Gregory Prosper COP 4331

Problem 2.1 (a)

Log in (Edited)

1. The mailbox owner carries out **Reach an Extension**.

2. The mailbox owner types the passcode, followed by the # key. (The default pass- code is the same as the mailbox number. The mailbox owner can change it—see **Change the Passcode**.)

3. The voice mail system plays the mailbox menu:
   Enter 1 to retrieve your messages.
   Enter 2 to change your passcode.
   Enter 3 to change your greeting.
   Enter 4 to delete all messages from a phone number

Delete a Message

1. The mailbox owner carries out **Log in**

2. The mailbox owner selects the "Delete all messages from a phone number" menu option.

3. The voice mail system speaks a prompt.
   Enter the phone number to delete all messages

4. The user types the phone number, followed by the # key.

5. All messages from specified phone number are deleted.

6. Continue to step 3 of **Log in**

Variation #1 Number Not Found

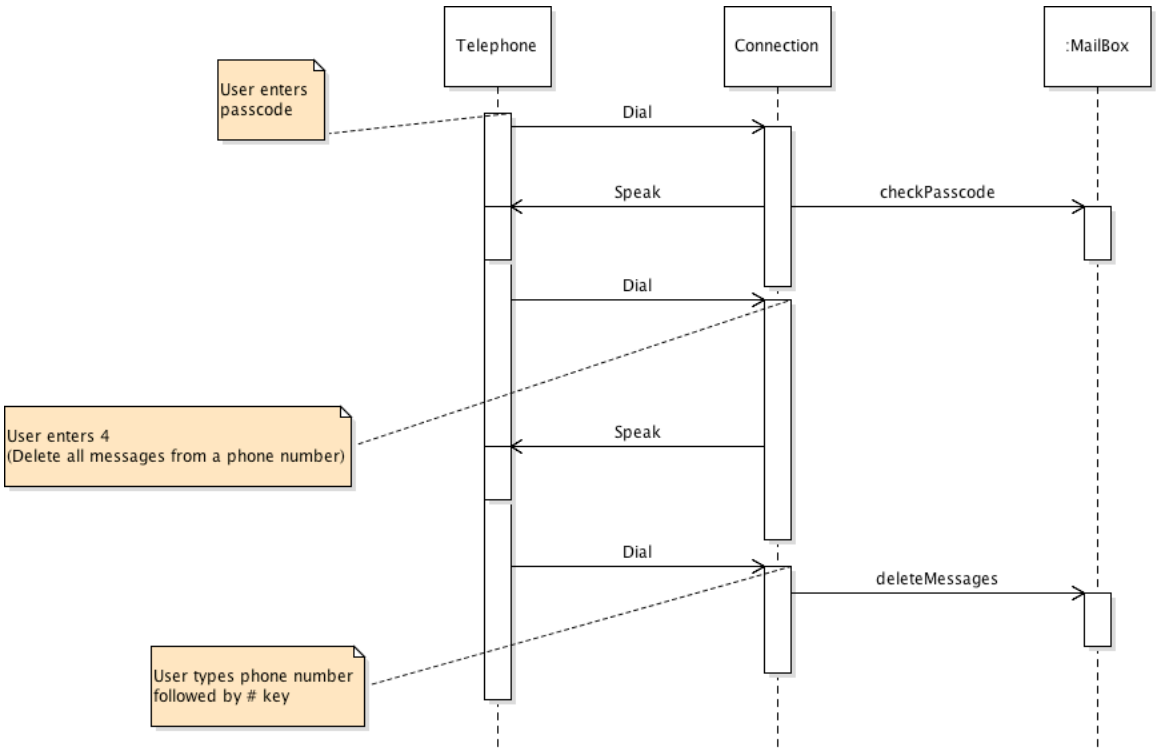
1.1. Start at step 4.

