Bradshaw Marina Case Study

When a business determines it needs a computer system, it works with a team of developers to design and develop the system.

- One of the first tasks for the development team is to analyze the business and identify the functions the system will perform.
- Next, the development team begins object-oriented analysis to identify the use cases and scenarios required and creating use case diagrams.
- Then the development team identifies required problem domain classes and creates the class diagram.
- Finally, the team develops sequence diagrams to model object interaction in the Bradshaw Marina case study; you are a member of the development team.

Exploring the Background of Bradshaw Marina

- Bradshaw Marina is a privately owned corporation that rents boat slips and provides boat services on Clinton Lake, a large inland lake located in the Midwestern United States. The lake was constructed in the 1970s primarily to provide flood control and generate limited amounts of electrical power. The U.S. Army Corps of Engineers manages the lake and restricts construction near its shores, creating an ideal natural wildlife habitat in addition to providing a beautiful park-like setting for boaters. Bradshaw Marina is the largest of the three marinas on the lake. The three marinas accommodate approximately 600 boats in slips: 450 sailboats and 150 powerboats. Bradshaw's boat population is around 350 sailboats and 75 powerboats, although it has plans to expand these capacities.

- Bradshaw Marina would like to have an **automated system to track its customers, the slips it leases, and the boats in the slips**. Initially, the system will simply maintain basic information for customers, slips, and boats, and perform basic day-to-day business tasks. These tasks include **creating a lease, computing the lease amount for a slip, and assigning a boat to a slip**. The marina also wants to use the system to search for information, such as vacant slips and slips leased to a specific customer.

- Bradshaw eventually wants to enhance the system so it can add boat service records, which will help track tasks such as **handling the boat**, **painting the bottom**, or **working on the engine**. Later, it will want to add **billing features** to the system and be able to use the system to generate **bills for both slip leases and boat services**, **record payments**, **send late notices**, and **produce accounts receivable** and **other accounting reports**. For now, Bradshaw wants the new system to include information on customers, slip, and boats.

Identifying Bradshaw Use Cases and Scenarios

The first step in the OOA process is **identifying use cases** that fall within the scope of the system. The main events of interest involve customers—**when a customer leases a slip**, **when a**

customer buys a new boat, and so on. Because these events involve customers, boats, and slips, the use cases also focus on customers, boats, and slips. Your development team initially spends a lot of time talking with Bradshaw staff about the events involving customers that result in use cases. Figure 1 illustrates the Bradshaw Marina use case diagram.

