Dissertation submitted in partial fulfillment of the requirements for candidate for MSc in Web Technologies.

A Frar ·· for Location Based

ıd

ımputing

National College of Ireland, Mayor Square, Dublin 1, Ireland

Supervisor

Dr. Keith Maycock

Contents

Abstract

This thesis investigates current technologies, social acceptance and culture of this generation to create a framework for location based advertising. Location aware services are becoming more prominent on the Internet over the past number of years, and this thesis is concerned with investigating if coupling the technologies of the present, with the social culture that spawned from the Web2.0 era, can be used to create a framework for the dynamic dissemination of advertisements to users based on their current location. The aim for such a platform is to provide a more meaningful, context-aware advertising experience for both users and advertisers, than that of traditional media channels such as newspaper and television.

1 Literature review and research question

1.1 Introduction

This chapter is involved with an investigation of the literature within the area of location based advertisements and location-aware services.

Across the literature, there are four prominent topics that are encountered and each of these topics is discussed below.

Location based advertising is defined by [Bruner II, 2007] as

"marketer-controlled information specifically tailored for the place where users access an advertising medium" ,

and has been around for years. It simply refers to positioning a physical piece of information in a position where it is most likely to be accessed; and each day, flyers and posters are strategically positioned near bustling streets in the hope that more people will see them.

Extending upon Bruner's definition, and coupling the ability to tailor the information with location-aware devices such as smart-phones, however, is a new concept. It is as a direct result of technological advances that such a concept may be feasible.

This research project is involved with the following research question:

> Can current technologies, social acceptance and culture of this generation be utilized to create a framework for location based advertising?

The goal of this dissertation is to determine if it is possible and to detail the design, architecture, and implementation of such a framework.

The literature involved in this area is categorized into the following areas.

- Overview of location based advertising
- Business models associated with location based advertising
- Current concerns towards location based advertising
- Potentials and possibilities of location based advertising

The following subsections investigate these areas in chronological order.

1.2 Literature Review

In this section, the key literature relevant to this dissertation is reviewed, and classified into one of the four prominent areas within the broad topic of location based advertising. The preceding section details the outcome of the review and the conclusions derived from it.

1.2.1 Overview of location based advertising

Location based services have a long tradition that date back to the 1970s when the US Department of Defence began operating the global positioning system. The global positioning system, or GPS as it is more commonly known, is a satellite infrastructure used for the identification of the current position of people and objects [Kauffman, 2004]. The process used for the identification of these entities is known as positioning, and is defined by [Kupper, 2005] as

> *"the process used to obtain the spatial position of a target".*

[Mauw,] defines a location based service as

> *"using the ability of a mobile device to determine it's own geographic location, to provide personalised services based on that location".*

This means that given that a user's mobile device has the ability to determine its geolocation at any given time, it could then be possible to query a service to determine if there is any useful information that could be disseminated back to the subject.

Similarly, Bobby L Rush, chairman of the United States House of Representatives Subcommittee of Commerce, Trade, and Consumer Protection, defines location based services as

> *"to some extent, a subcategory of of behaviour tracking, in that it can quickly, and cheaply, tell advertisers more than contextual advertising ever could about [a person's] preferences, habits, and patterns". [Kelley et al., 2010]*

The interesting contrast between the two above definitions ([Mauw, , Kelley et al., 2010]), is that the former is more user focused, and the latter more advertiser focused; with the former speaking of disseminating useful information back to the subject based on their geolocation data, and the latter speaking of communicating useful information back to the advertiser. This appears to be a prominent divide across the different literature on the topic.

On one side of the divide, you have those who appreciate the topic for the possibilities that it can offer the user, such as context aware suggestions, and discounts for local services. On the other side, those who appreciate the topic for the possibilities that is can offer the advertiser, such as trend / behaviour prediction and user preferences.

The one thing that both share in common however, is an appreciation for the great potential that lies in this area. It is predicted that in five years time, 2016, consumer and advertiser expenditure for location based services could reach $10 billion.[NewsoftheMedia, 2011]

Current moves and strategies incorporated by the modern day "big players" such as Google and Facebook[Facebook, 2011] further indicate that future services will be backed by location aware data. Facebook, for example, last year released Facebook Places[FacebookPlaces, 2011]. A service that allows users of the Facebook platform to "check-in" to virtual representations of real places. These check-ins are then shared with friends and venue owners. Foursquare[Foursquare, 2011], a similar service, allowed users to "check-in" to nearby places. The user with the most "check-ins" at a given location, was deemed the virtual mayor of that location. In some cases, retailers such as Starbucks, began providing rewards for people who obtained mayorships in their stores.[StarbucksFoursquare, 2011]

To accomplish tasks that are location dependent, there must be an effective means to determine the geolocation of an object. This is known as positioning, and it is defined as the process with which the spatial position of an object can be determined, and it occurs all around us all of the time.

For example, positioning enables the use GPS navigation in our cars. Positioning allows us to know when our flights are going to be delayed. Positioning lets us get directions to the nearest train station. Modern mobile devices provide multiple means for positioning. All mobile devices are track-able by their network operator using triangulation. Many mobile devices carry SIM cards which can be used to obtain raw radio measurements from the handset. Alternatively, devices that have the ability to connect to wifi APs can be more accurately mapped when indoors, for example, when connected to a wireless hotspot inside a shopping centre. Some methods however, are not usually directly accessible to the platform providers.

Additional means such as OS API's however allow the provider to access the device's geolocation data. Intelligent programming techniques allow the platform provider to work through the different methods and determine which are available. The provider can then choose to use the most accurate method.

Regardless of the method used to obtain the location data, location based services use this data to provide a contextual service.

1.2.2 Business models associated with location based advertising

Within the area of location based services, it is clear that two prominent use case / business models have been identified. The first of which is the push model, and the second, the pull model. The core difference between the two, is the point of control within the architecture. The latter, places the user in control, whereas the former removes an aspect of the control from the user.

The "push" taxonomy, encompasses a service that is controlled not by the user, but by the framework or the advertiser. The user does not explicitly request communication, but based on their geolocation, a message is communicated, or pushed, directly to them. This approach, much like the "pull" taxonomy, has both pros and cons. The pros of such an architecture is that it can be used to provide suggestions and recommendations to the user on something that they may not have previously been aware of. For example, if a subject generally spends a number of hours per week in an Italian restaurant, the system may build a profile of the subject and categorize them as somebody who enjoys Italian food. If the subject is then in a region that houses an Italian restaurant nearby, the system may suggest that the subject visits the restaurant. If the user profiling is correct, and the user was hungry or simply curious at the time of notification, then the system will have effectively provided a correct recommendation.

The flip side of such a service however, is that had the user simple worked at the restaurant, and actually disliked Italian food, then the notification would have been deemed obtrusive and spam-like. To solidify the above, findings by Tsang, Ho, and Liang [Tsang et al., 2004], indicated that users would evaluate mobile advertising negatively if they had not previously consented to receive such advertising.

Contrasting this model with the "pull" model, the user is required to "opt-in" to receive notifications from the framework. To ensure that the notifications are of the highest quality, the framework could allow the user to set preferences and on the type, or category of advertisements and notifications that they would like to receive.

1.2.3 Current concerns towards location based advertising

One of the prominent concerns with regards to location based services, is that of privacy. Rust, Kannan and Peng [Rust et al., 2002] define privacy as "*the degree to which personal information is not known to others*". *Placing* this definition into the context of location based services, the personal information is assumed to mean both location data, and preference data. Patrick Clarke [Clarke, 2011] defines location data as

> "*any data processed which indicates the geographical position of the terminal equipment (i.e the mobile handset) of a user*".

This location data can include the longitude, latitude, altitude, direction of travel, and time of recording. Location data is the core of any location based service, and Clarke goes on the provide recommendations on how best to handle the data. Clarke then advises that location data should only be processed if The user cannot be identified by the data. Such data is necessary for the provision of a value added service. The user has consented for their location data to be recorded.

Of the three points above, point three is of particular interest in that for a user to give consent to such an activity, there must be some base prerequisites in place. Firstly, the user should be well informed of what it is exactly that they are consenting to. The information for this must be freely available to the user. It should also be possible, in the same way that consent can be given, that consent can be withdrawn by the user at any stage. Withdrawing consent can be as simple as removing an applications, logging out, and opting out and depends on the situation.

Additionally, the user should have prior knowledge of the precise usage of any collected data, as in, how and why it is to be processed, and if the data will be released or provided to any third party sources. Should all said prerequisites be fulfilled, the user can knowingly, and meaningfully agree, or disagree, to have their location data recorded and processed. Clarke's recommendations are similar to that of Andrew Rimkus [Rimkus,] who iterates that L-commerce must provide users with the ability to control who knows their locations and how it's used.

1.2.4 Potentials and possibilities of location based advertising

Location based advertising has the potential to provide many benefits for both the advertiser and the consumer. The potential associated with location aware services has been noted for many years and is reflected in the first digital advertisement displays that changed the advertisement being displayed according to the time of day. The manufacturers of these digital signs were aware that engagement could be optimized by dynamically adapting the advertisement based on certain criteria, which in this particular case was time of day.

Later iterations of such devices included cameras that could detect demographics, and adapt the advertisement based on the determined primary demographic. As Partridge[Partridge,] stated,

> *"although the results were inaccurate, they only needed to be good enough to make advertisements that are adapted more effective than those that are not".*

Location based advertising using mobile devices is still in it's infancy, and whilst early iterations can be effective, it will be through contextual analysis and data mining the data generated, that it can further develop to provide an even better framework.

Hristova and O'Hare's[Hristova and O'Hare, 2004] Ad e-system demonstrated how user profiling through location patterns could provide a targeted audience for owners of a Chinese cuisine restaurant. The audience was composed of individuals who frequently visited Chinese restaurants, and were then categorized into the appropriate market segment. This type of audience classification would not have been possible through regular advertisement channels as the information would simply not have been available. The benefits outlined by Hristova and O'Hare's system were tailored towards advertisers, though using the same data in a different context, the benefits could have been tailored towards the user. The system could have created the same classification of users, but then provided suggestions of alternative Chinese restaurants. Furthermore, the system may have intelligently identified activity trends derived from all the users within that classification. For example, a derived trend of "39% of users of this restaurant are also frequent visitors to The Foobar Cafe" could be provided to the subject via a suggestion / recommendation engine. The benefits of such an engine are two-fold in that the user is receiving useful information, and the venue is being marketed to a targeted audience that are likely to engage.

One potential avenue for location based advertising is that of an auction-based distribution of locations to advertisers. Such an application could weight the different locations within a city based on the potential audience that may traverse through. Weighting factors such as size, number of shops an attractions, proximity to public transport, previous statistics, among others could then be used to build a profile for a particular location. Historical statistical data could be mined to determine that the likelihood that a user with an interest in modern art will traverse through this location is more than that of an avid reader passing through. Such a location can then be auctioned off to advertisers of modern art related information. The perceived value to advertisers of a more direct target can then be sold to the highest bidder.

Another interesting possibility for location based advertising would be a full integration between the advertising framework and a mobile phone operator. Coupling the mobile phone operator, which manages network user accounts, with a location based advertising framework, could enable dynamic network tariffs that are subsidised by the advertiser, via this 'push' based model. For example, looking at popular Pay-As-You-Go plans for the Meteor mobile network [Meteor, 2011], we can see that they provide deals based on a monthly fee ranging from €5 to €30. If this was coupled with an advertisements framework such that a user receives €0.50 for each received advertisement, it becomes clear that it would be possible to completely subsidise a user's monthly fee with between 10 and 60 advertisement per month. The upper scale of this range enumerates to 2 advertisements per day over the course of a typical month.