1.2. The voice mail system speaks a prompt.
The phone number you entered was not found

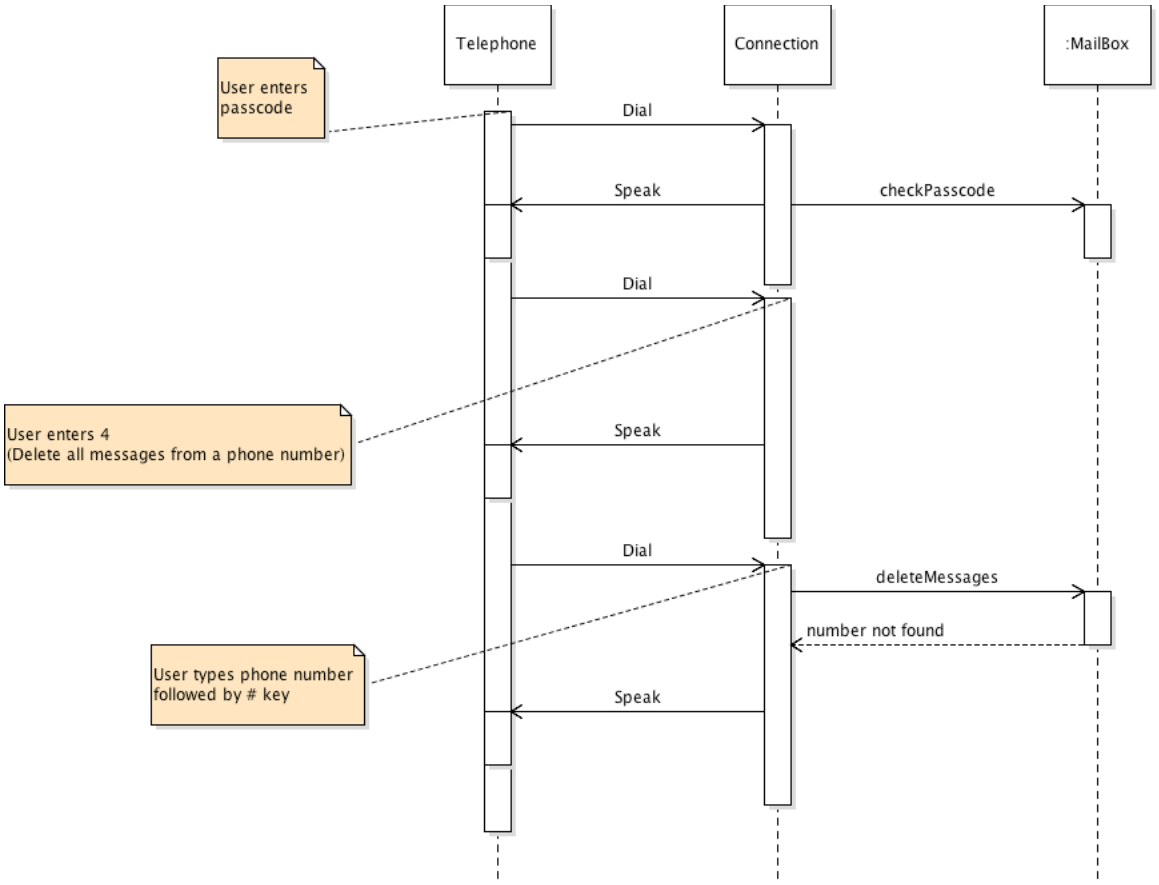
1.3. Continue to step 3 of **Log in**

Problem 2.1 (b)

