# Problem Identification and Requirements Analysis

## Case Study:

|  |  |
| --- | --- |
| Customer | Town hall |
| User | Administrator & User |
| Context of the problem | The biodiversity COP will be held in Cali between October 21 and November 01, 2024. This international event focuses on the conservation of biodiversity and ecosystem services. The aim is to develop an application to learn about and navigate places with biological diversity and the communities that protect them in the Pacific region. For species, details such as name, department, area, type, opening date, photo, responsible community, necessary economic resources, and inhabiting species must be provided. As for the products, the name, type, if it is handmade and the percentage of natural products used are specified. For communities, it includes the name, type, representative, cell phone number, number of inhabitants, problems faced and the inventory of products offered. For places with biological diversity, information such as name, department, area in square kilometers, type, date of opening, photo, responsible community, necessary economic resources and inhabitant species are required. In addition, it should be allowed to add products to a community's inventory, specifying the name, type, whether it is handmade, and the percentage of natural products used. |
| Functional Requirements | RF1 Register Species  RF2 Register Product  RF3 Register Community  RF4 Register Place  RF5 Entering a Product into a Community  RF6 Removing a Product from a Community  RF7 Modifying Species Data  RF8 Querying a Place's Information  RF9 Consult the information of the communities by department  RF10 Querying Community Information by Problem  RF11 Consultation of the place with the largest number of species  RF12 Query of the three largest places per square kilometer  RF13 Administrative Single Menu  RF14 Single Menu Queries |
| Non-functional requirements | RNF1 Intuitive & User-Friendly Interface |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF1-Register Species]* | | | |
| Summary | *The system must allow the registration and storage of the species taking into account the name, whether it is flora or fauna, a photo (access path) so that it can later be added to a place. If the registration was successful, it will show a favorable message, otherwise it will show an error message and the user will have to enter the data again.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Name | String | |  |
| Guy | String | | You must select between whether it is Flora or Fauna |
| Photo | String | | A path to the image |
| Result or Postcondition | Once all data has been entered, a successful registration message should be saved and displayed | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Successful Message | | String | Displayed when the record was successfully saved |
| Unsuccessful Message | | String | Displayed when there was an error in log storage |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF2-Register Product]* | | | |
| Summary | *The system should allow the registration and storage of products that can be made by the communities and then added to each community, of which the name, the percentage of natural products used in their manufacture, the type (food or handicrafts) and whether it was made by hand (Yes or No) are kept. If the registration was successful it will show a favorable message, if not, the system will show an unsuccessful registration message and the user will have to enter the information again.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Name | String | |  |
| Percentage of natural products used in their manufacture | Double | | xx.x% |
| Guy | String | | The user selects whether it's food or handicrafts |
|  | Handmade | Boolean | | The user selects whether the product was handmade or not |
| Result or Postcondition | Once all data has been entered, a successful registration message should be saved and displayed | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Successful Message | | String | Displayed when the record was successfully saved |
| Unsuccessful Message | | String | Displayed when there was an error in log storage |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF3-Register Community]* | | | |
| Summary | *The system must allow the registration and storage of the communities, for this the name must be stored, the type of community if it is Afro-Colombian, indigenous or Raizal, the name and cell phone of the representative, the number of inhabitants, the major problems faced by the community (lack of hospitals, schools, etc.) and products that the community offers must be added. with a maximum of 20 products per community. If the registration was successful it will show a favorable message, if not, the system will show an unsuccessful registration message and the user will have to enter the information again* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Name | String | |  |
| Type of community | String | | The user selects an Afro-Colombian, Indigenous, or Raizal community type |
| Representative's Name | String | |  |
| Representative's Phone Number | String | |  |
| Number of inhabitants | Int | | xxxx |
| Bigger Problems | Array | | Multiple issues can be selected |
| Products They Offer | Array | | It is selected from a list of products, the ones offered by the particular community |
| Result or Postcondition | Once all data has been entered, a successful registration message should be saved and displayed | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Successful Message | | String | Displayed when the record was successfully saved |
| Unsuccessful Message | | String | Displayed when there was an error in log storage |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF4-Register Place]* | | | |
| Summary | *The system must allow the registration of places, saving information on the name, the department, the number of square kilometers, the type (if it is a protected area, national park or private area), the date of inauguration, photo of the place, community that cares for it, economic resources necessary for good care, the species that inhabit it and the amount for each species selected. If the registration was successful it will show a favorable message, if not, the system will show an unsuccessful registration message and the user will have to enter the information again.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Name | String | |  |