Such a scenario could entail advertisers routing a number of advertisements through the network operator, with a particular demographic destination target that is triggered when the target enters a location. When the user receives the advertisement, a percentage of the operator's fee to the advertiser is discounted from the user's phone bill. This builds incentive for user's to engage in the 'push' model. A model, which if not implemented effectively, can be deemed invasive to the user. The incentive for user's to reduce their phone bill and also to receive interesting information would raise demand for advertisements. Advertisers routing their advertisements through the mobile network are assured that the information will reach the destination and not be deflected by native applications on the device being deactivated.

The environment created by a framework like this is that user demand of a particular mobile operator increases through being incentivised via having their phone bill wholly or partially paid for by advertisers. There are existing services that couple advertiser with mobile operators, but no services that couple advertisers, mobile operators, and the context provided by location data.

1.3 Conclusion

The conclusions derived from reviewing and analyzing the different literature are documented in this section. These conclusions, in addition to the results of initial market research, which can be found in the sectional named 'Evaluation and results', are used to drive the requirements for the Libero architecture. The requirements have taken into account the documentation, advice, and guidelines from the different authors.

The literature review presented a recurring definition of location based advertising and its core composing elements.

> *"Presenting more contextual information to the user based on their geographic location at a given point".*

Some interesting points made in the different literature refer to how the information is sent from the framework to the device, specifically, is the information requested, opt-in / pull, or is the information sent to the user without prior request, opt-out / push. It is clear that the choice of architecture model is dependent on the application that is being developed. A particular model may be well suited to one context but not another. From the literature, it is derived that each model has different base requirements and ultimately, is dictated by where the point of control lies within the system as a whole.

If the point of control is focused on the user, then the pull model is more suited, whereas if the point of control is focuses on the advertiser or marketer, the push model prevails.

For the Libero application, the point of control is centered on the user and therefore the architecture will encompass the pull model.

A very important topic presented in the literature is that of user privacy. Key points by Rimkus[Rimkus,] suggest that the user must give consent for their data to be processed, and to do so, in a meaningful way, there should be full disclosure as to how that data will be processed and to whom it will be visible.

Other suggestions include that the user be given the ability to stop their location data from being processed. This suggestion holds particularly valid with the results of initial market research where users specifically requested to know that they could easily terminate the data processing.

Taking these key points the Libero application will provide full disclosure to the user as to the data that will be collected, and what are the system's intentions with that data. Extending on advice from [Rimkus,], the Libero application will not be processing personal user data such as names, addresses, email addresses, phone numbers etc. The user, to Libero, will solely be identified by their devices UUID. Using the UUID ensures that Libero can identify users without the need to process sensitive information.

Location based services is still in it's infancy and a lot of research has been made into the area. If location based services are to success and be adopted by users, it is essential that the users neither feel vulnerable, nor that their data is unsafe. The culture rationale with regards to privacy in an online context has been to ensure that one's own PII (Personally Identifiable Information) is safe and not released to third parties without prior knowledge and consent.[1GoodReason, 2011, Brain, 2011]. Past and present success of applications providing interesting offers and deals from third party vendors to users have shown that users are quick to adopt and maintain relationships and usage with such frameworks [Gowalla, 2011, Groupon, 2011, Foursquare, 2011].

Having reviewed the different suggestions by literature authors, some of which have been adopted, others extended upon, and others negated, Libero will be developed to fulfill all requirements set about by the suggestions adopted and extended.

2 Theory and background

2.1 Overview

This chapter investigates the theory and background around location-aware services in the context of location based advertising, and details the evolutions within the area that have occurred of the years. It begins by identifying issues surrounding traditional advertisement media channels, such as inadequate advertisement placement. It then goes on to detail the evolution of location based services in the context of the web. From here, some potential use-cases for location based advertising are presented to the reader. Concerns with the initial use cases are detailed to the user, and potential solutions are described. This chapter concludes with an overview of the most important details of the evolution and provides derived conclusions with which the framework may be implemented.

2.2 Theory and background

Location based advertising is defined as

> "*marketer-controlled information specially tailored for the place where users access an advertising medium*" [Bruner II, 2007]

and means displaying information to a user where that user is most likely to engage with the information.

John Wanamaker coined the term

> "*half the money I spend on advertising is wasted; the trouble is, I don't know which half*",

and this terminology describes the difficulties that organizations face when advertising a product or service, with a common misconception that the more money an organization spends on advertising, the more effective that a particular campaign will be. And to an extent, this makes some sense. One could make the assumption that posting an advertisement in ten local newspapers would be more effective than posting an advertisement in a single newspaper. The problem however with this mentality, is that the users most likely to engage with such an advertisement may not be physically located where they could actually engage. A typical scenario of this would be a commuter who picks up a newspaper on their way to or from work, and sees an advertisement of interest towards the end of their journey, but because the advertised product or service is not in the immediate vicinity, they take no further action on it. The contrasting scenario would be the same commuter is walking from the train station to their place of work and they notice a similar advertisement for the same product or service. This advertisement is for a product that is only a few metres away from the user's current location, and the user engages with the advertisement and inspects the product. Thus, the marketer has tailored the information to be presented to the user where they are most likely to react positively.

Coupling this type of advertising with modern day technologies can provide an efficient, moreover, effective, experience for both the user and the advertiser by allowing the user to receive a more direct, personalized advertisement, and the advertiser to pinpoint 'where' the target will access the advertisement.

A survey in 2010 by Mindshare Ireland[MindshareIrl, 2011] determined that 87% of all phones in Ireland had the ability to connect to the Internet, and a study by Silicon Republic[Kennedy, 2011] shows that Smartphone usage in Ireland over the years has continually increased. These statistics reflect the evolution in the web development industry, where application developers have realised that smartphones are key to allowing users interact with their services whilst on the move. Some of the most popular web applications and services provide either a mobile interface or a dedicated mobile app. Some examples of such applications include Facebook, Basecamp[Basecamp, 2011], LinkedIn[LinkedIn, 2011] and many more. Moreover, some applications exist primarily as mobile applications, and only sometimes provide a secondary web application. Some examples of services like this include Foursquare[Foursquare, 2011] and Gowalla[Gowalla, 2011].

Regardless of primary medium, there has undoubtedly been a massive increase in the number of services that cater for mobile usage and access, by providing explicit mobile versions and interfaces.[jQuery, 2011] The great thing about smartphones and handheld devices is that we can carry them around with us all the time. There's no longer a need to carry around bulky laptops and PDAs to access online resources. This goes hand in hand with the user's desire to constantly be 'wired-in' and keep up to date with what's happening online and actively maintain their own online presence with activities such as posting a tweet or checking in to a virtual location.

As the Internet has evolved over the years, and web services are providing a more user-defined, more intimate online experience, users are willing to share more data with these services. If we go back before the Web2.0 era, it was not uncommon for the majority of online users to use aliases, alternate email addresses, or random unidentifiable usernames and avatars. Real identities were not ofter exposed online. Nowadays, however, this is not the case, and it would seem that the more of ourselves that we share with the Internet, the more tailored is our experience.

By creating 'public' profiles on social networking sites, we allow ourselves to be searched for, we allow pictures of ourselves to be posted. By letting search and recommendation engines to monitor our activities, we get the benefit of suggestions and recommendations. By adding our location to a post or a tweet, we provide more context and meaning to that piece of data. As twitter states in their geolocation documentation, *a tweet with the text "Fireworks" alone is not very meaningful, but when that same text is coupled with some location data, we know exactly where the fireworks are happening and it brings more context to the data.*

In addition to providing context, some services base the core of their product on location. Take Foursquare for example, which is a location based service that allows users to check-in to nearby virtual locations. These are virtual representations of physical places and locations. The idea behind this services was that users would check-in when they arrived at a location such as an airport, a coffee shop, a book store, anywhere. Foursquare would then tell the user which other Foursquare users were also checked-in to that place. To keep the user interested, various badges and awards were available for things like "*Most check-ins at a restaurant this week*", and "*Busy week (more than 3 late night check-ins)*".

Foursquare was quickly embraced, and is now values at over €95M.[Technorati, 2011] Soon, other companies began providing similar services, take Gowalla or Facebook Places for example.

The importance of Foursquare however, and it's relativity to this dissertation, is the fact that users are more than happy to share their location data. Foursquare demonstrated that John Smith is comfortable telling the world that he was in Wagamamas at 14:30 last Thursday. If our users are willing to share their location data with us, then we can provide more meaningful services to them, and this is the trail of thought for consumer benefits from location based advertising. This could be mapped to a benefit scheme for, for example John Smith wanting something. If, at a given point in time, we know that John Smith is in the Dublin 2 area, and if we know that John Smith was interested in Italian Food, then we could let him know that there's an Italian restaurant nearby.

A hot trend at the moment are services that provide discounts and savings to third party products and services. Many of these types of applications have appeared in the last year; with Groupon[Groupon, 2011] and Living Social[LivingSocial, 2011] being the most prominent. The workings behind both are as follows; a third party company says to application that they would be willing to provide a particular discount to a specified number of people. The application then emails this offer to it's customer base. If the specified number of people, as defined by the third party that is offering the service, sign up for the deal, then the deal is activated and those who opted in, are given an e-coupon to avail of the deal. If, however, less than the required number of people sign up, then the deal is void.

Coupling the topics of 'daily deals' with intelligent location aware context has the potential to provide an extremely useful service. Organizations could then offer incentives to potential clients and customers based on where they are located at any given time. AB Bagels could then send an offer for "a 10% discount on all bagels" to users near each of their stores. They could choose to run this advertisement campaign between 12:00 and 14:00 when most of the local workers would be on their lunch hour. Using the above theoretical use case, this has the makings of a very effective platform, though at this point however, it is lacking prevention of mass mailing all its users. By allowing AB Bagels to send out their "10% discount on all bagels" to all platform users in the chosen areas, we have actually sent all users a form of spam. Spam, defined as any unsolicited communications, was created through the fact that not all of the users would have wanted to receive the bagel discount advertisement.

The platform can remedy this by providing the option of an opt-in, also known as a pull model. A pull model, in the context of the user, is defined as "the user seeking the information (pulling it towards them)" as opposed to "the information being pushed out to the user".

Such a mechanism hands over the element of control to the user, and allows the user to opt for different categories of advertisement. At a high level, we could assign each advertisement into a category. Retail, Food & Drink, Homeware etc. Together, the above features form the ingredients for an effective platform. On the user's side, a more contextual, personalized, meaningful medium for receiving information of interest. On the advertiser side, a direct target at which to aim their message. A target that is more likely to engage with the message. This increased likelihood can be caused by the user's perceived value of the advertisement, given that the advertisement was sent directly to them and not mass sent to everyone.

2.3 Conclusion

Location based advertising is an exciting concept that has been inspired by the evolution of cultural rational towards more personalized Internet services, and the technological advancements in the area of mobile computing. The ability to obtain a person's location via their smartphone enables applications to process the position and dynamically disseminate data to that user. In addition to a service that is solely concerned with dynamic dissemination, the service could be coupled with a mobile operator to provide a benefit scheme such as subsidised tariffs. For advertisers and media buyers, this new environment provides a means of interacting with customers and reaching out to potential customers in a way that is not possible through the traditional media channels such as newspaper, television and radio.

Two prominent architectures have been identified within the area; the push model and the pull model. The former focuses the point of control around the entity sending the data, and the latter on the recipient. The usage of either model depends on the context in which the framework will reside, but ultimately can be adapted to accommodate a wide range of environments.

Choosing an appropriate architectural style however is not enough to build such a framework and adequate research and design must be done into the aspect of user privacy and data handling, such that the user has prior knowledge of the data that will be collected, the purpose for which it is being collected, and the details towards data ownership and how the data may be post-processed.

The two above areas can provide a stable foundation on which the service can be built.

3 Architecture and Implementation

3.1 Overview

The research question that this thesis investigates is

> Can current technologies, social acceptance and culture of this generation be utilized to create a framework for location based advertising?

This chapter is concerned with the architecture chosen, and the implementation steps used to build the framework. Based on the requirements for such a framework there needs to be a means by which the framework can obtain a user's location data, process that information and if necessary, disseminate some data back to the user.

