

Software Engineering – 02GSP

Books or notes are **not** allowed.

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

Surname, name, matricola _____

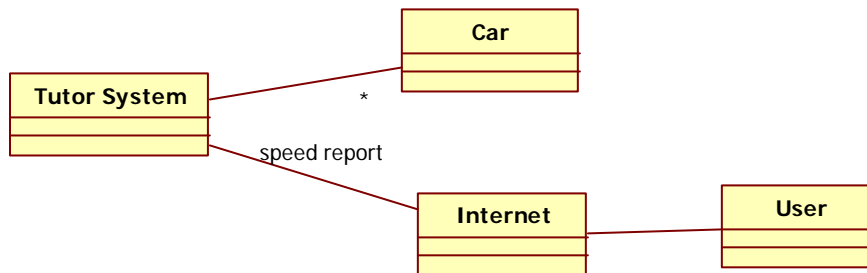
Speed monitoring system on highways (Tutor)

Traditional speed monitoring systems measure the instantaneous speed of a car on a street, following the car for few meters with radar based appliances.

The idea of the Tutor system instead is to measure the average speed of a car on a long distance (in the order of 1-10 kilometers). This is made by measuring the time when a car passes at point A, the time when the same car passes at point B, and doing simple math to compute the average speed. In point A and B on the highway a gate is built with one or more cameras that take pictures of cars passing, triggered by a motion sensor. The pictures, along with the time when they were shot, are sent to a computer based system for processing. The output is the average speed of each car passed through the gates.

In the following you should model this system, including hardware (cameras, computers) and software (notably functions provided).

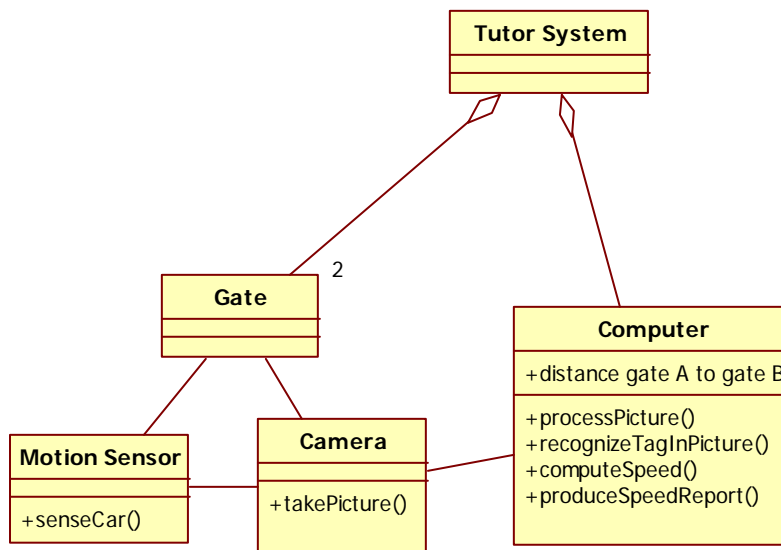
1 (13 points) – a. Define the context diagram (including relevant interfaces)



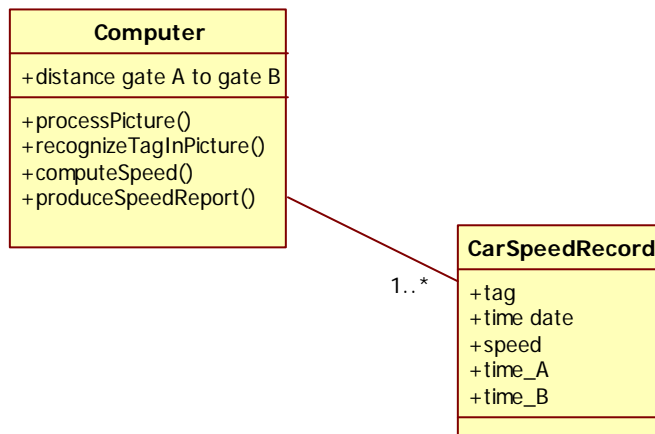
Interfaces: to cars (indirectly via motion sensor and camera), to network connection (to transmit car and speed reports). Speed report format should be described somehow (ex XML file with tag of car, time and date, speed measured). User accesses the system to initialize it, monitor it, and download reports.

