Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 3, Issue 5, May-2016

CROWD SOURCED DATA COLLECTION AND ANALYSIS

Akshay Thorat, Manish Raj, Rohit Yadav, Tauseef Khan.

Dept. of Computer Science Engineering, Savitribai Phule Pune University.
G.H.R.C.E.M, Wagholi, Pune, Maharashtra, India.

***Abstract* — Today's Smartphones can unlock the full potential of crowd sourcing and take eParticipation to a new level. Users are allowed to clearly contribute to difficult and original problem solving. Engagement of citizens is still challenging but the proliferation of smartphones with geolocation has made it easier than before. The paper introduces the environmental project called CROWDPATROL. This project is primarily intended to fight illegal waste dumps in the Pune District. The idea is to use the possible consciousness of the society about the environmental and economic drawbacks of illegal landfills. The Smartphones/tablets GIS (geographic information system) reporting application CROWDPATROL has been developed. This free mobile application enables to report illegal dumpsites and potholes. The project CROWDPATROL is intended for all those who don't like illegal sump sites in their cities or villages especially they want to do something against the bad attitude and behavior of irresponsible people or businessmen. The objective of this project is to contribute to solving problem of environmental pollution by illegal dumps in Pune and contribute to "SWACH BHARAT ABHIYAN".**

Keywords —GPS, GIS, Crowd Sourcing, Swach Bharat Abhiyan.

I. INTRODUCTION

Collection of massive amounts of data is one of the most labor intensive tasks in any industry. Various government bodies associated with civic duties such as transport department, municipal corporations are often burdened with the task for surveying before mitigating issues. The system proposed in the paper exploits crowd sourced architecture to collect data and report issues to various civic bodies in real time. Citizens are provided with a platform where they can report various issues using the app. The app runs on a GPS enabled smart phone and uses goe-location to find user's location, camera, Sensors to capture images of illegal dump sites and various other issues, which can then be reported to a central cloud infrastructure. The app runs on Android platform with API level 15 and above which targets more than 88% of available smart phones running Android.

II. PROPOSED SYSTEM

The proposed system enables users to report any issues in their locality by capturing images and uploading it to the system. The images contain geo-tagged information, essentially the GPS coordinates embedded in EXIF meta-data within image files which will be used to pinpoint the reported issue on an interactive map. The system consists of both web and android platform which can be used to view and report issues respectively. The proposed system is designed to be modular and draws inspiration from various existing systems. It extends several features of existing systems to insure a user-friendly experience built on top of widely available and existing technologies to incur no extra costs.

The proposed system consists of various modular components such as a user system to enable sign-up, sign-in and maintain minimal profile information of users. Users can view all the data collected over a period of time and also track the status of their reported instances in real time. An interactive map is published where one can view reports in real-time with dynamic report visibility based on social engagement, time of report and trust factor of reporter. Most of the data is stored in a NoSQL data store such as MongoDB which coupled with a NodeJS API server, forms the core of proposed system.

The system also contain an administration module for moderation of new reports, banning abusive users, generating reports etc. Overtime large amounts of data (reports) are collected and analysed; then visualized for easier consumption and published to the web where anyone can access it. The system also aims to integrate different social features such as – upvote, share etc to promote participation and increase visibility of reports.

Table 1 : Database Models

Database Model	Description
User	Minimal user profile information, access token, password hash
Marker	Latitude, longitude, timestamp, reporter uid, upvotes, flags, dynamic visibility score
MediaFile	Image path, image resolution, EXIF data, associated report id

Database models are normalized to encourage logically grouping, reduce data redundancy, ensure atomicity and integrity of transactions.

III. DESIGN & IMPLEMENTATION

Here is the system architecture which shows how the system will work.

- There will be 2 types of user
  -Authorized user
  -Anonymous user
- The user can login using his/her facebook/google+ account or can create a new user account.
  The authorized user will have the privilege of viewing, uploading/reporting an issue whereas the anonymous user will have only view privilege.

