Technical requirements for the tools to develop on the HeritageCare project

Report of the Project Activity 1.3 (GT.1)

Universidade do Minho (UMinho)	Portugal
Direção Regional da Cultura do Norte (DRCN)	Portugal
Centro de Computação Gráfica (CCG)	Portugal
Universidade de Salamanca (USAL)	Spain
Fundacion Santa Maria La Real (FSMR)	Spain
Instituto Andaluz del Património Histórico (IAPH)	Spain
University Clermont Auvergne (UCA)	France
University of Limoges (ULIM)	France

Date: 30-06-2017

No. pages: 22

Keywords: data base, equipment, inspection tools, forms

Table of Contents

1 Introduction 1

2 HeritageCare Expectations 2

3 Tentative Requirements for the HeritageCare Platform 2

4 Use-case specification for the software solution 3

4.1 System Overview 3

1 Introduction

Currently, no systematic policy for the preventive conservation of built cultural heritage exists in the South-West Europe. The actual approaches for inspection, diagnosis, monitoring and curative conservation are intermittent, unplanned, overpriced and lack a methodical strategy. Therefore, the ultimate goal of the HeritageCare project is the creation of a non-profit self-sustaining entity which will keep supervising the accomplishment of the methodology and the sustainability of the results once the project is concluded.

The main goal of this deliverable is to define the HeritageCare expectations and tentative requirements about the HeritageCare Software solution. In this, the focus was on the elicitation, analisis and definitions of the need and characteristics

The communication of needs, through the definition of requirements, will have a high importance for developing the solution that the team has proposed. A requirement portrays a condition that is needed to satisfy a porpuse or need.

This deliverable comprises the following structure:

Chapter 1 - A brief introduction and main objective of this report, which is being

presently described;

Chapter 2 – The identification of the HeritageCare expectations;

Chapter 3 – A description of the HeritageCare tentative requirements;

Chapter 4 – description of the use-case specification for the software solution.

2 HeritageCare Expectations

Expectation 1. Develop a new management system for the conservation and visualisation of heritage information, capable of increasing the efficiency of the resources intended for those purposes.

Expectation 2. Create a computer application that allows systematising procedures, as well as performing maintenance plans

Expectation 3. Have a platform were the inspector can anwser to the inspection form.

Expectation 4. Include in the database hBIM-based virtual models

Expectation 5. Have an efficient database capable of hosting a large amount of information.

Expectation 6. To have a data-processing system that serves both technicians and end-users (owners or government institutions) to access information in an organized way.

Expectation 7. Have a system that will incorporate the diferent levels of services.

3 Tentative Requirements for the HeritageCare Platform

Expectation	Tentative Requirements
3. Have a platform were the inspector can anwser to the inspection form.	3.1 Have offline access

Table 1 - Tentative Requirement

3.1 Have offline access

It is of high relevance that the inspector, when doing an inspection, have offline access to the inspection form.

4 Use-case specification for the software solution

Use-case diagrams are a crucial part in the elicitation of tentative requirements. This type of diagrams is the main tool used in discussion with stakeholders to validate and discover new requirements. The Use-Case diagram main objective is to represent all the different ways each actor interacts with the system. To do this, we use different levels of representation, as we can see in figure 1.

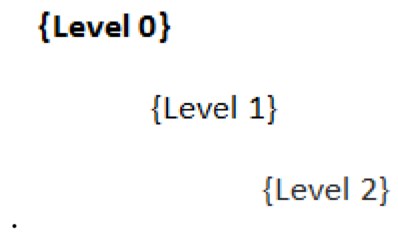

Figure 1 - Levels of representation

4.1 System Overview

To represent the HeritageCare software solution tentative requirements, the UML Use-Case Diagrams were used. Figure 2 represents the highest level of abstraction – Level 0.

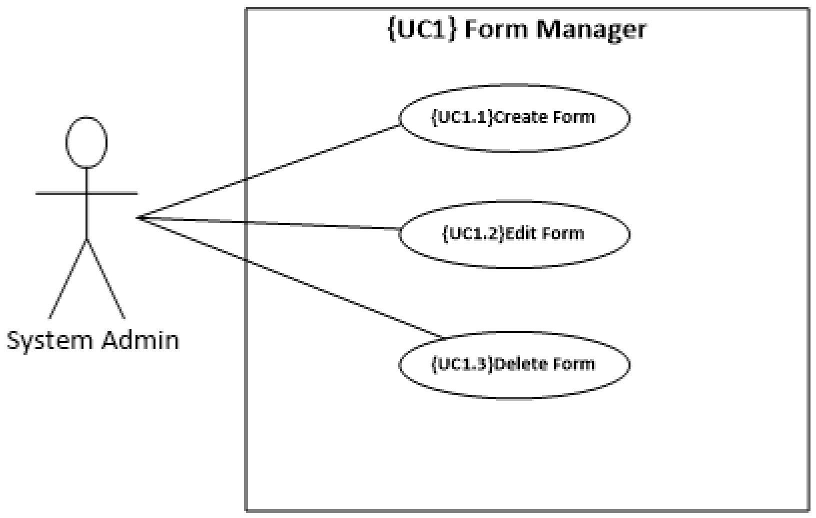

Figure 2 – HeritageCare Level 0

{UC1.1} Create Form: See Below.

{UC1.2} Edit Form: See Below.

{UC1.3} Delete Form: Allows the system admin to delete a form from the system. To do this, the form has to be present in the system ({UC1.1} Create Form).

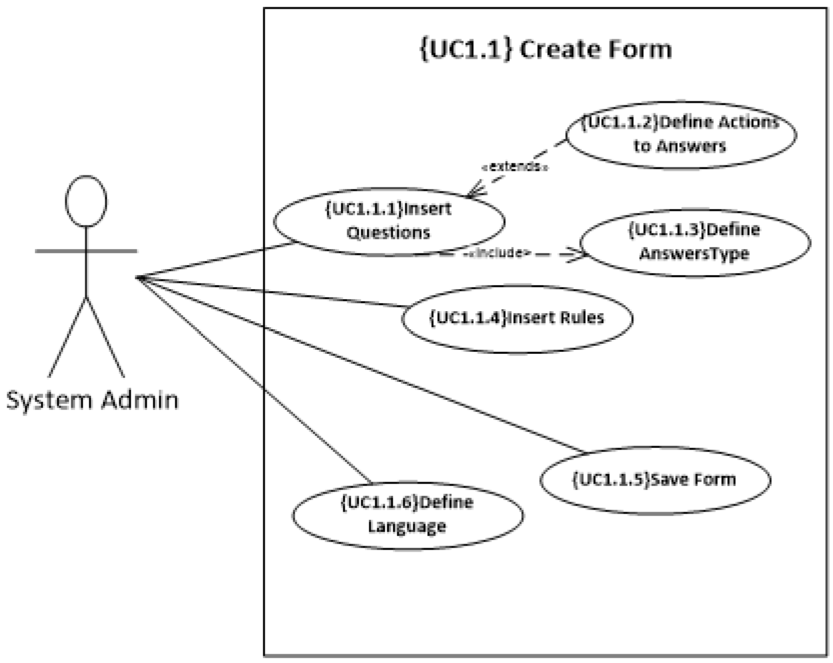

Figure 3 - {UC1.1} Create Form

{UC1.1.1} Insert Questions: Allows the system admin to specify the questions that will belong to the form. Later, the inspector will have to answer these questions while doing the inspection.

{UC1.1.2} Define Actions to Answers: See Below.

{UC1.1.3} Define Answers Type: Every time the System Admin inserts a question *({UC1.1.1} Insert Questions)* the type of answer must be specify. The type of answer can be text, numbers, a list of selective materials and a boolean.

{UC1.1.4} Insert Rules: See Below.

{UC1.1.5} Save Form: Allows the System Admin to save the form on the system. All the properties needed to create the form are stored (questions {UC1.1.1} Insert Questions and rules {UC1.1.4} Insert Rules). After saving, the form is available to be used.

{UC1.1.6} Define Language: Allows the System Admin to choose the language in which the form is going to be displayed (Portuguese, English, Spanish and French from the beginning). Later, the System Admin can add another language to a form, to do this, it has to execute the others functions such as {UC1.1.1} Insert Questions, {UC1.1.2} Define Actions to Answers, {UC1.1.3} Define Answers Type, {UC1.1.4} Insert Rules and {UC1.1.5} Save Form in the respective language.

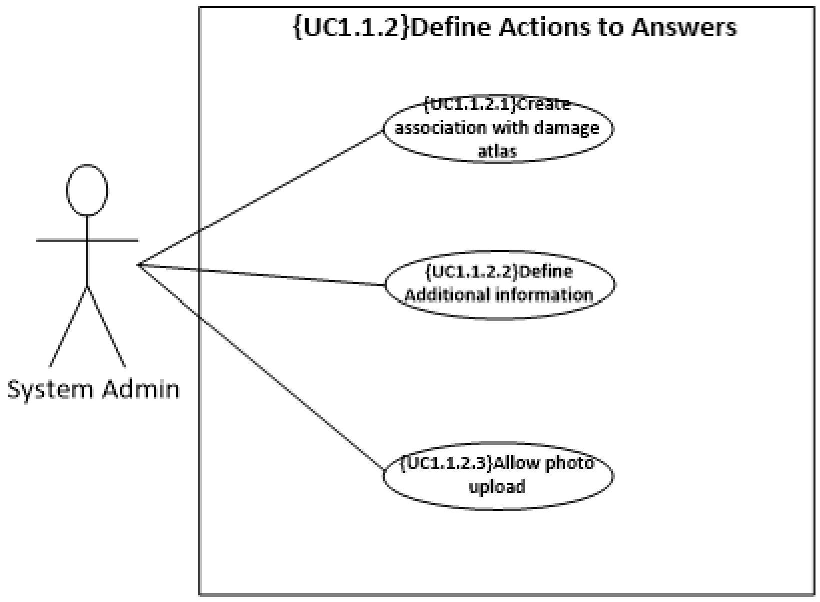
{UC1.1.2}Define Actions to Answers
{UC1.1.2.1}Create association with damage atlas
{UC1.1.2.2}Define Additional information
{UC1.1.2.3}Allow photo upload
System Admin

Figure 4 - {UC1.1.2} Define Actions to Answers

{UC1.1.2.1} Create an association with damage atlas: Allows linking an answer to the damage atlas. With this, it will be possible for the inspector, during the inspection, to identify a damage and consequently answer a question from the form, using the damage atlas.

{UC1.1.2.2} Define Additional information: Allows the System Admin to define additional information (text) to a question. This text will appear as a help to the user when it is answering a question

{UC1.1.2.3} Allow photo upload: Allows the system admin to define that an answer must be accompanied by a photo**.** For example, when the inspector sees a damage he should upload on the system a photo of that damage.

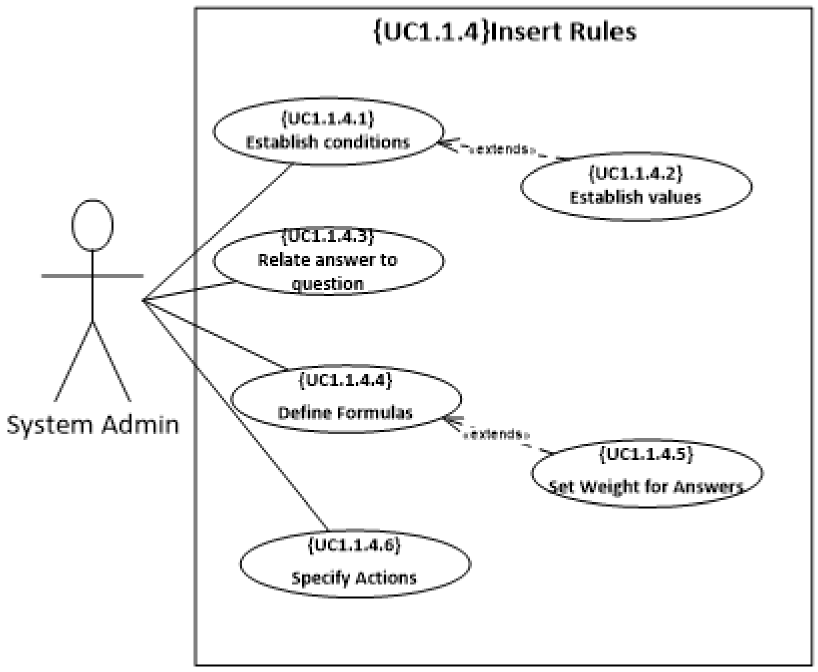
{UC1.1.4}Insert Rules
{UC1.1.4.1} Establish conditions
«extends»
{UC1.1.4.2} Establish values
{UC1.1.4.3} Relate answer to question
{UC1.1.4.4} Define Formulas
«extends»
{UC1.1.4.5} Set Weight for Answers
{UC1.1.4.6} Specify Actions
System Admin

Figure 5 - {UC1.1.4} Insert Rules

{UC1.1.4.1} Establish Conditions: Allows the System Admin to specify a series of rules based on conditions. The rules must define a condition (or a set of conditions) whose verification will cause a given action *({UC1.1.4.5} Specify Actions*). The condition will be based on checking the value of an answer (taken from *{UC1.1.4.3 } Relate answer to question*) for a given value (can be a comparison, if it is greater, less or equal, it can also check if the format of the answer is the correct one)

{UC1.1.4.2} Establish Values: Allows the System Admin, when setting rules based on conditions ({UC 1.1.4.1} Establish conditions) to add intervals of values that the answer must fall in. The interval defined is used to verify that the given answer in the form (in *{UC1.1.4.3} Relate answer to question)* belongs to that interval. For example, in *{UC1.1.4.1} Establish Conditions* if the System Admin defines that a damage must be rated between 0 and 3 and if the user rates a damage with 5 this may rise to an action, for example to an alert (*{UC1.1.2.2} Set Alerts*).

{UC1.1.4.3] Relate answer to question: The System Admin selects the answers (to the question that make up the form and that have been defined in *{UC1.1.1} Insert Questions)* that will be used in the application of the rules (both in *{UC1.1.4.1} Establish Conditions* and *{UC1.1.4.4} Define Formulas*).

{UC1.1.4.4} Define Formulas: Allows the System Admin to specify a series of rules based on formulas to be applied in order to obtain a given result. These formulas use the values of the answers given in the form (taken from *{UC1.1.4.3} Relate answer to question*). The System Admin, must specify how the calculation will be performed and whether different weights should be taken into account for the answer given in the form {UC1.1.4.5} Set Weight Answers) and at the end, a result is obtained. The value of the result and the way it is displayed is further defined in *{UC1.1.4.6} Specify Actions*.

{UC1.1.4.5} Set Weight for Answers: Allows the System Admin, in order to improve the configuration of some formulas, to set different weights to different answers (for example a damage on the roof has a different weight on the overall damage assessment of the building than a damage on the floor). When defining the formula *({UC1.1.4.4} Define Formulas*) the System Admin can specify that for that particular formula an answer has a particular weight.

{UC1.1.4.6} Specify Actions: the System Admin defined which actions (or set of actions) to execute after the rules have been applied. This use case is executed by the System Admin after defining the rules in *{UC1.1.4.1} Establish Conditions* or in *{UC1.1.4.4} Define Formulas*. These actions can be to show some text or to show a list of selectable items (in this context the grading options to evaluate a damage).

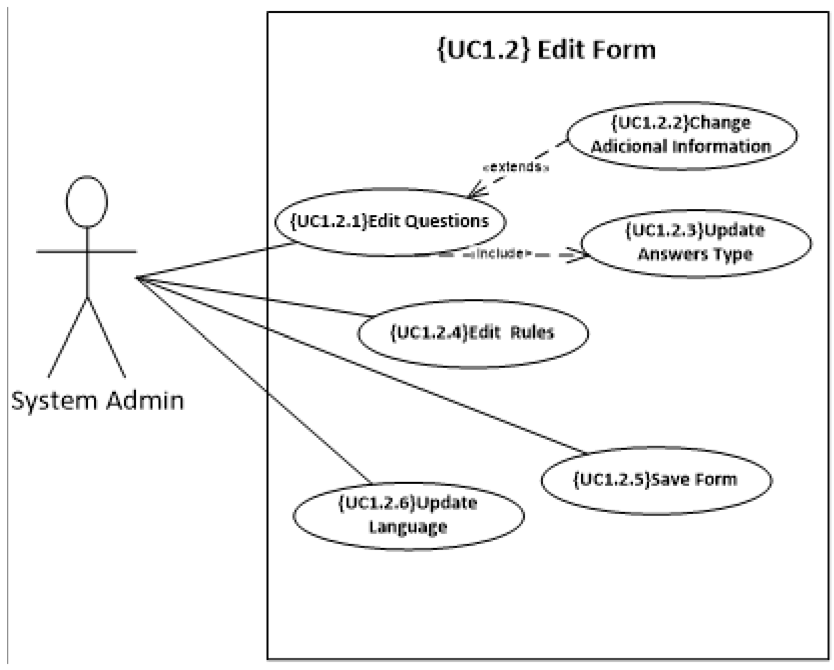

Figure 6 - {UC1.2} Edit Form

{UC1.2.1}Edit Questions: The System Admin can access the questions that make up the form. The inspector answers these questions when is doing an inspection. After accessing the form and loading its properties, the questions are displayed. Then the System Admin has the opportunity to change these properties. He can change the text that forms the question and add or delete questions. The System Admin can view the properties of the form provided.

{UC1.2.2}Change additional information: Allows the System Admin to change the additional information that appears when someone is answering a question. This additional information was defined in *{UC1.1.2.2} Define Additional information.*

{UC1.2.3}Update Answers Type: Every time the System Admin enters or changes a question, *({UC1.2.1} Edit Questions or {UC1.1.1} Insert Questions)* he must specify what type of answer that question must have.

{UC1.2.4}Edit Rules: See Below.

{UC1.2.5}Save Form: This use case ends the process of editing a form. The required properties that compose a form are stored in the repository *({UC1.2.1}Edit Questions, {UC1.2.2}Update Actions to Answers, {UC1.2.3}Update Answers Type, {UC1.2.4}Edit Rules and {UC1.2.6}Update Language).*

{UC1.2.6}Update Language:

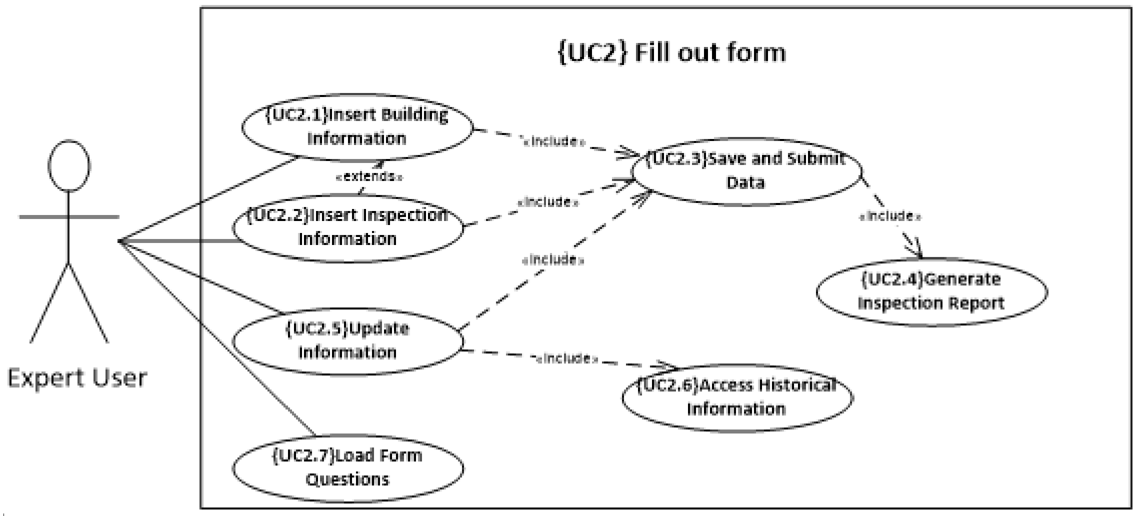
{UC2} Fill out form
{UC2.1}Insert Building Information
«extends»
{UC2.2}Insert Inspection Information
{UC2.5}Update Information
{UC2.7}Load Form Questions
«include»
«include»
«include»
«include»
«include»
{UC2.3}Save and Submit Data
{UC2.4}Generate Inspection Report
{UC2.6}Access Historical Information
Expert User

Figure 7 – {UC2} Fill out Form

{UC2.1} Insert Building Information: The inspector inserts the building information on the system. This information is about the building ID (building information). Every time this use case is executed, the data is stored in the system repository in *{UC.2.3} Save and submit* *data*. Each form field consists of a question (defined by the System Admin in *{UC.1.1.1} Insert Questions* or *{UC1.2.1}Edit Questions)* and an answer (the inspector must enter a value of the desired answer). The answers are defined by the System Admin in *{UC1.1.3} Define answers type* or *{UC1.2.3}Update Answers Type*.

{UC2.2} Insert Inspection Information: When an inspector is doing an inspection, he must answer the inspection form questions previously defined by the System Admin in *{UC1.1.1} Insert Questions or {UC1.2.1}Edit Questions*. In order to do an inspection, the inspector must have already inserted the building information in *{UC2.1} Insert Building Information*. Every time this use case is executed, the data is stored in the system repository in *{UC.2.3} Save and submit data*. In the case of a reinspection, some information, such as the damages previously observed can be present in the new inspection form *({UC2.6} Access Historical Information)*.

{UC2.3} Save and Submit Data: Past and current data is always stored in history. This data is about the building and the inspections performed (these data was inserted in *{UC2.1} Insert Building Information*, {UC2.2} Insert Inspection Information or {UC2.5} Update Information).

{UC2.4} Generate Inspection Report: After the inspection form is answered completely *{UC2.2} Insert Inspection Information*, the information is saved on the system *{UC2.3} Save and Submit Data*, and the inspection report is generated with the building and the inspection information (*{UC2.2} Insert Inspection Information and {UC2.1} Insert Building Information)* . This inspection report is in a .doc format for the inspector, in case of need, can edit it.

{UC2.5} Update Information: After an inspection, the inspector can update the building and the inspection information. This information is accessed with *{UC2.6} Access historical information*. This use case is only available if *{UC2.1} Insert building information or {UC2.2} Insert Inspection Information* were already executed. In the event of a reinspection, only the building information can be updated in the case of need, the inspection itself is considered a new one. In this case, some information such as the damages previously observed can be loaded on to the new inspection form (*{UC2.2} Insert Inspection Information)*. After being updated, the values are stored in the repository in *{UC2.3} Save and Submit Data*. As soon as the form is loaded, the answers to the questions are by default the last ones given; The user selects only the answers he wishes to update. Each form field consists of a question (define by the system admin in *{UC.1.1.1} Insert Questions or {UC1.2.1}Edit Questions)* and a field for the answer (the user enters the value manually or selects the answers through a pre-defined answer). Answers are defined by the System Admin in *{UC1.1.3} Define Answers Type* or *{UC1.2.3}Update Answers Type*.

{UC2.6} Access Historical Information: To update information in {*UC2.5} Update Information,* the data must be first stored *({UC2.3} Save and Submit Data).* Historical information can be later access to update information or in a case of a reinspection.

{UC2.7} Load Form Questions: Loads the questions from the form provided. The questions were defined by the System Admin in *{UC.1.1.1} Insert Questions or {UC1.2.1}Edit Questions*. After the questions are uploaded, the inspector can enter the building or the inspection information with *{UC2.1} Insert Building Information, {UC2.2} Insert Inspection Information or {UC2.5} Update Information.*

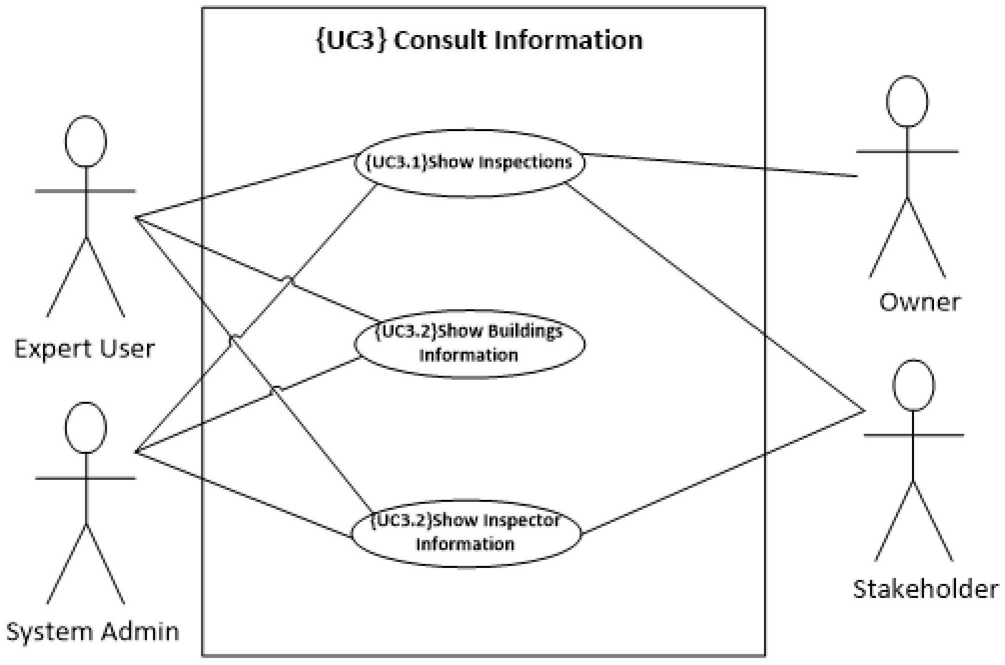
{UC3} Consult Information
{UC3.1}Show Inspections
{UC3.2}Show Buildings Information
{UC3.2}Show Inspector Information
Expert User
System Admin
Owner
Stakeholder

Figure 8 – {UC3} Consult Information

{UC3.1}Show Inspections: Allows to see an inspection or a list of inspections. The users can select a particular inspection and it can see the inspection in detail. For this, the building and the inspection form must have been filled out *({UC2.1} Insert Building Information and {UC2.2} Insert Inspection Information)*. This information can be access with *{UC2.6} Access Historical Information.*

{UC3.2}Show Buildings Information: Allows the Expert User, Owner or System Admin to view a list of buildings. He can select a building and see its information, which was previously inserted ({UC2.1} Insert Building Information or {UC2.5} Update Information). He can also see the inspections performed on that particular building in *{UC2.2} Insert Inspection Information*. This information can be access with *{UC2.6} Access Historical Information*.

{UC3.2}Show Inspector Information: Allows users to see a list of inspectors. If an inspector is selected, the systems will return all the inspections that he has performed.