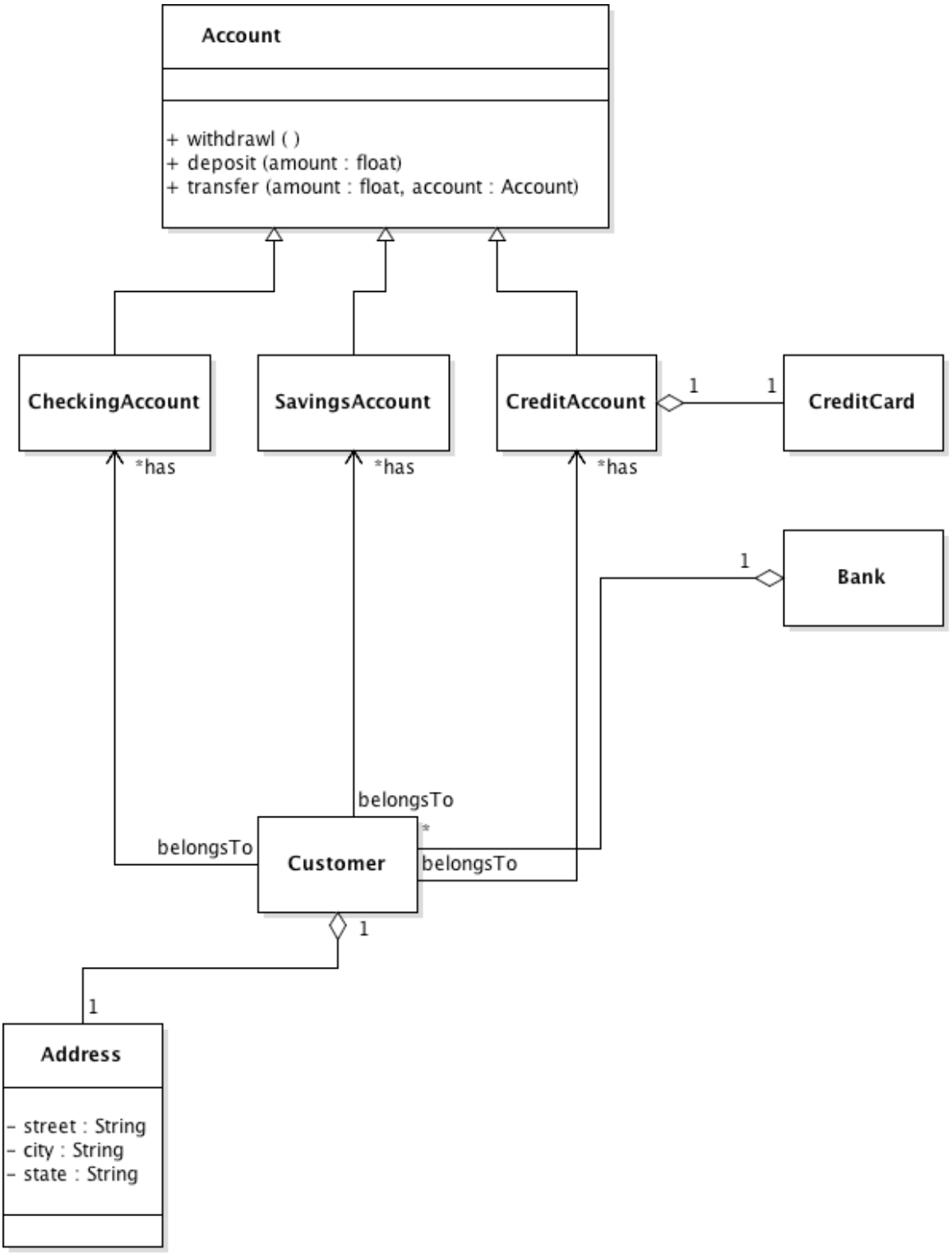
Normal Case:



Variation #1:


Telephone
Connection
:MailBox
User enters passcode
Dial
Speak
checkPasscode
Dial
User enters 4
(Delete all messages from a phone number)
Speak
Dial
deleteMessages
number not found
User types phone number followed by # key
Speak

Problem 2.2


Account
+ withdrawl ( )
+ deposit (amount : float)
+ transfer (amount : float, account : Account)
CheckingAccount
SavingsAccount
CreditAccount
CreditCard
1
1
*has
*has
*has
Bank
1
belongsTo
*
belongsTo
Customer
belongsTo
1
1
Address
- street : String
- city : String
- state : String