The previous chapter illustrated two different architecture model that could be applied to such a framework and determined that the usage of one model over the others was directly related to where the point of control lay within the system. This framework will implement the 'pull' model, where the user will be responsible for providing the location data to the server, via their smartphone, to request the advertisements.

Other topics highlighted by the previous chapter centres on that of data privacy and so users will be indentified by the device's unique identifier as opposed to any personal user data, to ensure that data privacy is not a concern.

The complete architecture of the Libero platform is comprised of three subsystems, and high level overview of each system follows, preceeded by a more indepth details of the subsystems.

The first subsystem is the web application and it is through this that the advertiser interacts. This system is responsible for advertisement CRUD and report viewing.

The second subsystem is the mobile application and it is concerned solely with the end user. It allows the end user opt-in and opt-out of the various categories. It also serves the function of background location tracking and processing. The Libero mobile client is compiled to run on the Android[Android, 2011] operating system. This decision was reached as a result of end-user market research.

The third subsystem is what I refer to as the platform core. The platform core is the glue that binds the web application with the mobile application. Although physically a part of the Rails[RubyonRails, 2011] application, it is not interacted with by the end-user nor the advertiser. It's primary goal is to manage the incoming requests from the mobile client, and when necessary, return an appropriate response.

Each subsystem maintains its own requirements, responsibilities and architecture. The following section will provide a detailed description of the architecture and implementation of each.

3.2 Proof of concept

The initial tasks involved with developing the Libero application were to prove the ability to communication from a mobile client to a remote web server. Throughout the development of this lightweight prototype, knowledge of the domain and the different technologies increased. Below, are the various partonomies of the prototype, and within each section, any significant findings are mentioned.

3.2.1 Web server and application

The prototype web application was build using the Sinatra[Sinatra, 2011] micro-framework. Sinatra is a DSL (Domain Specific Language) for creating web applications using the Ruby programming language[Ruby, 2011].

It provides the developer with a mechanism for defining a route, the HTTP verb associated with that route, and what the application should do when that route is requested. An extract from the initial prototype is seen below

```
DataMapper.setup(:default, ENV['DATABASE_URL'] || "sqlite3://#{Dir.pwd}/mydb.db")
class Ping
  include DataMapper::Resource
  property :id, Serial
  property :uuid, String
  property :longitude, Float
  property :latitude, Float
  property :created, DateTime
end

Ping.auto_migrate! unless Ping.storage_exists?

post '/' do
  lon = params[:longitude]
  lat = params[:latitude]
  uid = params[:uuid]

  ping = Ping.create( :longitude => lon,
                      :latitude => lat,
                      :uuid => uid,
                      :created => Time.now)
  200
end

get '/pings' do
  @pings = Ping.all(:order => [:created.desc])
  erb :pings
end
```

Stepping through the above piece of code we can see that we first define a model to encapsulate the attributes of a Ping. The format is property <propertyname>, <propertytype>

Once the model has been defined, we ensure that the database is created and migrated.

The next definition states that when a POST request is received to the root '/' web application, parse the parameters, create a Ping model, and respond with status 200 to the client. Similarly, if the server encounters a GET request to the '/pings' URL, lookup all pings in the database, sort by created descending, define an instance variable @pings to encapsulate the array, and return it to the client.

The above code extract was used to ensure proof of concept on the web application subsystem. It was tested by sending a GET request to each exposed route via CURL[CURL, 2011].

The next section details initial testing on the mobile client.

3.2.2 Mobile client

The requirements for the initial mobile client were that it have the ability to determine its own location, and that it be able to send that location data to a remote server.

The mobile application was written using HTML5 and JavaScript, though later versions of the application required native Java.

The framework used was Phonegap[Phonegap, 2011] and it provides Javascript bindings to native code which enables the developer to write code like the following

```
var getLocation = function() {
  var success = function(p) {
    $.ajax({
      type:'POST',
      data:{ longitude:p.coords.longitude,
          latitude: p.coords.latitude,
          uuid: device.uuid},
      dataType:"json",
      url:'http://stormy-sunrise-317.herokuapp.com/pings/new',
success: function(data) {vibrate();}
    });
  }
}
```

The above code demonstrates how the mobile client sent location data to the remote server by encapsulating it within an Ajax-style request.

Within the $.ajax function, an opts hash is passed in which defines the type of request to send, the data to be sent, the data-type expected back, the URL to send to, and the callback to be defined if a 200:OK response is received.

The application to which the above snippet belongs was then loaded an Android handset, and tested. The testing required the handset to obtain it's location every 30 seconds, and then to send it on the the server. The tester kept the handset in their pocket whilst commuting to work, and then the route generated from the incoming pings was tested against the tester's actual route.

3.2.3 Results

The results of the initial proof of concept were that the mobile application function as expected when the handset was active, but not when the handset entered sleep mode, as the accuracy of the results differed depending on the method used to obtain the handset location. For example, using GPS and WiFi enabled very accurate results, though using the mobile operator sometimes rendered results within a 55 metre radius.

Below is a screenshot of one ping that was received and mapped against Google Maps[GoogleMaps, 2011].

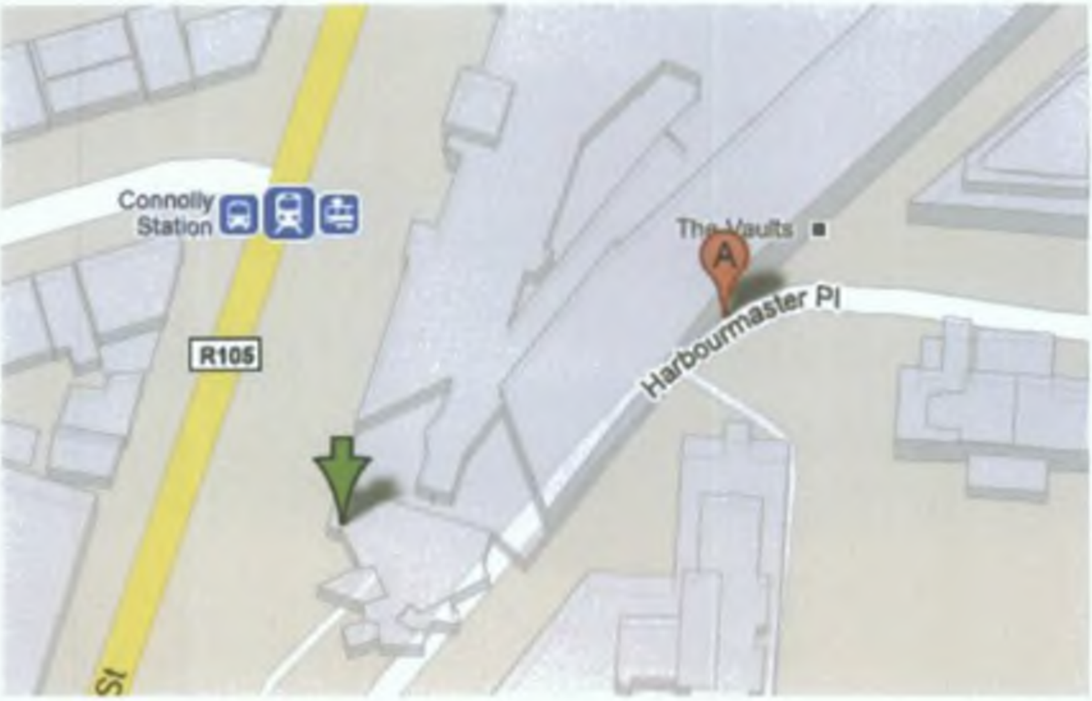


Figure 1 - Screenshot from Google Maps

The green arrow above indicates the position derived from reverse geocoding the longitude and latitude pair received to the server.

This result was extremely accurate as confirmed by the tester, who used this exit from Connolly Station when commuting to work.

3.2.4 Conclusion

The outcome of the proof of concept illustrated two areas that required further testing and development. Firstly, a solution needed to be formulated to ensure that the application had the ability to run in the background as opposed to just running when the handset is active. Secondly, the possibility that the mobile triangulation may be approximately 55 metres off must also be taken into account within the algorithms of the platform core.

Taking both the above points into account, the Libero platform now had a base that it could use to develop upon, and the following sections will details that implementation.

The next section details the web application and the following sections detail the mobile client and the platform core respectively.

3.3 Web application

The core functionality of the the web application is to provide advertisers and media buyers with the ability to create and modify ad campaigns, and to view reports and statistics associated with each campaign.

The framework used to build the web application is Ruby on Rails (Rails), and more specifically, edge Rails[Rails, 2011]. This decision to develop using the bleeding edge releases of Rails was to ensure that the project could benefit from the new features and toolkits that were created as a result of recent structural and architectural changes to the framework itself. Some of these benefits include the asset pipeline, an exciting new micro framework that embraces Rails 3.1's "Fast by default" message by promoting assets such as stylesheets and JavaScripts to first class citizens within the context of the framework.

These assets are now identified to the client by name and also by MD5 hashing their contents. This allows the client to cache the file until the contents change. In addition, the assets are also minified and compressed. Thus reducing the number of HTTP requests that must cross the wire. Another major feature is that Rails applications now support streaming, which means that the browser can now download the stylesheets and JavaScript while the server is still generating the response.

Rails itself is an open source web development framework that embraces convention over configuration and encourages modern day web design patterns such as the Model View Controller pattern for presentation, and the Active Record pattern, as coined by Fowler, for object-relational-mapping.

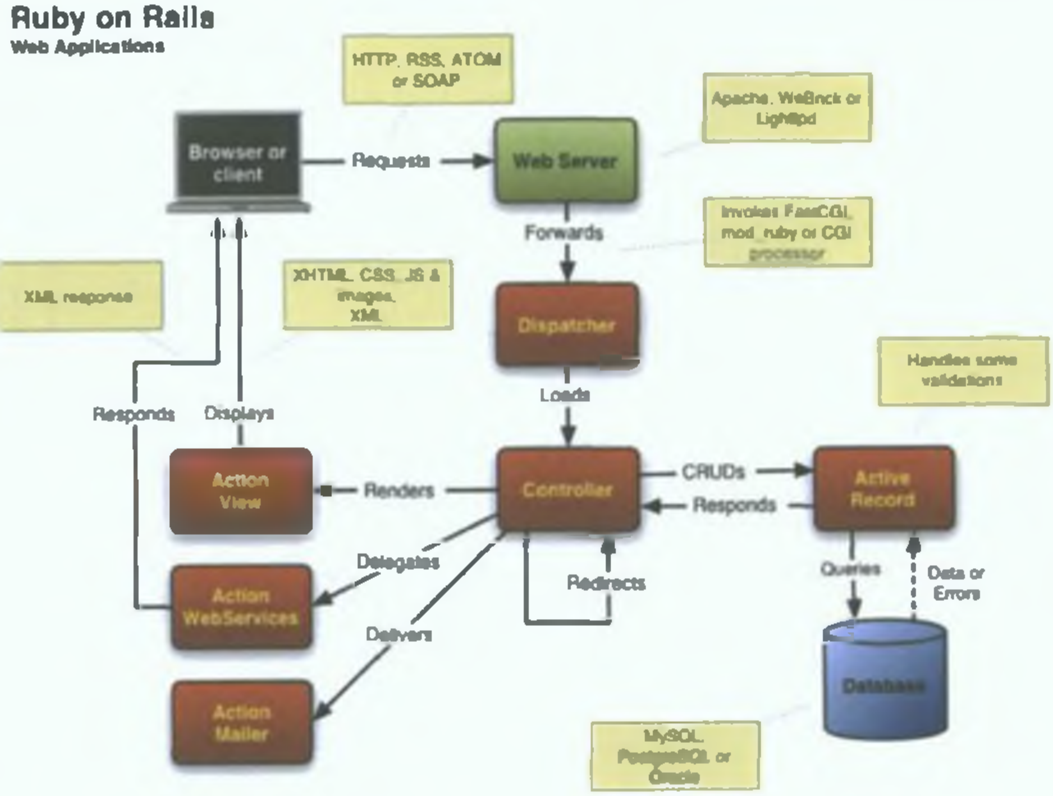


Figure 2: Ruby on Rails architecture

The above diagram illustrates the typical data flow and life-cycle within a Rails application.

