Requirements Document
Health-Watcher

Version 2.0

Citation for the case study:

Sérgio Soares, Eduardo Laureano, and Paulo Borba

"Implementing Distribution and Persistence Aspects with AspectJ."

In Proceedings of the 17th ACM conference on

Object-oriented programming, systems, languages, and applications,

OOPSLA' 02, pages 174–190. ACM Press, November 2002.

Also appeared in ACM SIGPLAN Notices 37(11).

The authors of the Use Cases Document are:

- An unnamed software company,
- Tiago Massoni,
- Sérgio Soares and
- Paulo Borba

Contents

Introduction

This document specifies the requirements for the City Hall Public Health System named HEALTH-WATCHER. This provides developers the required information for designing/implementing the system and maintaining it through testing and validation.

Document Overview

This introduction provides information to aid the understanding of this document, explaining the goals and conventions adopted in the text. It also contains a list of references to other related documents. In later sections the specification of the HEALTH-WATCHER system is detailed. These sections contain the following information:

Section 2 – System overview: depicts a global vision of the system, characterizing its scope and describing its users.

Section 3 – Functional requirements (use cases): specifies all the system's functional requirements. The flow of events, priorities, actors, inputs and outputs for each use case to be implemented are also detailed.

Section 4 – Non-functional requirements: specifies all system's non-functional requirements, divided into: usage requirements, liability, performance, security, distribution, pattern adapting, and hardware/software requirements.

Conventions, terminology and abbreviations

To understand this document correctly certain conventions and specific terminology need to be defined, which are described below.

Requirements identification

By convention, the link to the requirements is made by the name of the subsection where they are described, followed by the requirement's label, according to the format below:

[subsection name.requirement label]

For instance, the requirement [Data recovery.FR016] is described in a subsection called “Data recovery” and in a block labeled [FR016]. Similarly, the non-functional requirement [Liability.NF008] is described on the non-functional Liability requirements section, in a block labeled [NF008].

Requirements priority

To establish the requirements priority, the priority levels: “essential”, “important” and “desirable” were adopted.

Essential is a requirement the system cannot work without. Essential requirements must be implemented or the system will not work.

Important is a requirement the system can work without, but not in a satisfactory way. Important requirements should be implemented, but if not, the system can be deployed and used anyway.

Desirable is a requirement that does not compromise the system's basic features. The system can work in a satisfactory way without it. Desirable requirements are the ones that can be left to be implemented on further versions of the system in case there is not enough time to do so on the current version.

System overview

The purpose of the system is to collect then manage public health related complaints and notifications. The system is also used to notify people about important information regarding the Health System.

The Health Watcher system must also exchange information with the SSVS system (Sanitary Surveillance System). Initially, this exchange will involve the querying of sanitary licenses. Subsequently, when the SSVS has the Complaint Control module deployed, Sanitary Surveillance complaints will be exchanged between the two systems.

Broadening and related systems

With the deployment of HEALTH-WATCHER system, the Public Health System will considerably improve:

- The complaint control (registering and notifications).
- Quality of service regarding the dissemination of information; for example: vaccination campaigns, disease prevention, health guides, obtaining birth/death certificates and application details for a sanitary license.

The system will be managed by DIEVS and will exchange information with the Sanitary Surveillance system.

A citizen can access the system through the internet or dialing 1520, and make their complaint or ask information about the health services. In the event of a complaint being made, it will be registered on the system and addressed by a specific department. This department will be able to handle the complaint in an appropriate manner and return a response when complaint has been dealt with. This response will be registered on the system and available to be queried.

The system will be for public use in kiosks at several strategic points, on which the citizen will be able to register complaints and request information.

Users description

The HEALTH-WATCHER system will support the following users:

- Attendants/DIEVS staff
  Health System employee, placed on DIEVS.

- Citizen
  Any person who wishes to interact with the system.

Functional requirements (use cases)

The following actors have been identified on the system:

Citizen
Any person who wishes to interact with the system.

Employee
Health System employees (inspectors, attendants and managers).

Use cases associated to a citizen

Use cases pertaining to Citizen are the ones that follow:

- FR01 – Query information
- FR02 – Specify complaint

[FR01] Query information

This use case allows a citizen to perform queries.

Query Health Guide
The citizen might query:

- Which health units take care of a specific specialty.
- What are the specialties of a particular health unit.

Query Specialty Information
The citizen might query:

- Information about a complaint made by a citizen:
  - ✓ Complaint details.
  - ✓ Situation (OPENED, SUSPENDED, or CLOSED).
  - ✓ Technical analysis.
  - ✓ Analysis date.
  - ✓ Employee that made the analysis.

