondition and exception condition.

A post condition, as explained about describes states of the data and the system. An exception condition describes some non-normal situation in the processing, i.e. in the flow of activities, that must be handled in some way..

8. Compare/contrast business process and flow of activities for a use case. Explain how an activity diagram can be used to model both.

A business process is larger than a use case. A business process might include various manual business procedures both before and after the “business event” that causes the use case to occur. However, an activity diagram is a powerful model to describe all types of sequences of tasks and activities. The various swimlanes can represent various users or user groups as well as system activities.

9. What is the purpose of an SSD? What symbols are used in an SSD?

An SSD (system sequence diagram) is used to describe the messages that flow into and out of a system, i.e. between the system and the use case user. The symbols include:

- Stick figure for the actor
- Box with object name for the system object
- Vertical dashed lines for object lifelines
- Horizontal arrows for messages
- Horizontal dashed arrows for return data
- Comment box for comments

10. What are the steps required to develop an SSD?

1. Identify the input messages (from the activity diagram)
2. Describe the input message using the SSD message syntax
3. Add message conditions such as looping or true/false conditions
4. Add all output message data

11. Write a complete SSD message from the actor to the system, with the actor asking the system to begin the process for updating information about a specific product.

updateProductInformation (productID, updateInformation)

12. What is the name of the sequence diagram symbol used to represent the extension of an object throughout the duration of a use case?

Lifeline or object lifeline

13. What are the two ways to show a returned value on a sequence diagram?

Either with a return value on the left hand side of the equal (:=) sign, or with a return message shown by a dashed line and labeled with the return data.

14. What are two ways to show repetition on a sequence diagram?

Either with an asterisk on the message label or a loop frame around the message(s).

15. What are the three types of frames used on a sequence diagram?

Loop frame – repeat or loop the contents of the frame

Opt frame – send or do not send the message based on true/false condition

Alt frame – if-else alternative flows based on condition

16. What is the symbol for a true/false condition on a sequence diagram?

A true/false condition is shown in brackets []

17. What are the parameters of a message?

The parameters represent the input data, i.e. the data that is being passed to the destination object.

18. List the primary steps for developing a SSD.

Ooops, duplicate question. See question 10.

19. What is an object state?

An object state is the state of being of an object, and is usually measured by a set of values. It is comparable to a status condition.

20. What is a state transition?

A state transition is the movement of an object from one state to another state.

21. When considering requirements, states and state transitions are important for understanding which other diagram?

States and Transitions are part of the state machine diagram for an object class. Therefore, they help to understand the objects in the class diagram.

22. What UML diagram is used to show the states and transitions for an object?

States and Transitions are part of the state machine diagram, which describes the activity of the objects in an object class.

23. List the elements that make up a transition description. Which elements are optional?

Transition-name (parameters, ...) [guard-condition] / action-expression

Any of the three elements may be empty, i.e. are optional.

24. What is a composite state? What is it used for?

A composite state is a high-level state in that it may have other states and transitions inside of it. It is used to represent concurrent states. For example, a printer may be in the “on” state, which is a composite state, and it may be “idle” which is an internal state within “on.”

25. What is meant by the term *path*?

A path is a sequence of states and transitions. It may be a complete path to describe an entire origin to destination path, or it may be a “snippet” of a path and only contain a few states and transitions.

26. What is the purpose of a guard-condition?

A guard-condition determines whether or not a transition can fire. Usually first the trigger fires to notify the transition that it should execute, but before it begins execution, it tests to see if the guard-condition is true.

27. Identify the models explained in this chapter and their relationship to one another.

The two main models for requirements are the Use case model, which identifies the use cases or the “processes,” and the Domain model class diagram, which identifies the information or data for the system.

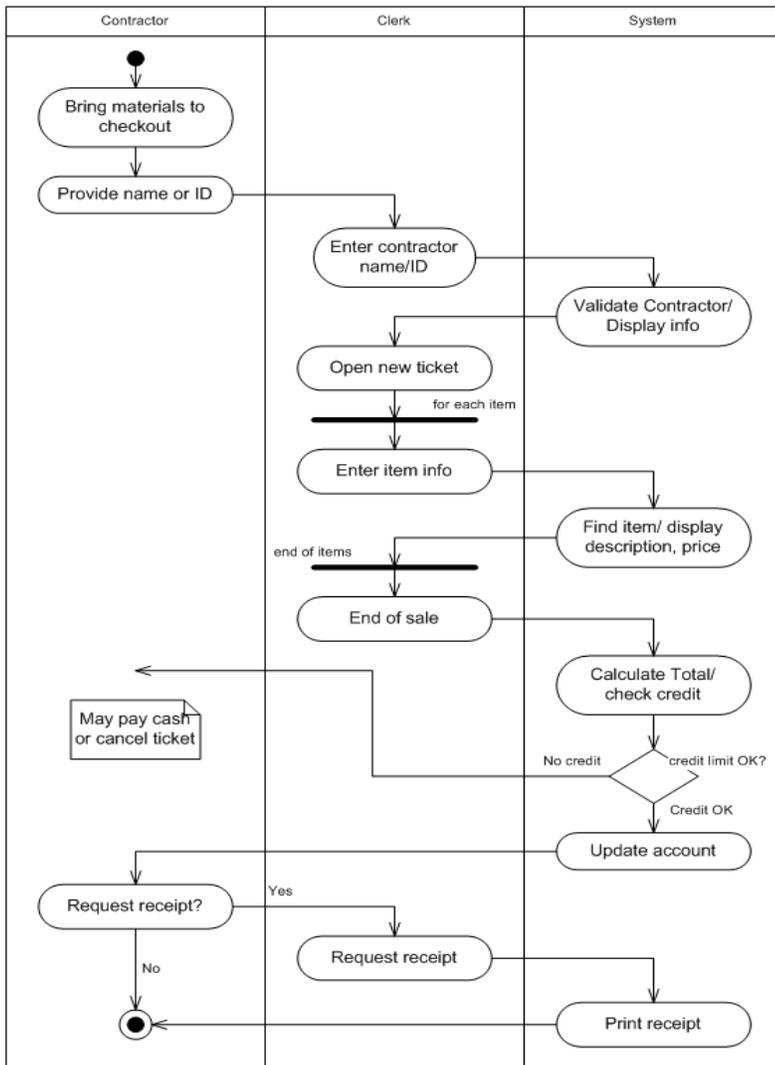
The Use case diagram is supported by Use Case Descriptions, Activity diagrams, and System Sequence diagrams. The Domain model class diagram is supported by State machine diagrams.

Problems and Exercises

1. After reading the following narrative, do the following:

- i. Develop an activity diagram for each scenario.
- ii. Complete a fully developed use case description for each scenario.

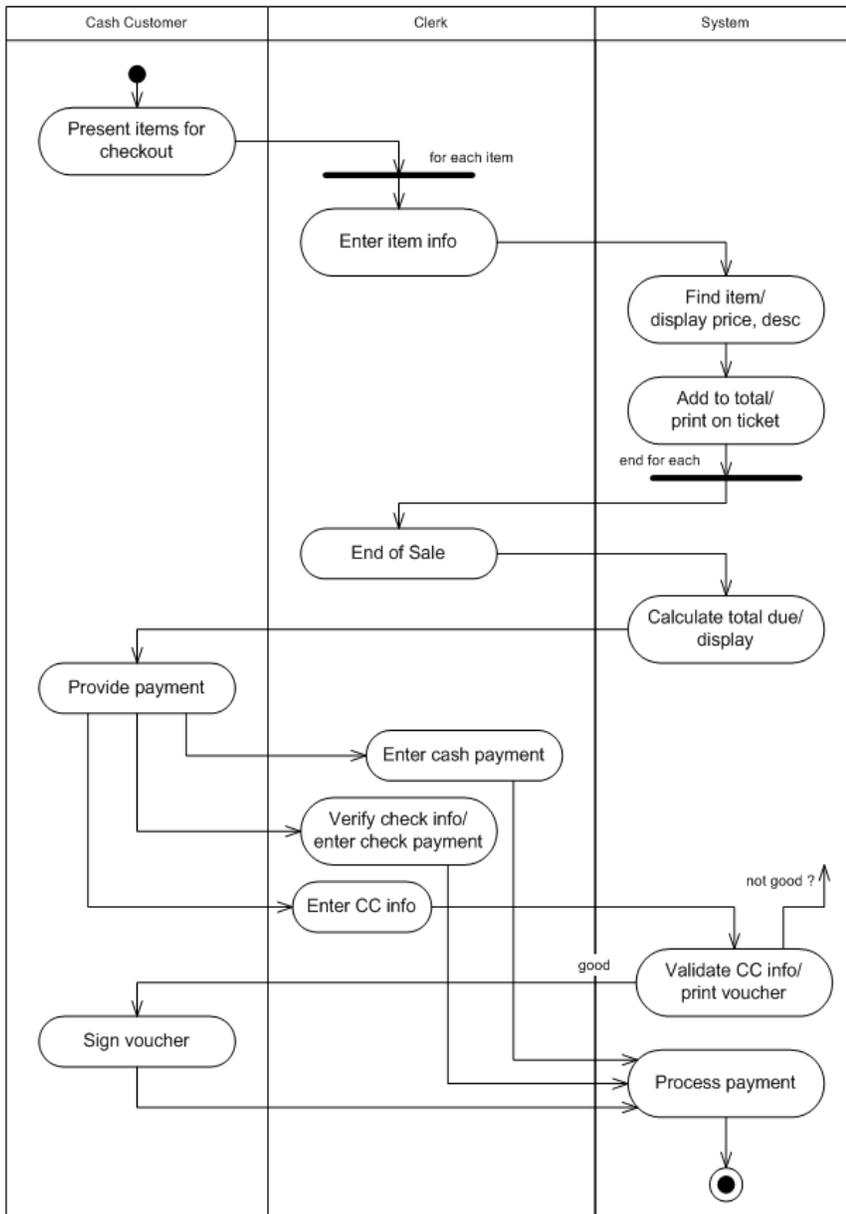
Contractor Sale activity diagram:



Contractor Sale fully developed use case description:

Use Case Name:	Create a new sale	
Scenario:	Create new sale to a contractor (on account sale)	
Triggering Event:	Contractor wants to purchase items.	
Brief Description:	A contractor wants to purchase items. The clerk rings up the items and then adds them to the contractor's account.	
Actors:	Sales clerk	
Stakeholders:	Sales clerk Accounting department Sales department	
Preconditions:	Customer account must exist. Inventory items must exist.	
Postconditions:	New sale is created. Sales line items are created and connected to the sale. Customer (contractor) account is updated.	
Flow of Activities:	Actor	System
	<ol style="list-style-type: none"> 1. Clerk enters contractor ID. 2. Clerk enters each item. 3. Clerk indicates the end of the sale. 4. If contractor wants receipt, requests receipt. 	<ol style="list-style-type: none"> 1.1 System validates contractor account. 2.1 System finds item in inventory, finds price, adds to total. 3.1 System calculates total and adds to contractor account. 4.1 System prints receipt.
Exception Conditions:	<ol style="list-style-type: none"> 1.1 If contractor account is out of balance, treat this sale as a cash sale, or stop process and send contractor to accounting clerk. 2.1 If system has information missing, sales clerk calls manager and manually enters information. 3.1 If contractor account balance is over the limit, treat as cash sale, cancel, or send contractor to accounting clerk. 	

Sale to public activity diagram:

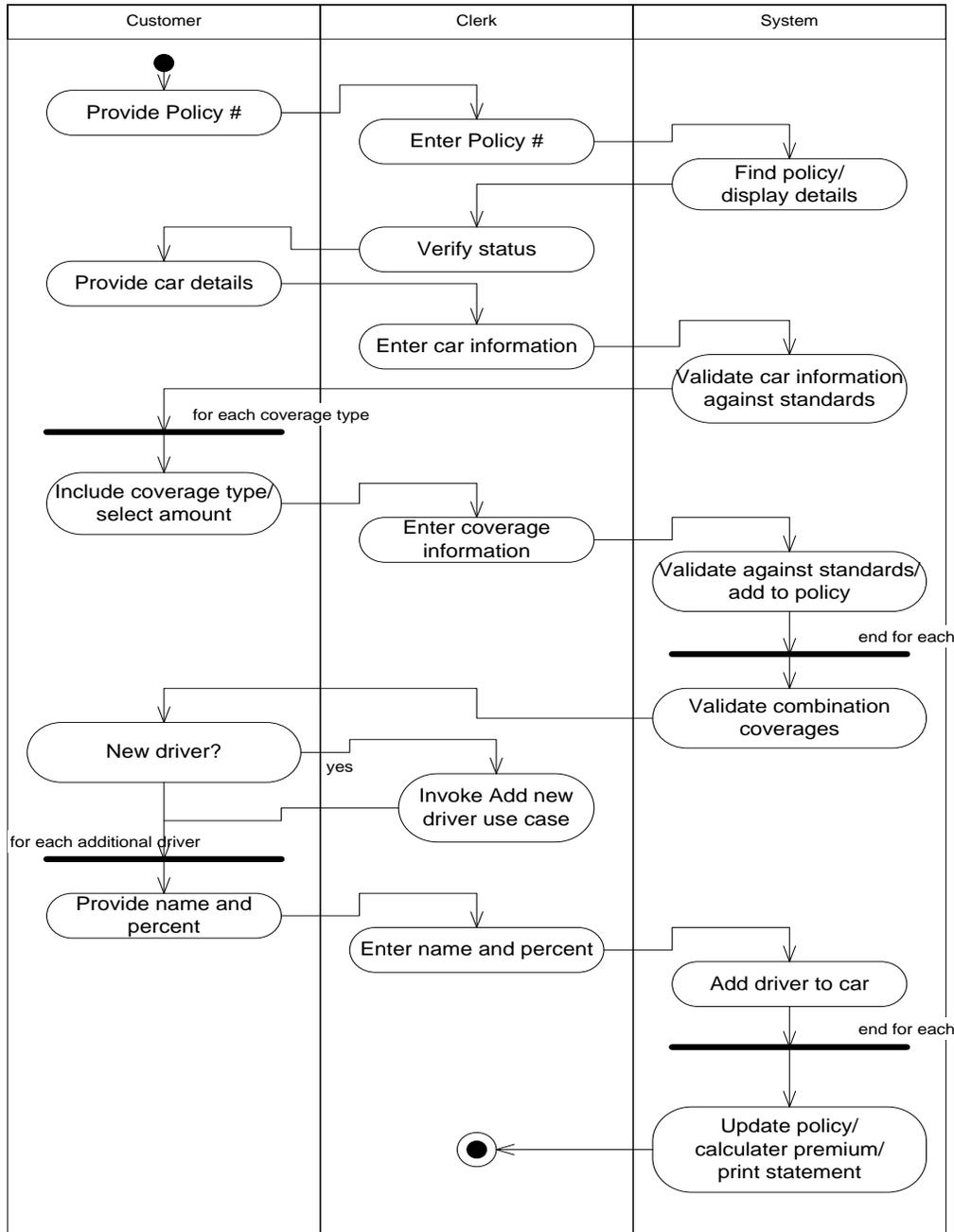


Sale to public fully developed use case description:

Use Case Name:	Create a new sale	
Scenario:	A new cash sale	
Triggering Event:	Cash customer wants to purchase items.	
Brief Description:	A cash customer wants to purchase items. The clerk enters the item ID, and the system creates a sales ticket. Customer pays with cash, check or credit card.	
Actors:	Sales clerk	
Stakeholders:	Sales clerk Accounting department Sales department	
Preconditions:	Inventory items must exist.	
Postconditions:	New sale is created. Sales line items are created and connected to the sale. Payment transaction is created.	
Flow of Activities:	Actor	System
	1. Clerk starts new cash sale. 2. Clerk enters each item. 3. Clerk indicates the end of the sale. 4. Clerk indicates type of payment and enters information.	2.1 System finds item in inventory, finds price, displays information, adds to total. 3.1 System calculates total. 4.1 System processes payment and creates payment transaction.
Exception Conditions:	2.1 If system has information missing, sales clerk calls manager and manually enters information. 4.1 If customer credit card fails approval, require cash or cancel sale.	

2. Based on the following narrative, develop either an activity diagram or a fully developed description for the use case of *Add a new vehicle to an existing policy* in a car insurance system.

Problem 7.



