Software Engineering

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

Surname, name, matricola _____

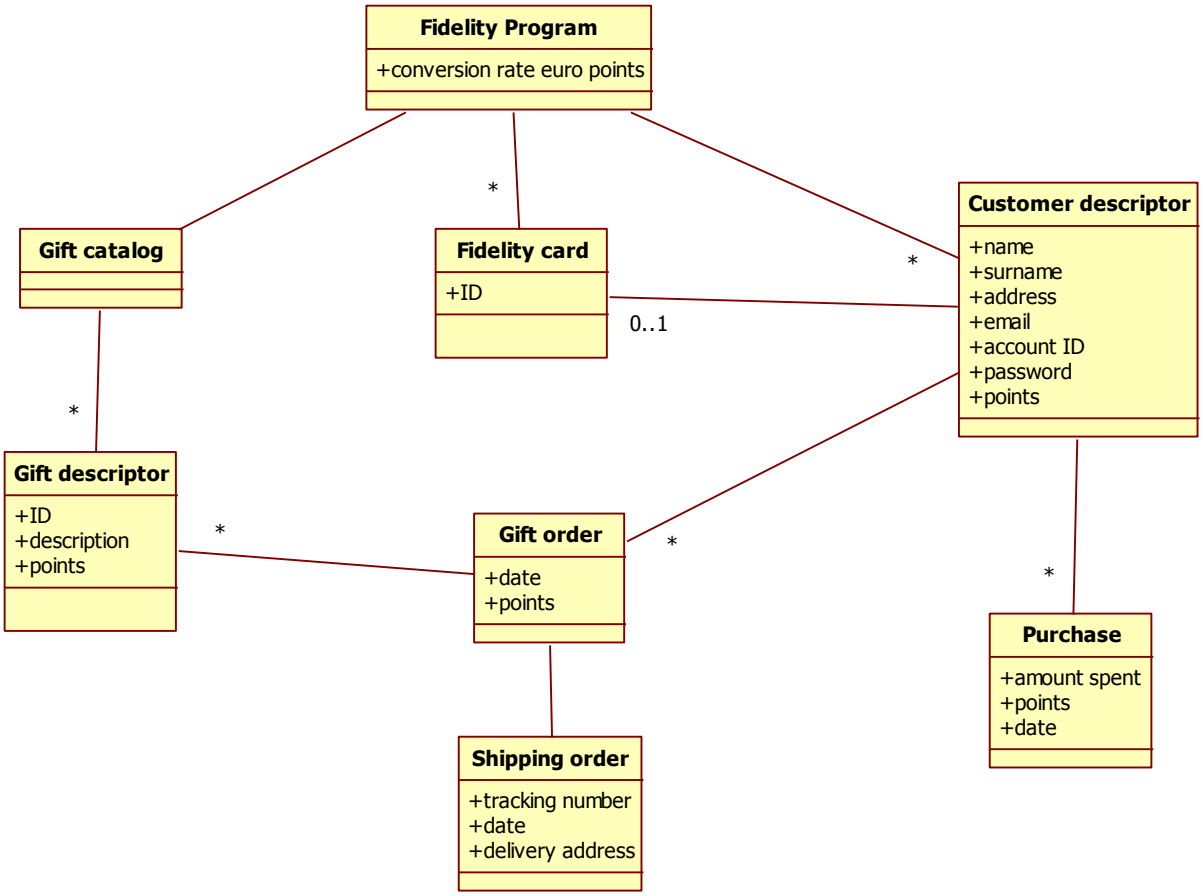
Fidelity program
Many companies offer fidelity programs, where customers obtain points each time they acquire a good / service, and can later exchange points for gifts.
Let's focus on the case of a retailer, with hundreds of stores in Europe.
A customer can enroll in the fidelity program, either online, accessing a web site, or in person, at a desk in a retail store.
In both cases the customer receives a card, and credentials to access his/her account on the web, via PC or app on smartphone.
At each purchase (both online or in a store) the customer can show the card and obtain points, at a certain rate (ex 10 euro – 1 point).
The customer, at any time, can access online his account, check the history of purchases and corresponding points. Further, she can consult the list of available gifts, and possibly order one. A gift requires a certain number of points, that are deducted when the gift is ordered. Gifts can be received at home or collected in a retail store. In both cases the customer can follow the state of delivery online, using a tracking number. Delivery is subcontracted by the retailer to a logistic company.
Ordering a gift may also involve a payment, by credit card.

In the following you should analyze and model the application that supports a fidelity program for the retailer company.

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

Actor	Physical interface	Logical interface
Customer	PC / smartphone	GUI
Clerk of retailer	PC	GUI
Administrator of retailer	PC	GUI
Credit card circuit	Internet connection	Web service (for payment)
Logistic company	Internet connection	Web service (parcel tracking)

Define the glossary (key concepts and their relationships) (UML class diagram) for the application



A customer could have more than one fidelity card (in case of loss, or expiry), however this case is not considered.

