

# Software Engineering

Books or notes are **not** allowed.

Write only on these sheets. **Concise** and **readable** answers please.

Surname, name, matricola \_\_\_\_\_

## *Gym company*

FITFIT is a company that manages many fitness centers, in many cities.

FITFIT aims to be a digital gym, where most paperwork is eliminated.

Anyone can become a customer by paying a fee that gives access to FITFIT facilities (any facility) for a certain amount of time (day, month, year).

Payments can be made only online, using a web site.

Access to a center is controlled automatically via turnstiles, in two ways. Using an RFID card, or using a NFC enabled smartphone. The card has to be purchased online (on the same website for payments) and is sent to the customer via courier.

Remark that also employees of FITFIT are granted access in the same way.

Access to the facilities requires not only a payment, but also a medical certificate. A scanned version of the certificate has to be uploaded on the web site. A certificate has a defined duration (one year or less).

Not only entrance is through the turnstile, but also the exit.

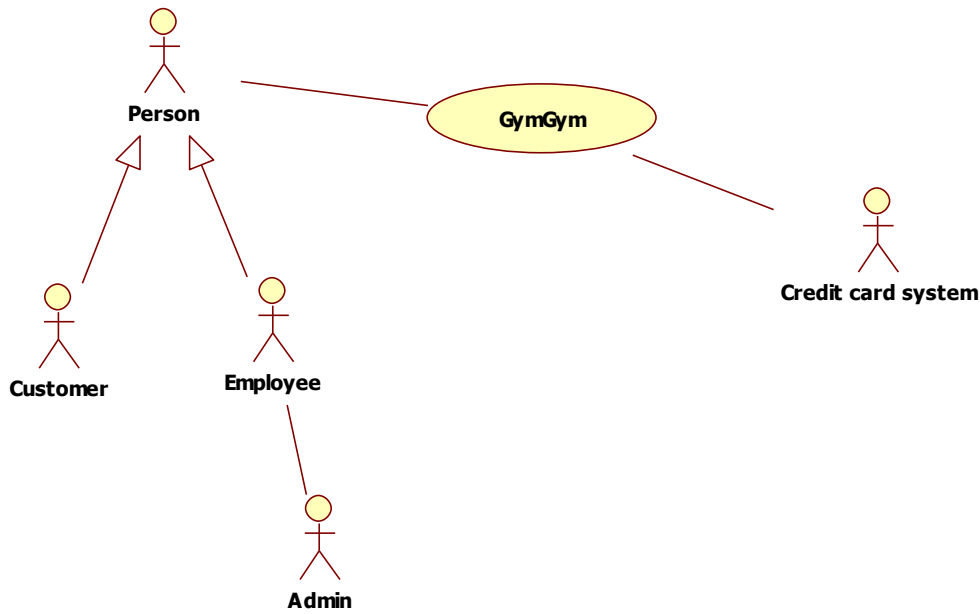
In the following you should analyze and model a client server application to support the FITFIT company.

1 – a. Define the **context diagram** (including relevant interfaces- remember this must be consistent with System design requested later in 1-f )

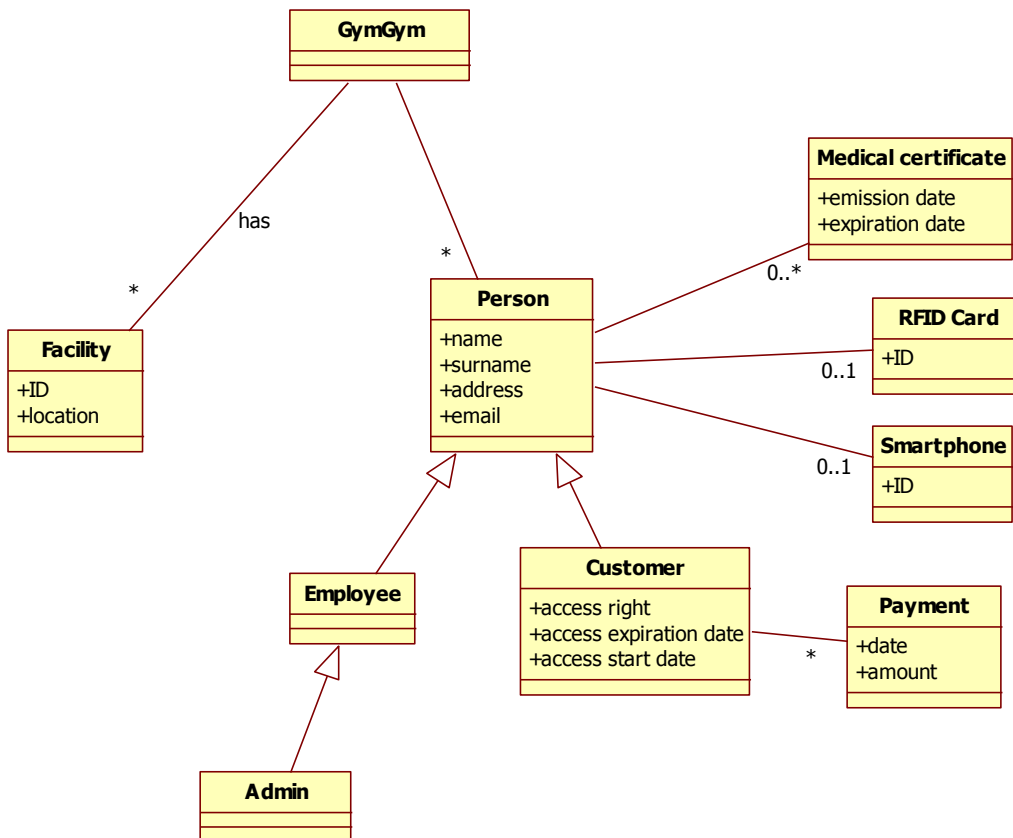
Actor	Physical interface	Logical interface
Customer	Smartphone /PC Rfid card / nfc	GUI
Employee	Smartphone /PC Rfid card / nfc	GUI
Credit card system	Internet connection	Dedicated protocol (do / undo payment on a credit card)
Administrator	PC	GUI
Card producer	Internet connection	Dedicated protocol (do /undo / modify order for a card dedicated to a certain person)

