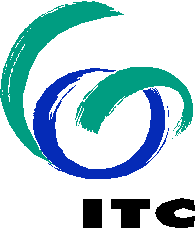
**Geo-Located Audio Tour Guide**

**Submitted By:**

* Aman Tiwari
* Appalla Sai Manoj
* Kanchi Gangadhar
* V Venkata Siva Reddy



FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

UNIVERSITY OF TWENTE, ENSCHEDE, THE NETHERLANDS

***iirs***

INDIAN INSTITUTE OF REMOTE SENSING,

INDIAN SPACE RESEARCH ORGANISATION, DEHRADUN, INDIA

Contents

[1. Introduction 3](#_Toc109400771)

[1.1. Motivation 3](#_Toc109400772)

[1.2. Geofence 3](#_Toc109400773)

[2. Methodology 4](#_Toc109400774)

[2.1. Flowchart 4](#_Toc109400775)

[2.2. BPMN Diagram 5](#_Toc109400776)

[2.3. Activity Diagram 5](#_Toc109400777)

[3. Results 6](#_Toc109400778)

[4. Conclusions 7](#_Toc109400779)

[5. References 8](#_Toc109400780)

[6. Appendix 8](#_Toc109400781)

[6.1. Code Snippet to access WFS Layer from Geoserver: 8](#_Toc109400782)

[6.2. Code Snippet to execute the logic 9](#_Toc109400783)

[6.3. Code Snippet to Convert text to speech 10](#_Toc109400784)

# Introduction

## Motivation

Travelling is and always will be an integral part of our lives. It is a rich experience which almost everybody cherishes. When we go to some unknown places, we like to get involved in the important places. To do so, most of the time we hire a tour guide which helps us with the area. But travel guide is most useful for large groups as it can be a bit costly. And nowadays, many people like to travel alone, and they want to travel in their own time without the influence of any other person or tour guide.

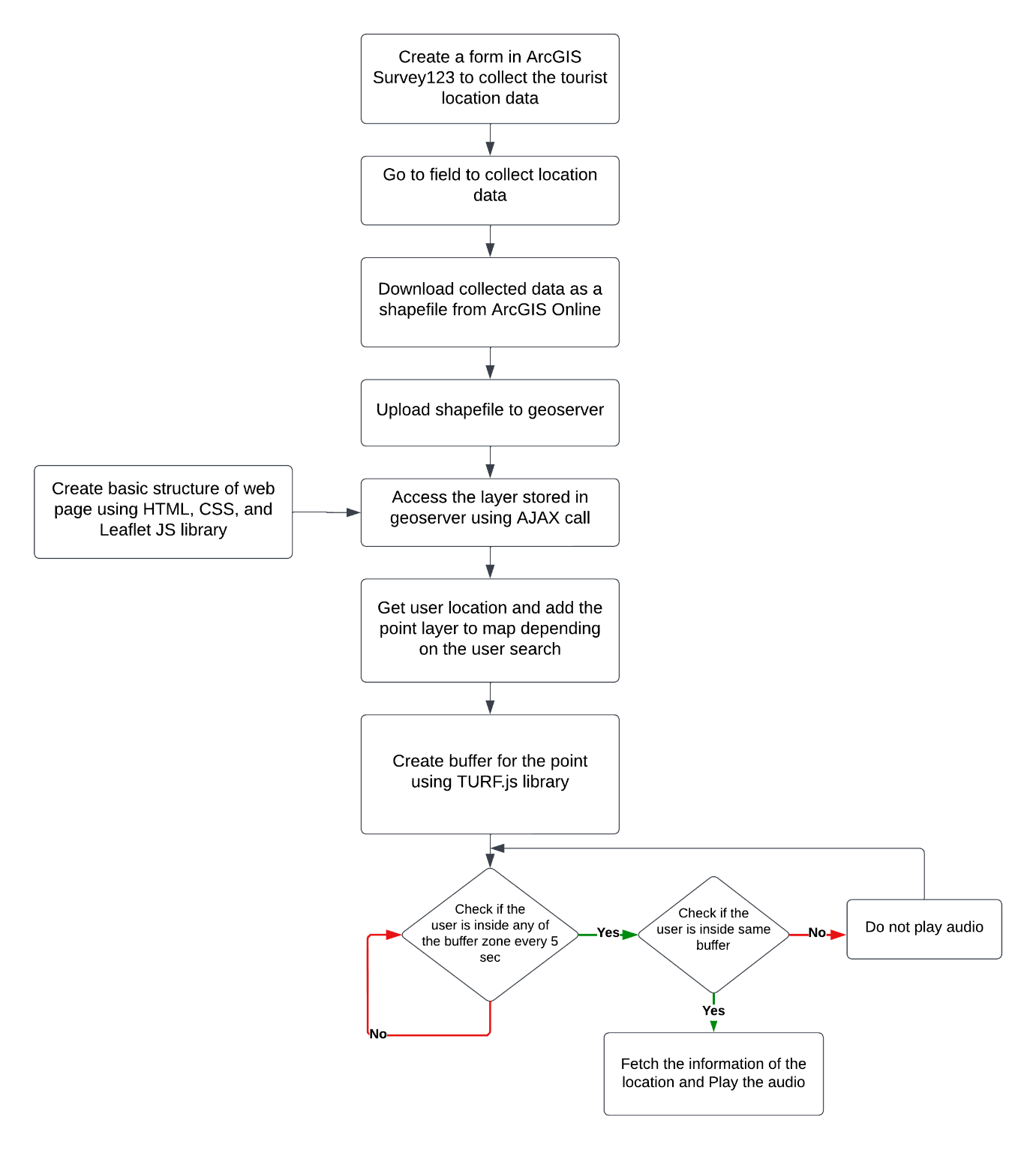
To tackle this issue geo-tagging of sound can be a unique solution. Around the area of interest, a geofence can be generated. Now this geofence can be used to trigger the audio. When the person enters in the vicinity of the region, a modulated voice starts narrating the information of the specified area. And when the person moves away from that buffered region, the voice stops. All of this can be managed through an app. The person requires to enable the location in the phone and the app will take their location and show the nearby point of interest.

## Geofence

Geofence is a virtual geographical area bounding a certain point in real world (Namiot & Sneps-Sneppe, 2013).They are non-changing entities which can be used around important buildings and places(Stevens & Atkins, 2021). It can be used to trigger a response whenever the user crosses the boundary. In the project, geofences is planned to be established around tourist locations, museums, parks, etc. When the user enters that zone, audio having the information about the region will be triggered. Geofences are the buffers along with an ability to trigger an event. The logic required to trigger an event is written using the turf.js library in JavaScript in the current project.