| Name of the department | String | | The user selects a department (Chocó, Valle, Cauca, etc.) |
| Number of square kilometers | Double | | xxKm |
| Guy | String | | The user selects a type of place (Protected Area, National Park or Private Area) |
| Opening Date | Date | | dd-mm-yyyy |
| Photo of the place | String | | Access to the image |
| Community that cares for her | String | | From the list of registered communities, the community belonging to the place is selected |
| Economic Resources | Double | | $xxxxxxx |
| Inhabiting Species | Array | | From the list of species in the system, the species that inhabit the place are selected |
| Number of species | Array | | The quantity of each selected species is specified |
| Result or Postcondition | Once all data has been entered, a successful registration message should be saved and displayed | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Successful Message | | String | Displayed when the record was successfully saved |
| Unsuccessful Message | | String | Displayed when there was an error in log storage |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF5-Entering a Product into a Community]* | | | |
| Summary | *The system must allow the user to register the products offered by the different communities and thus they are stored in the database. For this to happen, in order to register a product, the following information is needed: name of the product, percentage of natural products used in its manufacture, type (whether food or handicrafts) and also whether it is handmade or not.*  *In this way, the user enters the product information and in the event that any of the requirements is not correct, an "error" message will be shown to the user and it is enabled to re-enter the data correctly.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Name | String | |  |
| Percentage | double | | The user is asked to enter the percentage of natural products used in manufacturing, as follows:  1-100% |
| Type | String | | You must select whether the type of product is food or handicrafts. |
| Handmade | Boolean | | You must indicate if your product is handmade or not, the answer is as follows:  Yes/No |
| Result or Postcondition | Once all the data has been entered, a message must be saved and displayed | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Successful Message | | String | Displayed when the record was successfully saved |
| Unsuccessful Message | | String | Displayed when there was an error in log storage |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF6-Removing a Product from a Community]* | | | |
| Summary | *The system should allow the user to remove products offered by different communities. In order for this to happen, in order to register a product, the following information is needed: Product name*  *Thus, the user enters the name of the product to search for it in the database of this program, and thus it will be shown on the screen if the product is successfully deleted. In order for it to be deleted, it must be registered in the database, otherwise it will show an error for the user to re-enter correctly.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Community Name | String | | You must enter the name of the community you wish to enter, and it is required that this name be previously registered, otherwise an error will appear. |
| Name | String | | *Must be a valid name previously registered in the database* |
| Result or Postcondition | Once entered, the program locates the product thanks to the information of its name, in case it exists, it eliminates it, otherwise it shows "error" | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Successful Message | | String | Displayed when the record was successfully saved |
| Unsuccessful Message | | String | Displayed when there was an error in log storage |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF7-Modify Species]* | | | |
| Summary | *The system allows the user to modify the species according to the place in which it enters in this way. The user is also asked to enter the name of the place and then the name of the species previously recorded. After that, the system asks the user to enter the data they want to modify, whether it is the name, whether it is flora or fauna, their photo and the number of specimens in the place previously entered.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Name Place | String | | *You must select a place which has already been registered, otherwise it will show an error.* |
| Species name | String | | *You must select a species which has already been registered, otherwise it will show an error.* |
| Guy | String | | You must select between whether it is Flora or Fauna |
| Photo | String | | A path to the image |
| Result or Postcondition | Once the species data has been modified, it will be saved back to the database and a message will be displayed to the user | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Successful Message | | String | Displayed when the record was successfully saved |
| Unsuccessful Message | | String | Displayed when there was an error in log storage |

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier & Name | *[RF8- Consult the information of a place]* | | |
| Summary | *The system allows the user to consult the information of the places which have biological diversity, in this way it shows only outputs referring to the attributes of the place* | | |
| **Tickets** | *It doesn't have tickets* | | |
| Outputs | **Ticket name** | **Type of data** | **Condition Valid Values** |
| Name | String |  |
| Name of the department | String | The department to which the place belongs is evident |
| Number of square kilometers | Double | xxKm |
| Guy | String | The type of place is shown to the user |
| Opening Date | Date | dd-mm-yyyy |
| Photo of the place | String | Access to the image |
| Community that cares for her | String | It shows the community that cares for the place |
| Economic Resources | Double | $xxxxxxx |
| Inhabiting Species | Array | The list of species that inhabit the place is displayed |
