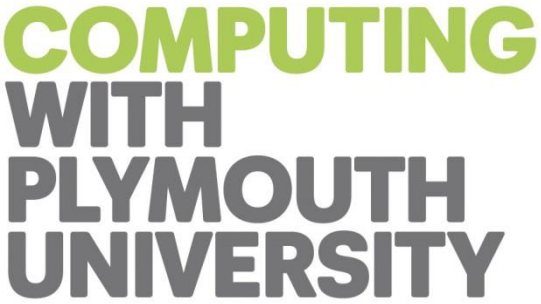


School of Computing and Mathematics

PRCO303

Final Stage Computing Project

BSc (Hons) Computing

Kevin Borrill

Festival: A system that allows Festivals mapping, line ups and news to be accessed on a mobile device.

2014/2015

Acknowledgements

I would like to thank the following people who played a key role in the process that was the Festival project:

My Supervisor: Martin Beck for his excellent advice.

Proof readers: Alison Gunn, Sarah Gunn and Andrew Borrill.

Course mates for general support during the project write up.

Abstract

This report explains the process that was undertaken for the Festival software development project. The report explores background research into technologies and similar applications. Project aims and objectives are set out. The method of approach is also considered by approaching a thorough research into development methodologies and considering timelines, risks and organisation.

The design of the developed project is set out through the explanation of considerations on technologies, high level architectural designs and user interface designs.

An analysis of the development explains how parts of the more complicated aspects of the system were implemented.

A final Project post-mortem is then undertaken which concludes that the method of approach, initial design and development was an overall success.

Contents

1. Introduction

Music Festivals are a great way for people to see their favourite bands, socialise and have some fun. The number of music festivals and their popularity in the UK has grown significantly in the 21st Century (Warman, 2010).

Research from PRS for music suggested that in recent years music festivals have generated a large amount of revenue for the UK economy (PRS, 2009), this means that for musicians, organisers and sponsors, music festivals are a strong source of income. To stay ahead of the competition, music festivals in the UK will no doubt have to put on an unforgettable experience for the festival goers to remember.

This report explores a project involved with the development of a Festival system that is aimed at bringing the experience of music festivals into the 21st century and give a competitive advantage to organisers by exploiting the increasing use of smart phones at festivals to deliver up-to-date content about a festival, such as line ups, location information and news, to its attendees.

2. Background

2.1. Technology at music festivals

Smartphones today are, by nature, designed to access information remotely. They are also becoming central to our communication and information needs (Butler, 2011). Coupled with this and that during 2014 1.3 Billion smartphones were sold worldwide (an increase of 27.7% from 2013 (IDC, 2015)) shows that at any festival, smartphones are going to be a popular device with festival goers; offering up-to-date information during a festival is one way that a festival organiser can meet the information needs of today's smartphone user.

One prominent issue with today's smartphones, which may present a problem being at a festival, is battery life. An experiment on the battery life of 60 different smartphones found that on average, the top three popular mobile platforms (iPhone, Android and Windows) had an average battery life of between 13.5 to 10 hours (Byrne, 2015). With most music festivals spanning a whole weekend it is obvious that festival goers' mobile devices would run out of power after a day.

One popular solution to keeping a smartphone from running out of charge has seen the rise in 'Charging stations' appearing at festivals. One popular provider, known as Charge Candy, offer a service where a festival goer can either hire a portable battery pack to stay charged on the go, or leave their phone to be charged and pick it up when it's full.

2.2 Similar applications

Appendix G is a document that conducts research into similar festival mobile applications. These mobile applications and how festivals maps are created give a good idea and provide the level of detail needed to create a system to improve on, most notably the glastonbury application. The level of detail in the map (locations of points of interests, walkways etc..) and scheduling information as well as the overall finishing, in terms of the user interface, of the product shows what users will expect to provide them with the experience they expect. The Bunbury application goes to show that even a static map can be just as effective. Meeting in the middle I believe that desiging a system that allows the festival organisers to

upload a previously designed map and then add points of interest on top of the map will work effectivly as festival organisers have usually already designed their own maps and would unlikely repeat efforts, for instance having to redraw roads and boundaries.

Through the analysis it can safely be determined that although there are some one off applications developed for a specific festival there is no generic,'off the shelf' product that would allow a festival to purchase and customise/set up for their own festival.

3. Project Aims and Objectives

3.1. Aims of the project

These are the aims of the project, as originally defined in the Project Initiation document (Appendix B).

- *To analyse similar systems (if any) to provide ideas for improvement to the design of the system.*
- *To investigate requirements of the end user that would allow the proposed system to deliver relevant information.*
- *To investigate requirements of festival organisers that would allow them to enter in relevant information to deliver to the end user.*
- *To develop a product/package that could be marketed towards festival organisers.*

3.2. System objectives

To create and improve the way that information is delivered to festivals goers in a way that they

- *They are quickly kept in the loop with scheduling of the acts.*
- *Can be alerted quickly to information such as cancellations and scheduling changes.*
- *Allow festival goers to easily find points of interest at a festival.*
- *Can advertise products and services at the festival to them.*
- *It will encourage them to advertise the festival via social media.*

3.3. Use Cases

Use cases define the user view of a system. They are normally accompanied by diagrams to visually represent actors and their tasks, along with descriptions of the tasks (Alhir, 1998).

To completely understand the objectives of both the system and the project, it is essential to explore the use cases for the proposed system. This has been achieved through the use of use case diagrams and descriptions see Appendix L and M.

4. Method of Approach

The method of approach for the development of this system will define how the project will be managed and organised. It will go into detail on possible methodologies and summarise on a chosen one, it will also outline the timeline that the project will undertake. As well as the methodologies and timeline, the risks associated with the project will also be described.

The PRINCE2 was chosen to govern the overall management of the project ('PRINCE' stands for PRojects IN Controlled Environments). This method is considered to be one of the world leaders in project management. Its activities that define how to direct, manage and deliver a project will greatly help the project to run smoothly throughout its course (Murray, 2011).

Software project management is an important part of developing a software product; both Sommerville (2007), Hughes and Cotterell (2009) agree that project management plays an important part in software engineering and that mismanagement of a project can result in it being over budget, over scheduled or even worse failing completely. Even though this project is small in comparison to the large scale projects Sommerville, Hughes and Cotterell might be referring to when discussing project management, Tsui and Karam (2007) mention that even some small software projects require some degree of project management.

4.1 Methodologies

This section will cover some well-known software process methodologies and conclude on the one that works the best for this project.

4.1.1 The waterfall model

Hughes and Cotterell (2009) describe the waterfall model as the 'classical' model for system development and summarise it as a 'one-shot' or 'once-through' process. As shown in Figure 1 the process follows a sequential step by step process in a downward order.

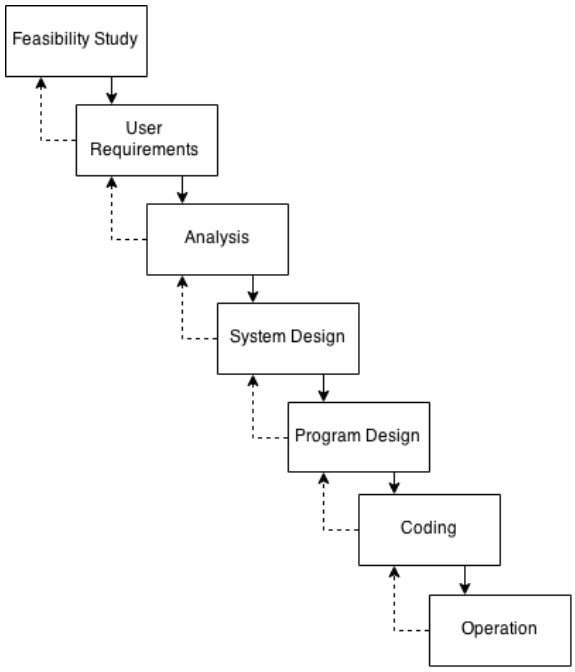


Figure 1 The waterfall model. Adapted from Hughes and Cotterell (2009)

The model allows some return to the previous step (called splash back) if an earlier stage requires some re-work. Hughes and Cotterell explain that it isn't encouraged. This rigidness of the model can be seen as a positive and a negative as on one hand it might be beneficial for some projects to have clear milestones set out at the end of each phase. This allows review points to make sure that the project is still valid and accurately determines easy forecasting of the project timeline. On the other hand it also means that projects should

avoid reworking the same task. If there is need for splashing back, it could endanger the project schedule.

Having each phase planned can be very useful for this project in terms of smooth running. It would however require needing to know and be very confident in the timeline of the project. Additionally, it does not allow for any chance to learn from what has been attempted and make unexpected changes; as this project is also attempting to try something relatively different, it could be seen as a learning curve, this could require occasionally having to re-attempt part of the project or make slight changes in the development. Due to waterfall's own rigid nature, and having too much splash back could therefore easily damage the delivery/finish date of the product.

4.1.2 The incremental model

The incremental model takes the project and breaks it down into sub components, It can be seen that each component is developed as a smaller 'mini' project. The result of each sub component gradually fulfils the requirements of the project as a whole. Figure 2 visualises the process of the model. Sommerville (2007) explains that this model differs from the waterfall approach as the incremental model only requires an outline of the required functionality in order of priority; waterfall would require a detailed requirements list in order to proceed. The outlined requirements for incremental would then be developed into a number of increments which would deliver a sub set of the required system.

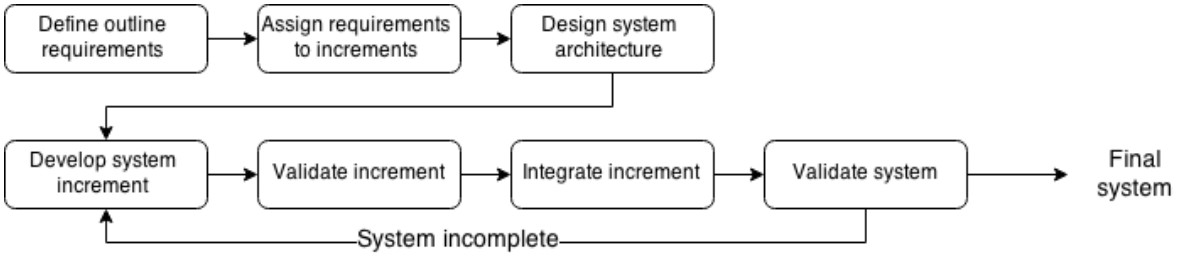


Figure 2 The incremental model. Adapted from Sommerville (2007)

A significant benefit of this model allows an early interpretation of how the system is looking which would allow the client to easily get used to and suggest changes to the system. In the case of applying this model to this project, it would easily allow an overview of what has been developed and what is still to be developed, any problems that arise with one stage of development will not mean the end of the project completely. It would just mean assessing the functionality that is required to determine if it's accomplishable or if any changes need making.

Motivation for the developers from working on each increment would be improved as they wouldn't be working on the same large project goals but more smaller short term goals as Hughes and Cotterell (2009) explain, job satisfaction is increased for the development team as they will be able to quickly see an outcome of their project at regular intervals instead of waiting till the end of the project to see a finished piece of work.

4.1.3 The spiral model

Sommerville (2007) describes the Spiral model (Figure 3) as something rather than a series of activities with some backtracking but represented as a spiral. Hughes and Cotterell describe it as a different interpretation of the waterfall model with greater emphases on the

success of each stage of the project. Each loop of the spiral signifies a phase of the project; each loop is split into four sections:

1. **Objective setting:**
   Objectives, constraints, a management plan and risks are identified.
2. **Risk assessment and reduction:**
   An analysis of each risk is carried out with steps taken to reduce it.
3. **Development and validation:**
   A development model is chosen depending on what is best suited to the type of work to be undertaken
4. **Planning:**
   The entire project is reviewed and a decision is made as to if it should continue onto the next loop.

Sommerville summarises that the main characteristic of this model is that each phase takes into account the risk at that point of the project. This is what sets it out from the majority of available models and makes it a strong contender for this project by allowing it to constantly adapt to the current situation at the end of a project phase

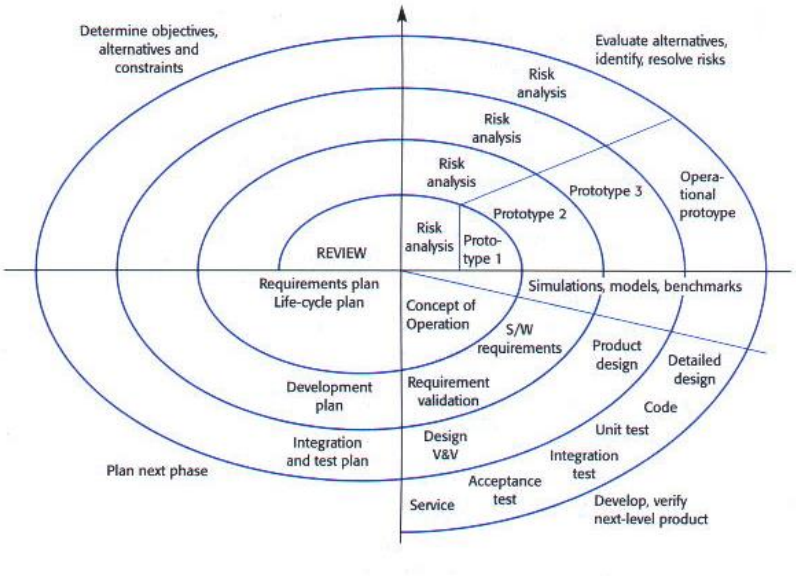


Figure 3 Boehm's Spiral model (IEEE, 1988)

Hughes and Cotterell (2009) describe this model as a different interpretation to the Waterfall model, applying it to this project should also take into consideration some of the areas of the waterfall model that don't fit well into this project. It might still require a strong, detailed plan for each of the loops as shown in Figure 3, the requirements elicitation is only done in the first loop, making this project very rigid to change. Even with the added benefit of being able to re-adjust the risks of the project throughout the whole process, it is something that could be easily implemented into most models.

Tsui and Karam (2007) mention that the Spiral model was developed from the experiences of large government software projects. Based on this fact, it might suggest that as this project is relatively small in size, therefore it might not be so appropriate.

4.1.4 Prototyping

To reduce ambiguity and increase the knowledge of the problem, we can implement prototyping. This can be seen as quickly as creating a basic, proof of concept version (or part) of the product without much expense (Hughes and Cotterell, 2009).

There are two different types of prototyping:

- **Throw away prototypes** are more of a proof of concept way to prove that the idea works. Sommerville (2007) suggests that creating an initial version of the system is useful to try out designs, show the concepts off and to discover what the actual problem is, and to show the solutions.
- **Evolutionary prototypes** are takes the initial version and continually develop to newer versions until the system is to a standard where it fulfils the requirements and is of standard that was originally promised (Hughes and Cotterell, 2009).

One of the biggest reasons to conduct prototyping is to reduce uncertainty by giving it a go. This allows better clarification of ambiguous requirements where there might be no previous example to follow; and also allows the end user to get involved with a tangible part of the system which then generates feedback to improve the end product. As this project is small in size and requires further experimentation to confirm all aspects of the initial requirements, it would be appropriate to create a throw away prototype (proof of concept) system to confirm that the idea works.

4.2 Conclusion

After assessing some methodologies that could have been suitable to this project, it is clear that the best available methodology to follow would be the incremental model. The incremental model has been chosen as it fits in the best with the size of this project. Both the waterfall and spiral models appear to accommodate more larger and defined projects.