Define the system design

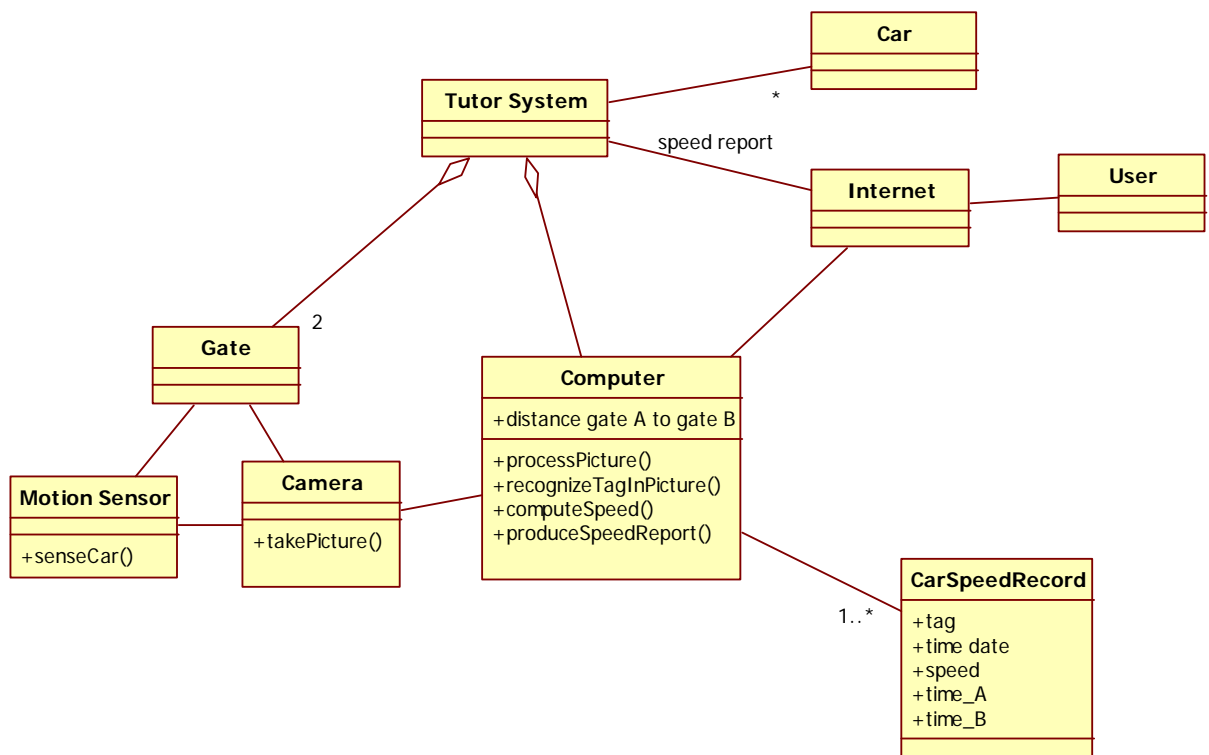
System design covers all parts of the system, both software and not software. Classes Gate, Motion sensor, Camera and Computer represent physical components of the system. Functions on Computer will be implemented by software.



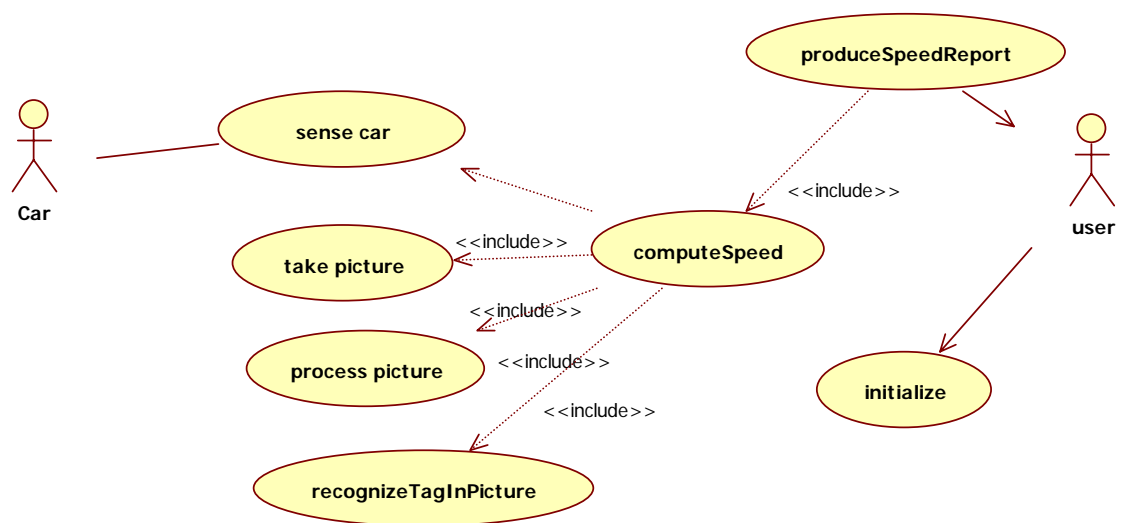
Define the class diagram.
Here we take the software view only



