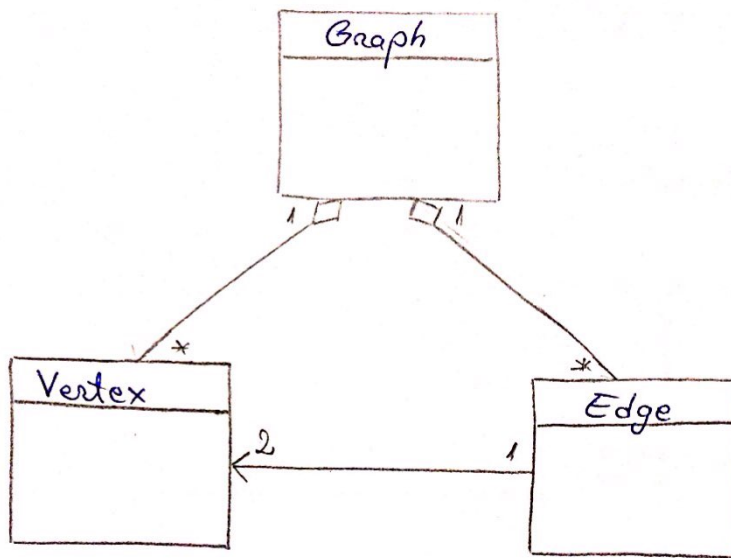
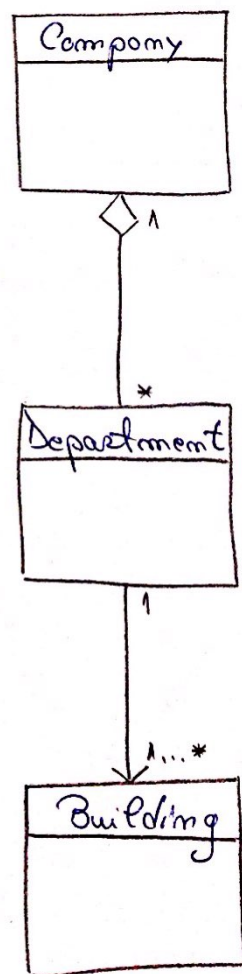


# PROBLEMS

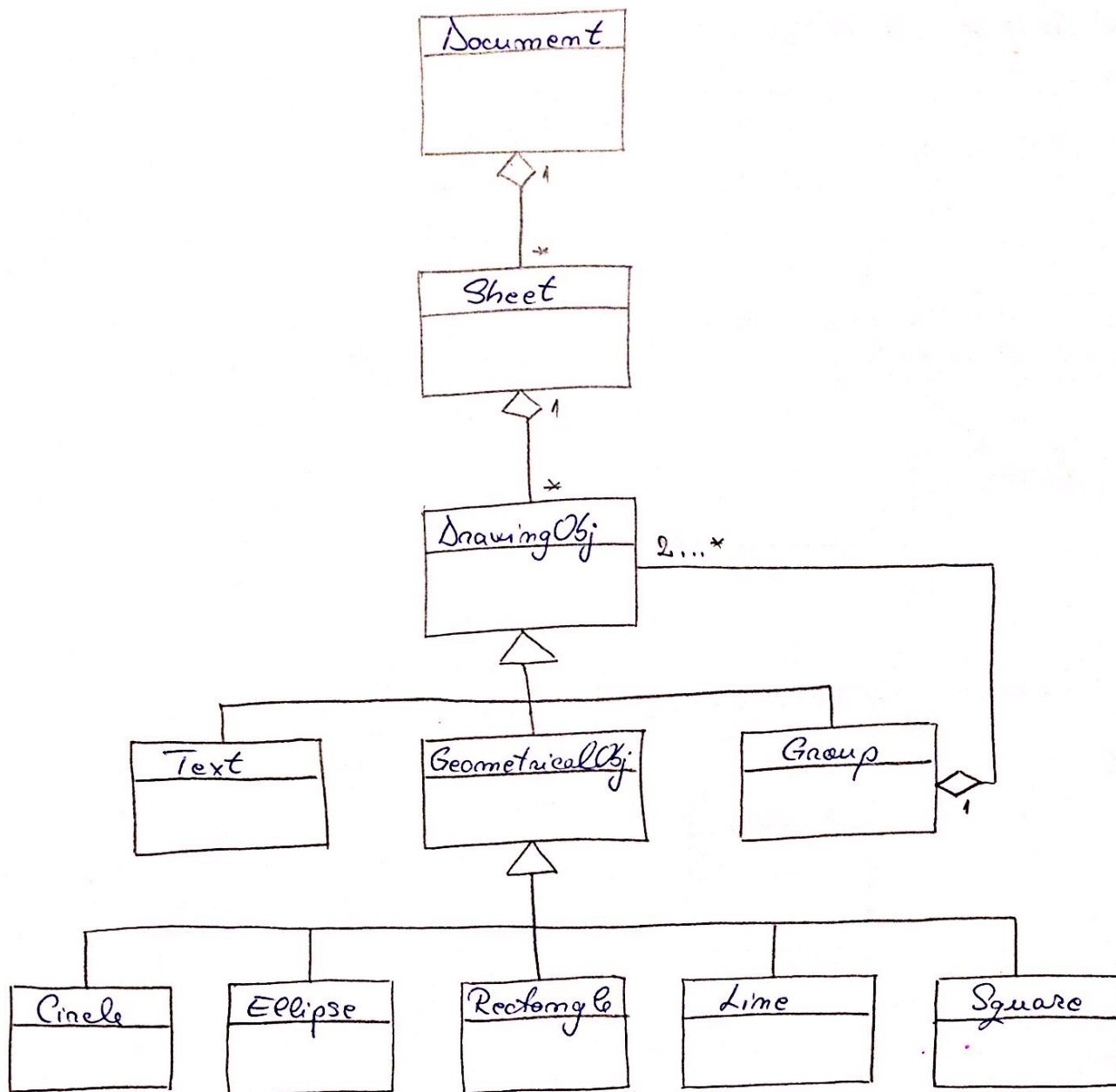
## P1: UNDIRECTED GRAPHS



## P2: COMPANY, DEPARTMENTS, BUILDINGS

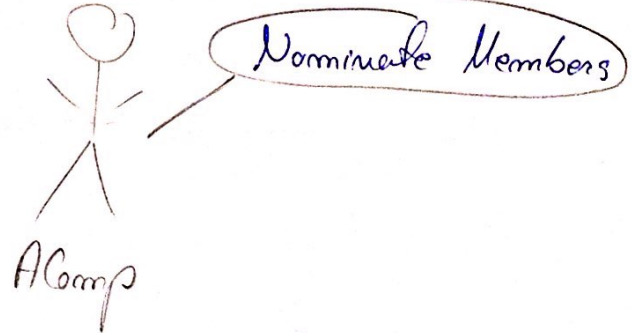
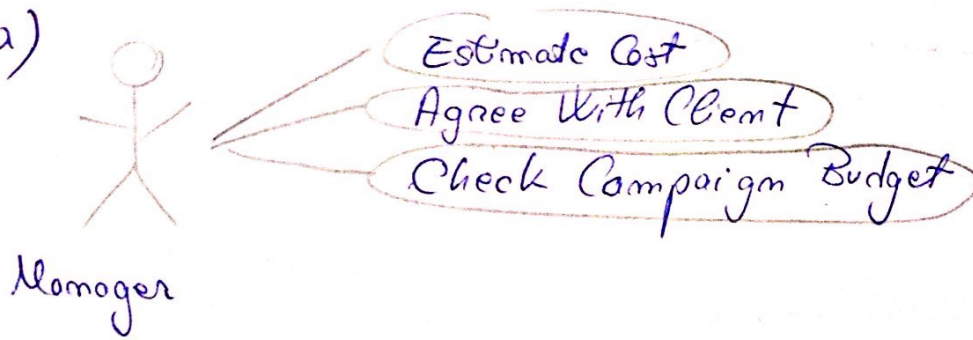


### P3 : GRAPHICAL DOCUMENT EDITOR

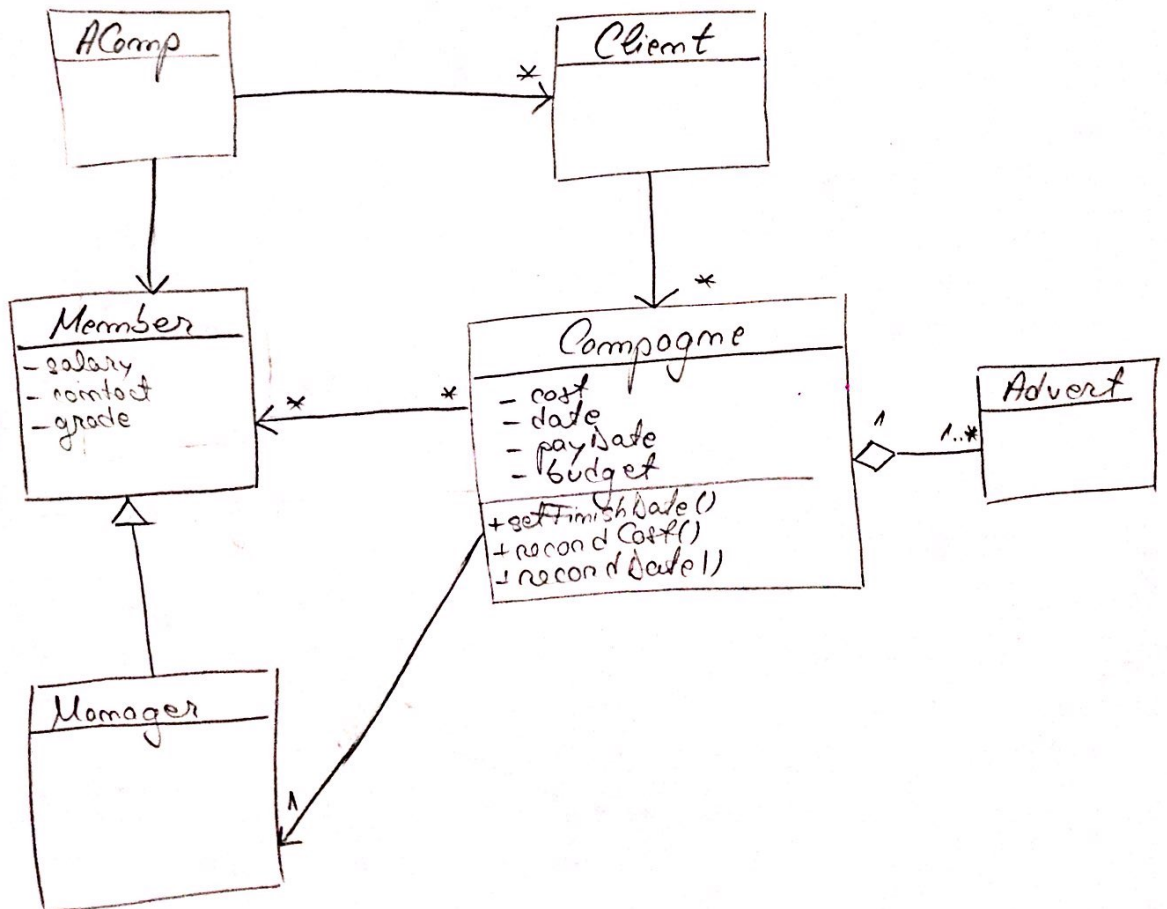


P4: A\*

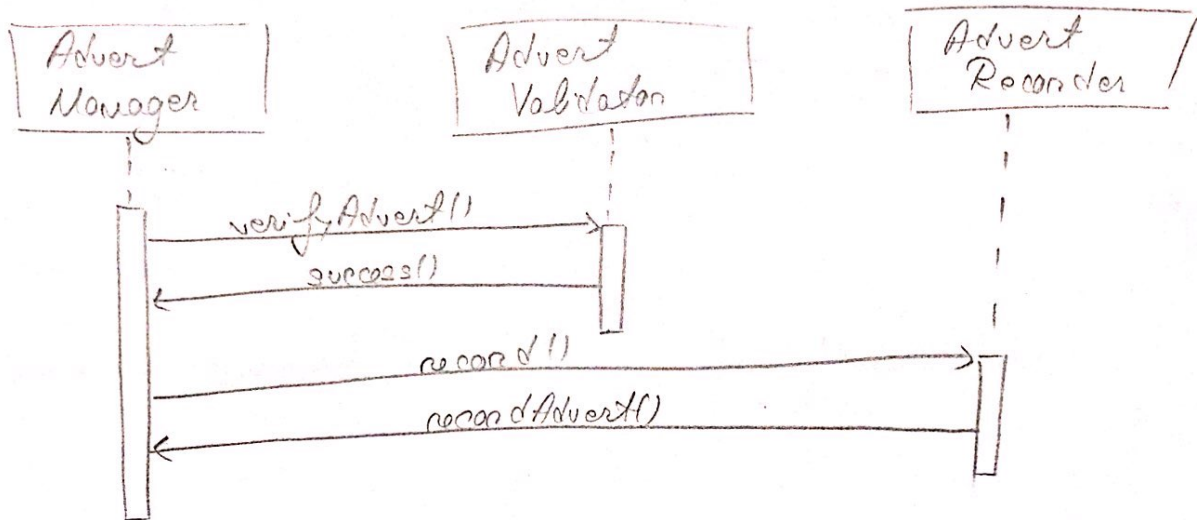
a)



b)