This project will make use of the incremental model by developing smaller parts of the system to build it up into the final product. The use of throw away prototyping will also accompany this model in the definition and designing phases. It will allow a proof of concept system to be quickly developed to prove that the methods and technologies needed to develop the system work as envisioned. This should also help to clarify any ambiguous requirements.

4.3 Project timeline

After choosing an appropriate methodology to guide the process of the project, a more detailed plan can be made to the timing and processes that this project will go though. Appendix H shows the designed timeline of the project from start to finish (October 2014 to April 2015). It has been identified into three main phases;

1. **The project start –** This phase focuses on the initiation, requirements elicitation and the overall design of what is to be produced (System architecture, Database design and GUI styling)
    a. **Initiation:** The initiation of the project defines and develops the project initiation document. This document sets up the initial project goals and deliverables. It also goes over a brief timeline of the whole project and risk assessments.

   b. **Requirements elicitation:** This part of the phase goes into refining the requirements through research into similar systems, research into technologies that can be used to implement the proposed system. A throw away prototype (proof of concept) will also be developed to prove that the idea does work.
   c. **Design:** This section develops a design that will be followed in the development of the system. It will produce an architectural design for the overall system by showing how all the parts interact and a database design. It will also develop an initial plan for the GUI.
2. **The development –** This phase has three increments, each one will develop a key part of the system to eventually form a complete product. It will develop the database and background web service that ties together the interfaces to communicate with the database. The interfaces are developed in the second and third increments. The festival organiser desktop application in the second and the festival goer's mobile application in the third.
3. **The 'Wrap up'**
   a. **Final system testing:** Although testing will take place at the end of each increment in the development phase, a final validation of the entire system will take place to confirm all of the project's objectives have been met.
   b. **Project report:** This document

Before the beginning of each stage in the project a stage plan was created so that that phase of the project ran smoothly. See Appendix C for detailed stage plans.

4.4 Risk Analysis

Sommerville (2007) simplistically summarises that a risk is something that you wouldn't want to happen in the project you are managing; it is something that may negatively impact the project, the organisation or the end product. It is therefore important to document and manage any risks that could occur during a project in order to effectively cope with the problem.

Appendix I shows the initial risk list table that was developed from the project initiation document. This table is central to understanding what the identified risks are, and what would be the impact if they occurred during the project and what the course of action and the contingencies were needed to fix the problem.

4.5 Project organisation

Communication plan

The plan regarding the communication for main feedback and discussion was set out with the project supervisor on a bi-weekly basis to discuss progress as well as any necessary extra meetings that might be required to discuss urgent/other matters involved with the project. All other communication was via email; this was usually used for smaller matters.

Control plan

To complete this project it was estimated that around 13-14 hours of work a week on the project would be sufficient. A fortnightly highlight report of the progress was made in the form of meeting records with the supervisor, doing so not only created a track of my

progress of the project but allowed the supervisor to keep track as well. Meeting records with the project supervisor can be found at Appendix F. For records, two interim reports were also created (Appendix D & E) to discuss progress made during the project.

5 Design

This section will look at the layout and architecture at a high level to give a representation as to what the role and the interactions of each part of the system plays.

Also, this section goes into discussion about each individual part of the system that was required in order to make up the product. Each part will include details regarding the requirements, possible development technologies and user interface design.

5.1 System

5.1.1 Overview

In order for the system objectives to be met, the system was broken down into three significant parts:

- An application for festival organisers to create, edit and publish a festival, as well as being able to send out news updates.
- A mobile application for festival goers to receive a festival, with news updates, and browse its contents.
- A web service that allows:
    - To receive and process a festival from a festival organiser and process news updates.
    - To allow access to the festival information by the festival goers, as well as pushing out news updates for a festival.

Figure 4 graphically represents the communication flow between each parts of the system. In essence the web service handles communications from the festival organiser application and the request for information by the festival goer mobile application (including pushing of news items).

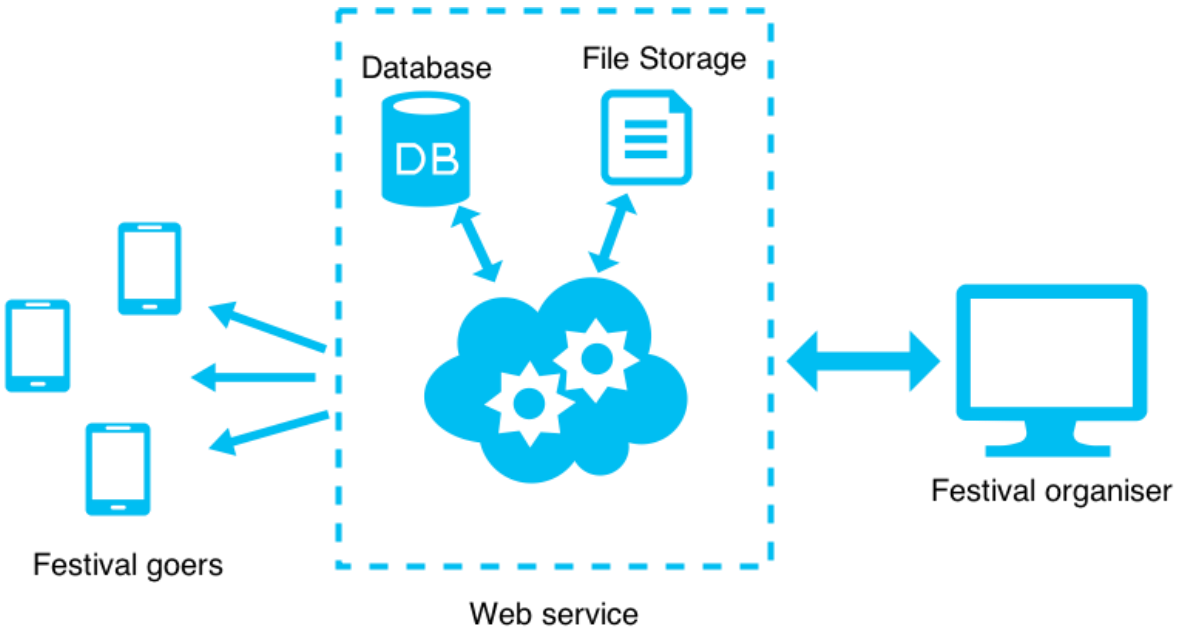


Figure 4 Architectural overview

The web service handles the communication to a database and file storage. The database will contain details regarding the festival organiser, a festival, its news items and festival goers (see 5.2.3 Database design for a detailed explanation of the database). The file storage will contain the main festival data, which includes image files, map data and line up data.

5.1.2 Activity Flows

Activity flows, sometimes referred to as Activity diagrams, are types of diagrams that represent the flow of control and information within parts of a system (Alhir, 1998).

The activity flows described in Appendix N explore the more critical features of the system where interaction between the Festival Goers' mobile application, Web service and the festival organisers' application will be required.

The features explored include:

- Viewing a festival map on a mobile device.
- Uploading a festival to the online access system (web service).
- Notifying mobile users of an important news item.
- Editing a schedule for a festival that is already online.

5.1.3 Programming language

An important factor of the design and a main influence of the quality of the end product will be the chosen programming languages that each element of the system will be developed in. This part of the design will look at the possibilities for the programming language that were considered for this project.

Java Solution

Java is an object-orientated programming language. Flanagan (2005) describes that Java was designed to make it easier for developers to write robust, bug free code. One of the key benefits of implementing a java solution include the notation that a developer can 'Write once, run anywhere', Flanagan goes on to say that this is the core value of the Java Platform. The benefit of using the java language for this project is code re-usage which could be heavily implemented between the festival organiser application, the web service and the festival goer mobile application. Code re-usage would mean a significant decrease in programming time, time that could be spent in ensuring the system meets all the requirements. There are further development tools that could be used to develop each part of the system, one example being using NetBeans to develop the Web service and Festival organiser application. Android Studio/Eclipse could be used to develop the mobile application (if the Android platform was the target).

C# (.NET) Solution

The .NET platform coupled with C#, another object orientated language, could be another possible alternative for at least the Festival organiser and Web service parts of the system. A mobile solution could also be developed for a windows mobile device. The .NET framework has an extensive library of classes and support that now allows a more open and cross compatible solution (within the bounds of Microsoft products) that would also allow some code reuse across the system (Landwerth, 2014).

HTML/JavaScript Solution

A web solution for this product was also a consideration. Using a combination of a server side programming language and HTML/JavaScript, the web service and festival organiser applications could be have been developed. Again, code reuse was a key quality that this solution represents. A mobile solution could have also been implemented with the use of a framework such as JQuery Mobile that offers a touch optimised solution for web sites accessed via portable devices (jquery.org, 2015).

Chosen language

Owing to Java's long term efforts and core value to be a cross platform programming language, the availability of multiple development tools and easily accessible support it was chosen to be the main development platform for the project. This decision means that as the Android SDK platform is Java based, it was chosen to develop a native mobile application for the festival goers. Implementing a Java solution would allow code reuse between the android mobile application, the festival organiser desktop application and the web service. This has been explained further with the development sections of this report.

5.2 Server Side (Web service)

5.2.1 Detailed requirements

The detailed requirements for the web service are as follows:

- Authorise Festival organisers to upload and change festival information.
  As part of making sure that only festival organisers can create and edit their own festival information, an authentication system will have to be implemented to ensure a secure system.

- Store the created map and festival data from the desktop application
  The web service must be able to handle the storing and retrieval of a festival file that contains data such as the image files required for a base layer map and Point of Interest images.

- Store the scheduling information
  The web service must be able to store the given scheduling information given to it from the desktop application.

- Allow access to a festival organiser's festival information.
  The web service must provide the means to allow methods to retrieve and update all the festival information from the festival organiser's desktop application.

- Allow the mobile application read access to the map and scheduling information
  The web service must provide the feature to allow the festival goer's mobile app to retrieve any festival information.

- Push out mechanism to send notifications to mobile users and keep a log of notifications
  The web service must be able to take a given message from the festival organiser application and push that message to all subscribed festival goers' mobile devices.

For the full list of requirements see Appendix Q.

5.2.2 Technologies

5.2.2.1 Hosting services

As the server side section of the project is developed to be accessed by festival organisers and festival goers it needs to be exposed on the internet in a secure and easily accessible way. One way of achieving this would be to use a provided web services solution that offers quick and secure access to the data required by the client applications. There are many solutions that are on offer. This part of the report will look into two of the most popular services, Amazon AWS and Microsoft Azure, which offer solutions such as relational database hosting, virtual servers and file storage.

Amazon AWS

Amazon Web Services (2015) (AWS) quotes itself as providing *"on-demand delivery of IT resources via the Internet with pay-as-you-go pricing".* In essence it is providing the means to offer cloud computing, which Amazon describes as a simple way to access servers, databases and other types of applications without the need to purchase expensive hardware to do the job.

As part of the design for the server-side system, a relational database service is required to hold the structure of a festival, its organisers and its attendees. RDS (Relational Database service), offered through AWS, makes it simple to set up, control and scale a MySQL, Oracle, SQL Server, or PostgreSQL database online. RDS offers the ability to manage time consuming database administration tasks, such as software installation, patching and backups so the developer/administrator can focus on other things. It also offers the ability to adjust capacity when required. For instance, if the deployed database was growing beyond its original size a simple allocation of more storage can be given.

To store the more complex festival data, such as the festival map image file, file storage will also be required. AWS provides a service called S3 (Simple Storage Service) which offers a fully redundant, high performance data storage infrastructure for any amount of any data. Like RDS, S3 allows scaling up its capacity and automatic backups. This, in essence, would future proof the storage needs for the festival system, allowing the storage to grow as it is required and allow any retrieval of lost data.

Another service offered by AWS, called EC2 (Elastic Compute Cloud), is a web service that offers resizable compute capacity in the cloud. Amazon claims that it is designed to make *"web-scale cloud computing easier for developers"*. EC2 provides a platform that allows developers to quickly obtain and boot new server instances within minutes; it also has a feature to quickly scale capacity depending on the computing requirements. Currently on offer by Amazon are a variety of preconfigured Linux and Windows distributions that can be selected and deployed automatically. An EC2 instance would be able to host a web service that the festival organiser and goer applications will require to compute and access data from the database.

Ease of accessing all these services that AWS offer when developing applications will also be important during the development phase. This is why amazon also offers a variety of SDKs and APIs to ensure developers can use their services easily on their chosen development platform. Currently Amazon provides support for Android, Browser integration, iOS, Java, .NET, Node.js, PHP, Python and Ruby.

In terms of pricing, Amazon offer a free tier to get developers started on their platform. This is available for 12 months, from registration and provides their services at a limited basis. After 12 months or if the usage goes above the limits they set out then charges apply. This is usually done on a pay as you go basis by charging for the resources used. The table on Appendix J shows a brief rundown of some of the pricing that is available for RDS, EC2 and S3.

Microsoft Azure

Microsoft's Azure platform (2015) describes itself as *"a growing collection of integrated services"* that allows developers to compute, store files and data, network and create applications in any operating system, language, tool and framework.

Unlike AWS, Azure appears only to promote using Microsoft's SQL server for database needs. It also promotes using Microsoft development tools, like Visual studio, to integrate applications. It does also include support for other development platforms such as .NET, Java, PHP, Ruby on Rails and Node.js. As well as this, It includes a 'near-zero' maintenance through a self-managed service which includes automatic set up and software patching.

Azure's data storage option, being very similar to Amazon's S3 offers high performance, low latency for input/output workloads which can be accessed under virtual machines.

In terms of virtual machines that could run a web service that the festival organiser and goer applications would require to compute and access data from the database, Azure has a broader range, in comparison to AWS, of operating systems to choose from, such as Windows, Linux, SQL Server, Oracle, IBM, SAP and BizTalk. With the ability to attach multiple premium storage options.

In terms of pricing, Azure offer a free $200 to get developers started on their platform. Charges apply on a pay as you go basis for the resources used. The table on Appendix K shows a brief rundown of some of the pricing that is available.

Although AWS and Azure offer very similar packages there are a few differences that set each other apart. Amazon appear to be more geared towards making the development on

their service a lot more easier by the use of allowing developers to easily integrate their services with developer's products through strong API/SDK support. This is in contrast with Azure who are very geared towards pushing Microsoft products, i.e. having a strong integration with Visual studio and SQL server. For this reason and that AWS have a free service for a whole 12 months, in comparison with a limited amount of money from Azure, the chosen web service platform was decided to be AWS.

5.2.3 Database design

The festival system relies on a database to store festival details, organiser information, festival goer information, news items and festival subscriptions. Below is a table that describes the purpose of each table in the database. Figure 5 shows an entity relationship diagram for the database that describes interactions between each table.

Table name	Description
Table Name	The primary purpose of this table is to hold the data of the festival. This is achieved by containing a URL field with the location of the saved data.
FESTIVAL	This is the table that holds the festival organiser information. It will be used to authenticate a festival organiser so they can make changes to their festival data.
FESTIVAL_ORGANISER	This is the area where news items will be stored for a particular festival.
NEWS_ITEM	This is the table that holds the information regarding a festival goer. It holds a special value that will be used for the Google Messaging service. This allows specific data to be sent to that festival goer's mobile device.
FESTIVAL_GOER	This is a simple table that allows a festival goer to sign up to a festival.
SIGN_UP	This is a simple table that allows a festival goer to sign up to a festival.

