|  |  |
| --- | --- |
| Nombre | [FR01] Query information |
| Descripción | 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. |
| Actor | * Citizen * Employee |
| Disparador | - |
| Flujo Básico | 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. |
| Flujos Alternativos | 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. |
| Precondiciones | The data to be queried must be registered on the system |
| Poscondiciones | The query result to the citizen |
| Requerimientos especiales | - |
| Prioridad | Important |

|  |  |
| --- | --- |
| Nombre | [FR02] Complaint Specification |
| Descripción | 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. |
| Actor | * Citizen * Employee |
| Disparador |  |
| Flujo Básico | 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. |
| Flujos Alternativos | 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. |
| Precondiciones | None |
| Poscondiciones | The complaint saved on the system |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |
| --- | --- |
| Nombre | [FR10] Login |
| Descripción | This use case allows an employee to have access to restricted operations on the Health-Watcher System |
| Actor | * Employee |
| Disparador |  |
| Flujo Básico | 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. |
| Flujos Alternativos | 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. |
| Precondiciones | None |
| Poscondiciones | Password validated by the system |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |
| --- | --- |
| Nombre | [FR11] Register Tables |
| Descripción | 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). |
| Actor | * Employee |
| Disparador |  |
| Flujo Básico | 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 |
| Flujos Alternativos |  |
| Precondiciones | Verified employee |
| Poscondiciones | Updated data on the tables |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |
| --- | --- |
| Nombre | [FR12] Update complaint |
| Descripción | This use case allows the state of a complaint to be updated. |
| Actor | * Employee |
| Disparador |  |
| Flujo Básico | 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 close; 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. |
| Flujos Alternativos | 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. |
| Precondiciones | The complaint must be registered and have the OPENED state.  Verified employee. |
| Poscondiciones | Complaint updated and with state CLOSED. |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |
| --- | --- |
| Nombre | [FR13] Register new employee |
| Descripción | This use case allows new employees to be registered on the system. |
| Actor | * Employee |
| Disparador |  |
| Flujo Básico | 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. |
| Flujos Alternativos | 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. |
| Precondiciones | Verified employee. |
| Poscondiciones | New employee registered on the system |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |
| --- | --- |
| Nombre | [FR14] Update employee |
| Descripción | This use case allows of the employee's data to be updated on the system. |
| Actor | * Employee |
| Disparador |  |
| Flujo Básico | 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. |
| Flujos Alternativos | On step 3, in case the name or the current password is missing/invalid, an error message should be showed. |
| Precondiciones | Verified employee. |
| Poscondiciones | Employee's data updated on the system |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |
| --- | --- |
| Nombre | [FR15] Update health unit |
| Descripción | This use case allows the health unit's data to be updated. |
| Actor | * Employee |
| Disparador |  |
| Flujo Básico | 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. |
| Flujos Alternativos | 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. |
| Precondiciones | Verified employee. |
| Poscondiciones | Health unit's data updated on the system. |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |
| --- | --- |
| Nombre | [FR16] Change logged employee |
| Descripción | This use case allows the currently logged employee to be changed. |
| Actor | * Employee |
| Disparador |  |
| Flujo Básico | 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]. |
| Flujos Alternativos |  |
| Precondiciones | Verified employee. |
| Poscondiciones | • First employee signed out and new employee logged-in. |
| Requerimientos especiales |  |
| Prioridad | Essential |

|  |  |  |
| --- | --- | --- |
| Nombre | Descripción | Prioridad |
| 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. | **Important** |
| 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. | **Important** |
| Performance | The system must be capable to handle 20 simultaneous users.  The response time must not exceed 5 seconds. | **Essential** |
| 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. | **Important** |
| Standards | The system must be developed according to the standards established by X1, responsible for the norms and standardization of systems for the City Hall. | **Important** |
| 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  • 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. | **Important** |
| User interface | The user interface must be implemented using Servlets. | **Essential** |