DayOut: Web and Mobile Application To Discover Deals, Events and Attractions

Karl Murray
B.A. (Mod.) Business and Computing
Final Year Project April 2014
Supervisor: Brian Gillespie

School of Computer Science and Statistics
O'Reilly Institute, Trinity College, Dublin 2, Ireland
Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

-----------------------------------------------------
---------------------------------
Name                                    Date
Acknowledgements

I would firstly like to thank my supervisor Brian Gillespie for his suggestions, guidance, and support, without which this project would not have been completed. I would also like to extend my gratitude to all my family and friends who supported me in the completion of this project.
Table of Contents

Declaration ........................................................................................................................................... 1

Acknowledgements .............................................................................................................................. 2

List of Figures ........................................................................................................................................ 6

Chapter 1: Introduction ......................................................................................................................... 7
  1.1 Aims .............................................................................................................................................. 7
  1.2 Motivations .................................................................................................................................. 9

Chapter 2: Background ......................................................................................................................... 10
  2.1 Flash Sale Sites .......................................................................................................................... 10
    2.1.1 Groupon .............................................................................................................................. 10
    2.1.2 LivingSocial ....................................................................................................................... 11
  2.2 The Rise Of Smartphones .......................................................................................................... 12
  2.3 Travel Companion Applications ............................................................................................... 14
    2.3.1 TripAdvisor ....................................................................................................................... 14
    2.3.2 Fodor’s City Guides ............................................................................................................ 15
  2.4 Conclusion .................................................................................................................................. 15

Chapter 3: Project Overview ................................................................................................................. 17
  3.1 Gathering Merchants Details ...................................................................................................... 17
  3.2 Storing The Data ....................................................................................................................... 18
  3.3 Retrieving ‘Advertisements’ ..................................................................................................... 18
  3.4 Viewing ‘Advertisements’ ......................................................................................................... 18
  3.5 Filtering User Requests ............................................................................................................. 18

Chapter 4: Design .................................................................................................................................. 19
  4.1 User Interface ............................................................................................................................. 19
    4.1.1Prototypes ........................................................................................................................... 19
    4.1.1.1 Web Application ........................................................................................................... 19
    4.1.1.2 Mobile Application ....................................................................................................... 21
  4.2 Architecture Design .................................................................................................................... 22
  4.3 Database Design ........................................................................................................................ 23
  4.4 PhoneGap Development Platform ............................................................................................ 24
    4.4.1 Ionic Framework ................................................................................................................. 25
  4.5 Software Technologies .............................................................................................................. 26
  4.6 Summary .................................................................................................................................... 27
8.1.1 Design Driven ...............................................................48
8.1.2 Available On All Platforms ........................................48
8.1.3 Tablet Optimization .......................................................49
8.2 Future Work ....................................................................49
8.2.1 Google Maps API .........................................................49
8.2.2 Update Server ..............................................................49
8.2.3 Mobile App Login ..........................................................50

Chapter 9: Conclusion ...........................................................51

References ........................................................................52
List of Figures

2.1 Groupon.ie (Home Screen)
2.2 LivingSocial (Home Screen)
2.3 LivingSocial (Merchant Center)
2.4 Geo-Fencing in current locations (city/country)
2.5 Finding attractions in Dublin
2.6 Fodor’s City Guide (Design Driven Innovation)

4.1 Web Application (Prototypes)
4.2 Mobile Application (Prototypes)
4.3 DayOut’s Architecture Design
4.4 Entity Relationship Diagram
4.5 PhoneGap Build
4.6 Visual Overview of Application Design

5.1 Fixed Menu Bar
5.2 Screenshot of regprocess.php
5.3 Registration tables.
5.4 Dashboard Home Screen
5.5 Segmented Controlling
5.6 Deals table
5.7 Two alternative views
5.8 Retrieving data from Attractions table
5.9 Deleting selected Attraction

6.1 Angular JS Architecture
6.2 Index View
6.3 AttractionService
6.4 AttractionIndexCtrl
6.5 Detailed View of advertisements
6.6 Screenshot of attraction-detail.html
6.7 Budget Finder Process
6.8 Detailed View on Home tab
Chapter 1: Introduction

Businesses have always been looking for new ways to advertise and share their products to the world and with recent technology advancements they found the best way to turn strangers into customers and promoters of their business, is through web and mobile technologies.

While businesses have been looking for new ways to advertise their products, buyers on the other hand have always been on the look for special offers with attractive prices. With the emergence of ‘flash sale’ websites in recent years both businesses and buyers have got what they wanted. Results have shown that flash sales grew at least twice as fast compared to any other online businesses [1]. Flash sale websites allow businesses to offer incredible discounts that are sometimes greater than half-off the original price, and the potential to generate hundreds to tens of thousands of sales from new customers within a single day. By attracting these huge numbers of customers, local and national merchants find it very hard to resist signing up to such sites. Not only have flash sales become popular online but also with a smartphone-driven boom in recent years. Businesses who advertise on these websites are now expecting the bulk of its sales in coming years to be carried out on smartphones and tablet computers [2].

1.1 Aims

While a significant number of flash sale applications exist, many rely on new customers to do well, as existing customers often lose interest due to offers not targeted at the individual but instead at the general public. In contrast to such approaches, this project proposes an alternative solution based on flash sale models.

The aim of this project is to develop a web and mobile application called DayOut that allows businesses to advertise and share their products (similar to flash sale sites) to people in a particular city. The aim is to supply users with an application where they can plan a day out and on another level, use it as a travel companion application to help users travelling to see the latest offers advertised by merchants in that particular city. By simply entering your budget and selecting your current city name, DayOut will bring up a list of offers for you and your friends and family to enjoy.
By targeting merchants and consumers the aim is to have them on separate platforms, giving a personalised feel for both. The web application will be targeted for merchants so they can add offers they would like to advertise and sell. These offers will be categorised under three headings – deals, events and attractions. Features will include:

1. Home Screen to display all relevant information to merchants
2. Registration page to sign up for the service
3. Sign-in form to access their personalised dashboard, which will include
   a. Screen to show merchants how to use their dashboard
   b. Library to see all their offers that have added
   c. Page to add their promotional offers which they want to sell

Along side the web application, the aim is to develop a mobile application using PhoneGap so it can easily target users across all smartphone platforms. Once a merchant adds an offer via the web application, the aim is to have the content pushed to the mobile application so local residents or visitors passing through the city can see the latest deals, events and attractions on offer. For the mobile application features will include:

1. A budget finder tab so that consumers can see what offers are available within their budget.
2. Three tabs to display the latest deals, events and attractions. Each advertisement will show the item name, photo, price and merchants name.
3. A detailed screen will be allocated to each advertisement, which will show all the information about that item provided, sent from the merchant.

This project shows that unlike existing applications, both merchants and consumers will have a separate platform and personalised feel. Merchants will have their own dashboard to view existing offers and can easily add new advertisements. On the other hand, consumers will easily be able to view all offers available on their smartphones, while also having a personal budget finder where they can narrow down their search to show exactly what they are looking for.
1.2 Motivations

The main motivation for developing this project is through personal experience. When traveling across Europe and South East Asia in recent years, one thing became evident, there seemed to be a lack of offers and advertising targeted at backpackers. This resulted in countless hours that were wasted in hostels looking for sites to see and activities to do. What DayOut’s mobile application can let you do, whether you’re traveling through a city or a local resident, is let consumers see what offers are available them.