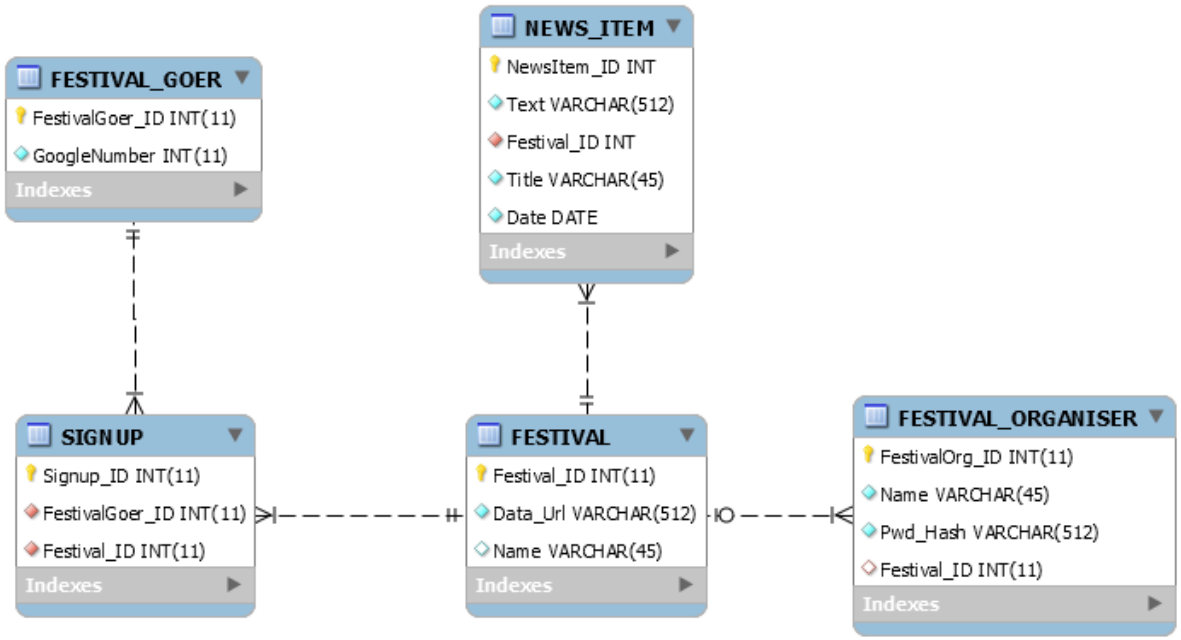


Figure 5 Entity relationship diagram for the database

5.2.4 Data access

A RESTful system was chosen as an access method as it is now becoming a popular alternative to such technologies as SOAP or SQL requests. REST also has the advantage of relying on simple HTTP actions such as GET, PUT, POST and DELETE (Belqasmi et al., 2012). This combined with plenty of support in the development community will make retrieval and submission of data to the web service easy to develop. Implementing a restful service will also assist in code reuse. As Java has been chosen as a development language, it should be a simple case of coding the http requests once, and reusing them in both the organiser application and the mobile application.

5.2.5 Administrator application

A small tool that will also interact with the web service will be an administrator application. This will allow the system admin to be able to simply add and remove the festival organisers. It will be a simple java application that will interact with the web service using the implemented REST API. Appendix O goes into a brief description of the GUI elements.

5.3 Festival Organiser application

5.3.1 Detailed requirements

The detailed requirements for the festival organiser application are as follows:

- Login to authorise use of uploading/changing data of a festival
  The festival organiser must be able to log in with their details to change or upload their festival.

- The ability to allow the user to upload a map (Image file) and then add 'points of interest' (pin drops) for all the locations required on top of the initial map.
  The application must support the ability to dynamically create a map and add/remove points of interest.

- Adding of scheduling information for each stage.
  The application must be able to add scheduling data for each day of the festival to a stage.

- Editing of the scheduling information in case of change or cancelation of any performance.
  The scheduling information must be able to be changed or removed durin, or before, a festival has started.

- The ability to push out notifications/alerts of any important news items regarding the festival to the users of the mobile application.
  The application must be able to send news items to the web service for processing. These news items are then handled by the web service and pushed out to the festival's subscribers.

For the full list of requirements see Appendix Q.

5.3.2 Technologies

The NetBeans IDE was chosen to develop both the Festival organiser application and the web service over the Eclipse alternative. NetBeans was proven to be the popular choice owing to including strong JDK support and coming with more advanced user interface design tools. Another more projected design orientated, reason owed to NetBeans having the tooling to automatically create a RESTful web service just from providing the database connection details.

5.4 Mobile application

5.4.1 Detailed requirements

The detailed requirements for the mobile application are as follows:

- Retrieve a pannable and zoomable detailed map of a music festival containing the locations of the points of interest; including camping locations and stages.
  The mobile application must include the ability to download a chosen festival from the web service and display the map developed by the festival organiser. It should include the points of interest.

- Each item on the map will be clickable to bring up a dialog of general information about the location.
  The mobile application must be able to interact with each point of interest on the map. On selecting a point of interest a description should appear.

- A schedule of each stage.
  Each stage should contain a scheduling of performances. It should be filter on selection of a date.

- The ability to create a personal schedule with notifications for favourite performances.
  The mobile application should allow the user to create a personal schedule of their own favourite performances.

- Notifications/Alerts to any important news items regarding the festival.
  The mobile application must allow the web service to send it messages if the user has subscribed to receive a notification about a particular festival.

For the full list of requirements see Appendix Q.

5.4.2 Technologies

Android SDK and Android Studio

As mentioned in section '5.1.3 Programming language' the Android SDK was chosen as a development platform for the mobile application as it is based on Java. Another reason the Android SDK was chosen owes itself to the popularity amongst smartphone users. Research carried out at the end of 2013 by Strategy analytics (2014) showed that the Android platform captured 79% of the market share throughout 2013. This research shows that picking Android will allow the mobile application to target a wide market. On top of Android owning a large market share, the developer already had experience in Android/Java programming. Using android as the development platform for the mobile application will allow the developer to use and improve upon their knowledge of Android programming.

Google's Android Studio has been chosen above the older Eclipse with Android development. It is clear from the Android documentation that Google are now supporting their product more than the eclipse tool as all of the tutorials have been updated with Android studio versions (Google, 2015). Using Android studio can be seen as a learning experience for the developer and to keep the mobile project in an up-to-date format.

Google cloud messaging service

Google Cloud Messaging for Android is a service that allows data to be sent directly from a backend service to users' Android powered devices. The kind of data that can be used includes lightweight push notifications that could tell the user that new information is available. (Google, 2015) This tool will easily integrate into the android application and will

be simple to develop on top of the web service. It will allow notification of new news items from the festival organiser right to the festival goers' android devices.

5.5 GUI Design

GUI designs were developed to visually demonstrate the layout and explain further the functionality of each part of the system. Appendix O shows the design for the System admin tool, the festival organiser tool and the festival goer mobile application.

6. Development Analysis

This part of the report looks at the more technical aspect of the project. It will explain the more complex parts of the development of the web service, the festival organiser application and the festival mobile application.

This section will also briefly explain a small prototype system that was developed, near the start of the project, to prove that the concept worked.

6.1. Initial prototype system

The initial prototype consisted of a rudimental web service that linked data from a very simple festival development tool with a basic festival viewer for a mobile application. The development of this system took place closer to the start of the project. It provided valuable knowledge including that a system like this could work.

This system was implemented on a closed network, for ease of accessing the data the festival development tool accessed a MySQL database that was controlled via a web service. Requests to and from the festival development tool used some basic RMI (Remote Method Invocation) to send the mapping data to the server. It was later discovered, on developing the mobile application, that android does not natively support RMI calls.

As a test, a RESTful service was implemented so that the android application could easily receive the festival data. This worked well using JSON and proved to be a much simpler mechanism for sending and receiving data. It also worked outside the closed network.

The prototype gave insight into the retrieval of data to and from the database. It demonstrated that RMI was not the solution and that a RESTful service that implemented JSON requests was far superior for this application.

6.2. Server side development

Amazon AWS

As mentioned in 5.2.2.1 Hosting services, there are external services that will be required to host a database, file storage and a virtual machine to host the web service. It was decided that Amazon AWS would be the right tool to implement this.

All parts were deployed using the AWS management console. Here, a developer can launch new instances of AWS products within minutes. The RDS, S3 and EC2 instances that were

launched were all set under the same VPC (virtual private cloud). This, in essence, was a private network that gave the developer control over the IP address range, creation of subnets, and configuration of route tables and network gateways.

The RDS instance was launched using the MySQL engine under a 'db.t2.micro' instance that Amazon offer under the free tier. It features 5 GB of SSD storage with automatic backups with a range of 7 days. The database was configured, as per the design, using MySQL Workbench which is a visual development tool for database architects that offers data modelling, SQL execution and other administration tools. (MySQL, 2015)

An S3 instance, as mentioned in the Design section, was required to store some of the public festival data created by the festival organisers. To create an S3 instance, what is called a 'Bucket' was created. A Bucket can be seen as a top level folder. Files and other folders are then stored within this folder. The developer was given full permissions to make any changes needed and the public was just given view permissions. For quick access this S3 instance was based in Ireland.

The EC2 instance, needed for containing the web service, was launched as a 't2.micro' Windows Service instance that was available under the free tier. It can be accessed from any machine that has the remote desktop port open with the correct Windows log in details.

Web Service

The primary function of the web service was to provide a RESTful service that linked the Organiser application and the mobile application with the Database. As discussed in the design, it was to be developed within NetBeans using the Java programming language. The web service was developed on the basis it was to be deployed on to a Glassfish server.

To save time hard coding RESTful classes to access the MySQL database, which was deployed via AWS, NetBeans offers an automatic system that pulls in the tables and generates the Entity Classes and RESTful services. All that NetBeans required was access to the database and for the developer to select the relevant tables.

A tutorial, located at https://netbeans.org/kb/docs/websvc/rest.html, was followed to do this.

To authenticate access within the web service, Basic Authentication was implemented on the Glassfish server. It was separated into two different Roles. The Admin, who could access everything, including PUT, GET, POST and DELETE requests to all the entities and the Organiser, who had access to retrieve and update festival information.

Push notifications

To implement this feature the Google Cloud Messaging API needed to be implemented on the web service and the Mobile application.

The Google cloud messaging (GCM) API requires two parts to function, a server and a client. The client part has been described in the development section for the Mobile application.

To implement the GCM service a project with Google had to be set up. This was so an API key could be obtained.

Upon receiving a news item from a festival organiser it is processed towards the database in the usual way. The web service then calls on the GCM API by pushing a notification using POST call with the following JSON content:

```
{
        “registration_ids”: “[signed up mobile id’s here]”,
        “data”: {
                “action”: “festival name + message”
                }
}
```

The google cloud messaging service then handles the push request and sends a notification to every device in the registration_ids field. Note the registration ids are stored in the FESTIVAL_GOER table.

System admin tool

The small system admin tool was developed to display the festival organiser details, create festival organisers and remove them. It uses the RESTful service to implement GET, POST, PUT and DELETE requests to the service.

The tool was developed in a way so that the system admin logs in with the admin logon details (through the implementation of Basic Authentication). After logging in, the tool then performs a GET request for the organiser details and displays that onto a table.

To manipulate the data, a PUT request is performed when the changes to the table is confirmed.

To add a new festival organiser, a new dialog is created which the admin will then fill out and confirm. A POST request is then implemented to send the data.

To delete a festival organiser a DELETE request is sent to the web service to remove the festival organiser, with a specific ID.

6.3. Festival organiser application

Networking

Any HTTP requests that are made by this application, have been separately threaded using the Thread and Runnable classes that are native to Java. This stops the application stalling on waiting for responses from the Web service.

Retrieving Organiser details

For the Festival organiser application to authenticate itself with the web service Basic Authentication was implemented. This was in a similar way to the System admin tool. Some code reuse was used here to generate and request authentication. After authentication has taken place, the requested festival organiser's details are retrieved via a GET request.

Upon receiving the festival organiser's details, the Festival data stored as a serialised file can be downloaded, de-serialised and loaded into the application. See uploading a festival for an explanation as to how a festival is serialised.

Interactive panel

Figure 6 shows the organiser view and the interactive map panel. Here the organiser can move, edit and create a map. This has been achieved by using the Swing GUI library that is native within NetBeans. Here a JLayeredPanel has been implemented. The bottom layer contains a 'map base' which is a bitmap of the map of the grounds for the festival. On top of this Jlabels with an icon are implemented to represent a point of interest. Using some basic transformation methods, it has been possible to click and drag elements within the map.

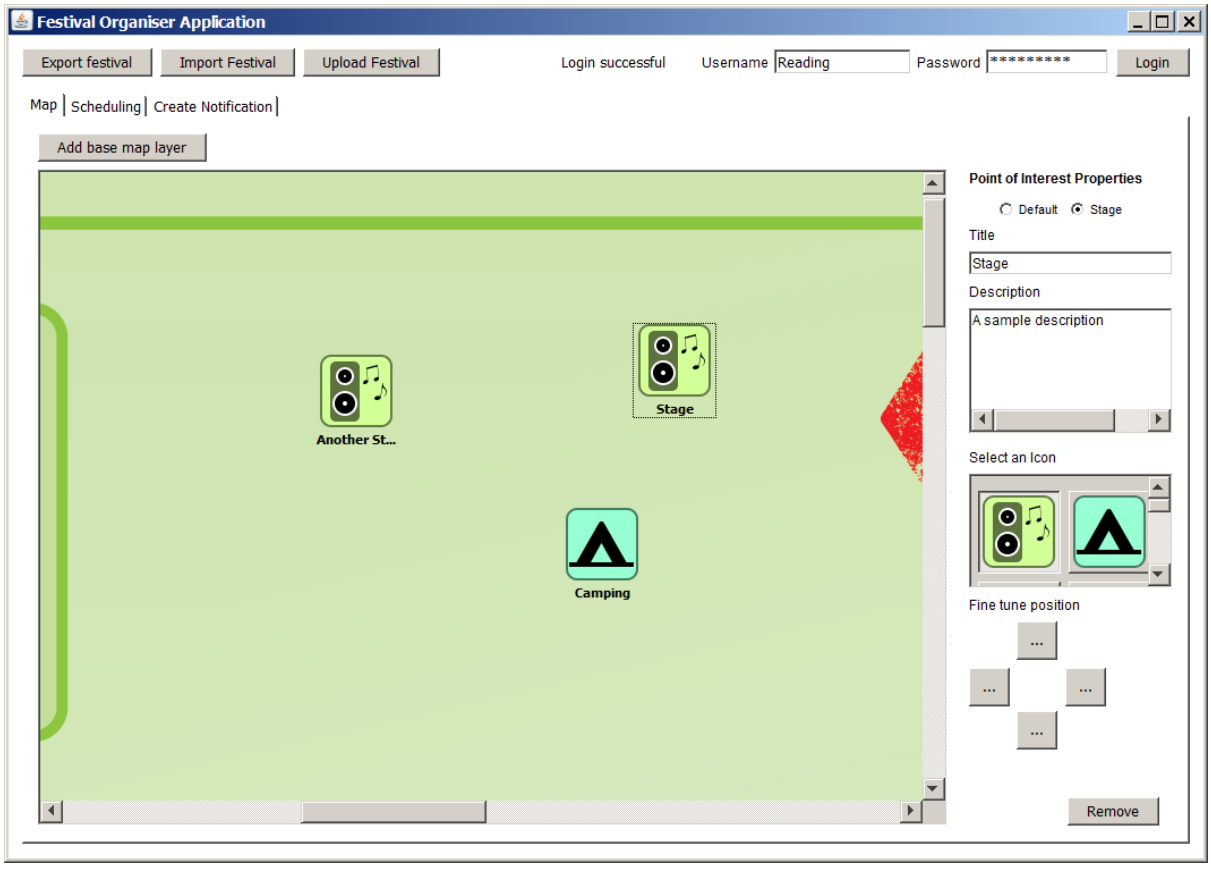


Figure 6 The organiser view and Interactive panel

Class structure

Appendix P shows the class structure for a Festival. This class structure was implemented by code reuse in both the Festival organiser application and the Festival goer mobile application.

