Theory ISW

Group:

Name:

Theory Evaluation

January 30th 2023

3h 30 min

Norms:

- **Do not open the exam** until it is indicated by a lecturer in the classroom.
- Write your answers using the attached pages (no extra sheets will be given). You may only extract the exam text page.
- Submission:
  - The student will identify him(her)self when delivering the exam answers.
  - The exam text and answers will be introduced in a box available.
  - The submission will be carried out in an orderly way. One student only will be delivering the exam at a time.

Software Engineering

Computer Science School

DSIC – UPV

Year 2022-2023

Software Engineering (ISW)

Theory Evaluation
January 30th 2023
ETSInf-UPV

NAME:
GROUP:

Time: 3 hours 30 minutes

Questions *(2 points)*

1. (0.75 *points*) Does a software methodology just only establish the set of activities to be carried out to produce and manage a software product? Justify your answer.

2. (*0.5 points*) What are the benefits of separating the business logic layer and the persistence layer in a 3-layered architecture? Justify your answer.

3. (*0.75 points*) What is a *Data Transfer Object* (DTO) and what is it used for?

Problems *(8 points)*

4. (*1.5 points*) Obtain the test cases for the following code fragment following the basis path technique, draw the flow graph, calculate the cyclomatic complexity, specify the independent paths and the associated test case for each path. The function "`findElement`" receives an array ("`v`") that may be "`null`", an int to be searched ("`elem`"), a maximum number of times to find it ("`max`") and it returns using an out parameter "`number`" the number of times that "`elem`" has been found in "`v`". It also returns the last position in the array (lastIndex) in which "elem" was found. Assume that only the indicated lines of code may produce an exception.

```
int findElement(int[]? v, int elem,int max,out int number){
    int index=0;
    int lastIndex = -1;
    number = 0;
    try
    {
        int length=v.Length; // It may produce null exception
        while (index <= length)
        {
            int v_elem = v[index]; //It may produce out of bounds exception
            if (v_elem == elem) {
                number++;
                lastIndex = index;
            }
            index++;
            if (number == max) break;
        }
    }
    catch {
        number = 0;
        lastIndex = -1;
        Console.WriteLine("An exception ocurred!");
    }
    return lastIndex;
}
```

5. *(4.5 points)* Given the following description obtain

   a) (2.5 points) the associated UML class diagram (**do not include methods nor attribute types).**
   b) (2 points) the structured UML use cases diagram.

The management of Collet of Mariola irrigation well is hiring ISWSoft to develop a software for managing its operations. The Collet of Mariola irrigation well is a community of users who water their fields. There are two types of users, owners and non-owners. The main difference between them is that owners have bought stakes of ownership (each stake has a unique code identifying it and a date of acquisition). Because of this ownership, owners have a discount on the fees that have to be paid because of watering services. On the contrary, non-owners have to pay the fees without any discount. The discount depends on the percentage of the well that has been acquired and it is calculated taking into account the number of shares bought by each owner and the water consumption. Therefore, each owner has a different discount applied on each season.

