

COMP9323 ANZ Insight API Project

**Brainstorming Report**

2014 Semester 2

**Lecturer**: Boualem Benatallah

**Mentor:** Amin Beheshti

**Group member:**

-Bani Kaur

-Diyin Zhou

-Nitish Gupta

-Yifan Fan

Table of Content

[**Project Summary / Introduction** 3](#_Toc396080904)

[**Technology/API Investigation** 5](#_Toc396080905)

[What is Data Visualization? 5](#_Toc396080906)

[How Data Visualization works? 5](#_Toc396080907)

[Analyze Current Mainstream Visualization APIs 6](#_Toc396080908)

[*Google Charts/Maps* 6](#_Toc396080909)

[*Chartio* 7](#_Toc396080910)

[*iChart* 7](#_Toc396080911)

[*Hightcharts* 8](#_Toc396080912)

[*Dimple* 9](#_Toc396080913)

[*Conclusion* 10](#_Toc396080914)

[**Brainstorming Q&A** 11](#_Toc396080915)

[*How to maintain the effective communication throughout the team?* 11](#_Toc396080916)

[*Where to store files including documents and program code?* 11](#_Toc396080917)

[*What is BPEL and IAAS?* 11](#_Toc396080918)

[*How to use BPEL and IAAS for our project?* 11](#_Toc396080919)

[*Where to deploy our project APIs/Web Applications?* 12](#_Toc396080920)

[*How to test our project APIs/Web Applications?* 12](#_Toc396080921)

[*Do we need to test the maximum capacity of our APIs/Web Applications?* 12](#_Toc396080922)

[*Why small business needs visualization?* 12](#_Toc396080923)

[*What services does ANZ currently have?* 12](#_Toc396080924)

[*What technologies do we need for the project?* 13](#_Toc396080925)

[*More quesions and answers goes here...* 13](#_Toc396080926)

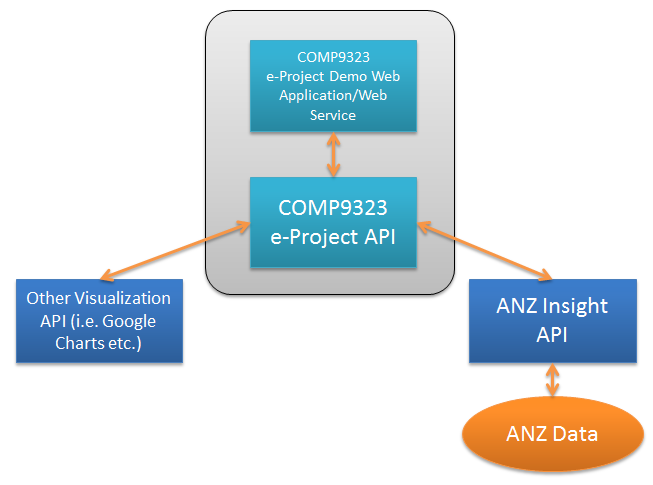
[**Evaluation of ideas** 14](#_Toc396080927)

# **Project Summary / Introduction**

This report aims to record, summarize and conclude the works done during the Brainstorming session in Phase 1 of the ANZ Insight Project of COMP9323.

In this project, we will be building Web applications/APIs mainly based on the ANZ Insight API provided by the ANZ Bank. ANZ Insight API provides various kinds of data located in ANZ bank systems. By taking advantage of this API, we are going to build Web applications focusing on the potential business needs of small businesses. The applications will enable them of reviewing the basic information of their business like sales, profit or debts etc. It is also supposed to deliver some sort of advanced functionality like dynamically interacting with the application to do analysis or prediction through graphical user interface.

In this phase, we will investigate the available APIs which would be used to visualize data, decide which of them can be used and which of them are mostly suitable for building the web application during the project through research and discussion. Possible research and discussion results regarding the pros and cons, background technologies and expendability of the third party API would also be included.



The above diagram illustrates the holistic view of what this project does and major components that the project will be interacting with.

Another major task at this phase is to do the brainstorming, we would come up with a host of issues we may have to practically deal with during the project, and suggest solutions which may address these issues. The issues/questions raised during the brainstorming session shall also be recorded and included.

Finally, since the ANZ API is only for research use, it is not commercialized yet. In addition, there’s also no firmed user/customer requirement at this moment. Thus at the beginning of this project, a crucial task emerges. We have to put ourselves into users’ shoes to assume the potential requirements of the customers (who are mainly users from small businesses). That may somehow decide the direction where the project goes.

# **Technology/API Investigation**

## What is Data Visualization?

In the Information Time today, various kinds of data are being generated in a unbelievable pace, like business transaction data, user management data, customer service data, bank transaction data, logistic data, financial data, inventory data and countless other forms of data. Either online or offline, they are stored somewhere and meant to be accessed by people who need corresponding information.

Data are cold, objective and abstractive. It costs humanity a great deal of time to read and comprehend the information that a set of data carries, especially out of the analytical purpose.

Data Visualization makes it much easier, quicker and more intuitive to grasp the meaning of the data, by aggregating, categorizing and transforming data into visualized objects in terms of pictures, charts, animations or even interactive and controllable components.

