ROLE OF API TECHNOLOGY IN IOT FOR A LOCATION-BASED SERVICE

by

AUSTIN AGBO
URN: 6346574

A dissertation submitted in partial fulfilment of the
requirements for the award of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

September 2015

Department of Computer Science
University of Surrey
Guildford GU2 7XH

Supervised by: Paul Krause

I declare that this dissertation is my own work and that the work of others is acknowledged and indicated by explicit references.

AUSTIN AGBO
September 2015

Abstract

This project covers location-based systems, touches on the future of IOT and how API technology allows this interoperability. During the project's life cycle, a web application that streamed events from an API was tested for validity and verification. Where the application is validated for a SUMO test questionnaire. This work is based around API services but also considers the integration of software with physical devices.

Acknowledgements

I'd like to thank my supervisor Paul Krause, for being of so much help, my parents for always been there for me and all the participants who took part in the software testing phase.

Contents

List of Figures

List of Tables

Glossary

A_f	The source message, being a sequence of f source symbols $a_1a_2 \dots a_f$
a_i	The i^{th} symbol in the source message, where $a_i \in S_m$
B_g	The decoded message, being a sequence of g source symbols $b_1b_2 \dots b_g$
b_i	The i^{th} symbol in the decoded message, where $b_i \in S_m$
C_h	The transmitted (compressed) message, being a sequence of h Tunstall codewords $c_1c_2 \dots c_h$
c_i	The i^{th} codeword in the transmitted message, where $c_i \in T_n$
D_h	The received (compressed) message, which for a complete Tunstall code is a sequence of h Tunstall codewords $d_1d_2 \dots d_h$
d_i	The i^{th} codeword in the received message, where $d_i \in T_n$ for a complete Tunstall code

Abbreviations

LBS	Location-Based Service
GIS	Geographic Information Service
API	Application Programming Interface
SDK	Software Development Kit
SAAS	Software As A Service
MVC	Model-View-Controller

Chapter 1

Introduction

1.1 Background

Firstly, an A.P.I. can simply be described as that which provides a developer with an interface that requires programming to access some content. Some other key definitions are described below: LBS – can be described generally as a name for new types of services, where location information is paramount to the service been provided (Ajam, Ahson & Ilyas 2010). Similarly termed services like location-aware service, location-related service, or location service mean the same thing and therefore can be used interchangeably. Now, we cannot discuss LBS without GIS, as the latter makes it possible to make use of location-based services. GIS – in a broad sense, can be defined as an information system, which is designed to provide data referenced by geographic coordinates or also described as a database system with particular practices for spatially-reference data, as well as a set of methods to work with this data (Foote & Lynch 1995). GIS should be thought as the socket while LBS is a plug. For a GIS to be efficient in this day and age (it should have an API), Then we can design our Location-based services to make use of this API. (Redraw diagram and reference from that LBS note)

Pull LBS: This refers to when a user initiates a service request to the service provider expecting certain content, information is then serviced specifically for the user's location. A simple example is - weather forecast, which will vary pending on the user's location. Being in the smart phone era, a user might check a mobile application every morning to find out what weather is going to be like for the day. Push LBS: In this case, the service provider initiates the service delivery, which means – location information may be requested of the user and then a value-added service is proposed. For example, a user can use his/her email account to register to the provider for a different service, and receive a weather focus every morning regardless if user still exists in recorded location. Tracking LBS: A service provider can continuously track their users, so in a fleet management company for example, their employees are tracked so their routes are optimized (Foote & Lynch 1995).

1.2 Project Overview

A lot of business opportunity is growing in the IOT movement across industries by the transformation of remote devices into intelligent objects that send and receive data to back-end platforms. Managing an API is a huge part of the interoperability, scale, and control for IOT. When web APIs, its management and a RESTful architecture are designed on Standard methods, immense value is provided in reducing the task of interoperability in heterogeneous systems that handle vast amounts of data.

1.3 Project Objectives

The system aimed to demonstrate integration of API services for a Location-based System.

1. Investigate location-based services IOT
2. Develop location-based application using API that hosts information service
3. Test application for validity
4. Prove why API technology is the most important link in IOT

1.4 Project Limitations

API technology is well known and popularly used by developers and a very essential tool but a lot of focus is been put into concern of IOT rather than the technological advancement it can bring to society. Though the interest is rising, we are not yet at the point where students are building IOT components rampantly as some on occasions they will require concrete knowledge of hardware and software. Thus can be considered a high learning curve experiment for a computer science student with no experience in electronic side of things. Also most IOT data was found on blogs as much university study hasn't yet been carried out on the topic.

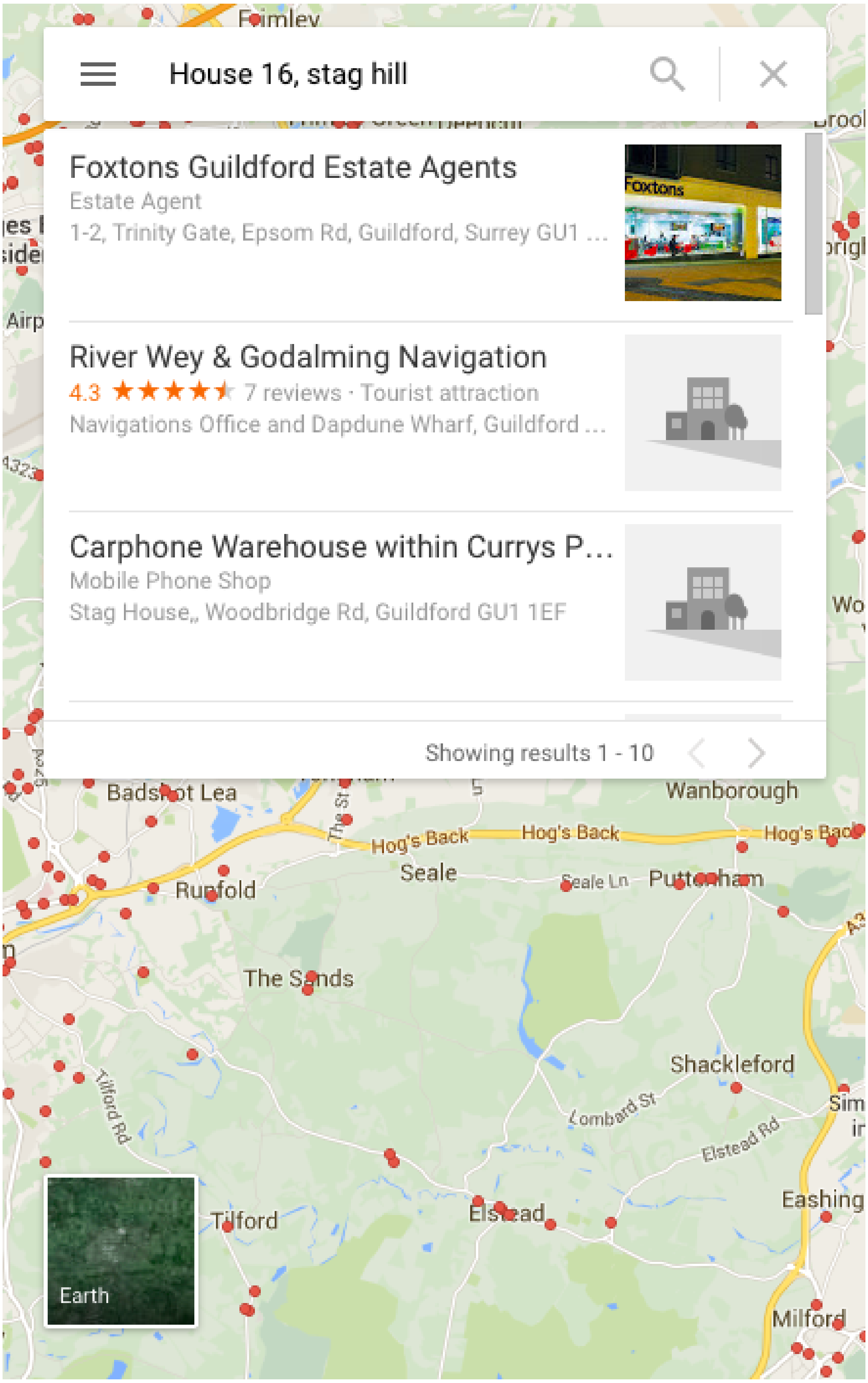
Chapter 2

Literature Review

2.1 LBS with API technology as resource

Asking for one's address over the Internet has become a thing of the past with some application providers. Rather the question becomes, *Can I use your location?* This is partly due to most applications having features that fetch location by utilizing user's position (longitude, latitude), with user's consent of course. Most of these applications might use an API to fetch this Geographical Information. One of the leading location-based services (i.e. SAAS) that provide this kind of API service is Google Map. As a web application, it provides an information system that can be consumed through mobile, tablets, laptops or any device capable of Internet Connectivity, where it's features include: Displaying related address information to a location that has being placed in its search engine and displaying information expected when phrases like *restaurants near me* , *cinema near me* are searched.

Google's API, which is quite useful to application developers in providing a comprehensive map for any application. As a user of this API, one can build applications utilizing it's map service as a step to then offering some sort of Information Service (Google 2005). The API which provides content and core functionality for the application designed for this study is a Concert and Events Information Service (http://api.eventful.com), its main site (http://eventful.com/) can be classified as LBS as it provides users with information based on their location. Simply visiting the website, it immediately provides a user with content of events occurring in the user's city. Regardless of this automated service, a user can specify and search for events in preferred locations.


House 16, stag hill

Foxtons Guildford Estate Agents
Estate Agent
1-2, Trinity Gate, Epsom Rd, Guildford, Surrey GU1 ...

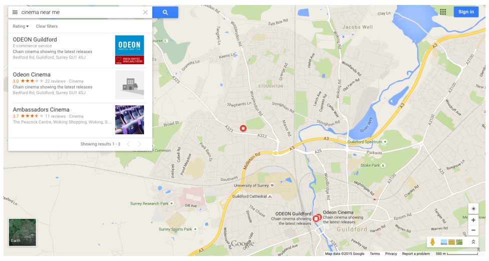
River Wey & Godalming Navigation
4.3 ★★★★★ 7 reviews · Tourist attraction
Navigations Office and Dapdune Wharf, Guildford ...

Carphone Warehouse within Currys P...
Mobile Phone Shop
Stag House,, Woodbridge Rd, Guildford GU1 1EF

Showing results 1 - 10

(a) Photo of Google Map Search Providing Other Information

Figure 2.1: Google Map

.5 

cinema near me

Rating ▾ Clear filters

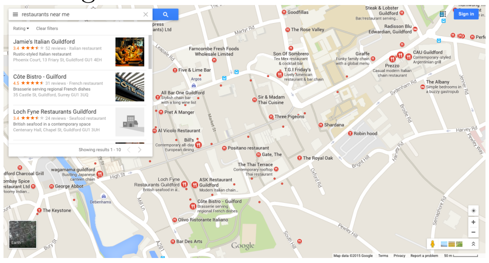
ODEON Guildford
E-commerce service
Chain cinema showing the latest releases
Bedford Rd, Guildford, Surrey GU1 4SJ

Odeon Cinema
3.0 ★★★☆☆ 22 reviews · Cinema
Chain cinema showing the latest releases
Bedford Rd, Guildford, Surrey GU1 4SJ

Ambassadors Cinema
3.7 ★★★★☆ 11 reviews · Cinema
The Peacock Centre, Woking Shopping, Woking, S...

Showing results 1 - 3

Figure 2.2: cinema near me

.5 

restaurants near me

Rating ▾ Clear filters

Jamie's Italian Guildford
3.4 ★★★★☆ 52 reviews · Italian restaurant
Rustic-styled Italian restaurant
Phoenix Court, 13 Friary St, Guildford GU1 4EH

Côte Bistro - Guilford
4.5 ★★★★★ 31 reviews · French restaurant
Brasserie serving regional French dishes
35 Castle St, Guildford, Surrey GU1 3UQ

Loch Fyne Restaurants Guildford
3.4 ★★★★☆ 24 reviews · Seafood restaurant
British seafood in a contemporary space
Centenary Hall, Chapel St, Guildford GU1 3UH

Showing results 1 - 10

Figure 2.3: restaurants near me

Figure 2.4: Google Search Providing LBS Information

2.2 Social Media

On a basic level, Social Media should provide individuals' platforms to construct profiles for said platform, accumulate list of other users sharing a connection, and interact with and beyond their list of connections. (Connections on most of the popular applications today would be described as friends, followers, matches etc.) Different sites/applications vary in degree to the type of connections they may support based on their functionality (danah m. boyd & Ellison 2007).