The card producer could be connected digitally, or not (this was not specified in the text). In case it is connected digitally it should appear in the context diagram (physical interface = internet connection, logical connection = dedicated protocol (do /undo / modify order for a card dedicated to a certain person))

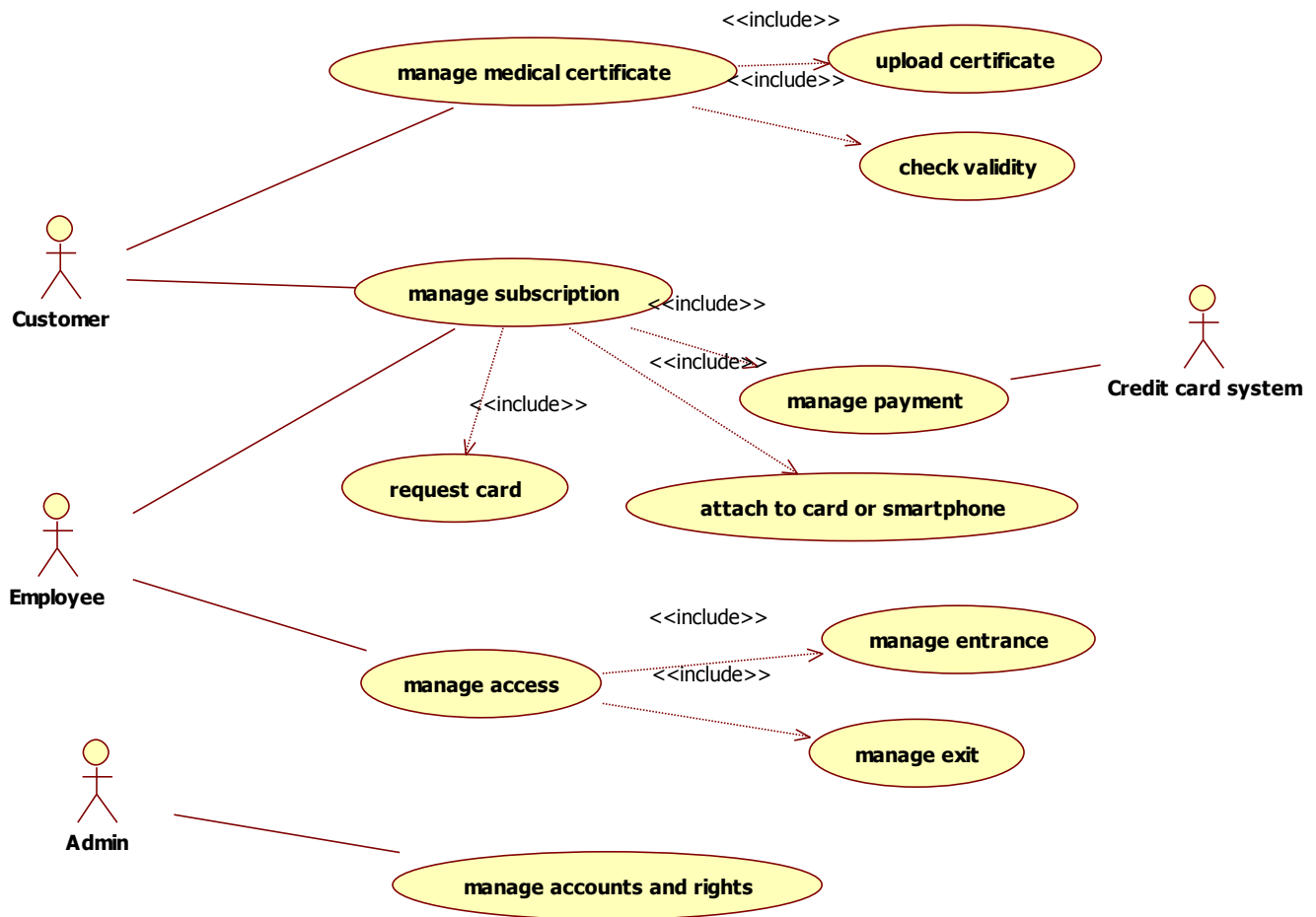
If the turnstiles are considered part of the system they should appear in the system design but not in the context diagram. Viceversa if the turnstiles are considered outside the system they should appear in the context diagram and not in the system design.



