**Problem analysis**

|  |  |
| --- | --- |
| **Client:** | * Cali’s mayor’s office |
| **Users:** | * Public officials |
| **Functional requirements:** | * Req#1: Register a community. * Req#2: Register a place. * Req#3: Register a product from a community. * Req#4: Register a species in a place. * Req#5: Eliminate a product from a community. * Req#6: Modify species data in a place. * Req#7: Creation of test cases. * Req#8: Consult the information of a place. * Req#9: Consult the information of the communities in a department. * Req#10: Consult the information of communities based on their problematic. * Req#11: Consult the name of the place with the most species diversity. * Req#12: Consult the three places with the biggest area. |
| **Context of the problem:** | * The 16th edition of the UN’s Climate Change Conference of the Parties (COP) is an important event to address the conservation of biodiversity and ecosystems. This year, the event is being held in Cali this year and for the reason, the Mayor’s Office is looking to make a program that will let them register, organize, and keep track of the biodiverse places, the communities that take care of those places, and the products the communities sell. * The public officials should be able to utilize the program to save information of these biodiverse places (and their species), communities and products. Additionally, the program should allow them to consult, modify or eliminate information depending on the type of data. Lastly, the program should allow the consultation of the most diverse place and the top three with the biggest area. * The program should be able to work with two separate menus (administrative and consultation). It should be able to be manipulated by multiple users at a time, be organized and be able to handle a big amount of information. |
| **Nonfunctional requirements:** | * Interoperability: The program should be able to handle multiple users at a time. * High usability: The program should be easy to understand while entering, editing, or consulting information. * High availability: The program should be working 24/7 and outage risks mitigated. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#1: Register a community. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Name | String | * Character > 2 * Mandatory |
| Type | CommunityType | * Must be one of the following options:   + Afro Colombian   + Indigenous   + Raizal * Mandatory |
| Representant’s Name | String | * Character >2 * Mandatory |
| Representant’s Phone Number | String | * Character == 10 * Mandatory |
| Population | Int | * Int > 0 * Mandatory |
| Greatest Challenges | Challenge | * Must be one of the following options:   + Absence of hospitals   + Lack of schools   + Inadequate access to clean water   + Insufficient access to food |
| Product Inventory | Product[] | * Empty array of 20 elements (it will be filled afterwards) * Mandatory |
| **Results or postcondition** | The program allows the user to register a new community using the name, type, representants name, representants phone number, population, greatest challenges, and the product inventory and a message of success will appear. In case the community is not registered a message of error, incomplete information or invalid information will print depending on the situation. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the community was registered successfully will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the community to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#2: Register a place. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Name | String | * Character > 2 * Mandatory |
| Department | Department | * Must be one of the following options:   + Choco   + Valle   + Cauca   + Nariño * Mandatory |
| Area | Double | * Double > 0 * Mandatory |
| Type | PlaceType | * Must be one of the following options:   + Protected area   + National Park   + Private area |
| Inauguration date | Date | * dd-mm-aaaa * Mandatory |
| Photo | String | * URL of the selected image * Only one image * Mandatory |
| Caregiving Community | Community | * Can only be one of the preregistered communities. * Mandatory |
| Economic Resources | Double | * Double > 0 * Mandatory |
| Species | Specie[] | * Empty array with 50 elements (it will be filled afterwards) * Mandatory |
| **Results or postcondition** | The place is registered with all the mandatory information: Name, department, area, type, inauguration date, photo, caregiving community, economic resources, and species. If done correctly, a message of success will be printed. If there are any errors, the appropriate messages of error, incomplete or invalid information will de be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the place was registered successfully will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the place to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#3: Register a product to a community. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Option | Int | * Must be one of the following options:   + 1 = Existent   + 2 =   + New * Mandatory |
| Community name | String | * The name is extracted from one of the preexisting communities already created. * Mandatory |
| Product name | String | * Characters > 2 * Mandatory |
| Natural materials percentage | Double | * Double between 0 and 100 * Mandatory |
| Type | ProductType | * Must contain one of the following options:   + Food   + Craft * Mandatory in case the Option 2 is selected |
| Handcraft | String | * Only options are:   + Yes   + No * Mandatory in case the Option 2 is selected |
| **Results or postcondition** | The user must select if he wants to register an existent product or a new one. For both cases, it should be entered the name of the community and the name of the product. If a new product is going to be registered, then it should also be provided the Natural materials percentage, the type, and the handcraft condition. After the user enters all the necessary information, a message that a new product or an existent product has been added to the community. Otherwise, a message of error, incomplete information or invalid information will be printed depending on the cause. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Added New Product | String | A message stating that the product was created and then registered successfully to a community will be printed. |
| Added Existent product | String | A message printing that the existent product has been added to the community. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the product to be registered will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#4: Register a species in a place. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Option | Int | * Must be one of the following options:   + 1 = Existent   + 2 =   + New * Mandatory |
| Place name | String | * The name should already make part of one of the preexisting places in the system. * Mandatory |
| Species name | String | * Character > 2 * Mandatory |
| Type | SpeciesType | * Must select one of the following options:   + Flora   + Fauna * Mandatory in case the Option 2 was selected |
| Photo | String | * URL of the selected image * Only one image * Mandatory if Option 2 was selected |
| Local population count | Int | * Int > 0 * Mandatory if Option 2 was selected |
| **Results or postcondition** | The user must select if he wants to register an existent species or a new species. For both cases, the user must input the Place name and the species name. If he selects Option 2, then the SpeciesType, a photo, and the local population count must be added. After the user enters all the necessary information, a message that a new species or an existent species has been added to a Place. If not, the appropriate message of error, incomplete information or invalid information will be printed depending on the situation. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Added New species | String | A message stating that the species was created and then registered successfully to a place will be printed. |