This solution is not only great for consumers but also for small, medium and large businesses as it provides them with a platform to advertise and sell their products. There are so many great offers to avail and places to see, and the advantage of DayOut is that it bridges the gap between the merchant and the consumer. Unlike existing applications were merchant’s register to add deal-of-the-day offers based on a wide range of products (Groupon or LivingSocial). DayOut focuses more on targeted advertisement for travel goers as it can provide them with the latest places to go and things to do through merchants showcasing their products and services.

While leading traveling sites like TripAdvisor provide great information and reviews of travel-related content. It’s user-generated content has often been criticised on their mobile application for being too text heavy. Features on DayOut (both web and mobile) will show user friendly content, with matching blue and white colour schemes while representing Meyer and Utterback’s product family, showing commonality of user interfaces and design elements [3].
Chapter 2: Background

The purpose of this chapter is to discuss the concept of flash sale sites and to analyse existing models and applications, while also looking at travel companion applications. This chapter outlines some of the technologies and concepts that underlie the solution proposed by this project. After explaining the general outline of these technologies, their implementation will be discussed later in Chapter 5 and Chapter 6.

2.1 Flash Sale Sites

Flash sales is an ecommerce business model in which a website offers products for sale, generally over a period of 24 to 36 hours. Potential customers register as members, which they receive online offers and invitations by email or social networks. By allowing merchants to advertise discounted products or services directly to the customers of the ‘deal company’, this allows them to build brand loyalty and quickly sell surplus inventory. While the deal company, like Groupon or LivingSocial receives a portion of its profit. Over the years these sites have continued to grow in popularity, although new concerns have arisen over the longevity of the concept, as close to 80% of deal users are new customers [4]. This chapter will now proceed to discuss existing applications.

2.1.1 Groupon

Groupon is one of the most famous flash sale sites available at the moment in Europe and within two years of launch it served in more than 150 markets and gained just over 35 million registered users [5]. It offers one ‘Groupon’ per day in each of the markets it serves and is primarily focused on the health, fitness and beauty markets, as reports show that females take up 62% of their purchases [6]. Groupon uses something it calls "collective buying power" to sell it's products. Unless a minimum number of deals are sold through Groupon, no one can buy it [6]. Giving each buyer the motivation to promote and share the deal in order for them to get it themselves resulting in free marketing for the business and Groupon through social networks and word of mouth.

Groupons design is unlike that of its competitors. Figure 2.1 shows their home screen, which targets a single (featured) product, resulting in more traffic been driven towards it. In terms of user interface, the biggest and brightest elements on the page are the title and
the buy button. Unlike normal online retailers, these big elements when combined with the rush of the countdown clock attract impulse buying from new customers.

![Groupon.ie (Home Screen)](image1)

**Figure 2.1** – Groupon.ie (Home Screen)

### 2.1.2 LivingSocial

In 2009, LivingSocial launched a flash sales website after two years of successfully being an application developer on Facebook. Since then LivingSocial has now more than 70 million members around the world [24]. Like Groupon, LivingSocial lets merchants (local or national) sign up and add deals to their website with a main goal to help people discover, share and enjoy great daily deals. LivingSocial has become the fastest-growing company in the social-buying category – mainly due to its acquisition driven strategies [24].

![LivingSocial (Home Screen)](image2)

**Figure 2.2** – LivingSocial (Home Screen)
Figure 2.2 shows LivingSocial’s home screen. This website doesn’t just show one featured product for sale, but instead targets eight products or services that are all located in your city. With its user-friendly display and targeted advertisements, LivingSocial shows the real power of collaborative buying sites.

On the other side, it is surprising to see that LivingSocial’s merchant center is very plain and none informative. Apart from saying that is provides deal performance metrics, customer data and tracks payments – it is very unclear how merchants can use LivingSocial to advertise and growth their brand image.

![LivingSocial Merchant Center](image)

**Figure 2.3 – LivingSocial (Merchant Center)**

### 2.2 The Rise Of Smartphones

With new features, capabilities and a seemingly endless array of apps, smartphones are increasingly becoming more important to our everyday lives. People of all ages are now using smartphones as personal communication and entertainment devices, mobile pocket computers and even personal digital assistants [8]. Now businesses have started to adapt to this so-called ‘smartphone-driven boom’. It’s too no surprise that a census report was released early this year showing that smartphones contributed to over $200 billion in offline and online sales last year, with a 188% increase in retail spending on mobiles [8].
However, while smartphones contributed to over $200 billion sales in 2013, mobile location advertising firm – xAd – found that 77% of smartphone-related transactions are actually happening within the store [9]. Showing that while people are searching for things online, they are actually buying them in store. That’s why many retail marketers are relying on techniques like geo-targeting – targeting mobile ads to consumers based on their location. Common techniques that are used:

1. Loyalty Geo-Fencing: reaching target consumers based on their current location.
2. Competitive Geo-Conquering: targeting consumers that are around their competitors retail location
3. Audience Target: targeting consumer based on their demographic and geographic factors – e.g. salary, buying pattern, location

Current CEO Jacques-Antoine Granjon of Vente-privee.com – a French ecommerce company that pioneered the model of online flash sales in 2001 – said "the smartphone is the future; one day people will no longer be in front of their computers" [2]. Vente-privee.com generated 40% of its 2013 revenue (€1.5 billion) via mobile devices and expects that to rise to 90% in the next five years. Like them, Groupon and LivingSocial have taken to smartphone driven sales – using some of the techniques mentioned above. Their mobile applications, seen in Figure 2.4, show their emphasis of geo-fencing in a particular city or country to target advertisements to users.

![Figure 2.4 – Geo-Fencing in current locations (city/country)](image-url)
Figure 2.4 also shows similarity in user interface between the mobile applications. This ‘tile’ view lets buyers easily see what the title, price and most importantly what the product looks like – without having to click in to each one.

2.3 Travel Companion Applications
Travelling alone or even with friends to a new place can give people a feeling of uncertainty. For many people it can be frightening to be in a foreign country where you do not know anyone and don’t know where you should go. However, since the emergences of smartphones the use of travel companion applications has allowed people to plan, organise and manage their trips away.

2.3.1 TripAdvisor
TripAdvisor is the world’s largest travel site, enabling users to plan their perfect trip. With over 75 million reviews, TripAdvisor offers trusted advice from real travellers and a wide variety of travel choices and planning features with seamless links to booking tools.

The mobile application lets users [11]:

- Browse millions of reviews, opinions, and photos by travellers
- Find the best hotels available, including Travellers’ Choice award winners
- Explore restaurants by food type, price range, and rating
- Discover great attractions to see in any destination
- Use ‘Near Me Now’ to discover options near you or any address you enter

Figure 2.5 – Finding attractions in Dublin
Figure 2.5 shows just how easy it is to see Dublin’s biggest attractions. With a few clicks, TripAdvisor shows you a map and list view of all the user-generated content in order of ratings. The mobile applications services are completely free to users, who provide most of the content – with the website supported by an advertising business model.

2.3.2 Fodor’s City Guides

Fodor’s City Guide is another great way for people to check out the culture around them. Not only is great for informing users of popular landmarks and attractions, but also the imagery to go with it has been an important design feature for them. The aim of Fodor’s City Guide is to make the iPhone and iPad a fun place to browse and learn different cities in the world [12]. Even though the application focuses on giving users the most valuable information in a few cities with beautiful imagery throughout the application, the drawbacks are quite substantial as there are not many cities that are featured. Currently the application is only available for 22 cities worldwide.