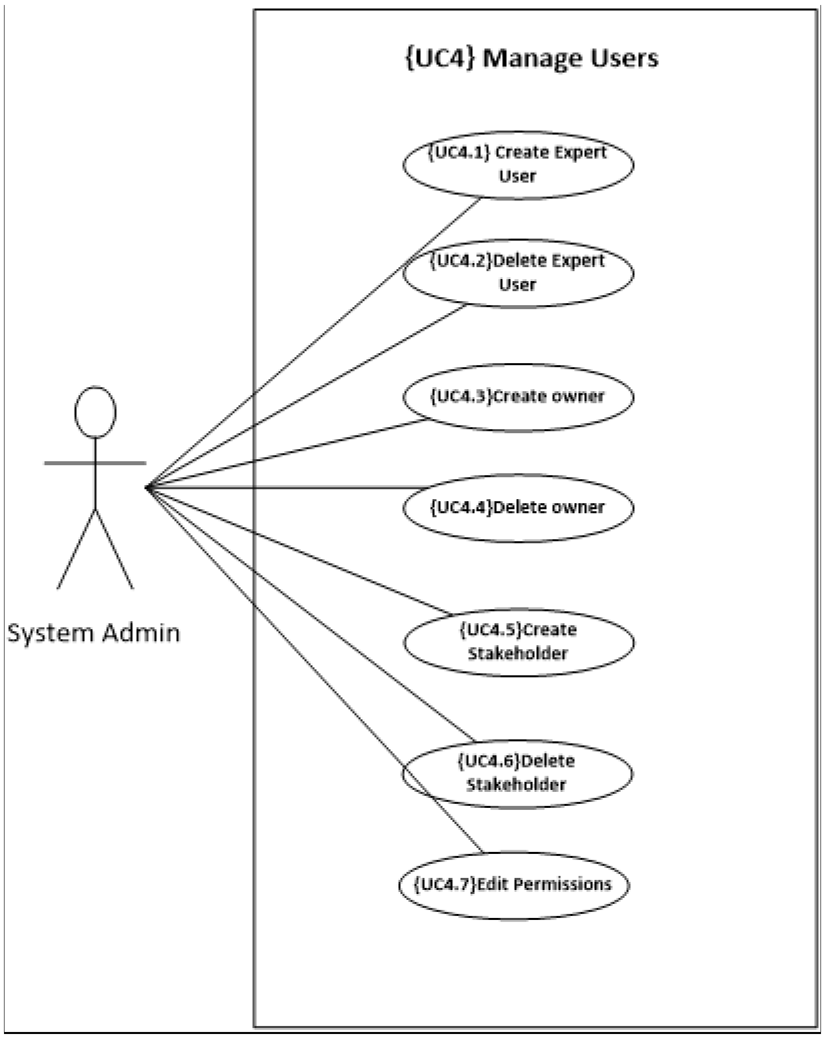
{UC4} Manage Users
{UC4.1} Create Expert User
{UC4.2}Delete Expert User
{UC4.3}Create owner
{UC4.4}Delete owner
{UC4.5}Create Stakeholder
{UC4.6}Delete Stakeholder
{UC4.7}Edit Permissions
System Admin

Figure 9 – Manage Users

{UC4.1} Create Expert User: The System Admin creates a new Expert User in the system. To create a new new Expert User the System Admin inserts the new expert user information

(Name, Adress, Email, Phone). Only registered Expert Users have access to information *({UC2} Fill out form and {UC3} Show results)*.

{UC4.2}Delete Expert User: The System Admin can remove from the database an Expert User that was created in *{UC4.1} Create Expert User*. After executing this use case, the Expert User no longer has access to the form execution features (*({UC2} Fill out form and {UC3} Show results)*.

{UC4.3}Create Owner: The System Admin creates a new Owner in the system. To create a new Owner the System Admin inserts the Owner information (adress, tax number, fiscal adress). Only the registered Owner have access to information.

{UC4.4}Delete owner: The System Admin can remove from the database an Owner that was created in *{UC4.3}Create Owner*. After executing this use case, the Owner no longer has access to the system.

{UC4.5}Create stakeholder: The System Admin creates a new Stakeholder in the system. To create a new Stakeholder the System Admin inserts the Stakeholder information (instituition, contact, name, address, email). Only registered Stakeholders have access to information.

{UC4.6}Delete Stakeholder: The System Admin can remove from the database a Stakeholder that was created in *{UC4.5}Create stakeholder*. After executing this use case, the Stakeholder no longer has access to the system.

{UC4.7}Edit Permissions: The System Admin can change the permission that the actors have with the platform.

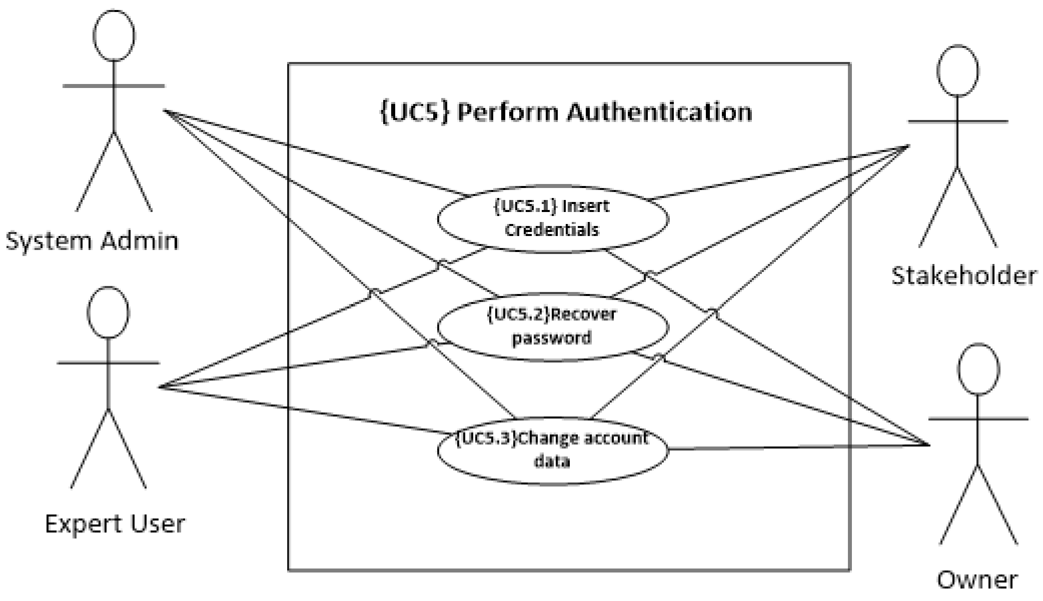
{UC5} Perform Authentication
{UC5.1} Insert Credentials
{UC5.2}Recover password
{UC5.3}Change account data
System Admin
Expert User
Stakeholder
Owner

Figure 10 – Perform Authentication

{UC5.1} Insert Credentials: Each user type (System Admin, Municipally, Inspector and Owner) enters the username and password that were assigned to them. If the information entered corresponds to the data of the respective register, access is given to the functionalities of the tool. After logging into the application the user, according to his profile (System Admin, Municipally, Inspector and Owner) has access to different functionalities.

{UC5.2}Recover password: All users can recover their password over email. The password is automatically sent to the registered email as soon as the user requests the password recovery.

{UC5.3}Change account data: All users can change the information related to their account.

