

**WILDLIFE RESERVE TOOL**

**User Manual**

**Asish Anirudhan**

**Tata Elxsi**

# Contents

[1. Contents 2](#_Toc495333078)

[2. Introduction 3](#_Toc495333079)

[3. Application Screens 3](#_Toc495333080)

[4. Design Pattern 6](#_Toc495333081)

[5. Installation Process 7](#_Toc495333082)

[6. The REST APIs 9](#_Toc495333083)

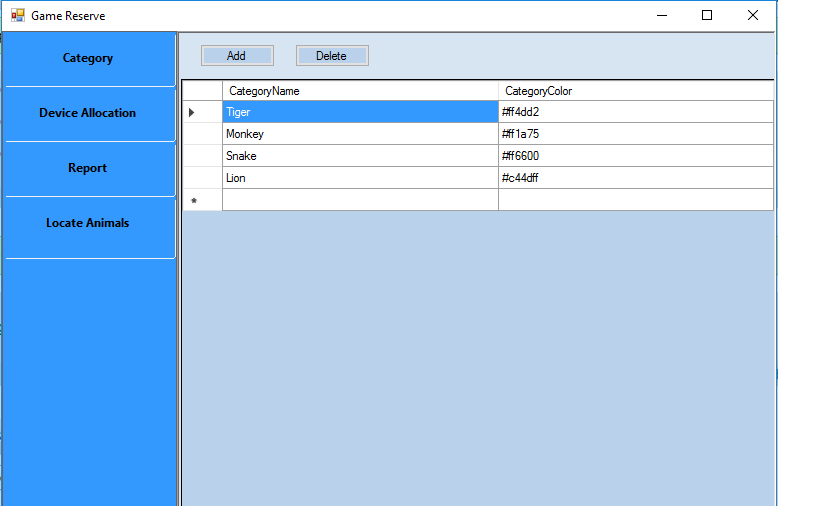
# Introduction

This document contains the information for the working of Wildlife Tracking Tool. This tool is used to track the animals over the period of time. It is used to keep an accurate account of their wildlife, and their movements. This tool is used to store the GPS coordinates for various wildlife to locate the animals over the period of time.

# Application Screens

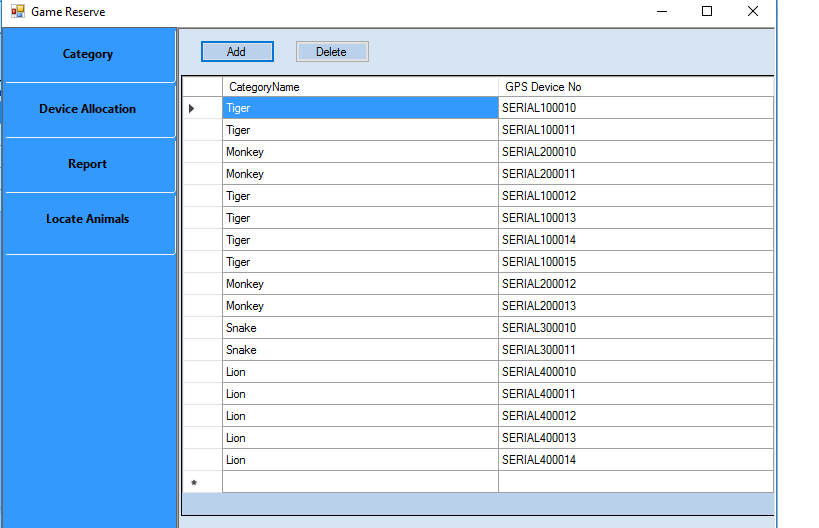
**Listing/Add/Delete Animal Category:**

User can see the different categories of animals. Also user can add/delete animal categories.



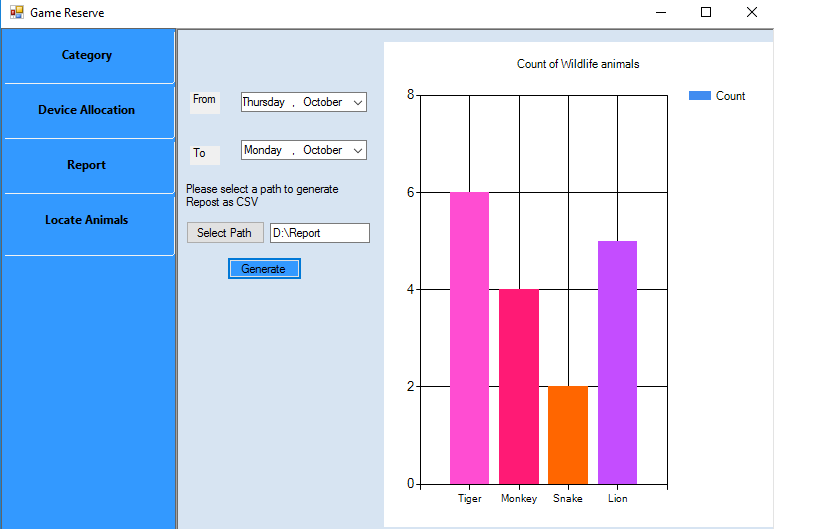
**Allocate GPS device to Animals:**

User can allocate GPS device to each animal in the game reserve. It will help to identify the latest location of each animal. Each device has a unique serial number, which will be recorded in DB once it is allocated to an animal.