Uploading a festival

To upload a festival, three stages must complete. Figure 7 shows the Upload Confirmation and Log dialog.

1. **Serialisation of the Festival into a file.**
   The Festival class that contains all the images, scheduling and points of interests is serialised into a file using the Java ObjectOutput class.
2. **Upload of the file to S3 storage.**
   The Amazon Java API is used here to connect to the 'bucket' in S3 and upload a public file:

```
        //Upload
s3Conn.putObject(new PutObjectRequest("festivaldata", fileName, inputFile));
        //Make Public
s3conn.setObjectAcl("festivaldata", fileName, CannedAccessControlList.PublicRead);
```

3. **Update of the database with the location of the new file in S3.**
   PUT request to the web service with the festival data and the new location of the newly serialised file.

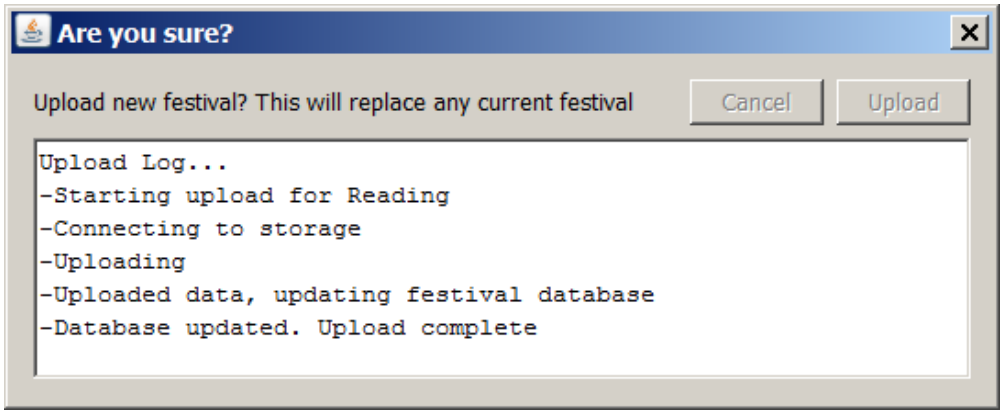


Figure 7 The festival upload dialog

Sending a news item to festival goers

This feature, although simple on the festival organiser application is a more advanced feature, as a whole, across the system.

This application plays a small role in just sending a POST request, containing a news item that is in a JSON format of the NEWS_ITEM table from the database, using the RESTful web service. See the section on push notifications in the Server side development for a greater explanation as to how this was developed to send the Posted news Item to the subscribed festival goers.

6.4. Mobile application

Receiving festival data

To receive the festival data, a GET request is made for a selected festival from the web service. This then retrieves the URL for the Festival file from the S3 storage (For details on

how the File is uploaded to S3 see 'Uploading a festival' in the Festival organiser Application section). Upon receiving the Festival file, it is de-serialised into a Festival class.

Displaying the festival information

To Display the Festival information, the map and POIs are displayed using a relative view. This view can then be translated via simple panning and pinch zooming. The POIs are displayed via an ImageButton that is part of the User interface classes that the Android SDK provides.

Scheduling information is displayed in a list view that is provided by a User interface class from the Android SDK, which filters by each stage and dates of a festival.

As the News information is data that can easily change, downloading the entire festival multiple times would be wasteful of the mobile user's data. It is therefore retrieved via a GET request from the web service instead.

Subscribing to a news item

To subscribe to a festival's news, a special method was created within the web service. Upon being called, via a GET request, if the requesting user is not subscribed to the given festival, an entry is made in the SIGN_UP table in the database and an entry is returned. If the SIGN_UP entry in the table exists for the user then it is removed. This method essentially flips if a festival organiser is subscribed or not.

Push notifications

To implement this feature, the Google Cloud Messaging API needed to be implemented on the web service and the Mobile application.

The Google cloud messaging (GCM) API requires two parts to function, a server and a client. The server part has been described in the development section for the Server side.

To be able to receive a notification the Mobile application must implement a few methods defined from the GCM documentation: https://developer.android.com/google/gcm/client.html.

The project has to be set up to retrieve the Google play services SDK which provides GCM. The next step is to register the mobile device with the GCM service (if it hasn't been already); upon doing so the phone retrieves a unique ID that defines the phone and the application.

To receive a notification from the web service, the application must set up a Service that receives and handles the given message. In the case of this application, a notification is given of the new news item on receiving a new message from GCM.

7. Testing Analysis

To confirm that the developed system conforms to its original requirements (See appendix R for each explained requirement in the design section) Integration testing was carried out.

The test plan was designed to test if the requirement met up to what it was originally expecting of the system. Out of the 16 tests carried out, 15 of them passed. This resulted in a 94% pass rate.

Below is an example and explanation of one of the tests carried out that passed and an explanation of the one requirement that failed.

Functional Requirements	Test Planned	Result to Pass	Achieved Result	Result
Web service				
The web service must authorise Festival organisers to upload and change festival information.	Test incorrect and correct login details.	Only the correct login details will be accepted.	Correct login details accepted and false ones rejected.	PASS

This example shows the requirement that states the web service should authorise the festival organisers. To confirm that this requirement has been met a planned test was carried out. As the test met the 'Result to Pass' criteria it passed.

Functional Requirements	Test Planned	Result to Pass	Achieved Result	Result
Mobile application				
The mobile application should allow the user to create a personal schedule of their own favourite performances.	Test that the mobile application can create a personal schedule.	A personal schedule is created.	No feature exists.	FAIL

This example shows the only failed requirement. This failed owing to the personal schedule feature not being implemented in the mobile application. See the critical evaluation for an explanation of this not being implemented.

8. Project Post-Mortem

8.1. Use of technologies

During the design of the Festival system the technologies were evaluated for their potential to make sure that they could be worked with by the developer and implemented in a way

that fulfilled the functional requirements. As shown in the previous section (7. Testing Analysis), 94% of the requirements passed the testing. This demonstrated that the deployed technologies had been implemented in a way that had fulfilled the original requirements.

Amazon AWS

Although a steep learning curve, Amazon AWS offered a variety of tools and services that allowed a robust backend to be deployed. After the education that took place in the design stage deploying the required services (Relational Database Service, Simple Storage Service and Elastic Compute Cloud) was a fast process. After being deployed they could be easily configured to work with the Festival system.

Java

Using Java (alongside NetBeans and Android Studio) performed as expected. Code reuse was an important part of reducing development time. It occurred with the more complicated parts of the programming, the REST/HTTP Connection classes, Festival data Classes and SQL data classes were just some of the areas where code was reused between the Organiser application and the Mobile application.

Android Studio/Android SDK

Originally it was specified in the Project Initiation document that Eclipse with the Android SDK would be used to develop the mobile application. As a learning exercise, and to stay with the supported IDE, Android Studio was chosen instead. As a learning exercise this paid off well, and in the end a mobile application that met all but one of its requirements was met. One issue that was experienced with Android Studio that severely increased the development time was issues within the IDE. The reason for the failed requirement (the user must be able to create their own personal schedule) not being met has been put down to these issues.

Android Studio, being an immature product had issues with freezing within the coding area and building with the Gradle tool. If this system were to be re-attempted it should be recommended that an alternative to Android Studio is chosen or stalled until a time where the defects that cause unproductivity within the product had been fixed.

8.2. Development Methodology

As stated in the Methodologies analysis that took place at the start of the project, the chosen development methodology was an incremental model. This allowed the development to be broken down into smaller, more achievable parts. 94% of the functional requirements being met it can safely be determined that this development methodology was applied well to this project. It could be argued that the development being broken down into smaller parts meant that if one increment overran it would push forward the next. If the project were to be undertaken again further research into combating overrunning schedules should be carried out.

8.3. Project management

The project management system that was employed for this project, as stated previously, was PRINCE2. The main part of this framework was the implementation of designed stages before any development had begun (See Appendix C for the stage designs). They gave a

beneficial system to create a process of how the project should proceed. It should be noted that eventually, due to development oversights (especially issues with Android Studio) the stage time constraints overran beyond any expected level.

Although the time constraints ran beyond what was originally expected it should be noted that the original contingency plans (from the original risk analysis, see Appendix I) did not need to be enacted, as no other risks were identified throughout the project it could be noted owing to the foresight during project planning and design, the project ran without any risks arising.

The reality that the project development overran due to technical faults with development software alongside the persistence by the developer to meet the functional requirements suggests that there was a heavier emphasis on focusing exclusively on the technical and development elements of the project, rather than the project management. In light of this it should be put forward that upon undertaking a similar project like this one more emphasis should be paid towards the management of the project.

8.5 Project summary

8.5.1 Achievements

The outcomes of this project had some notable achievements:

- **Deploying a web service with database and file support through Amazon AWS.**
  This was one of the biggest achievements of the project. The research and development showed its success for not only for the end product but as an educational tool for personal development. Amazon AWS has some big success stories (such as Dropbox, Spotify, Netflix, Uber, Samsung and even the CIA (Amazon AWS, 2015)) that go to show that these types of cloud services that they offer are becoming a relevant part to content delivery.

- **Google Cloud messaging integration.**
  This API from Google enabled notifications to be easily send out to festival goers. This enables festival organisers a fast content delivery platform that has never been tried before, at least in the UK.

- **Content delivery of Festival Scheduling and Mapping.**

- **Something 'new' attempted.**
  This kind of system that offers multiple festival organisers a platform to publish their festival to festival attendees has not been deployed before (as shown in the original background research).

8.5.2 Project Objectives analysis

- **To analyse similar systems (if any) to provide ideas for improvement to the design of the system.**

This project objective has been met, the research that went into similar systems and current technologies at music festivals demonstrated that although there were independent systems for music festivals, no existing systems allowed a repository of festivals to be stored and accessed.

- **To investigate requirements of the end user that would allow the proposed system to deliver relevant information.**
  Although existing technologies and similar applications were researched in detail it could be suggested that this objective was met. However, no primary research against the festival goers was carried out to reinforce the findings.

- **To investigate requirements of festival organisers that would allow them to enter in relevant information to deliver to the end user.**
  Although existing technologies and similar applications were researched in detail it could be suggested that this objective was met. However, no primary research against the festival organisers was carried out to reinforce the findings.

- **To develop a product/package that could be marketed towards festival organisers.**
  As 94% of the functional requirements passed the testing it can be safely assumed that a product aimed at the festival market has been developed correctly and marketing can now take place.

8.5.3 Lessons learnt

Throughout this project and through the analysis of the entire process lessons have been learnt, they are:

- Primary research of end users should be carried out to confirm other research
- Immature, new software should be assessed before being developed with to ascertain if there are any flaws that could impede the development.
- Further research into development methodologies should look at scheduling overruns.
- Technical development and Project management prioritisation should be assessed to ensure both have adequate attention.

8.5.4 Future Development

Future development of this project would like to see:

- Marketing of Festival towards festival organisers.
- Further testing to ensure a secure and stable system against high usage of the web service.
- Social media integration in the Mobile application.
- Location pinpointing on the map of the Mobile application.

9 Conclusions

Notwithstanding the one functional requirement not being met and two of the project objectives not being fully completed it can be viewed that this project and the delivered product can be seen as a success. Integration testing proved that the majority of the functional requirements had been met and some of the project objectives were achieved.

The overall goal of the project and its system was "*To create and improve the way that information is delivered to festivals goers".* The end result has produced a system that allows festival organisers to design and deliver their own festival and present this information right into the festival goers' smartphones. It should be noted that further development (see section 8.5.4. Future Development) should be undertaken before this system is deployed and marketed towards festival goers.

10 Bibliography

Alhir, S. (1998). *UML in a nutshell*. Cambridge: O'Reilly.

Amazon Web Services, Inc., (2015). *AWS | What is AWS - Cloud Computing with Amazon Web Services*. [online] Available at: http://aws.amazon.com/what-is-aws/ [Accessed 23 Mar. 2015].

Belqasmi, F., Singh, J., Bani Melhem, S. and Glitho, R. (2012). SOAP-Based vs. RESTful Web Services: A Case Study for Multimedia Conferencing. *IEEE Internet Comput.*, 16(4), pp.54-63.

Byrne, K. (2015). *Best battery life 2014 - 60 smartphones tested*. [online] Expert Reviews. Available at: http://www.expertreviews.co.uk/mobile-phones/1402071/best-battery-life-2014-60-smartphones-tested [Accessed 21 Mar. 2015].

Butler, M. (2011). Android: Changing the Mobile Landscape. *IEEE Pervasive Comput.*, 10(1), pp.4-7.

Charge Candy, (2015). *Charge Candy - More Than Just Phone Charging*. [online] Available at: http://chargecandy.com/ [Accessed 21 Mar. 2015].

EE, (2015). Glastonbury. [online] Available at: http://ee.co.uk/ee-and-me/entertainment-sport/music/glastonbury [Accessed 17 Mar. 2015].

Flanagan, D. (2005). *Java in a nutshell*. Sebastopol, CA: O'Reilly.

Google, (2015). *Android Studio Overview | Android Developers*. [online] Developer.android.com. Available at: http://developer.android.com/tools/studio/index.html [Accessed 25 Mar. 2015].

Google, (2015). *Cloud Messaging Support*. [online] Google Developers. Available at: https://cloud.google.com/tools/android-studio/messaging/ [Accessed 25 Mar. 2015].

Hughes, B. and Cotterell, M. (2009). *Software project management*. London: McGraw-Hill Higher Education.

Ianker.com, (2015). *ANKER*. [online] Available at: http://www.ianker.com/ [Accessed 21 Mar. 2015].

jquery.org, j. (2015). *jQuery Mobile*. [online] Jquerymobile.com. Available at: http://jquerymobile.com/ [Accessed 25 Mar. 2015].

Landwerth, I. (2014). *.NET Blog*. [online] .NET Blog. Available at: http://blogs.msdn.com/b/dotnet/archive/2014/12/04/introducing-net-core.aspx [Accessed 25 Mar. 2015].

Microsoft Azure, (2015). *What is Azure?*. [online] Available at: http://azure.microsoft.com/en-us/overview/what-is-azure/ [Accessed 24 Mar. 2015].

Murray, A. (2011) *PRINCE2® In One Thousand Words white paper*. [Online] Available at: https://www.axelos.com/case-studies-and-white-papers/prince2-in-one-thousand-words [Accessed 17 Mar. 2015].

MySQL, (2015). *MySQL :: MySQL Workbench*. [online] Mysql.com. Available at: https://www.mysql.com/products/workbench/ [Accessed 26 Mar. 2015].

PRS, (2009). *British festivals to generate £450m in 2009*. [Online] Prsformusic.com. Available at: http://www.prsformusic.com/aboutus/press/latestpressreleases/Pages/Britishfestivalstogenerate450min2009.aspx [Accessed 15 Mar. 2015].

Sommerville, I. (2007). *Software engineering*. Harlow, England: Addison-Wesley.

Strategy Analytics, (2014). *Android Captured 79% Share of Global Smartphone Shipments in 2013*. [online] Blogs.strategyanalytics.com. Available at: http://blogs.strategyanalytics.com/WSS/post/2014/01/29/Android-Captured-79-Share-of-Global-Smartphone-Shipments-in-2013.aspx [Accessed 25 Mar. 2015].

Tsui, F. and Karam, O. (2007). *Essentials of software engineering*. Sudbury, Mass.: Jones and Bartlett Publishers.