Problem 2.3

CRC Cards

Customer

- Store Customer info

System : Collaborates with **Rental Company**

- Manages Rental Companies
- Manages Reservations

Rental Company : Collaborates with **Car**

- Manages Cars

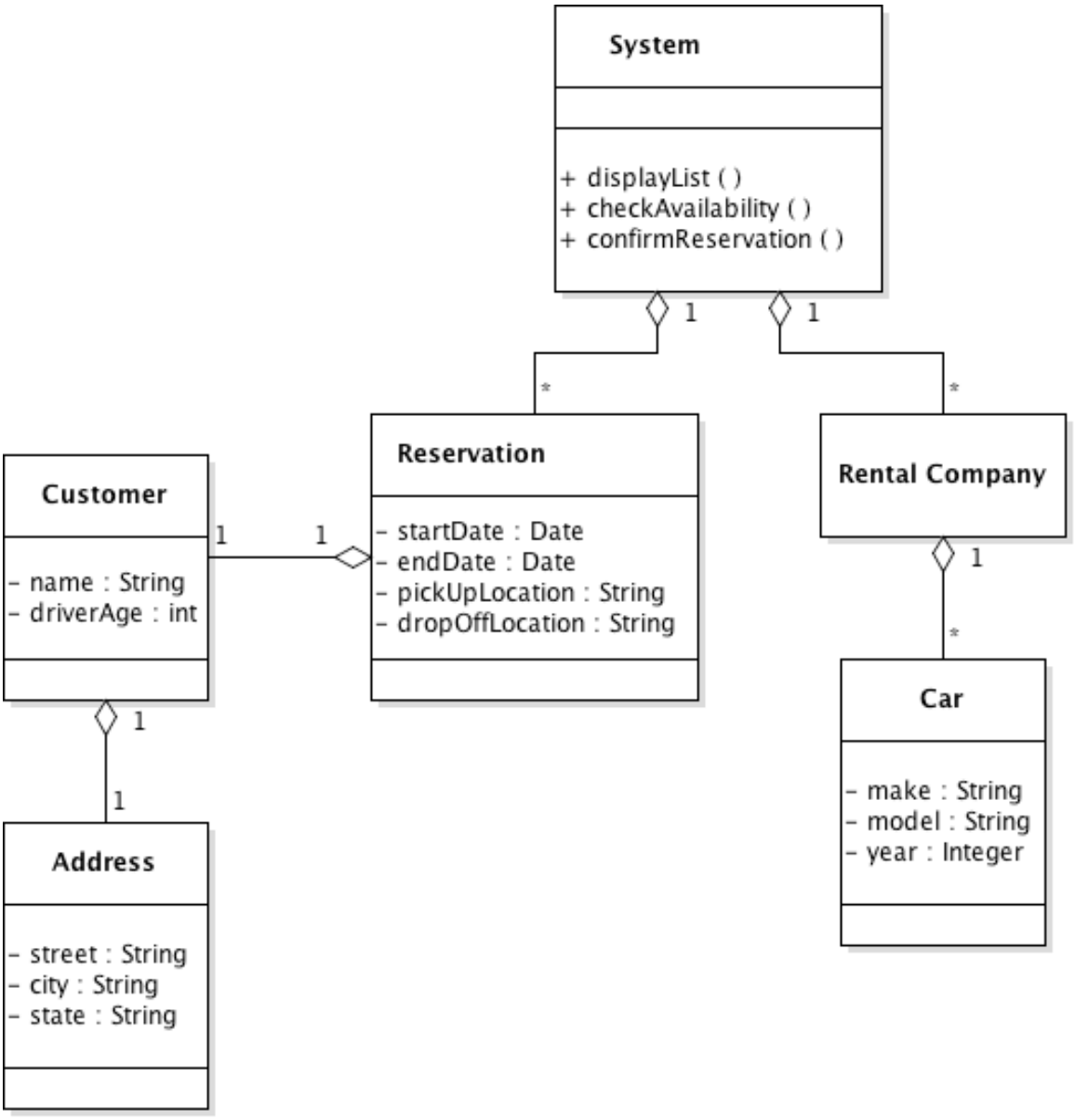
Reservation

- Store Reservation Info

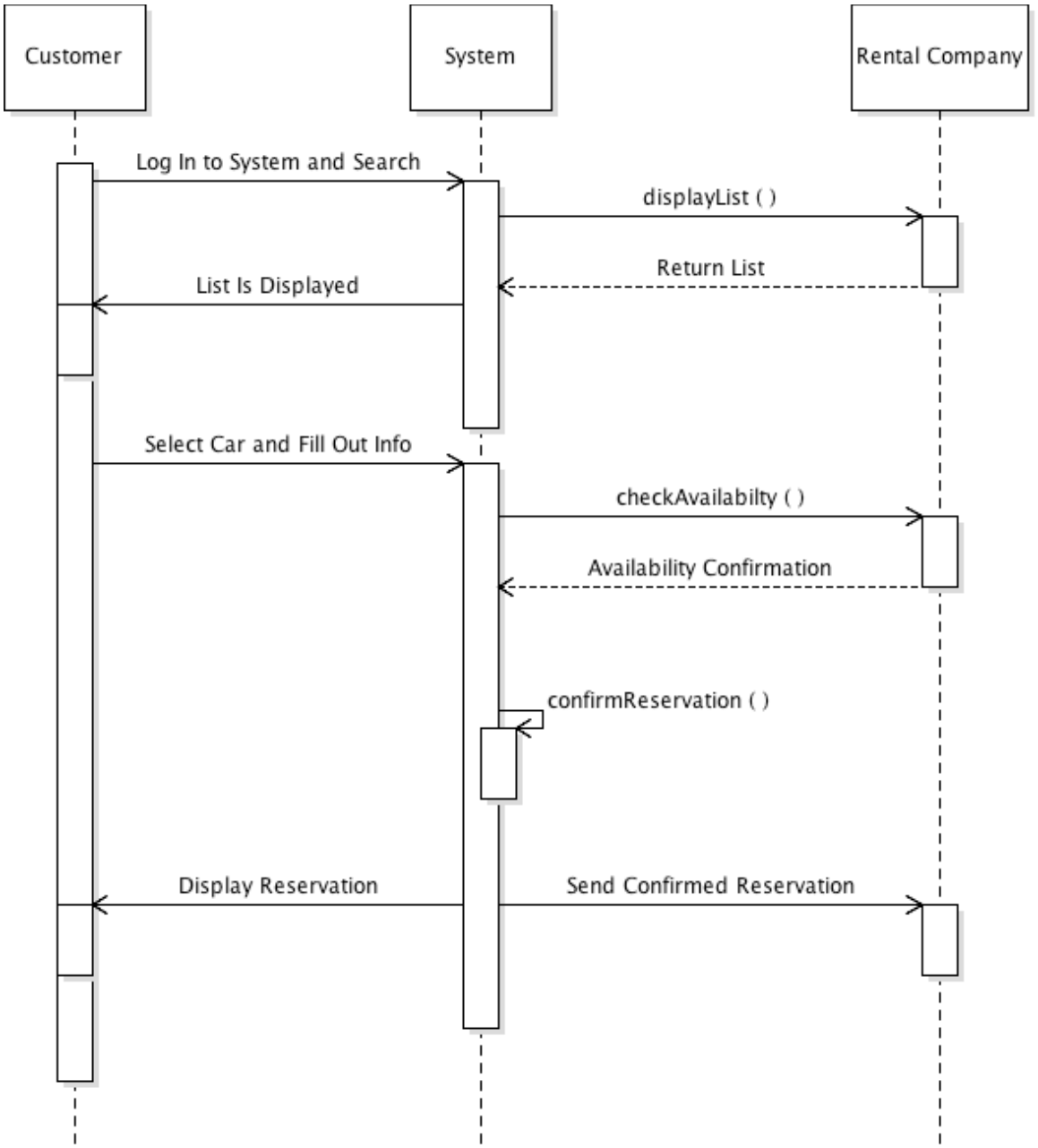
Car