- Information about diseases:
  - ✓ Description.
  - ✓ Symptoms.
  - ✓ Duration.

Priority: ☐ Essential ☑ Important ☐ Desirable

Inputs and pre-conditions:

- The data to be queried must be registered on the system

Outputs and post-conditions:

- The query result to the citizen

Main flow of events

1. The citizen chooses the type of query
   - 1.1 In the case of query on specialties grouped by health units, the system retrieves the list of health units stored.
     - 1.1.1 The system retrieves the details of each health unit such as its description and specialties.
     - 1.1.2 The list of health units is presented to the user on their local display.
   - 1.2 In the case of a query on health units grouped by specialties, the system retrieves the list of registered specialties.
     - 1.2.1 The system retrieves the details of each specialty such as its unique identifier and name.
     - 1.2.2 The list of specialties is presented to the user on their local display.
   - 1.3 In the case of a query on diseases, the system retrieves the list of diseases.
     - 1.3.1 The system retrieves the details of each disease type such as its unique identifier and name.
     - 1.3.2 The list of disease is presented to the user on their local display.
2. The citizen provides the data for the query
   - 2.1 In the case of a query on specialties grouped by health units, the citizen selects the health unit to be queried.
     - 2.1.1 A unique identifier representing the selected health unit is sent to the server.
     - 2.1.2 The system ensures the health unit information is consistent.
     - 2.1.3 The unique identifier is used by the system to search the repository for the selected health unit.
     - 2.1.4 The details of the selected health unit are retrieved including its specialties.
     - 2.1.5 The specialties for the selected health unit are returned to the user.
   - 2.2 In the case of a query on health units grouped by specialties, the citizen selects the specialty to be queried.
     - 2.2.1 A unique identifier representing the selected specialty is sent to the server.
     - 2.2.2 The system ensures the health unit information is consistent.

2.2.3 The unique identifier is used to retrieve the list of health units which are associated with the selected specialty.

2.2.4 The details of the health units and specialties are retrieved.

2.2.5 The retrieved health units are returned to the user.

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

2.3.1 The unique identifier representing the complaint to be retrieved is sent to the server.

2.3.2 The system ensures the complaint information is consistent.

2.3.3 The unique identifier is used to retrieve the complaint entry.

2.3.4 The system must determine the complaint type as to retrieve the appropriate information.

2.3.4.1 If the complaint is a special complaint the complainer's age, education level and occupation are retrieved (in addition to the standard complaint information).

2.3.4.2 If the complaint is a food complaint the meal which was consumed, the number of people who ate the meal, the number of sick people, etc. are retrieved (in addition to the standard complaint information).

2.3.4.3 If the complaint is an animal complaint the animal species and the number of animals affected (in addition to the standard complaint information).

2.3.5 The complaint with all the appropriate information is returned to the user.

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

2.4.1 The unique identifier representing the disease type to be retrieved is sent to the server.

2.4.2 The system ensures the disease type information is consistent.

2.4.3 The unique identifier is used to retrieve the disease type to query.

2.4.4 The symptoms for the selected disease type are retrieved.

2.4.5 The complete disease information is returned to the user.

3. The query results are formatted and presented to the user on their local display.

Alternative Flow

1.x and 2.x: A communication problem occurs.

1. Raise an error message

1.x.1 and 2.x.4: A problem occurs retrieving the complaint data.

1. The system retrieves the available information.
2. Raise an error message

2.3.3: An invalid complaint code is entered.

1. Raise an error message informing the user the complaint does not exist.

2.x.2: Consistent data cannot be assured.

1. The system abandons the data retrieval.
2. Raise an error message.

[FR02] Complaint specification

This use case allows a citizen to register complaints. Complaints can be:

Animal Complaint – DVA

- Sick animals.
- Infestations (rodents, scorpions, bats, etc.)
- Diseases related to mosquitoes (dengue, filariose).
- Animal maltreatment.

Food Complaint - DVISA

- Cases where there is a suspicion infected food being eaten.

Special Complaint - DVISA

- Cases related to several reasons, which are not mentioned above (restaurants with hygiene problems, leaking sewerage, suspicious water transporting trucks, etc.).

The three kinds of complaints have the following information in common:

- 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.
- Complaint state (mandatory), which might be: OPENED, SUSPENDED or CLOSED. When a complaint is first registers its state must be OPENED.
- The system must register the complaint registration date.

In addition to the above data, each complaint type has its own specific data, including:

Animal Complaint – DVA

- Type of animal (mandatory), amount of animals (mandatory), date problem was observed (mandatory).

- Problem location data: street, complement, district, city, state/province, zip code and telephone number. All of these fields are optional.

