

# **Tonight - An iOS app for event aggregation**

Final Report for CS39440 Major Project

*Author:* Ryan Clarke (ryc@aber.ac.uk)

*Supervisor:* Dr. Richard Jensen (rkj@aber.ac.uk)

8th April 2014

Version: 1.0 (Draft)

This report was submitted as partial fulfilment of a BSc degree in  
Computer Science (G401)

Department of Computer Science  
Aberystwyth University  
Aberystwyth  
Ceredigion  
SY23 3DB  
Wales, UK

## **Declaration of originality**

In signing below, I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for plagiarism and other unfair practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook of the Department of Computer Science.
- I understand and agree to abide by the University's regulations governing these issues.

Signature .....

Date .....

## **Consent to share this work**

In signing below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Signature .....

Date .....

## **Acknowledgements**

I would like to thank my dissertation tutor Richard Jensen for helping throughout the project.

I would also like to thank my parents for the support they have given me throughout my degree.

## **Abstract**

This report describes a project for helping users find events around them. When visiting a new area trying to find out what's going on can be difficult even for the most adept regular. Tonight tries to solve that problem by providing a mobile application that allows you to discover events that are going on around you. It does this by presenting the user with events that may be of interest to them via a 'My Feed' section. A user also has the ability to explore events by selecting a city/area then drilling down to a particular venue or category. The users is then able to view all of the details on a particular event, with the option to follow the event. By following an event the system can learn about the user and present them with suggestions of events, based on what similar users are also following. To do this I developed a Ruby on Rails server application that provided a REST based API to access the information that a user needed, as part of this I also developed a data mining module to be able to pull in information from various data sources. Using this in conjunction with an iOS application to present the data and add in functionality to the user on a mobile device.

# CONTENTS

<b>1</b>	<b>Background &amp; Objectives</b>	<b>1</b>
1.1	Background . . . . .	1
1.1.1	The . . . . .	2
1.1.2	Current Solutions . . . . .	2
1.2	Analysis . . . . .	4
1.2.1	Primary objectives . . . . .	4
1.2.2	Secondary objectives . . . . .	5
1.2.3	Hardware requirements . . . . .	5
1.3	iOS and Ruby on Rails . . . . .	6
1.3.1	iOS Development . . . . .	6
1.3.2	Server Development . . . . .	7
1.4	Process . . . . .	7
1.4.1	Server application . . . . .	7
1.4.2	iOS Application . . . . .	7
<b>2</b>	<b>Design</b>	<b>9</b>
2.1	Overall Architecture . . . . .	10
2.2	Some detailed design . . . . .	10
2.2.1	Even more detail . . . . .	10
2.3	User Interface . . . . .	10
2.4	Other relevant sections . . . . .	10
<b>3</b>	<b>Implementation</b>	<b>11</b>
<b>4</b>	<b>Testing</b>	<b>12</b>
4.1	Overall Approach to Testing . . . . .	12
4.2	Automated Testing . . . . .	12
4.2.1	Unit Tests . . . . .	12
4.2.2	User Interface Testing . . . . .	12
4.2.3	Stress Testing . . . . .	12
4.2.4	Other types of testing . . . . .	12
4.3	Integration Testing . . . . .	12
4.4	User Testing . . . . .	12
<b>5</b>	<b>Evaluation</b>	<b>13</b>
	<b>Appendices</b>	<b>14</b>
<b>A</b>	<b>Third-Party Code and Libraries</b>	<b>15</b>
<b>B</b>	<b>Code samples</b>	<b>16</b>
2.1	Random Number Generator . . . . .	16
	<b>Annotated Bibliography</b>	<b>19</b>

## LIST OF FIGURES

1.1	Line Up home screen . . . . .	2
1.2	Spotnight home screen . . . . .	3
1.3	Songkick home screen . . . . .	4
1.4	Image of Google Analytics for popular events site . . . . .	6