# Methodology

## Flowchart

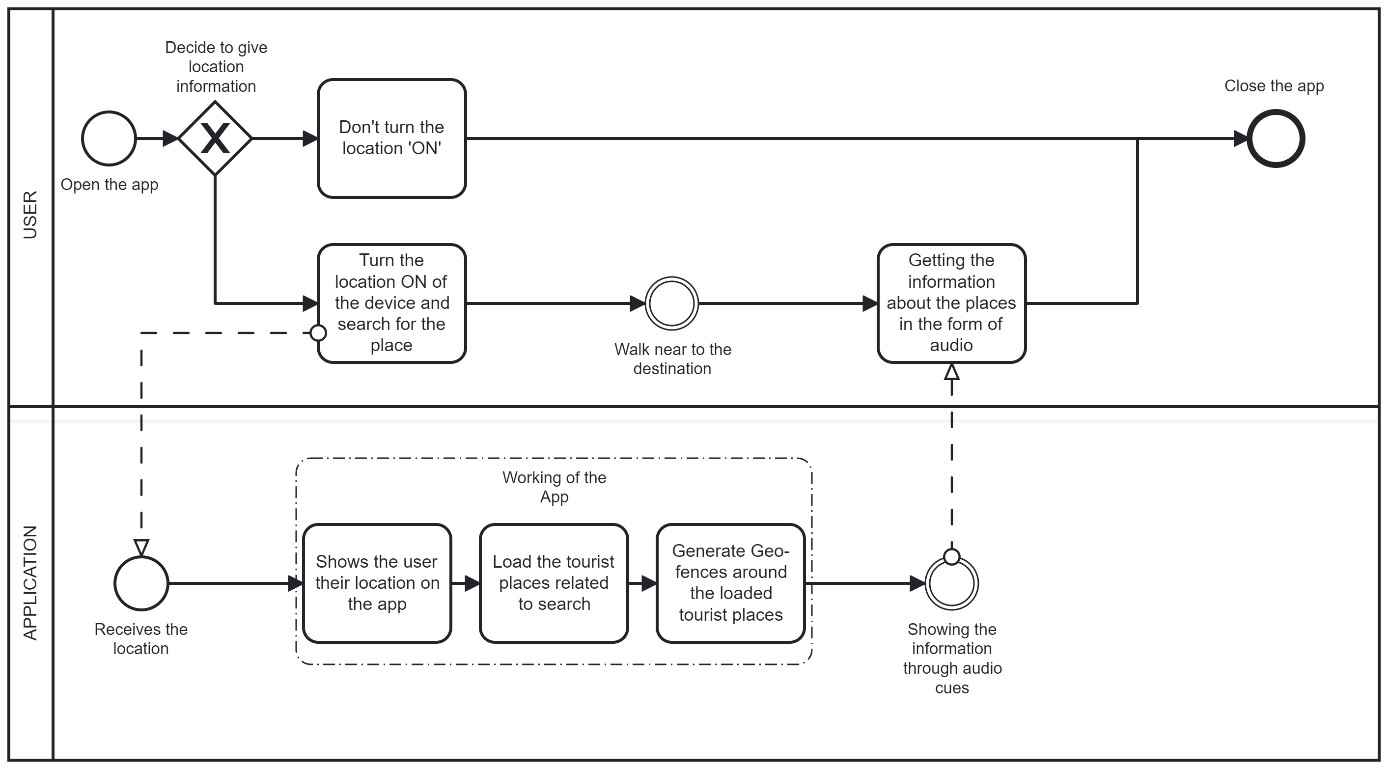


The first step was to collect the data from field. To do that, ArcGIS Survey123 was used. A form was created and then data was entered one point at a time. Once the data was captured, it was converted into the shapefile format and downloaded for further use. ArcGIS Online was used for this task. The downloaded shapefile was pushed onto the Geo Server to later provide it as WFS (Web Feature Services) layer. Parallelly, a basic web page was also created using HTML, CSS, and Leaflet JS library. On this web page, a request for the WFS layer in GeoJSON format is made to the geo server using AJAX call. After adding the WFS layer, the user’s location is acquired. And based on the user’s search, places are updated on the web page by removing the previous layers if any and adding the searched layers. A buffer is generated around the point of interest using TURF.js.

The upper tasks act as a prerequisite for the actual working of the web app. User’s location information gets updated every five seconds. Once the user enters a buffer zone, the audio is triggered. Since the user’s location is getting updated regularly, it checks if the user is still inside the same buffer or outside the buffer. If the user is inside the same buffer, the audio won’t repeat. Otherwise, it will stop. And if the user walks into some other buffer zone, the process gets repeated again.

Both the BPMN diagram and Activity diagram given below shows the working of the application in much more abstract way.