Some applications might provide a user with list of connections mined through user's location. One of the ways it is done, as mentioned earlier would be to fetch user's position and produce list of users on said platform in user's chosen range, therefore helping their users get acquainted to new people. In other words - socialize or network. Considering networking services that act as described below – location-based, is not far fetched as services most offer on a basic level are chat rooms based on one's location. Now they may be push/pull based on the scenario, if while just browsing a web page I randomly come across a pop up web page that says, *Hey, here are pictures of men and women using our service in your area, Connect with them!* - Push LBS. 2nd Scenario: I may sign up for a dating site and allow my location to be used to search for potential love interests that reside in a radius of 10km – Pull LBS. Social Media Description:

InstaMessage: Instagram – a simple picture/video sharing application does not come with a chat room feature, apart from being able to like and comment on photos, is not really conducive for social interaction between registered users (Morris/ 2012). Therefore developers of InstaMessage have used Instagram's API to further its functionality for its users. "InstaMessage allows you to connect directly with nearby users or anyone you like on Instagram, privately!" Its features include:

- Chat with friends or any instagram user privately
- Browse profiles of anyone you are interested in and start chatting immediately
- Discover popular InstaMessage users automatically via a smart recommendation engine
- Sign in with your Instagram Account, No registration
- View nearby InstaMessage users and begin chatting
- Block List: Prohibit selected users from contacting you

(By n.d.)

Now, the features of concern to an LBS study from the above listed, are being able to view nearby InstaMessage and Instagram users and begin chatting. (Note: To be able to chat one must be registered to both applications.) The location-based chat room is very crucial as to why a person would register to InstaMessage, as said person already has an Instagram account. Figure 1.5

Tinder: This service categorized as a lifestyle application is described as "a fun way to connect with new and interesting people around you" Where you simply swipe right to like or left to pass on the people recommended by the Tinder application. If a person you liked then likes you back or vice versa, this is considered a match and therefore one can chat with matches, getting to know them on the application (Tin n.d.). One of the features that made it very easy to use was enabling a Facebook login, using Facebook's API, tinder's developers built an application that would fetch your Facebook profile including pictures, interest and Facebook friends that use Tinder. Conveniently producing a profile for the user on the Tinder Application. Figure 1.6

Find my Friends: This app solely designed to allow easy location of friends and family using either iPhone, iPad or iPod touch. Just installing app and inviting friends to share locations from contact list or entering their email addresses. When a friend accepts invitation on their

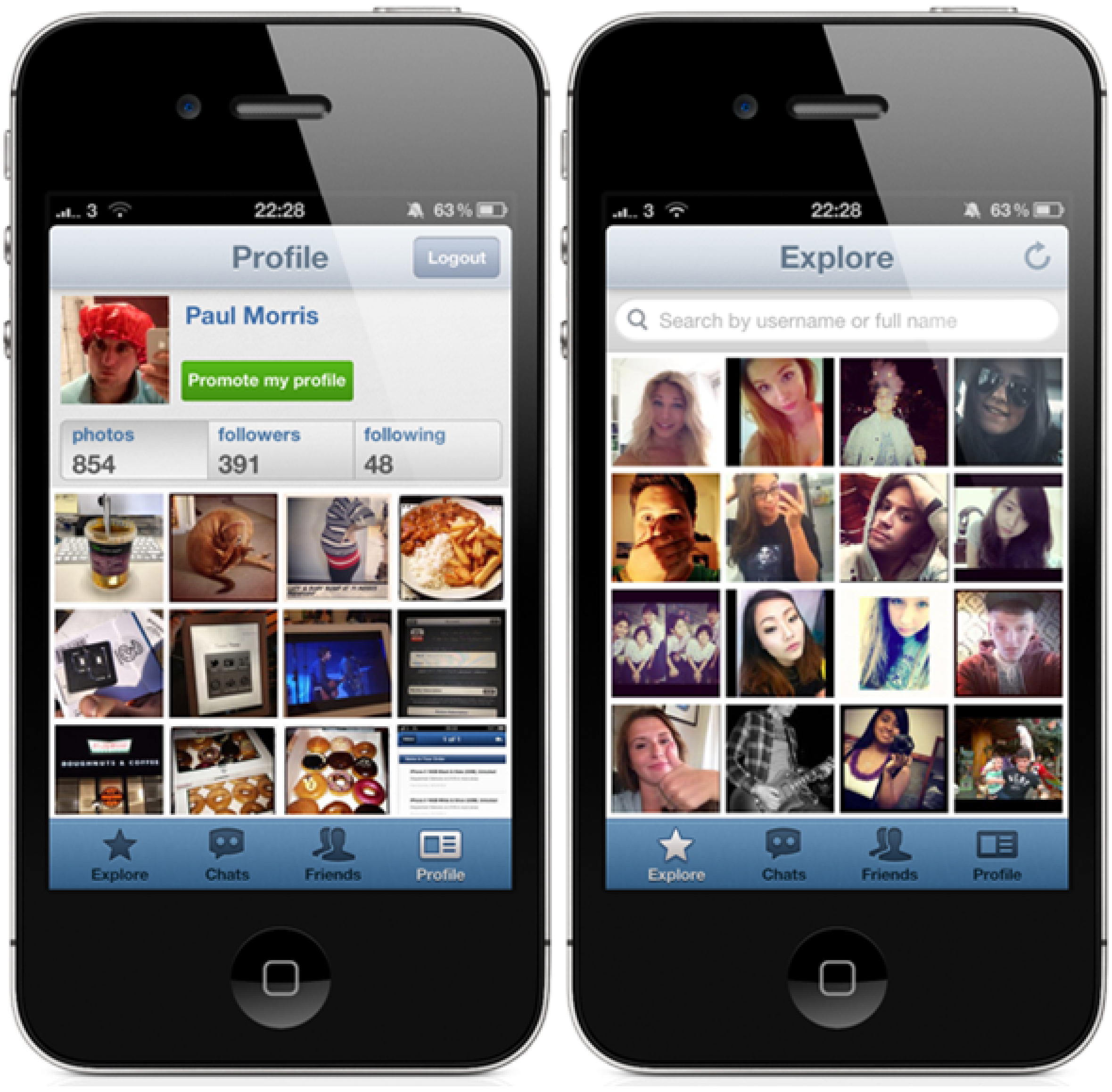


Figure 2.5: Instamessage on Mobile Device

device, using the app, one can begin following their information immediately and requests can be sent to follow their location. Location information can be hidden at anytime. It's application features allow setting up alerts to notify automatically when a friend is at the airport, a child leaves school, or family members home arrival. Alerts can also be set up to notify friends about location changes (Fin n.d.). Figure 1.7

Common thing about these applications being purely location-based, they have no API. This might be due to their small range of functionality. Other Applications with a larger number of the population may provide more developer features with very useful API's.

2.3 Social Media APIs

Services discussed in this section provide powerful A.P.I resource, which can be used in various ways, in terms of information transfer; Data can be fetched from a platform and spread across various platforms or in batches, mined for Big Data Analysis. Which means sometimes refine data to discover knowledge, or proving a hypothesis, and thereby developing a cycle that is used for commerce, advertising, customer service etc.

2.3.1 Popularly Used A.P.I. technologies

Facebook: The most widely known social media site, with its tag – either to login, like, share or all of the above exists on almost every blog site. The technology, which allows this – Its Application programming interface allows developers a whole variety of functions to utilize. Allowing Facebook login for iOS and Android Apps, and Websites. Facebook specifically recommends these best practices, which include:

- Ask for permissions in context – when users try to accomplish An action, trigger permission requests which is in context of particular permission. E.g. the Facebook app only asks for Location Services when people explicitly tap on the location button when updating their status.
- Include education to help someone understand how info from their Facebook Profile is used in your app E.g. when using the Facebook app, after tapping the Location button, a pre-permission dialog is displayed that clearly explains the benefit of turning on Location Services.
- Use Facebook Logins Button from their SDKs – The Login button that comes with Facebook's SDK's is easy to integrate it with and includes built in education that certifies a consistent Facebook Login experience for people across platforms.
- Testing Facebook Login flow – It is stressed how important it is to test Facebook's Login flow under a variety of conditions, built with a robust testing plan for a Developer to follow, available on their Developers web section.
- Submit app for Login Review if applicable – meaning, only apps requiring permissions beyond public profile, user friends and email. It is recommended; app is submitted for review

as early as possible in development lifecycle after Facebook Login has been integrated.

(Fac n.d.)

Twitter: An online social networking service that allows users to send and read 140-character messages called tweets. Its REST APIs provides programmatic access to read and write Twitter data. One can author a new tweet, read author profile and follower data, and more features. While to monitor or process tweets in real-time, A Streaming API exists for that purpose. This Streaming APIs allows developers at low latency, access to Twitter's global stream of Tweet Data. Twitter offers various streaming endpoints, with each customized to certain use cases. Public Streams: Where streams of the public data flowing through twitter, it is useful for following specific users or trending topics and data mining. User Streams: For Single-user streams, containing roughly all of the data corresponding with a single user's view of Twitter. Site Streams: This offers a multi-user version of user streams. Intended for servers, which must connect to twitter on behalf of several users. (Twi n.d.)

2.3.2 What are they presently used for ?

"Social networks are growing in popularity by the day, both for personal and business use, yet many IT and business executives remain wary of the risks posed by the online services and skeptical about potential benefits."

Though employers may not approve, an IDC survey of 4,710 U.S. workers shows 34% use consumer social networks like Facebook and LinkedIn for business, 9% employ micro blogging sites like Twitter for business as well (Brodkin 2010). On occasions one may sense everyone registered to Social Media is present with a planned agenda, it could be Political, Advertising, Friendship, Mentor, and Music/Business Promotion etc.

Facebook even provides a Marketing Tool to be used with its APIs, which provides access to its advertising platform. Its features include:

- Audience Management: Where creating and managing custom audiences and lookalike Audiences through its API. Data from CRM system, website events, or mobile app events can be used to target the right people.
- Ads Management: A complete suite of APIs can be built to create campaigns, manage

ads, and fetch metrics.

- Ads Insights: Provides API access for reporting and analytics purposes.
- Business Manager: This tool helps businesses and agencies manage their Facebook Pages, ad accounts and apps in one place.

It is very clear that these above features supersede the basic services of what social media would offer.

2.3.3 How do they contribute to the IOT?

Rising from the winter's flash flood earlier this year, the UK Government employed the services of Software developers and computer programmers who were provided access to 15-minute readings from every river level sensor in the UK. Among range of solutions raised was an app that would alert Twitter users to local volunteering opportunities (Neville & Allen 2014). This is possible due to twitter's development platform, Fabric, which includes their API; it helps developers build applications that can distribute information using Twitter accounts (Robles 2014). Bringing us to the Internet of things (IOT), which across industries is expected to produce intelligent objects that send and receive data to backend platforms. API management is response for a huge part of the interoperability, scale and control for these different things. Design standards for Web APIs, its management and a RESTful architecture simplify the task of interoperability across these various systems (GIS for example) that handle immense data.

The flood's called for a better early warning system, which resulted in connecting about 2.8 to 3 thousand river sensors to twitter. Now serving as a location-based application, a user only needs but subscribe on Twitter (follow user accounts) to the river points that might affect said user. Or subscribe to as many areas where loved ones reside so as to warn them in cases of similar events (Curtis 2014).

2.4 Privacy challenges of Location-Based Services

In the era of the smart phone, still Security with mobile devices are not new problems, these problems that need to be addressed can also be found in any device that provide connectivity to the Internet, there's risk when :

- (1) Managing assets stored in a device
- (2) Communication within trusted and non-trusted environments (this includes issues regarding privacy)
- (3) Interaction with critical IT infrastructures e.g Banks

Claims based on statistics show that millions of devices containing sensitive data might be misplaced into the wrong hands due to the best known practices for mobile security been simply insufficient. Strong user authentication both on client and server side is required to protect user from identity theft. Other problems that can arise include viruses(malicious software), attacks through Bluetooth, WLAN and GPRS - all these pose a threat that an average user may or may not be aware of. It is believed individuals are not too concerned about their privacy online(Hutter & Ullmann 2005), one will quickly sign up their real information including their present location to a site so as to make use of a service which might do home deliveries, book references etc. Still, Interface designers are expected to design systems that are privacy-sensitive due to public concern about privacy on the Internet, a study on 'Personal Privacy through Understanding' produces 5 pitfalls to help designers remember to build systems that helps users' understanding of a system's privacy implication (Lederer, Hong, Dey & Landay 2004). In Understanding:

- "Obscuring potential information flow" : the nature and extent of a system's potential for disclosure should not be obscured by design. Users should understand what the system's privacy implies so as to make a knowledgeable use of the system.
- "Obscuring actual information flow" : Users should understand any bit of information that is being disclosed to a third party, designs should not conceal the actual disclosure of information through a system.