## LIST OF TABLES

# Chapter 1

## Background & Objectives

### 1.1 Background

The UK music events and festival industry was estimated at 3.5 billion in 2010 and expected to raise to 4.2 billion in 2015. With roughly 530,000 full time equivalent jobs employed by 25,000 employers. [3] The industry itself is very diverse with a huge range of events going on every night, from intimate acoustic sets to all night raves, many people find themselves going to the same venue over and over again not aware of other events happening around them. Currently the promotion of events is limited to the users of certain social media sites, or purpose built websites for promotion of events. The tools that a promoter currently has at their disposal is limited and sporadic at the best, for an example The Rainbow Venues [8] use a mixture of their own site, social media (Facebook/Twitter) and the ticket masters. This brings to the core problem for a potential customer to discover the venue, they must be aware of the venue first to be able to find the information that they want. Even with use of social media the reach of a promoted event will only reach those that know of the venue or by those that 'share' the promotion to their friends.

With the cost of living going down people are looking further afield for evening entertainment, by not being from around the area they are at an immediate disadvantage. With the increase of popularity of smart phones many people will pick up their smart phones, and immediately search for an application to help them find events on the go. The current selection of applications are very limited to the style of events and the area they are based, which means a new application for each time they go away this is not ideal and will quickly fill up a smart phone.

At this time there is a wide selection of similar applications available to download from the Apple AppStore. Each of these applications differ slightly be it the way the information is presented or the key functionality they provide. Many of the applications are only suited towards a particular city or specific headlining artists, by doing this they are somewhat limiting the scope of their audience. Many of the solutions out there also currently only utilise data that's being input by employees, this means that only a representative of all the events happening are selected and presented by the application.



### 1.1.1 The

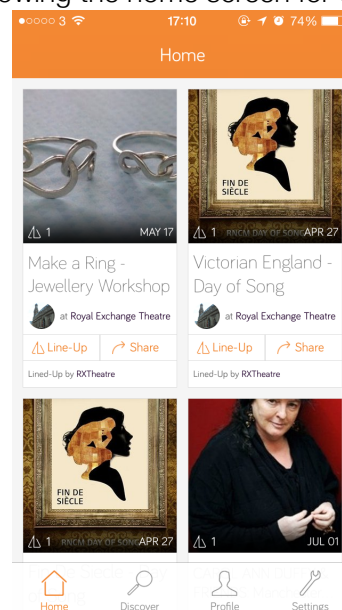
### 1.1.2 Current Solutions

Here are some of the current applications that are available to download from the Apple AppStore. All of the applications are free downloads, along with free registration to use the features. All of the applications as standard provide a feed of events (in a relevant order), and the ability to filter them in various ways.

#### 1.1.2.1 Line Up

Line Up [5] shows a wide variety of events in the Manchester area, it gives you the ability to add events to their 'Line Up' which is essentially a list of events that they are planning to go to or going to. It also gives you the opportunity to share the event via popular social network sites, increasing their own reach and allowing the users friends to see what they are attending/interested in. To discover events you are able to browse events by type of place, People, and all events. The application also allows you to follow other users that use the application, allowing a user to see what other users are attending. When viewing an individual event you can see a title, description, dates of the event, and the location.

Figure 1.1: Image showing the home screen for the Line Up application

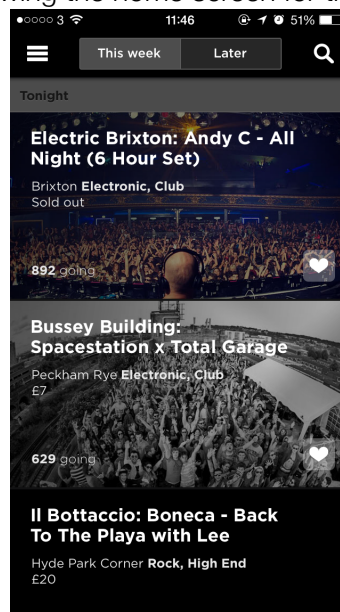


#### 1.1.2.2 Spotlight

Spotlight [7] shows selected events happening in the London area, it gives you the ability to purchase tickets for the events available through a 3rd party service. You can view the events that are happening either this week or later on, however you are able to apply filters for specific areas, style of music, and genres. The application allows you to like events, however

this does not seem to have any particular effect to the ordering of the events or any other aesthetic/functional item of the application. When viewing an individual event, you are able to view the location, description, price, images, people attending, and venue contact details. You are also able to share particular events through social networking sites; Facebook and Twitter.