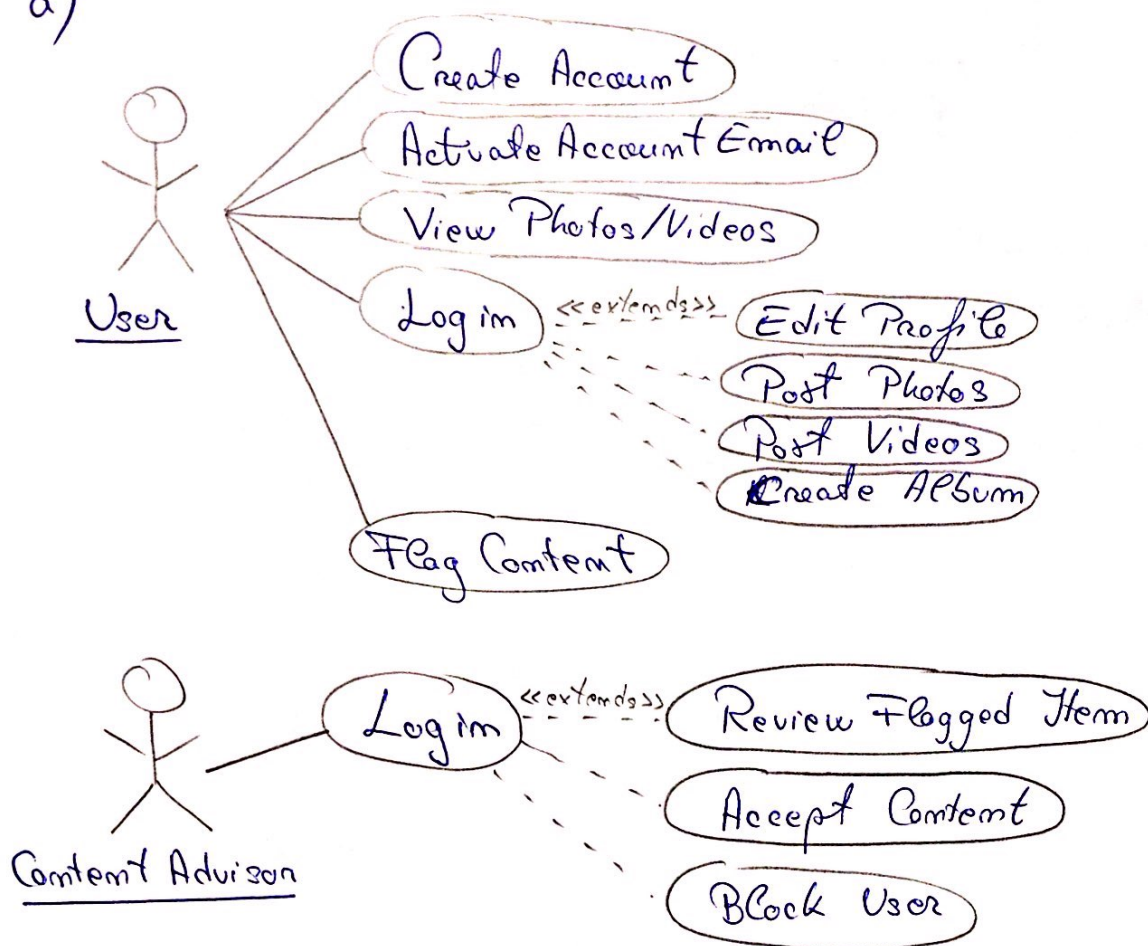


# c) Sequence Diagram: Add Advertisement

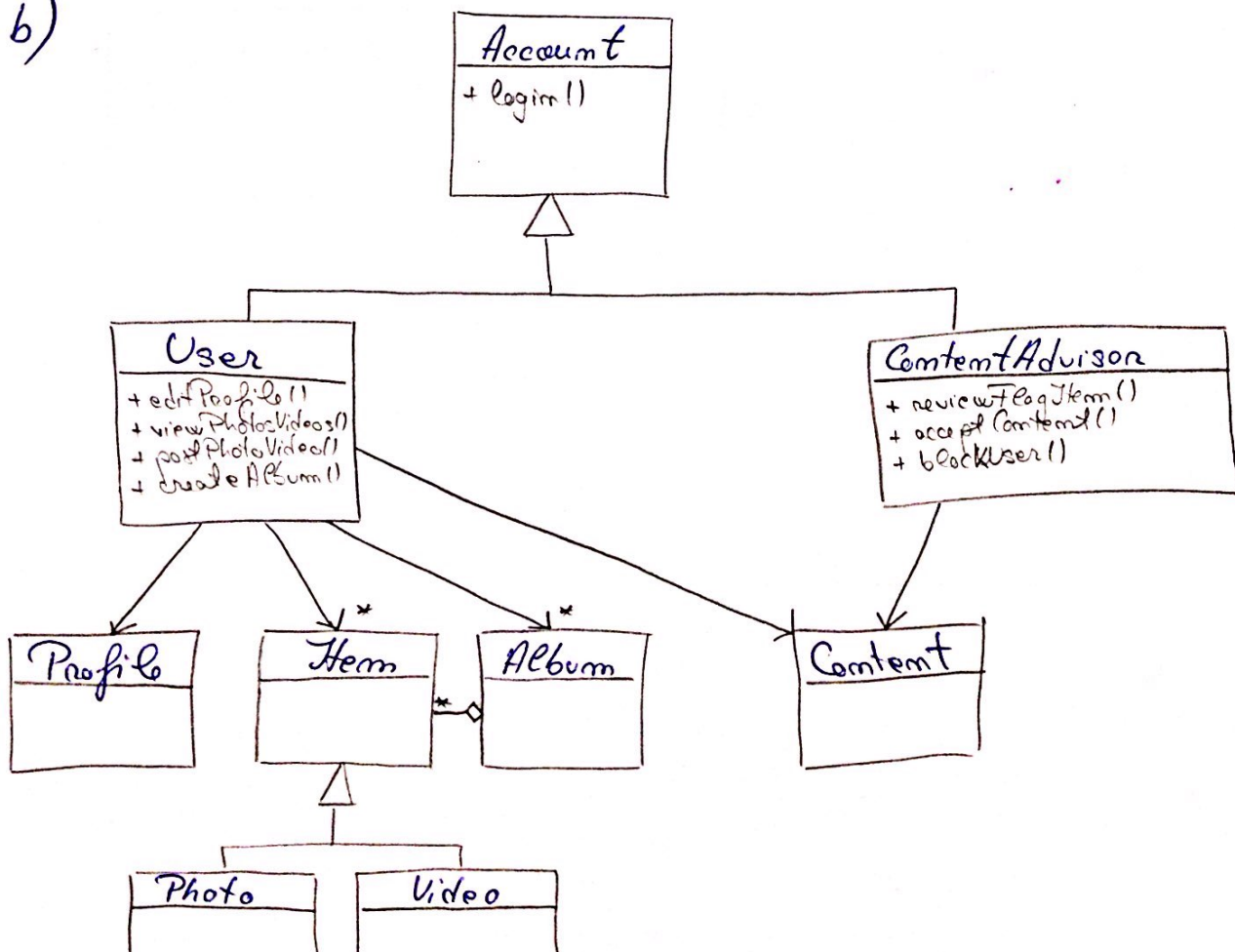