Starting the the browser, the user issues a HTTP request which is routed to the web server. Libero utilizes the Thin web server as it is not only lightweight, but scales well in a cloud environment. The web server forwards this request to the dispatcher, which is actually a ruby class that inherits from ActionController::Dispatcher.

It is the job of the dispatcher to route the request to the correct controller.

Controllers are the C in MVC and it is their job to handle requests, invoke any CRUD actions, load any required models, invoke services such as mailers, and pass the required data back to the view layer for presentation to the client.

As the diagram shows, the Rails controller will typically query the database via the Active Record interface, assemble the data required or the view, and pass controller over to Action View.

Action view will then respond to the client with typically HTML, Json, or some other data structure. In recent years, the use of XML as a responding format in Rails has declined in favor of Json (JavaScript Object Notation) which is a more lightweight data structure. In addition to the decreased overhead associated with Json, the increased usage of JavaScript on the frontend (and as of late, the backend too, see Node.js), has been less painful through the use of Json, which maps directly to JavaScript objects.

Sinatra / Rack was used during the prototyping stage but this work was later merged into the Rails application.

The RDBMS' used include SQLite3 for development, MySQL for testing, and PostgreSQL for production. SQLite3 is becoming an increasingly popular database for the development environment due to its simplicity and ease of setup. It works extremely well for early development and prototyping but it is not a production database.

Within the local, testing, and staging environments, MySQL. MySQL remains one of the most popular RDBMS in use across the software industry.

PostgreSQL however, is the default database back-end on the Heroku[Heroku, 2011] platform, and there can be subtle differences between itself and MySQL, specifically when raw SQL is used in the application, and as a result, it requires independent testing.

Git was consistently used for version control. It is fully distributed, supports various hooks, and makes branching and merging a pleasure. In addition to these features, Capistrano, a tool for automating deployments and releases, supports Git by default.

SCSS was used in place of CSS due to the functionality and ease of development provided by the use of variables, nested rules, mixins, and selector inheritance. This aids in keeping the code DRY (non-repeating), modular, and also reduces the size. jQuery was used as the core JavaScript library as it provides great support for DOM traversal and manipulation. The ability to extract common behaviours and implementations into plugins is a major productivity boost. Coupling this with a reduced development time, consistent Ajax support and method chaining, it makes a for a great tool for web development.

The project was developed with TDD (test driven development), using unit tests, functional tests, and integration tests. Mocking and test data was built using factory_girl and mocha alongside test/unit.

The ORM used, for the most part, was activerecord, however, during development, some aspects of the architecture used datamapper as the ORM. These aspects included early prototyping of mobile client communications with the core platform. Activerecord allows us to encapsulate a row of data from a database table, as a object construct. Rails' activerecord library provides the functionality outlined by Fowler[Fowler, 2002] and adds some nice features along the way. If no native SQL is used in the query interface, then there is no need to write different SQL statements for the different RDBMS. Activerecord takes care of mapping the native ruby code to be RDBMS agnostic. Within the application itself, the MVC presentation pattern for design was followed, and Fielding's REST model was also used when appropriate.

The following section describes, in detail, the inner working and structure of the web application. Diagrams will be provided, where deemed necessary, for ease of explanation and understanding.

The core of the application is built upon six models.

- User
- Campaign
- Category
- Location
- Advertisement
- Ping

3.3.1 User

This section deals with the user model, which represents the user entity of the system. Within Libero's web application, a user is deemed to be an advertiser or media buyer. The web application, as per the initial requirements, does not provide any functionality for the 'end-user'. Such functionality has been decoupled and responsibility handed over to the mobile client.

On the web application, (for the media buyer), the user uses the system for advertisement campaign CRUD functionality. Version one encompasses no billing functionality, though in future iterations, there will be a semi-variable costing structure applied, and the user will manage billing details and invoices through the web application.

The system does not store any plain-text credentials belonging to user, and uses a hash and salt to ensure the user data cannot be easily compromised. This functionality comes pre-baked with warden, the security strategy around which Devise was written.

The user model itself, inherits directly from ActiveRecord::Base and thus maps directly to a single row in the users table. Foreign key mappings and associations create the one-to-many relationship between the user instance and the campaign instances.

ActiveRecord::Base is the concrete class exposed by the activerecord framework. Subclassing this class ensures that the base inherits all CRUD actions on both it's own table, derived through explicit pluralization of the class name, and also on any external tables via foreign keys.

3.3.2 Campaign

This section describes the campaign model which encapsulate the attributes and behaviour of an advertisement campaign. It too inherits from ActiveRecord::Base. It's own attributes include start and end timestamps, a name, and a description. There are multiple associations pointing to and from the campaign instance. A user_id foreign key for example, points to the user instance that created / owns the campaign. Other foreign mappings include category_id (the category in which the campaign resides), location (the location associated with the advertisement), and the advertisement itself. Decoupling the advertisement from the campaign sets the foundations for more complex advertisement media such as audio and video. Along with these associations, another important mapping is that between a campaign and a ping. A ping, much likes its better known meaning, is a communication from a device, through the platform core, mapped to a particular advertisement campaign, such that he number of pings associated with a campaign directly correlates to the number of impressions associated with that campaign.

The start and end timestamp attributes are used to determine if a campaign is active and allows the platform and web application to scope campaigns by state. States include active, pending, and completed. A campaign that is in the pending state, is fully modifiable. A campaign only becomes static once it's start time has been reached.

Continuing on with the model definitions, each campaign has an associated category to which it is said to belong.

3.3.3 Category

This section details the category model which is comprised of a name string and encompasses a one to many association proxy to campaign models. Thus a campaign must belong to a category, and respectively, a category has many campaigns. This classification of a campaign to a category, provides the end-user with the ability to make an informed decision on the subject of advertisements that they are subscribing for. The categories are hardcoded into the system and are not modifiable by the advertisers. As the system evolves, additional categories can be created, and likewise, existing categories can be removed.

3.3.4 Location

This section details the location model which represents a longitude, a latitude, and a geocoded street address. When the user is first creating a location for a newly created campaign model, they are presented with a dynamic map container that is powered by Google Maps. Google Maps was used as through research and evaluation, it's API was the most concise and well documented of all Maps.

The dynamic map container is draggable, scrollable, and zoomable. I then attach a JavaScript event listener that listens for click events on the map. When a click event is triggered, the event handler then sends an Ajax POST request to the server. Specifically the create action of the Locations controller is invoked. For further reference, I will use the following annotation for a controller's action. Controller#action. The previous action, for example, can now be rewritten as Locations#create.

The Ajax request that is sent to the server, is comprised of a longitude and a latitude. These attributes are generated by mapping the click on the map, back to the corresponding geographic location data.

When the POST request is received by the controller, a new Location object is instantiated, and saved once it has been validated. This location is the central point of the campaign and all recipients must be located within a given radius of this point.

The web application makes it extremely easy to update and modify the location of a campaign, before the campaign begins. The edit view of the resource simple presents the user with another dynamic map object, much like the new action, with the only difference being that the existing location is displayed with a marker icon and a transparent blue area within the bounds of the circumference.

3.3.5 Advertisement

This section details the advertisement model which is the recipient facing content that is sent through to the platform core to the mobile client. When a device is being sent an advertisement, this is the content that is encoded and presented. For version one, the advertisement content can be a piece of text, though future iterations will include images and barcodes. Decoupling the advertisement from the campaign serves a dual purpose. Firstly, advertisers now have a non-public resource (the campaign description) which they can use for in-house data, and keep the advertisement content public. Secondly, this separation of concerns aids ensures that models do not become bloated by handling resources that are not directly related to themselves.

3.3.6 Ping

This section details the ping model which is associated with both the mobile client of the application and the web application. The ping is used to encapsulate the POST request of location data to the web application, and is comprised of a longitude, a latitude, the device's uuid, and a category list.

The uuid is the unique universal identifier of the handset and is used to identify the source of the incoming request as opposed the owner of the device. The category list is a list of integers, which map back to category identifiers, that inform the system which categories of advertisements that the device has opted into. A ping may have an associated campaign object.

The criteria for associating a ping with a campaign, is if it is located within the same area of as an active campaign, and also said campaign's category is in the category list. Simply, if the ping is received from an active campaign zone and the device has opted in to the advertisement category, then it is sent the advertisement content, and that campaign is associated with the ping and the uuid. The reason for association is that it allows the system to monitor the analytics of an advertisement campaign and report those back to the campaign owner. A ping that is a associated with a campaign, is known as an impression for that campaign. The number of impressions associated with a campaign, is the number of times that the advertisement content was sent to a device. It is important to keep a record of such variables, as advertisers traditionally can purchase media based on estimated number of impressions.

If the recipient opts to save the advertisement, or in application terms, add the advertisement to their wallet, then this is known as as engagement. Engagement is the number of users who opt to save the advertisement. Using these two variables, campaign owners can quickly derive how efficient a campaign was. A campaign with a high proportion of engagements against impressions can be deemed to be more successful than a campaign with little engagements against impressions.

During a campaign, the application monitors impressions and engagements, which are then presented back to the campaign owner. In future iterations of the product, the user will have the option of evaluating their campaign scores in the context of the category. This will allow the advertisers to know how their advertisements are performing against others in their category.

3.4 Mobile application

The mobile client for the Libero project is an Android application that was built using Java, JavaScript and HTML5, as opposed to traditional methods of solely Java and XML. Traditionally, applications for Android are built in Java, though recent trends are leaning towards leveraging the power of HTML5 and JavaScript to author native applications.

The framework chosen to build the client was Phonegap, an open source HTML5 mobile application framework that allows developers to write applications in markup and JavaScript, and then binds those scripts to native backend code. Essentially, JavaScript function definitions are acting as wrappers for native Java code.

Phonegap *is not the only* implementation of such a platform, but the choice to use it was encouraged by the fact that it is fully open source, under active development, and moreover, allows developers to write native code also, and occasionally it is necessary to get a more lower level access to the device than most frameworks permit. High level encapsulation works great a lot of the time, but often, especially for unique application types and speed requirements, it is necessary to write native code.

In the case of the Libero application, Phonegap did not provide access to the native *Android.notification* classes and thus, there was a requirement to use native code. The native code was written in Java and then a seperate JavaScript interface was defined.

The application was written and tested on both a HTC Hero and a HTC Sensation phone, and on Android releases 2.1 and 2.3.

The mobile client application has some core functional requirements that it must fulfill in order to be deemed usable.

- It must be able to determine it's own location.
- It must be able to periodically send that location data to a server.
- It must provide the user with a way of opting into, and opting out of advertisement categories.
- It must be able to send the category option data to the server with each location ping.
- It must be able to receive content from a remote server.
- It must be able to provide the user with a mechanism for saving and discarding advertisement.

The following section will deal with the architecture and implementation details of each of the above requirements.

3.4.1 Location

The ability for the device to determine it's own location is a core functional requirement of the mobile application. Given that the service that Libero provides is built upon location based services, a device must be location aware.

The client application uses any available resources to determine location. Such resources include WiFi, GPS, and Operator Network data. Some of these resources are more accurate than others, as is documented below.

If a WiFi connection is established, the the AP, or access point, is used to derive the geolocation data. If a GSM connection is available, then this is used, and likewise with GPS. If a device is GPS enabled, then the GPS module is used. The android operating system is provides a number of core classes to deal with location data. They can be found within the Android.Location namespace. With regards to the Libero application, I created a hook that is invoked when the application is initialized. The hook queries the current location of the device, encodes the longitude, latitude and uuid in JSON format, and sends the POST request to the server.

3.4.2 Communication

When the mobile application has queried its own location and obtained some data, it then encodes the location data, along with the device's UUID, and category watch-list into a Json structure. Json was used as the transport protocol as it is native to JavaScript, extremely lightweight as it does not bear the overhead of a typical XML document, and it is easily decoded by nearly every programming language.

This data is all wrapped in a single Ajax request and thus is non-blocking to the application, given Ajax's asynchronous nature. Once the request has been sent, the core application then determines whether or not it should send back an advertisement, depending on if there are any active campaigns within the category watch-list in the surrounding location.

If there is a campaign, a Json representation of the campaign's customer facing content is returned to the device, otherwise, nothing is returned. If content has been sent back to the device, a system notification (a small icon and message at the top of the Android UI) is created, and the phone vibrates. If the phone is not on silent, it will also beep, much like the receipt action of a text message. As mentioned previously, when the user receives an advertisement, a system notification is created. When the user selects the system notification, the Libero application is brought to the foreground and the user is presented with the home screen. On the platform side, the receipt of an advertisement is known as an impression for the advertisement's campaign and directly correlates to the number of users that seen the advertisement.

The user can then choose to engage with the advertisement by 'adding it to their wallet', or they can choose to ignore it. Typically, a user who is interested in the advertisement will save it to their wallet for future usage. On the larger scale of things, this is known as an advertisement engagement.

In addition to querying the location, sending it to the remote server, and receiving available advertisements, the mobile client also allows the user to choose which categories of advertisements they wish to receive.

The user, when modifying categories, is simply presented with a list of all categories and each has an associated check-box. All are checked by default and the user can then decide to uncheck the categories that are not interested in. Upon committing the changes for update, a request is sent to the server of the changes and the device's UUID. These changes are then persisted to the backend of the platform.

3.5 Platform core

The above sections deal explicitly with the web application and the mobile application. Physically, these are the only components of the system, though I will now explain what I refer to as the platform core.

The web application, used by the advertisement owners and media buyers, allow for the creation of advertisements and viewing of reports. The mobile application, used by the advertisement recipient, allows the user to toggle whether or not to send their location and to receive ads. The middle ground between the two, is known as the platform core.

The platform core is the area of the platform that receives the 'pings' from the mobile device, filters through the active adverts for ones matching the location data of the ping, encodes the advert content in JSON, sends the response directly back to the device, increments the impression counter, and when necessary, increments the engagement counter.

To accomplish the above, a single RESTful route is exposed that bypasses the authenticity validations provided by the Rails framework, which ensures that the incoming request does not need to forge agent types or authenticity tokens. This is achieved by skipping the "protect_from_forgery" filter in the before_filter chain. Once the request has been routed to the Pings Controller, the create action (Pings#Create) is then invoked with the location data and uuid parameters. In production, this is immediately persisted to the RDBMS. As I briefly touched on earlier, a Ping Model has the following attributes id:integer longitude:float latitude:float uuid:varchar(255) campaign_id:integer.

Id is simply a primary key for lookups, longitude, latitude and uuid are required attributes that must be present before the record can be persisted. The campaign_id foreign key is not a requirement, and will only be populated if an advertisement has been sent to the device. The criteria for this is that the location data sent by the device, is below the maximum threshold of the campaign's radius surrounding the central point of the location. For the time being, all campaigns are given a default radius that surrounds the point of creation. An interesting billing model that I intend to apply, is a dynamic range that allows the media buyer to pay more for a larger area.

Once a collection of campaigns have been found, that have fulfilled the requirements, they are then clustered into a single data structure. This data structure is compacted down to include only campaigns that are categorized in categories that the user has opted in the receive advertisements from. Version one of the application ensures fair selection chances by randomizing the order of the remaining elements, and then selecting the first element. In this way, all candidate campaigns had the same chance of being sent to a device.

When the advertisement is sent back to the device, the device beeps (if the handset is not silenced), it vibrates, and then creates a system status notification which resides in the status bar of the android UI. When the user responds to the status bar notification, they are presented with a notification that identifies itself as having been sent by Libero, it also is selectable such that when the user selects the notification icon, they are directed to the Libero application on their device.

It is at this point that that the user can view the advertisement, opt to save it, otherwise known as adding it to their wallet, or they can opt to discard it. A user will save an advertisement that they are interested in, and discard the advertisements that do not interest them. An advertisement of interest, that is added to the user's virtual wallet, is known as an engagement for that campaign. It indicates, to the advertisers, that the user was interested in their advertisement, and plans to engage with it.

3.6 Conclusion

This section detailed the architecture and implementation of the framework from proof of concept through to version one. The initial proof of concept required that the mobile client have the ability to obtain its location and to send that data to a remote server, which in turn, would log and store the data. From the proof of concept, the findings required implementation changes to the mobile client which were documented and developed.

By the end of development, a framework had been developed that provided advertisers with the ability to create advertisements through a web interfaces. This framework allowed users of the mobile application to have advertisement dynamically disseminated to their device when their location data was within the circumference of an active advertisement zone. The core requirements for such a framework were fulfilled during the architecture and implementation phase and also highlighted areas were further research and works could be carried out to further improve the framework; specifically in the area of mobile computing. These finding can be found in the conclusions section of this thesis.

4 Evaluation and results

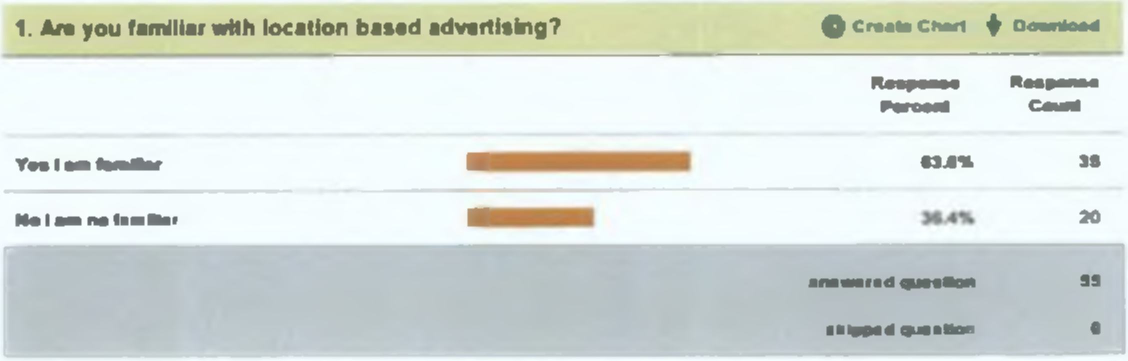
4.1 Overview

This chapter is involved with the evaluation of the project. Initially this chapter investigates the responses of a pre-build market research survey that was given to potential end-users of the platform. Following the response analysis, the chapter then goes on to evaluate the platform architecture and provide an account of the platform's testing. Finally, the chapter concludes with the evaluation outcome which then leads into the future work intentions for the system.

4.2 Market research survey

The market research survey for the Libero platform was drafted and sent out to potential end-users before the requirements of the system were created, with the intention of using the responses to drive the platform requirements. The survey was composed of 8 questions and was completed by 55 people. This section will display the original 8 questions along with the percentage and counts of responses.

The outcomes derived from each question and answer pair are documented following the answer analysis.



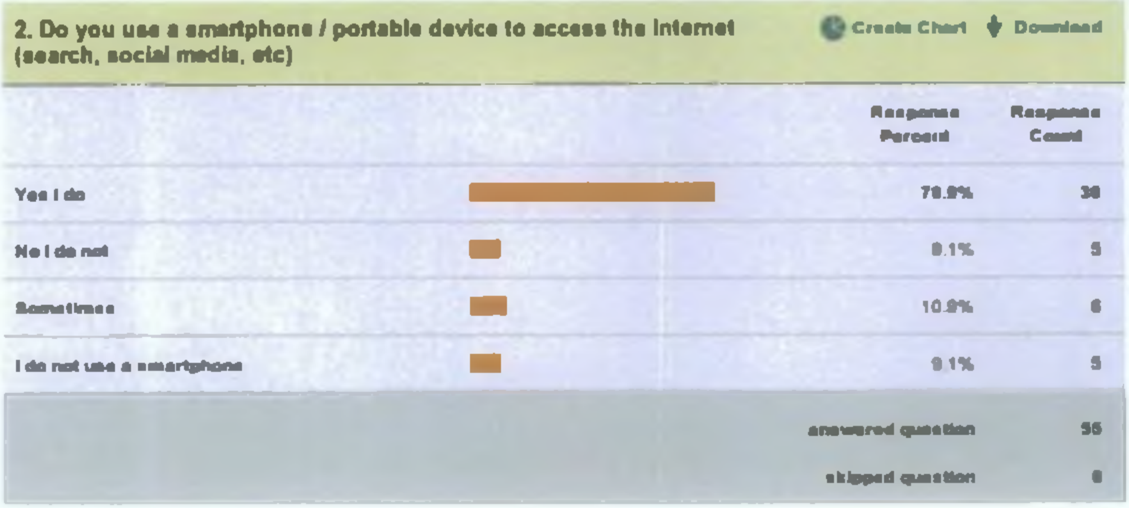
1. Are you familiar with location based advertising?

		Response Percent	Response Count
Yes I am familiar		63.6%	35
No I am no familiar		36.4%	20
		answered question	55
		skipped question	0

Figure 3: Market research survey question 1.

Question 1 was a broad question that was used to classify the audience responding into two categories. The first category being those that were familiar with location based advertising, and the second category housing those that were not.

From the responses, it is clear that 63.6% of the respondents feel that they are familiar with the area of location based advertising. It is also clear that 36.4% feel that they are not familiar with the area. With location based advertising being developing technology, it is great to know that people are already aware of the topic, and with regards to the audience of a survey, the majority are aware, or have some previous knowledge of the subject which may lead to more meaningful answers.



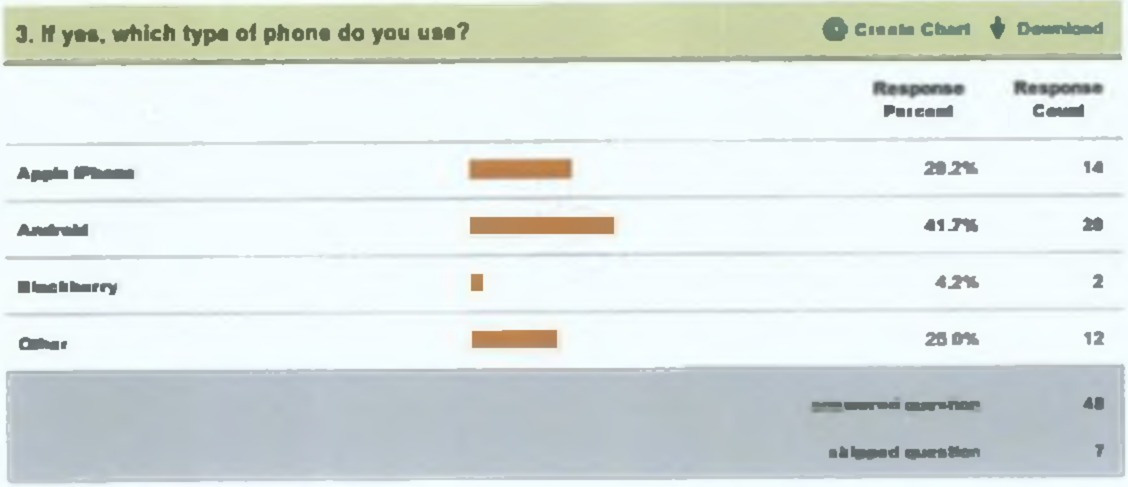
2. Do you use a smartphone / portable device to access the Internet (search, social media, etc)

	Response Percent	Response Count
Yes I do	70.9%	39
No I do not	9.1%	5
Sometimes	10.9%	6
I do not use a smartphone	9.1%	5
	answered question	55
	skipped question	0

Figure 4: Market research survey question 2.