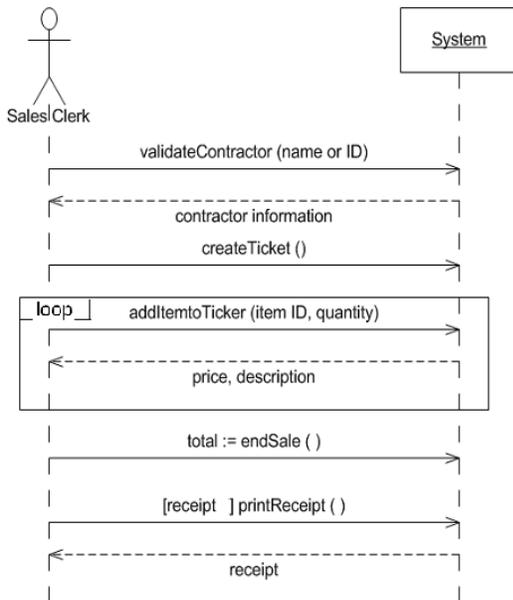
Use Case Name:	Add a new vehicle to an existing policy	
Scenario:	Telephone instance with customer and clerk	
Triggering Event:	Customer buys a new vehicle.	
Brief Description:	Customer provides car information, requests coverage with amounts, identifies drivers of the new car. System updates the policy.	
Actors:	Customer service clerk	
Stakeholders:	Customer Customer service department	
Preconditions:	Customer policy must exist and be up to date. StandardVehicle control tables for this vehicle type and year must exist. StandardCoverage tables exist.	
Postconditions:	New vehicle object created and connected to policy. Also connected to StandardVehicle. New coverage objects created and connected to vehicle. Also connected to StandardCoverage. New driver (InsuredPerson) (if necessary) created and added to policy. Existing drivers and percentages updated. Policy updated with new premiums.	
Flow of Activities:	Actor	System
	<ol style="list-style-type: none"> 1. Clerk enters customer information. 2. Clerk verifies policy is current. 3. Clerk enters car identification information. 4. Clerk enters each type of coverage customer requests, including deductibles and coverage amount. 5. Clerk indicates all coverages have been entered. 6. Clerk invokes <i>Add new person</i> use case if necessary. 7. Clerk changes driver percentages on this car and other cars. 8. Clerk indicates everything is complete. 	<ol style="list-style-type: none"> 1.1 System finds policy and displays details. 3.1 System validates that car has known standard. 4.1 System validates coverage requests. 5.1 System does combination validation on policy. 7.1 System updates driver information. 8.1 System updates policy, calculates new premium, prints new statement.
Exception Conditions:	<ol style="list-style-type: none"> 2.1 If policy is not current, clerk requests payment or collects necessary information. 3.1 If car type is not in system, clerk refers customer to underwriting to handle this situation. 4.1 If coverage requests are out of range, clerk asks customer for changed amount. 5.1 If some combination is invalid, return to step 4. 	

3. Given the following list of classes and associations for the previous car insurance system, list the preconditions and postconditions for the use case *Add a new vehicle* to an existing policy.

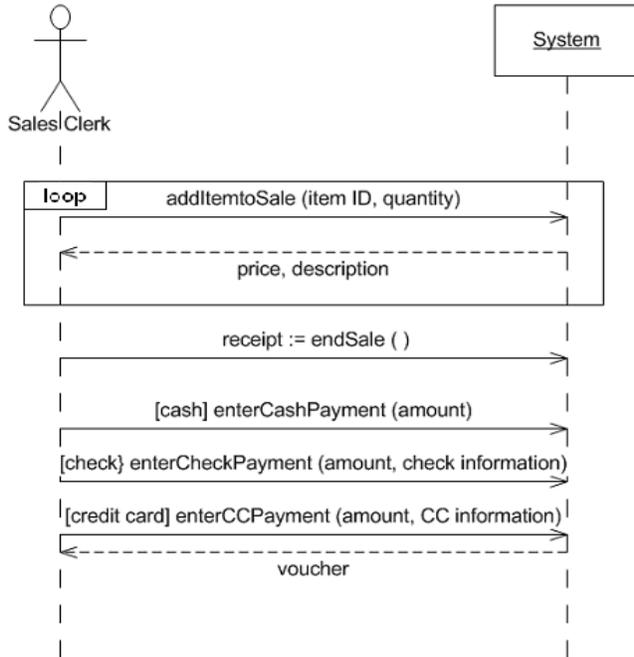
Preconditions:	Customer policy must exist and be up to date. StandardVehicle control tables for this vehicle type and year must exist. StandardCoverage tables exist.
Postconditions:	New vehicle object created and connected to policy. Also connected to StandardVehicle. New coverage objects created and connected to vehicle. Also connected to StandardCoverage. New driver (InsuredPerson) (if necessary) created and added to policy. Existing drivers and percentages updated. Policy updated with new premiums.

4. Develop an SSD based on the narrative and your activity diagram for problem 1.

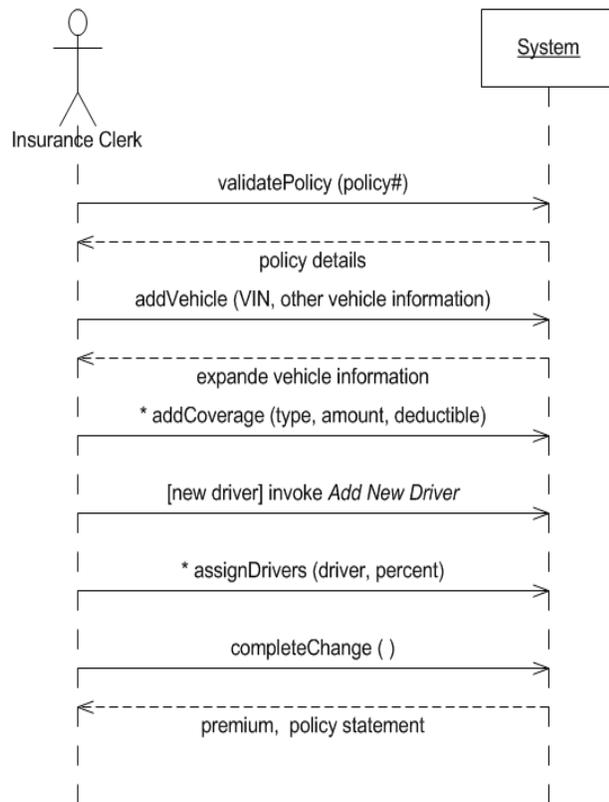
Contractor Sale:



Public sale:



5. Develop a n SSD based on the narrative or your activity diagram for problem 2.



6. Review the cellular telephone state machine diagram shown in Figure 5-21 and then answer the following questions. (Note that this telephone has characteristics not found in ordinary telephones. Base your answers only on the state machine diagram.)

i. What happens to turn on the telephone?

A person has to switch it on with some external event.

ii. What states does the telephone go into when it is turned on?

It goes into the composite state (unlabeled, but should have a label of ON). It goes into the *Quiet* state for one path. It can also go into *Charged*, *Low Warning*, or *Discharged*.

iii. What are the three ways the telephone can be turned off?

SwitchOff, *Quiet* and *Low Warning*, or *Discharged*.

iv. Can the telephone turn off in the middle of the Active (Talking) state?

Yes, if it is discharged, it leaves the *Discharged* nested state and the *On* composite state.

v. How can the telephone get to the Active (Talking) state?

Through the origin state of connecting or by answering from the origin state of ringing.

vi. Can the telephone be plugged in while someone is talking?

No. The state chart says it can only be plugged in from the *Quiet* state.

vii. Can the telephone change battery states while someone is talking? Explain which movement is allowed and which isn't allowed.

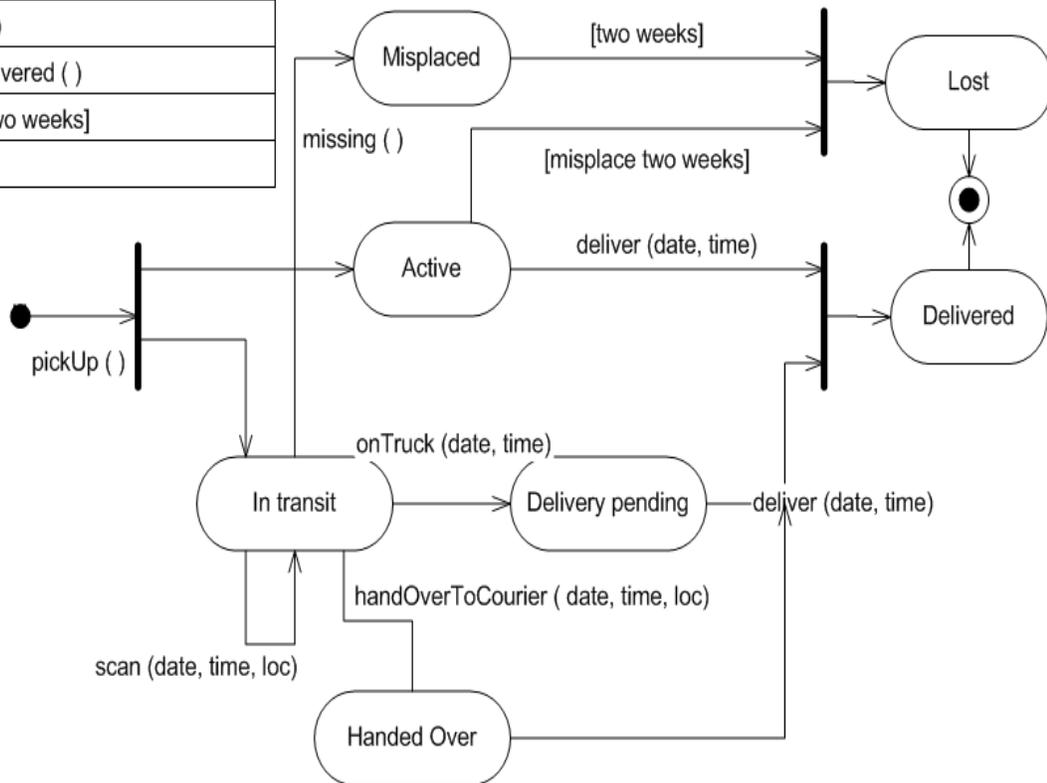
It can go from *Charged* to *Low Warning* and from *Low Warning* to *Discharged*. However, because it cannot be plugged in while someone is talking, it cannot move back up from *Discharged* to *Low Warning* or *Charged*.

viii. What states are concurrent with what other states? Make a two-column table showing the concurrent states.

