# Software Engineering

Books or notes are **not** allowed.
Write only on these sheets. **Concise** and **readable** answers please.

Surname, name, matricola ______________________________________________

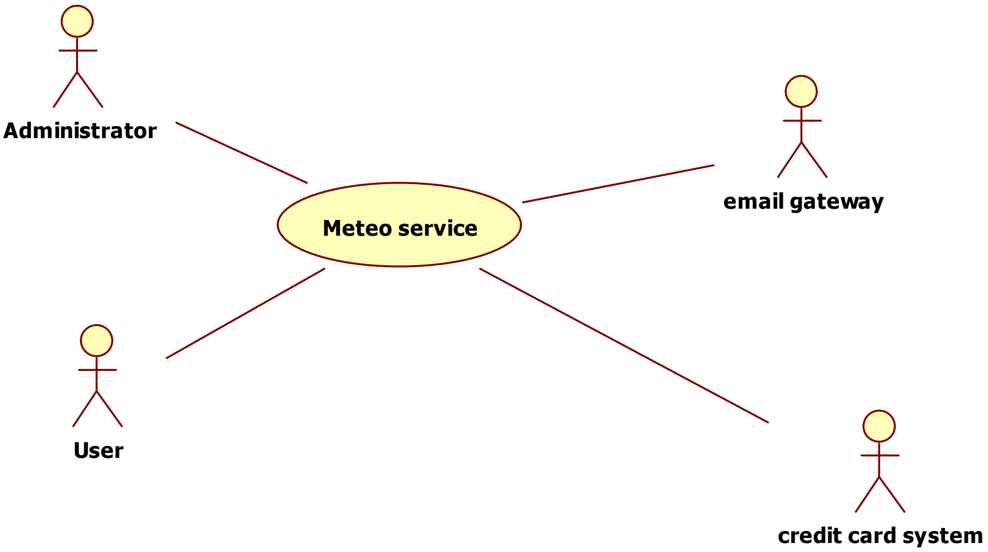
*Meteo Service*
A company wants to offer a sophisticated meteo service, based on a network of proprietary sensors. Each sensor collects weather data (temperature, pressure, humidity, wind, rain), stores it locally in digital form. Each sensor is connected via GPRS to the cellular network, and sends weather data every 15 minutes to a server. Considering Italy, around 1000 sensors are installed, in 20 regions. The server receives weather data and produces a current weather status report, every 15 minutes, and a weather forecast (with horizon 1, 3, and 5 days) every 3 hours. Some services are offered for free, some request a fee. For instance the 1 day weather forecast could be offered free, the 3-5 days forecast under payment of a time based (weekly, monthly yearly) fee (and this requires subscribing to the service to obtain an account, paying for it, etc. ). The company can configure as needed which services are free, which require a subscription.
The service is available on the web, and is accessible through any web browser.

In the following you should analyze and model the meteo service. This includes software, server, sensors.

1 (15 points) – a. Define the context diagram (including relevant interfaces)

| Actor | Physical interface | Logical |
|---|---|---|
| user | PC or smartphone | GUI |
| administrator | PC | GUI |
| Email gateway | Internet connection | Smtp, POP |
| Credit card system | Internet connection | https |

The sensors, as stated in the text, are part of the system. See system design.