| Number of species | Array | The quantity of each selected species is specified |

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier & Name | *[RF9- Consult the information of the communities of a department]* | | |
| Summary | *The system allows the user to consult all the information of the communities of a department along with each of its attributes, without the need to repeat the names of the communities since each name is unique and distinctive.* | | |
| Tickets | *It doesn't have tickets* | | |
| Outputs | **Ticket name** | **Type of data** | **Condition Valid Values** |
| Name | String |  |
| Type of community | String | The user is shown the selected community type |
| Representative's Name | String |  |
| Representative's Phone Number | String |  |
| Number of inhabitants | Int | xxxx |
| Bigger Problems | Array | The selected issue is displayed |
| Products They Offer | Array | The list of entered products is displayed |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF10-Consult* the information of the communities by problem*]* | | | |
| Summary | *The system allows the user to consult all the information of the communities by specific problems.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| Problem Finding | String | | *You must select an issue to query* |
| Result or Postcondition | Once you enter the search condition, the system will show you the communities that have that problem | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Name | | String |  |
| Type of community | | String | The user is shown the selected community type |
| Representative's Name | | String |  |
| Representative's Phone Number | | String |  |
| Number of inhabitants | | Int | xxxx |
| Bigger Problems | | Array | The selected issue is displayed |
| Products They Offer | | Array | The list of entered products is displayed |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF11-* Consultation of the place with the largest number of species*]* | | | |
| Summary | *The system allows the user to consult the places with the largest number of species* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
|  |  | |  |
| Result or Postcondition | The system, after validating and sorting the places with the highest number of species, will show the user all the attributes of the places. | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Name | | String |  |
| Name of the department | | String | The department to which the place belongs is evident |
| Number of square kilometers | | Double | xxKm |
| Guy | | String | The type of place is shown to the user |
| Opening Date | | Date | dd-mm-yyyy |
| Photo of the place | | String | Access to the image |
| Community that cares for her | | String | It shows the community that cares for the place |
|  | Economic Resources | | Double | $xxxxxxx |
|  | Inhabiting Species | | Array | The list of species that inhabit the place is displayed |
|  | Number of species | | Array | The quantity of each selected species is specified |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF12-* Consultation of the three largest places per square kilometer*]* | | | |
| Summary | *The system allows the user to check which are the 3 largest places per square kilometer* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
|  |  | |  |
| Result or Postcondition | The system, after validating and sorting, will provide information on three places that are the largest per square kilometer | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| Name | | String |  |
| Name of the department | | String | The department to which the place belongs is evident |
| Number of square kilometers | | Double | xxKm |
| Guy | | String | The type of place is shown to the user |
| Opening Date | | Date | dd-mm-yyyy |
| Photo of the place | | String | Access to the image |
| Community that cares for her | | String | It shows the community that cares for the place |
|  | Economic Resources | | Double | $xxxxxxx |
|  | Inhabiting Species | | Array | The list of species that inhabit the place is displayed |
|  | Number of species | | Array | The quantity of each selected species is specified |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF13-Single* Administrative Menu*]* | | | |
| Summary | *When the system starts, it will show a main menu in which it will have the "administrative panel" section which allows the managed user to choose between the different administrative operations (register, update and delete) after authenticating.* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
| User | String | | *Be compatible with the user stored in the system* |
| Result or Postcondition | The system, after validating that the user is found, will show you the menu of valid options for your user type. | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| 1) Enter a Community. | | Int |  |
| 2) Enter a Place, the name must be unique. | | Int |  |
| 3) Entering and/or removing a product from a community. | | Int |  |
| 4) Bringing a species into a place. | | Int |  |
| 5)Modify the species data at a location. | | Int |  |
| 6)Creation of test cases. | | Int |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier & Name | *[RF14-Single* Menu Queries*]* | | | |
| Summary | *When the system starts, it will show a main menu in which you will have the "administrative panel" section and the consultation options, which can be accessed by any user without the need to authenticate or have a user in the system* | | | |
| Tickets | **Ticket name** | **Type of data** | | **Condition Valid Values** |
|  |  | |  |
| Result or Postcondition | When the system starts, it shows the main options in which the user can choose the different queries | | | |
| Outputs | **Departure Name** | | **Type of data** | **Format** |
| 1)Administrative Panel | | Int |  |
| 2) Consult the information of a place, for which you must display its attributes and photo. | | Int |  |
| 3) Consult the information of the communities of a department, without repetition show the name of the community. | | Int |  |
| 4) Consult the information of the communities whose problems are: that they lack a hospital or that they lack a school. | | Int |  |
| 5) Consult the name of the place with the largest number of species. | | Int |  |
| 6) Check out the three largest places per square kilometer. | | Int |  |