In taking Action:

- "Emphasizing configuration over action" : Users should be enabled to practice privacy with the system as a natural result to engaging with the system. Designs should not be too complicated to an extent where excessive configuration is required to manage privacy.
- "Lacking coarse-grained control" : Top-level mechanism for halting to resume disclosure should be forgone when it is obvious disclosure is required.

- "Inhibiting established practice" : Designs should allow established social practice users are use to, be transferred to emerging technologies.

(Lederer et al. 2004).

2.5 Stakeholders

How does this ecosystem affect them individually? API providers: Companies like Facebook and Twitter, government bodies, organizations with valuable information as a resource. Been able to collect, host and provide this data requires responsibility and trust from the community.
Content Consumers/Users: Everyday people whose tweets and Facebook posts might be analyzed. They are made aware what their public data could be employed for via Terms and Conditions.
Mobile/Web Developers: Developers who directly fetch data from API resource, they could profit from building and selling these applications which might offer the same services as the provider, but might employ different interfaces.
Data Scientists: Data can be refined by employing machine learning algorithms to provide useful information for learning about a population to cure diseases or understand finances.
Advertisers: Using API technology as a valuable resource to marketing, hence providing them platforms where their marketing algorithms/machines could operate on with little interaction.
Government: Can always utilize any information that's free and available for the society good's by connecting them to other applications that easily educate or update all age groups on any necessary communal development.
Start-up Companies: They benefit of existing API providers allowing access to their data, by which they make it easier as they integrate with Facebook/Twitter. Therefore sometimes help increase usability of existing application while bringing attention to the unique services they offer.

2.6 Related Work of Year 2003

One of the questions being asked is, with the simple architecture of the Internet today, is it able to handle Web-distributed applications? These applications would require communication, cooperation and coordination. Web architectures would need to be improved without completely

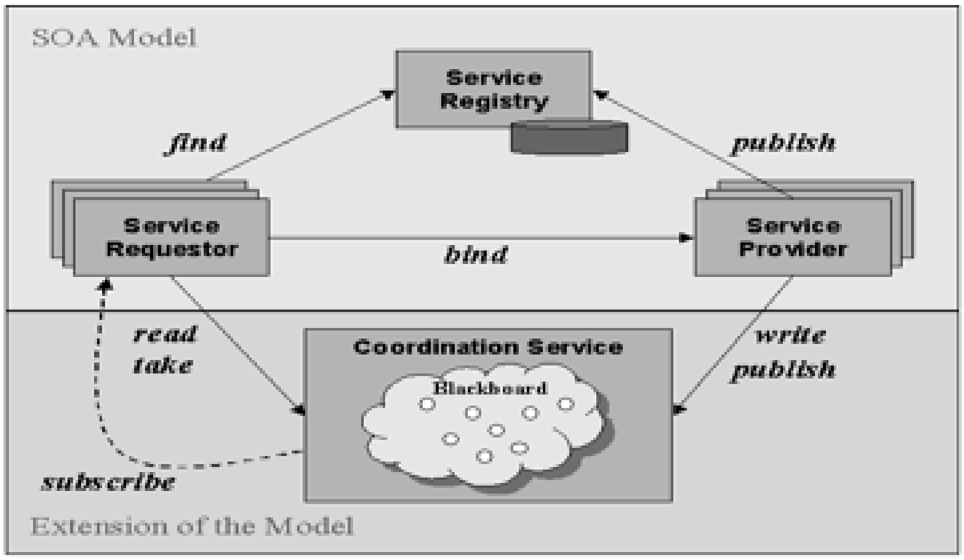


Figure 2.6: Extended SOA model

changing their original capabilities. [Ref]
A study that proposes an architectural pattern to support more flexible connections between these applications is primarily of interest. Inspired by the Blackboard architectural pattern, it is an extension of the Service-oriented Architecture. Where role parts that were added to the model are to communicate, synchronize requester and provider services. This is called a Service Coordinator.

Benefits include: Allows distributed services to collaborate distributively in time. Since writing services have no prior knowledge about its readers and vice versa, the collaboration can be described as uncoupled. Especially on the Internet where the different entities might posses little information about each other, it suits that environment adequately enough. Allows adaptable modeling of interfaces among services in dynamic environments. Supporting several data consumers with asynchronous and responsive interactions.

2.7 Review of Systems Built on API Technology

It is easy for one to come to a conclusion that building a business that strictly relies on an API is a bad idea, as a business has no control over the underlying foundation of its business. This may be true in some cases, but businesses like Kayak, which is a travel search company offering travel suppliers, on line travel agencies etc. Provides travel buyers 1-stop research solution to find cheapest fare, travel management tools and push location-based services like flight status updates, p ricing alerts and itinerary management. Kayak makes 55% of its revenue by offering advertising placement on its websites to travel suppliers and vendors, it costs their ad space buyers either cost per click or cost per impression while 45% of their revenue comes from distribution. (Team 2013) Kayak is assumed to make use of a solution from ITA Software, ITA provides QPX Express API - it is suited for entities that want to provide airfare pricing with shopping service for their customers with a self-service, pay as you go model.Designed to easily integrate with other business and available as a standard API on Google Developers. (Goia 2011)

Also Buffer is another business with its business model based on several API's, as it helps its user manage when to make a posting to social media page. It began aiming to schedule Twitter updates for smarter sharing, now increased its features to assist social media presence of its user across many networks. With its features including: Pre-set date and time to post to social media, which user can always adjust. Without opening the app, a user can add information to the Buffer queue from anywhere he/she is reading. Platforms it currently supports are

- Facebook: Profile, Page and Group
- LinkedIn: Profile and Company Page
- Twitter: Profile
- Google+: Business Pages
- Pinterest accounts (On Awesome and Business)

(Buf n.d.)

Chapter 3

System Analysis

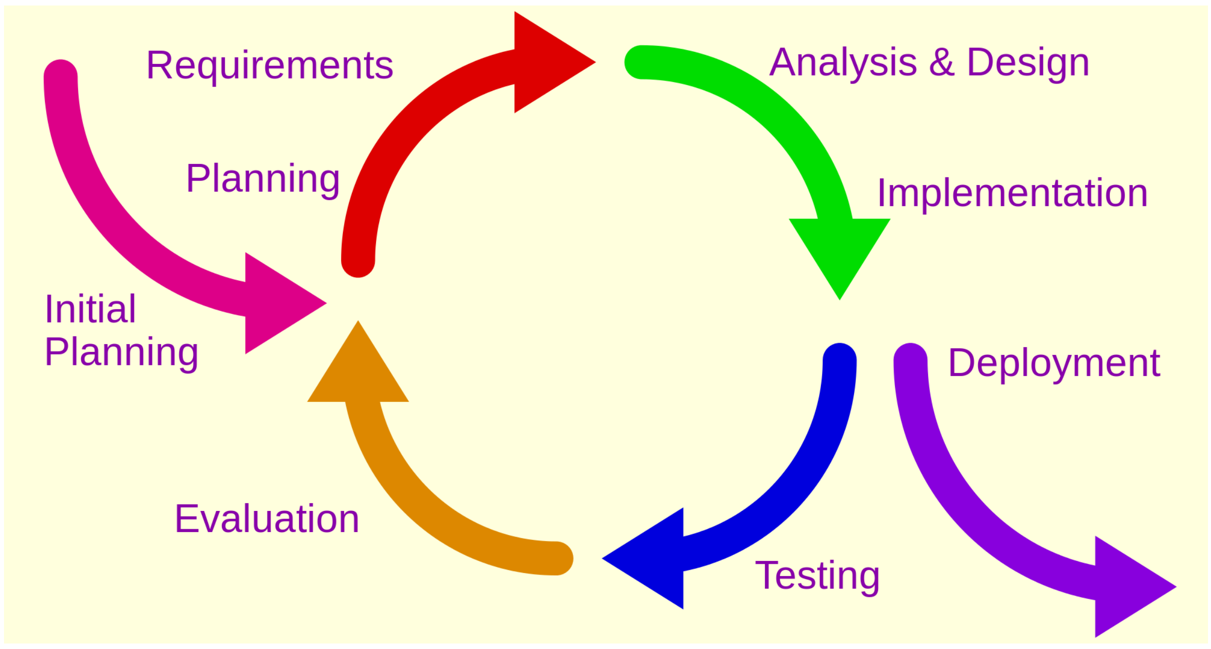
3.1 System Development Methodology

The development process employed in this study is in iterative fashion, planning, analyzing, designing and implementing the system in cycles. Producing functionality that was user focused, testable but also left room for advancement in terms of performance. An iterative process was required due to project's experimentation of technologies that were used to tackle problem. This problem which includes system requirements, project limitations, and a wide range of success measure; System Development proceeded with no prerequisite set of requirements except an objective to demonstrate API Integration for a location-based System. Requirements were raised and when necessary changed based on knowledge of what technologies allow to be easily integrated. Prototypes were presented for critique and then features developed to modern computing standards.

3.2 System Requirements

3.2.1 Functional Requirements

- The system should automatically fetch data from a Location-based API service
- The system should produce results for user to view
- The system should allow user log in with Facebook account
- The system should allow user to share events to Social Media



(a) Iterative

Figure 3.1: An example figure, with two parts

- The system should allow user to like an event
- The system should allow logged in user to comment on page

3.2.2 Non-Functional Requirements

- The system should offer a persistent front end
- The system should give user some options to select to produce results
- The system should allow user click through events without refreshing page

3.3 Analysis and Design Choices

During this phase, in picking the Events API, other API's like soundcloud and bandcamp were considered. The Events API provided easy access to the data using callbacks, other methods of extracting their data was well documented in the Events API Documentation. These other languages were considered since they allowed different ways to access their data. The method carried out in the system implementation is simply by choice due to factors of ease and comfort with the language and frameworks utilized.

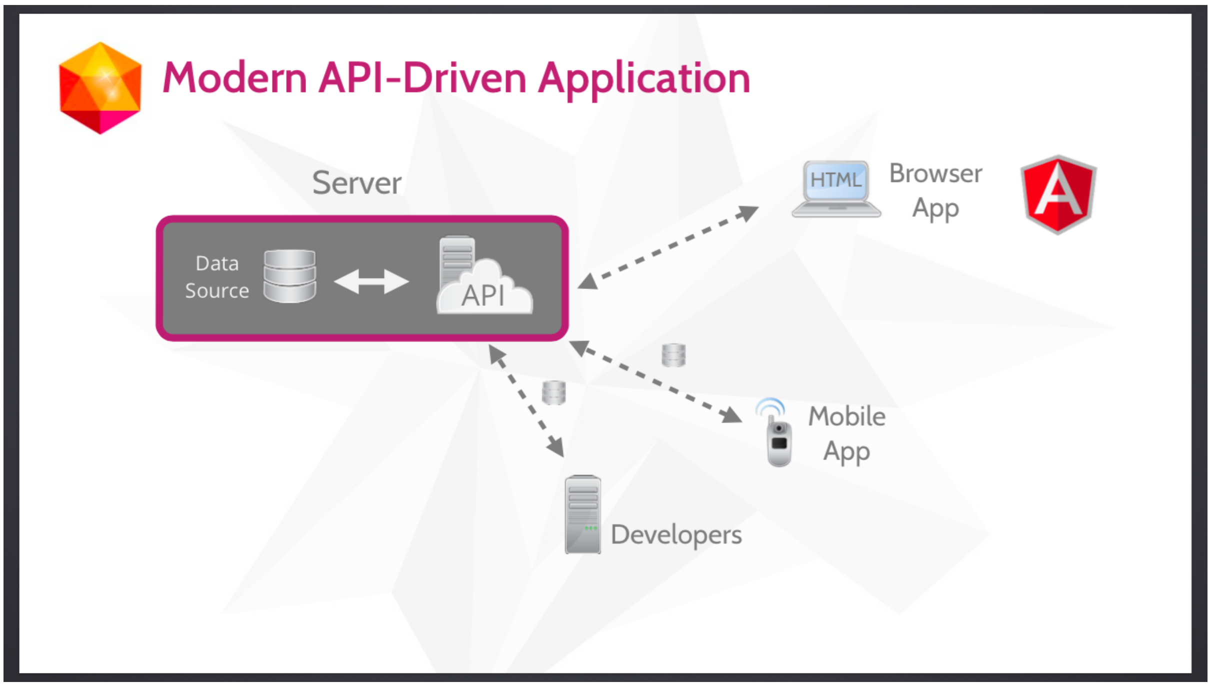
3.3.1 Frameworks/Libraries

Javascript was the underlying language in which these frameworks/libraries where utilized to build functionality. Node.js : An asynchronous event driven framework, designed to allow scalable network application. It is very useful for building server-side applications, a simple server was written called tester.js where, several connections could be handled concurrently due to the *call back* nature of the server. On every connection, a callback is launched, and when they are no calls the node is asleep. "HTTP is a first class citizen in Node, designed with streaming and low latency in mind", This makes it concrete foundation to also utilize other libraries/frameworks.

```
var static = require('node-static');
var http = require('http');

var port = 8080;

var file = new static.Server('./public');
\\Serving up the public folder that holds index, style sheets, scripts etc.

var app = http.createServer(function(request, response) {
  file.serve(request, response, function(error, result) {
    if (error) {
      console.error('Error serving ' + request.url + ' - ' + error.message);
      response.writeHead(error.status, error.headers);
      response.end();
    } else {
      console.log(request.url + ' - ' + result.message);
    }
  });
}).listen(port);

console.log('node-static running at http://localhost:%d', port)
\\{tester.js}
```