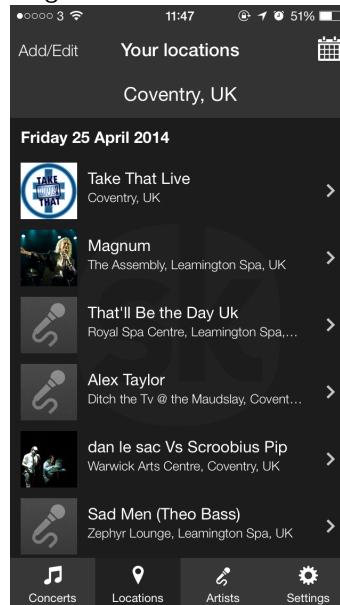
Figure 1.2: Image showing the home screen for the Spotlight application



### 1.1.2.3 Songkick

Songkick [6] offers a selection of artist specific events, Songkick allows you to see events happening in your area, and by artist. From here you are presented with the details of the event including the line up, location and an option to purchase tickets via a 3rd party service. Songkick also allows you to track an event and mark an event as attending, to use these functions you are required to sign up however casual use of the application does not require signing up. The application also allows you to follow artists and will suggest events that are happening near you within the artists you are following.

Figure 1.3: Image showing the home screen for the Songkick application



## 1.2 Analysis

With the main problem being users not knowing an area there is a prevalent set of key features that is required by the application for it to help solve the issue at hand. The most prominent being be able to grab the users current location and to be able to filter out the events that are happening near to their current location. However many people may not want to travel to the area only to find out that nothing that interests them is happening, so ideally the events should be categorised by area and then a user should be able to browse the areas they are visiting. Bearing this in mind, as the user is visiting a potentially new area they may require directions or some sort of map indicating the location of the event.

Another issue is that many of the previous systems require human interaction to find out about events and manually input these into to the system. Whilst this allows for a human based data validation it can be slow and time consuming to conduct, and result in only key events being selected and not provide a large enough range for their users. By use of data mining techniques we are able to pull in a must larger variety of events and therefore cater for a wider audience, thus allowing the user to actually discover new events that they may not have thought about. This relates to the first issue, by pulling in the details automatically from various API's we are able to half the work required by us and the promotions companies. Making the use of this service much more attractive to both parties, ultimately widening the stakeholders and potential audience.

### 1.2.1 Primary objectives

There a number of primary objectives for this task, these are listed below

**Personalised feed**

The application must produce a feed that's personalised to the currently logged in user

**Registration/Authentication**

The system must provide the ability to create a new user, and to login a previously registered user. They should also stay logged in until they have logged out of the account.

**Ability to follow events**

The application should provide functionality to follow an event to help facilitate the personalised feed.

**Categorise the events**

The application should allow the events to be presented in multiple ways including categories and areas.

**Show individual events**

The application must provide all of the relevant details of each event

**Connect to various API's**

The server must be able to pull in information from a number of different API's and

**Application - Server interoperability**

The application and server need to be able to pass information to each using some form of interoperability.

### 1.2.2 Secondary objectives

These objectives are not critical to the running of the application but provides extra functionality to the service.

**Location awareness**

The application should be of the location of the user, and only use the nearest events.

**Notifications**

The user should receive notifications then events have been updated based off the data mining application.

**User profile**

The application should show a users profile, showing the events they are following and their personal details.

**Link to purchase the events**

The application should provide some sort of back link to enable a user to purchase tickets through a 3rd party service.

### 1.2.3 Hardware requirements

**Application must run on > iPhone 4**

The iOS application must run on all iPhones that run the latest version of the iOS firmware.

### Server should run on cloud computing provider

The server element should run on a cloud computer to allow for high demand with ease.