- Store Car Info

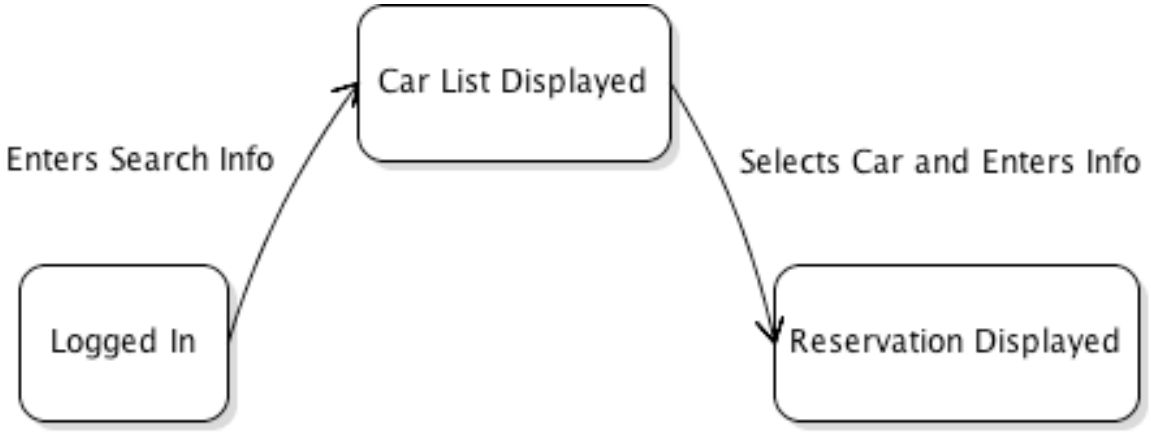
UML Class Diagram



Sequence Diagram



State Diagram



Problem 2.4

Inventory Collaborates with **Product**