Concurrent States	
Quiet, Dialing, Connecting, Ringing, Active, and Plugged In	Charged, Low Warning, Discharged

7. Based on the following description of a shipment made by Union Parcel Shipments, identify all the states and exit transitions and then develop a state machine diagram.

STATE — EXIT TRANSITION
active — delivered (), [misplaced two weeks]
in transit — onTruck ()
delivery pending — delivered ()
delivered — final ()
handedOver — delivered ()
misplaced — [for two weeks]
lost — final ()



8. Locate a company in your area that develops software. Consulting companies or companies with a large staff of information systems professionals tend to be more rigorous in their approach to systems development. Set up an interview. Determine the development approaches that the company uses. Many companies still use traditional structured techniques combined with some object-oriented development. In other companies, some projects are structured, whereas other projects are object oriented. Find out what kinds of modeling the company does for requirements specification. Compare your findings with the techniques taught in this chapter.

Answers will vary.

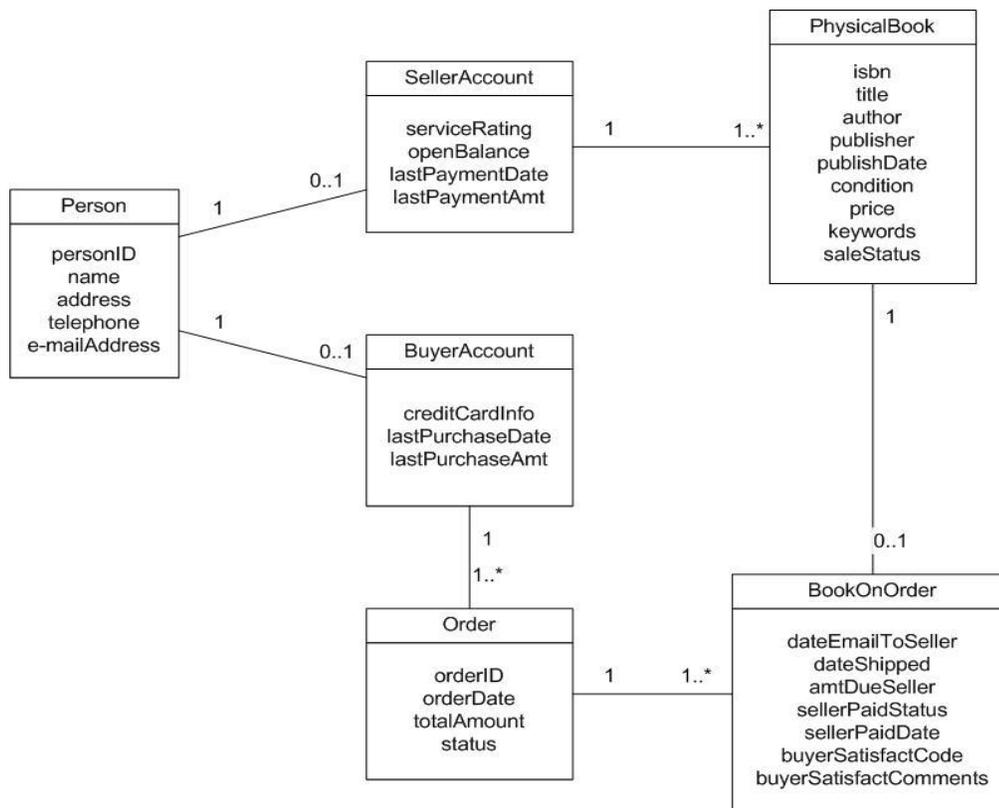
Solutions to End-of-Chapter Cases

Case Study: *TheEyesHaveIt.com Book Exchange*

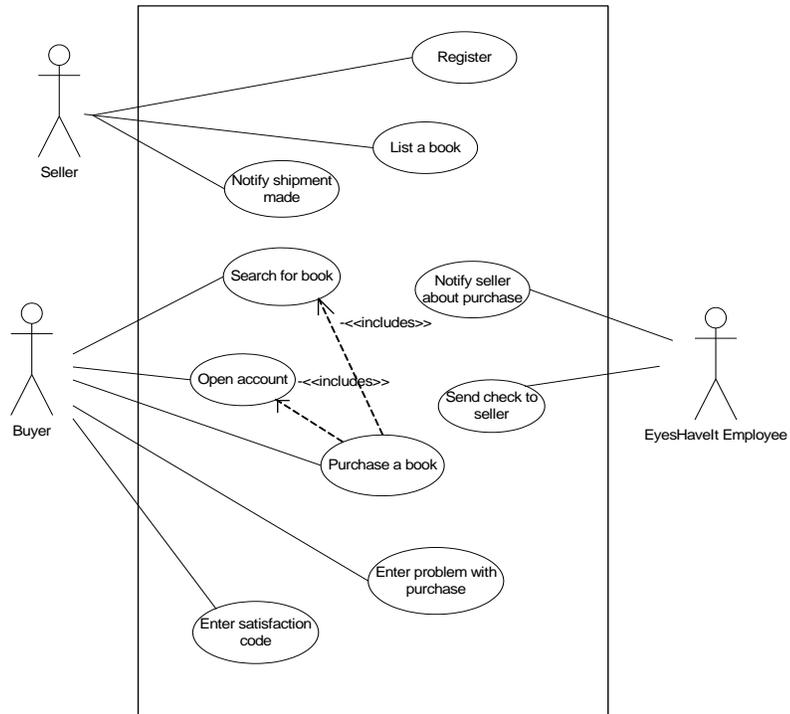
For this case, develop these diagrams:

1. A domain model class diagram
2. A list of uses cases and a use case diagram
3. A fully developed description for two use cases: *Add a seller* and *Record a book order*
4. An SSD for each of the two use cases in question 3

1. A domain model class diagram



2. A list of uses cases and a use case diagram



3. A fully developed description for two use cases: *Add a seller* and *Record a book order*

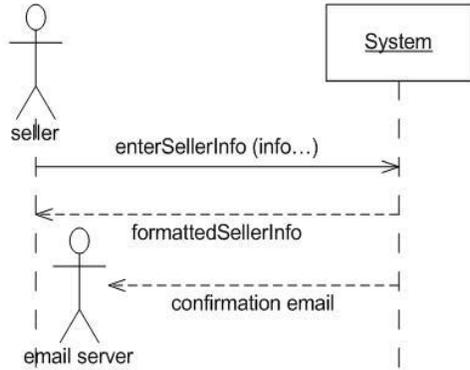
Use Case Name:	Register	
Scenario:	Register/add a new seller	
Triggering Event:	A new seller wants to sell books.	
Brief Description:	Seller decides he/she would like to list a book(s). Seller registers and receives a confirmation e-mail.	
Actors:	Seller E-mail server	
Stakeholders:		
Preconditions:	Seller must not exist in the system. Seller must have all information necessary to register.	
Postconditions:	Seller has an account to list books.	
Flow of Activities:	Actor	System
	1. Seller connects to EyesHaveIt.com and fills out registration form. 2. Seller submits registration form.	2.1 System notifies seller a confirmation e-mail will be sent. 2.2 System e-mails confirmation of registration to seller.
Exception Conditions:	1.1 If the seller already exists in the system, the system sends prior login and password to e-mail address. 2.1 If the seller was removed from the system for bad transactions/credit, the system sends the seller an e-mail notifying the seller of the situation and no account is created.	