Food Complaint - DVISA

- 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.

Special Complaint - DVISA

- 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.

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-conditions:

- None

Outputs and post-conditions:

- The complaint saved on the system

Main flow of events

1. The citizen selects the kind of complaint.
2. The system shows the specific screen for each type of complaint.
3. The system registers the kind, date and time of the complaints.
4. The citizen provides the complaint specific data.
5. The system saves the complaint.
   5.1. The information entered by the user is sent to the server.
   5.2. The system parses the data entered by the user.
   5.3. The system creates a new instance of the appropriate complaint type.

5.4. The system generates a unique identifier and assigns this to the new complaint.

5.5. The complainers address is parsed and saved.

5.6. The common complaint information is parsed and stored with the OPENED state.

5.7. The specific complaint data is then extracted and stored accordingly.

5.8. The system ensures the data is left in a consistent state.

6. The unique identifier is returned and presented to the user on their local display.

Alternative Flow

5.1: A communication problem occurs.

1. Raise an error message.

5.2: Invalid data is entered by the user.

1. Raise an error message.

5.5-5.7: A problem occurs storing the complaint.

1. The complaint entry is rolled-back.
2. Raise an error message.

5.8: Data consistency cannot be ensured.

1. The complaint entry is rolled-back.
2. Raise an error message.

Use Cases Related to Employee

The employee must login so that he/she can access the various operations of the system, which are:

- FR10 – Login.
- FR11 – Register tables.
- FR12 – Update complaint.
- FR13 – Register new employee.
- FR14 – Update employee.
- FR15 – Update health unit.
- FR16 – Change logged employee.

[FR10] Login

This use case allows an employee to have access to restricted operations on the Health-Watcher system.

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-conditions:

- None

Outputs and post-conditions:

- Password validated by the system

Main flow of events

1. The employee provides the login and password.
2. The login and password are sent to the server.
3. The system retrieves the employee details using the login as a unique identifier.
4. The system validates the entered password.
5. The result of the login attempt is presented to the employee on their local display.

Alternative flow

2: A communication error occurs.

1. Raise an error message.

3: A problem occurs retrieving the employee details.

1. Raise an error message.

4: The system cannot validate the employee.

1. Raise an error message.

[FR11] Register tables

This use case allows the registration of system tables. The following operations are possible: insert, update, delete, search and print.

The available tables include:

- Health unit (unit code, unit description).
- Specialty (code and description).
- Health unit / Specialty (health unit and specialty).
- Employee (login, name and password).
- Type of disease (code, name, description, symptom and duration).
- Symptom (code and description).
- Type of disease / Symptom (type of disease and symptom).

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-conditions:

- Verified employee

Outputs and post-conditions:

- Updated data on the tables

Main flow of events

1. The employee chooses the option to register (insert/update) in one of the tables.
2. The employee enters the data.
3. The system saves the data.

[FR12] Update complaint

This use case allows the state of a complaint to be updated.

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-conditions:

- The complaint must be registered and have the OPENED state.
- Verified employee.

Outputs and post-conditions:

- Complaint updated and with state CLOSED.

Main flow of events

1. The employee selects the update complaint option.
2. The system retrieves the list of all registered complaints.
   2.1. The complaint list is populated with general and complaint type specific data.
3. The list of complaints is returned to the employee.
4. The complaints are formatted and presented to the employee on their local display.
5. The employee selects the complaint they wish to update.
6. The complaint unique identifier is sent to the server.
7. The system ensures the complaint data is consistent.
8. The system retrieves the complaint entry.
9. The complaint is returned to the employee.
10. The complaint is formatted and presented to the employee on their local display.
11. The employee enters the conclusion.
12. The conclusion is sent to the server.
13. The complaint status is set to closed; the date the conclusion was entered is set in addition to the employee who dealt with the complaint.
14. The system ensures the complaint is left in a consistent state.
15. The complaint information is updated to store the new information.

Alternative Flow

2 and 8: An error occurs retrieving the registered complaints.

2. Raise an error message.

7 and 14: Data consistency cannot be ensured.

1. The complaint changes are rolled-back.

2. Raise an error message.

3, 6, 9, and 12: A communication error occurs.

1. Raise an error message.

15: An error occurs storing the updated complaint.

1. The complaint changes are rolled back.
2. Raise an error message.

[FR13] Register new employee

This use case allows new employees to be registered on the system.

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-conditions:

- Verified employee

Outputs and post-conditions:

- New employee registered on the system

Main flow of events

1. The employee selects the insert employee option.
2. The employee provides the following information about the new employee:
   - ✓ Name
   - ✓ Login ID
   - ✓ Password (with second password field for confirmation).
3. The employee confirms the operation.
4. The entered data is transmitted to the server.
5. The system verifies the entered data.
6. The system ensures employee data is consistent.
7. The system saves the new employee's data.