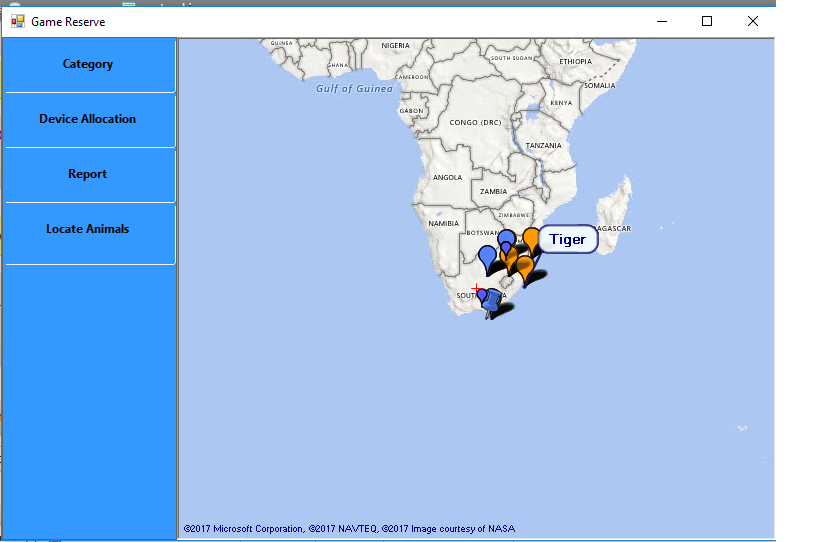
**Reporting:**

Application provides the report in excel format and generate bar chart of all animal categories and its count depends on the date of allocation.



**Locate Animals:**

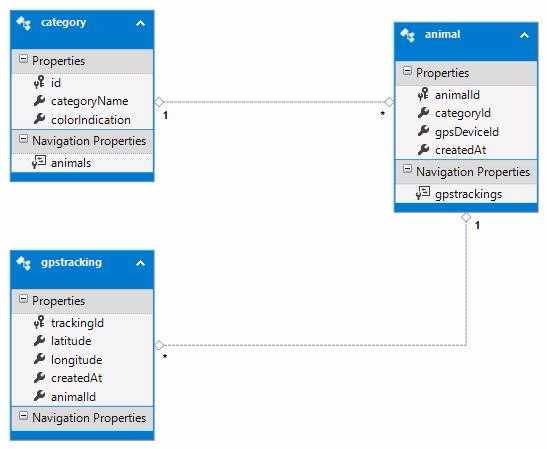
This tab provides the latest position of each animal and shown in a map. Unique symbol is used to represent each animal category in map.



# Html document generation

* For html document generation we are using third party tool named Doxygen.
* Install Doxygen
* From Wizard Option, select enter the Source code directory and Destination Directory.
* Form Mode Option, select the language (Optimize for Java or c# output)
* From Build Option, select the options to be documented.
* From Run Option, select Show HTML output.
* Can access the html document from repository /html\_doc/index.html

# Design Pattern



# Installation Process

1. **DB Setup**

* Mysql databse is used in this project.
* The dump file would be provided named **"game\_reserve.sql"**
* Open MySQL Workbench, Select the local instance, and enter the root password if prompted.
* Select Data Import option from Server>DataImport.
* Select the "Import from Self-Contained file" option and browse to select the DB backup dump provided.
* In the default Target Schema, select New and give the name of Schema that is to be created.
* Click Start Import button.
* Refresh the schemas at left hand side and the newly imported schema will be listed with all the tables and stored procedures.
* Update the "connection string" setting in **\Bin\WebServices\DBService\DBService\bin\web.config** file.

Change username, password and Database.

1. **Deployment Setup**

Installing IIS

* In Windows, access the Control Panel and click Programs.
* In the Programs and Features window, click Turn Windows features on or off.
* Select the Internet Information Services (IIS) check box, expand the node and ensure that the following items are selected.
* Then click OK.
* For the changes to take effect Restart PC.

Deploying services in IIS

* Open IIS by giving “inetmgr” in Run Window.
* Once IIS is opened, Expand the nodes in Connections panel (in left hand side of the window), to reach Default Web Site.
* Right Click Default Web Site and select Add Application.
* In Add Application pop-up window, give the folder name as alias name. For example, while hosting SomeService web service, give the alias name as SomeService.
* Browse the physical path to point to \Bin\WebServices\SomeService
* Click OK to complete adding the application to IIS.