# P5 : PHOTO/VIDEO SHARING WEB APPLICATION

a)

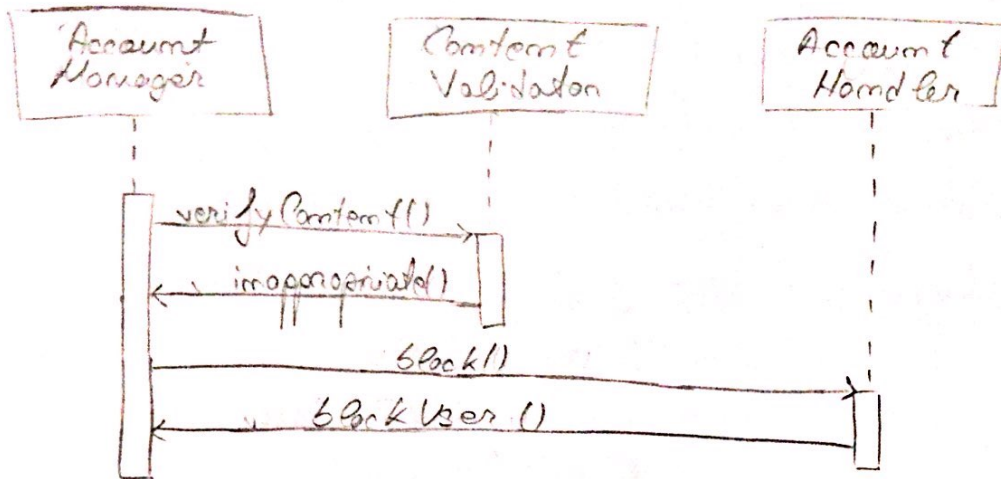


b)

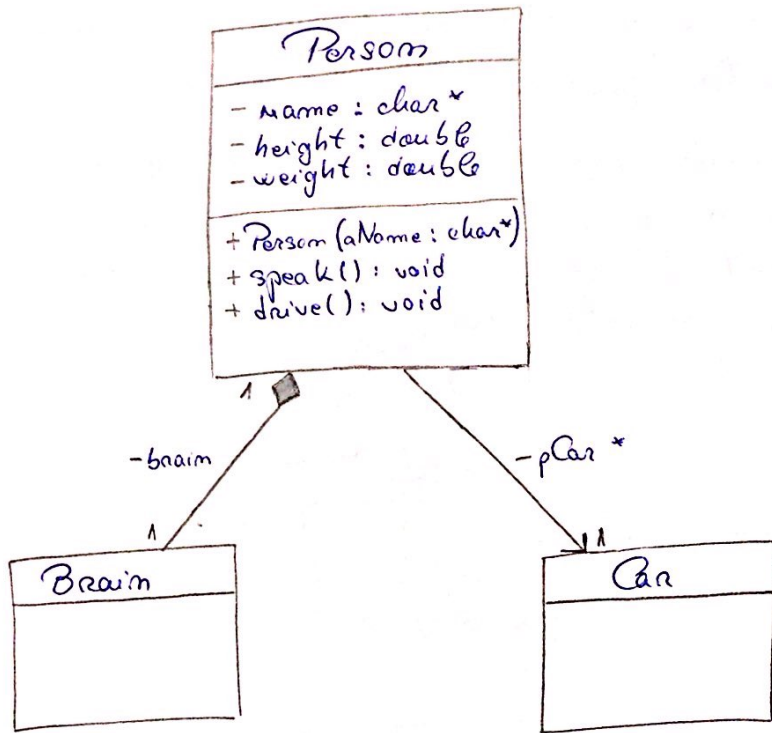




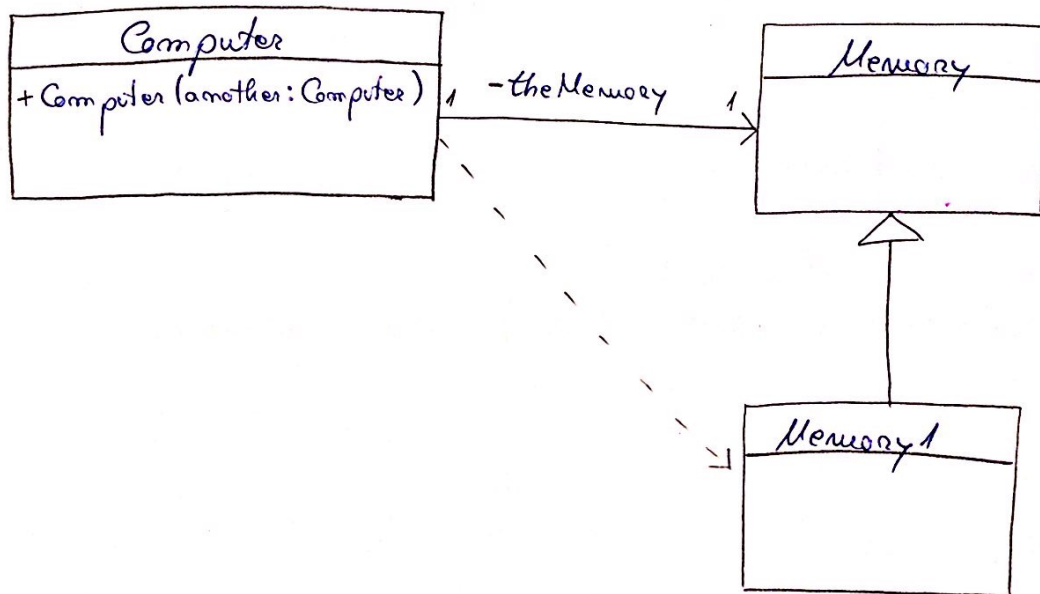
### c) Sequence Diagram - Block User



PG : PERSON



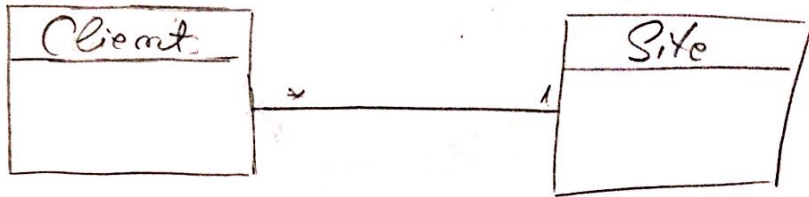
PG : COMPUTER



## PS: DESIGN PATTERN: OBSERVER

Acest design-pattern se folosește când există un subiect urmărit de mai multe persoane (subscribers).

Exemplu: Persoanele care sunt interesate de anumite oferte de pe un site se abonează la acel site și sunt notificate dacă apar modificări în cea ce privește ofertele.



```
public class Site
{
    private List<Client> clients = new ArrayList<Client> ();
    private int offer;

    public int getOffer() { return offer; }
    public void setOffer(int i) { this.offer = i;
                                notifyAllClients(); }

    public void attach(Client client)
    {
        clients.add(client);
    }

    public void notifyAllClients()
    {
        for (Client client : clients) client.update();
    }
}
```

```
public class Client
{
    protected Site site;

    public Client(Site s)
    {
        this.site = s;
        this.site.attach(this);
    }

    public void update()
    {
        System.out.println("Offer : " + site.getOffer());
    }
}
```