Alternative flow

2: Incomplete data entered.

1. Show a message informing the employee of the missing/incorrect data.

4: A communication error occurs.

1. Raise an error message.

5: The employee is already entered.

1. Inform the employee that the new employee is already entered.
2. Abandon the entry.

6: Data Consistency cannot be ensured.

1. The employee entry is rolled-back.
2. The employee is informed the employee cannot be inserted.

7: An error occurs storing the new employee's details.

1. The employee entry is rolled-back.
2. Raise an error message.

[FR14] Update employee

This use case allows of the employee's data to be updated on the system.

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-condition:

- Verified employee

Outputs and post-conditions:

- Employee's data updated on the system

Main flow of events

1. The employee chooses the update employee option.
2. The employee provides the data to be updated:
   - ✓ Name
   - ✓ New password (with second password field for confirmation)
   - ✓ Current password
3. The employee confirms the update.
4. The entered data is sent to the server.
5. The system verifies the entered data.
6. The system ensures the employee data is consistent.
7. The system stores the updated employee information.

Alternative flow

On step 3, in case the name or the current password is missing/invalid, an error message should be showed.

[FR15] Update health unit

This use case allows the health unit's data to be updated.

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-conditions:

- Verified employee

Outputs and post-conditions:

- Health unit's data updated on the system.

Main flow of events

1. The employee chooses the update health unit option.
2. The system retrieves the list of all health units.
3. The list of health units is returned to the employee.
4. The list of health units is formatted and displayed on the employee's local display.
5. The employee selects the health unit to be updated.
6. The unique identifier for the selected health unit is sent to the server.
7. The system ensures the health unit data is consistent.
8. The system retrieves the data for the selected health unit.
9. The data retrieved is returned to the employee.
10. The health unit data is formatted and presented on the employee's local display.
11. The employee alters the necessary data.
12. The updated information is sent to the server.
13. The system ensures the health unit data is left in a consistent state.
14. The system stores the updated health unit information.

Alternative flow

2, 8: A problem occurs retrieving the health unit information.

1. Raise an error message.

3, 6, 9, 12: A communication problem occurs.

1. Raise an error message.

7 and 13: Data consistency cannot be assured.

1. Any health unit updates are rolled-back.
2. Raise an error message.

14: A problem occurs storing the updated health unit data.

1. Any health unit updates are rolled-back.
2. Raise an error message.

[FR16] Change logged employee

This use case allows the currently logged employee to be changed.

Priority: ☑ Essential ☐ Important ☐ Desirable

Inputs and pre-conditions:

- Verified employee.

Outputs and post-conditions:

- First employee signed out and new employee logged-in.

Main flow of events

1. The employee chooses the change logged employee option.
2. The system shows the login screen, and from this point on, the flow will follow the one described in [Login.FR10].

Non-functional requirements

Usability

The system should have an easy to use GUI, as any person who has access to the internet should be able to use the system.

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

Priority: ☐ Essential ☑ Important ☐ Desirable

Availability

The system should be available 24 hours a day, 7 days a week. The nature of the system not being a critical system, the system might stay off until any fault is fixed.

Priority: ☐ Essential ☑ Important ☐ Desirable

Performance

The system must be capable to handle 20 simultaneous users.

The response time must not exceed 5 seconds.

Priority: ☑ Essential ☐ Important ☐ Desirable

Security

The system should use a security protocol when sending data over the internet.

To have access to the complaint registration features, access must be allowed by the access control sub-system.

Priority: ☐ Essential ☑ Important ☐ Desirable

Standards

The system must be developed according to the standards established by X[1], responsible for the norms and standardization of systems for the City Hall.

Priority: ☐ Essential ☑ Important ☐ Desirable

Hardware and software

This section lists the hardware and software to be used for the system to operate in a desirable fashion.

SOFTWARE

- One license for the Microsoft Windows for the workstation

HARDWARE

[1] The company name is confidential due to commercial reasons.

- One computer with: Pentium III processor, 256 MB of RAM memory, net card 3Com 10/100. This equipment shall be used by the attendant as a workstation.

Distribution

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.

Priority: ☐ Essential ☑ Important ☐ Desirable

User interface

The user interface must be implemented using Servlets.

Priority: ☑ Essential ☐ Important ☐ Desirable

Storage medium

The system must be flexible in terms of the storage format allowing the use of arrays or different databases (MySQL, Oracle, etc.)

Priority: ☑ Essential ☐ Important ☐ Desirable