## BPMN Diagram



## Activity Diagram

# Results

Figure : WFS layer from Geo Server displayed on Leaflet map

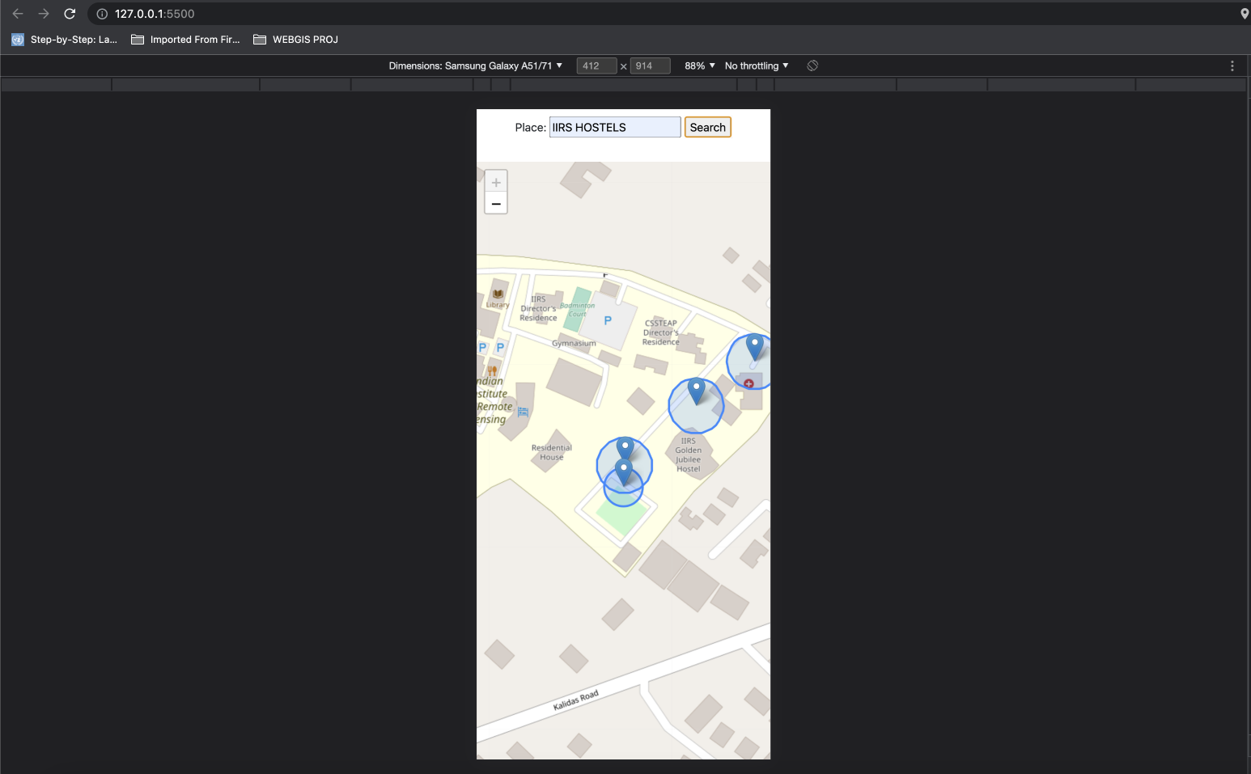
**

Figure : Locations along with geo fence loaded based on the user's search

A screen recording showing the working of the application is uploaded into the github along with the code which can be accessed from the below link:

<https://github.com/manojappalla/AudioTourGuide>

# Conclusions

Web technologies like leaflet and geo-server can be used to create applications for solving practical problems. JavaScript being a powerful scripting language allows for a standalone application. The working of the geo-tagged audio paves the path for more use cases. One such application can be used for the visually impaired people.

Currently, we are getting the location information from the users mobile. Better GPS devices can give us accurate results. Performance issues can be there in congested and indoor environment while using the mobile phones. In such cases, specialized hardware system can be used for accurate measurement.

Also the geoserver can be hosted in cloud technologies like AWS so that it will be possible to access it from anywhere. The webpage can also be hosted by storing the code in the tomcat webapps folder in the aws instance. This will help users to access it from anywhere.

# References

Namiot, D., & Sneps-Sneppe, M. (2013). Geofence and Network Proximity. In S. Balandin, S. Andreev, & Y. Koucheryavy (Eds.), *Internet of Things, Smart Spaces, and Next Generation Networking* (pp. 117–127). Berlin, Heidelberg: Springer Berlin Heidelberg.

Stevens, M., & Atkins, E. (2021). Geofence Definition and Deconfliction for UAS Traffic Management. *IEEE Transactions on Intelligent Transportation Systems*, *22*(9), 5880–5889. https://doi.org/10.1109/TITS.2020.3040595

# Appendix

## Code Snippet to access WFS Layer from Geoserver:

|  |
| --- |
| function loadLayer(){ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| place\_name = String(document.getElementById('place').value) |
|  |

|  |
| --- |
| let cqlf = `CQL\_FILTER=name\_of\_th='${place\_name}'` |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if(place\_name != null){ |
|  |

|  |
| --- |
| url = 'http://localhost:8080/geoserver/AudioTour/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=AudioTour%3Atourist\_locations&maxFeatures=50&outputFormat=application%2Fjson&' + cqlf; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| if (place\_name == ''){ |
|  |

|  |
| --- |
| url = 'http://localhost:8080/geoserver/AudioTour/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=AudioTour%3Atourist\_locations&maxFeatures=50&outputFormat=application%2Fjson'; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| $.ajax({ |
|  |

|  |
| --- |
| type: "GET", |
|  |

|  |
| --- |
| url: url, |
|  |

|  |
| --- |
| crossDomain: true, |
|  |

|  |
| --- |
| success: function(data) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| data\_gjson = data |
|  |

|  |
| --- |
| if(gjLayer && map.hasLayer(gjLayer)){ |
|  |

|  |
| --- |
| map.removeLayer(gjLayer); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| if(bufLayer && map.hasLayer(bufLayer)){ |
|  |

|  |
| --- |
| map.removeLayer(bufLayer); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| gjLayer = L.geoJSON(data).addTo(map); |
|  |

|  |
| --- |
| bufs = []; |
|  |

|  |
| --- |
| for(let i=0;i<data.features.length;i++){ |
|  |

|  |
| --- |
| bufs.push(turf.point(data.features[i].geometry.coordinates)); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| bufs = turf.featureCollection(bufs); |
|  |

|  |
| --- |
| bufs = turf.buffer(bufs, 0.05, {units: 'kilometers'}); |
|  |

|  |
| --- |
| bufLayer = L.geoJSON(bufs).addTo(map); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| }); |
|  |

}

## Code Snippet to execute the logic

|  |
| --- |
| currentLocation = turf.point([long, lat]) |
|  |

|  |
| --- |
| for (let i = 0; i < bufs.features.length; i++){ |
|  |

|  |
| --- |
| if(turf.booleanPointInPolygon(currentLocation, bufs.features[i])){ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| currentIndex = i |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (currentIndex != previousIndex){ |
|  |

|  |
| --- |
| for (let j = 0; j < data\_gjson.features.length; j++){ |
|  |

|  |
| --- |
| if (turf.booleanPointInPolygon(data\_gjson.features[j], bufs.features[i])){ |
|  |

|  |
| --- |
| // console.log(data\_gjson.features[j].properties['informatio']) |
|  |

|  |
| --- |
| textToSpeech(data\_gjson.features[j].properties['informatio']) |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| previousIndex = currentIndex |
|  |

|  |
| --- |
| console.log("your are inside") |
|  |

}

## Code Snippet to Convert text to speech

|  |
| --- |
| function textToSpeech(str){ |
|  |

|  |
| --- |
| let utterance = new SpeechSynthesisUtterance(str); |
|  |

|  |
| --- |
| speechSynthesis.speak(utterance) |
|  |

}