3.1 The system architecture is mainly consist of 4 modules:

1. User Module.
2. List Module.
3. Upload Module.
4. Visualization Module.

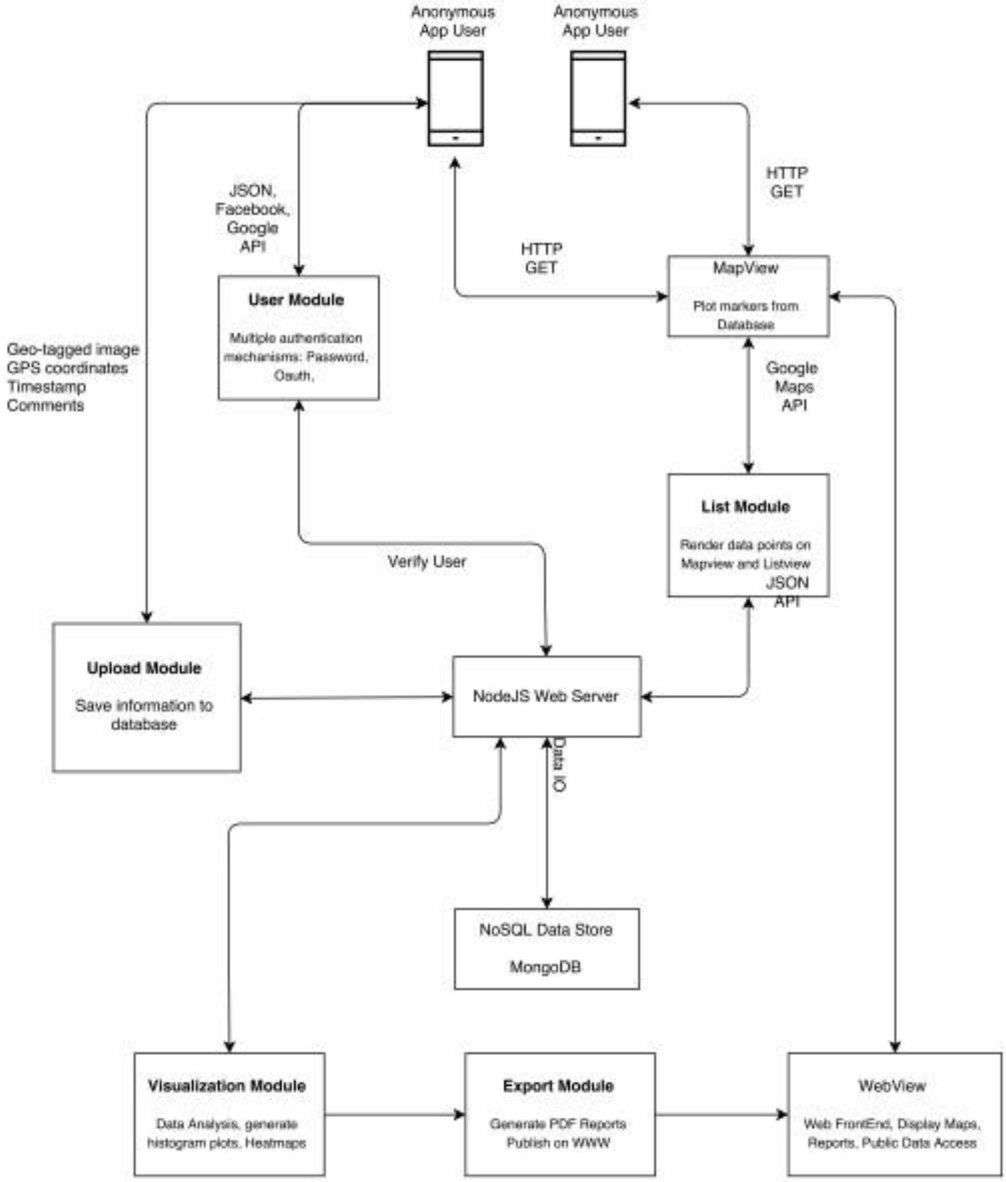
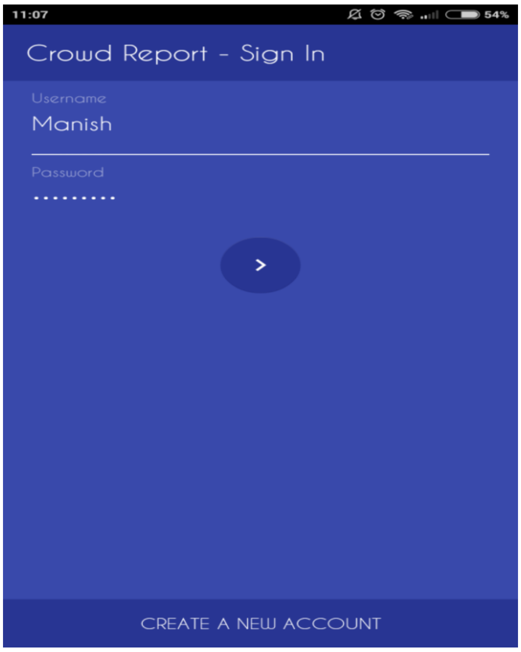