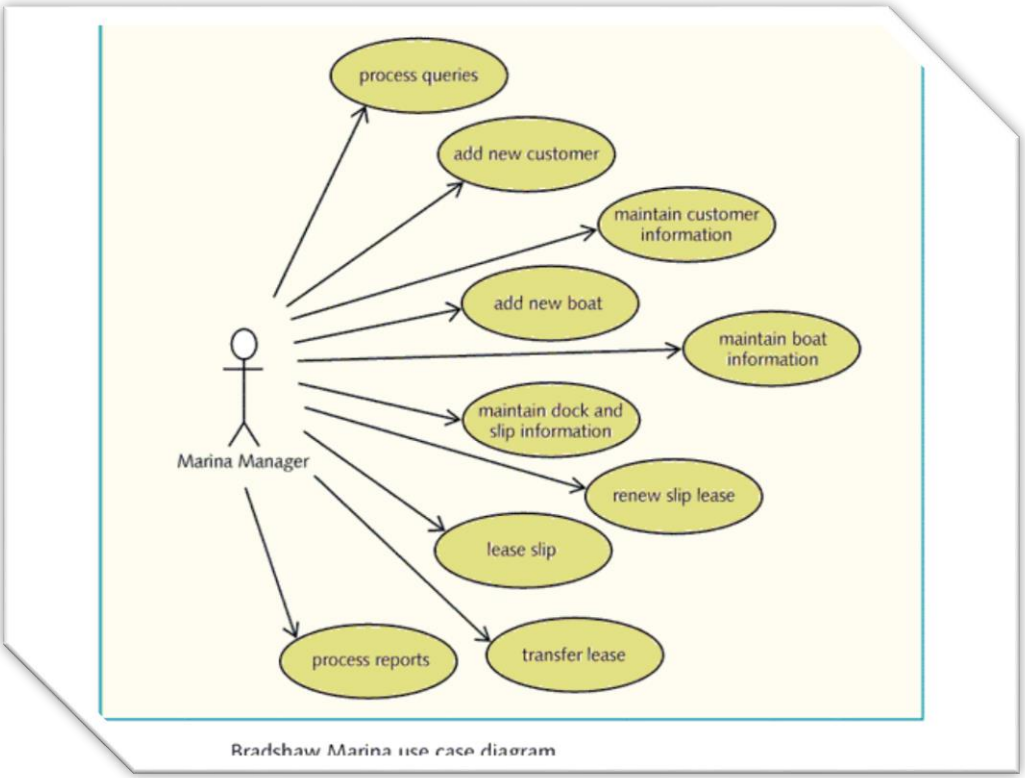


Figure 1: Bradshaw Marina use case diagram

For example, use cases involving customers might include add new customer and maintain customer information. A new customer is added when he or she leases a slip, and customer information is maintained whenever a customer changes address or phone number. Similarly, use cases involving leases include lease slip, renew slip lease, and transfer lease. Use cases involving boats include add new boat and maintain boat information. In addition, the system should maintain information about slips and the docks that contain slips. Finally, the system will need to process queries and reports. You work with other members of the development team to create the use case diagram indicating these use cases, which is shown in Figure 1.

Several scenarios could be associated with each use case, so you and your team might decide to **divide up the list of use cases and work separately on scenarios**. For example, the lease slip use case might have many scenarios you will want to discuss further with Bradshaw Marina. One scenario might be leasing a slip to an existing customer, and another scenario might be leasing a

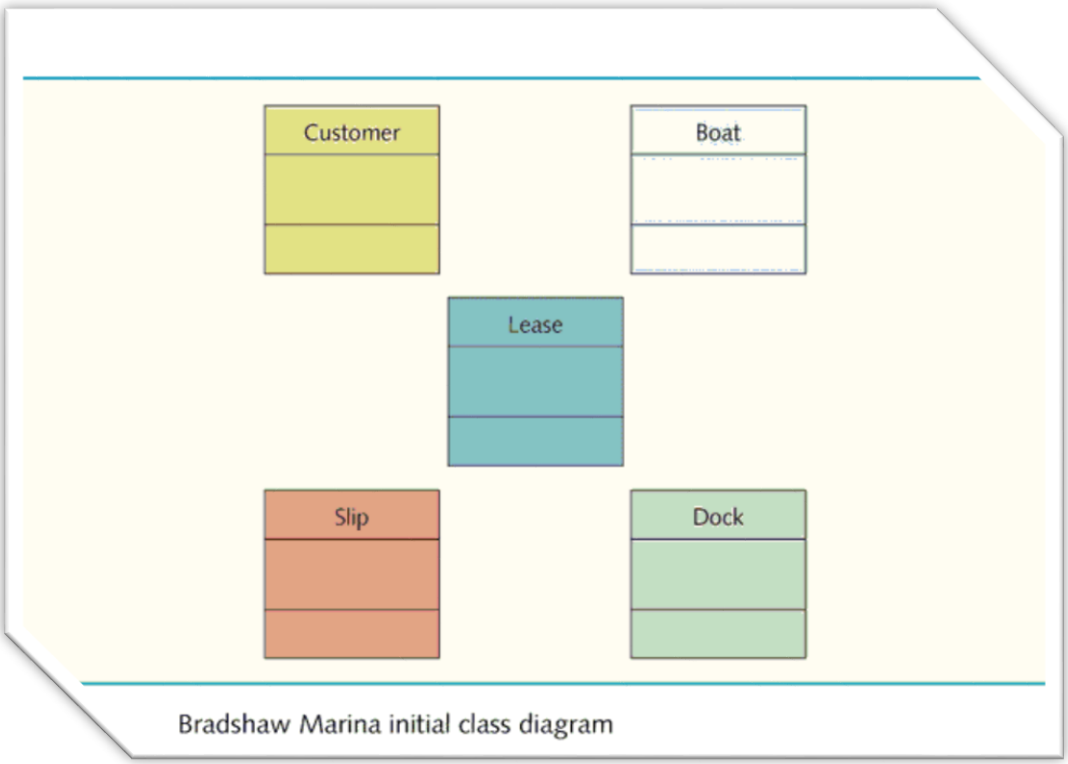
slip to a new customer. Further, another scenario might be leasing an annual slip to a customer, and another might be leasing a daily slip to a customer.

Scenarios can become very specific. It might take several attempts to create a comprehensive and mutually exclusive list of scenarios for the more important use cases. For example, given the situation involving existing customers, new customers, annual slip leases, and daily day leases, the scenarios for one use case might be finalized as follows:

- Lease annual slip to existing customer
- Lease annual slip to new customer
- Lease daily slip to existing customer
- Lease daily slip to new customer

Identifying Bradshaw Problem Domain Classes

Once you identify the use cases and scenarios, you explore the problem domain classes involved in the use cases. You and the team meet with Bradshaw Marina again to ask about the things that are involved in the work of the marina—in this case the customers, boats, leases, slips, and docks. The first step is to begin an initial class diagram that includes these potential classes, as shown in Figure 2.