Angular.js : Useful for writing client-side web applications, where HTML is used as the template language, extending its syntax so as to define applications components.Automatically synchronizing view's data with the JavaScript object(model). This is done using its 2-way data binding, it helps server-side communication by controlling the *asynchronous callbacks* using promises and deferrals.

Jquery : This is a JavaScript library, and like Angular it makes HTML document traversal and manipulation, event handling, animation and Ajax easier to do. It provides an easy API to use that works across several browsers. It combines extensibility and versatility, making it very



(a) Modern Design

Figure 3.2: Illustration of a modern API-Driven Application, with the Angular Framework used on the Web application

popular among JavaScript Programmers.

Bootstrap: A front-end framework for faster and easier web development, allows developers to focus on the back end of things while the front end is controlled by marking up HTML files with its syntax.

Concert and Events API : This provides a collection of events, taking place in various local markets around the world, from concerts, sports to singles events and political rallies. It offers a platform that allows a developer with an approved application key to leverage it's data, features and functionality.

3.3.2 Execution Flow

System would be demonstrated below in the various execution contexts in which the application's core functionality works:

```
\$(function() {if (navigator.geolocation) {
    // geolocation is supported
    navigator.geolocation.getCurrentPosition(success_handler, failure_handler);
  } else {
    console.log("Geolocation is not supported");
  }
})
```

Here, application checks if browser supports geolocation

```
var latlon = {};
var output = "<ul>";
var checker; //to check if API request was successful
var display;
var collection = {};
var pgsize;
```

Some Global Variables are Instantiated, that would be useful to individual functions in Application

```
function success_handler(position) {
  latlon.latitude = position.coords.latitude;
  latlon.longitude = position.coords.longitude;
  \$("<p>Using latitude&longitude: " + latlon.latitude + "," + latlon.longitude + "</p>").app
  event();
}
```

This fetches the user's location by position(latitude
longitude)

function refresher() event(); Triggers Event Refresh

```
function event()
{
var categoryid = document.getElementById("category").value;
var venue = document.getElementById("venue").value;
var within = document.getElementById("points").value;
pgsize = document.getElementById("eventsno").value;
var oArgs = {
  app_key: "{app_key}",
  where: encodeURI(String(latlon.latitude)+","+String(latlon.longitude)),
  within: String(within),
  category: String(categoryid),
  venue_name: String(venue),
  page_size: String(pgsize)
};
\\Arguements to make Events API request

checker = EVDB.API.call("/events/search", oArgs, function(oData)
\\Request Call back and store immediate response
{
    collection = oData.events.event;
\$.each(collection, function(index, value) {
\$("#venue").append("<option>"+ (value.venue_name) + "</option>");
});

event_handler();
if (checker == true)
{
console.log('There is Data, Submit to Start ANGULAR!!!!');
}
else
{
console.log('Sorry theres problem with API')
}

});
}
```

This triggers API request function

```
angular.module('eventsApp', [])
  .controller('MainCtrl', function(\$scope) {
  \$scope.submitData = function(){
  var self = this;
  var i = 0;
  var a = [];
  var j = 0;
  var phide;
  var nhide;
  self.events = collection;

  console.log(self.events);
  self.event = self.events[i];
  self.prevEvent = function() {
  self.event = self.events[--i];
  };
  self.nextEvent = function() {
  self.event = self.events[++i];
  if (collection.length === self.events[i])
  {
  nhide = true;
  }
  };

  while (0 === i - 1)
  {
  phide = true;
  }
  };
  });
```

Angular Controller that allows Client Interaction with Results from API request

3.3.3 Design Challenge

Asynchronous(ay-SIHN-kro-nuhs) : is Greek for "not with time", this describes objects/events that are not coordinated in time. A design issue encountered; When the Angular Controller expects Global Variable 'Collection' to store an object of the requested data, this occurs when the submit button is clicked

```
\$scope.submitData = function()
```

Due to the system automatically making callbacks which uses the default values for the search variables, it fetches data which is not displayed until the user clicks the submit button.This request to the Events Data-house,

```
EVDB.API.call("/events/search", oArgs, function(oData)
```

may execute only after all the remaining synchronous code in their respective code blocks have been executed. This is an interesting phenomenon as one would expect code written in chronological order to execute as such, but this occurs like so because the function is not executed immediately, it is being passed as an argument. That is the reason why this type of request is called a `callback`. Therefore callbacks are functions that execute asynchronously, programs like this may execute different functions at different times based on order and speed that earlier functions - http requests or file system reads occur. (GitHub)

When dealing with asynchronous programming where only way to receive a callback would be to remain in that execution context, it becomes complex when another technology needs to be coupled, storing expected content in global variables seems to be the most likely solution.

Variable	Description
latlon	user's latitude and longitude
output	list values using html <ul> tag
checker	boolean value for API request success
collection	object that stores content requested from API

Table 3.1: List of variables that help fetch content for client side

Variable	Description
categoryid	user's desired category for events heightvenue
user's preferred venue heightwithin	proximity of desired events from user heightpgsize
number of events user would like to view	

Table 3.2: Options for user to choose that display content

3.4 User Interface - justification and answer problem of such little scope to each variable

3.4.1 Variables User Options

Tables are in Table 3.2.

3.4.2 Responsiveness of Site

Bootstrap used as the framework for developing a responsive application, made the application conform when browser size is reduced, application would respond properly if accessed through a tablet.

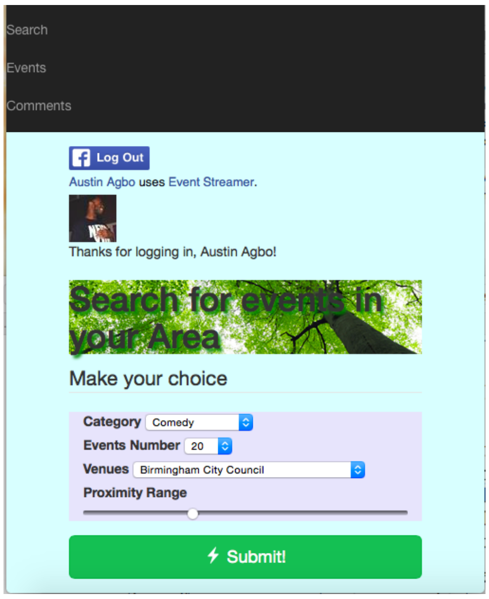
this system was built to also be responsive when viewed on a mobile, tablet and different browsers

3.4.3 Styling

A layout CSS was written to set margins of the body of the HTML file , styling the background of the header file with a controlled image. Jumbotron is a character of the bootstrap framework which was used to give the application a rigid feel.

```
body {
    background-color: #e0ffff;
margin-top: 2%;
margin-bottom: 2%;
```

.5



Austin Agbo uses Event Streamer.

Thanks for logging in, Austin Agbo!

Search for events in your Area

Make your choice

Category: Comedy

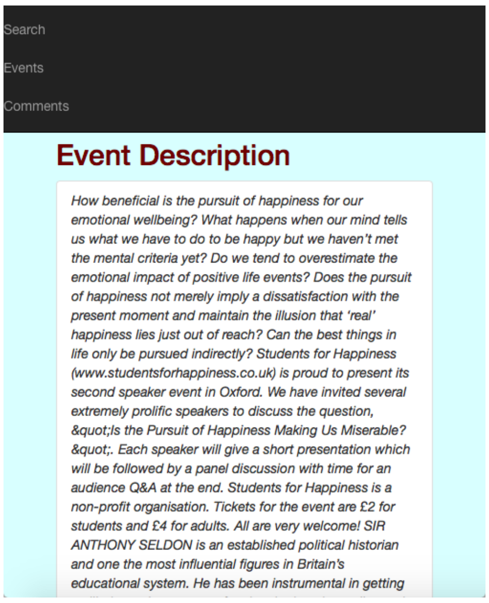
Events Number: 20

Venues: Birmingham City Council

Proximity Range

Figure 3.3: Compressed browser

.5



Event Description

How beneficial is the pursuit of happiness for our emotional wellbeing? What happens when our mind tells us what we have to do to be happy but we haven't met the mental criteria yet? Do we tend to overestimate the emotional impact of positive life events? Does the pursuit of happiness not merely imply a dissatisfaction with the present moment and maintain the illusion that 'real' happiness lies just out of reach? Can the best things in life only be pursued indirectly? Students for Happiness (www.studentsforhappiness.co.uk) is proud to present its second speaker event in Oxford. We have invited several extremely prolific speakers to discuss the question, &quot;Is the Pursuit of Happiness Making Us Miserable? &quot;. Each speaker will give a short presentation which will be followed by a panel discussion with time for an audience Q&A at the end. Students for Happiness is a non-profit organisation. Tickets for the event are £2 for students and £4 for adults. All are very welcome! SIR ANTHONY SELDON is an established political historian and one the most influential figures in Britain's educational system. He has been instrumental in getting

Figure 3.4: Compressed browser showing Event Description

```
margin-right: 5%;
margin-left: 5%;
}
h1 {
    background: url('/images/environment.jpg') repeat center center; width:1000;
text-shadow: green 0.1em 0.1em 0.1em;
}
h2 {
    color: rgb(100,0,0);
}

.jumbotron{
margin-right: 6%;
margin-left: 6%;
background:transparent;
}
```

Chapter 4

Software Documentation and Practice

Here we attempt to demonstrate the objectives of the project, focusing on two major types of documentation being the process of the project, and user requirements to utilize application and system implication.

4.1 Process Documentation

This section consists of documents produced during system development. A planned schedule, process quality, organizational and project standards (Sommerville n.d.). This was essential for management of the project phase by phase. Planned schedule was described with the Gantt chart below:

Project Plan:This project plan, consists of Research into Literary texts, API resources, frameworks, implementation and deployment of application.