Data Visualization is especially useful when the data need to be presented to public people, people with less technological background or high level management.

## How Data Visualization works?

The most basic way in which the Data Visualization works is described as following steps:

1. The service requestor prepares the data which needs to be visualized.
2. The data is presented to the Data Visualization API
3. Data Visualization API reads and processes the data
4. Some output i.e. visualized objects are generated and sent back to the service requestor in some ways.
5. The service requestor presents the visualized objects instead of/along with the original data.

The detailed ways of doing this, the background architectures or technologies will be discussed in later sections.

## Analyze Current Mainstream Visualization APIs

### **Google Charts/Maps**

**Description**:

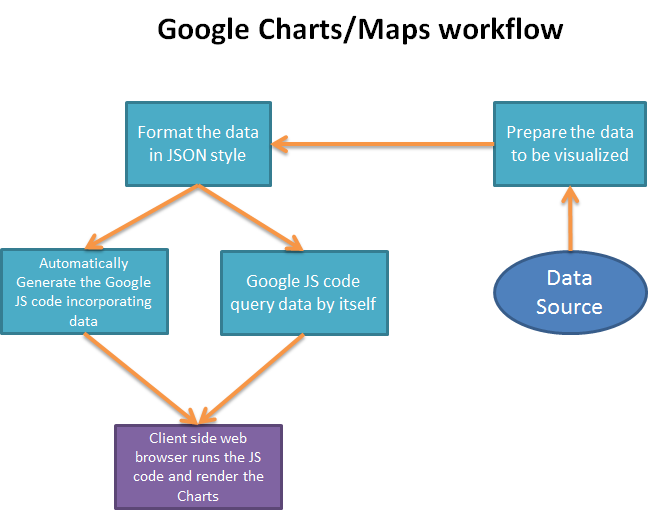
The Google Charts/Maps are JavaScript based solutions.

Basically, service requestor prepares data in JSON format, either in data files or dynamically generated codes. Then Google Charts/Maps JS class is invoked to visualize the prepared data.

An optional but recommended way of presenting data is to use Python to automatically generate the JS code to the client side.

In addition, Google JS code can also query data by itself, which may reduce the length of code generated and retrieve data at run-time.

Finally, client side web browser runs the JS code and renders the Charts.

******

**Pros**:

* Easy to implement and understand by service requestor
* No remote calls to google web services
* More computational load has been distributed to client side
* Good extandability
* Supports a number of data manipulation method in terms of Group or Join
* Event based architecture makes the real-time Data Visualization possible
* Basic interactive analytical functionality available
* A lot more APIs based on Google Visualization available

**Cons**:

* JS can be disabled or blocked at client side
* The capacity of processing large volume of data is unknown

### **Chartio**

**Description**:

**Pros**:

Cons:

### **iChart**

**Description**:

In order to take advantage of this API, developers need to prepare data in a quite simplified data format called TSV (Tab Separated Value), either generate these TSV files and place them in the web server with URL links to them, or prepare some JSP/Servlets generating output in this format dynamically.

Developers then need to setup a Data API Connector on the iChart website to direct the API to the output data. The iChart API reads the output data, populate the charts and store them on the iChart server.

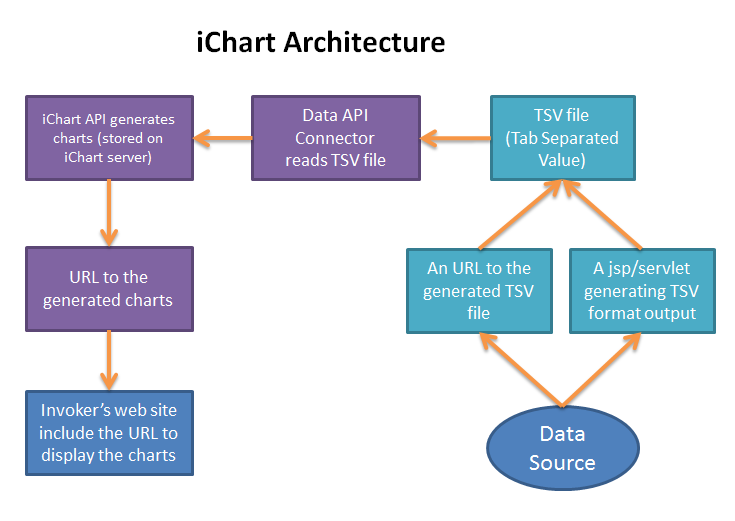
URLs pointing to the generated charts on the iChart server are provided, so that the developers can include these links on their own website.

**Pros**:

* Easy to configure and use
* Data format is simple, no complicated syntax to follow, and no software development experience requirement.
* Reduced load on the developers’ server.
* Real time data read and chart generation

Cons:

* iChart server capacity concern
* Extendibility concern
* No data analysis related functionality
* Whether the connector configuration can be automated is unknown



### **Hightcharts**

**Description:**

[Highcharts](http://www.highcharts.com/) is an amazing charting library written in JavaScript. It is free for non-commercial use if you want to run the library on a commercial website you'll need a license.

