Software Engineering

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

Surname, name, matricola ____

Motorcycle maintenance management
During the lifetime of a motorcycle many maintenance and repair interventions are needed, some scheduled, some not. It is important for the owner to keep track of these interventions. On the vehicle producer side, it is important to keep track of these interventions both for safety issues (recalls due to defects) and for customer care.

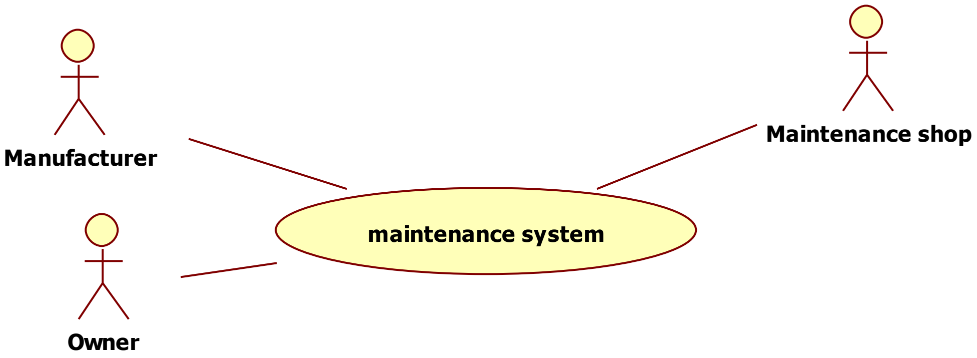
In the following you should analyze and model the application that supports web based maintenance management.
Key roles to be considered are the owner, the manufacturer (defines maintenance schedules and jobs), the maintenance shop (records maintenance interventions).

Key high level functions to be considered are:

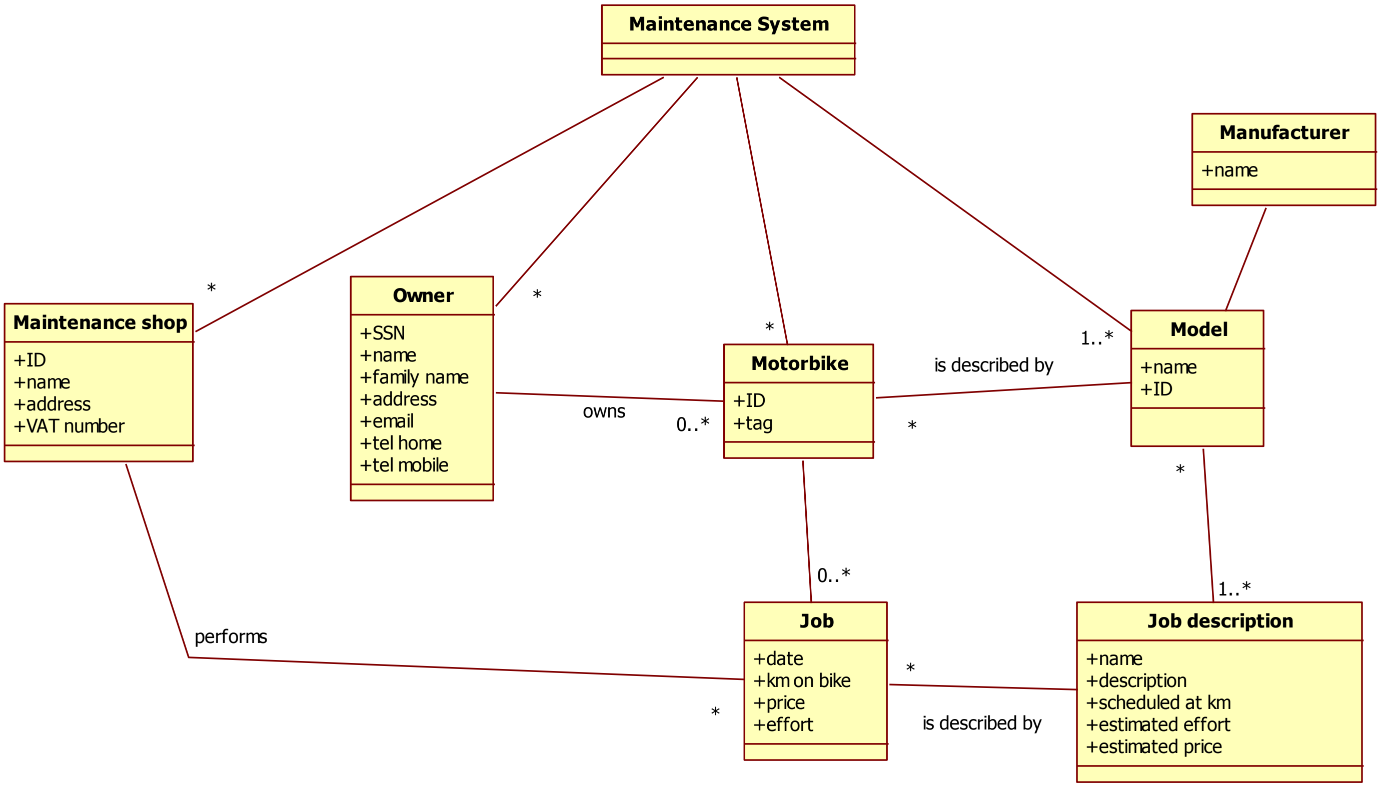
- Define the regular maintenance jobs (typically performed by vehicle manufacturer). Remark that jobs (both type and schedule) depend on the motorcycle model.
- Record a set of interventions on a motorcycle, due to normal maintenance or not (typically performed by a maintenance shop). Record also effort spent and cost for each job, to be used both for owner records and for payment.
- Remind owner about a scheduled maintenance
- Browse and analyze jobs for a motorcycle (typically performed by owner or maintenance shop)