- Manages list of products

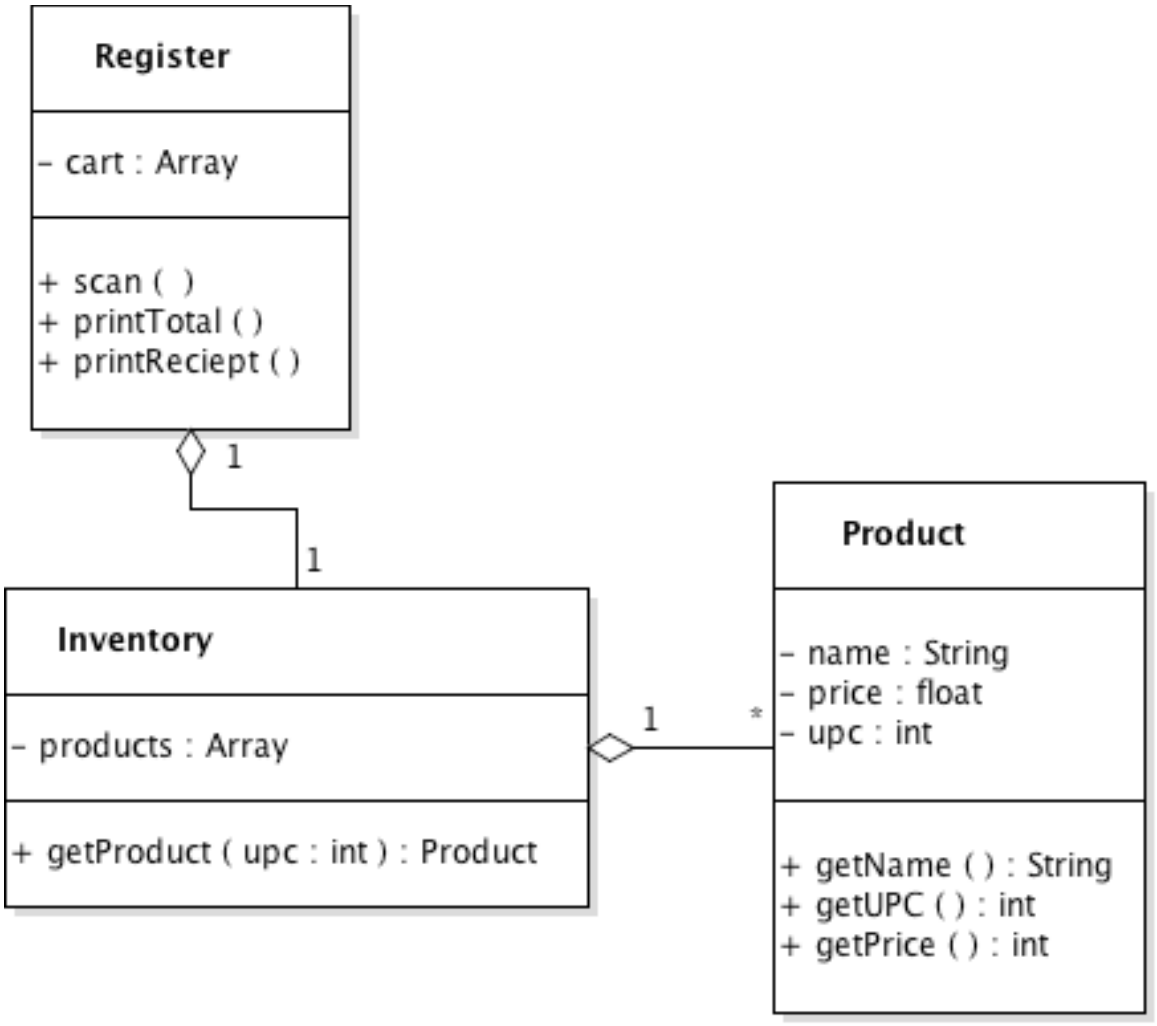
Product

- Contains product info

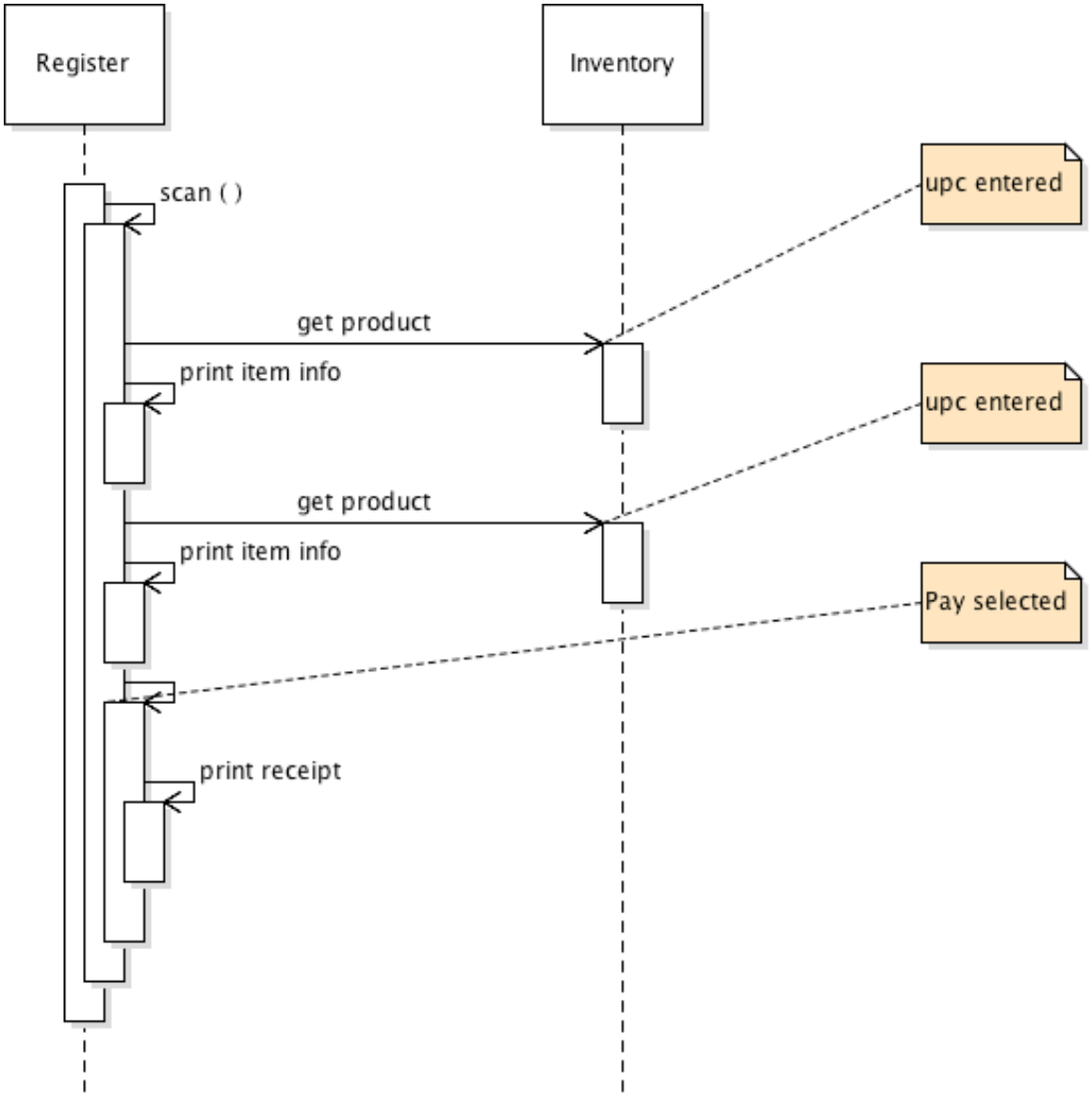
Register Collaborates with **Inventory**