Use Case Name:	Purchase a book
Scenario:	Purchase a book
Triggering Event:	A buyer decides to purchase a book from EyesHaveIt.com.
Brief Description:	Customer searches for a book(s) on EyesHaveIt.com. Customer selects from search results and adds a book(s) to the shopping cart. Customer then proceeds to checkout. If an account exists, the customer confirms purchase, and the system sends a confirmation e-mail to the customer. If an account doesn't exist, an account is created, the purchase is confirmed, and the confirmation e-mail is sent.
Actors:	Buyer E-mail server Seller
Stakeholders:	

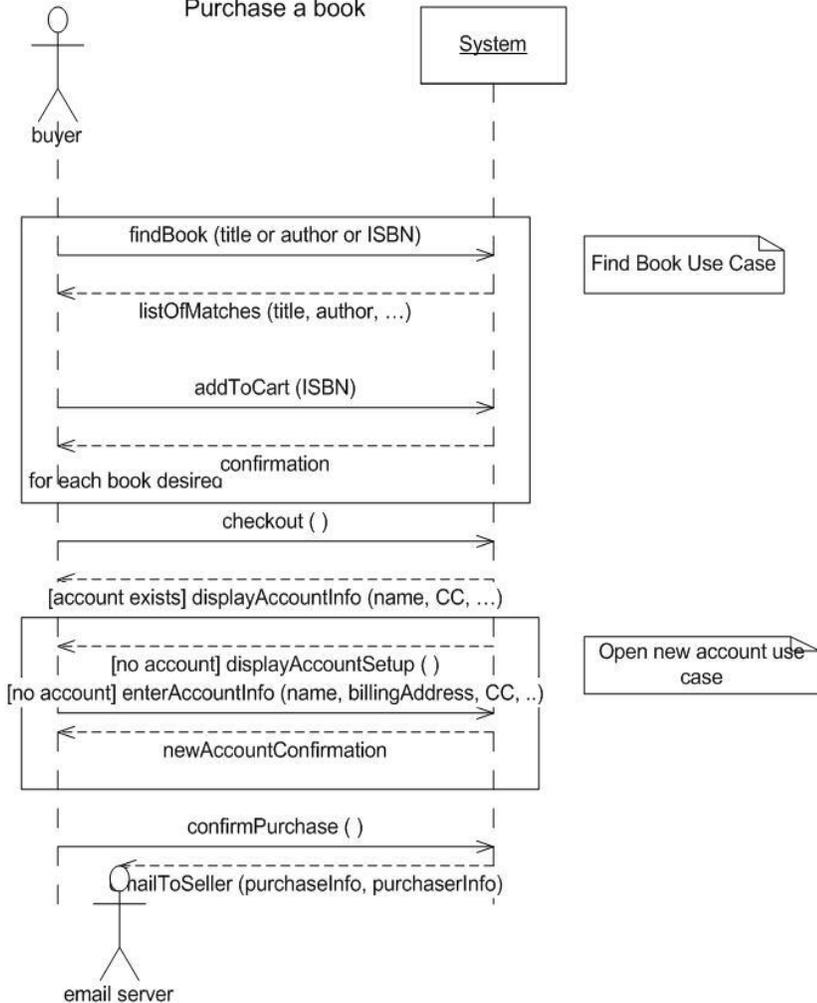
Preconditions:	Books and book information must exist in the system.	
Postconditions:	Customer account must exist. Order must be placed. E-mail must be sent to seller.	
Flow of Activities:	Actor	System
	<p>1. Buyer searches EyesHaveIt.com for a book(s).</p> <p>2. Buyer selects a book(s) to purchase from search results.</p> <p>3. Repeat steps 1 and 2 until all desired books are added to shopping cart.</p> <p>4. Customer proceeds to checkout by selecting the Checkout button.</p> <p>4a. If customer wishes to remove an item, he/she selects the item to be removed and then selects the Delete button.</p> <p>4b. If buyer wishes to add an item, he/she selects the Continue Shopping button and proceeds to steps 1 and 2.</p> <p>5. Buyer verifies displayed information.</p> <p>6. If information is incorrect or buyer account does not exist, buyer updates displayed information or enters new information into the registration form.</p> <p>7. Buyer confirms purchase.</p>	<p>1.1 Searches for all matches related to buyer's search criteria.</p> <p>2.1 Creates shopping cart. Adds selected item(s) to the shopping cart.</p> <p>4.1 Displays list of shopping cart items for verification.</p> <p>4a.1 Displays list of shopping cart items with deleted items removed.</p> <p>5. Displays buyer information.</p> <p>6.1 Updates new buyer information for existing account. 6.2 Creates account for new buyer and sends confirmation.</p> <p>7.1 Records order. 7.2 Sends e-mail to seller.</p>
Exception Conditions:	<p>1.1 If book is sold out, buyer cannot add book to shopping cart.</p> <p>2.1 If buyer account does not exist, a new account must be created. If buyer is rejected for new account based on credit, system sends buyer notification.</p> <p>3.1 If payment is rejected, system notifies buyer and seller.</p>	

4. An SSD for each of the two use cases in question 3

Register Seller



Purchase a book

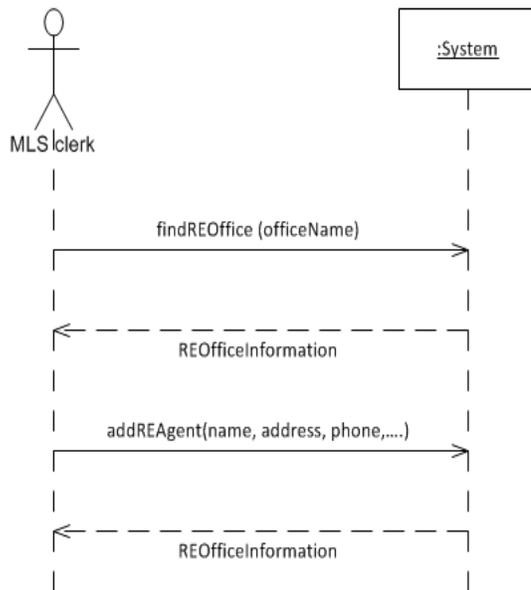


Running Cases: Community Board of Realtors

The Multiple Listing Service system has a number of use cases, which you identified in Chapter 3, and three key domain classes, which you identified in Chapter 4: RealEstateOffice, Agent, and Listing.

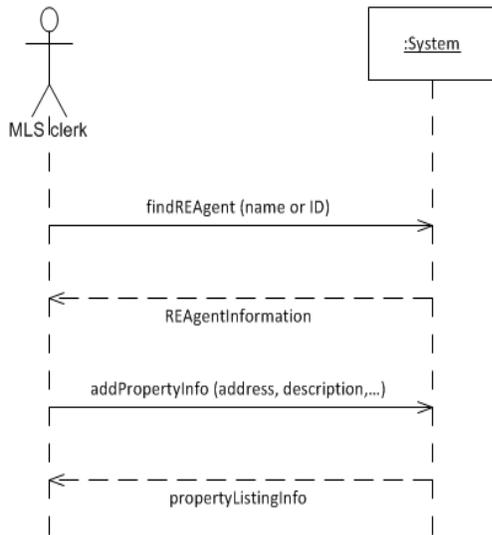
1. For the use case *Add agent to real estate office*, write a fully developed use case description and draw an SSD. Review the case materials in previous chapters and recall that the system will need to know which real estate office the agent works for before prompting for agent information.

Use case name:	Add agent to real estate office	
Scenario:	MLS clerk adding agent	
Triggering event:	New agent hired in a real estate office	
Brief description:	The correct real estate office is identified, and the new real estate agent information is entered into the system.	
Actors:	MLS clerk	
Related use cases:	Real estate office adds new agent (Web based version scenario)	
Stakeholders:	Real estate office, real estate agent	
Preconditions:	The real estate office must exist	
Postconditions:	Real estate agent is created and associated with real estate office	
Flow of activities	Actor	System
	1. Find correct real estate office 2. Enter new agent information	1.1 Display real estate office information 2.1 Create new agent record, including with relationship to real estate office
Exception conditions:	1.1 No real estate office found for requested id/name. Display not found message.	



2. For the use case *Create new listing*, write a fully developed use case description and draw an SSD. Recall that the system needs to know which agent made the listing before the system prompts for listing information.

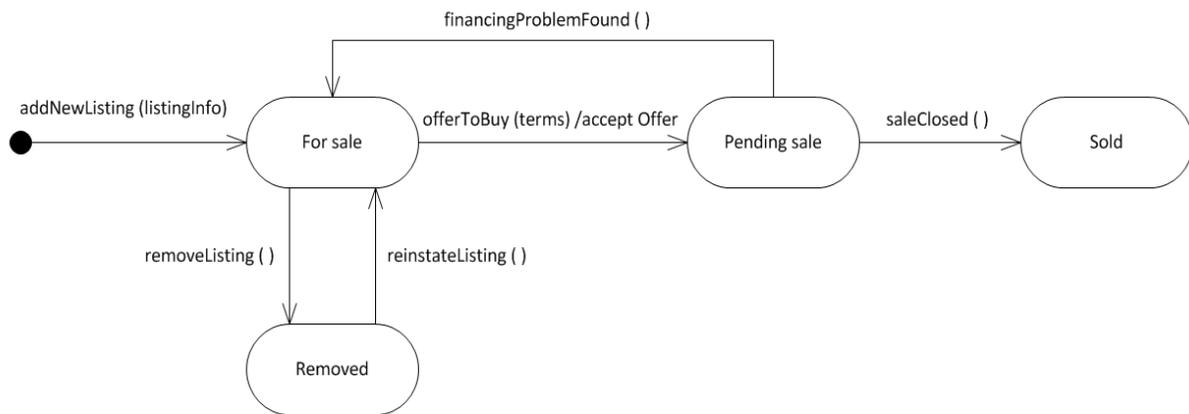
Use case name:	Create a new listing	
Scenario:	MLS clerk creates a new listing	
Triggering event:	New property is put up for sale	
Brief description:	The listing agent is identified and verified. The new property information is entered into the system, along with images etc.	
Actors:	MLS clerk	
Related use cases:	Real estate office/agent creates a new listing (Web version scenario)	
Stakeholders:	Real estate office, Real estate agent, Property owner	
Preconditions:	Real estate office must exist Real estate agent must exist	
Postconditions:	New listing must be created and associated with RE office and RE agent	
Flow of activities	Actor	System
	1. Find real estate agent 2. Enter new listing information	1.1 Display agent and office information 2. Create new property listing record, associated with agent. Display results.
Exception conditions:	1.1 Agent information not found. Display not found message.	