## 1.3 iOS and Ruby on Rails

### 1.3.1 iOS Development

The application itself will be based on the iOS platform for Apple iPhones, being programmed using Apples Objective C language. This decision was made based on some Google Analytics information, which was provided by a popular events company in Birmingham. Figure 1.4 shows us the statistics that's been produced by Google Analytics showing that over 70% of their traffic was coming from Apple iPhones. The programmer also had readily available access to an iPhone to assist with the development process underlined.

Figure 1.4: This image shows within a month their main traffic is through Apple iPhones, with secondary traffic of Android being a much smaller percentage.

<input type="checkbox"/>	Mobile Device Info ?	Acquisition
		Sessions ? ↓
		<b>16,106</b> % of Total: 59.73% (26,966)
<input type="checkbox"/>	1. Apple iPhone 	<b>11,343 (70.43%)</b>
<input type="checkbox"/>	2. Apple iPad 	<b>1,576 (9.79%)</b>
<input type="checkbox"/>	3. Samsung GT-I9505 Galaxy S IV 	<b>608 (3.77%)</b>

To develop iOS applications you are required to use Apples own IDE XCode 5 packaged into this is the iOS simulator, this allows me to test applications developed locally on the machine without needing to make a payment to the developers network. The simulator allows for all parts of the application to be tested, however I will need to manually set up locations to test the location awareness of the application. I will also be utilising a package manager called CocoaPods [1] which allows me to pull in various libraries with ease and

### 1.3.2 Server Development

The project will also require a server side element to be able to mine the data and provide the data to the application, this will be written in Ruby using the Ruby on Rails framework. Using the framework gives access to lots of functionality not built into the core of Ruby and provides a production ready environment. Due to the explosive nature of applications, it's been decided to use cloud computing to be able to effectively manage the demand of resources used by the application. For this it was decided the best cloud platform is Heroku [4] this was mainly due to the ease of deployment, which involved a git push to their remote server, and they provided a free tier during development.

## 1.4 Process

Mostly the methodology followed was a cut down version of agile, optimised for a single developer. This decision was made to allow the programmer to add in extra functionality as seen fit throughout the project, and to allow the design to flex with these requirements. To develop the bulk of the application feature driven development (FDD) was used as there were 3 main logical aspects to the project those being; Interoperability, iOS Application, and Data mining. Due to the nature of each of these sub sections, there will a slightly different sub-methodology for each. Each feature relies on one already being present bar the data mining feature, this is key to the order of features. The first feature developed was the data mining, which received information and input it into a database, the interoperability feature of the server element (REST API) and then the iOS application itself to use the server element of the project. Agile was also chosen as whilst we know the scope of the project currently, looking to the future it may be required to add in extra functionality based off the reception received. Inside of each feature was a set of iterations, these were primarily individual requirements for each feature.

### 1.4.1 Server application

The development of the server application will be following the test driven development (TDD) methodology, utilising the red, green, re-factor ideology. This was chosen as it will allow for flexibility in the overall design of the application, and will also ensure tests are being written for the functionality of the software. By following TDD it will also ensure that the code is the most efficient, as no more code is written except for what's required to pass the various tests.

### 1.4.2 iOS Application

The development of the iOS application was a lot more conventional, a feature was programmed with relevant unit testing implemented to ensure the functionality produced the correct output. A form of TDD was used as sometimes a test failed so it was required to go back and fix the code to ensure the test passed. After each feature was added, device testing was undertaken ensuring it all ties together well and works on the device.

## Chapter 2

# Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

## **2.1 Overall Architecture**

## **2.2 Some detailed design**

### **2.2.1 Even more detail**

## **2.3 User Interface**

## **2.4 Other relevant sections**



## Chapter 3

# Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex; perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

You can conclude this section by reviewing the end of the implementation stage against the planned requirements.