- Manages list of purchased products
- Responsible for scanning products

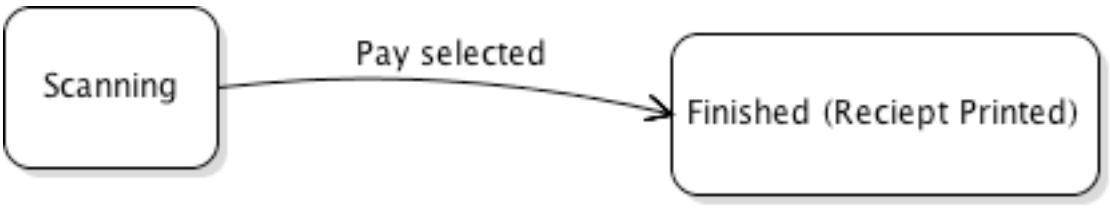
Class Diagram



Sequence Diagram



State Diagram



Code

Product.java

```java
package register;

/**
 * A class for the Product object.
 * */
public class Product {

	/**
	 * Constructs a Product object from the given parameters.
	 * @param name the name of the product.
	 * @param price the price of the product.
	 * @param upc the UPC number of the product.
	 * */
	public Product(String name, float price, int upc){
		this.name = name;
		this.price = price;
		this.upc = upc;
	}

	/**
	 * Retrieves the name of the product.
	 * @return the name of the product.
	 * */
	public String getName (){
		return this.name;
	}

	/**
	 * Retrieves the price of the product.
	 * @return the price of the product.
	 * */
	public float getPrice (){
		return this.price;
	}

	/**
	 * Retrieves the UPC number of the product.
	 * @return the UPC number of the product.
	 * */
	public int getUPC (){
		return this.upc;
	}

	private String name;
	private float price;
	private int upc;
}
```

Inventory.java

```
package register;

import java.util.ArrayList;

/**
 * A class for the Inventory object.
 * */
public class Inventory {

	/**
	 * Constructs a Inventory object.
	 * */
	public Inventory (){
		String[] names = {"Bread","Cookies","Pantry","Couch","T-
				 Shirt","Tv","Hammer","Shoes","Drawer","Basket","Belt"
				,"Toothbrush","Radio","Lotion","Slippers","Curtains"};

		float[] prices = {(float) 2.34,(float) 1.12,(float) 13.23,(float)
				532,(float) 5.99,(float) 1200,(float) 17.78,(float)
				34.99,(float) 32.89,(float) 2.12,(float) 3.45,(float)
				7.99,(float) 133,(float) 3.55,(float) 13.44,(float)
				198.10};

		//Creates products and adds them to products ArrayList
		for (int i = 0; i < prices.length; i++) {
			Product p = new Product(names[i],prices[i],i+100);
			this.products.add(p);
		}
	}

	/**
	 * Retrieves a product from the Inventory by UPC number.
	 * @param upc the UPC number of the product to be searched for.
	 * @return a product from the inventory with the matching UPC number.
	 * */
	public Product getProduct (int upc){
		for (Product p : this.products) {
			if (p.getUPC() == upc) {
				return p;
			}
		}
		Product p = new Product("",0,0);
		return p;
	}

	private ArrayList<Product> products = new ArrayList<>();
}
```

Register.java

```
package register;

import java.util.ArrayList;
import java.util.Scanner;

/**
 * A class for the Register object.
 * */
public class Register {

	/**
	 * Constructs a Register object.
	 * */
	public Register (){
		System.out.print("Register Initialized. Enter UPC numbers or Enter 'pay'
				to complete transaction\n");
	}

	/**
	 * Scans for UPC numbers until user inputs 'pay'.
	 * */
	public void scan (){
		Scanner in = new Scanner(System.in);

		while (in.hasNext()) {
			if (in.hasNextInt()) {
				int upc = in.nextInt();
				if (upc >= 100 && upc <= 115) {
					this.cart.add(this.inventory.getProduct(upc));
				}
				else {
					System.out.print("Invalid UPC number, must be within
							100 - 115 \n");
				}
			}
			else if (in.next().equalsIgnoreCase("pay")) {
				this.printReciept();
				break;
			}
			else{
				System.out.print("Invalid Input. Enter UPC number or
						'pay'.\n");
			}
		}

		in.close();
	}

	/**
	 * Prints the total to the console.
	 * */
	public void printTotal (){
		float total = 0;
		for (Product p  : this.cart) {
			total += p.getPrice();
		}
		System.out.printf("\nTotal: $%.2f", total);
	}
```

```
	/**
	 * Prints Every item in cart to console then prints total to console.
	 * */
	public void printReciept (){
		System.out.print("Reciept:\n");
		System.out.printf("%3s %15s %10s\n","UPC","Item","Price");

		if (this.cart.isEmpty()) {
			System.out.print("(No Items in Cart)");
		}
		else {
			for (Product p  : this.cart) {
			     System.out.printf("%3d %15s %10.2f \n",
			        p.getUPC(),p.getName(),p.getPrice());
			}
		}
		this.printTotal();
	}

	private ArrayList<Product> cart = new ArrayList<>();
	private Inventory inventory = new Inventory();
}
```

RegisterTester.java

```java
package register;

/**
 * A class to test Register class.
 * */
public class RegisterTester {

	/**
	 * Main class.
	 * @param args
	 */
	public static void main(String[] args) {
		Register register = new Register();
		register.scan();

	}

}
```