**Por: Camilo Carmona Valencia.**

|  |  |
| --- | --- |
| Client | Universidad Icesi, Dagma, CVC, Corpocuencas y la Alcaldía de Santiago de Cali |
| User | * Dagma |
| Functional requirements | * **R1:** The application must be able to register a wetland. * **R2:** The application must be able to register an animal to a wetland. * **R3:** The application must be able to assign an already registered specie to another wetland. * **R4:** The application must be able to register an event to a wetland. * **R5:** The application must be able to show the amount of maintenance visits the wetlands has in any given year. * **R6:** The application must be able to show the name of the wetland with less flora. * **R7:** The application must be able to show each wetland where a given specie is found. * **R8:** The application must be able to show all the information for all of the wetlands except for the events. * **R9 :** The application must be able to show the wetland with the highest number of fauna in it. |
| Problem context | Several state faculties and the icesi university are citing software engineers to create a program to manage the wetlands of Cali. |
| Nonfunctional requirements | * **Rn1:** The application must be easy to maintain and expand. * **Rn2:** The application must be easy to use. * **Rn3:** The application must be fast and efficient. * **Rn4:** The application must be safe and secure. |
| Process requirements | * **Rp1:** |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R1:** The application must be able to register a wetland. | | |
| Summary | The application must have the ability to register all the related data of a wetland and it must be saved into the data base. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Name | String | * Input of an already existing wetland |
| Location | String | * Input of an invalid location |
| Type | WetlandType (Enumeration(public/private)) | * Input of a value different from private or public |
| Size | String | * Input of a value less than or equal to zero |
| Photo’s link | String | * no |
| Percentage of completion of the environment manage plan | double | * input of a value less than zero |
| Zone Type | WetlandLocation(Enumeration(urban/rural)) | * input of a value different from urban or rural |
| Zone name | String | * no |
| General summary of activities | 1. Read and save the information 2. Add a new object of type wetland to the array of wetlands | | |
| Result | The new wetland is added to the array | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R2:** The application must be able to register an animal to a wetland. | | |
| Summary | The application must have the ability to register all the related data of a specie and it must be saved into the data base. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Name | String | * no |
| Scientific name | String | * no |
| Migratory status | Boolean | * input of a value different from true or false. |
| Type | SpecieType(Enumeration(fauna/flora)) | * Input of a value different form fauna or flora. |
| Specific type | SpecificType(Enumeration(Terrestrial/Aquatic/Bird/Mammal)) | * Input of a value different form Terrestrial, Aquatic, Bird or Mammal. |
| Wetlands where the specie is found | Wetland [] | * Input of an non-existent wetland |
| General summary of activities | 1. Read and save the information 2. Add the new specie to the database and to the wetlands where is found | | |
| Result | The new specie is added to the array | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R3:** The application must be able to assign an already registered specie to another wetland. | | |
| Summary | The application must have the ability to assign an already existing specie to a wetland it isn’t in yet. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Specie | Specie | * Input of a non-existent specie |
| Wetland | Wetland | * Input of a non-existent wetland |
| General summary of activities | 1. Find the wetland 2. Find the specie 3. Assign the specie to the wetland | | |
| Result | The specie is assigned to the wetland | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R4:** The application must be able to register an event to a wetland. | | |
| Summary | The application must have the ability to register all the related data of an event. | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Type | EventType(Enumeration(Maintenance/School Visit/ Bettering Activities/Celebration)) | * Input of something other than Manintence, School Visit, Bettering Activities, Celebration. |
| Organizer | String | * no |
| Price | double | * Input of a value less than zero |
| Description | String | * no |
| Date | Date | * no |
| Wetland | Wetland | * input of a non-existent wetland |
| General summary of activities | 1. Read and save the information 2. Add a new object of type Event to the array of Events in the specified wetland | | |
| Result | The event is assigned to the wetland | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R5:** The application must be able to show the amount of maintenance visits the wetlands has in any given year. | | |
| Summary | The application must have the ability to show the amount of maintenance events there are in a year given by the user | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Year | int | * Input is less than zero |
| Wetland | Wetland | * Input of a non-existent wetland |
| General summary of activities | 1. Find the wetland 2. Check all the events 3. Give you the amount of maintenance in the year given | | |
| Result | The number is printed | | |
| Outputs | **Output name** | **Data type** | Re-do condition |
| Number of maintenances in the year | int | No |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R6:** The application must be able to show the name of the wetland with less flora. | | |
| Summary | The application must have the ability to find the wetland with the least number of species of flora and show it. | | |
| General summary of activities | 1. Find the wetland with the least flora 2. Print it | | |
| Result | It is printed | | |
| Outputs | **Output name** | **Data type** | Re-do condition |
| Wetland with the least flora | Wetland | No |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R7:** The application must be able to show each wetland where a given specie is found. | | |
| Summary | The application must have the ability to show all the wetlands a specie inhabits | | |
| Inputs | **Input name** | **Data type** | **Retry condition** |
| Specie | Specie | * Input of a non-existing Specie |
| General summary of activities | 1. Search the specie 2. Return all of the wetlands where it is found | | |
| Result | It is printed | | |
| Outputs | **Output name** | **Data type** | Re-do condition |
| Wetlands where is found | String | No |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R8:** The application must be able to show all the information for all the wetlands except for the events. | | |
| Summary | The application must have the ability to display all of the information for all of the wetlands | | |
| General summary of activities | 1. Collects all of the information 2. Prints it | | |
| Result | It is printed | | |
| Outputs | **Output name** | **Data type** | Re-do condition |
| All wetlands | String | No |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | * **R9:** The application must be able to show the wetland with the highest number of faunae in it. | | |
| Summary | The application must have the ability to find the wetland with the highest number of species of fauna and show it. | | |
| General summary of activities | 1. Find the wetland with the most fauna 2. Print it | | |
| Result | It is printed | | |
| Outputs | **Output name** | **Data type** | Re-do condition |
| Wetland with the most fauna | Wetland | No |