3. Draw a state machine diagram showing the states and transitions for a Listing object.

Note: Answers will vary because students will have to brainstorm valid states for a Listing object. In our solution we will use the following states and exit transitions. (The term “Sale” is used to mean either “Sale” or “Lease”)

State	Exit transition
For sale	Offer to buy
Sale pending	Close the sale
Sold	
Removed	Reinstate listing



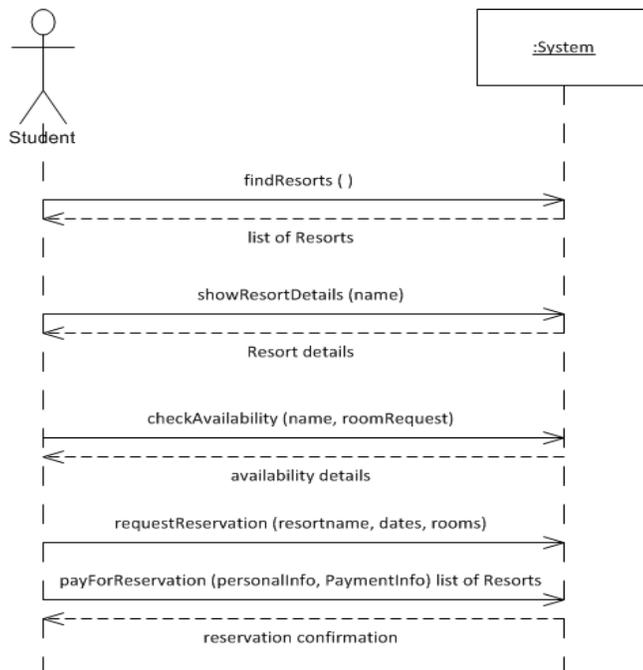
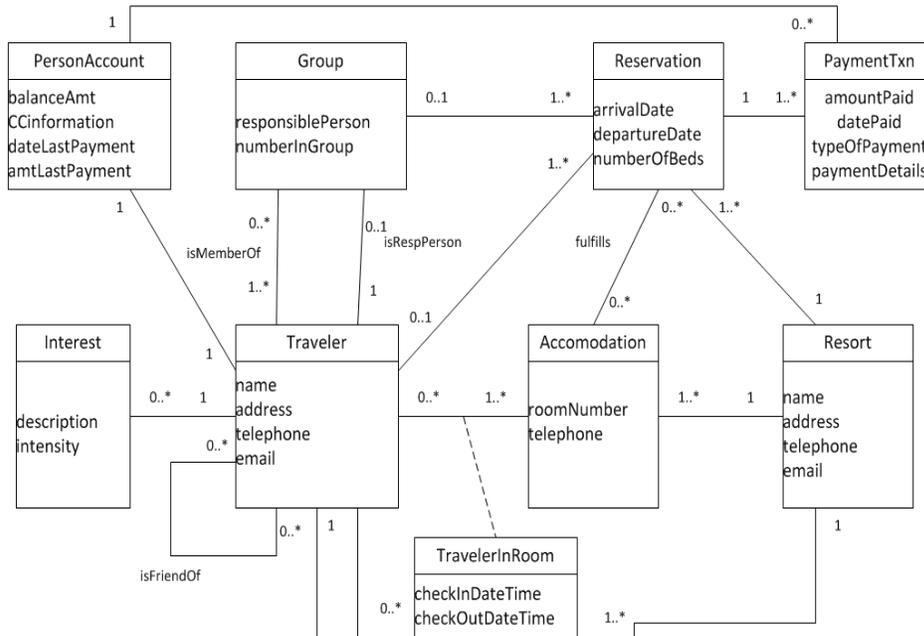
Running Cases: The Spring Breaks 'R' Us Travel Service

The Spring Breaks 'R' Us Travel Service system has many use cases and domain classes, which you identified in Chapters 3 and 4. Review the domain model class diagram to get a feel for the complexity of some of the use cases.

1. For the use case *Book a reservation*, write a fully developed use case description and draw an SSD. Review the classes that are associated with a reservation in the domain model to understand the flow of activities and repetition involved.

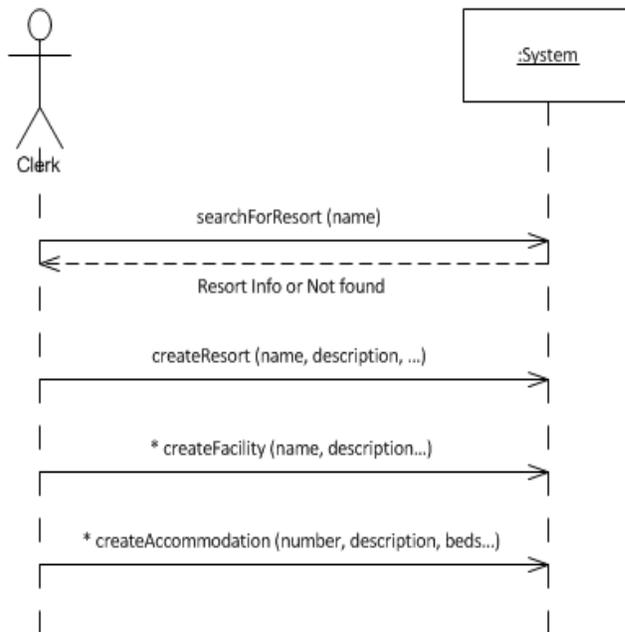
Use case name:	Book a reservation	
Scenario:	Book a reservation online	
Triggering event:	Student wants to make a reservation and initiates booking	
Brief description:	Student searches or browses the resorts. He/she checks accommodations and availability. Then he/she makes a reservation for either a single person or a group. (Allow both individual and group reservations.)	
Actors:	Student	
Related use cases:	Create individual account (includes Traveler) Create group account Add person to group (new use case previously undefined)	
Stakeholders:	Student, Resort	
Preconditions:	Traveler and Individual account must exist Group must exist (for group reservation) Resort must exist	
Postconditions:	Reservation must be created and associated with Resort and Group/Traveler Payment must be created and associated with IndividualAccount	
Flow of activities	Actor	System
	1. Find a resort (search or browse) 2. Check availability of accommodations 3. Choose reservation type 4. Enter reservation details 5. Enter reservation payment information	1.1 Display resort and accommodation information 2.1 Display accommodation availability information 4.1 Make reservation 5.1 Verify individualInfo and paymentInfo Create PaymentTransaction for Reservation Display confirmation Send email confirmation
Exception conditions:	5.1 Payment transaction fails	

Note: Based on the requirements of this use case, and the previously defined use cases, the partial domain model class diagram created in Chapter 4 needs to be enhanced with several more classes: Group, Reservation, PaymentTxn, PersonalAccount. The class diagram for Chapter 4 only focused on the social networking part. See the partial class diagram below.



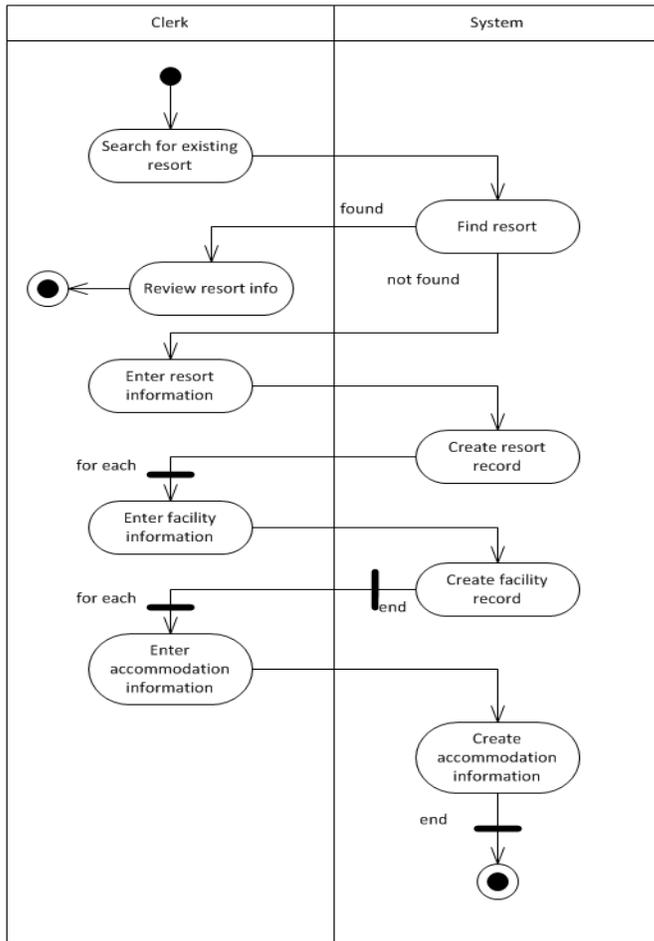
2. For the use case *Add new resort*, write a fully developed use case description and draw an SSD. Review the classes that are associated with a resort in the domain model to understand the flow of activities and repetition involved.

