International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

Embedded System for Dynamic Location-Based Advertisement using Google Maps API

Taher Ajnawala, Ranganath Pujari, Aaqib Shaikh, Balu Jaybhaye

Trinity College of Engg. and Research Pune, Maharashtra, India
Under the guidance of
Dr. S.B. Chaudhari

Abstract - *Advertising plays a very important role in today's age of competition. Advertising is one thing which has become a necessity for everybody in today's day to day life, be it the producer, the traders, or the customer. The world still uses the age old and primitive techniques for advertisement. Our objective is to increase and maximize the efficiency of these already existing advertisement systems. There are various ways in which the advertisement system can be improved. One such way is to make the advertisement system vary and change according to the location of the vehicle. This ensures that we capture the target audience in a much creative and interactive way by providing them information and advertisements that are related to their current location whilst travelling. The target audience gets a much better and personalized viewing experience if he gets content that is relevant and related to his current near his present location. We intend to create such an advertisement system that will be dynamic and location based using a single board computer that is The Raspberry Pi 3 to keep the production cost to a minimum with the essential modules such as GPS, GSM-Support, Display-Support and optional Wi-Fi Integration. The location services can be made available to the proposed system using Open Source Google Maps API.*

Key Words: **Dynamic, Location-based, Google Map API, Embedded System, Raspberry Pi.**

1. INTRODUCTION

1.1 Project Idea

Right from the industrial revolution, advertising has played a major role in promoting private and public businesses. Advertising is a form of marketing communication that is used to make people aware of a certain product or entity. In 21st century, advertisement has become ubiquitous but with the same traditional rudimentary ways and techniques. Due to these age old techniques, maximum potential of advertisement is not being achieved.

This brought up the idea of creating a system that will be digital and dynamic in nature. Also the idea of making the advertisements more localized was brought up with the intention of providing the customers and users useful and relevant advertisement that is around their present location.

Our project is a part of the smart city concept meaning the advertisement techniques used are Digital and advanced. Thus, our project will overcome issues like wastage of Physical Space, Reusability and Maintenance.

1.2 Motivation of the Project

The motivation of our project is to:

- The advertisement system that is in place now is primal in nature and does not make efficient use of all the resources available.
- Existing advertisement techniques are quite outdated. They need to be upgraded with current new and advanced technologies to get maximum results and outcomes.
- The traditional advertisement techniques waste a lot of space and resources because they use old and inefficient methods.
- Due to not being able to make use of new and advanced technologies of the market, they fail to capture the attention and interest of new generation of customers.
- Reusability is a big issue of the old advertisement system. Once an advertisement system has been put up in place it is very difficult to replace them or use them for other purposes. This leads to lot of resource wastage and system failure.
- Since most of the existing advertisement systems are not digital in nature, it becomes very difficult to maintain and keep them intact. Also the climate and weather changes can cause severe damage to the non-digital and trivial advertisement systems. Using digital techniques and methods can help prevent the

accidents that may arise due to weather changes and wear and tear.

1.3 Major Constraints

As this system makes use of the Raspberry Pi 3 as the central controller and a Sim808 module it has some constraints which pose an issue. Being a mobile system it faces the usual constraints of any mobile system. They are:

- Low Processing Power: Although the Raspberry Pi is equipped with a 1.2 GHz 64 bit quad core processor. It still has severe bottleneck while executing day to day tasks. On paper the processor my seem powerful, but in reality is it not able to function as per expectations. Boot up takes around 20 seconds, that being fast, for performing any normal operation you still have to wait for around 5 seconds before the system responds. This is noticeable during the first bootup of the system and the launching of our application. This constraint weighed in when we tried to play videos on the Chromium Browser installed natively on the RPi. Even basic HD video _le playback was significantly choppy on the browsers native HTML5 player.
- Low System Memory: This is one of the biggest constraints of the Raspberry Pi. It only has 1GB of system memory. Initially the proposed system was supposed to run on the RPis native web browser. After playing around with it for a while, we came to a conclusion that the system just wasn't capable of playing HD videos from the browsers HTML5 video player. As this is an Advertisement System which plays advertisements one after another without any delay, the RPi couldn't keep up with the changing video les. The browsers allocated page memory would run out far before it could play all the videos. This effectively crashed the browser and it would outright refuse to play any more videos.
- Low Storage Space: The Raspberry Pi is running on a 16GB ash storage device, out ofwhich only 15GB is usable. This limits the capabilities of the system to a great extent for future use. Currently the system OS which is Raspbian Jessie with Pixel requires 4GB of disk space alone and another 1GB of disk space is allocated to Page Memory. The support libraries for the project leave a footprint of about 200MB. This leaves us with the space of approximately 9GB. Assuming each video _le to be of 10MB, and their shorter version being 2MB we can store about 700 advertisements before running out of disk space. Alongside this, we face a risk of facing major system performance issues due to low disk space. Although, deploying in small region would pose no threat for this system, but at a broader sense it is problematic.