Anonymous App User
Anonymous App User
JSON, Facebook, Google API
HTTP GET
HTTP GET
Geo-tagged image
GPS coordinates
Timestamp
Comments
User Module
Multiple authentication mechanisms: Password, Oauth,
MapView
Plot markers from Database
Google Maps API
List Module
Render data points on Mapview and Listview
JSON API
Verify User
Upload Module
Save information to database
NodeJS Web Server
Data IO
NoSQL Data Store
MongoDB
Visualization Module
Data Analysis, generate histogram plots, Heatmaps
Export Module
Generate PDF Reports Publish on WWW
WebView
Web FrontEnd, Display Maps, Reports, Public Data Access

Figure 1. **Architecture of the system**

3.1.1 User Module:



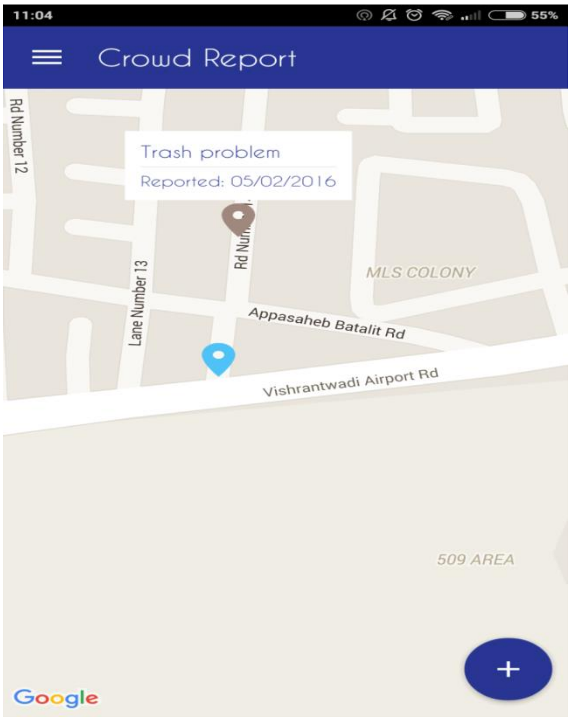
Username
Manish

Password
·········

This module handles all the user registration and login. It also provides the API for Login like

- Google
- Facebook
- Email/password

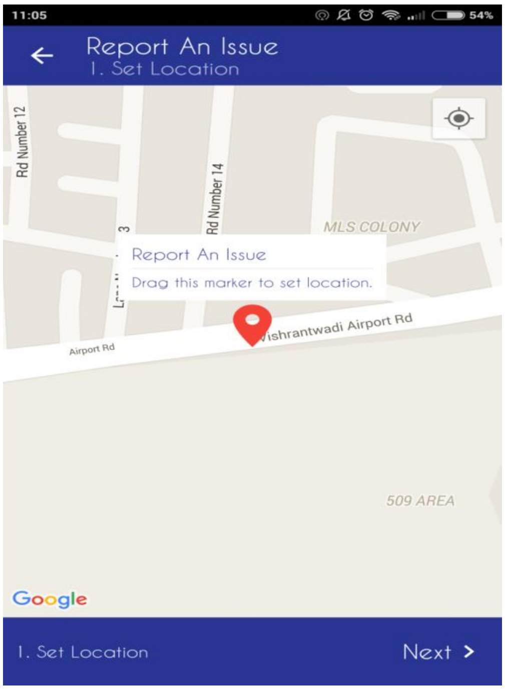
3.1.2 List Module:



Trash problem
Reported: 05/02/2016

After the user has successfully login, the List Module displays the various issues reported by users. The users have the filters to select from different instances like pot holes, illegal dumps, etc.