Use case name:	Add a new resort	
Scenario:	Add a new resort	
Triggering event:	A new resort contracts with SBRU to participate in the vacation program	
Brief description:	A new resort is added with descriptive information. Information about the accommodations available to this program are entered. Information about the facilities available for activities in this program are entered	
Actors:	SBRU clerk, Resort employee	
Related use cases:		
Stakeholders:	SBRU management, Resort management	
Preconditions:	Resort must not already exist	
Postconditions:	Resort is created Facilities are created and associated with the resort Accommodations are created for this resort	
Flow of activities	Actor	System
	1. Verify that the resort does not exist 2. Enter resort description 3. (loop) Enter facilities information 4. (loop) Enter accommodations information	1.1 Check database for resort information 2.1 Create resort record 3.1 Create facilities record 4.1. Create accommodations record
Exception conditions:	1.1 Resort already exists	



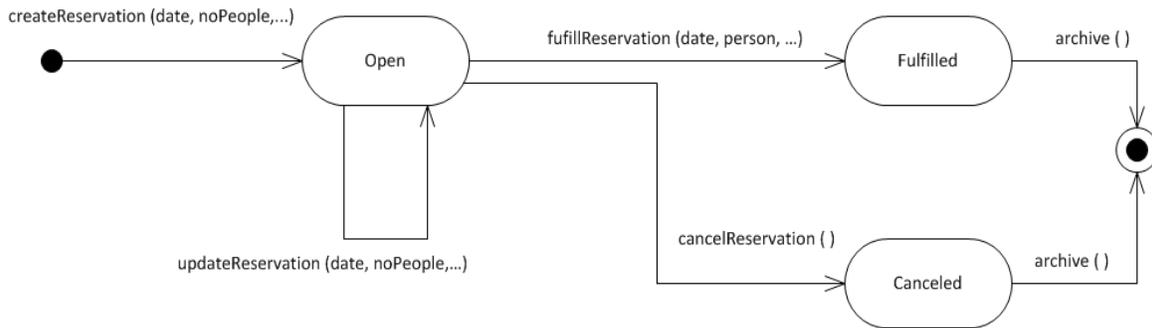
3. Draw an activity diagram to show the flow of activities for the use case *Add a new resort*.

Note: This activity diagram is based on the updated problem domain model.



4. Draw a state machine diagram showing the state and transitions for a Reservation object.

State	Exit transition
Open	Fulfill reservation, cancel reservation
Fulfilled	
Canceled	



Running Cases: On the Spot Courier Services

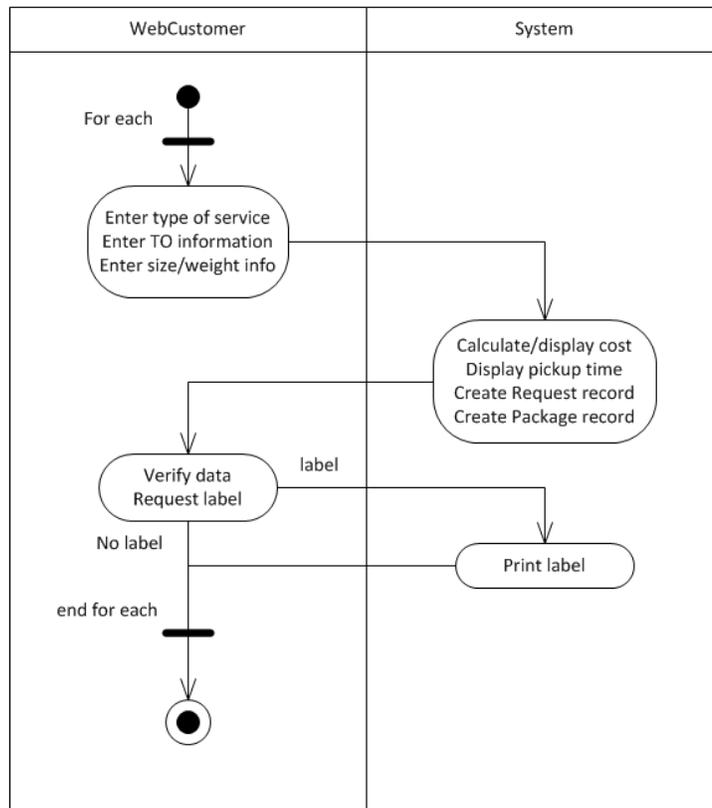
1. Based on this description, develop the following for the use case *Request a package pickup* and for the Web customer scenario:

- i. A fully developed use case description
- ii. An activity diagram
- iii. An SSD

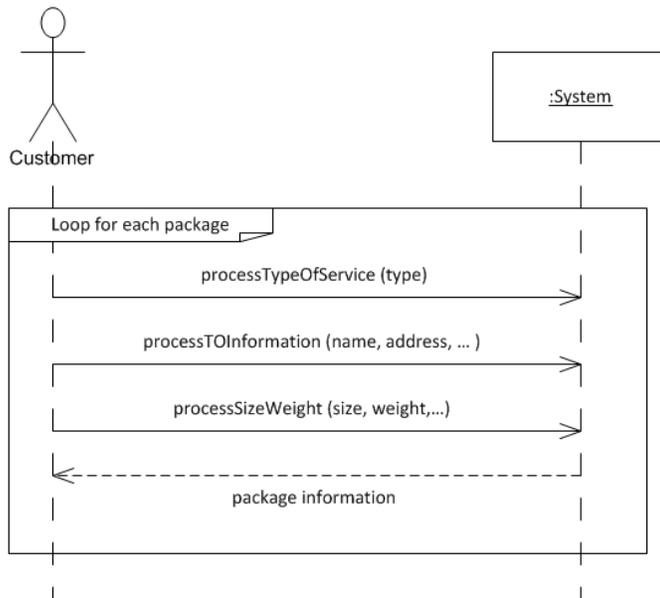
A fully developed use case description

Use case name:	Request a package pickup	
Scenario:	Web customer requests package pickup	
Triggering event:	Web customer has package(s) to be picked up and requests pickup	
Brief description:	User enters package information (TO address, type of service, size/weight). System returns the cost, expected pickup time, and prints label	
Actors:	Web customer	
Related use cases:	Request package pickup (phone in) Enter package info (pickup package)	
Stakeholders:	Bill, Customer, Delivery employee	
Preconditions:	Customer and customer account must exist	
Postconditions:	Pickup request is created and associated with Customer	
Flow of activities	Actor	System
	1. For each package 1.1 Enter type of service 1.2 Enter TO information 1.3 Enter package size/weight 1.4 Request label print	1.3.1 Display Cost 1.3.2 Display expected pickup time 1.3.3 Create Request record Create Package record 1.4.1 Print label
Exception conditions:	1.4.1 Label cannot print	

An activity diagram



System sequence diagram



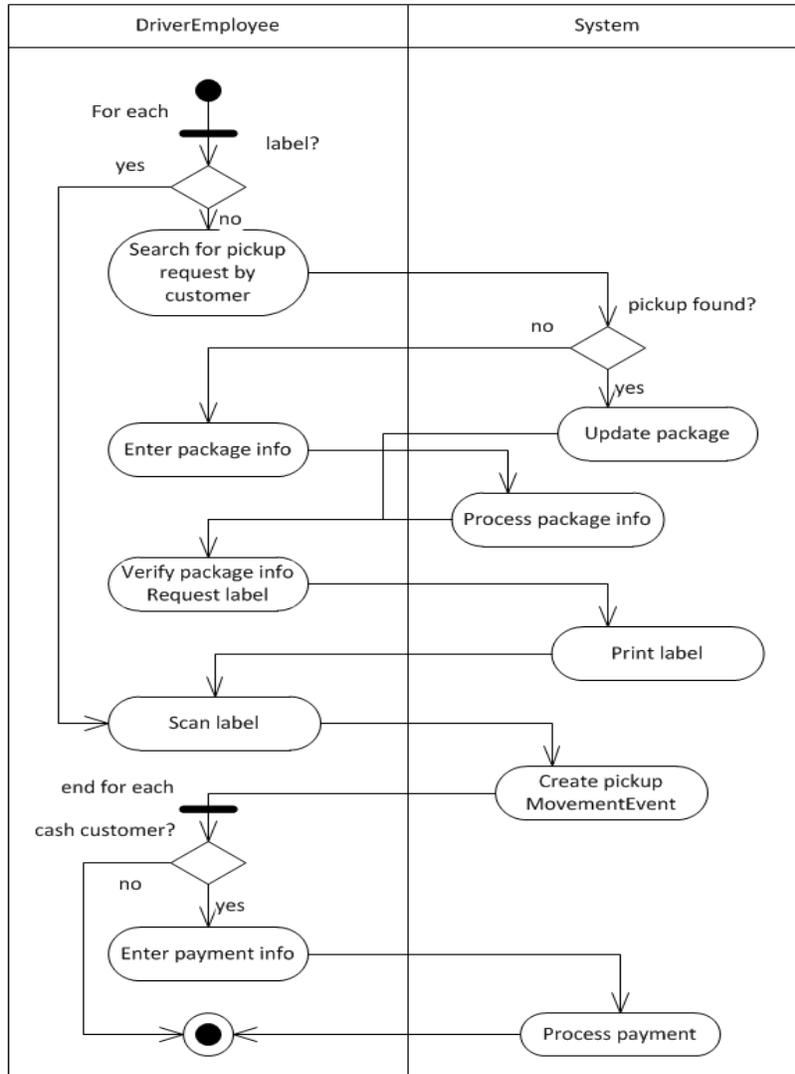
2. Based on the same description, develop the following for the use case *Pickup a package*:
- i. A fully developed use case description
 - ii. An activity diagram
 - iii. System sequence diagram