Warman, J. (2010). *How music festivals are singing the changes*. [Online] the Guardian. Available at: http://www.theguardian.com/business/2010/aug/27/music-festivals-record-industry [Accessed 15 Mar. 2015].

11 Appendix

A. User Guide

Installation:

System Requirements:

Festival organiser application

- Java 1.8 Compatible PC.
- Internet connection
- NetBeans IDE

Festival Mobile application

- Android Mobile Device
- Android Version > 4.1
- Internet connection
- PC with Android Studio.

The web service requires no set up as it is already set up.

How to run the Festival organiser application

- Load the project into NetBeans and resolve the dependencies from the supplied libraries.
- Build and run the application.

How to run the Festival Mobile application

- Copy the supplied .apk file on to the mobile device.
- Open and install the .apk file
- Run the installed Festival applicaition

NOTE: Appendix O shows the GUI elements that are detailed enough to be used as a visual user guide

B. PID

1. Introduction

Music Festivals are a great way for people to see their favourite bands, socialise and have some fun. With constant advancements in musical technologies, the increase of publicity/marketing and number of festivals taking place in the UK, many people are taking their summer take part in the festival experience.

Today festival organisers and sponsors are willing to put in the time and money to deliver a lifetime experience to promote bands and their own brands. On top of this festivals attract many different types of companies that want to sell their own product to festival goers (i.e. food, clothing, services). All this requires a lot of organisation, even for smaller festivals.

2. Business case (Background/Motivation section.)

2.1. Business need

With the push of mobile charging stations at festivals and the flood of portable battery packs in the market more and more festival goers are taking smart devices to festivals. My ambition is to take advantage of this and develop a system that allows people to make the most of a music festival to give them the best possible experience by delivering them up to date information regarding the festival.

Delivering up to date content and information to festival goers right to their mobile phones will allow festival organisers to not only advertise schedules, products and sponsors to the festival goers but will give them the opportunity to socially connect with the attendees of the festival.

With no all-round festival apps in the market that allow festival organisers to deliver maps, scheduling and advertisements to festival goers will allow me to develop a package to market to organisers upon completion of the proposed project. Not only will I have a potentially sellable product at the end of this project I will have also helped develop my own expertise in mobile/web app development as well as my database management knowledge.

2.2. Business objectives

To create and improve the way that information is delivered to festivals goers in a way that

- They are quickly kept in the loop with scheduling of the acts.
- Can be alerted quickly to information such as cancellations and scheduling changes.
- Allow festival goers to easily find points of interest at a festival.
- Products and services at the festival can be advertised to them.
- It will encourage them to advertise the festival via social media.

3. Project objectives

It is good to have in mind major project outcomes and deliverables to allow testing of the final system to show I have reached my goals. Below are the objectives that will be completed by the end of the project:

- To analyse similar systems (if any) to provide ideas for improvement to the design of the system.
- To investigate requirements of the end user that would allow the proposed system to deliver relevant information.
- To investigate requirements of festival organisers that would allow them to enter in relevant information to deliver to the end user.
- To develop a product/package that could be marketed towards festival organisers.

4. Initial scope

The final product developed with compromise of three parts:

- A mobile application aimed at festival goers.
- A web application/desktop application (more analysis required to determine best platform) for the festival organisers to use to set up and maintain the information* sent to the mobile application.
- A database and service to contain the information*, receive data from the web application and 'forward' it to the mobile application.

**the 'information' is defined in the initial feature list below.*

Essential Feature list:

A mobile application aimed at festival goers will deliver:

- A pannable and zoomable detailed map of a music festival containing the locations of the:
    - Stages
    - 3rd Party Companies; fast food stalls/marquees; merchandise stalls/marquees;
    - Bars
    - Other entertainment
    - Toilets/Showers
    - ATM machines
    - General convenience marquees

- Each item on the map will be clickable to bring up a dialog of general information about the location:
    - Stages shall contain a schedule of events.
    - Other places shall have opening times.
- A schedule of each stage.
- The ability to create a personal schedule with notifications for favourite performances.
- Notifications of any last minute changes to the scheduling of the performances.
- Notifications/Alerts to any important news items regarding the festival.

A Desktop/Web app aimed at festival organisers will deliver:

- The ability to allow the user to select a location for the festival and to then map out (or upload a map) by adding 'widgets' (pin drops) for all the locations required.
- Adding of scheduling information for each stage.
- Editing of the scheduling information in case of change or cancelation of any performance.
- The ability to push out notifications/alerts of any important news items regarding the festival to the users of the mobile application.

A Database to:

- Store the created map from the desktop/web app
- Store the scheduling information
- Keep a log of pushed out alerts/notifications
- Allow the mobile application read access to the map and scheduling information
- Allow the desktop/web app full access to the stored information.

Optional 'nice to have' features:

- Social media integration on the mobile application:
    - Links to festival social feeds, twitter and facebook
    - Ability to submit/share photos to twitter and instagram
- Map and schedule information will be available offline on the mobile application
- Forecasted weather for the festival on the mobile application
- The ability to pin point the mobile user's own tent and car on the map.
- The ability to save the mobile user's favourite places on the map.

5. Method of approach

To complete this project I will follow an incremental approach whilst following an iterative model. At the end of each increment I will have a prototype I can use to test any implemented functionality. This will allow me to quickly develop a system that will enable me to constantly re-assess functionality and design by analysing requirements at the end of

each iteration. Doing this allows me to adapt the feature list and design if need be throughout the development of the system.

I will follow three increments:

1. Develop a method of editing a map to store points of interest on a desktop application and display the map on a mobile device.
2. Make any changes based on analysis of first prototype and implement other required features.
3. Make any changes based on analysis of the second prototype and apply any 'nice to have' features.

Although a full evaluation will be needed during the project I am planning on using the Android development kit to develop the mobile application as it will enable me to quickly produce an application in a language I am already familiar with, Java. To limit having to re-do my work in another language I will base the desktop/web application in Java/Javascript, most likely using Netbeans as a development platform. For the database I am planning on using something like MySQL/SQL Server, further elicitation will be required to decide on what will be the best technology to use.

6. Initial project plan

Project Plan		
Stage	Deadline	Product/Deliverables/Outcome
1. Initiation	15/10	PID
2. Investigation and requirements	2/11	Analysis of existing any existing packages Requirements document Evaluation of possible development Technologies Proof of concept application
3. High level design	15/11	Design documents (Architecture; DB schema; modular decomposition; GUI style guide; …)
4. Increment 1	12/1	Prototype system, (Mobile application and Desktop/Webapp with database)
5. Increment 2	4/2	Updated prototype from feedback
6. Increment 3	4/3	Final full feature system.
7. System and user acceptance Testing	12/3	Final system as a package; user training
8. Assemble and complete final report	29/4	PRCO303 Report

Stage two plan

Task	Deadline	Outcome
Analysis of existing any existing packages/applications	16/10	Research and analyse any similar systems that do one or more of the requirements in this document.
Evaluation of possible development Technologies	21/10	Use previous analysis of existing packages to determine and finalise development technologies.
Requirements document	24/10	Use the analysis of existing applications and development technologies to create an initial requirements document.
Proof of concept application	2/11	Create an application to prove the integration of setting up a map on a computer and allowing a mobile device to retrieve and browse the map.

6.1. Control plan

To complete this project I will aim to roughly hit around 13-14 hours of work a week on this project, this roughly equates to just under 3 hours a day during the week, allowing me the weekend to catch up if I fall behind. I will aim to get every deliverable achieved by at least a week in advance, doing so will allow for any 'slippage' or unexpected work I may have to do. I will also product a fortnightly highlight report of the progress I have made, doing so will allow not only myself to keep track of my progress but allow my supervisor to keep track as well.

6.2. Communication plan

I will communicate with my project supervisor on a bi-weekly basis to discuss progress as well as any necessary extra meetings that might be required to discuss urgent/other matters involved with the project. I will also communicate via email for smaller matters.

7. Initial risk list

Risk	Probability	Impact	Action
Overrunning on schedule	Medium	Medium	I have allowed myself at least a week before deadlines to finish each deliverable, this gives me an overflow to achieve the work that has not been completed.
Difficulty learning new technologies	Medium	High	I will create a proof of concept system to test not that not only the technology works but I can develop

			with it.
Loss of work through corruption or deletion.	Low	High	Multiple backups of every piece of work I undertake will be created weekly. This will allow me to restore the project and only be a week behind at the most.
A required feature is too ambitious to be created owing to time or technology constraints.	Low	Medium	An analysis of requirements will be undertaken at the start and throughout the project, if one is too ambitious other features will still make the project successful.
Change in requirements	Low	High	If any changes of requirements are needed then a full analysis of how to fit in the new/changed features should be undertaken.

8. Quality plan

Initial Quality Plan	
Quality Check	**Strategy**
Mobile Functionality	Complete functionality checked at the end of stage 7 by testing every feature implemented to see if it meets the mobile application requirements.
Web/Desktop Application Functionality	Complete functionality checked at the end of stage 7 by testing every feature implemented to see if it meets the application requirements.
Database Functionality	Complete functionality checked at the end of stage 7 by testing every feature implemented to see if it meets the database requirements.
Requirements	Requirements checked during stage 2 to make sure they are correct, relevant to the business objectives, complete and achievable.
Usability and accessibility, Validation and user testing	During stage 7 the system will be tested against key HCI principles to make sure the interfaces are usable and accessible as well as further testing that the requirements of the system have been met.

C. Stage Plans

Stage two Plan

Task	Deadline	Outcome
Analysis of existing any existing packages/applications	16/10	Research and analyse any similar systems that do one or more of the requirements in this document.
Evaluation of possible development Technologies	21/10	Use previous analysis of existing packages to determine and finalise development technologies.
Requirements document	24/10	Use the analysis of existing applications and development technologies to create an initial requirements document.
Proof of concept application	2/11	Create an application to prove the integration of setting up a map on a computer and allowing a mobile device to retrieve and browse the map.

Stage 3 Plan

Task	Deadline	Outcome
Develop Architectural diagrams: • Use case Diagrams/Example Descriptions • Activity flows	07/11	A full understanding through architecture diagrams of how the final system will operate
Develop a GUI style guide.	15/11	A Styling guide created to show the GUI elements/Aspects of the: • Desktop Application • Mobile Application • Web Service/Server application
Amalgamate/tidy up proof of concept system to a development project to allow the start of development for Stage 4.	15/11	A single project/multiple projects taken from the proof of concept system that is effectively commented and has a clear and understandable flow to it. This will allow effective development to take place for stage 4

Stage 4 Plan

Task	Deadline	Outcome
Finish writing up GUI Styling	21/11	A GUI style guide that sets the basis for the Festival Organiser, Online access desktop applications and the Festival Viewer mobile application
First Interim report	02/12	A report highlighting: • tasks undertaken and outcomes • products produced and product quality • risks that have materialised and your response; changes to the risk list • schedule • resources (e.g., IT resources) • Learning undertaken & required
Design and Construct Database tables for holding Festivals and festival organisers	12/1	A database that should be accessible via the system
Start developing online access system.	12/1	The System should implement all basic functionality.

Stage 5 Plan

Task	Deadline	Outcome
Full write up of data model that was implemented for the online access system.	4/02	Write up of the recently created data model for records, to be used in final report and any documentation.
Investigate and Implement some basic authentication/restriction on REST service for the Online access system.	23/01	Restrict access to those who only need it and authenticate where necessary.
Develop the Online access system desktop application.	31/01	An application (as per the requirements) that interacts with the created background web service.
Review of Online access system, does it fulfil the requirements?	4/02	A review to see if all of the requirements have been met, if not, what was the reason behind this?
Start work on Desktop application for the festival organisers.	On-going into Stage 6	Make a head start on developing the desktop application that the festival organisers will use.

Stage 6 Plan

Task	Deadline	Outcome
Create a skeleton GUI of the desktop application.	6/2	A skeleton GUI for the Festival organiser application based on the original designs is developed.
Implement creating a pannable map on the desktop application and authenticating Organisers.	Mid February	The pannable map and authentication is built into the skeleton.
Implement the scheduling of the desktop application.	Late February	Scheduling feature is added.
Create a skeleton GUI of the mobile application.	1/3	A skeleton GUI for the Festival goer mobile application based on the original designs is developed.
Develop retrieving data from the web service onto the mobile device.	4/3	Mobile application can receive Festival information using REST from the web service.
Implement the map and scheduling views on the mobile device.	4/3	Mobile application can download and view the map and scheduling.
Implement the Google Cloud messaging service across the system	4/3	The system should be able to receive News items and push them to the mobile devices.
Create a test plan for Stage 7	On-going to stage 7	A test plan is created based upon the functional requirements.

Stage 7 Plan

Task	Deadline	Outcome
Carry out test plan	6/3	Test plan highlights areas that have failed and need work.
Analyse test results and fix problem areas.	9/3	Highlighted problem areas are fixed.
Fix any defects from development.	12/3	Defect rundown complete. Any identified defects are fixed.
Re-run testing.	12/3	Test highlights any areas that have failed.
Start Assembling final report material	On-going to stage 8	Materials, such as appendices and extra information needed for the final report is gathered.

Stage 8 Plan

Task	Deadline	Outcome
Draft Final Report	Late March	Draft report is developed for proof reading.
Proof read final Report	Early April	Report is refined and edited.
Repeat until satisfied with content	29/4	PRCO303 Report is produced that shows the development of the Festival product.

D. First Interim report

1. Introduction

The following Interim report contains an update on the progress of Festival viewer project so far by comparing what was planned to be produced and what has actually been produced along with reasons why.

The Festival viewer project is aimed at providing festival goers with up-to-date content about the festival they are currently at (content includes, mapping and scheduling information). The system will be split into three main parts; a festival organiser desktop application for festival organisers to build the required data, a web service/storage mechanism (dubbed the 'Online Access System') to host the information and a festival viewer mobile application to access the data in a way that it is easily accessible for the festival goer.

The first three stages (as outlined in the Project Initiation Document (PID)) of the project have been undertaken and completed with useful outcomes. These stages have primarily focused on research and creating a proof of concept system to create useful design documentation.

2. Tasks undertaken and outcomes

The table below goes into detail about each task that have been undertaken up to this point and if they deviated from the original plan[1] and why.