3.1.3 Upload Module:



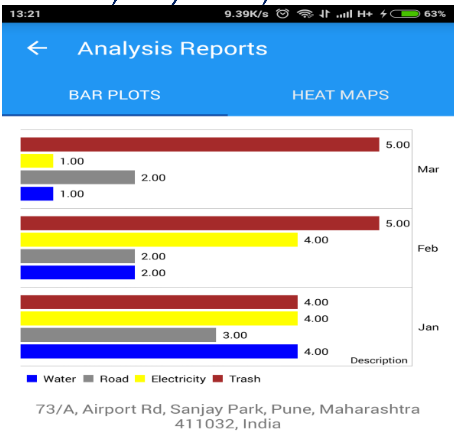
Report An Issue
Drag this marker to set location.

The Upload Module handles all the data collection part. Extract the GPS co-ordinates, uploads the images uploaded by the users on the server. It also scales and compresses the images to save bandwidth and saves data to mongodb database.

3.1.4 Visualization Module:

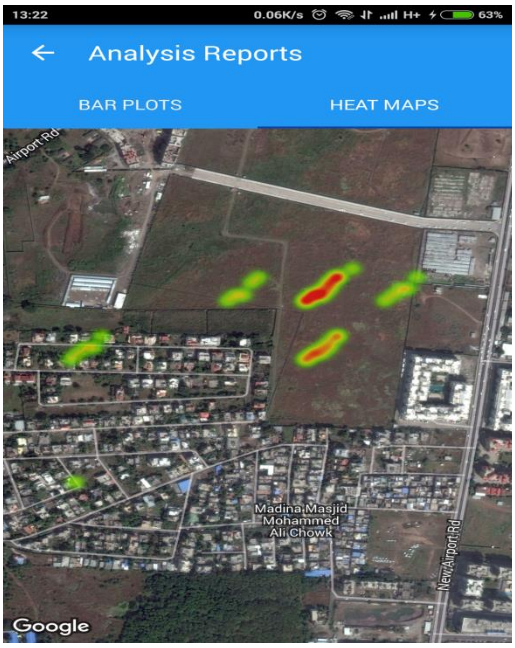
In this module, various algorithms are applied and Bar plots and Heat maps are generated according to the data present in the system. This data is nothing but the issues which are reported by different or same users.

3.1.4.1 Bar-Plots:



Bar plots are plotted on the basis of frequency count of the reported issues. This data is very clear to user to understand which area is more polluted or has maximum number of illegal dump sites.

2.1.4.2 Heat-Maps:



Here the issues reported by the users which are very severe and requires instant action are marked Red.

III. ALGORITHM:

The following algorithm, inspired by Reddit listing algorithm[7], is used to compute visibility of a report in order to determine the position and order of reports as they appear on the interactive map.[7]:

```
epoch := timestamp('1/1/1970')

function diff(date) {
  d = date - epoch
  return d.days * 86400 + d.seconds + d.microseconds)/1000000
}

// Compute visibility of a report
// up : number of upvotes to a reported
// flag : number of times the report has been flagged
// date : date-time when report was submitted
// return : a visibility score of reported
```

```
function visibility(up, flag, date) {
  score = up - flag
  order = log10(max(abs(score), 1))
  if ( s > 0 ) {
    sign = 1
  } else if ( s < 0 ) {
    sign = -1
  } else {
    sign = 0;
  }
  secs = diff(date) - 1134028003
  return round(sign * order + secs / 45000, 7)
}
```