Hightcharts is currently using some function of Google API. As a result, when we try to import and install Hightcharts API, we also need to import Google JavaScript library.

Hightcharts requires two files to run, [highcharts.js](http://code.highcharts.com/highcharts.js) and either jQuery, MooTools or Prototype or our own Hightcharts Standalone Framework which are used for some common JavaScript tasks. If you're installing Hightstock, the procedure is the same as below, except the JavaScript file name is highstock.js rather than highcharts.js.

**Pros:**

* Easy to configure and use
* Having Hightstock and Hightmap, which are also Visualization APIs
* Easy to understand the code of JavaScript library

**Cons:**

* Only provide some kinds of charts
* Also use some Google API



### **Dimple**

**Description:**

Dimple is an object-oriented API for business analytics powered by d3.

The aim of dimple is to open up the power and flexibility of d3 to analysts. It aims to give a gentle learning curve and minimal code to achieve something productive. It also exposes the d3 objects so you can pick them up and run to create some really cool stuff.

**Pros**:

* Easy to configure and use
* The function is simple and easy to understand all classes
* Use d3.js
* Easy to understand the code of JavaScript library

Cons:

* Only provide some kinds of charts
* Not impressive



## *Conclusion*

# **Brainstorming Q&A**

In this section, the questions raised and potential answers provided during the Brainstorming sessions have been recorded. In addition, the conclusions are also included if the team reaches a consensus on any of the individual question.

### **How to maintain the effective communication throughout the team?**

**1. A discussion group on Facebook**

**2. Routine weekly meeting, by taking everybody’s available schedule into consideration.**

**3. Everybody checks the posts on Facebook discussion group time to time.**

**4. For urgent issues, make contact with team members through cell phones.**

### **Where to store files including documents and program code?**

**1. Github, a web based version control utility.**

**2. Google Doc, a web-based file sharing utility.**

**3. Dropbox, a web-based file sharing utility, integrated with Facebook.**

### **What is BPEL and IAAS?**

**1.**

### **How to use BPEL and IAAS for our project?**

**1. Gith**

### **Where to deploy our project APIs/Web Applications?**

**1. Git**

### **How to test our project APIs/Web Applications?**

**1. Git**

### **Do we need to test the maximum capacity of our APIs/Web Applications?**

**1. Git**

### **Why small business needs visualization?**

1. Decision making
2. Marketing
3. Sales records
4. Strategic analysis

### **What services does ANZ currently have?**

1. **Small business bank account**

   a.free transactions

   b.competitive interest rates

    c.24 hour access to your money

1. **Business loans**

    Flexible repayment and interest rate options \*\*\*visualisation

1. **Business saving account**

  a.short and long term business savings solutions

  b.5 free transactions every month\* with no monthly account servicing fees#

  c.flexible terms

1. **Merchant services**

  a.pay-pass

### **What technologies do we need for the project?**

1. Cross-site request (Ajax JSONP, for some API requests )

2. Adopting framework (PHP Codeigniter framework)

   \*easy to learn

   \*fast to build

   \*have many functions ,such as email module, session module

3. JavaScript (easy to use. many API provide javascript service)

4. Security concern (low level requirement)

5. Database (MySQL, Derby)

### **More quesions and answers goes here...**

# **Evaluation of ideas**

The brainstorming has been divided into 4 sessions: mentoring, individual research, face to face discussion and task allocation, completing allocated tasks and final consolidation.

In first session, we had a meeting with mentor. Mentor introduced the background, overview and purpose of the project. Also a brief introduction with a Visualization API was given, leading us to the starting point of the brainstorming.

During the mentoring session, there were also some questions had been addressed, like the target audience (which initially be the staff of the ANZ bank, and eventually will be the users from small businesses), and current user requirements. Actually, there’s no specific user requirement right now, and we are still in a stage of research regarding the possible functionalities that we may offer based on the ANZ Insight API.

For the second session, a few holistic questions have been raised and each of the team members is required to do some research on the topics like:

* What is Data Visualization?
* How it works?
* What are the mainstream Data Visualization APIs and their pros and cons

Before the face-to-face group discussion, everybody needs to have a general idea about what we do, where we are and where to go.

At the third session, we conducted a face-to-face meeting, sharing consolidating some investigation results of the current Visualization APIs. Each of us gave a short presentation of the APIs that he/she researched. Also a brainstorming activity was performed during the meeting, questions were raised, and quick answers were given if possible. Solved or unsolved questions were recorded in the document. At the end of the meeting, everybody had been assigned two major tasks: 1. Review the consolidated API information and prepare reason why we should use that. 2. Solve the assigned questions in the unsolved question list.

For the last session, we meet again to finalize the technologies to use in terms of APIs and the architecture we would like to use for the project in terms of Web Services and Demo Web Applications. Also the unsolved questions had been ticked off by filling the works assigned and completed by every team member.

When discussing which API to use, we took the following elements into consideration: difficulty of implementation, extendibility, performance, advanced features like interactive operations or data manipulation etc.