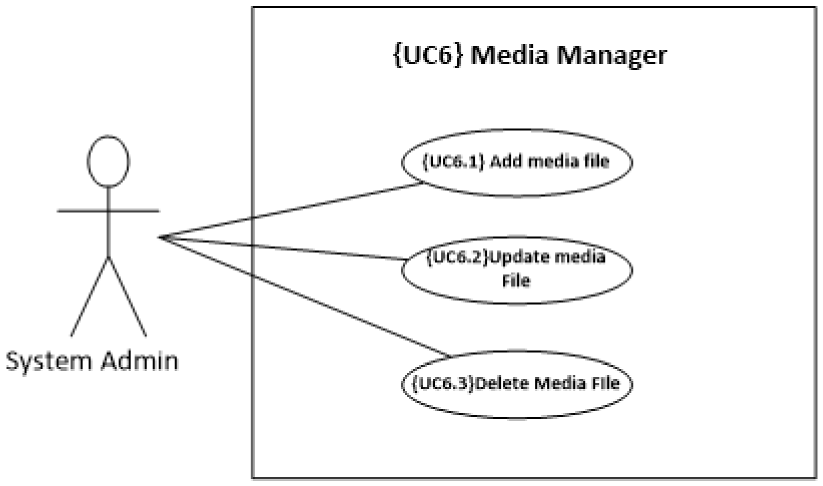

Figure 11 – Media Manager

{UC6.1} Add Media File: Allows the System Admin to add a media file in the system. This media file can be a photo, a video, a document or a 3D model. These media files are later attached in other functionalities of the platform, such as the forms.

{UC6.2}Update Media File: Allows the System Admin to update a media file that was already added to the system in *{UC6.1} Add Media File*.

{UC6.3}Delete Media File: Allows the System Admin to delete a media file.

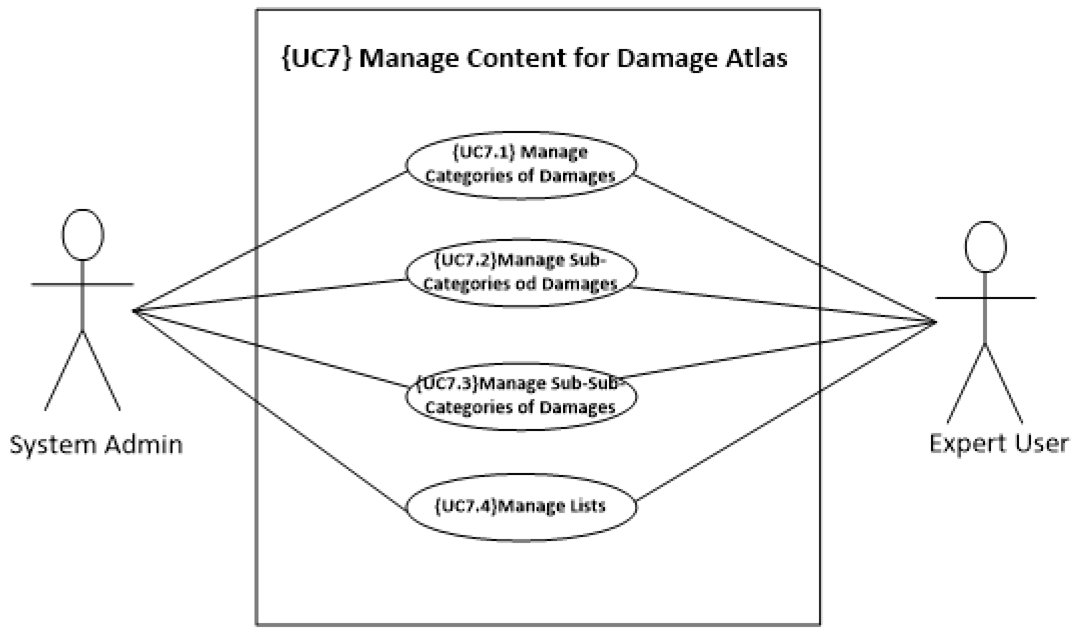
{UC7} Manage Content for Damage Atlas
{UC7.1} Manage Categories of Damages
{UC7.2}Manage Sub-Categories od Damages
{UC7.3}Manage Sub-Sub-Categories of Damages
{UC7.4}Manage Lists
System Admin
Expert User

Figure 12 –Manage Content for Damage Atlas

{UC7.1} Manage Categories of Damages: See Below.

{UC7.2}Manage Sub-Categories of Damages: See Below.

{UC7.3}Manage Sub-Sub-Categories of Damages: See Below.

{UC7.4}Manage Lists: Allows the system admin to create, delete and update lists present on the damage atlas (e.g., List of materials, list of construction elements and others).