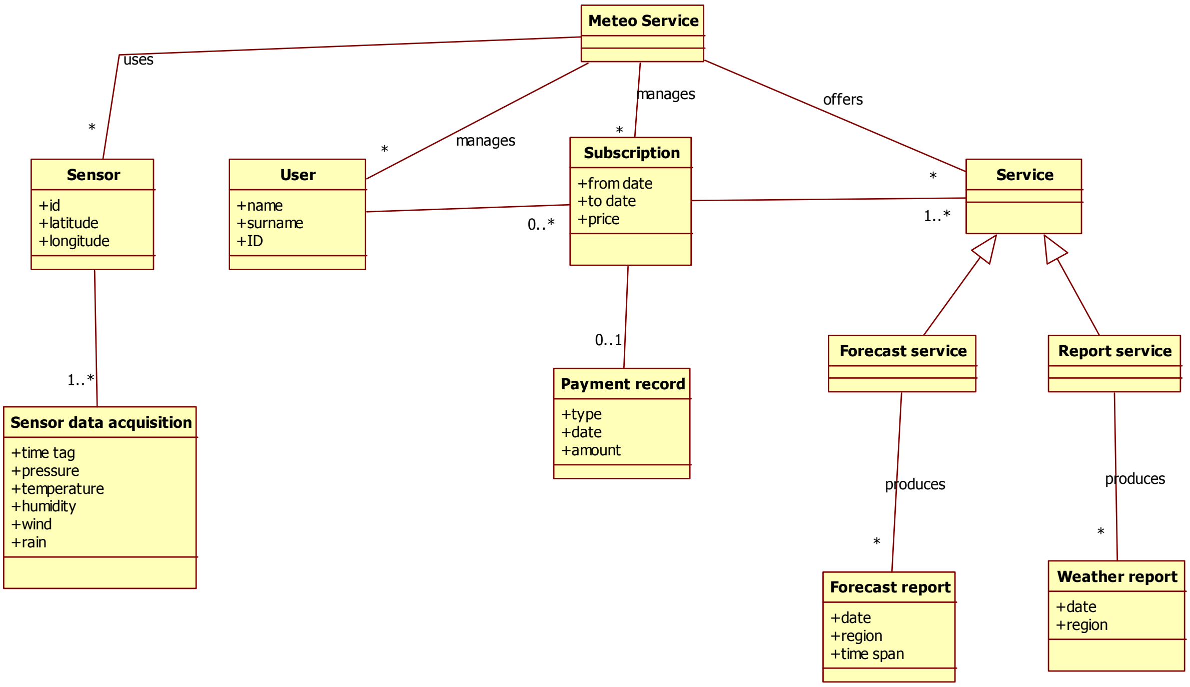
List the requirements in tabular form

| ID | Type (Functional Non Functional) | Description |
|---|---|---|
| 1 | F | Weather |
| 1.1 | | Produce weather report every 3 hours |
| 1.2 | | Produce weather forecast (on 1,3,5,days) every 3 hours |
| 1.3 | | Show weather report (national, by region, by spot) |
| 1.4 | | Show weather forecast (national, by region, by spot) (on 1,3,5,days) |
| 2 | F | Users, subscriptions, payments |
| 2.1 | | Add , modify, delete user |
| 2.2 | | Add, modify, delete subscription |
| 2.3 | | Login, logout |
| 2.4 | | authorize access to service |
| 2.5 | | Link subscription to user, to service |
| 2.6 | | Link payment to subscription |
| 2.7 | | Manage payment (credit card) |
| 3 | F | Sensors |
| 3.1 | | Add, modify, delete, configure sensor |
| 3.2 | | Acquire weather data from sensors every 15 min |
| 3.3 | | Verify status of sensor |
| 4 | F | Service |
| 4.1 | | Add, modify, delete service |
| | | |
| | | |
| | NF | Scalability. Number of users up to  X |
| | NF | Scalability. Number of sensors up to Y |

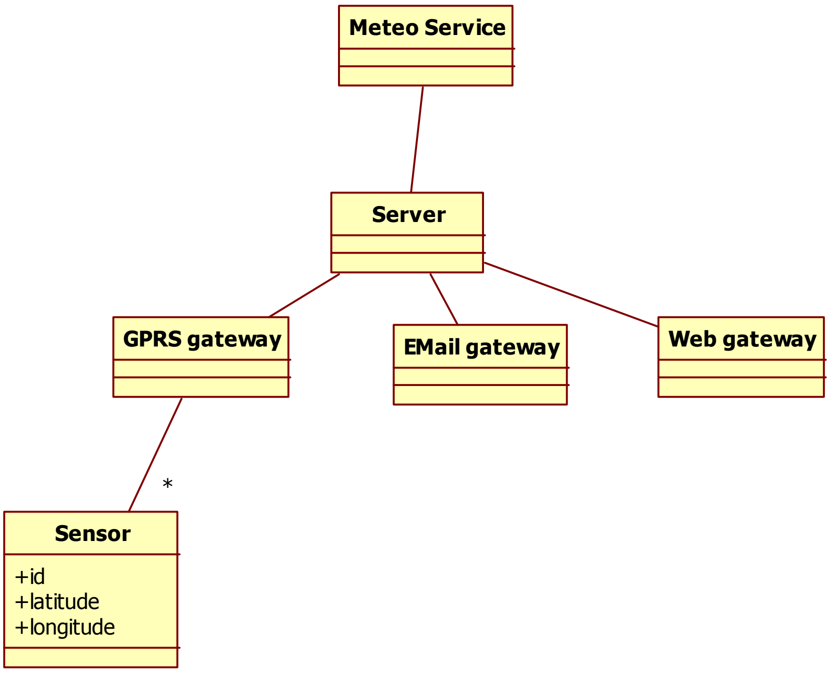
| | | |
|---|---|---|
| | NF | Performance. Response time for all functions < 0.5 sec |
| | NF | Performance. Sensor should be able to store locally weather data equivalent to at least 15 minutes of data collection |
| | NF | Reliability. Availability > 99% |

Many students have written requirements like ' server reads data from sensor, server stores data in db. However this concerns design decisions and should be avoided.
Another common mistake is having subclasses for free / paying services. This does not allow t change a service real time (from free to paying or viceversa)

Define the  key concepts and entities and their relationships (UML class diagram) for the system



Define the system design (key hardware and software components of the system, UML class diagram) for the system.  Describe the key design choices.