2.4 Conclusion

It’s no surprise that with the rise of mobile applications businesses are turning to smartphones to advertise their products. While flash sale sites have grown a huge amount in recent years, most seem to have no business model and show a lack of profitability to support there growth. Groupon for example takes 50% commission on all
sales and even though it has millions of new users a year you would assume there doing extremely well. However, since its IPO in 2011 ($26), the firms share price has dropped more than 70% to $7.84 [25]. Not only has this happened to them but also many other flash sales sites. Its shortfall has been backed up by xAd’s data, which stated that while customers are browsing online for deals, most consumers go to the actual store to avail of them. This evidence has backed up the proposal of using DayOut purely as an advertising tool for merchants to showcase their products or services to potential buyers rather than pushing for in-app purchases.

By using DayOut as an advertising tool, it would result in merchant’s not having to split any profits with flash sale sites. Past results have showed that by using flash sale sites like Groupon, LivingSocial or BuyWithMe Promotions, only 55.5% of businesses reported making money, 26.6% lost money and 17.9% broke even on their promotions [4]. While close to 80% of deal users were new customers, significantly fewer users spent beyond the deal’s value or returned to purchase at full price [4]. Other improvements that will be made for the web application is the nature of having a user friendly approach to the merchants – informing businesses exactly how to get the best out of DayOut.

In terms of the mobile application, it is clear to see the lack of travel companion applications available to really get the best out of a city. While leading travel site TripAdvisor is extremely good at seeing what attractions are best rated, it doesn’t really provide any information of upcoming events in a particular city, or exciting activities for friends and families to enjoy – that’s who DayOut is aimed at. Clear technologies of geo-fencing and targeted audiences have emerged as well. By using geo-fencing in DayOut’s mobile application, it can narrow down the advertisements to the users particular location and to make the application more personalized – targeted audiences can be utilized to target consumers based on their demographic as well.

Overall this section has looked at existing flash sales sites and travel companion applications to gather insight of technologies and concepts that underlie the design of how DayOut’s web and mobile application will be designed, this is discussed further in Chapter 3 and 4. While the implementation of the applications and will be discussed later in Chapter 5 and 6.
Chapter 3: Project Overview

The purpose of this chapter is to give a general overview of the approach taken in completing the project. The majority of the design decisions for both the web and mobile application were shaped by the research carried out in the relevant industries as outlined in Chapter 2.

At the highest level the web application lets merchants perform the following:

1. Proceed through a two-page registration process, were details are securely stored on server.
2. Login to a personalized dashboard.
3. Advertise under the headings of ‘Deals’, ‘Events’ and ‘Attractions’ by filling in the required fields in the form – again stored on server.
4. View an ‘Item Library’ to see their current advertisements.

At the highest level the mobile application lets users perform the following:

1. View the latest advertisements in their particular city, under the headings of ‘Deals’, ‘Events’ and ‘Attractions’ – data received from server
2. View a detail screen of each advertisement, giving users more information – again data received from server
3. Use the ‘Budget Finder’ screen to filter advertisements that are targeted at the users demographic – depending on the users query a personalized screen will be displayed showing a list of local activates to do and see.

These high level steps can be further broken into multiple sub steps. These sub steps are outlined below and indicate the structure of both applications and the remainder of the report.

3.1 Gathering Merchants Details

When merchants opt to register with DayOut, they will firstly go through a registration process to enter their email and password. After that they will proceed to the second and
final page where they are asked to fill in more detailed information about their business – including name, contact, address etc.

3.2 Storing The Data
While merchants are registering with DayOut the content they input is pushed to the database in a server using PHP scripting, where the password is encrypted. When merchants are adding offers to the dashboard, all content is again stored on the database using PHP scripts that connects to the server.

3.3 Retrieving ‘Advertisements’
When retrieving the data on the mobile application, an Ajax call is used to send a HTTP request to a string containing a URL – this URL is a PHP script and when executed, returns an array of deals, events and attractions in JSON form.

3.4 Viewing ‘Advertisements’
Smartphone users can easily view offers on DayOut. By opening the mobile application the user has a choice whether to view the latest deals, events or attractions. An index screen is designed for each, were the user clicks in to, to see a more detailed view – showing all of information that has been retrieved from the Ajax call.

3.5 Filtering User Requests
DayOut’s mobile home screen is an efficient way to filter user requests – based on their interests and budget, a MySQL query is send to the database. This query checks what the user has selected to view and compares their budget with item prices, returning offers that are both in their interest and within their budget.
Chapter 4: Design

This chapter introduces the design approach taken and different software technologies that are used throughout the web and mobile applications, briefly discussing why each was chosen and how they were designed and integrated.

4.1 User Interface

Having made the requirements in Chapter 3, the first steps to designing the applications were creating the user interfaces. The interfaces for both applications were mostly developed through the use of human computer interactions. Learning these techniques from previous modules, this study helped the design to produce applications that are:

1. **Useful** – accomplish what is required
2. **Usable** – do it easily and naturally, without danger of errors, etc.
3. **Used** – make people want to use it, be attractive, engaging etc.

For design proposes the main focus were to make the application useful, usable and used by both merchants (web) and consumers (mobile) because many applications that aren’t, are broken by design – making ‘human error’ highly likely, and sometime inevitable [26].

4.1.1 Prototypes

Before any technologies were considered, prototypes were made to envision how merchants and consumers may use their applications. Early prototyping allowed for feedback earlier in the development cycle and made it easier to change or throw away when problems were discovered. From receiving feedback from friends, family and co-workers high-fidelity prototypes were made using Balasmiq software [27]. Unlike previous versions these prototypes were designed to give the project guidance and show (roughly) what the project would look once completed.

4.1.1.1 Web Application

Figure 4.1 shows the high-fidelity prototypes for merchants using the web application. The home screen (Screen 1) is a one-page paradox scrolling that easily allows merchants to register, sign in, look at the features of DayOut and see the ‘About Us’ section. This simple design gives a merchant everything they need on one page. If they opt to register,
the merchant will be brought to a two-page registration process (Screen 3 and 4), where they enter all the required fields about their business.

Figure 4.1 – Web Application (Prototypes)
Once registered or signed in, the user is brought to their dashboard where they can submit advertisements or view their current library. To add an advertisement (Screen 5), merchants must first select an ‘Attraction’, ‘Deal’ or ‘Event’ related offer then simply fill in the required fields to submit their offer. Merchants can also view their current offers (Screen 6) and delete them at any moment if they wish to do so.

4.1.1.2 Mobile Application

**Figure 4.2** shows the high-fidelity prototypes aimed at the consumers using the mobile application. The Budget Finder Page is the home screen of the app. This page lets users choose what they’re looking for – returning a unique result displaying all offers within the categories and budget chosen.

![Mobile Application (Prototypes)](image-url)
Figure 4.2 also shows a mockup of how advertisements will look of the application. The index page shows a list view of all the current advertisements that have been added by merchants, showing all the relevant information – item photo, name and price. If an advertisement is clicked, the user will be brought into a detailed page showing all the information that has been provided by the merchant. These two pages will also be repeated for the Events and Attractions tab.

Having looked at the high-fidelity prototypes for both applications, this chapter will now proceed to discuss the architecture design – including the platforms and technologies that were used throughout the project, turning these prototypes into live applications.

4.2 Architecture Design

Figure 4.3 shows the system architecture for DayOut. This was a perfect choice for the project as web and mobile applications have mainly been powered by the three-tier architecture. Three-tier architecture is a client–server architecture that consists of a presentation, logic and data tier – each including established technologies [13]. The presentation tier is primarily focused on the user interface, which is made up by HTML, JavaScript and CSS on both applications. Behind the presentation tier is the logic tier. This can be written in leading technologies like .NET, Java, PHP, Python and Ruby on Rails – in this case PHP was chosen. Behind the logic and presentation layers sits the data tier, which is driven primarily by relational database management servers. 000webhost.com was the chosen platform to handle this projects MySQL database as it provides top class free web hosting services.

Connecting the web application to the database is done via PHP scripting, while the mobile application is connected by HTTP requests as PhoneGap is agnostic of back-end technologies and can work with MySQL using web protocols. PhoneGap applications generally do not talk directly to a database, so the client to application server communications was based upon HTTP request for PHP services using Ajax calls with data being received in JSON form.
4.3 Database Design

Having made the high-fidelity prototypes, it was clear to see how the database was going to be designed. The database design was targeted at merchants, as they are the only users that store the information. The database was setup using MySQL to handle all query executions. MySQL, also known as “My Sequel,” is the world’s most extensively used open source relational database management system that runs as a server providing multi-user access to a number of databases [28].

Figure 4.4 shows each of the tables needed for the project to function and the table relationships that relate to each other – done through an entity-relationship model to further enhance the description of the database. This diagram represents and describes the individual entities, relationships between them and the attributes associated with them, in regards to the database. The user table is the most important table in the database because all of the other tables revolve around it. A user may only have one lot of business and service information but can add several deals, events or attractions using their dashboard.
4.4 PhoneGap Development Platform

PhoneGap is a mobile development framework run by Adobe, which enables programmer’s to build applications for mobile devices using HTML5, CSS3 and JavaScript – instead of native applications such as Java (for Android) and Objective-C (for iOS) [29]. PhoneGap provides API’s that enable developers to access native operating system functionality using JavaScript, and because these APIs are consistent across multiple device platforms and built on web standards, the mobile application should be portable to other platforms with minimal to no changes [14].

PhoneGap suits this project perfectly as DayOut’s mobile application is designed for a broad array of people (local citizens and travel goers) using different mobile platforms. It’s cross platform native development framework enables this project to write an application for varies mobile phone platforms. Instead of focusing on coding one platform, like iOS or Android, PhoneGap allows for launch across seven different platforms.
A great advantage of using this development framework is the build service (Figure 4.5). PhoneGap Build compiles everything in the cloud, leaving developers without the headache of maintaining native SDKs. This build results in mobile applications always having the most up-to-date SDK for the platform developers are targeting. Integrating with PhoneGap Build allowed this project to offer a high-end, cost-effective product with rapid prototyping features and, above all, cross-platform development capabilities [15].

![Figure 4.5 – PhoneGap Build](image)

In the past PhoneGap has got a lot of criticism for its poor functionality and design because it’s a ‘hybrid’ application and not truly native – all layout rendering is done via web views instead of the platform’s native UI framework. Where before design used to suffer, now with the use of Ionic Framework it doesn’t have too.

### 4.4.1 Ionic Framework

Ionic is an open source front-end framework for developing hybrid mobile applications using a library of mobile-optimized HTML, CSS and JavaScript components for building highly interactive apps [16]. Its framework is modelled off of popular native mobile development SDKs, making it easier to design and built hybrid applications for iOS or Android. Clean, simple, and functional, Ionic has been designed to work and display beautifully on all current mobile devices.

Using Ionic Framework allowed the design of the mobile application mirror that of iOS7. Without this the application wouldn’t be attractive or engaging to the consumers – making it look more like a web app. Ionic also utilizes AngularJS in order to create a
framework most suited to develop rich and robust applications. It not only looks nice, but its core architecture is for serious app-development, with AngularJS tying in perfectly.

4.5 Software Technologies

Below is a list of the software technologies employed throughout the project. A brief justification is included, stating why each of these technologies was chosen.

1. HTML:
   • HTML was the mark-up language used for displaying the web and mobile application.

2. CSS:
   • Used for optimizing the look and formatting of the applications.

3. JavaScript:
   • JavaScript allowed the user to easily interact with the web applications home screen, using parallax scrolling.

4. PHP:
   • Used for general web development while also providing the bridge between both applications and the MySQL database.

5. MySQL:
   • MySQL is used to store and manage the data in the database, were it is easily compatible with PHP.

6. AngularJS:
   • AngularJS provided the mobile application with a clear structure - using a MV* pattern.

7. Ajax:
   • Ajax allowed the mobile application to retrieve data from the server asynchronously without interfering with the display and behaviour of the application.

8. JSON:
   • JSON allowed efficient data interchange between the server and mobile application.
4.6 Summary

This chapter provided high-fidelity prototypes for both applications that decided the breakdown of the design architecture. In summary, the design of this project required three components. The first component was the frontend using HTML, CSS and JavaScript for both applications. The second was the backend using PHP scripting, and the third involved the system’s database (MySQL). Chapter 5 will now discuss the implementation of the web application while Chapter 6 discusses the mobile application implementation.

Figure 4.6 – Visual Overview of Application Design
Chapter 5: Implementation - Web

This chapter introduces the business side of DayOut. The web application allows merchants to register to DayOut, so they can login to add their advertisements to consumers using the mobile application. This chapter now discusses how the web application was implemented, and some techniques that were used throughout it.

5.1 Home Screen

The home page has been developed using HTML, JavaScript and CSS. The purpose of this page is to show a merchant visiting the site all the relevant information they need. A fixed menu bar (Figure 5.1) including section headers and a login link is at the top of the page, which allows users to easily navigate through the web page without worrying about their current position. When the corresponding section header has been clicked, using jQuery API’s and Stellar.js, a parallax scrolling responsive framework was implemented to give clean and functional transition between sections on the page.

![Figure 5.1 – Fixed Menu Bar](image)

5.1.1 Home

This section is the first thing merchants see when viewing the page. This section was split in two using div containers. On the left side is an image that shows the user how their advertisements will look on the mobile application, while the right side provides them with the registration link and a video explaining what DayOut is and how it can help them grow their business. The video was implemented using video.js, an open source HTML5 video player that can be customized for the developers requirements.

5.1.2 Features, About Us & Pricing

Using HTML and CSS, these sections explain what DayOut is, why it was developed and who is was developed by, plus letting users know the pricing of DayOut. The Features section includes three icon images – each for deals, events and attractions. This was developed so that when a user hovers over an image, it would turn blue (like if it had been selected on the mobile application) making it attractive an engaging for the user.
5.2 Registration

When a merchant clicks the ‘Get Started’ button that’s on the applications home page, they automatically get redirected to the registration page, were they go through a two-page process.

5.2.1 Application

On the application side the look has been optimized to give merchants an easy and clear registration process – implemented using HTML, CSS and PHP (Exhibit 2). Colours, positioning and button sizes were all carefully considered in this process, to minimize human error in the process [17]. The first page is purely for merchants to create the account by registering their email address and password. The next and final page contains more information merchants need to input. Here they enter details regarding their business – name, contact number, address and their chosen industry.

5.2.2 PHP (Server side)

When the merchant fills out their details on both registration pages and clicks the submit button, the data is sent for processing to PHP scripts named regprocess.php and information.php. regprocess.php shown in Figure 5.2, does the following:

1. Makes a connection with the server (config.php)
2. Data is received from the application through HTTP POST requests.
3. `mysql_real_escape_string()` function is used for each attribute to escape special characters in a string for use in an SQL statement – a common function for preventing database attacks.
4. Checks if the submit button was pressed and if the emails confirm. Inside the if-statements a `mysql_query()` function is used to execute a query on the MySQL database, this query inserts the values straight into the User table in the database.
5. The users password is encrypted using the `md5()` function, which calculates the MD5 hash of the password using the RSA Data Security, Inc. MD5 Message-Digest Algorithm, and returns that hash [18].
6. Finally a session is created and stores the user id in `$_SESSION['id']`

On the final registration page (where merchants enter all there details) similar steps are taken as above. The `$_SESSION['id']` that was created in the previous PHP script is
carried forward (due to the session) and inserted into both tables so they can effectively be linked under the same user id. When the user fills in all the required fields and clicks the ‘Finish’ button, they then are automatically redirected to their dashboard. The data that was entered is then stored into the BusinessInfo and Services tables using `mysql_query()` INSERT statement.

Note: config.php is included in all scripts, as it connects the scripts to the database.

```php
<?php
include ('config.php');
$eemail = mysql_real_escape_string($_POST['email']);
$confirm_email = mysql_real_escape_string($_POST['confirm_email']);
$password = mysql_real_escape_string($_POST['password']);
$submit = $_POST['submit'];

if($submit){
    if($eemail===$confirm_email) {
        $insert = mysql_query("INSERT INTO User (email, password) VALUES ("$eemail", md5('$password'))");
        session_start();
        $_SESSION['id'] = mysql_insert_id();
        header('Location: /update_information.php?');
    } else {
        echo"PROBLEM!";
    }
}
?>
```

Figure 5.2 – Screenshot of regprocess.php

### 5.2.3 Database

Figure 5.3 shows the layout of the ‘User’, ‘Services’ and ‘BusinessInfo’ table contained in the database. As you can see the User table is linked via foreign keys to the other two tables. All three tables are used in the registration process.
5.3 Login Process

The login page was designed and implemented to give users a clear and simple process, while keeping with DayOut’s colour schemes. This meant minimizing human error with a big and bright colored log in button and a white form against a blue background.

By retrieving the login details from the user, these where compared against any existing users and password – implemented by a `mysql_query()` below.

```
$query = mysql_query("SELECT * FROM `User` WHERE email = ".$email." AND password = md5(".$password."));
```

If they matched (using the `mysql_num_rows()` function) a `session_start();` is implemented and the user will be redirected to their dashboard. If not, the user is notified that their username or password is invalid – meaning they either entered it their details wrong or they don’t exist on the database.

When a merchant has successfully registered or logged in they are automatically brought to their dashboard, where they are greeted with a personalized welcoming message. The dashboard contains two main pages for the merchant called ‘Add Items’ and ‘Library’ – these will now be discussed in detail. Like the other web pages, these were implemented using HTML, CSS and PHP. Again like the Home Screen, a fixed menu bar (Figure 5.4) was used for section headers and a logout link, which is at the top of the page, allowing users to easily navigate through the dashboard.
5.4 Add Items (Dashboard)
This page allows merchants to easily push their advertisements to the mobile application within seconds. As this page allows merchants to showcase their products / services to local residents and travel goers, it has to be considered the most important, therefore it was made so that the page was extremely user friendly and engaging for merchants to use.

5.4.1 Application
The ‘Add Items’ screen actually contains three separate pages as the button at the top (where merchants choose attractions, deals or events) is treated as segmented controlling. A segmented control is a horizontal control made of multiple segments, each segment functioning as a discrete button. The purpose is to allow users to interact with a number of different views in one page [18]. These three buttons (Figure 5.5) were implemented using .class selector styles in CSS.

![Segmented Controlling](image)

Figure 5.5 – Segmented Controlling

When a user selects one of the buttons (above), corresponding information will be asked for the merchant to fill in. These forms include all relevant information about the advertisement:

<table>
<thead>
<tr>
<th>Attractions</th>
<th>Deals</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attraction Name</td>
<td>Item Name</td>
<td>Event Name</td>
</tr>
<tr>
<td>Price</td>
<td>Picture</td>
<td>Event Picture</td>
</tr>
<tr>
<td>Information</td>
<td>Item Information</td>
<td>Venue</td>
</tr>
<tr>
<td>Address</td>
<td>Price</td>
<td>Event Information</td>
</tr>
<tr>
<td>Price</td>
<td>Expiry Date</td>
<td>Price</td>
</tr>
<tr>
<td>Valid Until</td>
<td>Contact Information</td>
<td>Event Date</td>
</tr>
</tbody>
</table>
For presentation purposes it’s only necessary to walk through one of the scenarios, as all three are implemented with the same strategy. How merchants add ‘Deals’ will be looked at in this case.

5.4.2 PHP (Server Side)

When the merchant fills out the information required on the ‘Deals’ page and clicks the submit button, the data is sent for processing to a PHP script called upload_deals.php., which does the following:

1. Makes a connection with the server (config.php)
2. Data is received from the application through HTTP POST requests and mysql_real_escape_string() is used for each attribute.
3. The script then performs a series of if-else statements to make sure all information entered is valid:
   a. Checks the image_type is valid – only gif, jpeg, jpg, pjpeg, x-png and png can be used.
   b. Checks that the image_size is smaller than 200KB.
   c. Checks to see if there was any errors, if not,
   d. Adds a timestamp to the image_name as well as a four digit random number before the image_type – making it close to impossible for image names to be the same.
   e. Checks if the image name already exists, if not,
   f. A mysql_query() function is used to execute a query on a MySQL database, this query inserts the values straight into the Deals table in the database.
4. If successful user is redirect to dashboard-item_success.php and if not the related error will be shown to the user.
5.4.3 Database

Figure 5.6 shows the layout of the ‘Deals’ table contained in the database. For merchants adding ‘Attractions’ or ‘Events’ the process / strategy used was the same, with the layout being very similar in the database.

![Figure 5.6 – Deals table](image)

5.5 Library (Dashboard)

This page was developed to allow merchants to view a table containing their current deals, events and attractions. Figure 5.7 shows the two alternative views this page can produce. The left is when a merchant has not added any adverts – showing a div container in the middle of the screen that includes a button, linking to the ‘Add Items’ page. However when merchants have active advertisements, a table will be produced showing the category (attractions, deals or events), name and price. Also on the right hand side is a small bin image where merchants can easily drop/delete any of their advertisements.

![Figure 5.7 – Two alternative views. Left: No current offers. Right: Current offers](image)

5.5.1 PHP (Server Side)

dashboard_library.php (screenshots above) is implemented in the following:

1. Makes a connection with the server (config.php)

2. When the merchant is logged in, all their attributes are retrieved from the Deals, Events and Attractions table using a mysql_query() function. e.g. 

   ```php
   $dataA = mysql_query("SELECT * FROM Attractions WHERE user_id = ".$_SESSION['id'].""");
   ```
3. The main body of the code is inside the if-else statement, which is used to check if there are any entities in the three tables. This is done using `mysql_num_rows()` function.
   a. If `mysql_num_rows()` equals 0 for all three tables, the screen on the left of Figure 5.7 is called – meaning there are no advertisements too show.
4. However, if there are entities in the ‘Deals’, ‘Events’ or ‘Attractions’ table then data is retrieved and represented in a table view.
5. The table view was developed using multiple while loops to retrieve all the merchants’ data. First the header of the table was written to display what each cell was used for. i.e. category, item name and price.
6. Three while loops in the body of the table view were implemented – one for each table in the database. Figure 5.8 shows the process that was used for Attractions table. The process was the same for the Deals and Event tables.

![Figure 5.8 – Retrieving data from Attractions table](image)

**Note:** When a user presses the ‘empty_trash-26.png’ image, a PHP script (Figure 5.9) is called to delete that particular deal, event or attraction – in turn instantly deleting it from the database.

```php
<?php
    include "config.php";
    mysql_query("DELETE FROM Attractions WHERE id=".mysql_real_escape_string($_GET['id']));
    header('Location: /dashboard-library.php');
?>
```

![Figure 5.9 – Deleting selected Attraction](image)

35
5.6 Extras

Throughout the web implementation the main CSS pages were optimized for Tablet devices using `@media screen and (max-width: 1024px) { … }` and `@media screen and (max-width: 768px) { … }`. These were included in the style sheets as many businessman and women own tablets today. With iOS7 and thousands of business apps, iPad transforms the way people work. Optimizing the web application allows merchants to register, login and add advertisements on the go. DayOut’s web application offers businesses a new way to think about advertising, in their offices and on the go.
Chapter 6: Implementation - Mobile

This chapter introduces the mobile side of DayOut. The mobile application allows users to discover the latest Deals, Events or Attractions within their city. The home screen lets the user become more engaged with the application by selecting the categories they are looking for and entering their budget. The application then returns a personalized list of local offers for you and your family or friends to enjoy. This chapter now discusses how the application was implemented, and some techniques that were used throughout.

Using PhoneGap enabled the mobile application to be developed using web technologies. With this, Ionic and AngularJS were used. Ionic utilizes AngularJS to provide the application structure, while Ionic itself focuses on the user interface. In other words, we see a match between the power of Angular and the beauty of Ionic UI [22]. AngularJS uses a MV* Pattern where the $scope object acts as a ViewModel, manipulated through a Controller function [23]. This allows the separation between the presentation and business logic while enhancing maintainability and productivity.

In order to display views on each tab, are number of steps are taken (Figure 6.1):

1. Once a user clicks on a tab, that view is displayed
2. By displaying the view, the view controller is called.
3. The view controller goes and calls the service.
4. The service retrieves the data
5. The view controller stores the data in the $scope item
6. As the view calls the view controller, the data is now available to the view

![Figure 6.1 – Angular JS Architecture](image-url)
6.1 Index View (Deals, Events, Attractions)

Using the prototypes made in Chapter 4, the design was first implemented. This was easily made possible by using mock data instead of retrieving the actual data needed. It allowed for simple and fast changes to be made without having to worry about the back-end of the project. The design was completely optimized using CSS and the first step that was taken was creating a div container for each advertisement – `#imageContainer` that had a width of 100% and height of 200px. Each container was filled with the advertisements picture, while the item name, business name and price were displayed at the bottom on the container. Again all optimized using CSS.

![Image Container and View Image Container](image.png)

Figure 6.2 – Index View. Left: Deals Tab. Middle: Events Tab. Right: Attractions Tab

The view for the Deals, Events and Attractions tab were all implemented with the same strategy (Figure 6.2). For presentation purposes it’s only necessary to walk through one of the scenarios – Attractions will be looked at in this case.

6.1.1 Services & Controllers

Once the design was implemented, the next phase was now swapping the mock data for the actual data that had been uploaded by the merchants. This was done in `services.js`, where a service is created corresponding to a particular controller – in this case `AttractionService` is created (Figure 6.3), which does the following:

1. Creates a variable called `attractions`.
2. An `ajax()` method is used to perform an AJAX (asynchronous HTTP) request to a URL called `http://dayout.uphero.com/dayout-app/php/attraction_summary.php`. 

![Tab Bar](image2.png)
a. This PHP script connects to the server and executes a `mysql_query()` function that selects all attractions and returns its name, price, image and business name from the ‘Attractions’ and ‘BusinessInfo’ tables.

b. Using `mysql_fetch_array()`, all the entities get stored in an array called `$result_formatted` – were `echo json_encode($result_formatted)`; is used to return the JSON representation of a value.

3. An error function is called if the request has failed. This function simply makes a console log informing the developer that there was a problem connecting.

4. However if the request is successful, a success function is called returning all the data from the server into the `attractions` variable that was created at the start.

5. Finally `return attractions;` is used to return all the data to the service.

```php
/*
 * ATTRACTIONS SUMMARY
 */
.factory('AttractionService', function() {
  // Ajax call thats returns a JSON array full of data relating to 'ATTRACTIONS'
  var attractions;
  all: function() {
    jQuery.ajax(backEndURL = "attractions_summary.php", {
      async: false,
      cache: false,
      error: function() { console.log("oh no"); },
      success: function(data) {
        attractions = data;
      }
    });
    return attractions;
  }
});
```

Figure 6.3 – AttractionService

Now that the data has been received in `AttractionService`, the controller needs to call it. This is done in `controller.js` where a controller is created corresponding to a particular view. In this case it’s the `AttractionIndexCtrl` is created. The first thing the controller does is call the corresponding service (`AttractionService`) and store it in the `$scope` item.

```
$scope.attractions = AttractionService.all();
```

By storing the data in the `$scope` item, the data is now available to the view. Within this controller a variable called `loading` was developed to show a loading image every time the tab was clicked. A function was also implemented to reload the data from the service every time a user pulled down on the screen to refresh the page – see Figure 6.4.
Now that the data is available to call, a list view was developed to display all attraction information with a link to its detailed screen. The strategy that was used to retrieve and display data for the Attractions tab was the same to that of Deals and Events, which can be seen on Figure 6.2. This chapter will go discuss how the ‘Detailed View’ was developed.

6.2 Detailed View

When a user clicks on an advertisement in the Deals, Events or Attractions tab, the detailed view is displayed. While the underlying design is the same for each tab, the content that’s in it isn’t. Figure 6.5 shows what the detailed view looks like.
Like the ‘Index View’, the design was completely optimised using CSS. Each one had a very similar look but displayed slightly different information, as some attributes were different. This can be seen in **Table 6.1**.

<table>
<thead>
<tr>
<th>Deals</th>
<th>Events</th>
<th>Attractions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Image</td>
<td>Image</td>
</tr>
<tr>
<td>Item Name</td>
<td>Item Name</td>
<td>Image</td>
</tr>
<tr>
<td>Business Name</td>
<td>Business Name</td>
<td>Business Name</td>
</tr>
<tr>
<td>Price</td>
<td>Price</td>
<td>Price</td>
</tr>
<tr>
<td>Deal Expires</td>
<td>Date of Event</td>
<td>Valid Until</td>
</tr>
<tr>
<td>About This Deal</td>
<td>Venue</td>
<td>Address</td>
</tr>
<tr>
<td>Contact Number</td>
<td>About This Event</td>
<td>About This Attraction</td>
</tr>
<tr>
<td>Website</td>
<td>Contact Number</td>
<td>Contact Number</td>
</tr>
<tr>
<td></td>
<td>Book Online</td>
<td>Website</td>
</tr>
</tbody>
</table>

**Table 6.1 – Information that’s displayed on the Detailed View**

Again the detailed view for advertisements in the Deals, Events and Attractions tab were all implemented with the same strategy – Attractions will be looked at, as it’s only necessary to walk through one of the scenarios.

### 6.2.1 Services & Controllers

The services that were created for the detailed view were similar to that of the index view. For Attractions the `AttractionDetailService` calls an `ajax()` method that performs an AJAX request to a URL called “http://dayout.uphero.com/dayout-app/php/attraction_detail.php?attraction_id=”+nId. This PHP script returns all the information in the Attractions table to that particular ID. Meaning that when a user clicks on an advertisement, a PHP script will process returning only the information that’s linked to that ID and not all Attractions. Like all, this service returns a JSON array.

The controller then calls the service (`AttractionDetailService`) and stores the clicked items data in the `$scope` item below:

```javascript
$scope.attractions = AttractionDetailService.get($stateParams.id);
```
Once the controller has been called and assigned to the tab view, the data can now be used in the HTML document. Figure 6.6 shows how the front-end of the detailed view for the Attractions tab was programmed. As the data is retrieved without affecting any of the user interface, the attraction variables can easily be called within HTML tags that where designed to display the data. This process was kept efficient as all the data that was retrieved from the service was called in this document and displayed on the mobile application.

**Note:** the strategy for displaying the detailed screen for an Attraction is the same for the Deals and Events tabs.

![Figure 6.6 – Screenshot of attraction-detail.html](image)

### 6.3 Budget Profile

This is the home screen of the mobile application. This view interacts with users by asking them to enter their budget and to select what they’re looking for. Then the application returns data based on their requirements – see Figure 6.7.

![Figure 6.7 – Budget Finder Process. Process goes left to right](image)
Figure 6.7 shows the process of how users navigate through the budget finder tab. In order to implement this successfully a number of steps were taken:

- The HTML and CSS was written first to optimize the look of the page and to make Deals, Events and Attractions checkboxes (Boolean values).
- The default was set so that all Boolean values were false and budget equaled 0 – this was done in the controller corresponding to the view under $scope.filter:

```javascript
$scope.filter = {events: false, deals: false, attractions: false, budget: 0};
```

- Once a user has selected their checkboxes and entered their budget, a the `navigate()` function is called in the controller:

```javascript
$scope.navigate = function(){ $location.path( "/tab/home/budget/" + $scope.filter.deals + "/" + $scope.filter.events + "/" + $scope.filter.attractions + "/" + $scope.filter.budget );};
```

- This function then sends the request to `home-budget.html` where it is linked to a service called `BudgetService`. This service calls an `ajax()` method that performs an AJAX request to a URL called “http://dayout.uphero.com/dayout-app/php/budgetprocess.php”.
- This PHP script firstly creates an array called `$result_d`. It then checks if the ‘Deals’ checkbox is true. If so, it returns all the data from the Deals table where the deal item price is less than or equal to the budget entered and stores it in the array. This process is then repeated for the Events and Attractions checkbox.
- The corresponding controller called `BudgetIndexCtrl` then calls the service and makes the data available to the view.

Once that’s done, an Index View (see 6.1) is displayed to the user. But this time its personalized directly at them – it shows what the users are looking for, within their budget. The final thing that needed to be done was distinguishing what type the advertisement was because this Index View, unlike others, displays advertisements that are a combination of Deals, Events and Attractions. The user may select one, two or all checkbox’s, therefore displaying more than one type.
A function was made (Figure 6.8) to check what type the advertisement was and if clicked, the user then would be redirected to the detailed view corresponding to the type of the advertisement, while remaining on the home tab.

```javascript
// this function is done to look for the TYPE of offer
$scope.navigate = function(arg) {
  if (arg.item_type == "DEAL") {
    $location.path("/tab/home/budget/deal/" + arg.id);
  }
  if (arg.item_type == "EVENT") {
    $location.path("/tab/home/budget/event/" + arg.id);
  }
  if (arg.item_type == "ATTRACTION") {
    $location.path("/tab/home/budget/attraction/" + arg.id);
  }
}
```

Figure 6.8 – Detailed View on Home tab

This was implemented, as it was an efficient way to go from the Budget Index View to a Detailed View (see 6.2). If the advertisement were a ‘Deal’, the DealDetailCtrl would be called, and if the advertisement were an ‘Event’, the EventDetailCtrl would be called etc. Doing this enabled the development to be structured and efficient, as a new controller didn’t have to be created for each tab.

6.4 Notes

While the application was design for users to discover the latest Deals, Events or Attractions within their city, under time limits and demonstration purposes, the mobile application was only shown how it would work in Dublin. Both applications however were designed so in the near future more cities could easily be added, without having major changes to the code written.

Having discussed how the web and mobile application was implemented, the report now continues onto Chapter 7, where this chapter focuses on a business model that has been constructed for the future of DayOut.
Chapter 7: Business Model

This chapter showcases DayOut’s business model using a Business Model Canvas (see poster on presentation slides). This is used as a strategic management template for developing new business models that is a visual chart with elements describing a firm’s value proposition, infrastructure, customers and finances [19]. It assists firms in aligning their activities by illustrating potential trade-offs. With this business model design template, a firm like DayOut can easily describe its business model.

7.1 Key Partners

In order for DayOut to gain great exposure, partnership with Guinness Storehouse and Dublin Zoo is essential. Last year these new sites where the top two attractions to go and see in Ireland, together attracting over 2 million people. Having Guinness Storehouse and Dublin Zoo showcase and advertise DayOut for merchants and consumers would be vital for rapid growth. Partnerships with popular activity-based firms like Jump Zone Indoor Trampoline Park in Sandyford and Santry or popular events like Oxygen or Love Life Festival would showcase the great power of using DayOut. Support and resources are needed to make this project successful.

7.2 Key Resources

The key resources for DayOut would be capital investors – money to improve, test and maintain high quality of both applications. Enterprise Ireland supplies great funding for start-up firms called the Competitive Start Fund. The purpose of this is to accelerate the growth of start-up companies that have the capability to succeed in global markets [21]. With said funding, software engineers and business graduates are needed to help develop new features, and promote the applications. Bringing people in also offers fresh ideas.

7.3 Key Activities

A big part of business’ today is keeping the communication between the business and programming ends active. Being a Business & Computer Science student it is great to know the fundamentals in both, because without good communication with both parts of the business, problems can arise. Managing both application sales is another activity that can be easily done by sales figures received from web subscriptions and mobile downloads from related app stores. For each application, advertising will be done
separately. For the web application (potential) partners Guinness and Dublin Zoo will advertise their use of DayOut, hopefully attracting new businesses to join (network effort). For mobile users, advertisements will be targeted all across the city as well the Airport – attracting new arrivals to Dublin. Finally, once the applications are live, customer feedback is vital to make improvements for the future.

7.4 Value Proposition
DayOut’s web application offers merchants a real-time marketing tool that allows them to add advertisements within seconds. This may not be a unique experience for merchants as many advertising methods currently exist, but for consumers using the mobile application it will be. Currently there are no existing applications that let friends or family enjoy a day out by selecting what they are looking for and entering a budget.

<table>
<thead>
<tr>
<th>Web Features</th>
<th>Mobile Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login / Register</td>
<td>Discover more, within your budget</td>
</tr>
<tr>
<td>Add advertisements in real-time</td>
<td>See the latest deals, events and attractions</td>
</tr>
<tr>
<td>View library containing active ads</td>
<td>Contact merchants in seconds</td>
</tr>
<tr>
<td></td>
<td>View merchants website / booking page</td>
</tr>
</tbody>
</table>

Table 7.1: Features of DayOut

7.5 Cost Structure
DayOut’s cost structure contains three parts:

1. Staff wages: paying programmers to improve the development of the applications, but also for ongoing wages for app support, maintenance and updates. A small potion will be for the salaries of the business graduates.
2. Promotion: includes advertising both applications.
3. Overhead costs and equipment.

7.6 Channels
The main channel that DayOut will use is on the web – as the web application is targeted for all merchants. Other channels that will be used are mobile application stores across all smartphones/handheld devices:

1. The App Store (Apple iOS Devices)
2. Google Play (Android)
3. Ovi Store (Nokia)
7.7 Customer Segments
The potential market for the web application is any small, medium or large business with access to Internet connection that is looking for a cheaper alternative way to advertise their product or service. DayOut will also be targeting any smartphone user that is looking for a new way to view the best deals, events and attractions near them – aimed at both the local residents and active travel goers.

7.8 Customer Relationships
The relationship with merchants is of big importance. Feedback will be taken on board to improve merchants experience using DayOut. However, the relationship with the customer won’t really be a big part of the business as the application is self-service; they view what’s near them and within their budget. Users control what they want to see but if they have any problems with the mobile application, DayOut will offer in-app online support. However, users will act as co-creator of our service. DayOut need them for the service to be successful, otherwise why would merchants want to advertise it on DayOut’s platform.

7.9 Revenue Streams
There are a few options available for the DayOut pricing strategy. The web application will have three pricing options, (1) a yearly subscription fee of €149 per month (€1,788 per year) allowing users to have 1 active advertisement, (2) a yearly subscription fee of €399 per month (€4,788 per year) allowing users to have 5 active advertisements and (3) a 30 day subscription fee of €499 allowing users to have 5 active advertisements. This pricing strategy is a lot cheaper than what existing ‘deal’ companies offer. Groupon takes 50% all of sales, resulting in a lot of companies actually making a loss. A commission free service allows merchants to advertise their products/services in DayOut for as low as €149 a month.

7.10 Summary
This chapter has shown a brief Business Model that will hopefully be implemented in the near future. It is extremely important to set a Business Model / Plan early in a start-up as it gives the founders some guidance and targets to achieve. Chapter 8 now evaluates the project as a whole, discussing the success and future works of DayOut.
Chapter 8: Evaluation

The purpose of this chapter is to review the project as a whole. First, the successes that were achieved are discussed, followed by a discussion of the opportunities for future work related to the project.

8.1 Project Success

The aim of this project from the outset was to create two applications. One, a web application where merchants register to add advertisements under the headings of Deals, Events and Attractions and also view a list of their advertisements where they can manage and delete them. The aim was to have the advertisements pushed to a mobile application, which was the second application aimed towards the consumers. These advertisements were to be presented in a list, viewable on any mobile application to any user looking for the latest deals, events and attractions around them, within their budget. To this extent, the project was a success. Additionally, to the author’s knowledge, an application such as this has not been implemented before; while there are similar applications out there out the moment, nothing quite like this was been developed, so this is a great success.

8.1.1 Design Driven

While the functionality of the applications worked exactly to plan, the aim was to keep both applications as user friendly and engaging as possible. By watching and learning many tutorials over the year, optimizing the look and feel for both applications was really a big focus for this project. An invaluable amount of knowledge was absorbed throughout the development process.

8.1.2 Available On All Platforms

Using the PhoneGap development framework it was great to be able to deploy the mobile application to all platforms. At the beginning of the project, the aim was to target all consumers, not just people using Android or iOS smartphones. Using PhoneGap was a bit of a gamble as it is only new in mobile development. In a sense, the choice of PhoneGap was a gamble, albeit a calculated one, that paid off, as the mobile application that was developed can now be deployed to all smartphones, without sacrificing any functionality or design requirements that were set from the beginning.
8.1.3 Tablet Optimization

Over the years many businesses have used tablet devices to try and increase productivity and profitability for their business [23]. This was one of the main motivations for optimizing the web application so merchants could easily add / delete advertisements on the go. While the applications aim was to be developed for only desktop use, the end result saw it developed for both desktops and tablet devices. This has great benefits for merchants, as they don’t have to be in the office or at home to go onto the web application. Advertising any time from anywhere can be quite a powerful tool for businesses.

8.2 Future Work

As is the case with any software project, many difficulties were encountered along the way, with plenty of room for future work / updating. The following are some of such areas:

8.2.1 Google Maps API

The next step in the development of DayOut is to add Google Maps to the mobile application. At this current time, when the merchant inputs a venue to an ‘Event’ or an address for an ‘Attraction’ the data is only displayed in text form on the mobile application. The aim in coming months is to use Google’s Geocoding API to turn the string address into longitude and latitude coordinates and pin the coordinates on a map. This could be a lot more beneficial for the users as they can easily view where the location is on the map and use directions to easily find the venue.

8.2.2 Update Server

Another important improvement that’s needed is updating the current version of DayOut’s server. At the moment the project is using MySQL version 5.1.1, which doesn’t allow the use for SHA-2 security (5.5.1 is needed). Using SHA-2 encryption for password protection is a lot more secure then md5 hashing. This update isn’t just better for security reasons but also for speed and reliability. While navigating through the web application, sometimes pop-up advertisements appear, something merchants don’t want to see. By updating to a paid (cloud) hosting service this will improve the experience for merchants using DayOut in the future.
8.2.3 Mobile App Login

Developing a mobile login screen is something that will be done down the line. By gathering information from potential consumers using the mobile application could give DayOut the power to sell this information to merchants. Letting merchants not just target people in a particular location but also between a certain age, or gender is something that can make DayOut really unique. While this is done on social networking sites like Facebook and Twitter, it was not been implemented on any travel companion applications or flash sale sites as of yet. Therefore creating a unique selling point for merchants using DayOut.

Finally, real world deployment isn’t far away for DayOut. Adding in geo-fencing perimeters around locations (countries or cities) can easily be done using Google Map APIs. Doing this could see DayOut being used in any city across the world, and as the applications are designed for this to be implemented in the future – it wouldn’t be extremely hard to accomplish.
Chapter 9: Conclusion

This report has outlined the successful implementation of an application capable of gathering merchant details, storing them on a server, retrieving them and displaying them on a mobile application available on all smartphone devices. To the author’s knowledge, this is the first application that takes in advertisements from merchants and displays them on a mobile application where users can select what they are looking for and enter in their budget. Bringing up a list of local deals, events and attractions for friends and family to enjoy. This is not to say that the proposed solution is complete; it is simply a successful prototype aimed at giving merchants an alternative way to advertise and showcase their products and services, while at the same time giving consumers a chance to discover more within their city. An application capable of this has huge market potential as it could potentially help users across the globe when traveling or at home, looking for something great to do – while benefiting the merchant, therefore is an appealing candidate for future work.

In undertaking this project, a more rounded knowledge of not just computer science but also how the development of web and mobile applications are so important for businesses and consumers today. Existing applications in different areas exist already; a common problem has shown that businesses aren’t making the profits they had hoped for as ‘deal companies’ are charging them too much. While consumers on the other hand are fed up of advertisements they’re not interested in – this project has demonstrated that if approached in the right frame of mind (benefiting both merchants and consumers), it can be quite the opposite.
References


