

Assignment 1 (Requirements Elicitation and Documentation)

Exercise 1 (Requirements Elicitation)

You have been given the role of specifying the requirements document for an online flower shop.

The client has been running this shop for many years and has become friends with both his customers and his suppliers. However, the client would like to streamline the process of distributing flower purchases and the purchasing of bulk flowers. He currently has one messenger for the delivery of flowers and only three suppliers of bulk flowers that he receives from Holland. One of his biggest concerns in "modernizing" his flower shop is that he may lose the personal touch offered by his shop.

His current customers appreciate his touch for determining the right flowers for the right moments in their lives. The usual customers are normally given an in-house account, which they may pay at the end of the month. New or inconsistent customers must pay either at delivery or by credit card over the phone if the flowers are a gift.

He has found that most of the mistakes in the process of ordering flowers is with new customers. A new customer must provide him with an address and possibly a credit card every time they call in for a delivery. Furthermore, if there is a discrepancy in the address or credit card, he has found that the flowers are usually delivered too late and the client has already found another shop to purchase them from. Or, that the client did not exist at all!

He is also interested in determining which flowers are popular during the major events of the year. One of the issues in working with his suppliers is to determine which flowers to get, and how many of the flowers. Waste of flowers is very expensive and he would like to use this "modernization" as a way of helping cut down on this waste.

He is currently more interested in modernizing his relationship with his clients and wishes to see how this works before risking a modernization with his suppliers.

He currently has an internet connection at the flower shop attached to his older PC. His shop already has a point of sales system for in store sales. The interface for the online store will be handled by his daughter who is currently taking lessons in web design.

It is your responsibility to elicit requirements and write a document capturing these requirements to begin discussions with the customer. Start by determining the actors of the system. Write down the scenarios of the system (incl. exceptional one) and determine any open issues. Once you have determined the open issues, discuss these with the flower shop owner (your assistant). Think about how you could generalize the scenarios into use cases. Once you have done this, work out the non-functional requirements of the system.

Exercise 2 (Design and Documentation)

You are given the following classes:

```
class Edge
{
    Node from;
    Node to;
    int distance;
}

class Graph
{
    List<Edge> edges;
    Node source;
    Node destination;

    void addEdge(Edge e)
    {
        edges.Add(e);
    }

    List<Node> shortestPath()
    {
        ...
    }
}
```

Since computing the shortest path in a graph is an expensive operation, we would like to avoid re-computing it unless it is necessary.

1. Come up with a design to accomplish this.
2. How does this design influence the client-visible documentation for class **Graph**?
3. Can you document your design for the implementor/maintainer? If so, what contracts could be provided?
4. Are there alternative designs that achieve comparable results?

Exercise 3 (Design)

You have seen different list implementations in the lectures, one of which uses reference counting to reduce the need for expensive cloning operations:

```
class ListRep<E> {
    E[] elems;
    boolean shared;

    ListRep(ListRep<E> o) {
        elems = o.elems.clone();
        shared = false;
    }
}

class List<E> {
    ListRep<E> rep;
    int len;

    List(ListRep<E> r, int l) {
        rep = r;
        len = l;
    }

    void set(int index, E e) {
        ListRep<E> r;
        if (rep.shared) {
            r = new ListRep<E>(rep);
        } else {
            r = rep;
        }
        r.elems[index] = e;
        rep = r;
    }

    List<E> clone() {
        rep.shared = true;
        return new List<E>(rep, len);
    }
}
```

1. Come up with scenarios in which this implementation could be further improved to eliminate unnecessary cloning operations.
2. How could you modify the implementation to eliminate this inefficiency?
3. Are there still scenarios in which this new implementation could be made more efficient?
4. How could you modify the new implementation to eliminate this inefficiency?