# The REST APIs

**1. To Add a Category**

Method: POST

URL : http://localhost:<port>/CategoryService.svc/SaveCategory

Request Body : {"categoryName":"Giraffee","colorIndication":"#8B0056"}

Response: {

"categoryName": "Giraffee",

"colorIndication": "#8B0056",

"id": 13,

"message": "Created sucessfully"

}

**2. To Retrieve all Category**

Method: GET

URL : http://localhost:<port>/CategoryService.svc/GetAllCategories

Response: {

"GetAllCategoriesResult": [

{

"categoryName": "Tiger",

"colorIndication": "#ff4dd2",

"id": 1

},

{

"categoryName": "Monkey",

"colorIndication": "#ff1a75",

"id": 2 },]}

**3. To Retrieve A Category when Id is specified**

Method: GET

URL : http://localhost:<port>/CategoryService.svc/GetCategory/1

Response: {

"GetCategoryResult": {

"categoryName": "Tiger",

"colorIndication": "#ff4dd2",

"id": 1

}

}

**4. To Update a Category**

Method: POST

URL : http://localhost:<port>/CategoryService.svc/UpdateCategory

Request Body : {"categoryName":"Tiger9","colorIndication":"#FF0000", "id":1}

Response: {

"categoryName": "Tiger9",

"colorIndication": "#FF0000",

"id": 1,

"message": "Updated Successfully"

}

**5. To Delete A Category**

Method: GET

URL : http://localhost:<port>/CategoryService.svc/DeleteCategory/9

Response: {

"categoryName": "Zebra",

"colorIndication": "#ac397",

"id": 9,

"message": "Deleted successfully."

}

**6. To Add an Animal**

Method: POST

URL : http://localhost:<port>/AnimalService.svc/AddAnimal

Request Body : {"categoryId":1,"gpsDeviceId":"SERIAL100090"}

Response: {

"animalId": 43,

"categoryId": 1,

"createdAt": "2017-10-09 12:44:44",

"gpsDeviceId": "SERIAL100090",

"message": "Created sucessfully"

}

**7. To Retrieve all Animals**

Method: GET

URL : http://localhost:<port>/AnimalService.svc/GetAllAnimals

Response: {

"GetAllAnimalsResult": [

{

"animalId": 1,

"categoryId": 1,

"categoryName": "Tiger9",

"createdAt": "10/5/2017 12:00:00 AM",

"gpsDeviceId": "SERIAL100071"

},

{

"animalId": 2,

"categoryId": 1,

"categoryName": "Tiger9",

"createdAt": "10/5/2017 5:00:00 AM",

"gpsDeviceId": "SERIAL10002"

}]}

**8. To Update an Animal**

Method: POST

URL : http://localhost:<port>/AnimalService.svc/UpdateAnimal

Request Body : {"animalId":1,"categoryId":1,"gpsDeviceId":"SERIAL100071"}

Response: {

"animalId": 1,

"categoryId": 1,

"createdAt": "2017-10-05 00:00:00",

"gpsDeviceId": "SERIAL100071",

"message": "Updated sucessfully"

}

**9. To Delete an Animal**

Method: GET

URL : http://localhost:<port>/AnimalService.svc/DeleteAnimal/36

Response: {

"animalId": 36,

"categoryId": 1,

"createdAt": "2017-10-07 17:58:56",

"gpsDeviceId": "SERIAL100098",

"message": "Deleted successfully."

}

**10. To Get Total Count Of Animals By Category**

Method: GET

URL : http://localhost:<port>/AnimalService.svc/GetTotalCountOfAnimalsByCategory /2017-10-05/2017-10-05

Response: {

"GetTotalCountOfAnimalsByCategoryResult": [

{

"categoryName": "Tiger",

"colorIndication": "#ff4dd2",

"id": 1,

"totalAnimals": 2

} ]}

**11. To Add Tracking**

Method: POST

URL : http://localhost:<port>/TackingService.svc/AddTracking

Request Body : {"latitude":-30.2555,"longitude":31.2555,"gpsDeviceId":"SERIAL20002"}

Response: {

"animalId": 4,

"createdAt": "/Date(1507534165000+0530)/",

"latitude": -30.2555,

"longitude": 31.2555,

"message": "Created sucessfully",

"trackingId": 10

}

**12. To Get Latest Position of Animal**

Method: GET

URL : http://localhost:<port>/ TackingService.svc/GetLatestPositionOfAnimal Response: {

"GetLatestPositionOfAnimalResult": [

{

"animalId": 1,

"catId": 1,

"categoryName": "Tiger",

"colorIndication": "#ff4dd2",

"createdAt": "/Date(1507145400000+0530)/",

"gpsDeviceId": "SERIAL100071",

"latitude": -25.966688,

"longitude": 32.58052,

"trackingId": 2

}]

}