

# **Mobile Development of Philippine Climate Outlook Through Data Driven Documents and Web-View Applications**

## **Abstract**

Knowing the amount of rainfall in a region is a necessity for people to know if they need to plant crops or to evacuate areas based in the heaviness of precipitation. The purpose of this paper is to provide a solution to easy access of the data that is already provided by government agencies. The goal of this paper is to discuss how the development and the use of a mobile app designed to carry the information from a website on SARAI. The technology used in this application is PhoneGap, which is a mobile web-view application that allows the programmer to use familiar scripts in coding within mobile applications and allows for a high percentage of code reuse when coding for both web and mobile applications. In joint with this technology, Data Driven Documents or D3 is used to display the maps dynamically with the data supplied and less processing time than what GIS provides.

## **Introduction**

The vast land wealth and the tropical climate of the Philippines mean that our country has a lot of land for farming but is also very prone to natural disasters like typhoons. Many citizens have the need to know forecasts about factors of climate including average temperature and amount of rainfall in an area for their own safety and use. These data can also be used by farmers so they would be able to ascertain the type of crop they would need to plant in a certain timeframe or when to plant a certain type of crop. This leads to the need to have easy access to the information provided. With the spread of phones built from open source casual desktop users have migrated to using their portable handheld equivalent: smartphones.

Several countries have websites that show climate outlook in varying degrees and varying ways. For example, USA's National Weather Service, Japan's Meteorological Agency and Russia's Hydrometcenter has a page on their website dedicated to displaying this information. The Philippines' PAG-ASA has a website that show these data in the form of colored maps but these are not easily accessible when not connected to the internet. The mobile application will be largely offline, and will only need to connect to the internet every time information is updated.

## **I. Review of Related Literature**

Visualization of Data is a great way for people to be able to know how their data works without having to do much of an effort regarding it. Rows upon rows and columns upon columns of data and information can be stored easily in a spreadsheet but these data aren't easy to understand without the help of some tools or a lot of skill with regards to the given information. For example, when you see the rise and fall of the values in the stock market, for example, numbers could be dizzying in large amounts, and people have long since proven to understand and retain illustrations and visual cues better than other types of sensory input. Current technologies used in the display of location-oriented datasets, for example rainfall forecasts, are of the following: a) using

Geographical Information Systems (GIS) available to the public, for example Google Maps or Apple Maps. However, previous systems using GIS for these factors are limited due to the processing speed and the amount of data that GIS uses to display what is needed, therefore, a more basic visualization is needed for a more convenient use of the provided data.

Geographical Information Systems work in a variety of levels, and is used to power many different decisions that might affect a lot of people. GIS, put into simple terms, does four major things: 1) it creates geographical data, 2) manages geographical data, 3) analyzes these data and 4) displays it in the form of a map. Several different types of people contribute to make a GIS work, and that includes cartographers, land surveyors, analysts, programmers. Although GIS are incredibly useful for making these decisions, many GIS are already loaded with a lot of information about a lot of things across the globe and not specifically focused locally. This has good effects and bad effects to the visualization of data. A large quantity of data means that the length of time needed for the information to be loaded is longer but in this certain scenario where we only focus on the forecast per municipality per province in the Philippines, such amount of detail is not needed.

This is where SVG, D3JS and GeoJSON come in. D3JS stands for Data Driven Documents JavaScript. D3JS is a JavaScript Framework that was made for the display of and/or visualization of objects given a certain amount of data. Given that it is an open source JavaScript framework, simply sourcing the D3JS file would be all the preparation needed for your coding environment.

JSON behave similarly to objects in javascript and GeoJSON files are a subset of JSON. GeoJSON are relatively small files which have details about certain regions or provinces that would be useful in illustrating these said regions. There are different open source GeoJSON files in GitHub and other project platforms and could be used in a project without further need to use other resources. GeoJSON files have ‘features’ and some of these features may include the name of the certain region or the country or the geographic parent of the certain region but the most important part of the GeoJSON files is its *geometry*. The geometry of an object in a GeoJSON file has two important properties: 1) the type and 2) the coordinates. The type of the GeoJSON object is a structure that lets you know how the coordinates would be laid out or how much information you would be able to put in the coordinates. For example, a Point would only need two coordinates in the form of a Point while a LineString would need you to have an array of points. What we use in mapping geographical borders are the type MultiPolygon. A MultiPolygon is an array of array of array of Points. These points construct the region according to the coordinates of the widely-used Geographic Coordinate System (latitude and longitude), also called the *e* that you see in regular maps. The Philippines for example sits ~13 degrees North of the Equator and ~122 degrees East of the Prime Meridian.

**Figure 0.** An Example of a GeoJSON object, type: Point

```
{  
  "type": "Feature",  
  "geometry": {
```

```

    "type": "Point",
    "coordinates": [121.7, 12.8]
  },
  "properties": {
    "name": "Philippines"
  }
}

```

Angular JS is a robust JavaScript framework that works well with creating single-page applications due to its two-way binding of variables wherein changing that values on the HTML main file will reflect on the JavaScript controller file without the use of additional commands. Binding an html element to a variable inside the controller would guarantee that there would be no misplacement of their values so long as proper coding practices are used. Angular JS has been used in many different applications that we use daily. Good Examples of these are PayPal, Netflix and Youtube(for PS3). This is used so that the programmer no longer has to use jquery for each and every action regarding the display of variables in the HTML file.

SARAI (<http://www.sarai.ph>) already hosts a web application for Rainfall Outlook for the next six months and the purpose of this project is to provide a mobile app that has the functionalities of the website.

## II. Methodology

The technology used in this application is PhoneGap, which is a mobile web-view application that allows the programmer to use familiar scripts in coding within mobile applications and allows for a high percentage of code reuse when coding for both web and mobile applications.

The user interface is made by the HTML skeleton and local CSS and the system functionalities use JavaScript. The initial data is loaded via a csv file. The format of the csv file is as follows: 1) The first row is the headers where the title of the first column doesn't matter but the headers of the values of each month must be in *text* format (can be achieved by pressing *ctrl+I* for the selected cells using Excel as Excel automatically converts any type of date inputted into cells into a date format), and preferably contain both the month and the year of the values for each column; 2) the contents of each of the rows must start with the name of the province followed by each of its municipalities.

The data from the csv file is parsed using a preset list of the names of all the provinces of the Philippines and is stored in an array of objects which we will call *provinceArray* in this document as shown in Figure 1 where "Array(6)" contains the values of each month for each municipality/province given by the csv file. Headers of the csv file are put aside to a smaller array to avoid repetition in the already large object provided by the parsing.

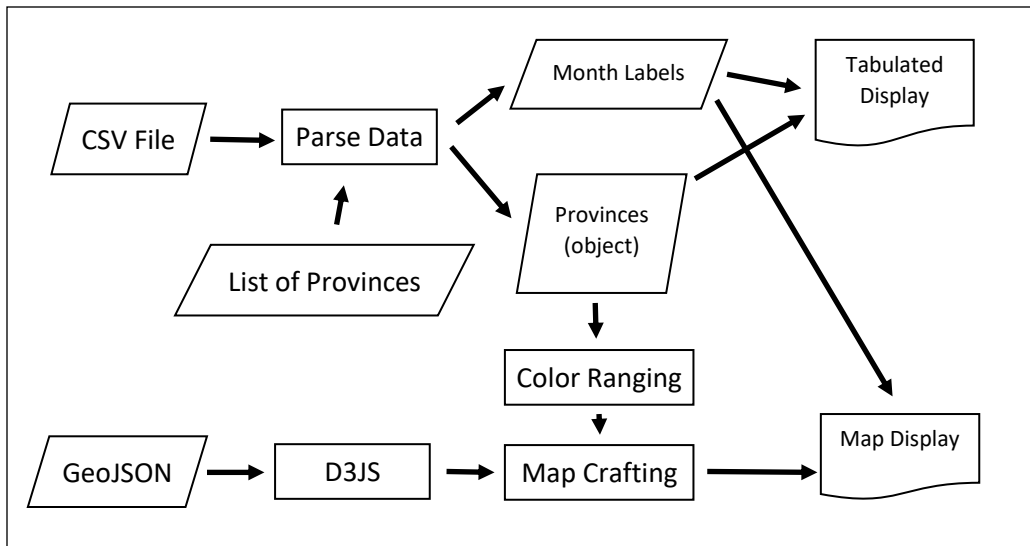
**Figure 1.** Object storage of parsed data from csv file.

```
[
  {
    id: 0,
    province: "Abra",
    array: Array(6),
    municipalities: [
      {id: 0, Municipality: "Bangued", array: Array(6)},
      {id: 1, Municipality: "Boliney", array: Array(6)},
      ...,
      {id: 26, Municipality: "Villaviciosa", array: Array(6)}
    ]
  },
  ...,
]
```

After the files are parsed, the maps are created using D3JS and GeoJSON files. The GeoJSON file for the Philippine provinces is downloaded from GitHub and is used as the path attribute of the <svg> tag. The colors of each province are taken from the *provinceArray* object presented in Figure 1 with the corresponding nth map to the nth element of the array for each province. The color scheme is selected by the user upon the first startup of the app, and is given four ranges according to number of millimeters of rainfall in that region. The ranging is applied to the corresponding value and then is assigned as the color of that province of that month. Six maps are crafted by D3JS in SVG and the headers of the previous columns of the csv file are put as a label of each map. Consequently, the initial display of the app selects the first province in the list to show the summary of the rainfall for that province for the next six months by selecting the 0<sup>th</sup> element of the *provinceArray* object and rounding the two-decimal number to a round natural number. Each display box shows that month/year, the rain in mm and is colored according to the selected scheme.

Two-way binding is used to display the tabulated form of the data with their respective coloring based on ranges.

**Figure 2.** Flowchart Summary of Methodology



### III. Results and Discussion

The app starts up with none of its settings and data setup. The processing is front-loaded, so the app, once it's fully loaded no longer needs to add any middle processes that slow down the use of the app. Initial loading of the data takes around 40 seconds to 45 seconds, averaging to below a minute at 43 seconds. This loading only takes place the first time that the user opens the app, or after an application update has been downloaded. The loaded data is stored inside the phone's internal storage, provided that the user allows the app to access files. This lessens the processing time that the app needs for its startup.

The home page consists of dropdowns of all provinces and municipalities, and once selected, the displayed details will correspond to the selected province/municipality. It also shows the Six Month Nationwide Rainfall Outlook that shows the amount of rainfall in a province by color ranging the provinces in shades of blue as shown in Figure 3, the lighter the color, the less rainfall there is in an area.

Updating the information of the app is done by updating the app itself. Since the available information for rainfall in an area spans 6-months, the data of the app changes monthly. This can be done by downloading the current version of the app either through Google Play Store (once available) or the QR code.

The home page consists of dropdowns of all provinces and municipalities, and once selected, the displayed details will correspond to the selected province/municipality.

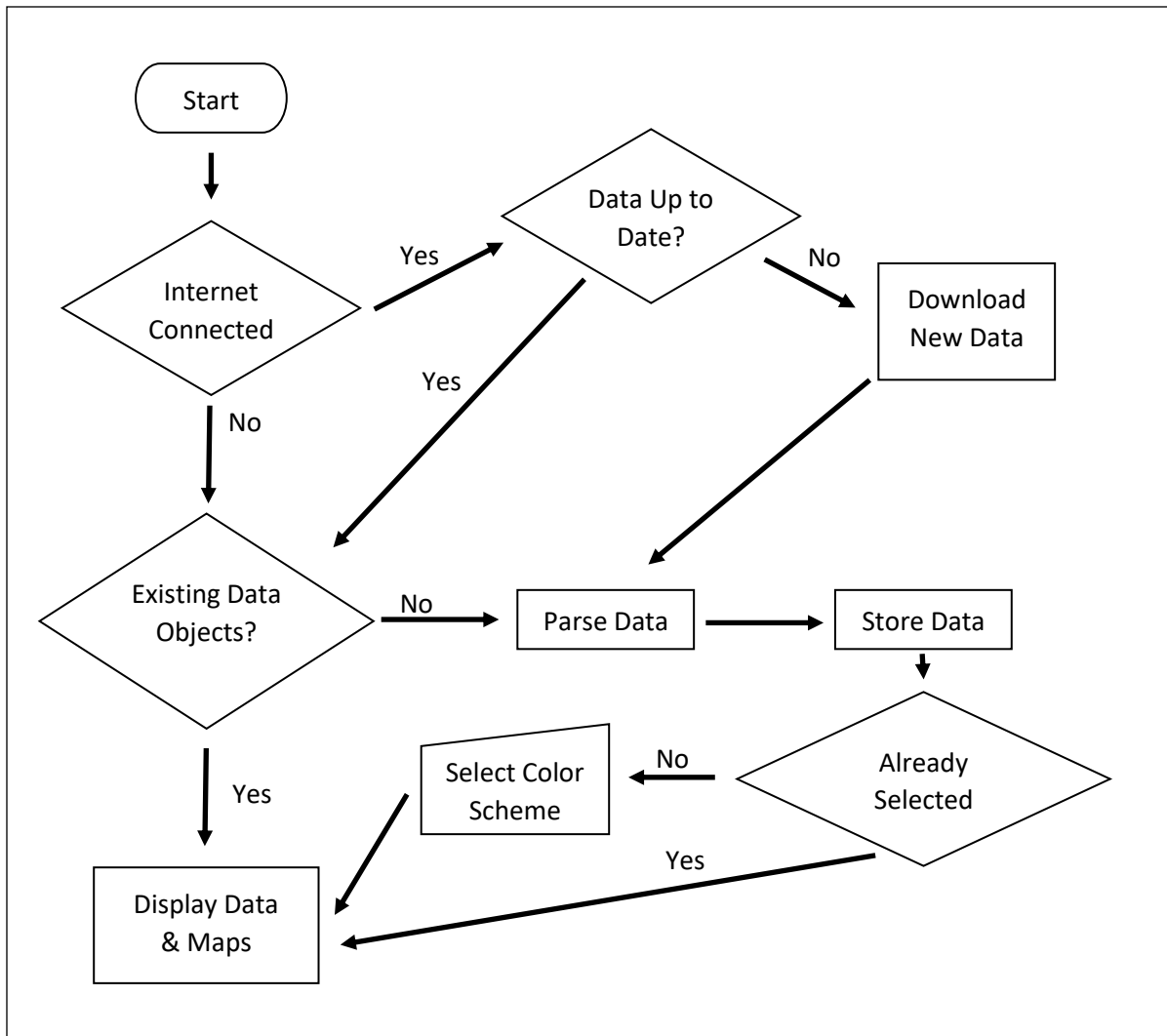
Figure 3 is where the application can be downloaded from. It is hosted by PhoneGap Build.



**Figure 3.** PhoneGap QR Code

Since the data changes monthly, the app determines whether or not your current files are up to date and automatically downloads new data as soon as the device connects to the internet, if the user opts for the app to ‘update automatically’. This can only happen once the application opens, because the app does not run as a background process. They can also choose to update manually, in the settings and updates tab.

**Figure 2.** Flowchart Summary of Use Cases



#### IV. Conclusion

The Rainfall Outlook Application for the Philippines was successfully created to display the rainfall in an province or municipality with the stipulation of simplicity and lesser strain on the computational device. The application performs better when many of the components have been preloaded or stored into a data object that can be re-accessed at a later date.

Visualization of Data helps make decisions for those that need there information because people recognize and recall visual cues better than other sensory input.

#### REFERENCES:

[http://wiki.geojson.org/GeoJSON draft version 6](http://wiki.geojson.org/GeoJSON_draft_version_6)

<https://gisgeography.com/what-gis-geographic-information-systems/>

<https://researchguides.library.wisc.edu/GIS>