Task	Description/Outcome	Current Status
Produce a Project proposal	This provided the initial description of what the system will be. This was accepted by the project supervisor allowing creation of the PID.	Finished and accepted
Produce a Project Initiation document	This task was designed to outline what the proposed system is to be, the Business case behind the system, the project objectives, the initial scope, the method of approach, and the initial project plan. This was accepted by the project supervisor.	Finished and accepted
Research into similar systems	Developed within the original plan this was a key task to make sure that it was worth proceeding with the project as well as gaining inspiration for the proposed system. Doing this gave confidence that this idea was relatively unique and motivated development of a proof of concept system.	Finished
Research to refine details of the intended feature list.	This task was a deviation from the intended plan as it came to light that the feature list was vague on highlighting what Mobile platform would be targeted, whether the festival organiser application would be a desktop application or web application and how will the system allow a mobile device to access a map created by a festival organiser.	Finished
Produce a proof of concept system (poc)	This task was planned for the end of the second stage of the project plan, the development of this overran by a couple days as it was crucial to make sure that the key feature of being able to create a map on a desktop and access it on a mobile device. The completion of the poc demonstrated that it was possible to develop this key feature as well as motivation further to continue with this concept.	Finished
High level designs (Architectural and GUI)	The High level designs of how the proposed system will operate and look are key to showing any stakeholders some progress of the design as well as helping to start development of the system. As any extra research and development takes place the	Initial designs complete/ On going

	design may slightly deviate. This task will therefore be an on-going process.	
Research into database/storage systems	This is a new task that arose from the High level designs that it would be easier to develop a small database used for authenticating the festival organisers on the system. This research is currently on going and will highlight why this is required, what resources should be allocated for this and table designs.	On going

3. Products produced and product quality

Product	**Description and Level of quality**	**Current Status**
Project initiation document	This document's (described in section 2) quality was good enough to be accepted by the project supervisor to allow proceeding with the proposed project.	Finished
Analysis of similar systems[2]	This document was the produce of the task to research in to similar systems. It shows the descriptions and screenshots of each system that had similar aspects to what was intended to be produced. The quality of the document is up to standard as it gave some insight into what the current market of Festival mapping looks like.	Finished
Decisions based on Stage 2 research[3]	This was the deliverable from the research from the similar systems as well as the refinement of the feature list. The quality of this work allowed documentation of decisions in an effective way which led on to developing a proof of concept system.	Finished
Proof of Concept system	The proof of concept system was a big milestone in the second part of the project. It was a real life development (and by the end a produced system) that demonstrated a system that allowed a festival organiser to create a basic map with points of interest and then upload that map to an online service which was then accessed by an android mobile application. The quality of this work is what is to be expected from a proof of concept, its sole purpose has illustrated that the development of the main feature of the system using current technology and skillsets can take place.	Finished

Activity flows and use case diagrams with descriptions[4]	Producing activity flows and use case diagrams allowed the processes of the system, without getting involved with any of the technicalities of the design of the software, to be shown. They have also contributed to the on-going refinement of the requirements documentation. These documents will most likely go through some changes if any extra research takes place.	Initial designs complete/ Ongoing
Detailed Stage 3 and 4 plans[5]	At the end of each stage a detailed plan of the next stage has been developed. As the first and second stages are defined in the PID, the third and fourth have been defined in two separate documents.	Finished
GUI designs and descriptions[6]	These designs are crucial in displaying to the stakeholders of the project as well as a reference for future development. As these are so crucial a good level of detail has been provided with each design with descriptions about each element displayed. This document will most likely go through some changes if any extra research takes place.	Initial designs complete/ Ongoing
Requirements list[7]	This is a list that has been adapted from the original in the PID. It reflects the most up-to-date changes including priorities and difficulties for each requirement.	Finished

4. Risks that have materialised and my response; changes to the risk list

Up to this point only one risk has arisen and that has been overrunning on a schedule on the development of the proof of concept application, the response was to allocate some extra time out of the normal schedule to deal with the overrun, this worked well and allowed development of the proof of concept without impacting the project. The next phase of the project, the development, will have a higher probability of a risk arising extra attention to the risk list from the PID will be given.

5. Schedule

On average there has been between fourteen and seventeen hours a week work for this project. This has allowed effective work up to this point to produce the products listed in section three. Use of some free time at the weekend will act as a 'buffer' for making up any hours that need to be made up.

In the Initial project plan Stages 4, 5 and 6 have been devoted to the development of the system. Stage 4 indicated that a Prototype system will be developed. With the research undertaken at the start of the project I have updated stage 4 to focus primarily on the research and development of the underlying database that will store the information needed to authenticate festival organisers and hold festival data, without this I will not be able to continue to develop the rest of the system, it is key to make this a priority throughout this stage. Any time left after the database development will be given to preparing for development of the rest of the system in stages 5 and 6.

6. Resources

Up to this point, making use of different resources to produce the products outlined in section 3 has worked well to get the products finished with a high level of quality.

The majority of the documents have been created with the use of Microsoft Office; Word has been used to document research, analysis and diagrams (Activity flows and Use cases) have been created via Visio. A program called Pencil has been used to develop the set of detailed GUI designs.

For development of the proof of concept application Netbeans has been utilised for the Festival organiser application and the online access system. For the mobile application the Android development toolkit in Eclipse has been used. In both instances Java has been the main programming language.

A big resource that has been a great advantage has been the fortnightly meetings with the project supervisor throughout the duration of the project, these have been extremely useful in getting a thought process about general organisation rolling and receiving valuable feedback from produced material.[7]

Research will be undertaken in the next stage to determine an appropriate resource for the proposed database.

7. Student learning undertaken & required

To develop the proof of concept application I had to enhance my knowledge on using Java to display layered items on top of a base image. On top of this I also had to refresh my experience in dealing with JSON data (I will be using JSON to transmit information from the Online access system to the mobile devices). I expect that as the development starts and continues it will be a constant

To develop the documentation already produced I relied on my knowledge from the computing degree I am currently undertaking as basic fundamentals taught throughout my university education have been heavily focused towards effective design and planning.

8. Evaluation of progress

So far the project has not thrown up any strenuous issues that I have not been able to cope with. The products produced so far have all been to schedule, apart from the proof of concept application. Sticking to my planned schedule and making use of the resources available to me has been the main reason for me staying on track. Jumping in to creating a proof of concept application gave me a valuable time to practice my development procedures as well as furthering my learning. If I continue down this path I see no reason why I shouldn't be able to meet the business and project objectives set out in the PID.

E. Second Interim report

1. Introduction

The following Interim report contains an update on the progress of Festival viewer project since the first interim report by comparing what was planned to be produced and what has actually been produced along with reasons why.

The fourth (as outlined in the changes to scheduling in the first interim report) of the project has been undertaken and completed with useful outcomes. The fifth stage is currently underway and making good progress, regular meetings with the supervisor have also continued, see appendix A for meeting records.

2. Tasks undertaken and outcomes

The table below goes into detail about each task that have been undertaken up to this point and if they deviated from the original plan[1] and why.

Task	Description/Outcome	Current Status
Research into database/storage systems	This is a new task that arose from the High level designs that it would be easier to develop a small database used for authenticating the festival organisers on the system. This research is currently on going and will highlight why this is required, what resources should be allocated for this and table designs.	Finished
Development of Database and JSON Web service	Creating a suitable database and RESTful web service will allow a major backbone of the service that will provide functionality to the Desktop applications and the mobile application. This has to be finished before any real further development can take place.	Finished
Investigation into authentication of the JSON Web service	As having the product working in the 'cloud' it can be susceptible to attack, research is needed into an effective way to protect the database via the JSON web service.	On Going
Development of desktop applications	When the development of the JSON Web service with database is finished the development of the desktop applications (The system admin application and the Festival creator application) can start.	On Going

Write up Table of contents for the final report and write a draft Chapter for the final report	To get a grasp of the final report and to get feedback of the project a Table of contents and Draft chapter should be drafted. (See the attached items with this submitted report)	Finished

3. Products produced and product quality

Product	**Description and Level of quality**	**Current Status**
Database/Storage design document	A brief design document detailing the structure of the underlying database deployed to Amazon RDS and a quick write up about the RESTful service created in Netbeans, see appendix B for the design document.	Finished
Database	A MySQL database that holds the information regarding the festival data, mobile users and festival organisers deployed and currently live on the Amazon RDS service.	Finsihed
JSON Web service	A RESTful web service created via Netbeans that links to the database online.	Finished
Detailed Stage 5 plan	At the end of each stage a detailed plan of the next stage has been developed. As the first and second stages are defined in the PID, the third and fourth have been defined in two separate documents. See appendix C for the stage 5 plan	Finished

4. Risks that have materialised and my response

Since the report of the first interim there have been no major or notable risks that have developed.

5. Student learning undertaken & required

To develop the JSON restful Web service along with its underlying database I had to enhance my knowledge on using Java to develop the RESTful service. On top of this I also had to refresh my experience in dealing with JSON data as this is how the applications will receive data to and from the database.

As the RESTful web service and Database needed to be hosted on the cloud I decided to make use of the Amazon AWS services. They offer full database support (RDS), online storage (S3) and remote servers (EC2). This was a good choice as all of these services could be launched for free on a trial basis in minutes. Further setup and usage needed a bit more education, fortunately there are plenty of online tutorials and documentation to take advantage of.

To develop the documentation already produced I relied on my knowledge from the computing degree I am currently undertaking as basic fundamentals taught throughout my university education have been heavily focused towards effective design and planning.

6. Evaluation of progress

So far the project has not thrown up any strenuous issues that I have not been able to cope with. The products produced so far have all been to schedule. Sticking to my planned schedule and making use of the resources available to me has been the main reason for me staying on track. Creating a suitable database and RESTful web service has allowed a major backbone of the service that will provide functionality to the Desktop applications and the mobile application.

F. Supervisory Meeting records

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 07/10/14

Items Reported
Delivery of draft PID (To be discussed below)

Topics discussed
• My idea for my final year project. • Initial draft version of PID. • Specific ideas for programming languages to use for the project. • How specific the requirements should be. • Some discussion of ideas for requirements. • Discussion of implementation of either a standalone application or web application for the tool to set up a festival. • My own personal time management.

Action Plan	By whom?	By when?
• Refine and develop essential and optional requirements as well as finishing the PID.	-----	15/10/14 (Latest)
• Develop a proof of concept application of my idea.	-----	ASAP after PID
• Re-read and understand documentation from DLE	-----	ASAP

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 21/10/14

Items Reported
Draft research into similar applications Start of a proof of concept using google maps API

Topics discussed
Concern rose over how the map is going to be created on the festival organiser's side of the system, is it a map created within the application or is it a map uploaded and then given extra information? Assumption needs to be taken to proceed with a clear plan. The Draft research into similar applications needs to be tidied up and concluded to be of any use. Pushing on with a proof of concept was also discussed, once a clear assumption has been made about how the map will be developed and development environments then the proof of concept can be delivered.

Action Plan	By when?
• Tidy up and conclude research into festivals and similar applications. • Decide on assumption from research of similar systems and how festivals create a map for the best way to deploy a map in the desktop/web application. • Develop a proof of concept system of deploying a map and reading it on a mobile device.	ASAP ASAP After research has finished By next meeting, (04/11)

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 04/11/14

Items Reported
Proof of concept application finished Documentation of decisions for final feature list based on research Stage 3 plan Modified feature list

Topics discussed
Discussion of how the proof of what features the proof of concept application had been developed as well as a high level description of how that was achieved. The discussion of the POC also prompted some discussion regarding the level of detail on what should be shown as the user zooms out of the map on the mobile device. The stage three plan was also discussed, the 'architectural diagrams' part needs to be more precise to show what is actually being produced. The given feature list should be given a priority on each item as well as the difficulty associated with that. An optional 'nice-to-have' feature list needs to be drafted.

Action Plan	By whom?	By when?
• Stage 3 plan to be more precise	-----	ASAP
• Update feature list to be more precise in regards to priority and difficulty	-----	Before next meeting

• Draft an optional feature list • Execute Stage 3 plan	----- -----	Before next meeting ASAP

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 18/11/14

Items Reported
• Updated feature list • Gui Styling • UML Diagrams • Playing around with Notifications results • Stage 4 plan

Topics discussed
The feature list should say in which direction the priority should be. Looked at example maps online to notice that 3rd party stalls are never really shown. Only areas where they are all located. This should be reflected in the feature list. GUI elements discussed, most notably: • Positioning of the icons on the map x, y coordinates pointless, investigate • How does the user Save changes/ Save as/ Upload. What is the order, process of it all? Are there any more scenarios that could crop up?

Action Plan	By whom?	By when?
Detailed write up of GUI screen grabs.	-----	Next meeting
Tidy up all documentation to be added to 1st Interim report.	-----	Next meeting
	-----	
1st Interim report	-----	Next meeting

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 02/12/14

Items Reported

- First Interim report

Topics discussed

This meeting primarily focussed on the First Interim report.
Went through it together and paid most of the attention on the appendix as this demonstrated the work I had done so far on the project and showed that the report was good enough quality to be submitted.

It was also discussed that I should have in mind a milestone I wished to achieve before I finished for the Christmas holidays to make sure that upon returning to working on the project I would know where to start from. I decided that it would be a good idea to design a small data structure (for the database that holds info regarding users and festival data) so I could implement it straight away after the holidays.

Action Plan	By whom?	By when?
Submit Interim report	-----	03/12/14
Write up a small data structure	-----	Before Xmas hols

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 28/01/15

Items Reported

- Draft Table of contents
- Progress over the holidays

Topics discussed

The table of contents needs to be more detailed and have more relevance with explanations as to what goes into each chapter and sub chapter.

A draft chapter is also needed along with the second interim report.

An demonstration of using Outline in Word was given by the supervisor to show how to create a table of contents in a productive way.

Work on the project that took place over the holidays was also discussed; most notably the development of the underlying database and web service using Amazon AWS services was discussed.

Action Plan	By whom?	By when?
Re work Table of contents	-----	04/02/15
Write Draft Chapter	-----	04/02/15
Submit 2 Interim report		04/02/15

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 12/02/15

Items Reported

- Updated table of contents

Topics discussed

Discussion based around the updated table of contents.

- No numbers needed on anything outside the main write up (i.e. appendix etc...)
- The requirements topic should be moved into the Method of approach topic, research should be before the method of approach.
- Need to accurately define what is to go into the objectives, business and project goals. (Are they all needed?)
- Contents is a bit short on length, would equate to about 2000 words per section at the moment.
- More detail needed, i.e. what is the API, Amazon web services, SDKs being used, development methodologies, why REST, MySQL and Amazon?

Discussion also on the Sample chapter.

- Sample chapter was the Method of approach section.
- Update to updated contents (when it gets updated).
- Quite light on content, tables should also be in the appendix.
- Use references to back up claims

Action Plan	By whom?	By when?
Re work Table of contents	-----	Next meeting
Re work draft chapter	-----	Meeting after next.

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 25/02/15

Items Reported

- Been focussing on development rather than any kind of write up.
- Good progress made with the development

Topics discussed

- The demo needs to be arranged with Martin and Chris. Send out an email to organise
- Reminder of the poster showcase, 27th April
- Push forward with the contents and Chapter

Action Plan	By whom?	By when?
Get the table of contents finished	-----	Next meeting

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 18/03/15

Items Reported
• Re-defined Table of contents

Topics discussed
• Table of contents : • Location of different sections • Merging technologies research into one bigger topic. • Test plans • Future development • Marking Criteria • Sample Chapter

Action Plan	By whom?	By when?
Get the sample chapter finished	-----	Next meeting
Update table of contents with recommendations.	-----	

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

PRCO303 Computing Project – Supervisor Meeting Record

Student Kevin Borrill **Supervisor** Martin Beck **Date** 25/02/15

Items Reported
• Draft Chapter

Topics discussed
• Draft chapter: • Be aware that the report is only 10k max words. Draft chapter too large for Method of Approach. Look for repetition when cutting back. • Mention the use of PRINCE2 • Move back the numbering. Goes in too deep.

Action Plan	By whom?	By when?
Final work towards development and Project write up.	----- -----	

These forms should be used contemporaneously to record brief details of the substantive meetings with your supervisor, and aggregated to form an appendix of the final report.

G. Similar Applications research

The following analysis gave useful input in identifying and refining the feature list of the proposed system as well as identifying any other potential users and markets that the product could be aimed towards. Below is the analysis of the similar mobile applications that were researched in the music festival market.

Glastonbury App by EE

Figure 8 Screen captures from the Glastonbury application.

The Glastonbury application (Figure 8) (currently unavailable to download), developed by EE is the most similar to the mobile application side of the system that has been proposed. With news, an interactive map and a schedule list for its key features shows that the big festivals are genuinely noticing the same need for a smarter solution to keeping festival goers updated that the proposed system has identified.

Bunbury Music Festival, Ohio.