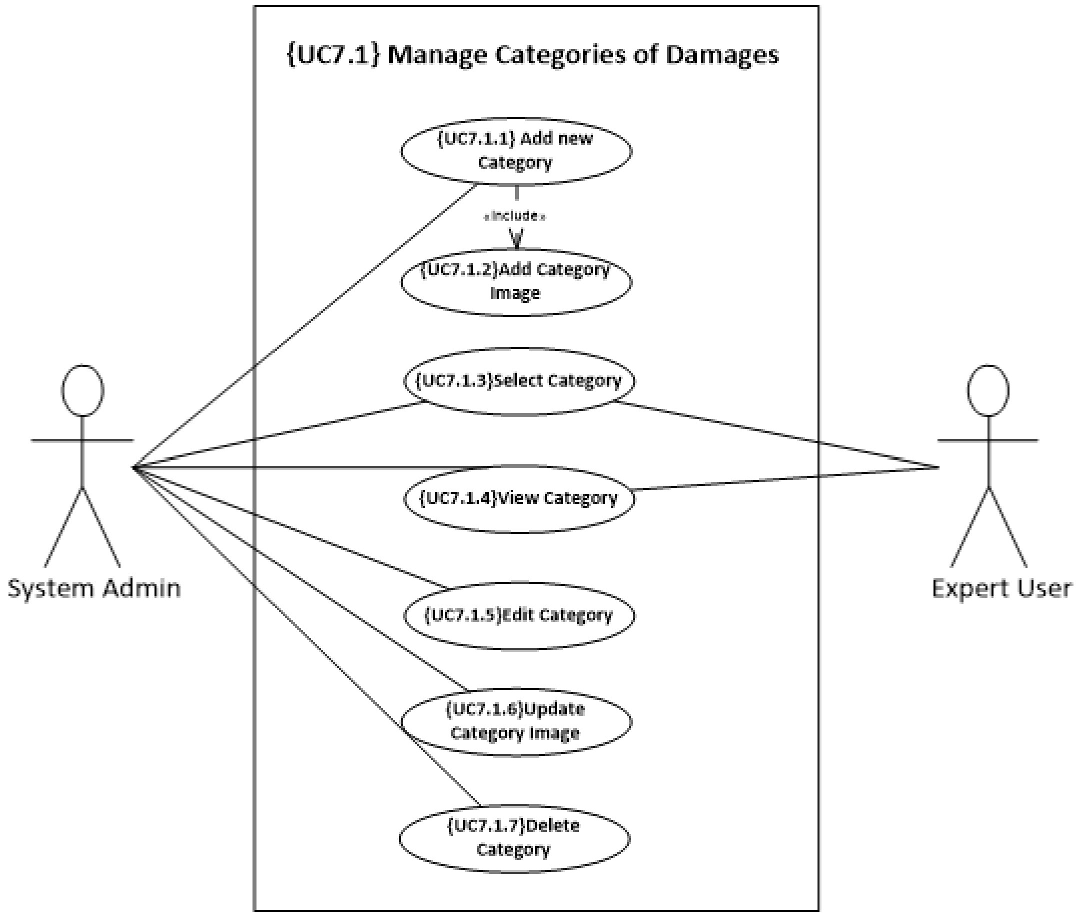

Figure 13 - {UC7.1} Manage Categories of Damages

{UC7.1.1} Add new Category: Allows the System Admin to add a new damage category. To do this, an automatic code is inserted and he has to insert a name. Every time this use case is executed the System Admin must enter an image of damage category *({UC7.1.2}Add Category Image).*

{UC7.1.2}Add Category Image: The System Admin after creating a new damage category *({UC7.1.1} Add new Category*) must insert a damage category image. The image must contain a subtitle.

{UC7.1.3}Select Category: Allows the Expert User or the System Admin to select a category from the damage atlas. The Expert User, when doing an inspection *({UC2.2} Insert Inspection Information*) can select a damage category from the damage atlas directly from the Inspection form.

{UC7.1.4}View Category: Allows the Expert User or the System Admin to view a category from the damage atlas. The Expert User, when doing an inspection *({UC2.2} Insert Inspection*

Information) can view a damage category from the damage atlas directly from the Inspection form.

{UC7.1.5}Edit Category: Allows the System Admin to edit a damage category. The System Admin can edit the name and image of the category. To do this, the damage category has to have been created in *{UC7.1.1} Add new Category*.

{UC7.1.6}Update Category Image: Allow the System Admin to change a damage category image.

{UC7.1.7}Delete Category: A System Admin can delete a Damage Category. When a category is deleted the damage sub-categories, and sub-sub-category are also deleted (*{UC7.2.1} Add new Sub-category*)

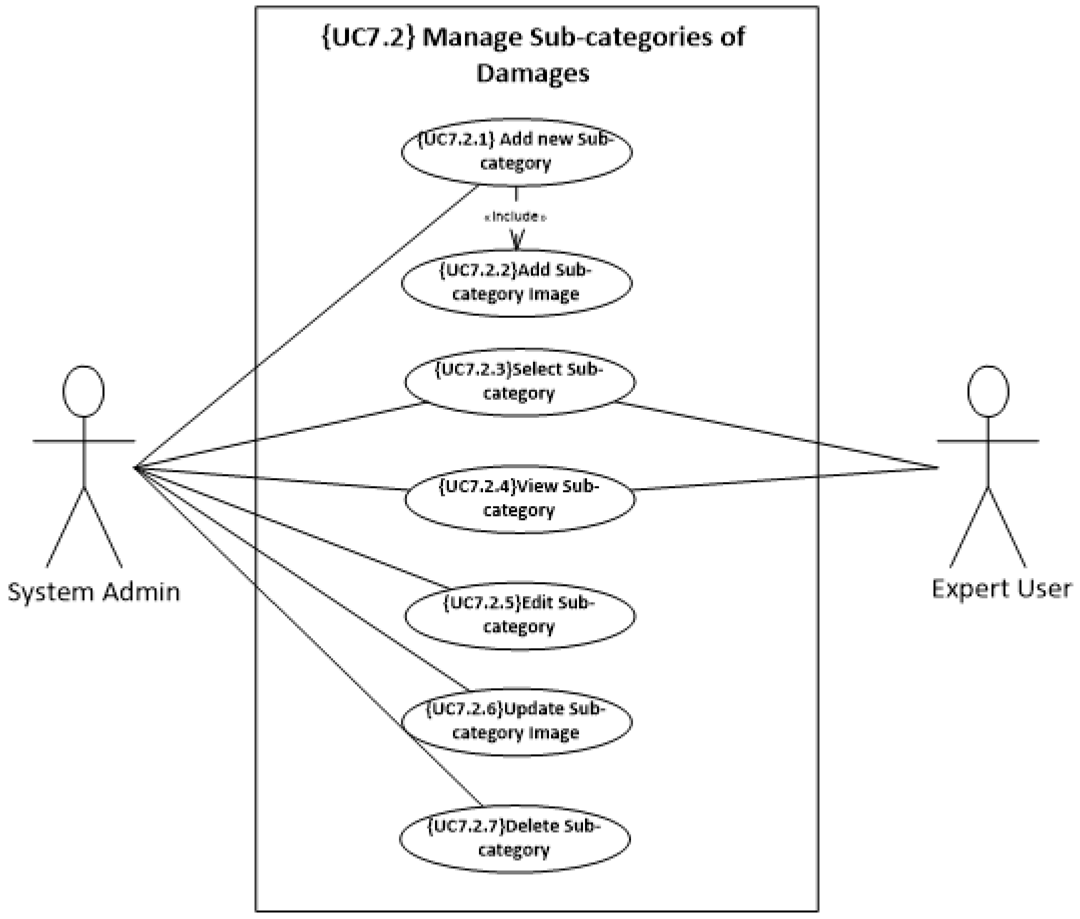
{UC7.2} Manage Sub-categories of Damages
{UC7.2.1} Add new Sub-category
«include»
{UC7.2.2}Add Sub-category Image
{UC7.2.3}Select Sub-category
{UC7.2.4}View Sub-category
{UC7.2.5}Edit Sub-category
{UC7.2.6}Update Sub-category Image
{UC7.2.7}Delete Sub-category
System Admin
Expert User

Figure 14 -{UC7.2}Manage Sub-Categories of Damages

{UC7.2.1} Add new Sub-category: Allows the System Admin to add a new damage sub-category. To do this, he has to insert the code, name, description, Necessary parameters to characterize the damage, possible causes, possible consequences, interrelations between damage, possible mitigation/prevention action, and equivalent terms to be found in other glossaries. Every time this use case is executed the System Admin must enter an image of damage sub-category *({UC7.2.2}Add Sub-category Image).*

{UC7.2.2}Add Sub-category Image: The System Admin after creating a new damage sub-category *({UC7.2.1} Add new Sub-category*) must insert a damage sub-category image. The image must contain a subtitle.

{UC7.2.3}Select Sub-category: Allows the Expert User or the System Admin to select a sub-category from the damage atlas. The Expert User, when doing an inspection *({UC2.2} Insert Inspection Information*) must first select a damage category *({UC7.1.3}Select Category)* from the damage atlas. Only after selecting the damage category, the Expert User is able to select the damage sub-category directly from the Inspection form.