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

Actor	Physical interface	Logical interface
Owner	PC / smartphone	GUI
Manufacturer	PC	GUI
Maintenance shop	PC	GUI



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



1-c List the **requirements** in tabular form (do not forget to list important NF requirements)

ID	Type (Functional Non Functional)	Description
1	F	Motorcycle management
1.1 1.2 1.3	F	Add a motorcycle, modify a motorcycle, attach motorcycle to owner
2	F	Motorcycle Model management
2.1 2.2	F	Add a model, modify a model
3	F	Job description management
3.1 3.2	F	Add job description, attach job description to motorcycle model
4	F	Motorcycle Owner management
4.1 4.2	F	Add owner, modify owner
5	F	Motorcycle management
5.1 5.2 5.3	F	Add a motorcycle, attach motorcycle to motorcycle model, attach motorcycle to motorcycle owner
6	F	Job management
6.1 6.2 6.3 6.4	F	Add job, modify job, attach job to job description, attach job to motorcycle
7	F	Scheduled maintenance management
7.1 7.2	F	Compute scheduled maintenance job for a motorcycle, send reminder to owner
	NF	Performance. All functions should have response time < 0.5sec
	NF	Security. All functions and data about Motorcycle Model accessible only by Motorcycle manufacturer (Similarly for other functions)

1-d Define one **scenario** describing the recording of job J on motorcycle M.

Precondition: Owner O owns vehicle M, J feasible on M

Postcondition: job J performed and recorded on M

Step	Description	Req ID
1	O brings M to maintenance shop	
2	Job J is performed on M. shop employee look for M, finds it	5.1
3	Employee looks in job descriptions attached to M's model	3.1
4	J is recorded on M	6.1
5		

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

boolean checkAmount (String amountAsString, double amountAsNumber)

On a check the amount to be paid is usually written as a number and as text. In European checks the amount in number should be written as 100,10 or 100.10 , the amount as text should be written as hundred/10 or hundred euro /10 . Of course the amount should be positive, and <= 10000 euros. And finally the amount as expressed in the two forms should be equivalent.
The function above is used by a bank to verify incoming checks. Assume that amountAsString and amountAsNumber are obtained by an optical character recognition system that processes paper checks. Assume also that amountAsString never contains capital letters, and that the amount is expressed in English only.

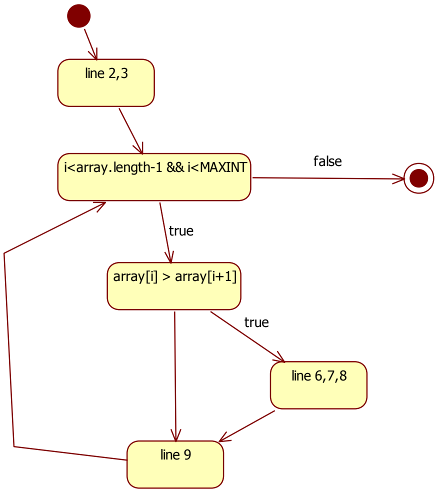
Ex. checkAmount (“hundred/10”, 100.10) → true
checkAmount (“hundred /00”, 100.10) → false
checkAmount (“fendred /10”, 100.10) → false

amountAsNumber	AmountAsString	Number == string	Valid	Test cases
[mindouble, 0.00[	valid	Y	N	
		N	N	
	Not valid, no English	-	N	
	Not valid, format	-	N	
	Not valid, no number	-	N	
[0.00, 10000.00]	valid	Y	Y	
		N	N	
	Not valid, no English	-	N	
	Not valid, format	-	N	
	Not valid, no number	-	N	
]10000.00, maxdouble]	valid	Y	N	
		N	N	
	Not valid, no English		N	
	Not valid, format		N	
	Not valid, no number		N	

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 f(int[] array){
2  int temp;
3  int k= array.length -1;
4  for (int i=0; i<array.length-1 && i<MAXINT ; i++){
5      if (array[i] > array[i+1]) {
6         temp = array[i];
7         array[i] = array[k];
8         array[k] = temp;
9      }
10   }
11 }
```



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 4	4 or less	50% for sure T4 and T5 may be unfeasible	T1: TT and FT T4: TT and TF T5: TT and FF
Loop line 4			T1, T2, T3
Path	$2^{array.lenght-1}$	Since length of array can be very high coverage is close to zero	

Write test case ID (T1, T2 ..) in the rightmost column, and test cases here

T1({2,1})

T2({1})

T3 ({2,1, 3})

T4 an array longer than MAXINT

T5 an array as long as MAXINT

4 (1 points) – Function A calls function B, that calls function C. You want to apply bottom up integration. How do you proceed?

Test C, test B+C, test A+B+C

5 (1 point) –Considering GIT, what are the three project sections that it defines, and how are they used?

See slides

6 (1 point) –– What is the typical lifecycle for a change? (draw states and transitions)

See slides

7 (1 point) –What measures can be used to evaluate the quality of software?

Fault density, MTBF, user satisfaction (questionnaire)

8 (1 point) – Describe shortly the Adapter Design pattern

See slides