# SYNOPSIS REPORT

# ON

IP Locator

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Title** | **Page No.** |
| 1 | Abstract | 2 |
| 2 | Introduction | 2 |
| 3 | Literature Review | 3 |
| 4 | Requirements | 3 |
| 5 | Problem Statement | 3 |
| 6 | Objectives | 3 |
| 7 | Design | 4-5 |
| 8 | Methodology | 6-8 |
| 9 | PERT chart | 9 |
| 10 | References | 9 |

1. **Abstract**

IP addresses and logs are the fundamental blocks of networking and system reports that are often ignored. IP addresses are the given names of a system that it is known by throughout the network. The gateway to every network attack is the reconnaissance of fetching the IP address details of a machine. People can look at their IP addresses and know whether they have their VPN on or not. Logs are the next most important blocks of information that can intercept any malicious activity both on the internet as well as on the machine - since each system creates a log report of every action performed on a system. Our goal is to analyze logs to find any vulnerabilities and suspicious activities.

# Introduction

Concisely, IP addresses are the identifiers that make devices on a network available for communication and allow information to be exchanged between them. They also carry geographic information. There must be a means for computers, routers, and webpages to be distinguished on the internet. A method for achieving this is provided by IP addresses, which are crucial to the operation of the internet. No two IP addresses are the same. The Internet Assigned Numbers Authority (IANA), a part of the Internet Corporation for Assigned Names and Numbers, produces and distributes them mathematically (ICANN). To assist maintain the security of the internet and make it accessible to everyone, ICANN was founded in the United States in 1998. Every time someone registers a domain on the internet, they do so through a domain name registrar, who then pays ICANN a nominal fee to do so.Another subject that is considered an important part of the networking and Ip addresses is the algorithms that is used for the logic building. An algorithm is defined as a finite set of instructions that are followed in problem solving operations or calculations. Mainly when it comes to finding the Ip address of a machine the type of algorithms that come in our mind are the searching and sorting algorithms like Binary search. Binary search is a searching algorithm that is used for sorting the data with which we are working by repeatedly dividing the search interval in half. The main idea behind the use of binary search is to sort the data in such a manner that it reduces the time complexity of the program and thus helps us in finding the results more quickly and accurately.

# Literature Review

We can find a lot of websites related to IP location on the internet that not only give us the IP address of a target but also other essential details that are useful for an attacker.

The most prominent one of them is WhatIsMyIpAddress.com whose features are listed below:

* + It can find the IP address of a system version 4 as well as version 6.
  + It can detect whether a client is requesting IP sitting behind a proxy server or not.
  + It can fetch other attributes of an IP address too like the location of the client, and Internet Service Provider.
  + It can retrieve the DNS by inputting its IP address.

# Requirements

1. Java for programming the backend for the application. The project will be running on Java SDK 17 - since v17 offers long-term support which means the mentioned SDK will be getting updates for the next four years.
2. Java has classes like Arraylist that will be helpful in achieving goals and tackling problems that will be forthcoming.

# Problem Statement

Even though there are websites that can provide the IP address, most of those websites are using APIs of the specific company that is providing with data set of the IP addresses. The only way to get the location is to import their APIs.

# Objectives

* 1. To understand how IP addresses are mapped.
  2. To check the IP address of a system which will include localhost IP and Public IP
  3. To read the .csv format file containing the data including network addresses and splitting them, taking input from the user and comparing them.
  4. To read the .csv format file containing data mapped as geoid and the corresponding continent and country.

# Design

# 

Data flow diagram (Level - 0)

# Module 1

# network,geoname\_id,registered\_country\_geoname\_id,represented\_country\_geoname\_id,is\_anonymous\_proxy,is\_satellite\_provider