During implementation, parts of the systems(chunks of code) were tested to provide functions that if executing with no errors and at least performing at an expected rate of efficiency. The process of the project was simple since it did not involve several individuals, steps in implemen-

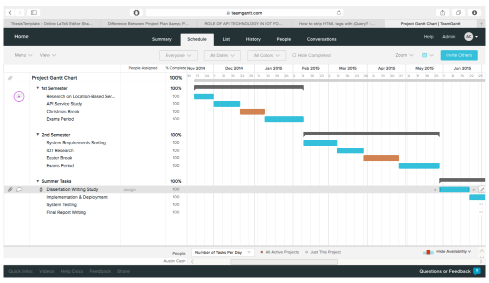
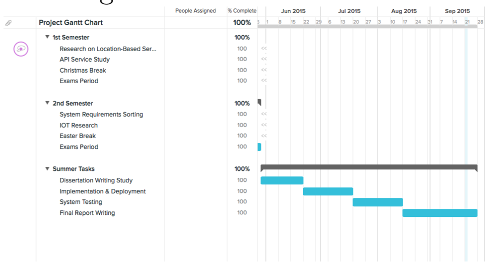
.5 

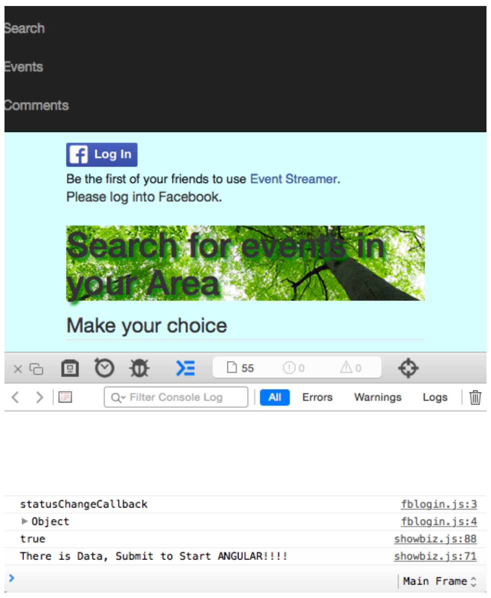
Figure 4.1: Process Plan



Project Gantt Chart	People Assigned	% Complete
		100%
▼ 1st Semester		100%
Research on Location-Based Ser...		100
API Service Study		100
Christmas Break		100
Exams Period		100
▼ 2nd Semester		100%
System Requirements Sorting		100
IOT Research		100
Easter Break		100
Exams Period		100
▼ Summer Tasks		100%
Dissertation Writing Study		100
Implementation & Deployment		100
System Testing		100
Final Report Writing		100

.5

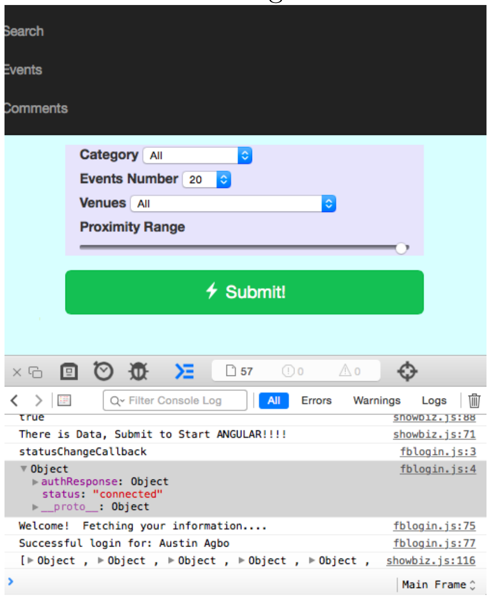
Figure 4.2: Process Plan


Search
Events
Comments
Log In
Be the first of your friends to use Event Streamer.
Please log into Facebook.
Search for events in your Area
Make your choice
55
0
0
Filter Console Log
All
Errors
Warnings
Logs
Main Frame

```
statusChangeCallback                                  fblogin.js:3
▸Object                                               fblogin.js:4
true                                                  showbiz.js:88
There is Data, Submit to Start ANGULAR!!!!            showbiz.js:71
```

.5

Figure 4.3: 'True' is a result to API call being automated and successful on load up



```
true                                                  showbiz.js:88
There is Data, Submit to Start ANGULAR!!!!            showbiz.js:71
statusChangeCallback                                  fblogin.js:3
▾Object                                               fblogin.js:4
  ▸authResponse: Object
   status: "connected"
  ▸__proto__: Object
Welcome!  Fetching your information....               fblogin.js:75
Successful login for: Austin Agbo                     fblogin.js:77
[▸Object , ▸Object , ▸Object , ▸Object , ▸Object ,    showbiz.js:116
```

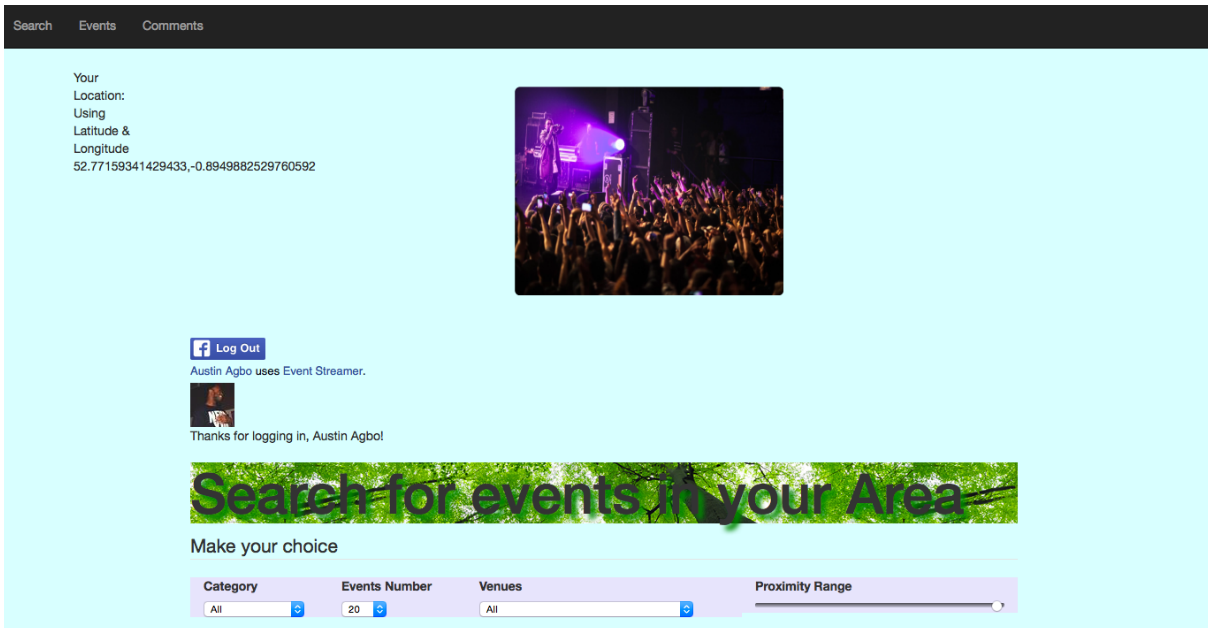
.5

Figure 4.4: Shows objects of events request by user made, fetched and log in to Facebook is authenticated

tation of application required approval from supervisor iteratively.

In terms of cost estimates, no costs pertaining to the project were acquired during its life cycle as the frameworks/libraries utilized are open source.

Testing: Apart from final usability testing, Debugging and testing of features where carried out by sending expected variable outputs to 'console.log()' so as to be sure correct results were produced.



Your Location: Using Latitude & Longitude 52.77159341429433,-0.8949882529760592

Austin Agbo uses Event Streamer.

Thanks for logging in, Austin Agbo!

Make your choice

Category: All

Events Number: 20

Venues: All

Proximity Range

(a) Load up page

Figure 4.5: On MacBook Air; It shows I am already logged in as a Facebook User, and Automatically fetches my location.

4.2 User Documentation

This is the part of the documentation, where users of the system would need to read and understand to make effective use of the system.It is been structured to serve for different user tasks depending on level of expertise. The 2 major parts are : For End-users and System Administrators.

4.2.1 End-users

You need to go under the hood taking pictures and showing how data pops up

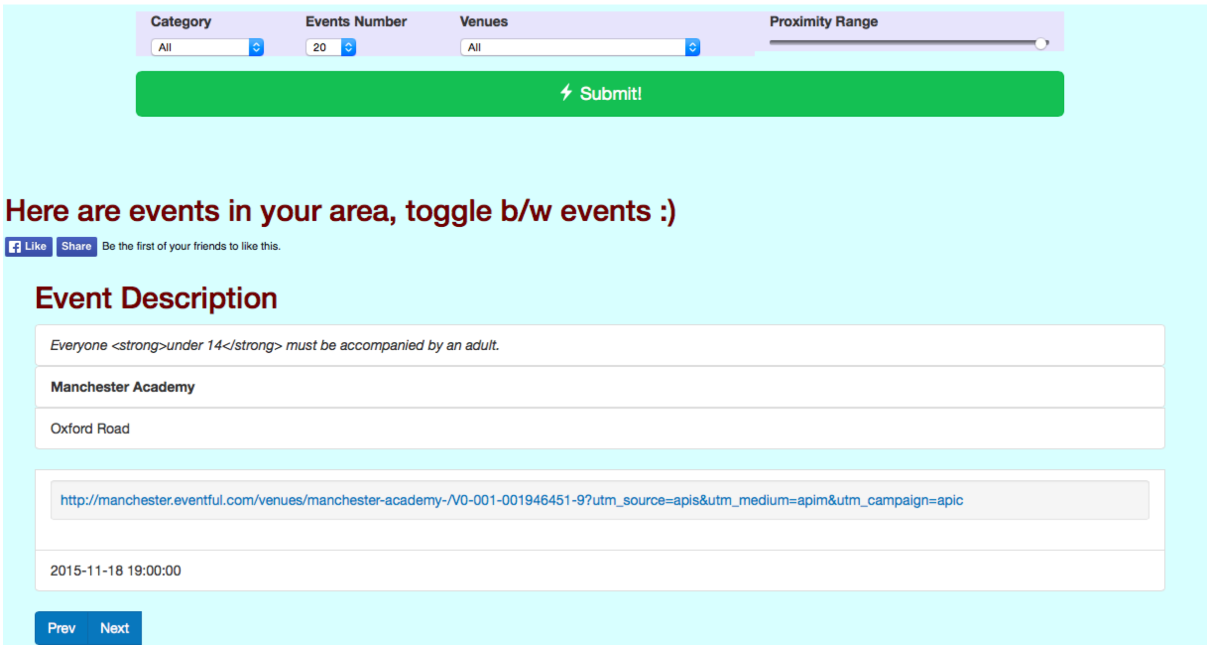
End-user : This software can be used to search for events where there is Internet connectivity, with a browser that fully supports Geo-location API. First browsers with full support include Google Chrome 5.0, Internet Explorer 9.0, Mozilla Firework 3.5, Safari 5.0, and Opera 16.0 (W3schools.com). User can login with already created Facebook account so as to allow sharing of events to the social media site. Application also scales to be utilized on mobile/tablet browsers. A demonstration of how to operate the on-line application is shown below:

First Step: If the application is completely loaded up, it displays like

Second Step: if you decide to click submit on the default options, meaning Category = All, Events Number = 20, Venues = All, Proximity Range = 100 (within location radius)

User can redefine Search by selecting options

Administrator : This application does not contain a special interface to monitor its services, but an administrator would need to be aware that deployment on any cloud or physical servers would entail installing all dependencies. This dependencies are specified in the application folder in the file package.json .



Here are events in your area, toggle b/w events :)

Event Description

Everyone <strong>under 14</strong> must be accompanied by an adult.