Question 2 was designed to determine what percentage of the audience would be enabled to use the service, which is is targeted at smart-phone owners.

As mentioned within the literature review, Silicon Republic [Kennedy, 2011]stated that their study determined that smart-phone usage in Ireland was increasing. The responses to the above question indicate that smart-phone usage is in the high third quartile. This result is composed of those who continually (a) or periodically (c) use a smart-phone to access the Internet. 18.2% of the audience would not currently be eligible to use the platform as they do not use / own a smart-phone.



3. If yes, which type of phone do you use?

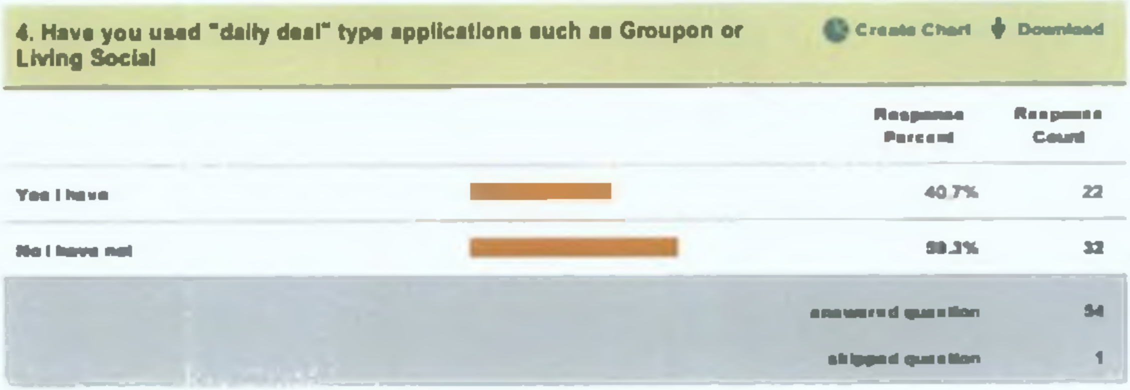
	Response Percent	Response Count
Apple iPhone	29.2%	14
Android	41.7%	20
Blackberry	4.2%	2
Other	25.0%	12
answered question		48
skipped question		7

Figure 5: Market research survey question 3.

Question 3's responses were comprised of those who answered (a) or (c) to question 2.

This question was designed to determine which of the popular handset operating systems were being used by the audience. As we can see, the Android operating system was leading the poll with 41.7% of respondents using that particular OS. Apple's iOS operating system followed with 29.2%, whilst Blackberry and others accounted for a quarter of the responses. Conclusions derived with regards to the survey was that whilst it is clear that Android led the poll, it would have been worthwhile to include some other options that the 25% of 'other' responses could have chosen.

With regards to the Libero application, it was concluded that the mobile client would be built initially for the Android operating system as the market survey showed that Android held the market share.



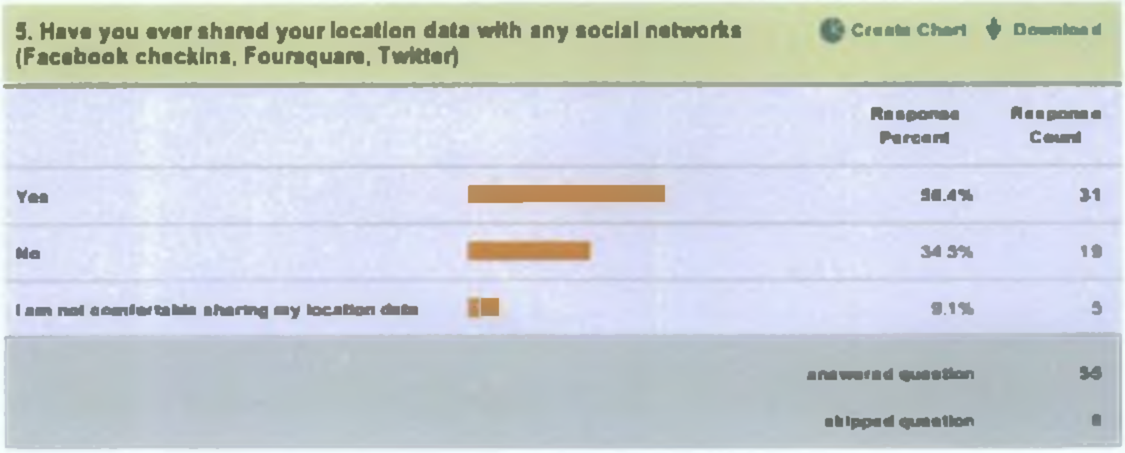
4. Have you used "daily deal" type applications such as Groupon or Living Social

	Response Percent	Response Count
Yes I have	40.7%	22
No I have not	59.3%	32
answered question		54
skipped question		1

Figure 6: Market research survey question 4.

Question 4 was used to determine if any of the respondents had ever used 'daily deal' type services such as Groupon of Living Social. The reasoning behind this questions was due to the fact that such services are not only quite new, and determining a take-up number for them may correlate to a take-up number of location based advertising, but also that these services are essentially a push-model advertising platform and knowing their user base numbers would allow us to classify the audience into 2 categories; firstly, those that subscribe to deal and advertisement services, and then those who do not.

From the responses, we can see that the majority of respondents have not used such services.



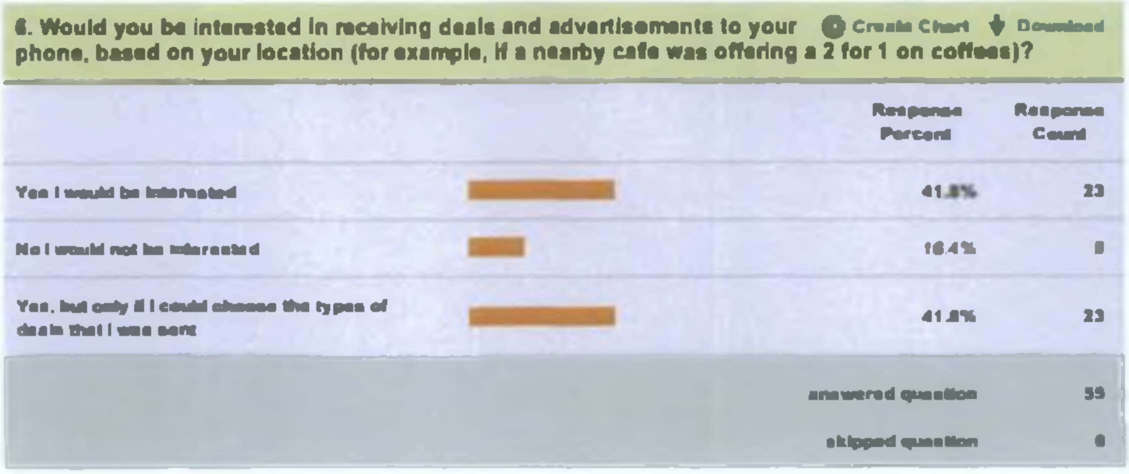
5. Have you ever shared your location data with any social networks (Facebook checkins, Foursquare, Twitter)

	Response Percent	Response Count
Yes	56.4%	31
No	34.5%	19
I am not comfortable sharing my location data	9.1%	5
	answered question	55
	skipped question	0

Figure 7: Market research survey question 5.

The core of location based advertising is having the ability to determine a user's location data. Having the ability to determine the location data is not solely enough to enable the platform, and we must obtain the permission of the user to collect and process their location data. These suggestions were made by Clarke in his work on Regulating Location Based Advertising.[Clarke, 2011]

The responses however, will aid us in determining if the audience are comfortable sharing their location data with social networks and web applications. Interestingly enough, the majority of users have shared their location data with social networks and only 9.1% said that they were not comfortable sharing their data. Using this data, we could assume that the majority of people are comfortable sharing their geolocation data with third party applications.



6. Would you be interested in receiving deals and advertisements to your phone, based on your location (for example, if a nearby cafe was offering a 2 for 1 on coffees)?

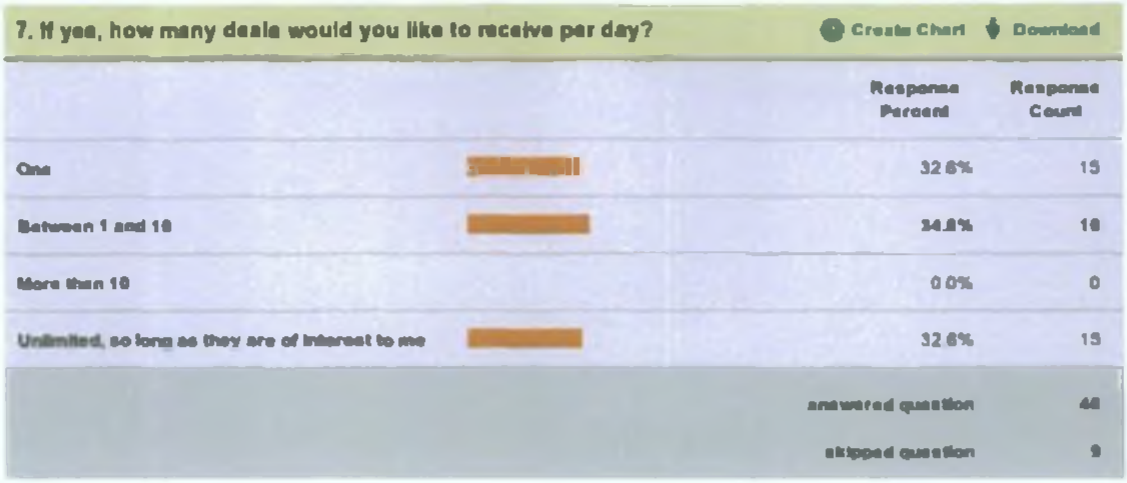
		Response Percent	Response Count
Yes I would be interested		41.8%	23
No I would not be interested		16.4%	9
Yes, but only if I could choose the types of deals that I was sent		41.8%	23
		answered question	55
		skipped question	0

Figure 8: Market research survey question 6.

The above question indirectly asks the audience if they would be interested in using the Libero platform. The results of this question would loosely correlate to the perceived feasibility of the platform and from the results above, we can see that 83.6% of the audience would be interested in using the Libero platform, and half of those interested would only use the platform if they had control over the data that they were being sent.

The conclusions derived from the responses are that the service must entail a 'pull' model that places the element of control on the user, and moreover, the platform must classify advertisements into various categories and allow the user to opt-in to and opt-out from the various categories. The results indicate the feasibility of such a platform and begin to drive the platform requirements.

Though we still need a way of determining the quantity of advertisements that a user is comfortable to receive on a daily basis. This is answered by the responses to the next question.



7. If yes, how many deals would you like to receive per day?

	Response Percent	Response Count
One	32.6%	15
Between 1 and 10	34.8%	16
More than 10	0.0%	0
Unlimited, so long as they are of interest to me	32.6%	15
	answered question	46
	skipped question	9

Figure 9: Market research survey question 7.

32.6% of the respondents indicate that they would like to receive only one advertisement per day from such a platform. 34.8% indicate that they would be happy to receive between one and ten advertisement per day and 32.6% indicate that they would be happy to receive any number of advertisements so long as they are of interest to them. What is interesting its that 0% of the respondents were happy to receive an unlimited number of advertisements without the guarantee that they would be on interest to them.

Conclusions derived correlate to the previous conclusions in that advertisements must be classified to allow category level opt in and opt out.

It was also determined that although the initial implementations of the Libero platform would place the quantity control on the platform side, later versions for GA (General Availability) would move this control to the client and allow the end user to dynamically modify the quantity of receipt.

The final question was an invitation for users to provide any other feedback on such a system. It was an optional question and there were 13 responses to it.