1-b Define the **glossary** (key concepts and their relationships) (UML class diagram) for the application



1-c Draw the Use Case Diagram for the application. For each Use Case give self-explainable long names, or a short textual description



1-D List the **NON functional requirements** that you deem important for the application

ID	Description
1	Privacy. Data of a customer must not be visible to other customers
2	Performance – response time of all functions < 0,5 sec
3	Usability – any customer with at least one year experience in using a smartphone must be able to use all functions with no training in less than 1 minute

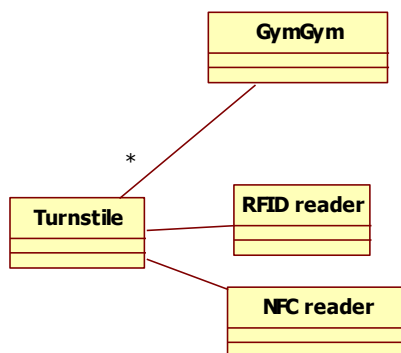
1-e Describe below the scenario specific to a customer who accesses a facility of FITFIT.

Precondition: customer has card, customer has a valid subscription, customer has valid medical certificate

Postcondition: customer is inside facility

Step	Description
1	Customer presents card C at turnstile T
2	System reads card C, finds attached customer CU
3	System checks subscription expiration date $\geq$ today
4	System checks medical certificate attached to CU, expiration date $\geq$ today
5	System opens turnstile T

1-f Define the system design model (using UML class diagram)



2 (7 points) -Define black box tests for the following function, using equivalence classes and boundary conditions.

The function receives the grades of a student on her courses (for simplicity exactly six grades for six courses are considered) and computes the average. Grades can be from 18 to 30, or 30Laude == 33. The average is computed excluding the best and worse grade.

```
double average( int grade1, int grade2, int grade3, int grade4, int grade5, int grade6);
```

ex.  $\text{average}(30,18,26,28,27) \rightarrow 27$  (computed excluding 30 and 18,  $(26+27+28)/3 = 27$ )  
 $\text{average}(25,24,28,27,26) \rightarrow 26$  (computed as  $(25+26+27)/3$ )

partitions to be combined: [minint, 17] [18, 30] [31, 32] [33] [34, maxint]

boundary: try edges (17, 18, 30, 31, 32, 33) try with all equal grades, try with two or more max grades equal, two or more min grades equal

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**.

Write control flow graph here

```

1      int compute_tax(int wage) {
2
3          int ranges[] = {6000, 18000, 36000};
4          int amount_due = 0;
5          int level=0;
6
7          for(int i=0;i<3;i++){
8              if(wage>ranges[i])
9                  level++;
10         }
11         if(level==1)
12             amount_due = 500;
13         else if(level == 2)
14             amount_due = 1500;
15         else if(level == 3)
16             amount_due = 3000;
17         return amount_due;
18     }

```

Coverage type	Number of test cases needed to obtain 100% coverage	Coverage obtained with test cases defined (%)	Test cases defined
Node	3	100%	T1,T2, T3
Edge	4	100%	T1,T2, T3, T4
Multiple condition line 8	Not a multiple, 2 are enough (even only 1 because of for cycle line 7)	100%	T1
Loop line 7	3, but not controllable	33%	Any input
Path	For in line 7, in theory $2^3$ , in practice 4 only Line 11 to 16: 4 Should be $4*4$ However paths in the two parts of the function are correlated, so overall 4 paths Feasible		

Write test case ID (t1, T2 ..) in the rightmost column, and test cases here

4 (1 points) – Describe the key concepts of Scrum

See slides

5 (1 point) – Given a set of functional requirements, often many designs are possible. How to select one design option versus other ones?

Given that all designs satisfy the functional requirements, selection should be done considering NON functional requirements

6 (1 point) – In the context of configuration management, what is the ‘copy modify merge’ approach and what are its pros and cons?

See slides

7 (1 point) – What is the definition of ‘exhaustive testing’ ? Is it possible?

Try all possible test cases (for a function, a class, a program). Normally not feasible due to virtually infinite number of test cases

8 (1 point) – What is ‘mutation testing’ and what is its goal?

Evaluate how a test suite is good by injecting errors (mutations) in a program and verifying how many mutations are caught by the test suite