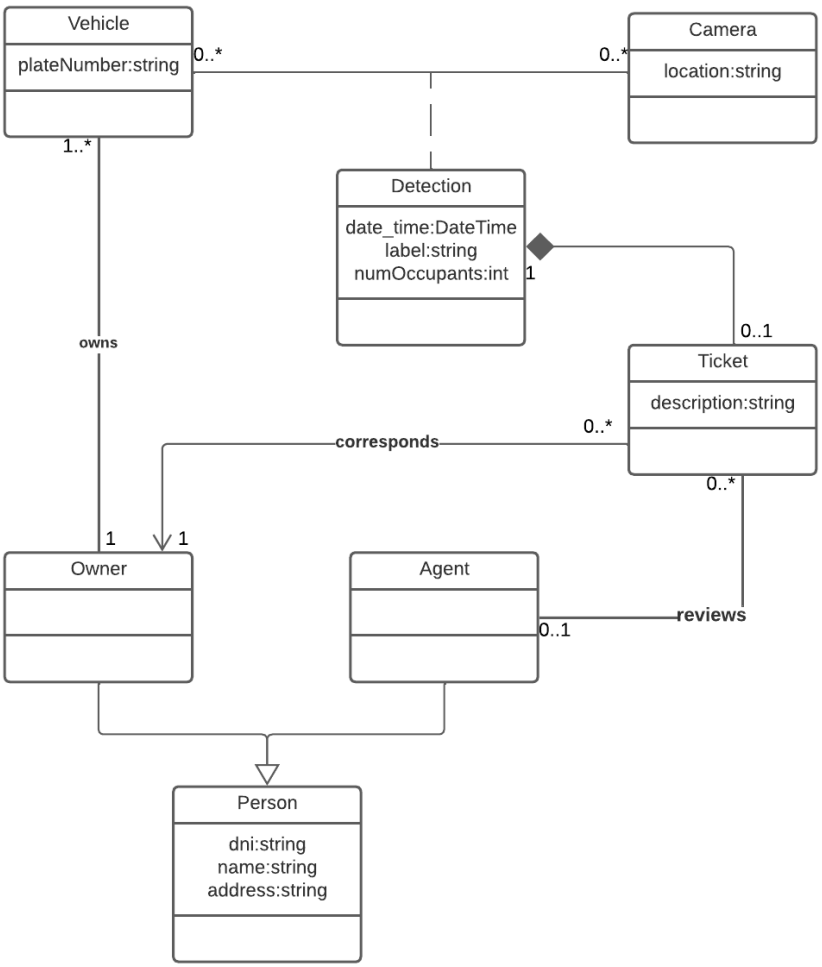
The users (owners or non-owners) may have different fields to water. Watering fields are grouped in terms of watering branch lines depending on the main tube that provides water to the fields. Currently, the well has 8 watering branch lines, but new branches could be created if there is demand of water for new fields that require new main pipes. Each watering branch line is identified with a code and has a length expressed in kilometers.

The software to be developed must support adding new users, changing the type of user non-owner/owner and the acquisition of shares. Particularly, a potential user may make a request to be added in the system. To this end, the potential user will provide his/her data (DNI, first name, last name, address, phone number and email) and optionally, the number of ownership shares he/she wants to buy and/or the fields that need to be watered (province, city, the area code where the field is located and the field id). If the new user creation request contains fields to be watered, the request must be reviewed by the maintenance manager who will study the feasibility of watering the fields before the new user creation request is confirmed.

Any share may be bought by several owners and in this case, it must be specified the percentage of ownership of the share for each owner. A user may also request the watering of new fields (a user may change from non-having any fields to water to having some). This request must be reviewed by the maintenance manager who must approve the watering of new fields. All these actions may also be done by the maintenance manager on behalf of the users.

Finally, the system will automatically generate at the end of the month the bank orders to charge for the watering fees or for the payment of dividends depending on the type of user (owner or non-owner). The generated watering fees will include the date, the cost of the fee and the amount of water (cubic meters) used for watering. The generated dividends for owners will contain the date, the dividend money amount and a number that indicates the position in the earnings ranking of owners.

6. *(2 points)* Given the following UML class diagram:



Note: Person is an abstract class and there is a navigation restriction between Ticket and Owner.

a) *(1 point)* Obtain the C# design following the design patterns studied (do not add any class methods).

b) *(0.5 points)* Obtain the C# design of the constructor with parameters for each class (just the header, do not implement them. Do not define the constructor without parameters).

c) *(0.5 points)* Using the constructors from the previous question implement the necessary code to create a system consistent with the diagram for the following requirement: *create a vehicle that has been detected by a camera and that a ticket has been generated.* Use any arbitrary values for the remaining attributes.

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:

Name:
Group:
