

## Integrated Project 3 (IP3) Project Specification January 2019 DRAFT

<b>1</b>	<b>INTRODUCTION</b>	<b>2</b>
1.1	DISCUSSION OF SPECIFICATION	2
1.2	TECHNOLOGIES	2
1.3	DELIVERABLES	3
1.4	STRUCTURE OF THE SPECIFICATION DOCUMENT	3
<b>2</b>	<b>SPECIFICATION</b>	<b>4</b>
2.1	OVERVIEW	4
2.2	DETAILED SPECIFICATION	5
2.2.1	NON-FUNCTIONAL SPECIFICATIONS	5
2.3	FUNCTIONAL SPECIFICATIONS	5
2.3.1	OVERVIEW PAGE	5
2.3.2	AUTHORS PAGE	5
2.3.3	EARTHQUAKE MAPPING PAGE	5
2.3.4	WEATHER PAGE	6
2.3.5	INNOVATE: CREATE USEFUL AND INTERESTING VISUALISATIONS	6
2.4	TUTORIAL PAGES	6
2.4.1	GEOJSON OVERVIEW	7
2.4.2	EARTHQUAKE DATA	7
2.4.3	WEATHER DATA	7
2.4.4	JAVASCRIPT TECHNIQUES	7
2.4.5	GROUP-SELECTED DATA AND VISUALISATION TECHNIQUES	7
<b>3</b>	<b>GENERAL ADVICE ON SOFTWARE DEVELOPMENT</b>	<b>7</b>
3.1	TOOLS	8
<b>4</b>	<b>DATA FEEDS THAT ARE USED IN THE EXAMPLE CODE</b>	<b>8</b>
4.1	GEOJSON	8
4.2	WEATHER DATA FROM APIXU	8
<b>5</b>	<b>GOOGLE MAPS HELP</b>	<b>9</b>
<b>6</b>	<b>SAMPLE CODE</b>	<b>9</b>
6.1.1	API KEYS	9
6.1.2	CODE REPOSITORY	9
<b>7</b>	<b>EXAMPLES OF DATA</b>	<b>10</b>
7.1	EARTHQUAKE DATA	10
7.2	WEATHER DATA	13
<b>8</b>	<b>BIBLIOGRAPHY</b>	<b>14</b>
<b>9</b>	<b>SOME LINKS TO WEB-BASED VISUALISATION METHODS</b>	<b>14</b>

## 1 Introduction

Students will work in groups (of ideally six students) and will undertake the following tasks over one trimester: the creation of a web application that demonstrates how to access remote data sources via APIs and how to process and visualise this data through appropriate coding practices. As well as containing the implementation of the technologies, the web application will also deliver tutorial pages that explain where to source data, aspects of data content and structure, API-access, visualisation techniques, etc.

The project will involve research, conceptualisation, problem-analysis, design, implementation, test and evaluation of a web application that accesses a range of online data sources; specific extracts of these data sources are to be presented within the web application, including the use of Google Maps and other appropriate visualisation techniques. There is a **core aspect** to the specification and also an **innovation section** where students are free to select online data sources and deliver appropriate visualisations using any available technologies. The project will demand real-world skills in team-building, project management, report writing and presentation skills.

It is suggested that students undertake the following tasks in the order presented:

- Read this document
- Run the Google Maps code examples and examine them in detail; it would be good if students could work in pairs to undertake this task. All members of the team should be able to fully understand the examples after this point.
- Re-read this document.
- Create a list of deliverables based on the specification
- Divide tasks so that all group members have responsibilities. There is a range of activities and some will be more technical than others.
- Build a task list. Consider using Trello as a project management tool (see section 3 of the specification).

### 1.1 Discussion of Specification

You can seek clarifications, report problems or suggest improvements to the draft specification document at:

<https://docs.google.com/spreadsheets/d/1aALQ2LRjuXjWgBeLDxalgEjQQvzYpdrNAuThpoi1CvY/edit?usp=sharing>

### 1.2 Technologies

Students are free to utilise any technologies that they wish (however, we will not be expecting students to purchase any technologies).

For one part of the core aspect of the project you will be using Google Maps and we have provided some sample code; technologies used in the example code include HTML, CCS and JavaScript (including JQuery and Google Maps libraries). Please note that these are simple functional examples and do not make use of best practices for separating content, layout and JavaScript.

The web application can be implemented with client-side code and makes API calls to access data from external sources; no server-side code is required, although students are free to use server-side technologies if demanded by the innovative aspects of the implementation.

### 1.3 Deliverables

The following deliverables are required:

- A Group Report and a web application (70% of total assessment)
- Individual student reflective report (20% of total assessment)
- Presentation (10% of total assessment)

### 1.4 Structure of the specification document

The remainder of this specification document has the following sections:

- A specification of the required content of the web application
- A brief introduction to the GeoJSON format used by the example earthquake data sources
- An introduction to the example data sources (earthquake data and weather data).
- A set of code examples to introduce the following:
  - Loading data sources
  - Processing the data sources to extract specific data content
  - Using Google Maps
    - Loading a map
    - Placing and labelling markers on the map
    - Creating marker pop-up infoWindow content
    - Executing code when a marker is clicked
    - Creating clusters (groups) of markers when they are close to each other on a map.
  - Use of a specific library to facilitate dynamic creation of buttons on web pages. The buttons are configured to load specific data sources.

## 2 Specification

The specification includes an overview, followed by a more detailed list of requirements.

### 2.1 Overview

The web application will comprise a set of web pages and a suitable navigation structure to allow the user to access these pages. The styling, layout and implementation technology of the web application is to be defined by the student groups.

The specification has functional and non-functional requirements and these are equally important when it comes to the assessment of the project deliverables. The assessors will be looking at the project development and management processes as well as the technical development.

The following non-functional requirements must be followed:

- The application interface must be responsive, allowing it to execute within browsers on a range of devices (from mobile to desktop).
- The application must be well-structured (separation of code and content) and make use of structured styling techniques.
- The web application should be deployed to a publically-accessible web server.

The application must have the following specified pages:

- An overview page that explains the purpose and content of each aspect of the of the application.
- An author page.
- An earthquake mapping page with a set of buttons and a map. The buttons control a range of data feeds that are to be visualised on a map. A range of mapping features is to be employed.
- A weather page that allows a user to click on a map or enter a location name and to view weather information for that location.
- Tutorial pages
  - A page that introduces GeoJSON
  - A page that explains earthquake data fundamentals
  - A page that explains weather data fundamentals
  - A page that explains specific JavaScript techniques used to access and process data

The project requires students to innovate by researching the availability of other interesting data sources and visualising these in attractive ways:

- Create four additional pages that contain visualisations of data feeds selected by the student group. Rich, highly-visual representations should be utilised.
- Add some tutorial pages
  - Pages to explore the data feeds
  - Pages to explain the use of libraries/techniques used for the visualisations

## 2.2 Detailed Specification

### 2.2.1 Non-Functional Specifications

- The application should be *responsive* - “Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones)”<sup>1</sup>. Students are free to use any responsive framework.
- In pages that visualise data, you should fit the content so that the pages do not use scrolling.
- Pages should be as aesthetically attractive a possible
- The navigation controls should be in a regular format/position for each page of the application.
- Any keys used in the application should be protected as far as possible from reuse!
- Best practices: The application must separate content, layout and executable-code, linking to the appropriate external files, organised in a sensible manner.
- HTML5 files should validate at <https://validator.w3.org/> as properly defined HTML5
- CSS should validate at [https://jigsaw.w3.org/css-validator/#validate\\_by\\_upload+with\\_options](https://jigsaw.w3.org/css-validator/#validate_by_upload+with_options) as CSS Level 3
- The web application should be deployed to a publically-accessible web server.
- The development process should make use of a version control system (such as Git on GitHub or Bitbucket) that can support access by all group members.

## 2.3 Functional Specifications

### 2.3.1 Overview Page

This should explain the purpose and content of the IP3 Project. It should include a link to a version of the project specification document in a suitable web-readable format (such as PDF).

### 2.3.2 Authors Page

An author page that provides a brief introduction to each of the group members and describes the rolls/activities of each group member within the project.

### 2.3.3 Earthquake Mapping Page

To gain the fundamental understanding of the programming methods to be used on this page, students should refer to the example code discussed in section 6 of this report. You will find examples of all aspects that are required for your web application. You need to integrate these to implement the pages as specified below.

The Earthquake Mapping page should comprise the following:

---

<sup>1</sup> HTML Responsive Web Design (n.d.) W3schools.Com, [Online]. Available at [https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp) (Accessed 25 January 2019).

- A set of dynamically created buttons to select all feeds that are available from the following page: <https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php>
  - See the code example: IP3\_Map6.html . You will have to extend the number of buttons to cover all the groups ('Past Hour', 'Past Day', etc) and the individual items for each group ('Significant Earthquakes', 'M4.5+ Earthquakes', etc).
- A Google Map that is freshly populated by clicking any button. The map:
  - Must include clustered markers
    - See examples: IP3\_Map4.html
  - Must include an Google Maps infoWindow for each marker.
    - The infoWindow must have the textual content that is included in example: IP3\_Map3.html. The infoWindow should also contain a URL **which is different to the one on the example infoWindow**; when clicked, this URL should open an additional browser tab and dynamically populate this page with information relating to the geographical location. It is up to the group to research and find a suitable data feed (with API) that uses latitude and longitude as an index and has interesting data.

#### 2.3.4 Weather Page

This page should allow the user to enter specify a location using two different methods: 1. latitude and longitude, or 2. location name, and then click a button. The page then displays a current weather graphic image and as much useful information as you can find within the weather feed. [See example: IP3\_Map5.html as a starting point; you need to explore what data is returned from the apixu.com weather service API call. Use the explorer at: <https://www.apixu.com/api-explorer.aspx> to see the data that is returned]. Feel free to use any other weather service if you like. See any online weather site (such as BBC weather) to get some ideas about how you can present this data as useful weather information.

#### 2.3.5 Innovate: Create Useful and Interesting Visualisations

One of the major parts of the project is the freedom for groups to innovate by investigating new sources of data and the use of appropriate visualisation techniques. You are required to provide four implemented examples of data visualisation by sourcing appropriate data and web-based data-visualisation techniques.

Here are some interesting examples:

<http://www.nytimes.com/newsgraphics/2013/09/07/director-star-chart/index.html>

<https://flowingdata.com/2017/01/24/one-dataset-visualized-25-ways/>

<https://toplink.weforum.org/knowledge/explore>

<http://www.worldbank.org/> See the video on the first page.

See section 9 for some links to useful visualisation techniques.

## 2.4 Tutorial Pages

The tutorial pages should provide useful guidance on a range of data and techniques used in creating the web application. Each page should also provide links to a range of external tutorial pages that provide more detailed information.

#### 2.4.1 GeoJSON Overview

Introduce GeoJSON so that a reader who does not already know about this can gain a basic understanding.

#### 2.4.2 Earthquake Data

A page that explains earthquakes, earthquake data fundamentals and where this data can be sourced.

#### 2.4.3 Weather Data

A page that explains weather data fundamentals.

#### 2.4.4 JavaScript techniques

A page that explains specific JavaScript techniques used to access and process data.

#### 2.4.5 Group-selected data and visualisation techniques

Pages that describe the visualisation techniques and data sources that are used for the group-selected visualisation pages.

### 3 General Advice on Software Development

- Go through all the code examples, running them and examining each line of code. It is best to work in pairs to do this task.
- Run through the examples in order (from IP3\_Map1.html onwards).
- Plan to divide the project into short sprints and keep track of all tasks using a project management tool such as Trello (see below).
- When starting to build your own code, add one new feature at a time. When a new feature is working, then save that code as a working version (and don't touch it again!). Use this version as the starter code for the next version. If something goes wrong, you can simply start with the previous working version.
  - To help with this, use a version control system such as Git, hosted on GitHub or Bitbucket.
- Save all of your versions on ALL of the following: Online repositories (Cloud providers/GitHub/Bitbucket) | Local storage on your computer | USB storage
  - Yes – ALL of these. Don't be lazy about this. Something is bound to go wrong, somewhere, at some time. When you tell us that you have lost all your stuff, we will just say: "well, it's good that everything is backed up". If you tell us that you didn't back-up, then we will tell you that you don't get any marks for lost stuff.
  - Back-up frequently. How often? Well, how much work do you want to have to redo if you lose something? Your time is valuable, so save frequently and back up often.
- Split up tasks with your team members ASAP.
- Wireframing

- Use a lo-fi wireframing tool (such as Balsamiq <https://balsamiq.com/wireframes/> <https://balsamiq.com/givingback/free/classroom/> ) We have requested a license key for the class to use. This is a rapid and useful way of playing with layouts without being concerned with detail.
- Adobe have a free version of their XD tool: <https://www.adobe.com/uk/products/xd.html>
- Project management
  - Consider using Trello as a project management tool. Do some research on using Trello on software development projects. <https://trello.com/home> <https://trello.com/inspiration/engineering>

### 3.1 Tools

Use development tools that you familiar and confident with. Test your code using (for example) Chrome browser and its Developer Tools. You can easily add breakpoints to your code so that you can stop and examine variables. You can easily breakpoint within the response of an ajax call so that you can see the data returned from an API call.

Use a good quality editing environment for your code, such as the free Visual Studio Code environment or a good IDE that supports web technologies.

## 4 Data Feeds that are used in the example code

### 4.1 GeoJSON

GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON). It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties, and their spatial extents.

GeoJSON uses a geographic coordinate reference system, World Geodetic System 1984, and units of decimal degrees [1]. The examples that you will use are based on *points* (latitude, longitude). The page at [2] explains GeoJSON at an introductory and comprehensive level.

You can see many examples at: <http://geojson.xyz/>

You will be accessing GeoJSON data for earthquakes:

U.S Geological Survey. The page at [3] explains the GeoJSON format and also has links to predefined data feeds that you can access. The supplied HTML code examples show exactly how to access this data.

### 4.2 Weather Data from APIXU

The example code sources weather data from apixu.com [4]. Register for the free APIXU service. Try the API explorer at: <https://www.apixu.com/api-explorer.aspx>. This will provide an example of the data returned for a weather query. Documents and help are at:

<https://www.apixu.com/doc/>

Other weather services are available, so feel free to look at alternatives.



## 5 Google Maps Help

Documentation and examples are available for Google Maps at:

<https://cloud.google.com/maps-platform/>

The following documentation is especially useful:

<https://developers.google.com/maps/documentation/javascript/get-api-key>

<https://developers.google.com/maps/documentation/javascript/adding-a-google-map>

<https://developers.google.com/maps/documentation/javascript/marker-clustering>

<https://developers.google.com/maps/documentation/javascript/maptypes>

## 6 Sample Code

### 6.1.1 API Keys

In order to run the example code (see: 6.1.2 below) and also run your own code, you will need to register for the following API keys (free):

- Google Maps: <https://developers.google.com/maps/documentation/javascript/get-api-key>
- APIXU (for Weather data). Register at: <https://www.apixu.com/pricing.aspx> using the Free plan and its 'Get Started' button.

You will need to use your own API keys in the sample code. Once you have your keys, just do a search in the supplied HTML example source code files for the following keywords:

- YOUR-GOOGLE-MAPS-API-KEY (search for this)
  - In files: IP3\_Map1.html, IP3\_Map2.html, IP3\_Map3.html, IP3\_Map4.html, IP3\_Map5.html
  - Replace 'YOUR-GOOGLE-MAPS-API-KEY' with your own Google Maps API key
- YOUR-APIXU-KEY (search for this)
  - In files: IP3\_Map5.html
  - Replace 'YOUR-APIXU-KEY' with your own apixu API key

### 6.1.2 Code Repository

Sample code is made available in the folder at:

[https://drive.google.com/drive/folders/1-NOZYNxYP2NHd\\_v7YJ-yJUfX2cSg7Bka?usp=sharing](https://drive.google.com/drive/folders/1-NOZYNxYP2NHd_v7YJ-yJUfX2cSg7Bka?usp=sharing)

**Remember that you have to provide your own API keys!** See section **6.1.1** above.

File	Overview
IP3_Map1.html	Web page with Google map. Click button to populate map with earthquake markers from a specific USGS data feed. Markers are not labelled. Note that these examples use data sources that end with <b>.geojson</b> . Do NOT use sources that end with <b>.geojsonp</b>
IP3_Map2.html	Web page with Google map. Click button to populate map with earthquake markers from a specific USGS data feed. Markers are labelled to show the magnitude of the earthquake.

IP3_Map3.html	Web page with Google map. Click button to populate map with earthquake markers from a specific USGS data feed. Markers are labelled to show the magnitude of the earthquake. Each marker has an infoWindow that includes a URL that is part of the GeoJSON data. The infoWindow URL opens in the same browser tab.
IP3_Map4.html	Web page with Google map, earthquake markers and clustering. Click on a cluster to zoom in to see markers (and maybe other clusters).
IP3_Map5.html	Web page with Google map and markers labelled with earthquake magnitude. Accesses weather data when clicking on a Google Maps marker. The weather data is used to populate a weather image and a brief weather summary.
IP3_Map6.html	Constructing a set of buttons where each one make a different ajax request, each button associated with a specific URL.

## 7 Examples of Data

### 7.1 Earthquake Data

The data is shown as an illustration. The data will obviously change over time.

Feed: [https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5\\_hour.geojson](https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_hour.geojson)

```
{
  "type": "FeatureCollection",
  "metadata": {
    "generated": 1533742458000,
    "url": "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_hour.geojson",
    "title": "USGS Magnitude 2.5+ Earthquakes, Past Hour",
    "status": 200,
    "api": "1.5.8",
```

```

    "count": 1
  },
  "features": [
    {
      "type": "Feature",
      "properties": {
        "mag": 2.6,
        "place": "6km ENE of Mammoth Lakes, CA",
        "time": 1533741217560,
        "updated": 1533742270324,
        "tz": -480,
        "url": "https://earthquake.usgs.gov/earthquakes/eventpage/nc73066416",
        "detail": "https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/nc73066416.geojson",
        "felt": 2,
        "cdi": 2.2,
        "mmi": null,
        "alert": null,
        "status": "automatic",
        "tsunami": 0,
        "sig": 104,
        "net": "nc",
        "code": "73066416",
        "ids": ",nc73066416,",
        "sources": ",nc,",
        "types": ",dyfi,focal-mechanism,geoserve,nearby-cities,origin,phase-data,scitech-link,",
        "nst": 30,
        "dmin": 0.003796,
        "rms": 0.05,
        "gap": 53,
        "magType": "md",
        "type": "earthquake",
        "title": "M 2.6 - 6km ENE of Mammoth Lakes, CA"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          -118.9095001,
          37.6525002,
          2.43
        ]
      },
      "id": "nc73066416"
    }
  ]
}

```

Weather Data: OVER/

## 7.2 Weather Data

The data is shown as an illustration. The data will obviously change over time.

Here is a request that uses latitude and longitude as parameters:

[http://api.apixu.com/v1/current.json?key=PUT\\_YOUR\\_KEY\\_HERE&q=-5.5221,151.7805](http://api.apixu.com/v1/current.json?key=PUT_YOUR_KEY_HERE&q=-5.5221,151.7805)

```
{
  "location": {
    "name": "Buka",
    "region": "East New Britain",
    "country": "Papua New Guinea",
    "lat": -5.52,
    "lon": 151.78,
    "tz_id": "Pacific/Port_Moresby",
    "localtime_epoch": 1533743239,
    "localtime": "2018-08-09 1:47"
  },
  "current": {
    "last_updated_epoch": 1533742339,
    "last_updated": "2018-08-09 01:32",
    "temp_c": 20.1,
    "temp_f": 68.2,
    "is_day": 0,
    "condition": {
      "text": "Partly cloudy",
      "icon": "//cdn.apixu.com/weather/64x64/night/116.png",
      "code": 1003
    },
    "wind_mph": 3.4,
    "wind_kph": 5.4,
    "wind_degree": 214,
    "wind_dir": "SW",
    "pressure_mb": 1012.0,
    "pressure_in": 30.3,
    "precip_mm": 0.0,
    "precip_in": 0.0,
    "humidity": 86,
    "cloud": 9,
    "feelslike_c": 20.1,
    "feelslike_f": 68.2,
    "vis_km": 18.0,
    "vis_miles": 11.0
  }
}
```

## 8 Bibliography

- [1] <http://geojson.org/>, "GeoJSON," [Online]. Available: <http://geojson.org/>.
- [2] "More than you ever wanted to know about GeoJSON," [Online]. Available: <https://macwright.org/2015/03/23/geojson-second-bite.html>.
- [3] "GeoJSON Summary Format," [Online]. Available: <https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php>.
- [4] "Apixu," [Online]. Available: <https://www.apixu.com/>.
- [5] <http://geojson.org/>. [Online]. Available: <https://tools.ietf.org/html/rfc7946>.

## 9 Some links to web-based visualisation methods

These are just a few links to allow you to get started. You will want to undertake your own research.

<https://www.quora.com/What-are-the-best-tools-for-web-based-data-visualization>  
<https://bigdata-madesimple.com/20-web-based-visualisation-tools/>  
<https://www.slant.co/options/10577/alternatives/~d3-js-alternatives>