List the requirements in tabular form (do not forget to list important NF requirements)

ID	Type (Functional Non Functional)	Description
1	F	CRUD customer descriptor and related account information
2	F	Login, Authenticate and authorize customer, logout
3	F	CRUD Gift descriptor
4	F	CRUD Gift order
5	F	CRUD Purchase (this function could be already implemented by retail store management software)
6	F	CRUD Shipping order (actual tracking may happen on the web site of the logistic company, or here via a proxy to web site of logistic company)
7	F	Interact with logistic company (send shipping order, receive tracking number) Remark that the tracking function is offered by the logistic company – the retailer may redirect the customer to the logistic company web service
8	F	Interact with credit card system (send payment information, receive feedback)
	NF	Privacy, user should R/W only his her data. Credit card info should be canceled after usage
	NF	Security, payments should be started only by authorization of card owner
	NF	Performance, all functions (except payment) should complete in <1sec

List the user level goal Use cases

Customer or clerk: register
Customer: Order gift (Browse gifts, select gift, order)
Customer: Track gift delivery
Clerk: Manage gifts (add, modify)
Admin: Manage customer accounts (add, delete, modify)

Select one use case from the point above and describe it

Name	Order gift
Scope	Fidelity program management
Level	User goal
Intention	A customer wants to use her points to get a gift
Primary actor	Customer
Precondition	Customer has a card and an account (has subscribed to the fidelity program), has enough points to obtain a gift
Main success scenario	1 Customer logs in, 2 browses available gifts, 3 selects gift. 4 System checks if customer has enough points for the gift. 5 System asks shipping address. 6 System confirms gift transaction and detracts points to the customer. 7 Customer logs out 8 System starts procedure to package and ship gift.
Extensions	3b gift not available 4a not enough points 4b Payment needed 4c payment failed

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

double telCallCost(int startTime, int endTime , int zone)

This function computes the cost of a phone call. startTime and endTime are in seconds over a 24 hours day. The cost is based on a cost per second rate, that depends on the zone called. Besides, calls longer than 5 minutes get a 10% discount

The cost rates are:

- Zone 1: 1 cent /sec
- Zone 2: 2 cent /sec
- Zone 3: 3 cent /sec

Ex. telCallCost (1, 10, 1) → 9
telCallCost (1, 10, 2) → 18
telCallCost (1, 10, 3) → 27
telCallCost (1, 301, 1) → 300 - 10% = 270

60x60x24 = 86400 assuming seconds start from 0
Another assumption; calls can start on one day and end the next day – this means endTime < startTime is acceptable – this means also that max duration of a call is implicitly <= 24 hours

Max number combinations 3x3x3x2x2x2

startTime range	endTime range	Zone range	Endtime> startTime	Call lenght	Test case
[minint, 0[	-	-	-	-	T(-1, 10, 1; err)
[0, 86400[	[minint, 0[	-	-	-	T(1, -1, 1; err)
	[0, 86400[	[minint, 0]	-	-	T(1,1,-2; err)
		[1,3]	N	<5 min	T(86351, 1, 1; 50) Tb(86301, 1, 1; 100)
				>5min	T(86101, 1, 1; 270)
			Y	<5min	T(2, 102, 1; 100)
				>5min	T(0, 300, 1; 270)
		[4, maxint]	-	-	T(1,3,5; err) Tb(1,3,4; err)
	[86400, maxint]	-	-	-	T(1,90000,1; err) Tb(1,86400, 1; err)
[86400, maxint]	-	-	-	-	T(90000, 1, 1; err)

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 void heapsort(int heap[], int no){
2               int i, j, c, root, temp;
3   for (j = no - 1; j >= 0; j--)
4    {
5        temp = heap[0];    heap[0] = heap[j];
6          heap[j] = temp;        root = 0;
7        do
8       {
9            c = 2 * root + 1;
10           if ((heap[c] < heap[c + 1]) && c < j-1)
11               c++;
12           if (heap[root]<heap[c] && c<j)
13           {
14               temp = heap[root];  heap[root] = heap[c];
15                    heap[c] = temp;
16           }
17           root = c;
18       } while (c < j);
19    }
20   }
```

Coverage type	Number of test cases needed to obtain 100% coverage	Coverage obtained with test cases defined (%)	Test cases defined
Node	1	100	T1
Edge	1	100	T1
Multiple condition line 10	4 or less	100	T1
Loop line 3	3	100	T1 many T2 zero T3 one
Path	Depends exponentially on no, not feasible		

T1: heap[5,2,3, 6], 4
T2: heap[], 0
T3: heap[1], 1
T4 1,2 - 2 1

4 (1 points) – What are Function Points used for? How are they defined?

See slides

5 (1 point) – What kind of problem ‘traceability’ identifies?

Linking requirements to design components / functions to code components / functions to test cases

6 (1 point) – A defect on a sw product signaled by a customer has been allocated to a developer, who has found the corresponding error. What happens next?

Fix the defect, rebuild,test, regression test, decide release to attach the patch to, release

7 (1 point) – Describe the key concepts of the waterfall process

See slides

8 (1 point) – Describe shortly the ‘MVC’ Design pattern

See slides