

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** 4](#_Toc396168380)

[**Technology/API Investigation** 6](#_Toc396168381)

[What is Data Visualization? 6](#_Toc396168382)

[How Data Visualization works? 6](#_Toc396168383)

[Analyze Current Mainstream Visualization APIs 7](#_Toc396168384)

[*Google Charts/Maps* 7](#_Toc396168385)

[*Chartio* 8](#_Toc396168386)

[*iChart* 8](#_Toc396168387)

[*HighCharts* 9](#_Toc396168388)

[*Dimple* 10](#_Toc396168389)

[*Drupal Visualization API:* 11](#_Toc396168390)

[*jQuery Visualize* 12](#_Toc396168391)

[*DYGRAPH* 14](#_Toc396168392)

[*Conclusion* 15](#_Toc396168393)

[**Brainstorming Q&A** 16](#_Toc396168394)

[*How to maintain the effective communication throughout the team?* 16](#_Toc396168395)

[*Where to store files including documents and program code?* 16](#_Toc396168396)

[*What is BPEL and IAAS?* 16](#_Toc396168397)

[*How to use BPEL and IAAS for our project?* 17](#_Toc396168398)

[*Where to deploy our project APIs/Web Applications?* 17](#_Toc396168399)

[*How to test our project APIs/Web Applications?* 17](#_Toc396168400)

[*Do we need to test the maximum capacity of our APIs/Web Applications?* 17](#_Toc396168401)

[*Why small business needs visualization?* 18](#_Toc396168402)

[*What services does ANZ currently have?* 18](#_Toc396168403)

[*What technologies do we need for the project?* 18](#_Toc396168404)

[*Do we need to provide comparison feature?* 18](#_Toc396168405)

[*What kinds of visualization do we need?* 19](#_Toc396168406)

[*What kinds of data we can get from ANZ Insight?* 19](#_Toc396168407)

[*In which way the ANZ Insight provides us the data?* 19](#_Toc396168408)

[*What kinds of data the users are supposed to view?* 19](#_Toc396168409)

[*How we deal with the compatibility problem?* 19](#_Toc396168410)

[*What database should we use?* 19](#_Toc396168411)

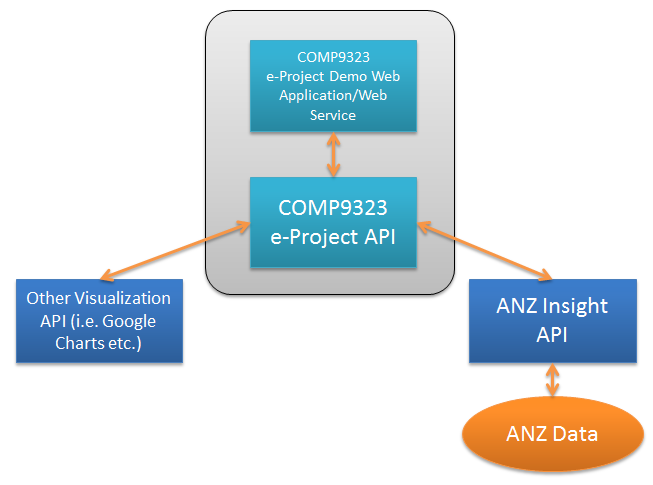
[**Evaluation of ideas** 20](#_Toc396168412)

# **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 no firmed user/customer requirement at this moment either. 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 at an 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**:

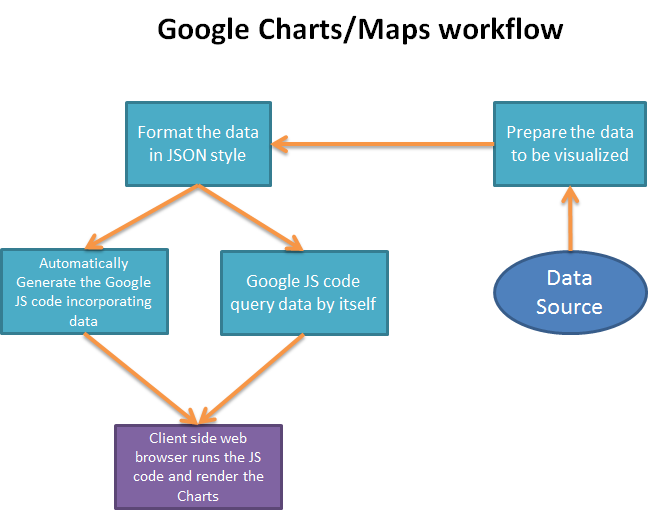
The Google Charts/Maps are JavaScript based solutions.