The complete class diagram, considering all parts of the system (hardware and software) follows (it is simply the aggregation of diagrams shown above):



Define the use case diagram



Recognizing the tag of a car in a picture is the most delicate function of the system, both for speed (must be performed before the next car passes) and precision (tag must be identified without errors).

Define one scenario describing a successful measurement of the speed of a vehicle.

Precondition: car with tag_c passes between gates A and B

Postcondition: average speed for car with tag_c computed and stored

Step	Description
1	Motion_A sensor detects motion (car) and triggers camera_A to take picture
2	Camera_A takes picture
3	Camera_A sends picture to computer
4	Computer processes picture to recognize tag, recognizes tag_c, generates CarSpeedReport with time_A and tag_c
5	Motion_B sensor detects motion (car) and triggers camera_B to take picture
6	Camera_B takes picture
7	Camera_B sends picture to computer
8	Computer processes picture to recognize tag, recognizes tag_c, searches CarSpeedReport with tag_c, sets on it time_B
9	Computer computes average speed = distance A – B / (time_B – time_A) and store it in CarSpeedReport of car with tag tag_c

Define one scenario describing an unsuccessful measure.

Precondition: car with tag_c passes between gates A and B

Postcondition:

Step	Description
1	Motion_A sensor detects motion (car) and triggers camera_A to take picture
2	Camera_A takes picture
3	Camera_A sends picture to computer
4	Computer processes picture to recognize tag, tag is unrecognized

2 (8 points) -Define black box tests for the following function

```
double averagePositive (int array[]);
```

The function receives an array of integers, and computes the average, but considering only the positive numbers.

Ex. averagePositive ({1,2,3,4}) → 2.5

Ex. averagePositive ({1,-2,3}) → 2

Criterion	Valid class	Invalid class	Boundary Condition
Number of elements in array	1 and more	zero	1, infinite
Sign of elements	All positives Positives and negatives	All negatives (result is error or zero? To be clarified)	
Type of elements in array	int	Char, string, float, double	
Range or each element in array	Minint to maxint	>maxint <minint	Minint, minint +1, minint -1 Maxint, maxint -1, maxint +1

3 (6 points) – For the following function define the control flow graph, choose the best coverage achievable, and define test cases to obtain it

```
double averagePositive(int[] array){
    double sum = 0; int n = 1;
    for (int i = 0; i<array.length; i++)
        if (array[i] > 0){
            sum = sum + array[i];
            n++;
        }
    return sum/n;}

```

Coverage condition(s) selected: _____

Node coverage is limited.

Condition coverage is better, but it still does not cover the for

Decision coverage in this case is the same as condition coverage

Loop coverage is useful for the for cycle

Path coverage is not achievable

In summary a good approach is condition coverage + loop coverage.

4) (1 point) Describe the lock-modify-unlock change management policy, and its advantages/disadvantages.

User puts lock on CI, user modifies CI, user unlocks CI. Until lock is set, only one user can modify the CI. Advantage: no concurrent, incoherent modifications. Disadvantage: bottlenecks on CI needed by many users, deadlocks if user forgets to unlock a CI

5) (1 points) What is an inspection? How does it work?

See slides

6) (1 points) What is a deadline? Give an example of a deadline in a construction project.

Calendar date when a task/deliverable has to be completed/issued. Ex: roof ready by may 31 2009.

7) (1 point) According to the Cocomo model, how are related duration and effort?

$TDEV = 3 PM (0.33 + 0.2 * (B - 1.01))$ with $B=1$ in the simplest model

Writing the formula was not necessary, key point was saying that duration (TDEV) depends on effort (PM) with exponential formula, exponent <1

8) (1 point) What is a build? What are the related problems?

Process (including compilation, link, possibly testing and other activities) to produce an executable starting from source code modules and libraries. Problems: dependencies among modules, finding and using the right modules in term of type and version. Popular tools: Make, Ant, Maven.

9) (1 point) In the context of verification and validation, describe a static analysis technique

Inspection or dataflow analysis or control flow analysis.