# 1.0.0.0/24,2077456,2077456,,0,0

# 1.0.1.0/24,1814991,1814991,,0,0

# 1.0.2.0/23,1814991,1814991,,0,0

# 1.0.4.0/22,2077456,2077456,,0,0

# 1.0.8.0/21,1814991,1814991,,0,0

# 1.0.16.0/20,1861060,1861060,,0,0

# 1.0.32.0/19,1814991,1814991,,0,0

# 1.0.64.0/18,1861060,1861060,,0,0

# 1.0.128.0/17,1605651,1605651,,0,0

# The data is arranged in the format mentioned above in the .csv files.

# Module 2

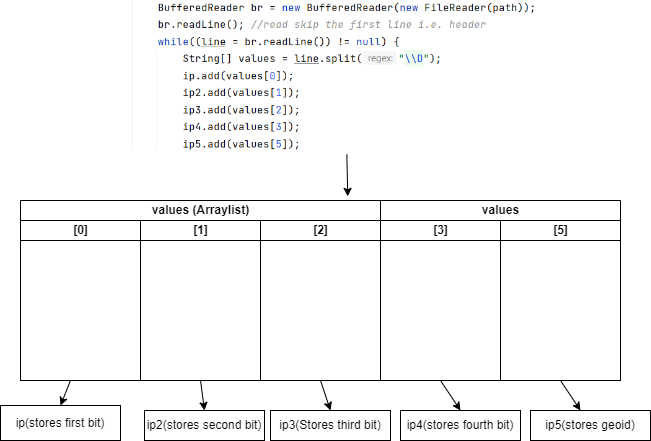
# The code below Fetches the local host IP address.

import java.net.InetAddress;  
  
public class ipaddress{  
 public static void main(String[] args) {  
 try{  
 *//get public address and hostname of the localmachine* InetAddress myIP = InetAddress.*getLocalHost*();  
 System.*out*.println(myIP.getHostAddress());  
 }  
 catch (Exception ex){  
 System.*out*.println(ex.getMessage());  
 }  
 }  
}

# Module 3

# The above code reads the .csv file splits it in five different arraylist.

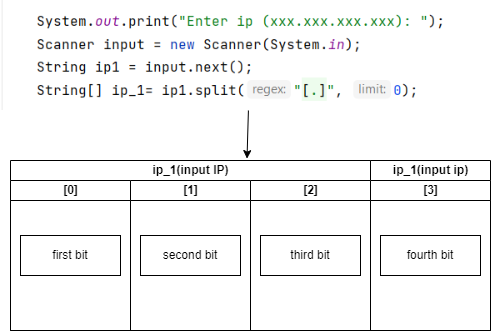
try {  
 BufferedReader br = new BufferedReader(new FileReader(path));  
 br.readLine(); *//read skip the first line i.e. header* while((line = br.readLine()) != null) {  
 String[] values = line.split("\\D");  
 ip1.add(values[0]);  
 ip2.add(values[1]);  
 ip3.add(values[2]);  
 ip4.add(values[3]);  
 ip5.add(values[5]);  
 }  
} catch (IOException e) {  
 e.printStackTrace();  
}



Data is split using regex and stored in Arraylist

1. The code below takes the user input as IP and splits it into four arraylist.

System.*out*.print("Enter ip (xxx.xxx.xxx.xxx): ");  
Scanner input = new Scanner(System.*in*);  
String inputip = input.next();  
String[] input\_ip= inputip.split("[.]", 0);



Input ip is split and stored each bit is stored separately

1. The code below finds the first occurrence of the first bit of the input IP in the arraylist of the mapped IP first bits.

static int firstoccurrence(int[] arr, int target, int start, int end) {  
 while (start <= end) {  
 int mid = start + (end - start) / 2;  
  
 if ((mid == 0 || arr[mid - 1] < target) && arr[mid] == target)  
 return mid;  
 else if (target > arr[mid])  
 start = mid + 1;  
 else  
 end = mid - 1;  
 }  
 return -1;  
}

1. The code above finds the last occurrence of the first bit of the input IP in the arraylist of the mapped IP first bits.

static int lastoccurrence(int[] arr, int target, int start, int end) {  
 while (start <= end) {  
 int mid = start + (end - start) / 2;  
  
 if ((mid == end || arr[mid + 1] > target) && arr[mid] == target)  
 return mid;  
 else if (target < arr[mid])  
 end = mid - 1;  
 else  
 start = mid + 1;  
 }  
 return -1;  
}

Next, we use the same methods for the first occurrence and last occurrence but the starting and end points of the list changes after finding the precursor start and end points. This is done because the first octate of the IPv4 address will remain the same for all the combinations for the remaining octates of the IPv4 address.

1. After finding the exact same IP in the dataset, the corresponding Geoid is found using the above code and input in the next module.

static int search\_ip4(int[] arr, int target, int start, int end) {  
 while (start <= end) {  
 int mid = start + (end - start)/2; *// same as (start + end) / 2* if(target < arr[mid]) {  
 end = mid - 1;  
 } else if (target > arr[mid]) {  
 start = mid + 1;  
 } else {  
 return mid;  
 }  
 }  
 return -1;  
}

* **Module 4**

1. The data is present in the above format in the .csv file.

geoname\_id,locale\_code,continent\_code,continent\_name,country\_iso\_code,country\_name,is\_in\_european\_union