A fully developed use case description

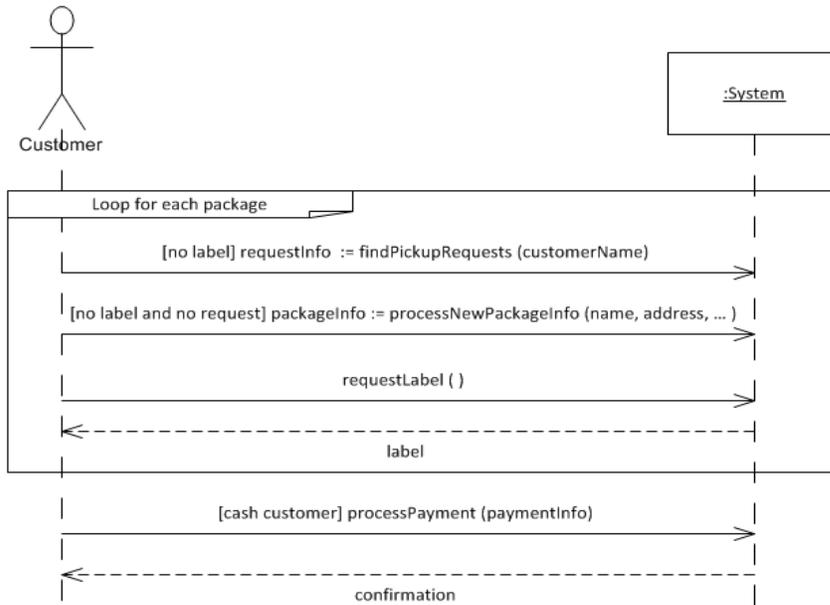
Use case name:	Pickup a package (Enter package pickup information)	
Scenario:	Delivery employee picks up package	
Triggering event:	Delivery employee arrives at customer's location and picks up a package	
Brief description:	Delivery employee verifies package with pickup request information, OR enters new package information. If cash customer, process payment.	
Actors:	Driver	
Related use cases:	Create customer and customer account Accept payment Scan package (movement)	
Stakeholders:	Customer, Driver, Bill	
Preconditions:	Customer should exist (else invoke Create customer use case)	
Postconditions:	Create package record and connect with Customer, PickupRequest, Movement event	
Flow of activities	Actor	System
	1. If package has NO label, search for request by Customer name	1.1 Access pickup info, update package

	<ul style="list-style-type: none"> 2. If pickup request not found, enter new package info 3. Request print label 4. Scan package label 5. If cash customer, enter payment info 	<p>If no pickup info, display none</p> <ul style="list-style-type: none"> 2.1 Process new package info 3.1 Print label 4.1 Update pickup request, package Create pickup MovementEvent 5.1 Process payment
Exception conditions:	5.1 Invalid payment data	

An activity diagram

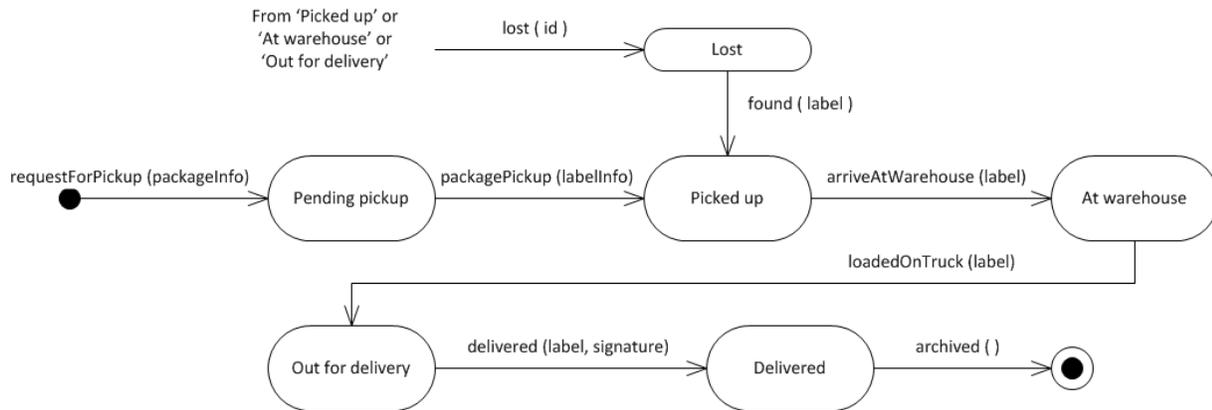


A system sequence diagram.



3. Develop a state machine diagram describing all the possible status conditions for a Package object.

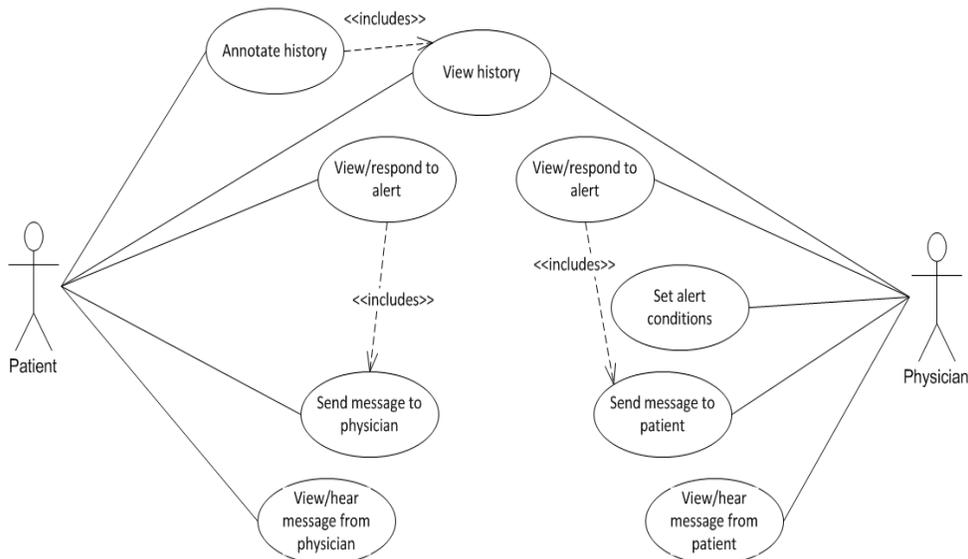
State	Exit transition
Pending pickup	packagePickup ()
Picked up	arriveAtWarehouse ()
At warehouse	loadedOnDeliveryTruck ()
Out for delivery	delivered ()
Delivered	none
Lost	found ()



Running Cases: Sandia Medical Devices

Figure 5-22 shows a set of use cases for the patient and physician actors. Answer the following questions and/or complete the following exercises:

1. Which use cases include which other use cases? Modify the diagram to incorporate included relationships.



2. Consider the use cases *View/respond to alert* and *View history*. Both actors share the latter, but each has a different version of the former. Why do the actors have different versions of the

View/respond to alert use case? Would the diagram be incorrect if each actor had his own version of the View history use case? Why or why not?

View/respond to alert: Even though the names are the same, the detail steps might be very different. It would be possible to make them different scenarios of the same use case. But making them separate use cases also works since the actors are different, the steps are different, and the system responses are different.

View history: View history does the same activities no matter who the actor is. In this case it would not make sense to have separate use cases since the processing steps, the data, the activities are both the same.

3. Develop an SSD for the View history use case. Assume that the system will automatically display the most recent glucose level, which is updated at five-minute intervals by default. Assume further that the user can ask the system to view glucose levels during a user-specified time period and that the levels can be displayed in tabular form or as a graph.