Below are some of the responses

- *Sounds like a great idea!*
- *It would be good for the likes of nightclubs or bars to promote offers*
- *User selection of discounts etc is totally the key for me - if I can program in my favourite shops and restaurants and get offers only from them, then I'm in!*

- *Would be handy enough but would need to be heavily moderated as there's a fine line between an offer or 2 and spamming ones phone with lots of unwanted nonsense.*
- *I find it invasive and very frustrating to recv txtxs advertising a product or a discount on flights, etc (through Meteor) or when I enter such sales environments as Dundrum Shopping Centre.*
- *Good idea especially if the service was available on holidays outside of ireland.if i had a smart phone i would be interested in location based advertising but within limits that would not affect other applications*
- *I would also like to easily stop my subscription*

An interesting point made above was that some users may find it useful to be able to filter advertisements at the brand level as opposed to the category. Others were able to visualize the platform in use as a way of promoting nightclub discounts.

One the negative side, one respondent made the point that they find it evasive to receive such data. Overall, it is concluded that there is a market for such a platform and the majority of the responses indicated that a pull model architecture with the user in point of control would be preferable.

These responses drove the requirements for the Libero application. These requirements have been included in the appendix of this dissertation.

4.3 Testing

4.3.1 Web application

The web application was developed in a TDD (Test Driven Development) style throughout. TDD is a tradition that has long been emphasized within the Ruby community. TDD required that before any code is written for the domain logic or application, we first write unit tests that define the behaviour of the models within the framework.

Below is an extract of the code used to drive development of the the campaign model.

It uses Factories as a replacement for Fixtures for test data setup.

```
class CampaignTest < ActiveSupport::TestCase
  def setup
    @campaign = Factory(:campaign)
  end

  test "a campaign must have a name" do
    @campaign.name = nil
      assert !@campaign.valid?
  end

  test "a campaign description is optional" do
    assert @campaign.valid?
      @campaign.description = "Optional description"
    assert_equal true, @campaign.valid?
  end

  test "a campaign must have a start date" do
    @campaign.start = nil
    assert_equal false, @campaign.valid?
    @campaign.start = Time.now
    assert_equal true, @campaign.valid?
  end
end
```

The above code snippet will instantiate a Campaign object before each test function is invoked. The test function accepts a String denoting the test description, and then a block. The code within the block is executed and the assertions will pass or fail.

Initially we will write the above code and run the test suite to ensure that the tests are failing, as we haven't yet wrote the domain logic code.

The next thing we do is write the domain logic and run the test suite to ensure that the tests are passing. Once the tests are passing, we can then go and refactor the code.

This practice is know as test-driven development and is much practiced by developers practicing agile development.

The purpose of a unit test is to test one unit of code. In the case of MVC, a unit test generally refers directly to a particular model, and simple references other model through the associations proxy when necessary, though this type of behaviour is generally tested within the integration tests test-suite.

By creating a detailed test-suite, that we run before we check code into version control, we can ensure that there is no divergence within the code-base.

4.3.2 Mobile client

The requirements of the mobile client was to access the device's geolocation data, periodically ping that data to a remote server, and if necessary, be able to receive advertisement responses from the server.

The client runs in the background of the device and should be unobtrusive to the user until there is information to disseminate.

Upon development of the client, the core testing to ensure that it fulfilled the requirements was to take it on a test-drive and analyze the results.

As I mentioned in the prototype subsection within the architecture and implementation section, the application was installed onto 2 Android devices, HTC Hero and HTC Sensation, each with a different version of the operating system. Android 2.1 and Android 2.3 respectively. The user then commuted to work and the results were logged to a remote web service.

Initial testing found that the application was not running in the background when the device would enter sleep mode. This meant that location data was only sent to the user when they were using the phone. This did not fulfill the requirements of the mobile client so the code needed to be rewritten.

Upon redevelopment of the code, the application was given the functionality to override the superclass' OnPause event which allowed it to run in the background.

Below is the section of code that did just that.

```
@Override
public void onPause() {
    super.onPause();
    super.appView.loadUrl("javascript:try{
                                    PhoneGap.onResume.fire();
                                    }catch(e){};");
    super.appView.resumeTimers();
}
```

The above code was then retested by loading the updated application back onto the handsets and again testing the user's commute to work.

The application was instructed to send the location data every minute, and this time, it did ping the location every minute.

The device sent a Json data structure comprised of the handset's unique id, longitude and latitude.

The pings received were then mapped out on Google Maps and the trail generated was cross referenced with the user's route, which traversed 7 train stations from the North of Dublin, to the city centre.

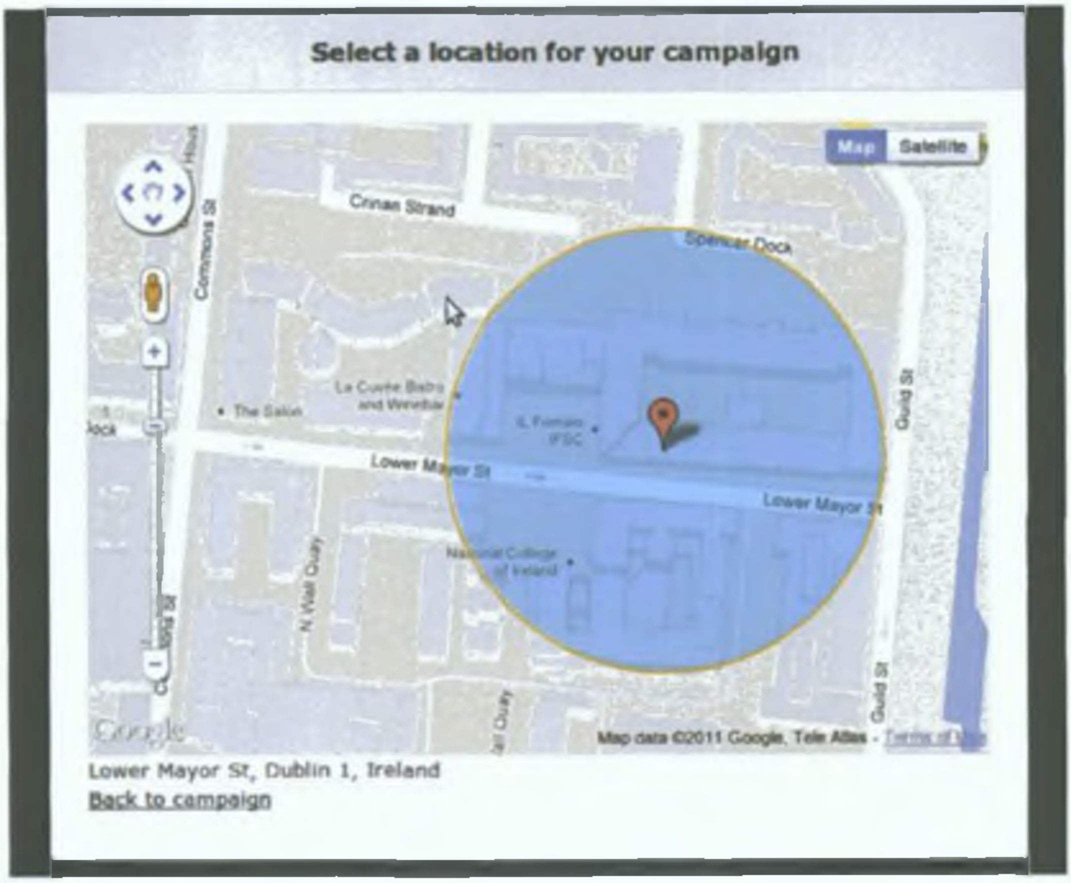
The findings were accurate at any given point with a 50 metre each-way radius taken into account when the application was using the mobile device's network operator to triangulate the position.

4.3.3 Platform core

The platform core was tested by simulating advertisements being created at known points throughout the city, and traversing through those points with the application activated on the handset.

The platform, due to the ~50 metre 'each-way' accuracy of the mobile operator triangulation set a 100 metre radius.

Below is a snippet of the "Create a campaign location" form that shows the 100m radius that surrounds the advertisement central location.



Select a location for your campaign

Lower Mayor St, Dublin 1, Ireland

Figure 10 - Web application, new location screenshot

The tester then traversed through this location and ensured that an advertisement was received. Upon receipt of an advertisement, the campaign impression count was to be incremented and if the user chose to 'Add to wallet', then the engagement count too needed to be incremented.

The ratio of impressions to engagement is then used to measure the efficiency of that campaign.

Once this section of the platform was deemed working, the next section was concerned with allowing the user to enable and disable categories.

Once the implementation was completed, the testing was much the same as before except this time we predefined categories of interest and ensured that only advertisements belonging to that category were returned to the tester.

4.4 Conclusion

The Libero platform is primarily comprised of an unobtrusive mobile client and a web application. The platform glues the pair together.

The web application is bound with a test suite of unit tests, integration tests, and functional tests. This test suite must be passing before domain code can be committed to version control. This automated testing helps to keep the application defect free and prevents divergence.

The mobile client required manual testing and once installed onto a client device, the client physically needed to travel around and traverse through active advertisement campaign zones. These interactions with the zones were then monitored by a remote web service.

The problem with this type of testing is that when bugs are encountered, they cannot be immediately tested as the application needs to be reinstalled, and the client needs to test again by physically traversing the regions. This process takes time and is not efficient so development has begun to create a remote deployment interface where the user can have the application update remotely and the developers can make changes to the code without the need to reinstall onto the device.

With regards to the platform itself, the research question has been answered. It is in fact possible to use current technologies to build a platform for location based advertising. The framework has not been released to the public yet, but the market research and direct public response has indicated that there is a market for such a platform.

The requirements for release would be to finalize the billing model(s) that can be applied, and then release the web application to advertisers. The mobile client needs to be deployed to the Android marketplace where users can easily download it to their device.

In addition to the above, the mobile client need to be tested on other mobile operating systems such as iOS and Blackberry, though the Phonegap platform is OS agnostic and should take care of the native binding.

5 Conclusions and future works

5.1 Introduction

This dissertation was concerned with the research question

> Can current technologies, social acceptance and culture of this generation be utilized to create a framework for location based advertising?

The Libero project has shown that it is in fact possible to build a framework for location based advertising using current technologies, coupled with the social rationale and culture of this generation. The project took into account many finding and suggestions in the area and extended upon some of those suggestions.

The market research that was conducted during the early stages of the project life-cycle, has shown that there is a market for such a platform, and that users are willing to share their location data in exchange for meaningful, location-aware advertisements and offers.

Through the lifecycle of the project, research and works were performed into the areas of web application development, mobile clients and computing, and the framework platform core. Future works have been identified and are detailed in the preceding section.

5.2 Future works

5.2.1 Introduction

This sections investigates future works and research that can be performed by extending onto this thesis. This section has been divided into three sections that each represented the core systems of the framework; web application, mobile client, and platform core. In addition, there are suggestions made towards the area of remote mobile application versioning and management.

5.2.2 Web application

The web application is the subsystem of the platform that is aimed towards the advertiser and is not intended to be used by the end-user (client).

The web application fulfills the requirements of the framework by providing an interface through which the advertisers can manage advertisement campaigns and view the results and reports of those campaigns, but there is still some functionality that could be added to improve the usability and the feasibility of a business model.

It should provide *billing* plans for the advertisers and implement recurring billing via one of the many payment gateways such as.PayPal or Realex.

Further to this, some of the testers requested that a campaign should have multiple advertisements and locations. Currently the application binds a one-to-one mapping between campaign and advertisement. Though some of the potential advertisers spoken to had advised that it would make more sense for a campaign to be comprised of multiple advertisements.

This feature does not prevent the application from going live nut it would be a nice feature to implement over the coming weeks.

5.2.3 Mobile client

The mobile client could provide 'sharing' functionality when an advertisement is received. This would allow users to drive up the impressions and engagement of the advertisements by sharing the advertisement with friends and connections. The requirement for this is that the user authenticate the application to access social networks on their behalf, for example, data could be shared on Facebook using their Open Graph API.Another additional feature that would be nice for the mobile client would be providing directions from current location to advertisement location once the advertisement has been engaged.