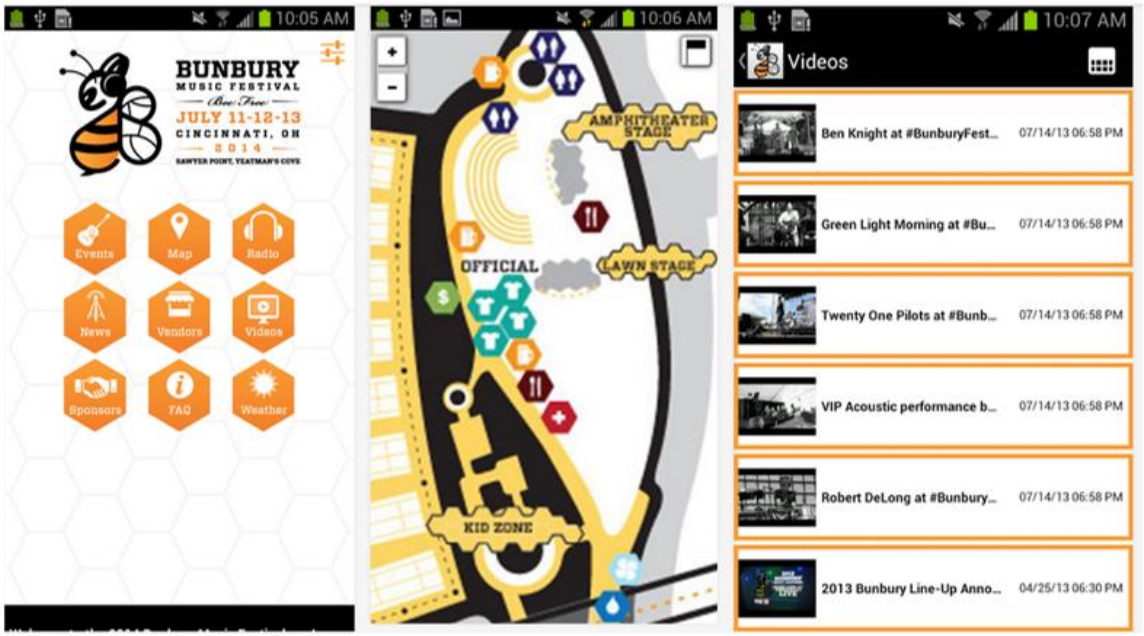


Figure 9 Screen captures for the Bunbury Festival application

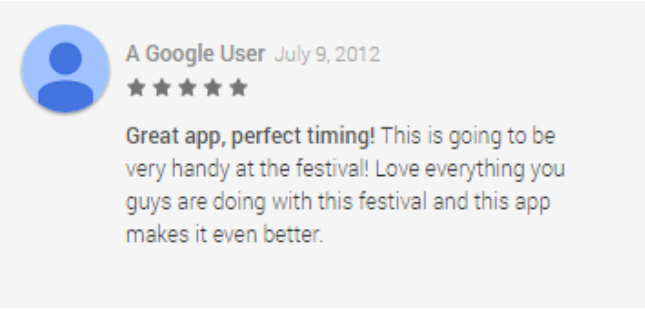


Figure 10 A positive review from the google play store for the Bunbury Festival application

With positive reviews online (Figure 10) and simple information such as a scalable map and an events list the Bunbury music festival application, Figure 9, is a simple yet effective application to provide users with some basic details about the festival, again this is another one off application showing that festival organisers want to retrieve details out to the festival goers to a mobile device. This application also favours using a static, pre-existing map instead of a completely re-designed map like the Glastonbury application.

Festival ready mobile application

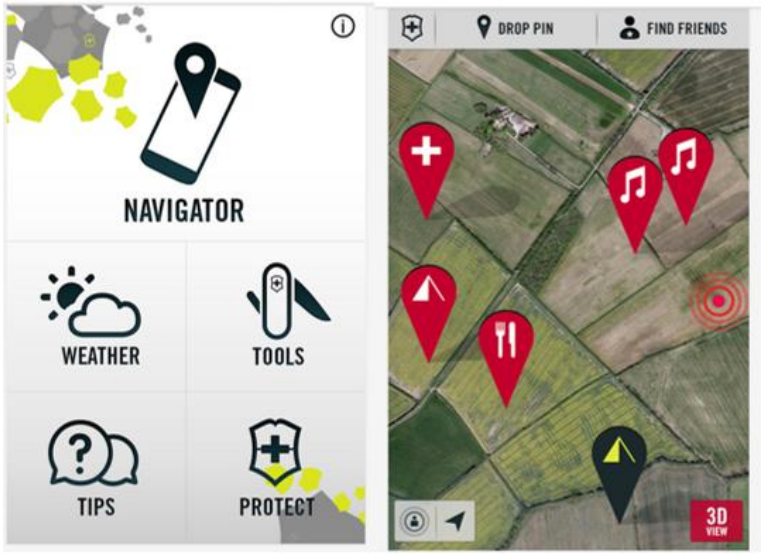


Figure 11 Screen captures from the festival ready application

Festival Ready, Figure 11, available on the google play store, is more of a utilities application aimed for the festival market. Using an inbuilt map, users can 'build' their own map to help them navigate around a festival. As well as this, the application features simple tools such as a weather forecaster. The simplistic design of the user interface in the application makes it very easy to use and understand making it stand out as an attractive and finished mobile app.

Standard maps created by festivals, usually for flyers/websites

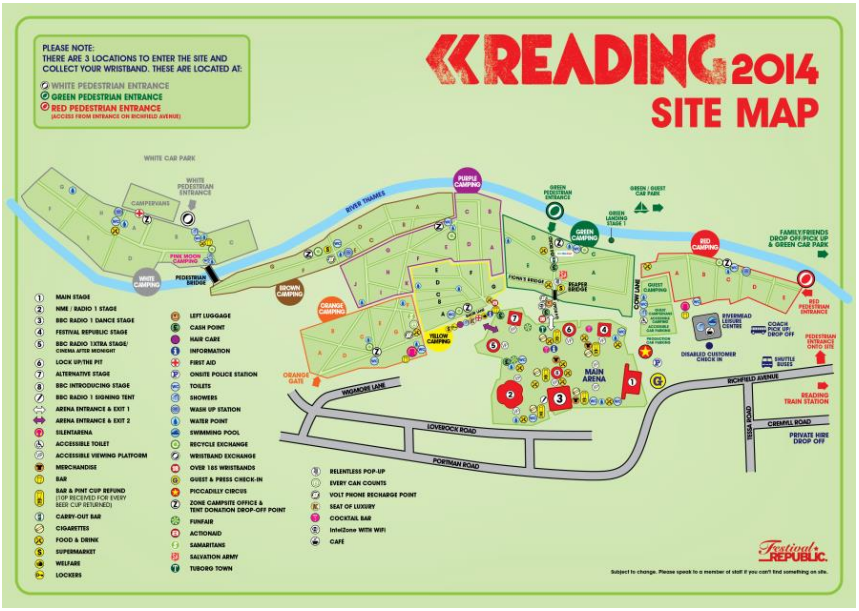
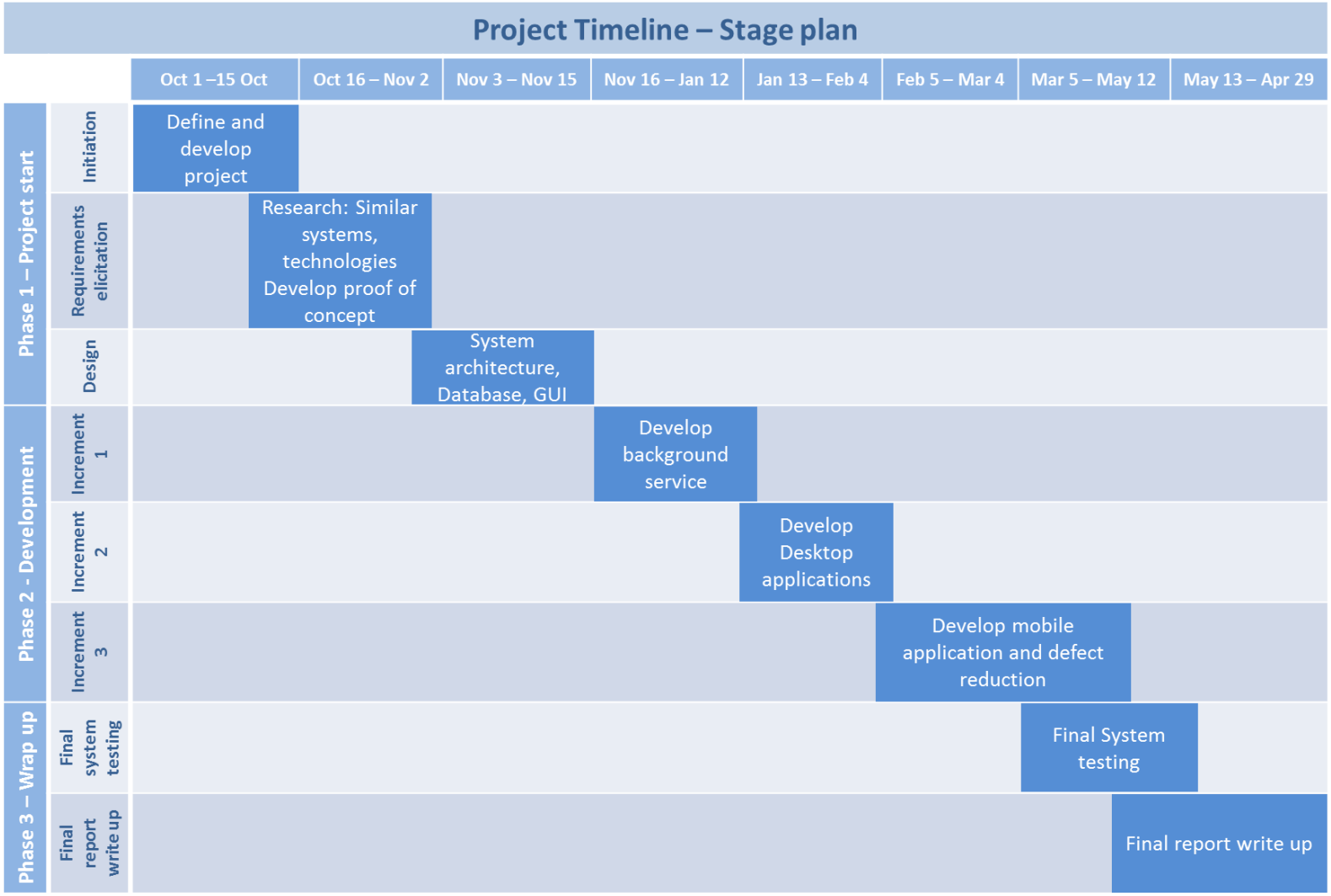


Figure 12 The Reading music festival 2014 Site map

As well as smart solutions it is also important not to ignore what a lot of festivals currently produce, which is the standard printable maps usually seen on flyers.

The Reading 2014 map, Figure 12, is a good example of how organisers like to display the festival grounds to the festival goers. One notable flaw with these is the big list on the left of the map for the key to what the iconography means. One piece of information that is quite useful is the boundaries and the walk ways/roads displayed, this could potentially be used as an interactive map on a mobile device if the image was loaded into a browser.

H. Project Timeline



I. Initial Risks List

Risk	Probability	Impact	Action (Contingency)
Overrunning on schedule	Medium	Medium	Allow at least a week before deadlines to finish each deliverable; this gives an overflow to achieve the work that has not been completed.
Difficulty learning new technologies	Medium	High	Create a proof of concept system to test not that not only the technology works but it can be developed with it.
Loss of work through corruption or deletion.	Low	High	Multiple backups of every piece of work that is undertaken will be created weekly. This will allow restoration of the project and only be a week behind at the most.
A required feature is too ambitious to be created owing to time or technology constraints.	Low	Medium	An analysis of requirements will be undertaken at the start and throughout the project, if one is too ambitious other features will still make the project successful.
Change in requirements	Low	High	If any changes of requirements are needed then a full analysis of how to fit in the new/changed features should be undertaken. Risks should also be re-assessed at this stage.

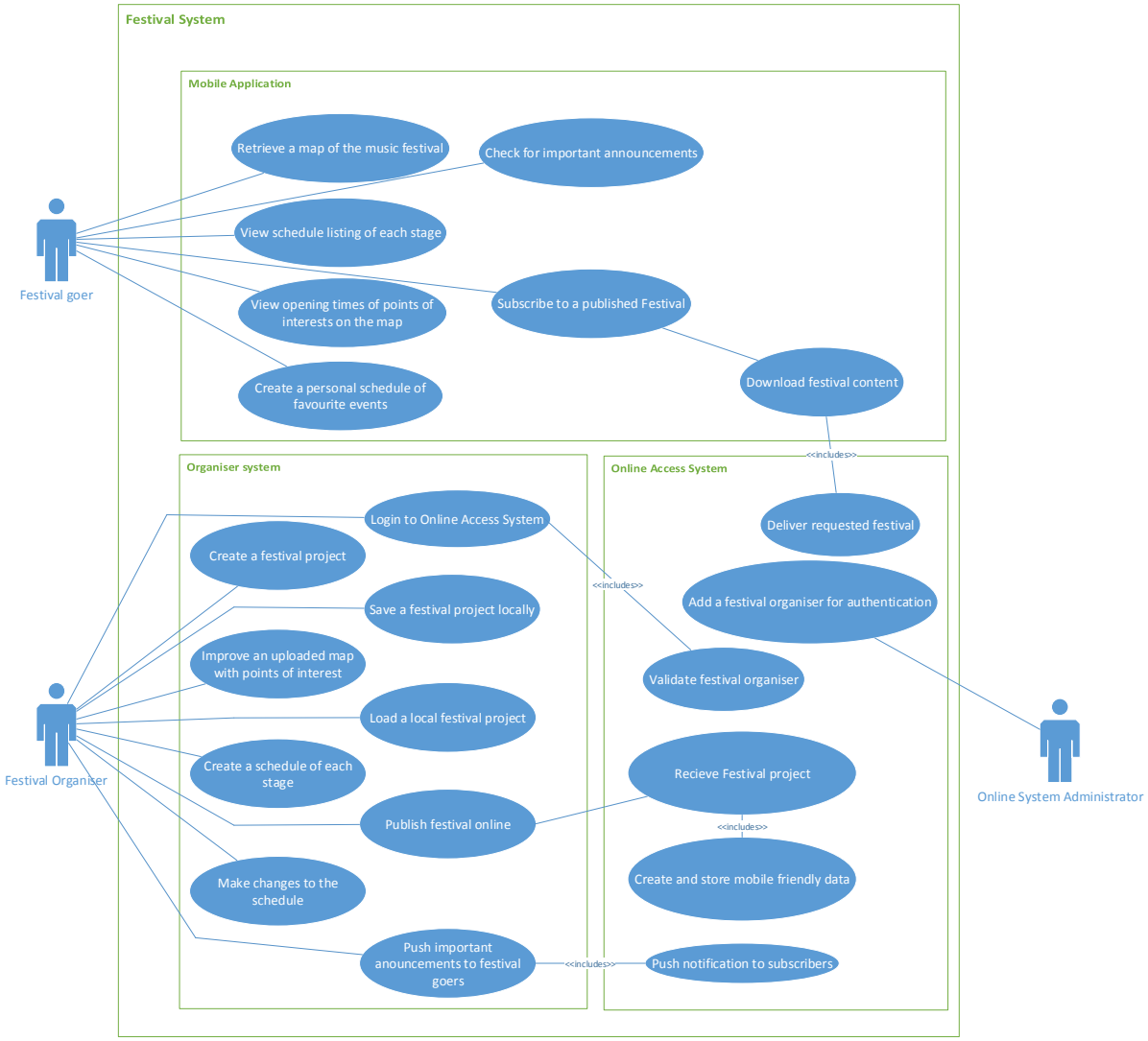
J. Amazon AWS pricing table

Product	Free tier (12 Month) Usage	Beyond free tier (USD)
EC2	**750 hours** per month of Linux, RHEL, or SLES t2.micro instance usage **750 hours** per month of Windows t2.micro instance usage Run one instance at a time or multiple instances simultaneously	(Based on a windows instance) **t2.micro** $0.018 per Hour **t2.small** $0.036 per Hour **t2.medium** $0.072 per Hour Note: Other sizes and prices are available: http://aws.amazon.com/ec2/pricing/
S3	**5 GB** of Standard Storage **20,000** Get Requests **2,000** Put Requests	**First 1 TB / month** $0.0300 per GB **Next 49 TB / month** $0.0295 per GB **Next 450 TB / month** $0.0290 per GB Note: Other sizes and prices are available: http://aws.amazon.com/s3/pricing/
RDS	**750** Hours per month of single-AZ Micro DB Instance usage **20 GB** of DB Storage: any combination of General Purpose (SSD) or Magnetic **20 GB** for Backups (with RDS Magnetic storage; I/Os on General Purpose [SSD] are not seperately billed) **10,000,000 I/Os**	**db.t2.micro** $0.018 per Hour **db.t2.small** $0.036 per Hour **db.t2.medium** $0.073 per Hour **db.m3.medium** $0.095 per Hour Note: Other sizes and prices are available: http://aws.amazon.com/rds/pricing/