## Chapter 4

# Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Have you tested your system on “real users”? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

### 4.1 Overall Approach to Testing

### 4.2 Automated Testing

#### 4.2.1 Unit Tests

#### 4.2.2 User Interface Testing

#### 4.2.3 Stress Testing

#### 4.2.4 Other types of testing

### 4.3 Integration Testing

### 4.4 User Testing

## Chapter 5

# Evaluation

Examiners expect to find in your dissertation a section addressing such questions as:

- Were the requirements correctly identified?
- Were the design decisions correct?
- Could a more suitable set of tools have been chosen?
- How well did the software meet the needs of those who were expecting to use it?
- How well were any other project aims achieved?
- If you were starting again, what would you do differently?

Such material is regarded as an important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things and room for improvement with any project. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

Review the discussion on the Evaluation section from the lectures. A recording is available on Blackboard.

# Appendices

## Appendix A

# Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

Apache POI library □ The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the client's existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [?]. The library is released using the Apache License [?]. This library was used without modification.

## Appendix B

# Code samples

## 2.1 Random Number Generator

The Bayes Durham Shuffle ensures that the psuedo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs [?].

```
#define IM1 2147483563
#define IM2 2147483399
#define AM (1.0/IM1)
#define IMM1 (IM1-1)
#define IA1 40014
#define IA2 40692
#define IQ1 53668
#define IQ2 52774
#define IR1 12211
#define IR2 3791
#define NTAB 32
#define NDIV (1+IMM1/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0 - EPS)

double ran2(long *idum)
{
    /*-----*/
    /* Minimum Standard Random Number Generator */
    /* Taken from Numerical recipies in C */
    /* Based on Park and Miller with Bays Durham Shuffle */
    /* Coupled Schrage methods for extra periodicity */
    /* Always call with negative number to initialise */
    /*-----*/

    int j;
    long k;
    static long idum2=123456789;
```

```
static long iy=0;
static long iv[NTAB];
double temp;

if (*idum <=0)
{
    if (-(*idum) < 1)
    {
        *idum = 1;
    }else
    {
        *idum = -(*idum);
    }
    idum2=(*idum);
    for (j=NTAB+7;j>=0;j--)
    {
        k = (*idum)/IQ1;
        *idum = IA1 *(*idum-k*IQ1) - IR1*k;
        if (*idum < 0)
        {
            *idum += IM1;
        }
        if (j < NTAB)
        {
            iv[j] = *idum;
        }
    }
    iy = iv[0];
}
k = (*idum)/IQ1;
*idum = IA1*(*idum-k*IQ1) - IR1*k;
if (*idum < 0)
{
    *idum += IM1;
}
k = (idum2)/IQ2;
idum2 = IA2*(idum2-k*IQ2) - IR2*k;
if (idum2 < 0)
{
    idum2 += IM2;
}
j = iy/NDIV;
iy=iv[j] - idum2;
iv[j] = *idum;
if (iy < 1)
{
    iy += IMM1;
}
```

```
    if ((temp=AM*iy) > RNMx)
    {
        return RNMx;
    }else
    {
        return temp;
    }
}
```



# Annotated Bibliography

- [1] Cocoa Pods, "CocoaPods.org - The Dependency Manager for Objective C. ," <http://cocoapods.org/>, 2014.
- [2] GitHub, "GitHub - Hosted GIT Server," <http://www.github.com/>, 2014.
- [3] HBAA, "Opportunities for Growth in the UK Events Industry," <http://www.hbaa.org.uk/sites/default/files/Opportunities%20for%20Growth%20in%20the%20UK%20Events%20Industry.pdf>, 2014.
- [4] Heroku, "Heroku - Cloud Application Platform," <https://www.heroku.com/>, 2014.  
The cloud server being used to deploy to a production ready state
- [5] Line Up, "Line Up - Events in Manchester," <http://lineupnow.com/>, 2014.
- [6] Songkick, "SongKick - Concerts, tour dates, and festivals for your favorite artists," <http://www.songkick.com/>, 2014.
- [7] Spotnight, "Spotnight -Nights out in London," <http://spotnightapp.com/>, 2014.
- [8] The Rainbow Venues, "The Rainbow Venues," <http://www.therainbowvenues.co.uk/>, 2014.
- [9] Trello, "Trello - Project Management," <http://www.trello.com/>, 2014.