In addition to adding features and improving usability of the client, significant work and research could be performed into the area of mobile client management, specifically when the application has not been deployed via an application deployment system such as Google Market[Android, 2011] or the Apple Appstore[Apple, 2011]

Presently, when applications are deployed via one of the above systems, developers can release newer versions and deploy patches but those changes only become live once the user accesses the above systems and opts to update. This is not done as easily when the application file has been deployed manually to a device. The benefits of deploying manually is that the developer does not need to pay registration and recurring fees to the distributor, nor do the applications need to be approved by distributor teams. A mechanism that would allow applications to determine if an update was available and update accordingly, unless the update jeopardizes the security of the phone would a great tool for all mobile developers.

5.2.4 Platform core

The platform core does not currently prevent multiple advertisements from choosing the same location for advertisement campaigns. An interesting billing model that was initially chosen to be applied to the platform was that of competitive bidding between advertisers for prime location. The biddings could be used to ensure that only a single advertisement within each category is active at a given location at any given time.

Similarly, on the billing front, the platform currently provides a default radius of 100 metres around a location point. The platform could however, choose to apply a dynamic radii to a campaign and charge a higher fee for a higher area of coverage.

5.3 Conclusion

The area of location based advertising provides a range of possibilities and services and we will no doubt see an increase in the number of location aware services that are released over the coming years. This thesis has not only highlighted the possibilities of coupling current technologies with social acceptance and culture, but also highlighted areas where further research is required to provide a more efficient framework.

6 Appendix

6.1 Abbreviations / Acronyms

GPS: Global Positioning System

SIM: Subscriber Identify Module

AP: Access Point

API: Application Programming Interface

OS: Operating System

UUID: Universal Unique Identifier

PII: Personally Identifiable Information

CRUD: Create-Read-Update-Delete

Rails: Ruby on Rails

DSL: Domain Specific Language

HTTP: Hyper Text Transfer Protocol

MVC: Model View Controller

RDBMS: Relational Database Management System

DRY: Don't Repeat Yourself

TDD: Test Driven Development

ORM: Object Relational Mapper

SQL: Structured Query Language

REST: Representational State Transfer

UI: User Interface

JSON: JavaScript Object Notation

6.2 List of figures and diagrams

Bibliography

[1GoodReason, 2011] 1GoodReason (2011). :http://www.1goodreason.com/blog/blog/2010/10/22/fa missing-culture-privacy/. Accessed August 15,2011.

[Android, 2011] Android (2011). :http://www.android.com/. Accessed August 1,2011.

[Apple, 2011] Apple (2011). :www.apple.com/iphone/apps-for-iphone. Accessed August 4,2011.

[Banerjee and Dholakia, 2008] Banerjee, S. S. and Dholakia, R. R. (2008). Mobile advertising: does location-based advertising work? *International journal of mobile marketing*, 3.

[Barwise and Strong, 2002] Barwise, P. and Strong, C. (2002). Permission-based mobile advertising. *Journal of interactive marketing*, 16.

[Basecamp, 2011] Basecamp (2011). :http://www.basecamphq.com/. Accessed August 1,2011.

[Bausch and Han, 2006] Bausch, S. and Han, L. (2006). Social networking sites grow 47 percent, year over year, reaching 45 percent of web users.

[Brain, 2011] Brain, M. (2011). :http://blogs.howstuffworks.com/2011/01/18/todays-uproar-facebook-broadcasting-private-phone-number-and-address-data/. Accessed August 15,2011.

[Bruner II, 2007] Bruner II, G. C. (2007). Attitude toward location-based advertising. *Journal of interactive advertising*, 7.

[Caffrey, 2000] Caffrey, J. J. (2000). *Wireless location in CDMA cellular radio systems*. Kluwer Academic.

[Clarke, 2011] Clarke, P. (2011). Regulating location based services.

[Cleffand and Gidofalvi, 2008] Cleffand, E. B. and Gidofalvi, G. (2008). The legal aspects of a location-based mobile advertising platform. *International Journal of Intellectual Property Management*, 2.

[Cuneo, 2005] Cuneo, A. Z. (2005). Marketers get serious about the 'third screen'. *Advertising Age*, 6.

[CURL, 2011] CURL (2011). :http://curl.haxx.se/. Accessed August 1,2011.

[Driscoll, 2001] Driscoll, C. (2001). Are consumers interested in wireless internet location-based services? *20*, RCR Wireless News.

[Ducoffe, 1995] Ducoffe, R. H. (1995). How consumers assess the value of advertising. *Journal of current issues and research in advertising*, 17.

[Facebook, 2011] Facebook (2011). `http://facebook.com/`. Accessed August 1, 2011.

[FacebookPlaces, 2011] FacebookPlaces (2011). `http://www.facebook.com/about/location/`. Accessed August 19, 2011.

[Foursquare, 2011] Foursquare (2011). `http://foursquare.com/`. Accessed August 1, 2011.

[Fowler, 2002] Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison Wesley.

[Gezici et al.,] Gezici, S., Kobayashi, H., and Poor, V. H. A new approach to mobile position tracking.

[Google, 2011] Google (2011). `http://google.com/`. Accessed August 1, 2011.

[GoogleMaps, 2011] GoogleMaps (2011). :http://maps.google.com/. Accessed August 1,2011.

[Gowalla, 2011] Gowalla (2011). `http://gowalla.com`. Accessed August 25, 2011.

[Grattan, 2002] Grattan, E. (2002). M-commerce: The notion of consumer consent in receiving location-based advertising. *Canadian Journal of Law and Technology*, 3.

[Groupon, 2011] Groupon (2011). `http://groupon.com/`. Accessed August 1, 2011.

[Haghirian and Madlberger, 2005] Haghirian, P. and Madlberger, M. (2005). Consumer attitude toward advertising via mobile devices - an empirical investigation among austrian users. *Proceedings of the European Conference on Information Systems*.

[Heroku, 2011] Heroku (2011). :http://heroku.com/. Accessed August 1,2011.

[Hristova and O'Hare, 2004] Hristova, N. and O'Hare, G. (2004). Ad-me:wireless advertising adapted to the user location,device and emotions. *37th Hawaii Int. Conf. on system sciences.*

[Jing et al., 2011] Jing, N., Yao, Y., and Ru, Y. (2011). Modeling and analyzing user contexts for mobile advertising. *International Journal of Handheld Computing Research*, 2.

[jQuery, 2011] jQuery (2011). :http://jquerymobile.com. Accessed August 16,2011.

[Kaasinen, 2003] Kaasinen, E. (2003). User needs for location-aware mobile services.

[Kauffman, 2004] Kauffman, M. (2004). Location-based services. *The Morgan Kaufmann Series in Data Management Systems*, 1.

[Kelley et al., 2010] Kelley, P. G., Bensich, M., Cranor, L. F., and Sadeh, N. (2010). When are users comfortable sharing locations with advertisers.

[Kennedy, 2011] Kennedy, J. (2011). :http://www.siliconrepublic.com/business/item/21782-o2-says-smartphone-usage-up/. Accessed August 1,2011.

[Kupper, 2005] Kupper, A. (2005). *Location based services, Fundamentals and Operation.* Wiley.

[Lin, 2010] Lin, Y.-L. (2010). The types, industry chain and applications of mobile advertising.

[LinkedIn, 2011] LinkedIn (2011). :http://www.linkedin.com/. Accessed August 1,2011.

[Liu, 2008] Liu, L.-k. (2008). A study on the consumer attitudes toward location-based mobile advertising.

[LivingSocial, 2011] LivingSocial (2011). :http://livingsocial.com. Accessed August 2,2011.

[Mauw,] Mauw, S. Location-based services: a short introduction with a focus on security and privacy.

[McCormick, 2006] McCormick, A. (2006). Who do advertisers want to see their ads? *New Media Age*, 24.

[Merisavo and Kajalo, 2007] Merisavo, M. and Kajalo, S. (2007). An empirical study of the drivers of consumer acceptance of mobile advertising. *Journal of interactive advertising*, 7.

[Meteor, 2011] Meteor (2011). :http://meteor.ie/. Accessed August 1,2011.

[MindshareIrl, 2011] MindshareIrl (2011). :http://sociable.co/2011/02/09/mobile-ireland-2011-smartphone-usage-in-ireland-infographic/. Accessed July 19,2011.

[NewsoftheMedia, 2011] NewsoftheMedia (2011). `http://newsofthemedia.com/2011/06/location-based-services-worth-10-billion-2016/`. Accessed August 4, 2011.

[Okazaki and Barwise, 2009] Okazaki, S. and Barwise, P. (2009). Has the time finally come for the medium of the future? research on mobile advertising. *Journal of Advertising Research*, 51.

[Partridge,] Partridge, K. Activity-based advertising, techniques and challenges.

[Phonegap, 2011] Phonegap (2011). :http://www.phonegap.com/. Accessed August 1,2011.

[Rails, 2011] Rails, E. (2011). :http://edgeguides.rubyonrails.org/. Accessed August 1,2011.

[Rashid et al.,] Rashid, O., Coulton, P., and Edwards, R. Providing location based information/advertising for existing mobile phone users. *Special Issue: Papers from the Fourth International Conference on Mobile Business*, 12.

[Rimkus,] Rimkus, A. Location commerce and privacy. *Wireless Review*, 17.

[Rodden et al.,] Rodden, T., Friday, A., Muller, H., and Dix, A. A lightweight approach to managing privacy in location-based services.

[Ruby, 2011] Ruby (2011). :http://www.ruby-lang.org/en/. Accessed August 1,2011.

[RubyonRails, 2011] RubyonRails (2011). `http://rubyonrails.org/`. Accessed August 1, 2011.

[Rust et al., 2002] Rust, R. T., Kannan, P., and Peng, N. (2002). The customer economics of internet privacy. *Journal of the academy of marketing science*, 30.

[Sinatra, 2011] Sinatra (2011). :http://www.sinatrarb.com/. Accessed August 1,2011.

[StarbucksFoursquare, 2011] StarbucksFoursquare (2011). `http://mashable.com/2010/03/11/foursquare-starbucks/`. Accessed August 12, 2011.

[Technorati, 2011] Technorati (2011). :http://technorati.com/blogging/article/20-million-thoughts-on-foursquares-success/. Accessed August 1,2011.

[Thorson, 2000] Thorson, E. (2000). The interactive advertising model: How users perceive and process online ads. *Journal of Interactive Advertising*, 1.

[Troutman and Timpson, 2008] Troutman, M. and Timpson, S. (2008). Effective optimization of web sites for mobile access: The transition from ecommerce to mcommerce. *Journal of interactve advertising*, 9.

[Tsang et al., 2004] Tsang, M. M., Shu-Chun, H., and Ting-Peng, L. (2004). Consumer attitudes toward mobile advertising: An empirical study. *International Journal of Electronic Commerce*, 8.

[Twitter, 2011] Twitter (2011). `http://twitter.com`. Accessed August 29, 2011.

[Unni and Harmon, 2007] Unni, R. and Harmon, R. (2007). Perceived effectiveness of push vs. pull mobile location-based advertising. *Journal of Interactive Advertising*, 7.

[Xu et al., 2009] Xu, H., Oh, L.-B., and Hock-Hai, T. (2009). Perceived effectiveness of text vs. multimedia location-based advertising messaging. *International Journal of Mobile Communications*, 7.

[Xu et al., 2010] Xu, H., Te, H.-H. T., Tan, B. C. Y., and Agarwal, R. (2010). The role of push-pull technology in privacy calculus: The case of location-based services. *Journal of Management Information Systems*, 26.

[Xua et al., 2008] Xua, D. J., Liaob, S. S., and Lid, Q. (2008). Combining empirical experimentation and modeling techniques: A design research approach for personalized mobile advertising applications. *Decision Support Systems*, 44.

[Zeng, 2009] Zeng, F. (2009). Social factors in user perceptions and responses to advertising in online social networking communities. *Journal of Interactive Adcertising*, 10.