| Added Existent species | String | A message printing that the existent species has been added to the place. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the species to be registered to a place will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#5: Eliminate a product from a community. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Community Name | String | * The name should already make part of one of the preexisting communities in the system. * Mandatory |
| Product Name | String | * The name should already make part of one of the preexisting products in the system. * Mandatory |
| **Results or postcondition** | After the user inputs the community’s name and the product, the system will erase the product from the inventory of the community and a success message will be printed. If there are any errors, an error message, an incomplete information, or an invalid information will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the product was successfully eliminated form a community will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow product to be deleted from the community will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#6: Modify species data in a place. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Place name | String | * The name should already make part of one of the preexisting places in the system. * Mandatory |
| Species name | String | * The name should already make part of one of the preexisting species inside a place in the system. * Mandatory |
| Type | SpeciesType | * Must select one of the following options:   + Flora   + Fauna * Optional |
| Photo | String | * URL of the selected image * Only one image * Optional |
| Local population count | Int | * Int > 0 * Optional |
| **Results or postcondition** | The system should allow the user to enter the name of a preexisting species inside a preexisting place so the information can be modified. Afterwards, the user must enter the values of the variables he wishes to change (Type, Photo, and/or local population count). Next, the program shows a message of success showing that the information of the species inside the place was modified. In other cases, the message of error, incomplete information or invalid information will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the information of a species inside a place was modified correctly will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the information of a species inside a place be modified will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#7: Creation of test cases. | | |
| **Summary:** | ?????????????????????? | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
|  |  |  |
|  |  |  |
| **Results or postcondition** |  | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#8: Consult the information of a place. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Place name | String | * The name should already make part of one of the preexisting places in the system. * Mandatory |
| **Results or postcondition** | After the user enters the name of the place, the system will display all the information that is stored about that place. If the information is not printed, a message of error, incomplete information or invalid information will be printed instead (depending on the situation). | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Place information | String | The entire information of a place will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the information to be printed |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#9: Consult the information of the communities in a department. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Department name | Department | * Must select one of the following options:   + Choco   + Valle   + Cauca   + Nariño |
| **Results or postcondition** | The system will print out all the information (grouped by community) about all the communities that are inside the selected department. If the information can’t be printed, it will show a message of error, incomplete information, or invalid information depending on the cause of the error. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Community information | String | The entire information of the communities that are registered inside the same department will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the information to be printed will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#10: Consult the information of communities based on their problematic. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Main problematic | Challenge | * Must select one of the following options:   + Absence of hospitals   + Lack of schools |
| **Results or postcondition** | The system will print out all the information (grouped by community) about all the communities that show either of those main problematics (absence of hospitals or lack of schools). If the information can’t be printed, it will show a message of error, incomplete information, or invalid information depending on the cause of the error. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Community information | String | All the information stored about the communities that had the selected problematic registered, will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the information to be printed will be printed. |
| Incomplete information | String | A message stating that not all the mandatory fields where filled will be printed. |
| Invalid information | String | A message stating that there are mistakes in the inputted information (don’t follow the conditions) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#11: Consult the name of the place with the most species diversity. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| N/A | N/A | N/A |
| **Results or postcondition** | The system will use the information stored of every place and then compare which has the most registered species. Afterwards it will order it and print the name of the most diverse place. If the name is not printed, a message of error will appear. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Most diverse place | String | A message stating that the place was registered successfully will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the place to be registered will be printed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#12: Consult the three places with the biggest area. | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| N/A | N/A | N/A |
| **Results or postcondition** | The system will compare the area information of the registered places, organize it, and print the names of the top three places with the biggest area. If the names are not printed, a message of error will appear. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Success | String | A message stating that the place was registered successfully will be printed. |
| Error | String | A message stating there was an error of unknown cause that didn't allow the place to be registered will be printed. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier and Name:** | Req#13: Selecting the menu | | |
| **Summary:** |  | | |
| **Inputs** | **Input name** | **Data Type** | **Conditions for valid values** |
| Menu Type | Enumeration | * The user must select one of the following options:   + Administrative   + Consulting * Mandatory |
|  | Administrative menu | Enumeration |  |
|  | Consulting menu | Enumeration |  |
| **Results or postcondition** | The program allows the user to select the menu he wants to use (administrative or consulting). After selecting one of the types of menus, the options should be displayed which then he can access. If the menu is not displayed, then an error message will be printed. | | |
| **Outputs** | **Output Name** | **Data type** | **Format** |
| Menu | String | A message printing the name of the menu and displaying all the options inside each type of menu |
| Error | String | A message stating there was an error of unknown cause that didn't allow the place to be registered will be printed. |