

## Identify and Specify Concerns

### Usability

<Concern name="Usability">

<Requirement id="1">

The system should have an easy to use GUI, because it might be used by any person who has access to the internet.

<Requirement id="1.1">

The system should have an on-line HELP to be consulted by any person that uses it.

</Requirement>

<Requirement id="1.2">

Several functionalities of the system should provide context sensitive help for the user (e.g., explain what are the types of complaints).

</Requirement>

</Requirement>

<Requirement id="2">

The user interface must be implemented using servlets.

</Requirement>

</Concern>

### Availability

<Concern name="Availability">

<Requirement id="1">

The system should be available 24 hours a day, 7 days a week.

</Requirement>

<Requirement id="2">

Because of the nature of not being a critical system, the system might stay off until any fault is fixed.

</Requirement>

</Concern>

### Performance

<Concern name="Performance">

<Requirement id="1">

The system must be capable to handle 20 simultaneous users.

</Requirement>

<Requirement id="2">

The response time must not exceed 5 seconds.

</Requirement>

</Concern>

### Concurrency

<Concern name="Concurrency">

<Requirement id="1">

The system must be capable to handle 20 simultaneous users.

</Requirement>

</Concern>

## Distribution

```
<Concern name="Distribution">
  <Requirement id="1">
    The system should be capable of running on separate machines. For
    example, the system core could be running on one machine and the servlets on another.
  </Requirement>
</Concern>
```

## Security

```
<Concern name="Security">
  <Requirement id="1">
    The system should use a security protocol to send data over the internet.
  </Requirement>
  <Requirement id="2">
    To have access to the complaint registration features, the use must be
    allowed by the access control sub-system.
    <Requirement id="2.1">
      The employee provides the login and password.
    </Requirement>
    <Requirement id="2.2">
      The system validates the typed password.
    </Requirement>
  </Requirement>
</Concern>
```

## Login

```
<?xml version="1.0"?>
<Concern name="Login">
  <Requirement id="1">
    The employee provides a login name and password.
  </Requirement>
  <Requirement id="2">
    The system validates the login name and password.
  </Requirement>
</Concern>
```

## Standards

```
<?xml version="1.0"?>
<Concern name="Standards">
  <Requirement id="1">
    The system must be developed according to the standards established by X ,
    responsible for the norms and standardization of systems for the City Hall.
  </Requirement>
</Concern>
```

## Persistence

```

<?xml version="1.0"?>
<Concern name="Persistence">
  <Requirement id="1">
    The persistence mechanism should store data about:
    <Requirement id="1.1">
      Complaints.
    </Requirement>
    <Requirement id="1.2">
      Employees.
    </Requirement>
    <Requirement id="1.3">
      Health Units.
    </Requirement>
    <Requirement id="1.4">
      Deceases.
    </Requirement>
    <Requirement id="1.5">
      Specialties.
    </Requirement>
    <Requirement id="1.6">
      Citizens that complain.
    </Requirement>
  </Requirement>
  <Requirement id="2">
    The system must be flexible in terms of the storage format allowing the use
    of arrays or different databases (MySQL, Oracle, etc.).
  </Requirement>
</Concern>

```

### Query\_Information

```

<?xml version="1.0"?>
<Concern name="Query_Information">
  <Requirement id="1">
    The citizen chooses the type of query.
  </Requirement>
  <Requirement id="2">
    The citizen provides the data for the query.
    <Requirement id="2.1">
      In the case of a query on specialties grouped by health units, the
      citizen selects the unit to be queried.
    </Requirement>
    <Requirement id="2.2">
      In the case of a query on health units grouped by specialties, the
      citizen selects the specialty to be queried.
    </Requirement>
    <Requirement id="2.3">

```

In the case of a query on complaints, the citizen provides the complaint code.

```
</Requirement>  
<Requirement id="2.4">
```

In the case of a query on diseases, the citizen selects the disease to be queried.

```
</Requirement>  
</Requirement>  
<Requirement id="3">  
    The system shows the query result.
```

```
</Requirement>  
</Concern>
```

### **Register\_Complaint**

```
<?xml version="1.0"?>  
<Concern name="Register_Complaint">  
    <Requirement id="1">  
        The citizen informs the kind of complaint.  
    </Requirement>  
    <Requirement id="2">  
        The system registers the kind, date and time of the attendance.  
    </Requirement>  
    <Requirement id="3">  
        The system shows the specific screen for each type of complaint.  
        <Requirement id="3.1">  
            Animal Complaint – DVA •Animals apprehension. •Control of vectors  
            (rodents, scorpions, bats, etc.) •Diseases related to mosquitos (dengue,  
            filariose). •Animals maltreatment.  
        </Requirement>  
        <Requirement id="3.2">  
            Food Complaint - DVISA •Cases where it is suspicious the ingestion  
            of infected food.  
        </Requirement>  
        <Requirement id="3.3">  
            Diverse Complaint - DVISA •Cases related to several reasons, which  
            are not mentioned above (restaurants with hygiene problems, leaking  
            sewerage, suspicious water transporting trucks, etc.).  
        </Requirement>  
    </Requirement>  
    <Requirement id="4">  
        The citizen provides the generic data. Complaint data: description  
        (mandatory) and observations (optional); Complainer data: name, street,  
        complement, district, city, state/province, zip code, telephone number and e-  
        mail. All these fields are optional.  
        <Requirement id="4.1">
```

Specific animal complaint data: Type of animal (mandatory), amount of animals (mandatory), date problem was observed (mandatory).