Define one scenario describing a user that accesses the 3 day forecast (requires a subscription)
Precondition: User U is registered and has subscribed subscription S (that includes right to service 3 day weather forecast)

Postcondition:3 day weather forecast is presented to user U

| Step | Description | Req ID |
|---|---|---|
| 1 | U logs in | 2.3 |
| 2 | U requests 3 day forecast | |
| 3 | System authorizes access to service | 2.4 |
| 4 | System shows 3 days forecast | 1.4, 1.2 |
| 5 | | |

2 (8 points) -Define black box tests for the following function, using equivalence classes and boundary conditions.

double rechargeCard( double *amount*)

the function computes the amount to recharge on a telephone card. *Amount* is the amount paid by the user, in euros. Amount must be a multiple of 5. The function

- Truncates amount to the lowest multiple of 5 if *amount* is not exactly a multiple of 5
- After this truncation (if needed) it returns
    - The same amount if *amount* < 10 euros.
    - Amount plus a 10% bonus if *amount* >= 10 euros
    - Amount plus a 20% bonus if *amount* >= 100 euros

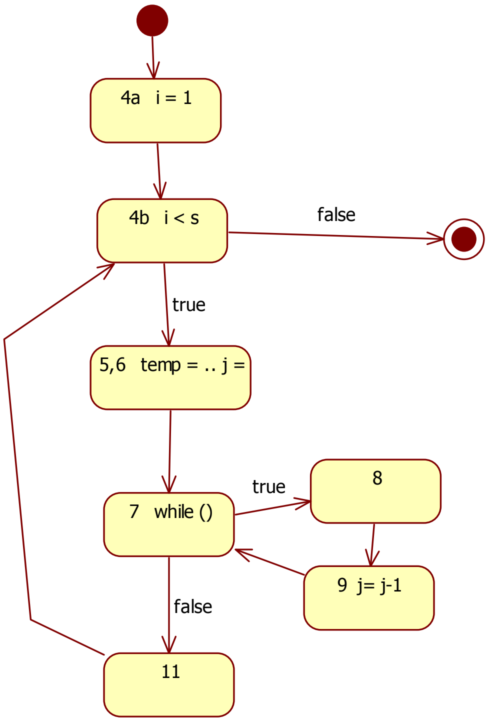
Ex:

rechargeCard ( 5) → 5
rechargeCard ( 5,5) → 5
rechargeCard ( 10,5 ) → 11
rechargeCard ( 20 ) → 22
rechargeCard ( 200 ) → 240

| Amount range | Amount multiple of 5 | Valid / Invalid | Test case |
|---|---|---|---|
| ]minint, 0[ | Y | I | |
| | N | I | |
| [0, 10[ | Y | V | |
| | N | V | |
| [10, 100[ | Y | V | |
| | N | V | |
| [100, maxint[ | Y | V | |
| | N | V | |

3 (7 points) – For the following function define the control flow graph, and define test cases to obtain the highest possible node coverage, edge coverage, multiple condition coverage, loop coverage, path coverage.
For the test cases, **write only the input value**.

```
1 int sort(int[] a, int s){
2  int i, j, temp;
3
4  for(i=1;i<s;i++){
5      temp=a[i];
6      j=i-1;
7      while((temp<a[j])&&(j>=0)){
8      a[j+1]=a[j];
9          j=j-1;
10      }
11      a[j+1]=temp;
12   }
13
14 }
```



| Coverage type | Number of test cases needed to obtain 100% coverage | Coverage obtained with test cases defined (%) | Test cases defined |
|---|---|---|---|
| Node | 1 | 100% | T1{3,2} |
| Edge | 1 | 100% | T1 |
| Multiple condition (line 7) | 4 | 75% | TT yes<br>TF not feasible<br>FT yes<br>FF yes |
| Loop - for line 4 | 3 | 100% possible | T1, one loop<br>T2{2} zero loop<br>T3{3,2,1} more loops |
| Loop - while line 7 | 3 | 100% possible | T4{2,3} Zero loop<br>T3 more loops<br>T1 one loop |
| Path | Very high, limited by length of array (value of s) | practically unfeasible | |

4 (1 points) – Describe briefly the MVC pattern

5 (1 points) –  Describe briefly the inspection of a requirements document

6 (1 points) – In the context of configuration management, what is a baseline?

7 (1 points) – In the context of project management, give the definition of ‘slack time’