2. APPLICATIONS

- Advertisements in:
  1. Public Transports like:
     -Local Bus
     -Trains
  2. Private transports like:
     -Auto Rickshaws
     -Car pooling
     -Cab services

3. IMPLEMENTATION

3.1 Advertisement System

The Advertisement Web Application consists of the following files that are required for managing and running of the system.

- Data.json
- Properties.json
- RadialMapping.py
- GeoLocation.py
- Schedule.py

3.1.1. Data.json

As mentioned earlier, the system uses json files as a method to store and parse data from one module to another. In the data.json file, the entire information about the clients or customers details along with the associated advertisement data that is stored in the system. The data.json file contains two different arrays in which the data storage is split up into.

- Advert
- Default list

3.1.2. Properties.json

The properties.json contains the essential properties on which various parameters of the system are defined. These properties can be manually viewed and changed as per the

requirement and need of the administrator. These properties afect the working of the advertisement system and cause the system to work as per our satisfaction.

The properties.json contains three properties that are very important:

- Radius
- Advertisement Slot
- Time Slot

3.1.3. RadialMapping.py

The radialmapping.py file is important for checking if the advertisement slots of a particular are filled with advertisements or not. As mentioned above in the properties.json file, each particular area has a maximum number of advertisements slots. This determines the maximum number of advertisements That can be allocated and played in that particular area. The radialmappig.py is used to when the administrator wants to add an advertisement in the system. When the administrator enters a particular location, then the radialmapping.py is responsible for checking if there are advertisement slots available or not. If there are slots available then this module adds the advertisement to the database and the system thus making the system dynamic and exible. The radialmapping checks each and every advertisement if it lies in the display radius of the system. If a particular advertisement lies in the current display radius, then it checks the advertisement slot property of the properties.json and if the current number of advertisement slots is less than the maximum number advertisement slots then it will add the advertisement to the system. With the use of this module, the system is able to manage and control the number of advertisements which can cause any unexpected working of the system or even its failure. This module makes sure that the system in not overloaded with advertisements that the system will be unable to play and informs the administrator if the system is already full.

RadialMapping.py requires both the json _les that are stored in the system. RadialMapping uses the radius property from the property.json to create the virtual circular area to make the comparisons with each advertisement and map them to the final advertlist. It also uses the latitude, longitude and advertlist from the data.json to carry out its functionality. It uses the latitude and longitude to calculate the distance of each of the advertisements from the current location of vehicle. If any of the advertisements do fall inside the radius then it appends them to the final advertlist that needs to be displayed by the system.

3.1.4. GeoLocation.py

The geolocation.py class is responsible for accessing the current location of the system. This is achieved by creating a UART Connection to the SIM808 module which is connected to the RPi using Tx and Rx serial headers. The library required for enabling this communication is called PySerial. The current location is obtained by sending two serial AT commands in succession as follows:

- **AT+CGPSPWR** : Used to power on the GPS module and initialize it to make connections with the geolocation satellites
- **AT+CGPSINF** : This AT command is used to request the SIM Module to provide with the necessary geolocation details. These details include:
  - Latitude
  - Longitude
  - Altitude
  - Course on Ground (direction)
  - No. of satellites connected
  - Speed

3.1.5. Schedule.py

This class is the core of the Advertisement System. It is responsible for Eficiently scheduling each and every advertisement such that, the most relevant advertisements are displayed first and also that no advertisement is left out from being displayed at least once. The schedule class consists of 3 methods viz:

- ScheduleLong
- ScheduleShort
- ScheduleDefault

3.2 Administration Web Application

The web application is the front end of the system that the administrator can view and control over. Through Java, JavaScript, HTML, CSS, Ajax and other technologies, application specific methods such as drawing on the screen, playing audio, and access to the keyboard and mouse are all possible. The current system makes use of the above mentioned technologies to make a web application that is quick, responsive and lightweight so that the UI is interactive, attractive and efficient. Many services have worked to combine all of these into a more familiar interface that adopts the appearance of an operating system. General purpose techniques such as drag and drop are also supported by these technologies. The system makes use of user

defined CSS and HTML techniques to create an interactive experience that does not require page reloading. We have used Ajax, a web development technique using a combination of various technologies, is an example of technology which creates a more interactive experience. In this system, Ajax is used to change the webpage contents dynamically and without having to reload the page.

Index.html page is core front end of the system. This html page creates, manages and controls the entire database as well as the system properties. This is the front end module of the system that allows the administrator to make changes into the system without having to deal with the programming aspect. The index.html is split into various sections depending on the working and functionality:

3.2.1 Upload section

- Adding advertisements
- Adding Default Advertisements
- Checking for available slots in particular selected location
- Error handling And Management

3.2.2 Display section

- Name of the company/client
- Priority of the advertisement
- Name of advertisement file stored in system
- Name of High priority advertisement file stored in system (If any)
- The location where the client wants his advertisement to be displayed

3.2.3 Manage System section

The manage system section of the administration web application does exactly what the name suggest. The manage section is divided into two parts. One part allows the admin to delete the advertisements that are stored in the database. The deletion process allows the admin to remove any data that in no longer required. The other part is where the system properties are displayed and defined. Properties: The working of the system is dependent on the properties that are defined in the second part of the section. This part first displays the current system properties which informs the user about the current working of the system. Below the display part of this Section all the properties; the time constraints, the advertisements and the radius of collecting advertisements are all defined. The manage section helps the administrator to change the working of the system based on his requirements and needs. This manage section helps us the system to be exible and modular in nature. The administrator does not need to go into the code of the system to change the working. The system becomes very user friendly and efficient when the system control is in the administrator's hand.