Manchester Academy

Oxford Road

http://manchester.eventful.com/venues/manchester-academy-/V0-001-001946451-9?utm_source=apis&utm_medium=apim&utm_campaign=apic

2015-11-18 19:00:00

(a) Load up page

Figure 4.6: Displaying the events based on default arguments, 20 events within 100 radius to click through

.5

Figure 4.7: drop down list of categories

.5

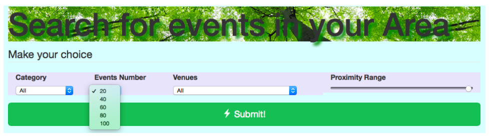


Figure 4.8: drop down fixed list of events number

.5

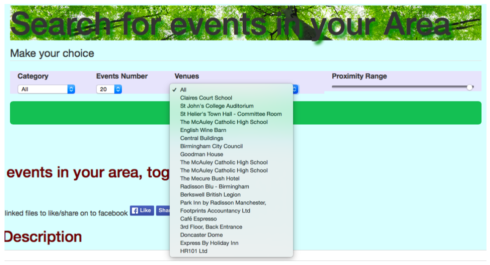
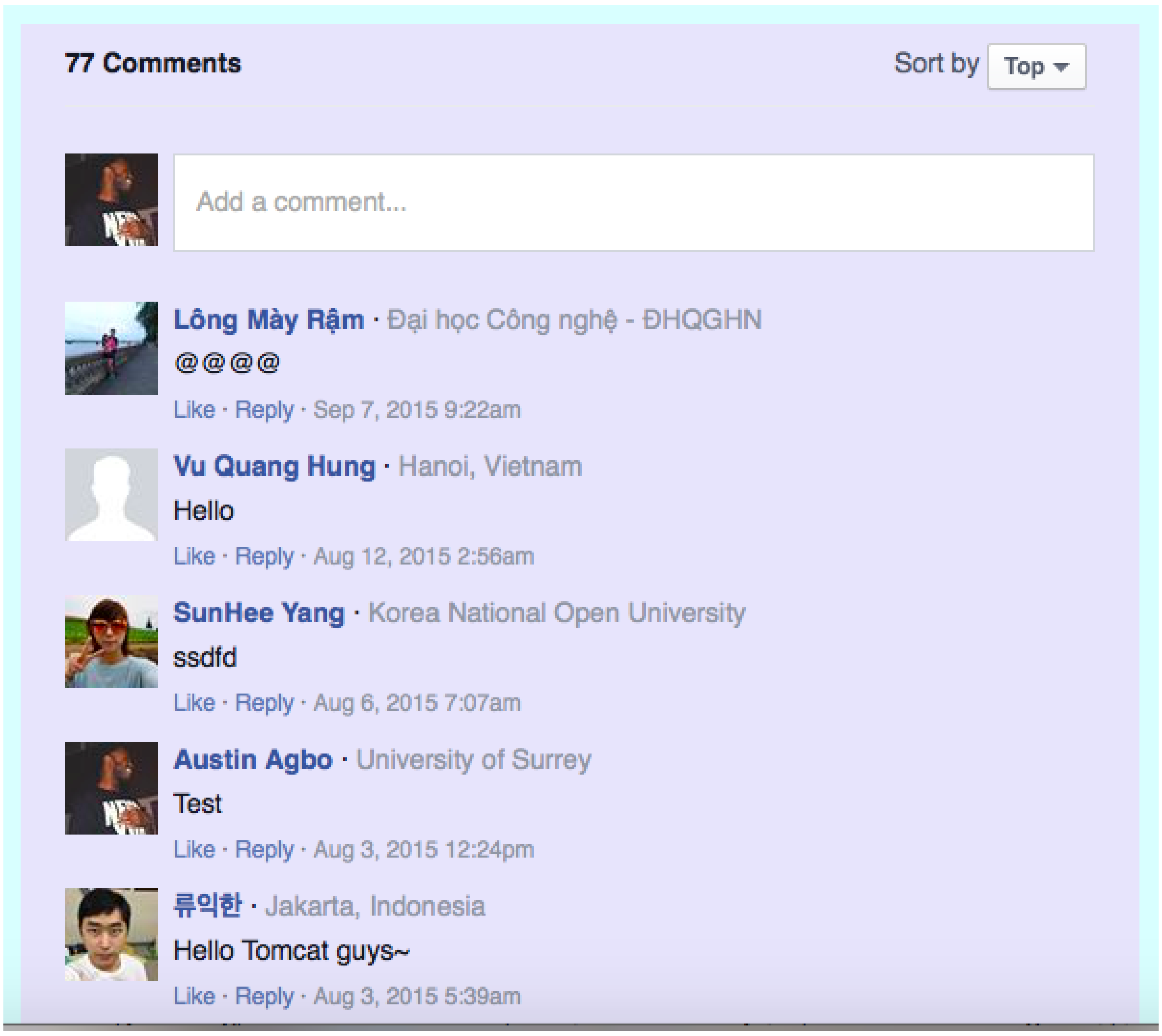


Figure 4.9: drop down list of venue list streamed from API


77 Comments Sort by Top

Add a comment...

Lông Mày Rậm · Đại học Công nghệ - ĐHQGHN
@@@@
Like · Reply · Sep 7, 2015 9:22am

Vu Quang Hung · Hanoi, Vietnam
Hello
Like · Reply · Aug 12, 2015 2:56am

SunHee Yang · Korea National Open University
ssdfd
Like · Reply · Aug 6, 2015 7:07am

Austin Agbo · University of Surrey
Test
Like · Reply · Aug 3, 2015 12:24pm

류익한 · Jakarta, Indonesia
Hello Tomcat guys~
Like · Reply · Aug 3, 2015 5:39am

(a) Face book plug-in to comment

Figure 4.10: User can comment on page

.5

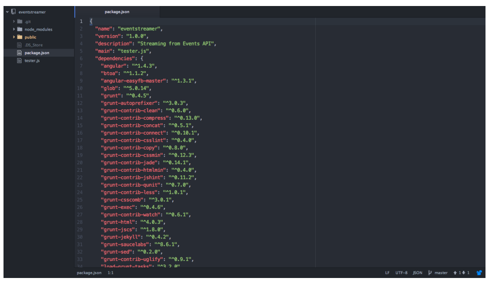
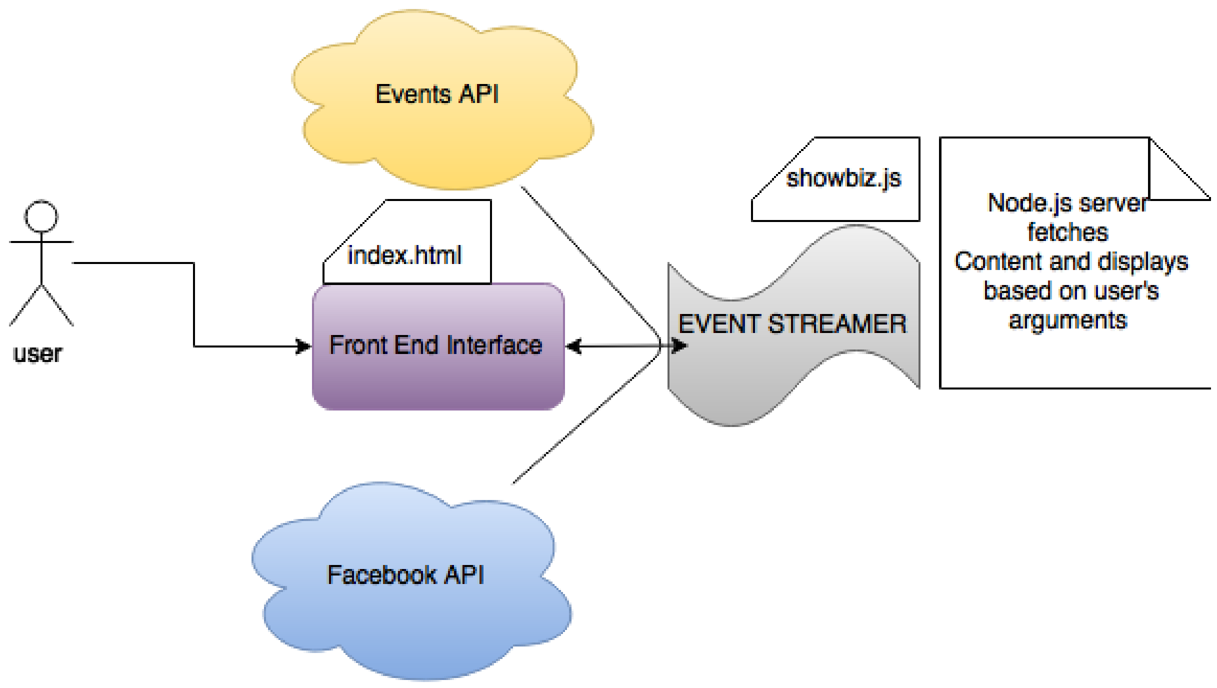

```
{
  "name": "eventstreamer",
  "version": "1.0.0",
  "description": "Streaming from Events API",
  "main": "tester.js",
  "dependencies": {
    "angular": "^1.4.3",
    "btoa": "^1.1.2",
    "angular-easyfb-master": "^1.3.1",
    "glob": "^5.0.14",
    "grunt": "^0.4.5",
    "grunt-autoprefixer": "^3.0.3",
    "grunt-contrib-clean": "^0.6.0",
    "grunt-contrib-compress": "^0.13.0",
    "grunt-contrib-concat": "^0.5.1",
    "grunt-contrib-connect": "^0.10.1",
    "grunt-contrib-csslint": "^0.4.0",
    "grunt-contrib-copy": "^0.8.0",
    "grunt-contrib-cssmin": "^0.12.3",
    "grunt-contrib-jade": "^0.14.1",
    "grunt-contrib-htmlmin": "^0.4.0",
    "grunt-contrib-jshint": "^0.11.2",
    "grunt-contrib-qunit": "^0.7.0",
    "grunt-contrib-less": "^1.0.1",
    "grunt-csscomb": "^3.0.1",
    "grunt-exec": "^0.4.6",
    "grunt-contrib-watch": "^0.6.1",
    "grunt-html": "^4.0.3",
    "grunt-jscs": "^1.8.0",
    "grunt-jekyll": "^0.4.2",
    "grunt-saucelabs": "^8.6.1",
    "grunt-sed": "^0.2.0",
    "grunt-contrib-uglify": "^0.9.1",
    "load-grunt-tasks": "^3.2.0"
```

Figure 4.11: First half of package.json

.5

.png .pdf .jpg .mps .jpeg .jbig2 .jb2 .PNG .PDF .JPG .JPEG .JBIG2 .JB2 .eps

Figure 4.12: Second half of package.json



(a) System's Architecture

Figure 4.13: User can comment on page

This package.json file also includes how to test the application, the command to run for testing is 'node tester'.

4.2.2 System Documentation

System Architecture and API Dependencies at its core:

To maintain the system, some considerations would need to be put forward and analysed.

4.2.3 Technical Maintenance

Disadvantage of employing one API service as providing the core functionality of the application is in cases where the Event API is not responsive or down due to server problems, thus system would face similar problems as well. For this application to be maintained and function properly, backup API services need to be on stand by so the system redirects to fetching its data from an alternative source that is not connected to the main source. Since the application has a Model-View-Controller Architecture that's all in one. Even though the application offers providable scalability as one of gains of node.js framework. When the application needs to be developed further into a large application with several developers involved and other complex functionality implemented, it might be of good service to split the parts - angular framework for the front end, a database that models user's transactions under an administrator, and node.js controller as the server.

4.2.4 Business Maintenance

Running a business on another entity's API, contracts/agreements need to be drawn to protect the interest of both parties, the ramifications in getting value from another business, it is only fair that if profits are made on the system's side, the provider should get a share for maintenance of the data provided.

4.3 Risk Assessment and Mitigation

For the system Administrator and business owner would have to consider the huge Risk in situations like this when data is corrupt, not flowing, or an unfair cost is unexpectedly placed on the cost. Risk of privacy issues are low for the API streamer due to no storage of information - Privacy issue mitigated by incorporating social media log in so risk is carried on by facebook or user who decides to log in so as to comment, share or like.

Chapter 5

Project Findings

5.1 Application Structure

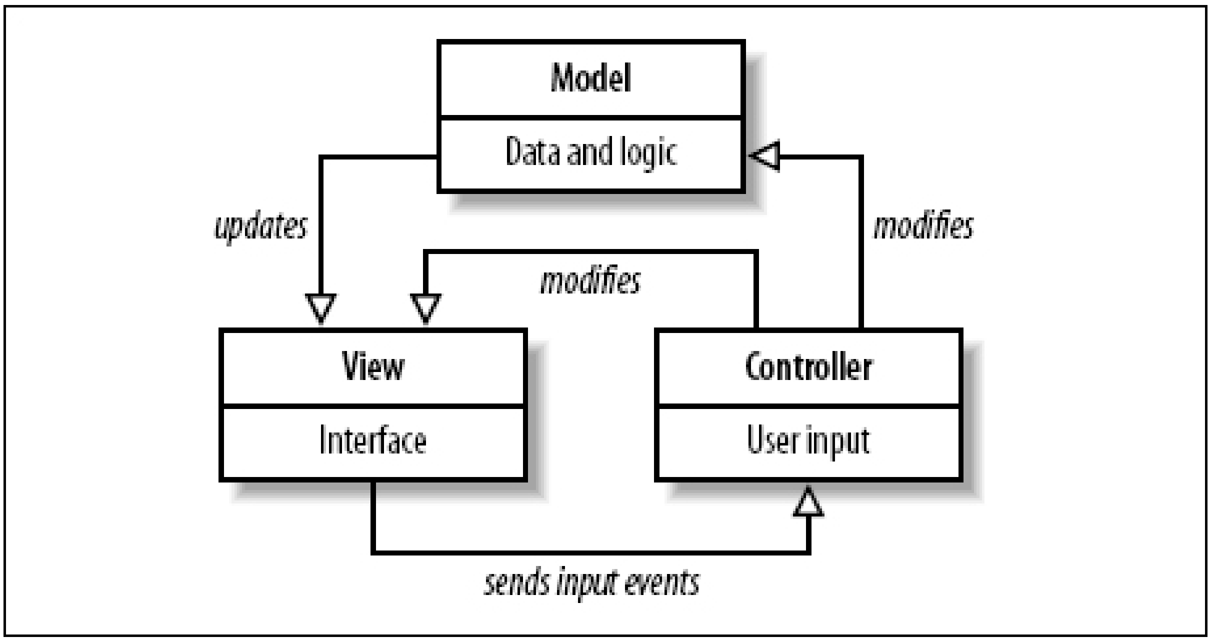
From the routinely practice of MVC, in almost every application, a benefit of this practice is making either of the components easily replaceable. (*model-view-controller* n.d.)