</Requirement>

<Requirement id="4.2">

Specific food complaint data: Victim's name (mandatory). Victim's data: street, complement, district, city (or closest one), state/province, zip code and telephone number. All of these fields are optional. Amount of people who ate the food, amount of sick people, amount of people who were sent to a hospital and amount of deceased people. All mandatory. Location where the patients were treated, suspicious meal. All optional.

</Requirement>

<Requirement id="4.3">

Specific special complaint data: Age (mandatory), academic qualifications (optional), occupation (optional). Street, complement, district, city, state/province, zip code and telephone number of the closest location to the complaint location. All optional.

</Requirement>

</Requirement>

<Requirement id="5">

The system saves the complaint (with the OPENED state), return the code for the attendance, so that the citizen can take note and query for the situation of his/her complaint.

</Requirement>

</Concern>

## Register\_Tables

<?xml version="1.0"?>

<Concern name="Register\_Tables">

<Requirement id="1">

The employee will be able to register in one of the tables.

<Requirement id="1.1">

The employee chooses the option to register (insert/update/delete/search/print) in one of the tables.

</Requirement>

<Requirement id="1.2">

The employee informs the data.

<Requirement id="1.2.1">

Health unit (unit code, unit description).

</Requirement>

<Requirement id="1.2.2">

Specialty (code and description).

</Requirement>

<Requirement id="1.2.3">

Health unit / Specialty (health unit and specialty).

</Requirement>

<Requirement id="1.2.4">

Employee (login, name and password).

```

        </Requirement>
        <Requirement id="1.2.5">
            Type of disease (code, name, description, symptom and
            duration).
        </Requirement>
        <Requirement id="1.2.6">
            Symptom (code and description).
        </Requirement>
        <Requirement id="1.2.7">
            Type of disease / Symptom (type of disease and symptom).
        </Requirement>
    </Requirement>
</Requirement>
</Concern>

```

### Update\_Complaint

```

<?xml version="1.0"?>
<Concern name="Update_Complaint">
    <Requirement id="1">
        The employee will be able to update complaint.
        <Requirement id="1.1">
            The employee activates the options update complaint.
        </Requirement>
        <Requirement id="1.2">
            The employee enters the complaint number.
        </Requirement>
        <Requirement id="1.3">
            The system shows the complaint information.
        </Requirement>
        <Requirement id="1.4">
            The employee types the conclusion.
        </Requirement>
        <Requirement id="1.5">
            The system updates the complaint conclusion and the information
            about the employee who gave it.
        </Requirement>
    </Requirement>
</Concern>

```

### Register\_New\_Employee

```

<?xml version="1.0"?>
<Concern name="Register_New_Employee">
    <Requirement id="1">
        The employee will be able to register new employees on the system.
        <Requirement id="1.1">

```

The employee provides the following information about the new employee: Name, Login Password (with special field for confirmation).

</Requirement>

<Requirement id="1.2">

The employee confirms the operation.

</Requirement>

</Requirement>

</Concern>

### **Update\_Employee**

<?xml version="1.0"?>

<Concern name="Update\_Employee">

<Requirement id="1">

The employee will be able to update employee's data on the system.

<Requirement id="1.1">

The employee chooses the option update employee.

</Requirement>

<Requirement id="1.2">

The employee provides his/her data to be updated: Name, New password (with extra field for confirmation), Current password.

</Requirement>

</Requirement>

</Concern>

### **Update\_Health\_Unit**

<?xml version="1.0"?>

<Concern name="Update\_Health\_Unit">

<Requirement id="1">

The employee will be able to update health's data on the system.

<Requirement id="1.1">

The employee chooses the option update health unit.

</Requirement>

<Requirement id="1.2">

The employee provides the health unit data to be updated.

</Requirement>

<Requirement id="1.3">

The employee changes the necessary data.

</Requirement>

</Requirement>

</Concern>

### **Change\_Logged\_Employee**

<?xml version="1.0"?>

<Concern name="Change\_Logged\_Employee">

<Requirement id="1">

The employee will be able to change a logged employee on the system.

```
        <Requirement id="1.1">
            The employee chooses the option change logged employee.
        </Requirement>
    </Requirement>
</Concern>
```

## **Error and Exception Handling**

```
<?xml version="1.0"?>
<Concern name="Error_and_Exception_Handling">
    <Requirement id="1">
        Several functionalities might raise errors while the user interacts with the
        system and require different handling techniques.
    </Requirement>
    <Requirement id="2">
        General errors that apply to most cases are due to missing information.
        <Requirement id="2.1">
            Display the message employee information was not provided.
        </Requirement>
        <Requirement id="2.2">
            Display the message complaint information missing.
        </Requirement>
        <Requirement id="2.3">
            Display the message name or current password is missing.
        </Requirement>
        <Requirement id="2.4">
            Display the message health unit data is missing.
        </Requirement>
    </Requirement>
    <Requirement id="3">
        Other errors might be related to entering invalid data and the error handling
        mechanism should try either to avoid that or to raise the error and suggest
        the correction.
        <Requirement id="3.1">
            Display the message employee is invalid.
        </Requirement>
        <Requirement id="3.2">
            Display the message the password or the employee is not valid.
        </Requirement>
    </Requirement>
</Concern>
```