Bradshaw Marina initial class diagram

Figure 2: Bradshaw Marina initial class diagram

One of the first questions your team asks is about docks and slips. Sometimes the users talk about docks and sometimes they talk about slips. It turns out (after quite a few explanations and sketches) that Bradshaw Marina defines a **dock as an entire floating structure that boat owners walk on to get to their boats** (see Figure 3).

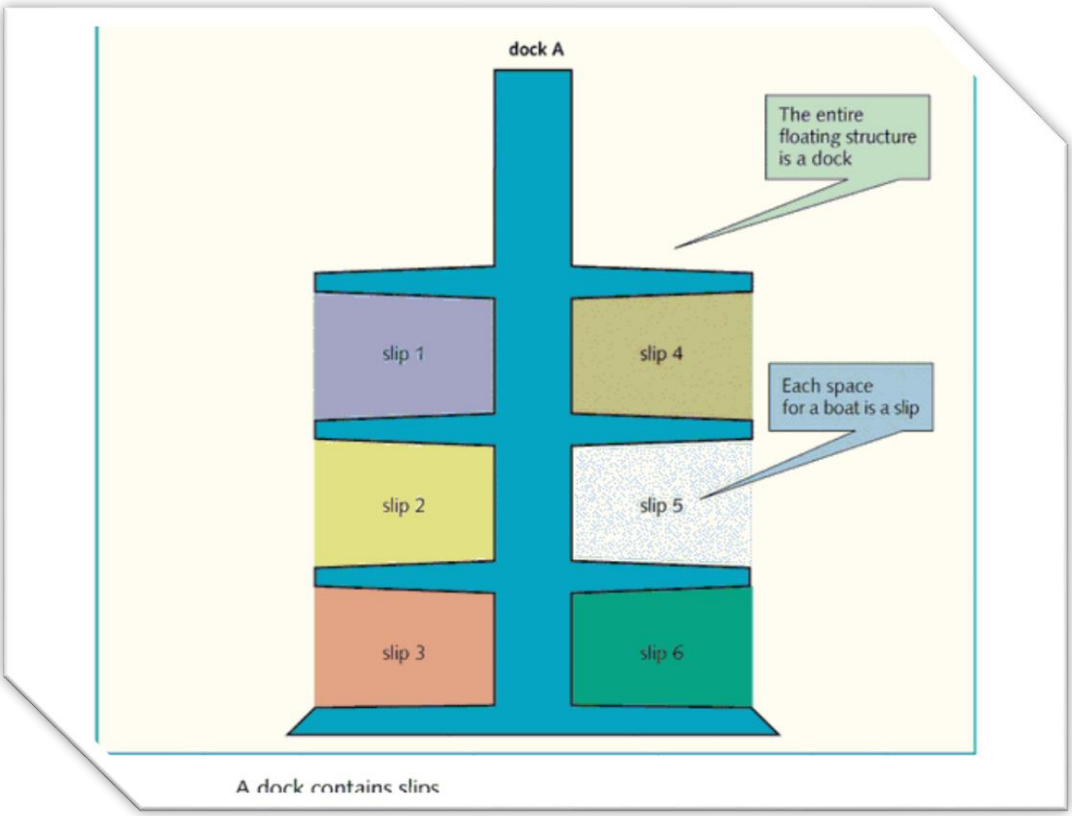


Figure 3: A dock contains slips

As you continue to look more closely at the initial classes, you see that you need more specialized information about boats, slips, and leases in the system. You refine the classes to show the generalization/specialization hierarchies that will require inheritance, as in Figure 4.

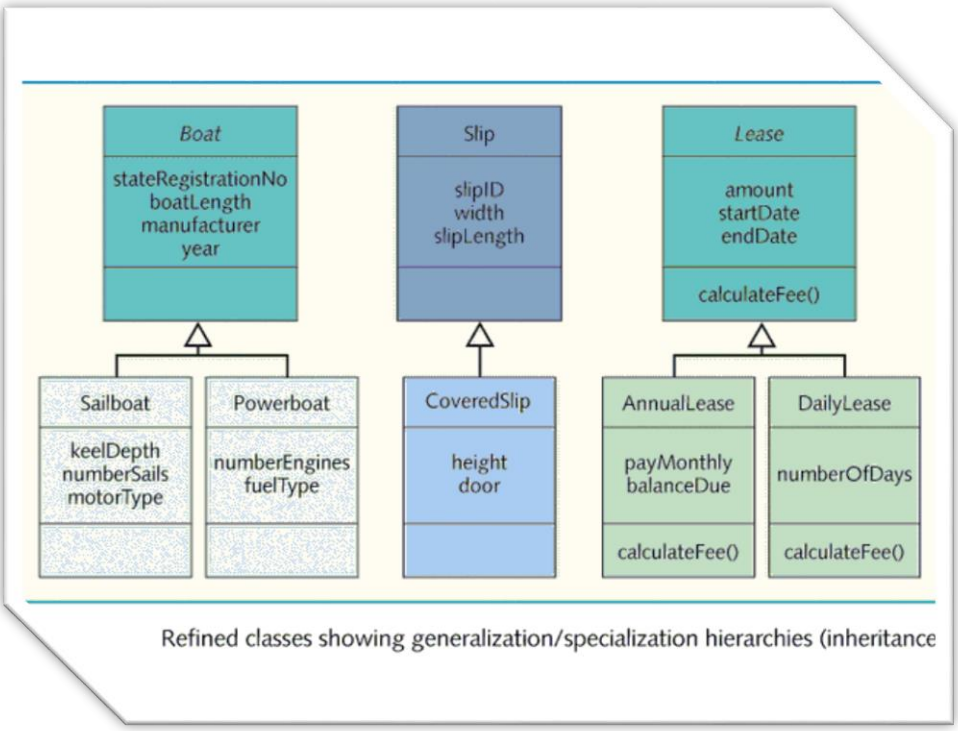


Figure 4: Refined classes showing generalization/specialization hierarchies

Because a boat must be either a sailboat or a powerboat, the Boat class is an **abstract class** (shown in italics), meaning it is only used for inheritance. The Lease class is also an abstract class because any lease must be a daily lease or an annual lease. A slip, on the other hand, might be a regular slip or a covered slip. A covered slip is a special type of regular slip, therefore, the Slip class is a concrete class (not abstract).

Once these details are finalized for boats, slips, and leases, the other classes – Customer and Dock –are considered. The complete class diagram showing all problem domain classes is shown in Figure 5.

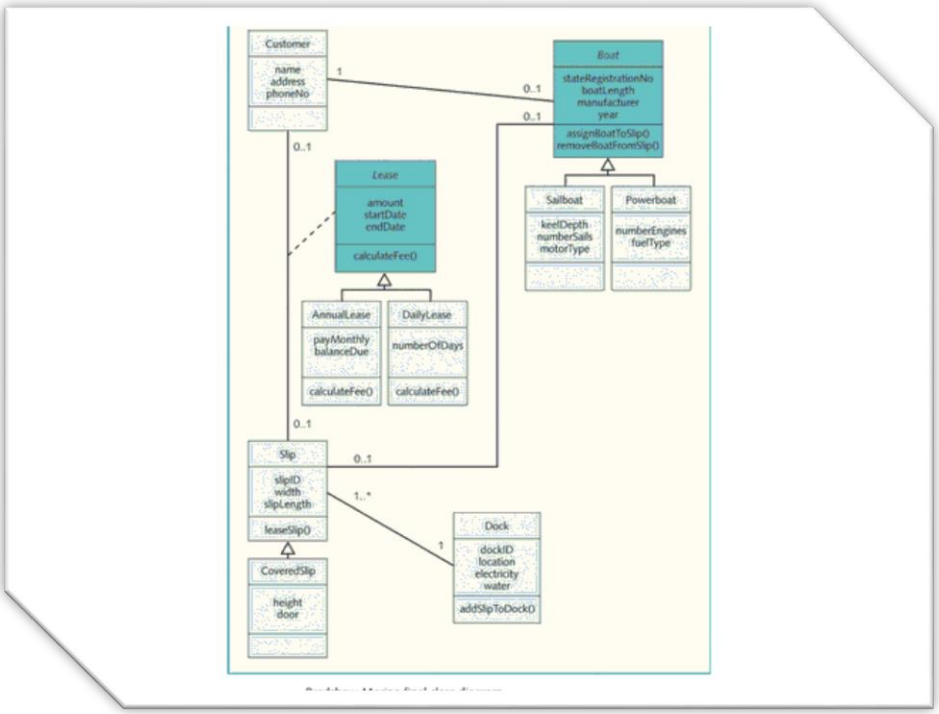


Figure 5: Bradshaw Marina final class diagram

Additional methods are included for Slip, Boat, and Dock. Not all methods are included on the class diagram, particularly during the early stages of OOA, but important methods are often included. You complete the class diagram by identifying and modeling the association relationship among classes. A final association relationship is between slip and customer. A customer optionally leases a slip and a slip is optionally leased by a customer. But this association is really more complex. In fact, the Lease class defined previously exists as a byproduct of the association between customer and slip, so Lease is called an **association class** and is attached to the association line with a dashed line.

Creating a Bradshaw Sequence Diagram

The development team already indicated some methods on the class diagram for Bradshaw Marina, as shown earlier in Figure 5. Please examine Figure 6, which illustrates the sequence diagram for scenario "Lease Annual Slip to Existing Customer".

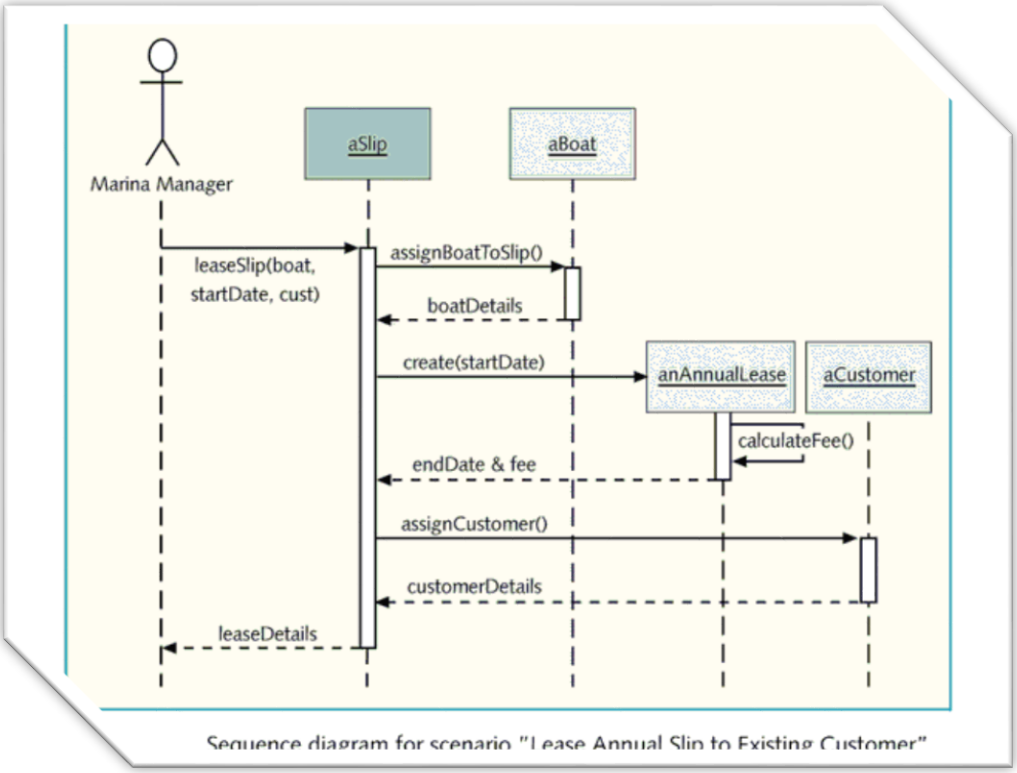


Figure 6: Sequence diagram for scenario "Lease Annual Slip to Existing Customer"

In this scenario, the actor is the marina manager. To lease a slip, the actor sends the leaseSlip message to a specific slip object (aSlip), supplying identifying information about the boat, the start date, and the customer. The leaseSlip method of aSlip interacts with other objects to complete the scenario.

The slip sends the assignBoatToSlip message to the boat, for example, which returns boat details. Finally, the slip returns all of the lease information to the actor—boat details, lease details, and customer details. This sequence diagram is a logical model showing only problem domain objects. As you move from OOA to OOD, you will expand the diagram to show the GUI objects the actor interacts with and the data access classes that handle interaction with the database.

Additional features

1. **The Bradshaw Marina case study describes**
    a. Initial system requirements,
    b. Some additional functions they want to add in the future, and
    c. Even more functions they want to include eventually. List at least three use cases for the functions desired in the second phase of the project. List at least four use cases desired for the third phase of the project.

Phase 2 Use Cases (any three)

Schedule boat service

- Send boat service reminder
- Record completion of boat service
- Produce boat service monthly reports

Phase 3 Use Cases (any four)

- Produce lease bills
- Produce boat service bills
- Record bill payment
- Produce late notice
- Record credit default record
- Produce billing reports

2. **Consider the additional Bradshaw Marina requirements for recording boat service for customers.** What additional problem domain class would you add to the class diagram? What are the association relationships between this class and other classes? Might this class be expanded to a generalization/specialization hierarchy? If so, what are some potential subclasses? Are there any other problem domain classes you might add? Draw the complete class diagram for Bradshaw that include all existing classes plus one or more new classes.

3. **Add a Boat Service class.**
   It would be associated with a Boat (not just Customer because you want to maintain service records for the boat to pass on to a new owner). In fact it would probably be an association class between Customer and Boat, attached to the association line with a dashed line (like Lease). .
   There might be special types of Boat Service with different attributes or different methods for calculating the charges, perhaps Haul Out Service and Repair Service.

   Eventually there might be a Service Transaction class that records the transactions applicable to each service, such as deposit, partial payment, credit, and refund. There might be Service Types that are associated with each Boat Service or there might be Service Parts used for a repair service.

   The class diagram would vary depending upon the specific classes used.

Figure 6 shows a sequence diagram for Bradshaw scenario lease slip to existing customer.

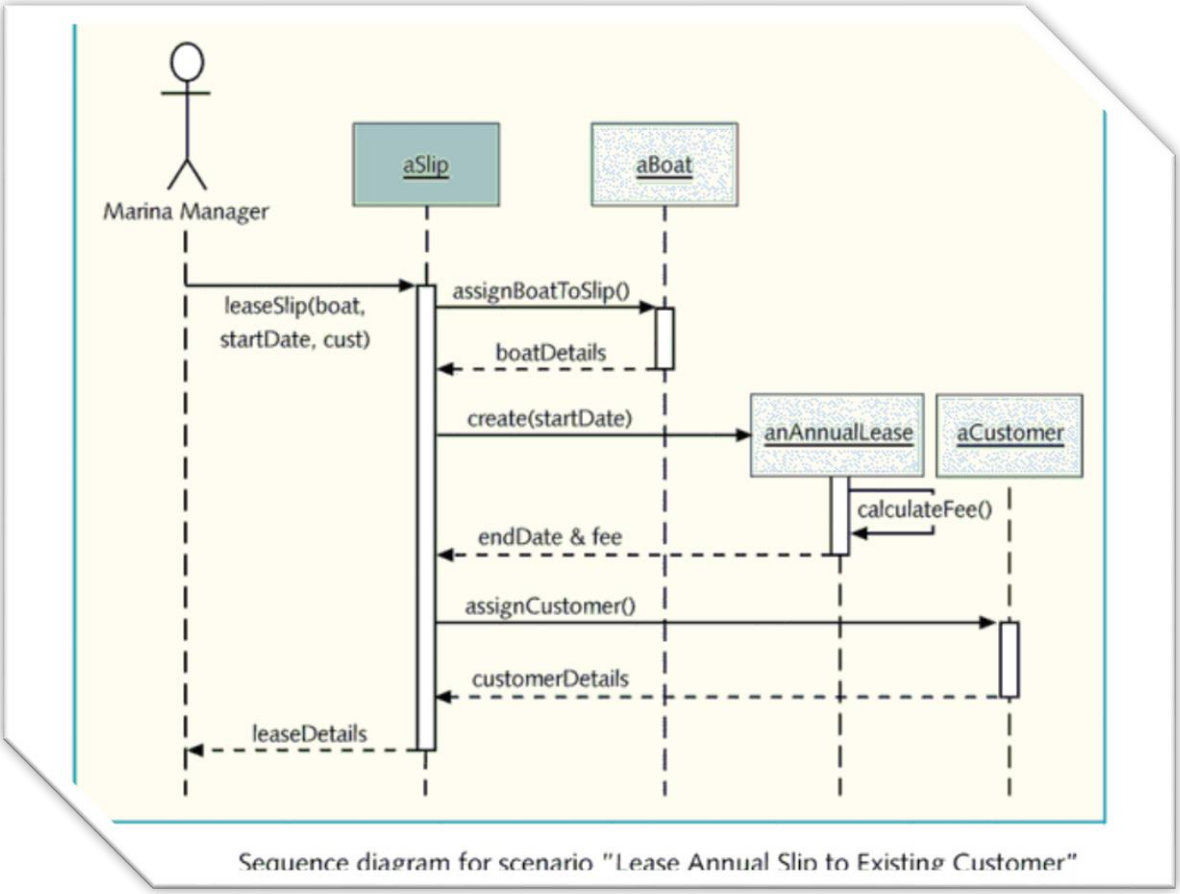
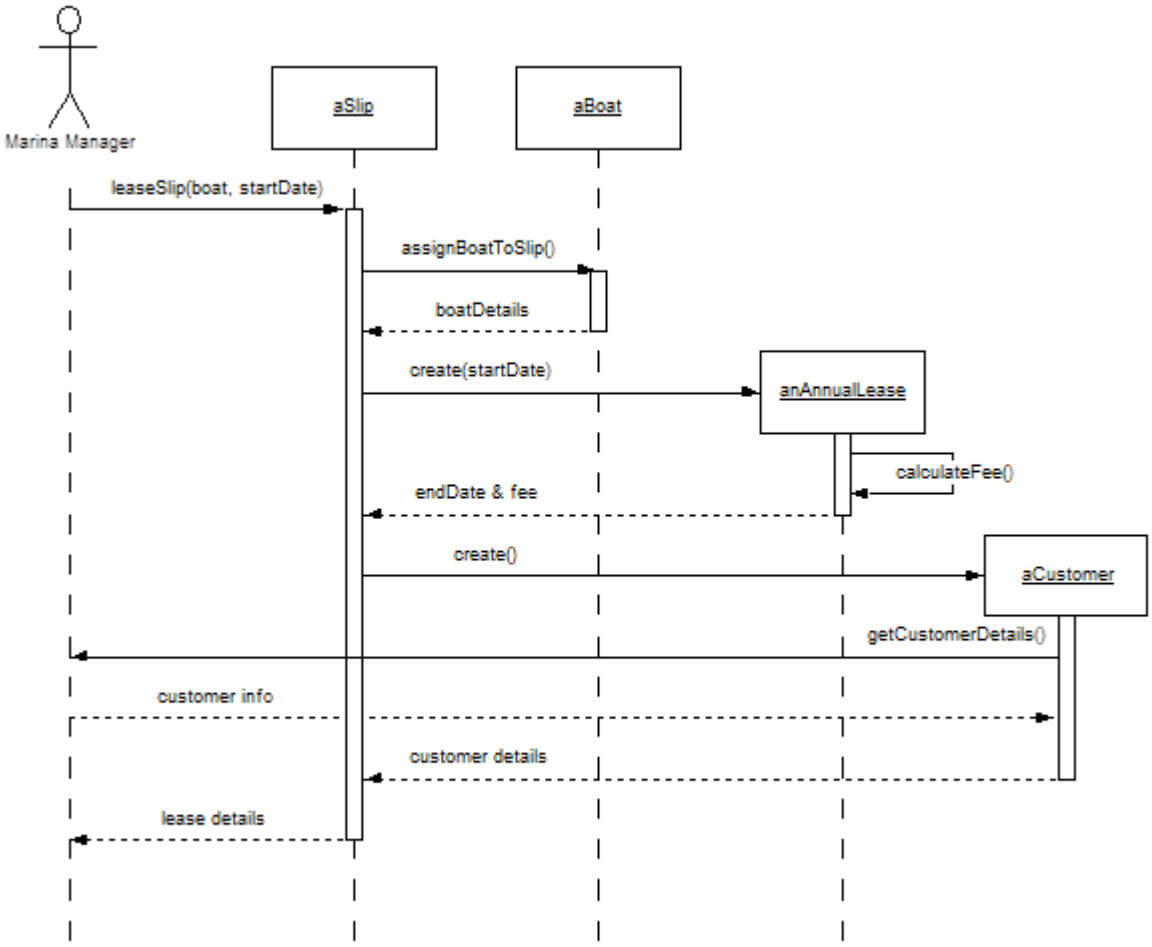
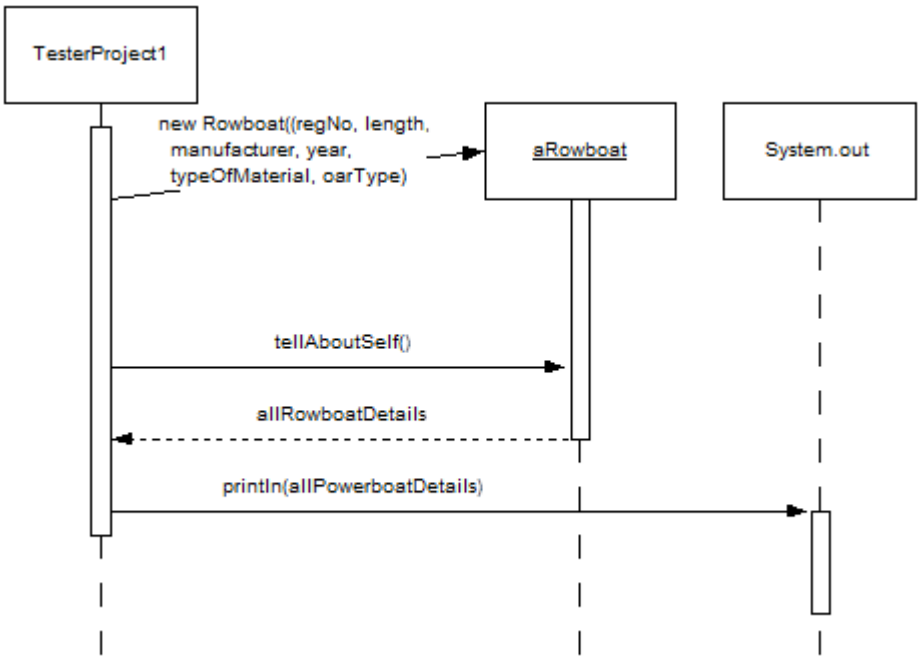


Figure 7: Sequence diagram for scenario "Lease Annual Slip to Existing Customer"

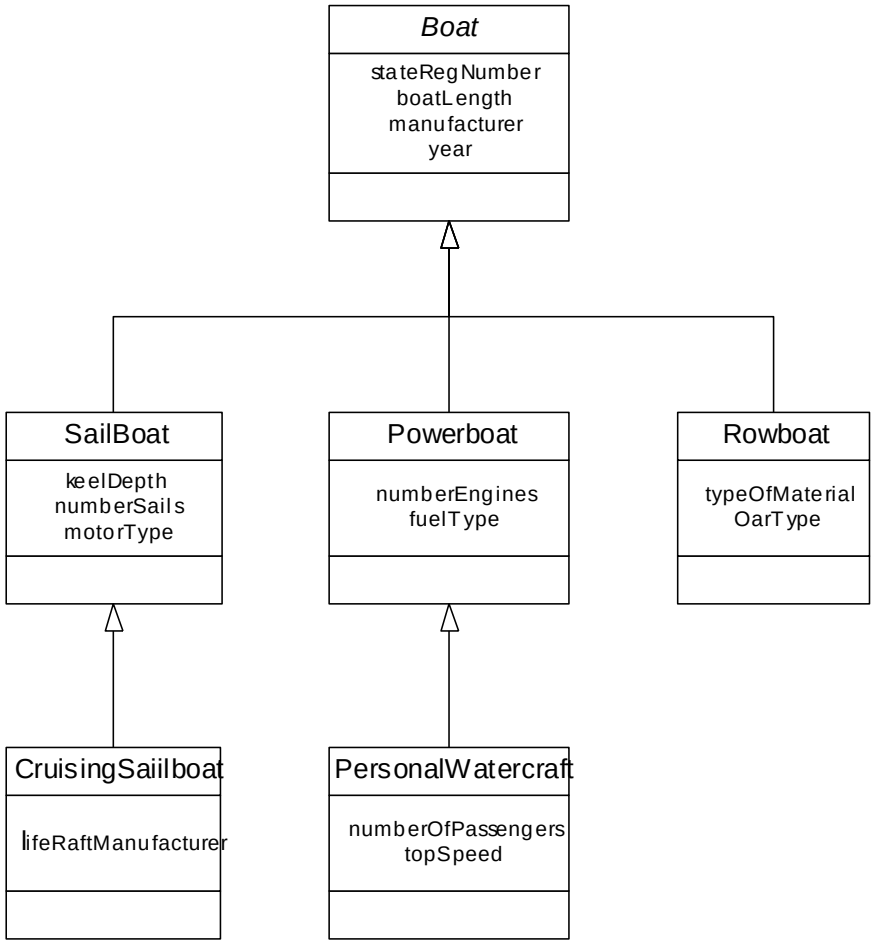


1. **Write an additional subclass for Boat named Rowboat.** A few specific attributes might apply to a rowboat, specifically type of material (wood, fiberglass, or inflatable), and oar type (paddles, wood oars, metal oars). Include accessor methods for the new class, a tellAboutSelf method that overrides then invokes the superclass method, and a tester program that tests the class. Draw a sequence diagram for the tester program.



Sequence diagram:

2. **Consider a special type of boat named PersonalWatercraft.** Should it be a subclass of Boat or Powerboat? Discuss. Draw the generalization/specialization hierarchy for all types of boats in this chapter based on your decision, including CruisingSailboat, and Rowboat discussed previously. Class diagram assuming PersonalWatercraft is subclass of Powerboat:


Boat
stateRegNumber
boatLength
manufacturer
year
SailBoat
keelDepth
numberSails
motorType
Powerboat
numberEngines
fuelType
Rowboat
typeOfMaterial
OarType
CruisingSaiilboat
lifeRaftManufacturer
PersonalWatercraft
numberOfPassengers
topSpeed