49518,en,AF,Africa,RW,Rwanda,0

51537,en,AF,Africa,SO,Somalia,0

69543,en,AS,Asia,YE,Yemen,0

99237,en,AS,Asia,IQ,Iraq,0

1. The below code reads the .csv file, splits the data and saves them in arraylist in different arraylist named geo\_id, con\_id and coun\_id.

try {  
 BufferedReader br = new BufferedReader(new FileReader(path));  
 br.readLine(); *//read skip the first line i.e. header* while((line = br.readLine()) != null) {  
 String[] values = line.split(",");  
 geo\_id.add(values[0]);  
 con\_id.add(values[3]);  
 coun\_id.add(values[5]);  
 }  
} catch (IOException e) {  
 e.printStackTrace();  
}

1. The code below is used to find the index where the same geoid is stored.

static int search\_ip4(int[] arr, int target, int start, int end) {  
 while (start <= end) {  
 int mid = start + (end - start)/2; *// same as (start + end) / 2* if(target < arr[mid]) {  
 end = mid - 1;  
 } else if (target > arr[mid]) {  
 start = mid + 1;  
 } else {  
 return mid;  
 }  
 }  
 return -1;  
}

# Methodology

# First the user will access the website and click on the button to request the IPv4 of the device.

# The front-end made using java swing will fetch IPv4 using the InetAddress class of the java.

# The IPv4 address fetched using the front-end will go into the backend program which will then output the location corresponding to IPv4 address.

# 

# The program will work in the following way: -

# a) The data set in the .csv format is read using BufferedReader class of java line by line.

# b) Since the data set is in the .csv format, the entries are separated with commas.

# c)The entries are then stored using Arraylist class of the java because Arraylist provides

# us with dynamic array.

# d)The entries of .csv file are split using the split () method of java using regex.

# e) Each digit before decimal is stored in a different array. Therefore, four Arraylist are made.

# f) The input IP is then split using the same method mentioned above.

# g) Then the first and last occurrence of the first bit of the IP is searched used binary

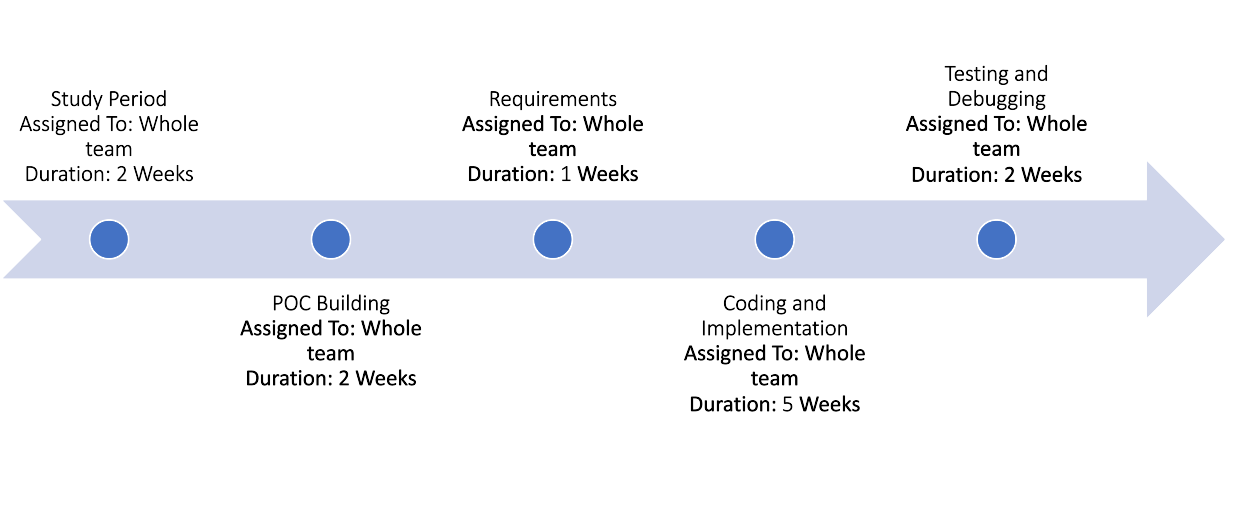
# search.

# h) The first and last occurrence is then used as input for the binary search for the second bit and so on for the last bit.

# i) After finding the corresponding index for the required IP, the program then finds the corresponding geoid.

# f) The country class then uses the binary search find the corresponding continent and country.

# PERT Chart

****

1. **References**

[1] Computer Networking - A Top-Down Approach, 8th Edition by James F. Kurose University of Massachusetts, Amherst Keith W. Ross Polytechnic Institute of NYU – Chapters - [4] section - [4.3]- The Internet Protocol.