UML Exercises

Problem 1

Given the following problem description, draw a UML class diagram representing the software system design you would design. Include as many details as possible.

You need to design a software system that handles installation requests for a cable company. Installation requests are serviced on a first-come first-serve basis. An installation request is identified by an id number and the technician it is assigned to. An installation request includes information about a customer, and type of service requested (digital o analog). The cable company has two types of technicians: those that install analog systems and those installing digital ones. Installing an analog system requires connecting a cable to the back of the TV and setting up the TV; installing a digital system requires making a connection to a company digital box, and a separate connection to the TV. In addition to installing systems, any technician can generate a bill for a customer, and answer any general questions about services the company offers.

Problem 2

Define the UML Class Diagram associated with the classes/interfaces below.

```
public interface ElectricalDev { public double watts(); }

public class Remote {
        private TV tv;
        public Remote(TV tv) { this.tv = tv; }
        public void changeChannel(int channel) { tv.changeChannel(channel);}
}

public abstract class TV implements ElectricalDev {
        private String brand;
        private int channelNum;
        public Remote remote = new Remote(this);

        public TV(String brand) {this.brand = brand; channelNum = 1;}
        public String toString() {
                return "Brand: " + brand + " Channel: " + channelNum;
        }
        public void changeChannel(int channelNum) {
                this.channelNum = channelNum;
                processAndDisplaySignal();
        }
        protected abstract void processAndDisplaySignal();
}

public class HighDefTV extends TV {
        private String digitalConverter;

        public HighDefTV(String brand, String digitalConverter) {
                super(brand);
                this.digitalConverter = digitalConverter;
        }

        public String toString() { return super.toString() + ", Converter: " + digitalConverter; }
        public void processAndDisplaySignal() {System.out.println("Process/Display HD signal."); }
        public double watts() { return 100; }
}
```