In building, an application around an API, it is safe to say it is not a full application by it self without been connected to the API's data banks. Though, this application, which is simply a user interface to utilizing some or all of the functions, the API provider's application gives access to. Utility value in a newly developed interface can be combing data from different API's or some sorts of refinery for data that proves to be uniquely valuable to its users . Looking at this application as a user interface, then breaking this down into MVC allows for a more controllable and work sharing development among developers for a complex system.The MVC which the application contains is described as so:

5.1.1 Model

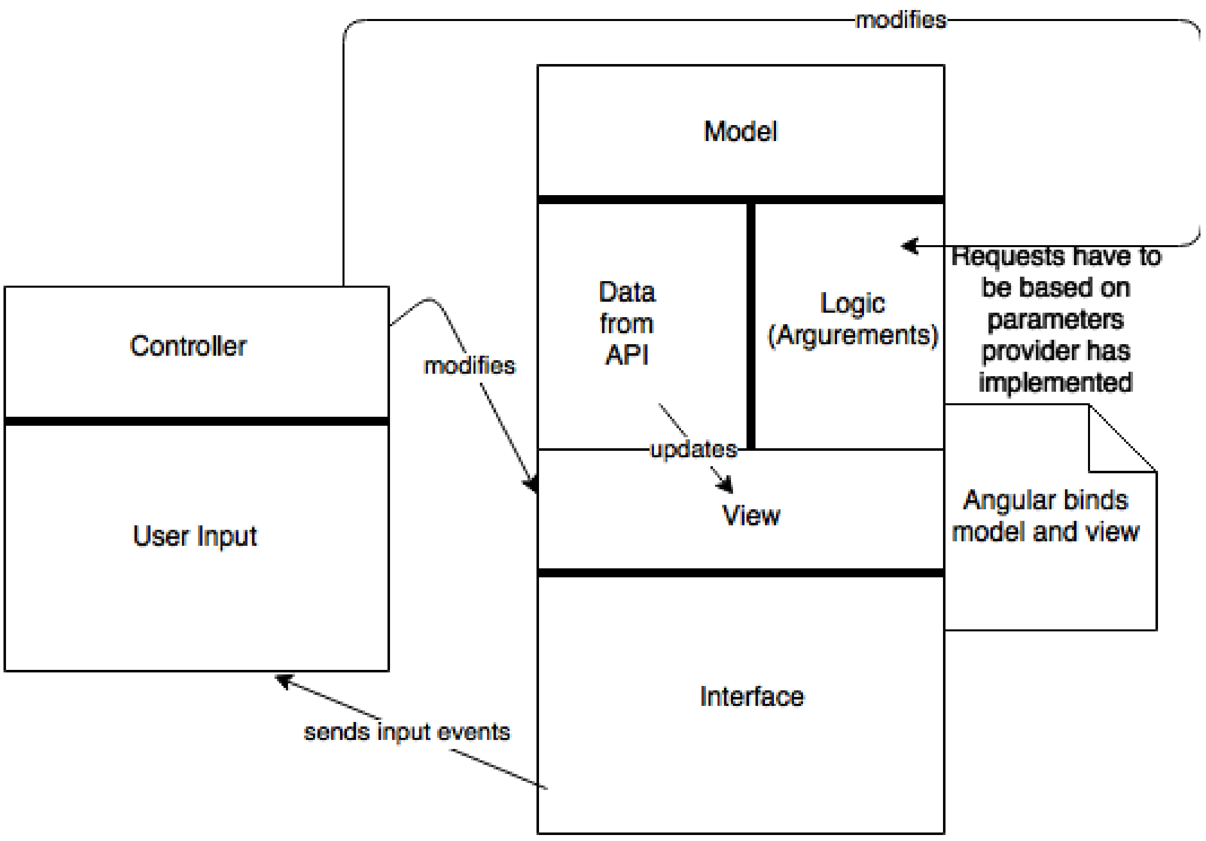
Not too many parameters of data where to be modeled for the user, so writing its code for a separate folder or file like other regular practices e.g ruby on rails would have been unnecessary, also a database was not useful as it might be considered redundant to create storage for data that is already stored, especially if it is accessible and reliable most of the times. This can be accounted for as, not once did the provider fail in it's API produce. Due to its consistency, creating variables to model the choices that were required to hold user's choices was easier and keeps the application efficient in serving information.

```
var categoryid = document.getElementById("category").value;
var venue = document.getElementById("venue").value;
var within = document.getElementById("points").value;
pgsize = document.getElementById("eventsno").value;
var oArgs = {
  app_key: "bPDvt9DqVLJPqVL8",
  where: encodeURI(String(latlon.latitude)+","+String(latlon.longitude)),
  within: String(within),
  category: String(categoryid),
  //performers: String(performers),
  venue_name: String(venue),
  page_size: String(pgsize)
};
```



(a) Normal MVC diagram

Figure 5.1: Model-View-Controller



(a) Revised MVC

Figure 5.2: Model-View-Controller

5.1.2 View

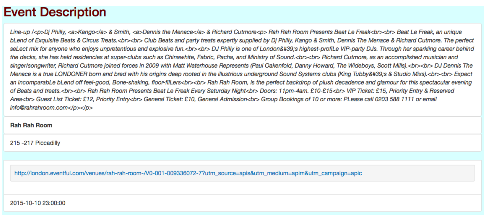
The view was plainly marked up and described by the HTML, Angular.js, Bootstrap syntax, this allowed for easy dom manipulation, CSS styling, and responsiveness. A graphic designer can carry out a lot more virtuoso by employing SVG libraries, Processing, Raphael, or CSS animations. This shows the HTML part that connects the angular controller to the view for manipulation.

```
<body ng-controller="MainCtrl" data-spy="scroll" data-target=".navbar" data-offset="50">
<div class="container">
<h2>Event Description</h2>
<ul class="list-group">
<em><li class="list-group-item" ng-bind = 'event.description'></li> </em>
<address>
<strong>
<li class="list-group-item" ng-bind = 'event.venue_name'></li>
</strong>
<li class="list-group-item" ng-bind = 'event.venue_address'></li>
</address>
<li class="list-group-item" ><div class="well well-sm"><a ng-bind = 'event.venue_url'></a></d
<li class="list-group-item" ng-bind = 'event.start_time'></li>
</ul>
<div class="btn-group">
<button class="btn btn-primary" ng-click="prevEvent()">Prev</button>
<button class="btn btn-primary" ng-click="nextEvent()">Next</button>
<div>
</div>
</body>
```

5.1.3 Controller

Note, we are still talking about a controller for a user interface, but for this application it does very little. To apply more refinery or further functionality to data that is being streamed, this controller functionality written in Angular would allow a whole load of implementation of decision making into it. This would be addressed more in the sections considering machine learning (5.5). For now it is simply used to implement a next/prev function for listed events.

```
angular.module('eventsApp', [])
  .controller('MainCtrl', function(/$scope) {
  /$scope.submitData = function(){
  var self = this;
  var i = 0;
  var a = [];
  var j = 0;
  var phide;
  var nhide;
  self.events = collection;

  console.log(self.events);
  self.event = self.events[i];
  self.prevEvent = function() {
```


Event Description

*Line-up /<p>Dj Philly, <a>Kango</a> & Smith, <a>Dennis the Menace</a> & Richard Cutmore<p> Rah Rah Room Presents Beat Le Freak

 Beat Le Freak, an unique bLend of Exquisite Beats & Circus Treats.

 Club Beats and party treats expertly supplied by Dj Philly, Kango & Smith, Dennis The Menace & Richard Cutmore. The perfect seLect mix for anyone who enjoys unpretentious and explosive fun.

 DJ Philly is one of London&#39;s highest-profiLe VIP-party DJs. Through her sparkling career behind the decks, she has held residencies at super-clubs such as Chinawhite, Fabric, Pacha, and Ministry of Sound.

 Richard Cutmore, as an accomplished musician and singer/songwriter, Richard Cutmore joined forces in 2009 with Matt Waterhouse Represents (Paul Oakenfold, Danny Howard, The Wideboys, Scott Mills).

 DJ Dennis The Menace is a true LONDONER born and bred with his origins deep rooted in the illustrious underground Sound Systems clubs (King Tubby&#39;s & Studio Mixs).

 Expect an incomparabLe bLend off feel-good, Bone-shaking, floor-filLers

 Rah Rah Room, is the perfect backdrop of plush decadence and glamour for this spectacular evening of Beats and treats.

 Rah Rah Room Presents Beat Le Freak Every Saturday Night
 Doors: 11pm-4am. £10-£15
 VIP Ticket: £15, Priority Entry & Reserved Area
 Guest List Ticket: £12, Priority Entry
 General Ticket: £10, General Admission
 Group Bookings of 10 or more: PLease call 0203 588 1111 or email info@rahrahroom.com</p></p>*

Rah Rah Room

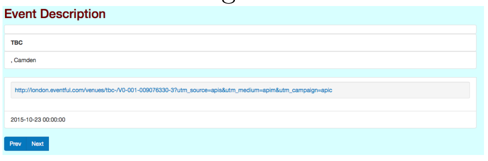
215 -217 Piccadilly

http://london.eventful.com/venues/rah-rah-room-/V0-001-009336072-7?utm_source=apis&utm_medium=apim&utm_campaign=apic

2015-10-10 23:00:00

.5

Figure 5.3: Data encoded with html tags which a user shouldn't be able to see


Event Description

TBC

, Camden

http://london.eventful.com/venues/tbc-/V0-001-009076330-3?utm_source=apis&utm_medium=apim&utm_campaign=apic

2015-10-23 00:00:00

.5

Figure 5.4: Description of event is missing

Figure 5.5: When data is data and missing!

```
self.event = self.events[--i];
};
self.nextEvent = function() {
self.event = self.events[++i];
};
};
});
```

5.2 Data dirt or loss in coupling Applications

Difficulties that can easily be noticed in fetching large data from an outside source, even by users of the application include dirty and/or loss of data. Dirty data can be defined as records in a database containing errors. It's causes range from duplicates, incomplete or outdated data(i.e. not updated), and bad parsing of record fields from separate systems. It should be also noted that TDWI has an estimate of dirty data costing over 600 billion U.S. dollars for U.S. businesses every year. There are several methods that can be implemented to fix data considered dirty but for data loss which is preventable but not fixable in our situation, defined as error conditions where information is destroyed by failures in storing, transmitting, or processing it's data (searchcrm.techtarget.com n.d.).

5.3 Test Study

Features that are implemented where built in iteratively but the final test phase was essential in critique of the application as a whole. Where, the two most important concepts considered were validating(bold) the correctness of the end product that is with respect to the user needs requirements then verifying(bold) by evaluating the system components to determine whether the end product of the development phase satisfies the conditions that were imposed at the beginning of that phase. (BS7925-1, 1998)

Test Scenario:
Been that any individual who can make use of a computer and is accustomed to going out for events should be able to use the application, it is not really as important the expertise of the

sample test users. So going by SUMI questionnaire based method to measure software quality from a user's perspective. This process was utilized in abstracting if the application satisfies the needs it was intended for. The sample test users each spent about 20 minutes using the software, though SUMI is intended for users with experience with the software to evaluate it. Assumptions were, users have made use of similar applications like the one been tested and since its a relatively small application in terms of its functionality, one doesn't need long time experience spent to evaluate if it would be useful to a person as the user(Validation). The overall feedback gained from the users is they believed the system needed a better User Interface, this view was shared by 90 percent of the users that were tested. Since measuring the level at which a developer can say, users would be pleased with the look and feel of the application is unquantifiable except by several users proving an opinion to be the most popular. To answer the last question in the questionnaire,a text box was provided for users to voice their opinion on what they thought needs improvement the most, for the 10 users their answers read like ("the event submission and results", "The overall design layout and information architecture needs to improve as it is difficult to interpret the content on the website", "Design", "Overall look and feel, to attract more users.", "nothing", "design", "background colour", "Interface", "Layout/presentation (general improvement and consistency) [needs to look more orderly, professional, attractive]... Also prompts, information, and the like... Though somewhat straightforward atm, you can begin to wonder exactly what certain changes have done or should do I.e. Category and event number changes, you think 'is it working as intended?' ", "Functionality and design"). Their responses need to be taken at face value, as a business it always proves profitable to keep the user satisfied in what they require of a system. More detail into questionnaires and evaluation of testing exists in chapter 7.