Algorithm for kernel generation (for weighted data points) when generating heatmap overlay on top of Google maps.

```
function generateKernel[] ( radius, sd) {

    kernel[] = {radius * 2 + 1}
    for (i = -radius i <= radius ++i) {
        kernel[i + radius] = e ^ ((-i * i) / (2.0 * sd * sd))
    }
    return kernel
}

function convolve[][] ( grid[][], kernel[][] ) {

    radius = floor( kernel.length / 2 )
    dimOld = grid.length
    dim = dimOld - 2 * radius
    lowerLimit = radius
    upperLimit = radius + dim - 1
    intermediate[][] = {{dimOld, dimOld}}
    x = 0, y = 0, initial = 0, val = 0

    for (x = 0 x < dimOld ++x) {
        for (y = 0 y < dimOld ++y) {
            val = grid[x][y]
            if not val  == 0.0 {
                xUpperLimit = (upperLimit < x + radius?upperLimit:x + radius) + 1
                initial = lowerLimit > x - radius?lowerLimit:x - radius
                for (x2 = initial x2 < xUpperLimit ++x2) {
                    intermediate[x2][y] += val * kernel[x2 - (x – radius)]
                }
            }
```

```
        }
    }

    outputGrid[][] = {{dim, dim}}

    for(x = lowerLimit x < upperLimit + 1 ++x) {
        for(y = 0 y < dimOld ++y) {
            val = intermediate[x][y]
            if not val == 0.0  {
                yUpperLimit = (upperLimit < y + radius?upperLimit:y + radius) + 1
                initial = lowerLimit > y - radius?lowerLimit:y - radius
                for (y2 = initial y2 < yUpperLimit ++y2) {
                    outputGrid[x - radius][y2 - radius] += val * kernel[y2 - (y – radius)]
                }
            }
        }
    }
    return outputGrid
```

V. CONCLUSION AND FUTURE SCOPE

In conclusion, the proposed system turns out to be more cost effective by utilizing the power of crowd to report different social issues while making use of existing technology ecosystem. From an administrative point of view, a widely available technology stack and seemingly easier implementation and deployment means most civic bodies can seamlessly adopt, deploy and maintain the system.

During testing of initial prototype, it was observed that the simple design flows and user-friendliness of the app resulted in quicker and wider adoption by both technically adept users and laymen.

The facts from statista.com as showed in figure 3 shows sufficient market penetration by Android smartphones at about 45% of total mobile operating systems which further strengthens our conclusion of wider adoption of the platform.

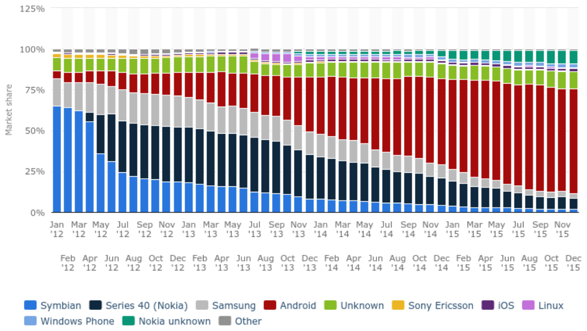


Figure2: Android market share 2015

The transparent nature of the system, lacking in existing solutions, helps reduce administrative costs and also instils trust in users.

Furthermore, the system can be extended to include a range of other issues that require crowd participation such as reporting instances of disease, accident-prone areas, civic issues, tagging stray animals et cetera.

VI. ACKNOWLEDGEMENTS

This work is affiliated by the Department of Computer Engineering, G. H. Raisoni College of Engineering and Management (GHRCEM) – Pune, Maharashtra. India. We would like to thank our project guide Mrs. Seema Udgirkar, HOD Mrs. Poonam Gupta and all faculty of Computer department, GHRCEM for their valuable guidance and help.

VII. REFERENCES

[1] Kubásek, M. (2013). ―Mapping of Illegal Dumps in the Czech Republic-Using a Crowd-Sourcing Approach.‖, vol., no., 30 May. 2013

[2] Sen, R., "RasteyRishtey: A social incentive system to crowdsource road traffic information in developing regions," in Mobile Computing and Ubiquitous Networking (ICMU), 2014 Seventh International Conference on , vol., no., pp.171-176, 6-8 Jan. 2014

[3] Heipke, Christian. "Crowdsourcing geospatial data." ISPRS Journal of Photogrammetry and Remote Sensing 65.6 (2010): 550-557

[4] Suri, Manik V. "From Crowdsourcing Potholes to Community Policing: Applying Interoperability Theory to Analyze the Expansion of ―Open311." Berkman Center for Internet & Society at Harvard University (2013).

[5] Google Maps API intro http://developers.google.com/maps/documentation/android-api/intro

[6] J. Prasanna Kumar "Inferring Location from Geotagged Photos" - IJARCSSE, ISSN: 2277 128X - Vol 4, 9 Sept. 2014.

[7] https://medium.com/hacking-and-gonzo/how-reddit-ranking-algorithms-work-ef111e33d0d9