The properties that can be displayed and changed are:

- Current TimeSlot
- Current Radius (kms)
- Slots per Area

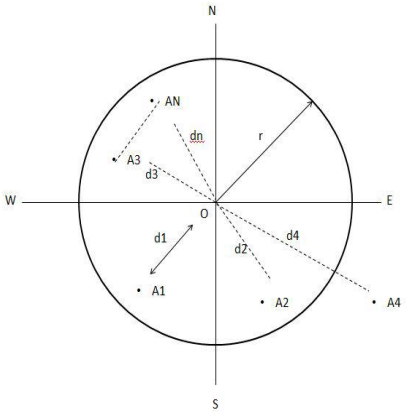
3.2.4 Deletion

The manage system also helps the admin to delete previous advertisements if they are no longer required. The system allows us to delete a single advertisement, all advertisements or all default advertisements if they want to be removed from the system in order to accommodate other needs.

3.3 Algorithm Details

Implementation and architecture is described in this section. It consists of two algorithms mainly. One is to schedule the advertisements according to the parameters and displaying them on the system mentioned and other algorithm is to check if the advertisements slots are full or not.

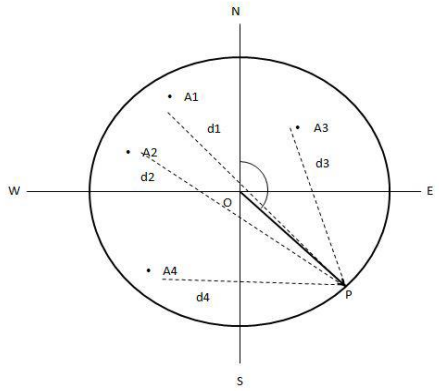
a) Radial Mapping While Uploading Advertisements:



- Step 1: Enter details of the advertisement >Name >Priority >Location
- Step 2: If priority is high then upload 2 files
- Step 3: else upload only 1 _le
- Step 4: Geo-coding process of the location specified
- Step 5: Check availability of slots[d(A,O)<=R]

- Step 6: If available allow upload of advertisement to system
- Step 7: else ask for another location
- Step 5: Stop

b) Schedule Advertisements:



- To schedule the advertisements eficiently, we need not consider only the current location of the vehicle 'O' but also the future location i.e the direction the vehicle is moving toward.
- P is the point at which the vehicle will reach after 't' amount of time. The geo-location of 'O' can be calculated by using the SIM808 module. But we need to calculate the geo-location of point 'P'.
- This can be done with the help of 'theta' i.e the direction in which the direction is headed in.
- 'P' can be calculated such as:
  Px = [Ox + (rcos(Theta))]

  Py = [Oy + (rsin(Theta))]

where, Px and Py = the co-ordinates of Point 'P'. Ox and Oy = the coordinates of Point 'O'(current location of vehicle).

Theta = course over ground i.e direction or angle with respect to NORTH direction. r = radius for mapping.

After calculating the coordinates of 'P' we need to calculate the distance from each and every advertisement in the radius with P. This is done with the help of Euclidean Distance Formula:

$d(P,An)= \sqrt{ (Px - Ax)^2 + (Py - Ay)^2 }$

Thus the advertisements are arranged in a decreasing order of distance. This ensures that only the relevant advertisements are displayed first and the remaining ones have a chance to be played in the next cycle of advertisement display.

3. CONCLUSIONS

By this proposed project we observe that the existing means of advertisement are inefficient and inadequate, thus they need to be replaced advanced and modern techniques.

Hence, we infer that the future for advertisement can be improved vastly by making use of:

- Dynamic digital systems
- Location-based advertisement
- Personalized experience

REFERENCES

[1] Pankaj Virulkar, "Location based advertisement publishing by using Wi-Fi and QR codes", IEEE, 2015.

[2] Md. RashidujjamanRifat, "A Location Based Advertisement scheme using OpenStreetMap", IEEE, 2012.

[3] Hiroaki Higaki, "Angle-Based Location Advertisement for Location-Based Ad-Hoc Routing Protocols", IEEE, 2012.

[4] O. Rashid, "Implementing location based information/advertising for existing mobile phone users in indoor/urban environments", IEEE, 2015.

[5] Rasberry Pi Official Webpage, http://www.rasberrypi.org/

[6] Raspberry Pi Libraries and Documentation, http://www.rasberry.org/downloadresources/

[7] SIM808 GSM/GPRS/GPS module, https://www.adafruit.com/product/2637