{UC7.2.4}View Sub-category: Allows the Expert User or the System Admin to view a sub-category from the damage atlas. The Expert User, when doing an inspection *({UC2.2} Insert Inspection Information*) can view a damage sub-category from the damage atlas directly from the Inspection form. The Expert User must first select a damage category *({UC7.1.3}Select Category)* to be able to see a damage sub-category.

{UC7.2.5}Edit Sub-category: Allows the System Admin to edit a damage sub-category. The System Admin can edit the name and image of the sub-category. To do this, the damage category and sub-category must have been created in *{UC7.1.1} Add new Category and {UC7.2.1} Add new Sub-category*.

{UC7.2.6}Update Sub-category Image: Allows the System Admin to change a sub-damage category image.

{UC7.2.7}Delete Sub-category: A System Admin can delete a Damage sub-category. When a sub-category is deleted the damage sub-sub-categories is also deleted (*{UC7.2.1} Add new Sub-category*).

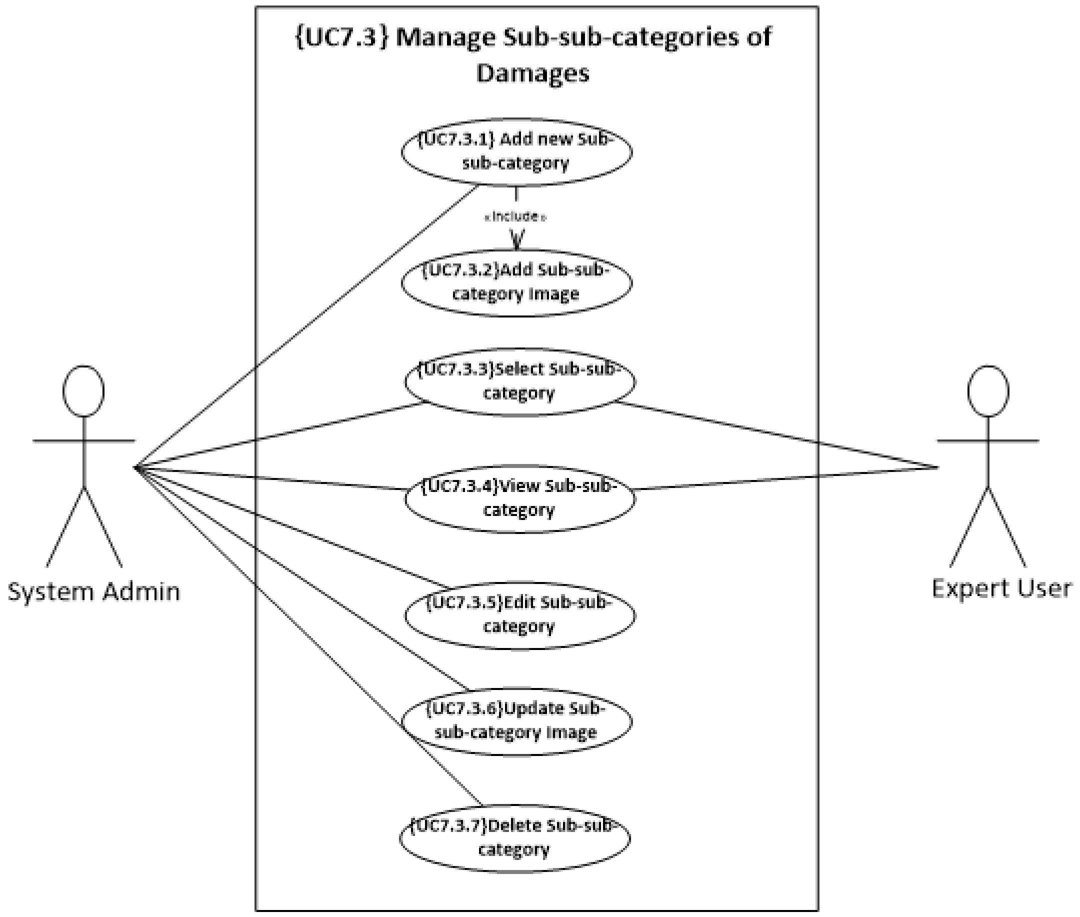
{UC7.3} Manage Sub-sub-categories of Damages
{UC7.3.1} Add new Sub-sub-category
«include»
{UC7.3.2}Add Sub-sub-category Image
{UC7.3.3}Select Sub-sub-category
{UC7.3.4}View Sub-sub-category
{UC7.3.5}Edit Sub-sub-category
{UC7.3.6}Update Sub-sub-category Image
{UC7.3.7}Delete Sub-sub-category
System Admin
Expert User

Figure 15 - {UC7.3}Manage Sub-Sub-Categories of Damages

{UC7.3.1} Add new Sub-sub-category: Allows the System Admin to add a new damage sub-sub-category. To do this, he has to insert the code, name, and description, Necessary parameters to characterize the damage, possible causes, possible consequences, and interrelations between damage, possible mitigation/prevention action, and equivalent terms to be found in other glossaries. Every time this use case is executed the System Admin must enter an image of damage sub-sub-category *({UC7.3.2}Add Sub-sub-category Image).*

{UC7.3.2}Add Sub-sub-category Image: The System Admin after creating a new damage sub-sub-category *({UC7.3.1} Add new Sub-sub-category*) must insert a damage sub-sub-category image. The image must contain a subtitle.

{UC7.3.3}Select Sub-sub-category: Allows the Expert User or the System Admin to select a sub-sub-category from the damage atlas. The Expert User, when doing an inspection (*{UC2.2} Insert Inspection Information*) must first select a damage category *({UC7.1.3}Select Category)* and the damage sub-category *({UC7.2.3}Select Sub-category)* from the damage atlas. Only after selecting the damage category and sub-category, the inspector is able to select the damage sub-sub-category directly from the Inspection form.

{UC7.3.4}View Sub-sub-category: Allows the Expert User or the System Admin to view a sub-sub-category from the damage atlas. The Expert User, when doing an inspection (*{UC2.2} Insert Inspection Information*) can view a damage sub-sub-category from the damage atlas directly from the Inspection form. The Expert User must first select a damage category *({UC7.1.3}Select Category)* and the damage sub-category *({UC7.2.3}Select Sub-category)* to be able to see a damage sub-sub-category.

{UC7.3.5}Edit Sub-sub-category: Allows the System Admin to edit a damage sub-sub-category. The System Admin can edit the name and image of the sub-sub-category. To do this, the damage category, sub-category and sub-sub-category must have been created in *{UC7.1.1} Add new Category, {UC7.2.1} Add new Sub-category and {UC7.3.1} Add new Sub-sub-category*.

{UC7.3.6}Update Sub-sub-category Image: Allows the System Admin to change a sub-sub-damage category image.

{UC7.3.7}Delete Sub-subcategory: A System Admin can delete a Damage sub-sub-category.