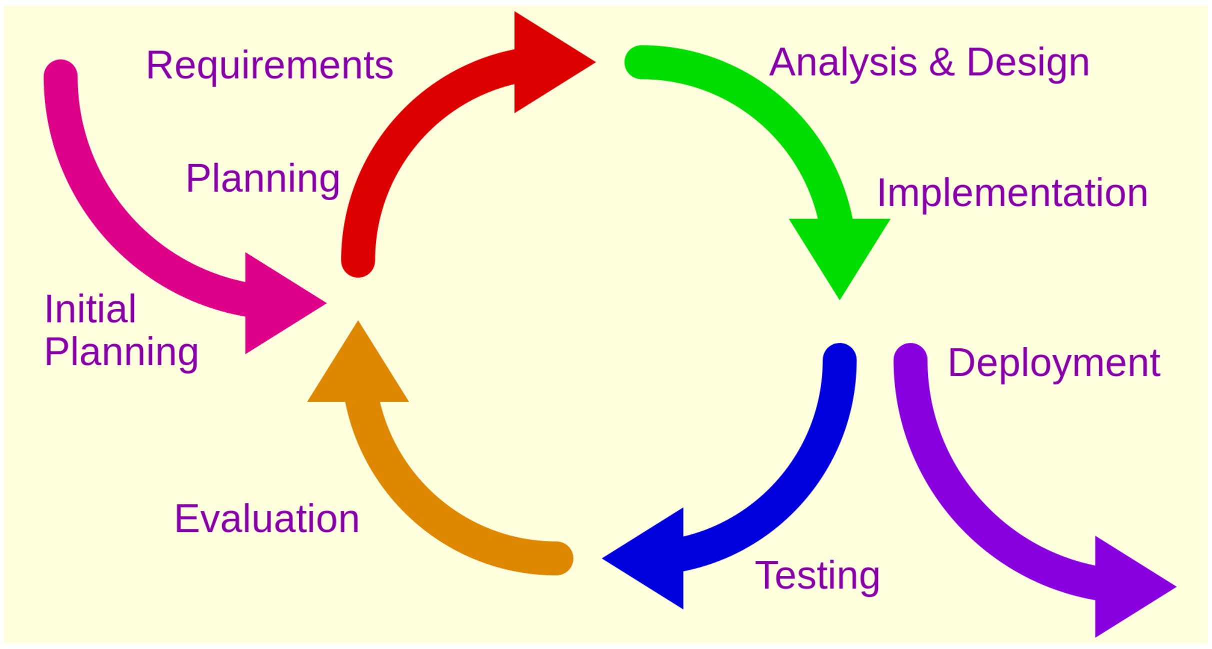
In respect to verification of the system, it is found that the list of requirements stated in the system analysis phase were met to a degree. Roughly about 77 percent of the required system was achieved. So the system automatically fetches data from a location-based API service, produces results for user to view, allows user log in with Facebook account, Allows logged in user to comment while offering a persistent front end, options to select, and allowing user click through events without refreshing the page.

5.4 Application in IOT

An afterthought of functionality to be implemented in the controller for future research purposes is machine learning. Being able to predict the parameters a user is likely to pick based on user's history would add substantial value to the application that will make it stand out in the market place of event notifiers.This might then imply some sorts of storage e.g. data cache. Optimising API technology so devices can update data in real time will really be pushing the ambitions of the project. E.g. a sensor notifying its followers on twitter when soil in parts of the earth is volatile for an earthquake to occur will really be powerful and useful technology. A connection to a 'physical thing' would have demonstrated the project's ambitions in a grand way.

5.5 Future use of API's in IOT

The Google Car, which while still in development is another prime example of the usefulness of being able to program into an Information Service. As it can possibly make use of the Google Map to direct itself intelligently(Machine to Machine).



(a) Iterative

Figure 5.6: Google's ‘Driverless Car’ launches in Saudi Arabia (Ratjaka n.d.)

Chapter 6

Objectives Evaluation

6.1 Location-Based services IOT

A lot of detail was found while researching location-based services including description of existing ones, especially the more popular services. It is clear that they provide this location service as the core component of the business but there are several tools which allow interoperability with their API services. Also addressed the issues that were involved with privacy.

6.2 Developed Application

An application that displays events in one's area was successfully achieved and hence a demonstration of what can be achieved using API technology was implied. Though simple design which was heavily criticized by users during the test phase, it achieves information transfer through separate technologies by standard practical means It does show it is possible to refine information to suit users location which entirely means it is just as good functionality wise as applications discussed in the Literature Review. Now does it bring anything relatively new to location-based applications ? No. It does however show a

6.3 Application Validation

The application was unit tested during development and efficiently tested at the final stage of the project life cycle which evidently displayed the flaws users found in using it.

6.4 API for IOT

The project does not succeed in demonstrating IOT due to an absence of a physical thing like a sensor.

Chapter 7

System Testing

7.1 Final Test Phase Questionnaire

Though unit testing was carried out through out the whole project life-cycle, to get genuine feedback from the users they were informed to not provide any unbiased answers, as it would not help the research, as biased answers were only a disservice. Straight forward and honest answers were required so as to critique the system as a whole and provide a basis on which further development can begin with. The application was deployed to the AWS Elastic Bean Stalk (http://default-environment-vimk6pgm3k.elasticbeanstalk.com) for testing and Google Form were used to present the questionnaire to the users. In SUMO fashion, pick statements they agree with, disagree with and undecided where most appropriate based on the Events Streamer Application. Apart from the last 5 questions who demand the user writes answer in own words, the rest of the questions are multiple choice are can be categorized as a positive or negative test outcome. The questions are:

- This software responds too slowly to inputs. -
- I would recommend this software to my colleagues +
- The instructions and prompts are helpful +
- This software has at some time stopped unexpectedly. -
- Learning to operate this software initially is full of problems. -
- I sometimes don't know what to do next with this software. -
- I enjoy the time I spend using this software. +
- I find that the help information given by this software is not very useful. -
- If this software stops it is not easy to restart it. -
- It takes too long to learn the software functions. -
- I sometimes wonder if I am using the right function. -
- The way that system information is presented is clear and understandable. +
- I feel safer if I use only a few familiar functions. -
- The software documentation is very informative. +

- This software seems to disrupt the way I normally like to arrange my work. -
- Working with this software is mentally stimulating. +
- There is never enough information on the screen when it's needed. -
- I feel in command of this software when I am using it. +
- I prefer to stick to the functions that I know best. -
- I think this software is inconsistent. -
- I can understand and act on the information provided by this software. +
- This software is awkward when I want to do something which is not standard -
- There is too much to read before you can use the software. -
- Tasks can be performed in a straight forward manner using this software. +
- Using this software is frustrating. -
- The software has helped me overcome any problems I have had in using it. +
- The speed of this software is fast enough. +
- I keep having to go back to look at the guides. -
- It is obvious that user needs have been fully taken into consideration. +
- There have been times in using this software when I have felt quite tense. -
- The organization of the menus seems quite logical. +
- The software allows the user to be economic of keystrokes. +
- Learning how to use new functions is difficult. -
- There are too many steps required to get something to work. -
- I think this software has sometimes given me a headache. -
- Error messages are not adequate. -
- It is easy to make the software do exactly what you want. +
- I will never learn to use all that is offered in this software. -
- The software hasn't always done what I was expecting. -
- The software presents itself in a very attractive way. +
- Either the amount or quality of the help information varies across the system.+
- It is relatively easy to move from one part of a task to another. +
- It is easy to forget how to do things with this software. -
- This software occasionally behaves in a way, which can't be understood. -
- This software is really very awkward. -

- It is easy to see at a glance what the options are at each stage. +
- Getting data files in and out of the system is not easy. -
- I have to look for assistance most times when I use this software. -
- What, in general, do you use this software for?
- How important for you is the kind of software you have just been rating?
- How would you rate your software skills and knowledge?
- What do you think is the best aspect of this software, and why?
- What do you think needs most improvement, and why?

7.1.1 Response Pie Charts Evaluation

Pie charts were used to visualise responses from the questionnaires, were for all the questions - about 25 of the multiple choice responses agree where it's marked positive and disagree where its marked negative.The basis for the number 25 is that more than 50% are in that group for each question. (See the pie charts for more detail) This means 25 questions out of 48 multiple choice questions, validate the application through the SUMO Test.

7.2 Conclusion

A location-based service was built, deployed and tested throughout the project life cycle that allows user to view events that were in proximity to the user.

Bibliography

Ajam, N., Ahson, S. A. & Ilyas, M. (2010), 'Location-based services and privacy".

Brodkin, J. (2010), 'Facebook, twitter becoming business tools, but cios remain wary'.
URL: `http://www.networkworld.com/article/2241880/security/facebook--twitter-becoming-business-tools--but-cios-remain-wary.html`

Buf (n.d.), *About Buffer.*
URL: `https://buffer.com/`

By (n.d.), *InstaMessage - Chat with Instagram users and nearby friends.* Version 2.2.2.

Curtis, S. (2014), 'How twitter will power the internet of things'.
URL: `http://www.telegraph.co.uk/technology/twitter/11181609/How-Twitter-will-power-the-Internet-of-Things.html`

danah m. boyd & Ellison, N. B. (2007), *'Social Network Sites: Definition, History, And Scholarship'. Journal of Computer-Mediated Communication*, Journal of Computer-Mediated Communications.

Fac (n.d.), *Facebook Login Best Practices.*
URL: `https://developers.facebook.com/docs/facebook-login/best-practices`

Fin (n.d.), *Find My Friends.* Version 4.0.1.

Foote, K. E. & Lynch, M. (1995), 'Geographic information systems as an integrating technology: Context, concepts, and definitions'.

Goia, M. (2011), 'How website aggregators like kayak work'.
URL: `https://www.quora.com/How-do-website-aggregators-like-Kayak-work`

Google (2005), *Google Maps APIs.*
URL: `https://developers.google.com/maps/`

Hutter, D. & Ullmann, M. (2005), Security in pervasive computing, Springer.

Lederer, S., Hong, J. I., Dey, A. K. & Landay, J. A. (2004), 'Personal privacy through understanding and action: Five pitfalls for designers'. University of California - Berkeley and Carnegie Mellon University and University of Washington.
URL: `http://repository.cmu.edu/cgi/viewcontent.cgi?article=1077&context=hcii`

model-view-controller (n.d.).
URL: `http://www.moock.org/lectures/mvc/`

Morris/, P. (2012), *InstaMessage For iPhone Lets You Send Private Messages To Instagram Users.*
URL: `http://www.redmondpie.com/instamessage-for-iphone-lets-you-send-private-messages-to`

Neville, S. & Allen, K. (2014), ‘Uk government to publish flood risk data’.
URL: `http://www.ft.com/cms/s/0/dcb1b524-8144-11e4-b956-00144feabdc0.html#axzz3n1F4sLXC`

Ratjaka, A. (n.d.), ‘Google’s ‘driverless car’ launches in saudi arabia’.
URL: `http://www.themideastbeast.com/google-car-launches-in-saudi/`

Robles, P. (2014), ‘Will twitter’s fabric connect the iot?’.
URL: `http://www.programmableweb.com/news/will-twitters-fabric-connect-iot/analysis/2014/10/29`

searchcrm.techtarget.com (n.d.), ‘Dirty data definition’.
URL: `http://searchcrm.techtarget.com`

Sommerville, I. (n.d.), ‘Software documentation’. Lancaster University.
URL: `http://www.literateprogramming.com/documentation.pdf`

Team, T. (2013), ‘How kayak’s business model creates value’.
URL: `http://www.trefis.com/stock/kyak/articles/162414/how-kayaks-business-model-creates-value/2013-01-11`

Tin (n.d.), *Tinder*. Version 4.6.1.

Twi (n.d.), *Twitter Documentation*.
URL: `https://dev.twitter.com/overview/documentation`