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
* Compatible with mobile device

**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.

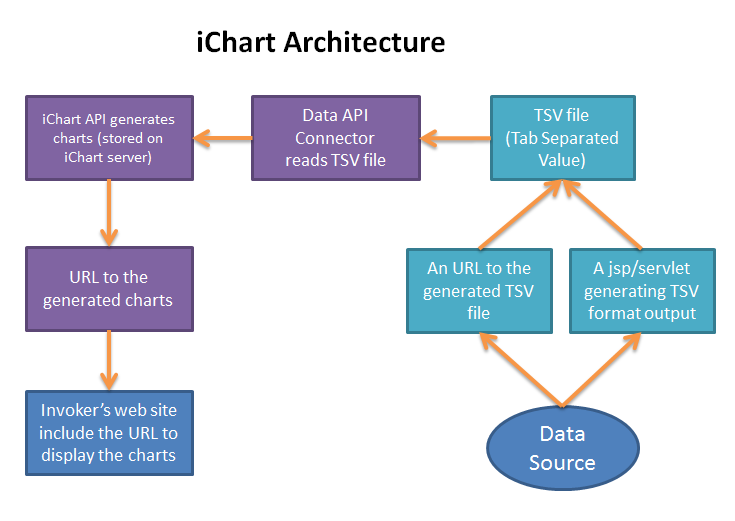
URLs pointing to the generated charts on the iChart server are provided, so that the developers can include these links on their own website.

**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



### **HighCharts**

**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.

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

HighCharts requires two files to run, [highcharts.js](http://code.highcharts.com/highcharts.js) and either jQuery, MooTools or Prototype or our own HighCharts 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

### **Drupal Visualization API:**

**Description:**

It is an open API based on Java. The underlying framework is Google visualization1 and Highcharts2. The system plugin, which is used, is CTools. The overall structure supports interactive visualizations and all the functionalities of the Google Charts. And in a similar fashion it includes features from Highcharts.

**Pros:**

1. Various types of Metadata support includes,
2. Geospatial of data
3. Social tagging of data
4. Interactive data analysis
5. Dynamic graph generation with real time data addition
6. Nice compatibility, cell phone web browser support for both Android and IOS
7. It supports multiple data source from raw data, JSON to database.
8. JS based technology, no need to install additional plugins at client side.
9. It supports interactive operations like adjusting filters or even drill down/up.
10. Good documentations and substantial amont of support information.
11. Free and open source.

**Cons:**

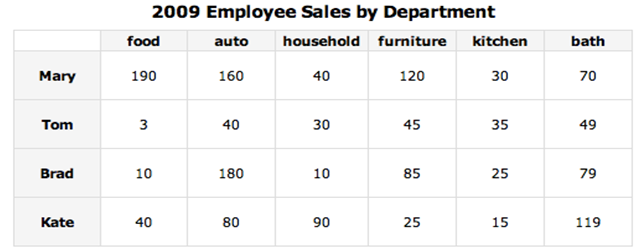
1. Performance issue when running at specific versions of Android OS.
2. Performance issue when running at lower versions of IE.

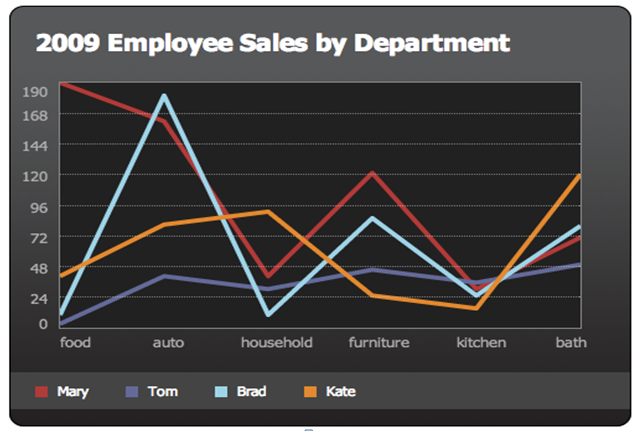
Reference links:

<https://developers.google.com/chart/interactive/docs/gallery>

<http://www.highcharts.com/>

### **jQuery Visualize**





**Description:**

It is a simple and easy tool for data analysis. It can simply convert an HTML table into a chart. It supports:

1. Pie charts
2. Bar charts
3. Line charts
4. Area charts

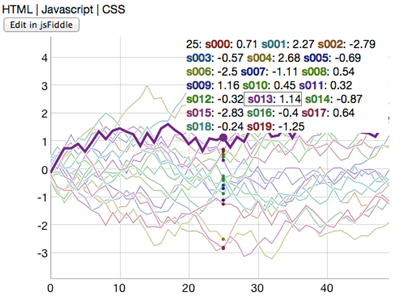
**Pros**:

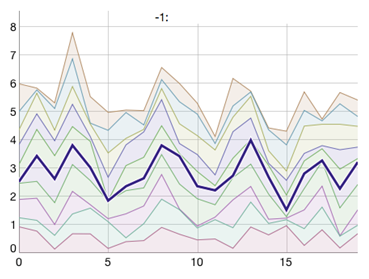
1. It draws charts directly from an HTML table. This means we don’t need to put the data into a special format. This implies that if there is a table for the data sets, a single jQuery call can create the above mentioned charts.
2. The styling is usable in many situations. Also one can style almost every element with CSS. However the elements in the actual chart are supported by JavaScript.

Cons:

1. It is not interactive i.e. no animation option and no hovering over a data point.
2. Not enough options/ functionalities offered as compared to its counter APIs.

### **DYGRAPH**



**Pros**:

1. It can handle huge sets of data without slack and getting hanged.
2. It is interactive unlike Jquery which doesn’t have point touch features.
3. One can customize it and can help different varieties of charts etc using dygraphs.
4. It handles time series very well.
5. Displays values on mouseover (this makes it easily discoverable)
6. It is extremely compatible and the touch zoom facility is also available for the android version.

## *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.

4. Facebook file sharing.

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

BPEL stands for Business Process Expression Language, which is a standardized language describing the business process of an application. It is expecially useful when developing large scale enterprise applications.

IAAS

IAAS stands for Infrastructure as a Service, which is a concept of Cloud computing.

Amazon provides the infrastructure through its cloud service. Customers/users can subscribe a certain amount of fully furnished web servers with specified system capacity. Customers/users do not need to care about the detailed maintainance of these servers, they pay and they get the control of these services in terms of web servers. It’s an idea place to deploy and publish the WS or Web Applications for this project.

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

For BPEL

1. It needs excessive effort to learn and understand BPEL.
2. It’s hard to incorporate in this project
3. It’s best for complicated enterprise projects, not fit for this one.

Thus we need to keep a cautious attitude towards BPEL.

For IAAS

Amazon is the only and ideal IAAS service provider we can come up with, especially when it provides free services for study use to university students. We would follow the instructions given in COMP9322 about how to use IAAS to take advantage of this service in this course and this project.

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

1. Best at IAAS
2. Free for student use

Similar question and answer as previous one.

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

1. Manual test
2. Test cases may be needed.
3. Everyone needs to take part in the test activity
4. Automation test for capacity in terms of large volume of data and huge number of concurrent user access.

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

Yes, but the priority is low, since the project is a research on how we can provide by taking advantage of ANZ Insight. System capacity is not an urgent issue. We will consider it if we have time.

### **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)

### **Do we need to provide comparison feature?**

Yes, we can.

Even if the Google Charts does not support this, it is possible for us to implement this feature by our own. We can try to display the information of the data which are ready to be compared, and calculate the gaps between them.

### **What kinds of visualization do we need?**

1. Charts
2. Interactive objects
3. Animations
4. Timeline
5. Filters
6. Groupping
7. Drill down/up

### **What kinds of data we can get from ANZ Insight?**

TBD.

### **In which way the ANZ Insight provides us the data?**

TBD.

### **What kinds of data the users are supposed to view?**

TBD.

### **How we deal with the compatibility problem?**

There’s no obvious way of solving the current compatibility issues. It all depends on the API we are about to use. In additon, the purpose of this project is to explore the possibility of how to present the data to the users, therefore the compatibility issues would not be an urgent problem to be tackled with.

### **What database should we use?**

1. MySQL easy to use and setup, fully featured
2. Derby is too small, lack of functionality, not scalable, poor performance when data go large and not reliable.

# **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 and 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, compatibility, advanced features like interactive operations or data manipulation , documentation and support etc.