K. Microsoft Azure Pricing Table

Product	Pricing (USD) Basic Tiers
Virtual Machine (including storage)	**A0:** 1 Core, 0.75 GB Ram, 20 GB Disk, $0.018/hr **A1:** 1 Core, 1.75 GB Ram, 40 GB Disk, $0.077/hr **A2:** 2 Core, 3.5 GB Ram, 60 GB Disk, $0.154/hr **A3:** 4 Core, 7 GB Ram, 120 GB Disk, $0.308/hr **A4:** 8 Core, 14 GB Ram, 240 GB Disk, $0.616/hr Note: Other sizes and prices are available: http://azure.microsoft.com/en-us/pricing/
SQL Database	5 Throughput units, 2 GB Database storage, 7 Days point in restore time, $0.0067/hr Note: Other sizes and prices are available: http://azure.microsoft.com/en-us/pricing/

L. Use Case Diagram



M. Use Case Descriptions

To properly understand the use case diagram, example use case descriptions can show the processes involved with actions from the actors and the responses from the system.

ACTOR: **FESTIVAL ORGANISER**

Name: Create a festival map.
Actor: Festival organiser
Description: A festival to be created as part of the project for a festival by the Festival organiser.
Preconditions: The festival organiser must have a basic map of the festival in an image or pdf format.
Post conditions: A map of the festival grounds with points of interest and stage icons added on top.

Actor action	System Response
Select the map designer from the menu	
	Display the map designer form
Select an image/pdf file of the grounds/layout of the festival	
	Image/PDF displayed on a pannable frame
Select to add a Point of interest/Stage on top of the map	
	Menu displayed to add a description and an icon
Repeat adding points/stages until finished with design.	
The administrator saves the project	
	Map information successfully stored locally

Name: Create a stage schedule	
Actor: Festival organiser	
Description: Create a schedule for the stages on the map	
Preconditions: None	
Post conditions: A schedule for a stage will have been created	
Actor action	**System Response**
Select the Schedule designer from the menu	
	Display the Schedule designer form
Select/Or create a Stage	
	Blank schedule for stage shown
Add performances to the schedule with a date and time	
Save the schedule	
	Schedule saved to the project.

Name: Log in to Desktop Festival Organiser System	
Actor: Festival Organiser	
Description: Validate authorisation with the Online access system	
Preconditions: None	
Post conditions: Festival organiser can upload/make changes to a festival	
Actor action	**System Response**
Select the login function from the main menu	
	Check online access and load login menu
Fill in login details	
	Wait for authorisation from the Online access system

Name: Create alert to push to Festival goers	
Actor: Festival Organiser	
Description: Create an alert that is deemed necessary to push straight to the festival goers' notification panes	
Preconditions: Festival Goers must be 'Subscribed' to the festival	
Post conditions: A notification is pushed to the festival goers mobiles	
Actor action	**System Response**
Select create notification	
	Show notification panel
Type a text notification and hit send	
	Send to Online access system
	Online Access system pushes to subscribed devices

ACTOR: **ONLINE SYSTEM ADMINISTRATOR**

Name: Create a Festival organiser user	
Actor: Online System Administrator	
Description: Create a festival organiser so that they can log on and create a festival on the Festival Organiser system	
Preconditions: Festival Organiser has paid a subscription/fee	
Post conditions: Festival Organiser has authorisation to use the system and upload content	
Actor action	**System Response**
Select add Festival organiser from the main menu of the Online access system	
	Loads form to create a new user
Fill in username and generate a password and hit create	
	Checks if username exists if not returns successful, if it does returns error.

ACTOR: FESTIVAL GOER

Name: Register to a festival	
Actor: Festival Goer	
Description: Festival goer registers to a festival so they can view its content and receive notifications	
Preconditions: None	
Post conditions: Festival goer can now have access to the mapping and scheduling of the festival as well as receiving important announcements	
Actor action	**System Response**
Select 'find a festival' from main menu	
	Festival list downloaded from online access system and presented to user.
Select to 'Subscribe/show/download' festival	
	Phone added to list of subscribers.

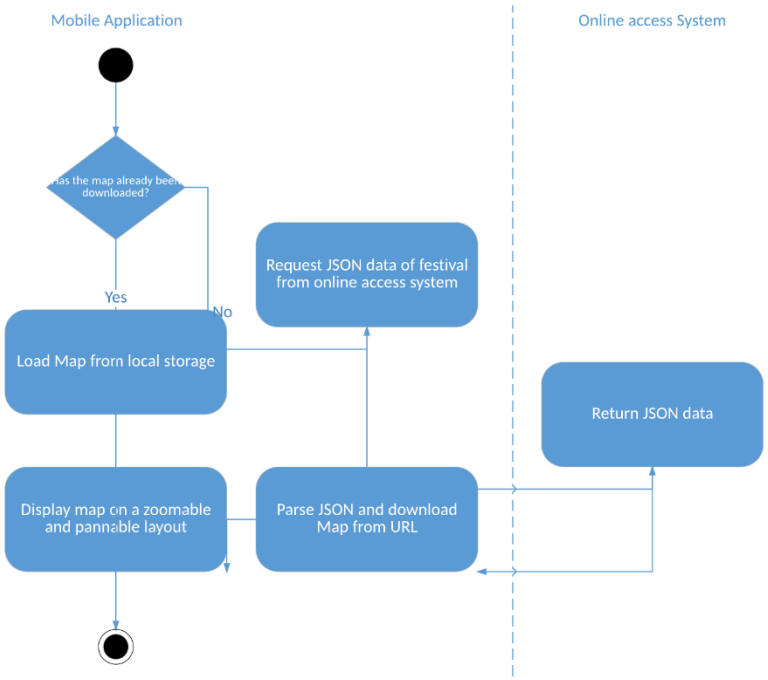
Name: Find a stage and check out the schedule for a particular day	
Actor: Festival Goer	
Description: A festival goer may wish to find out what acts are performing on a particular stage on a particular day	
Preconditions: Is subscribed to the festival	
Post conditions: User is aware of the schedule	
Actor action	**System Response**
Select 'Map' from the festival menu	
	Display map from downloaded content
Browse to a stage by panning/zooming to its location and touching the stage.	
	Load a pop up with the schedule information (Each day has a separate list, viewable by 'swiping through' each day)
Swipe to a particular day and read information	

Name: Add performance to personal schedule	
Actor: Festival Goer	
Description: The Festival goer may wish to create a personal schedule of their own choosing and reminders of what they would like to see.	
Preconditions: None	
Post conditions: A performance is added to personal schedule	
Actor action	**System Response**
Select a schedule for a stage via the map or via the schedule list menu.	
	Load the requested schedule.
Select the 'add' button next to a performance.	
	Add schedule to list if exists or create one if it doesn't.

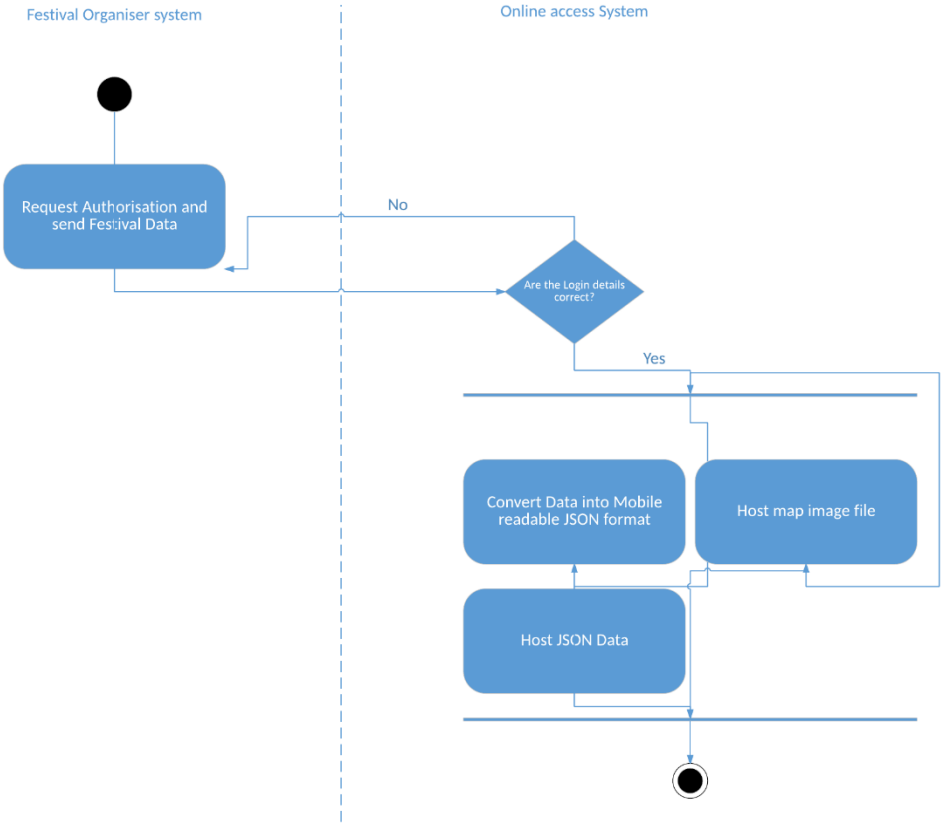
Name: Read an important announcement from a festival	
Actor: Festival goer	
Description: The Festival goer may wish to read an announcement about the festival	
Preconditions: Notification of new news pushed to festival goers mobile device	
Post conditions: User aware of new information	
Actor action	**System Response**
Pulls down notification panel and selects news item from list.	
	Display news item
Read news Item	

N. Activity Flow/Sequence Diagrams

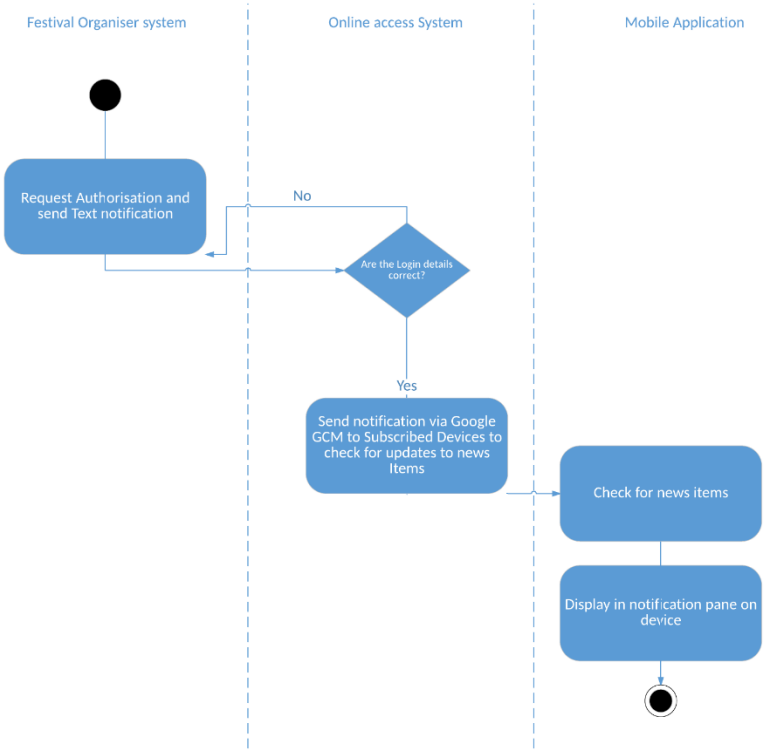
Viewing a festival map on a mobile device



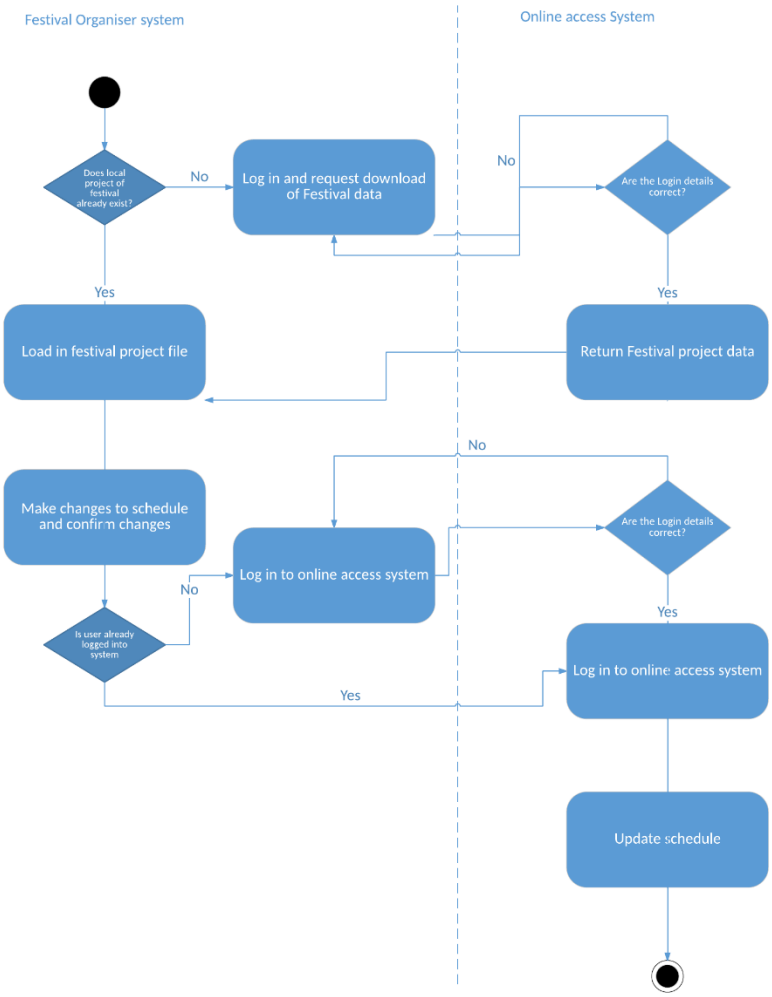
Uploading a festival to the Online access system



Notify Mobile Users of an Important news Item

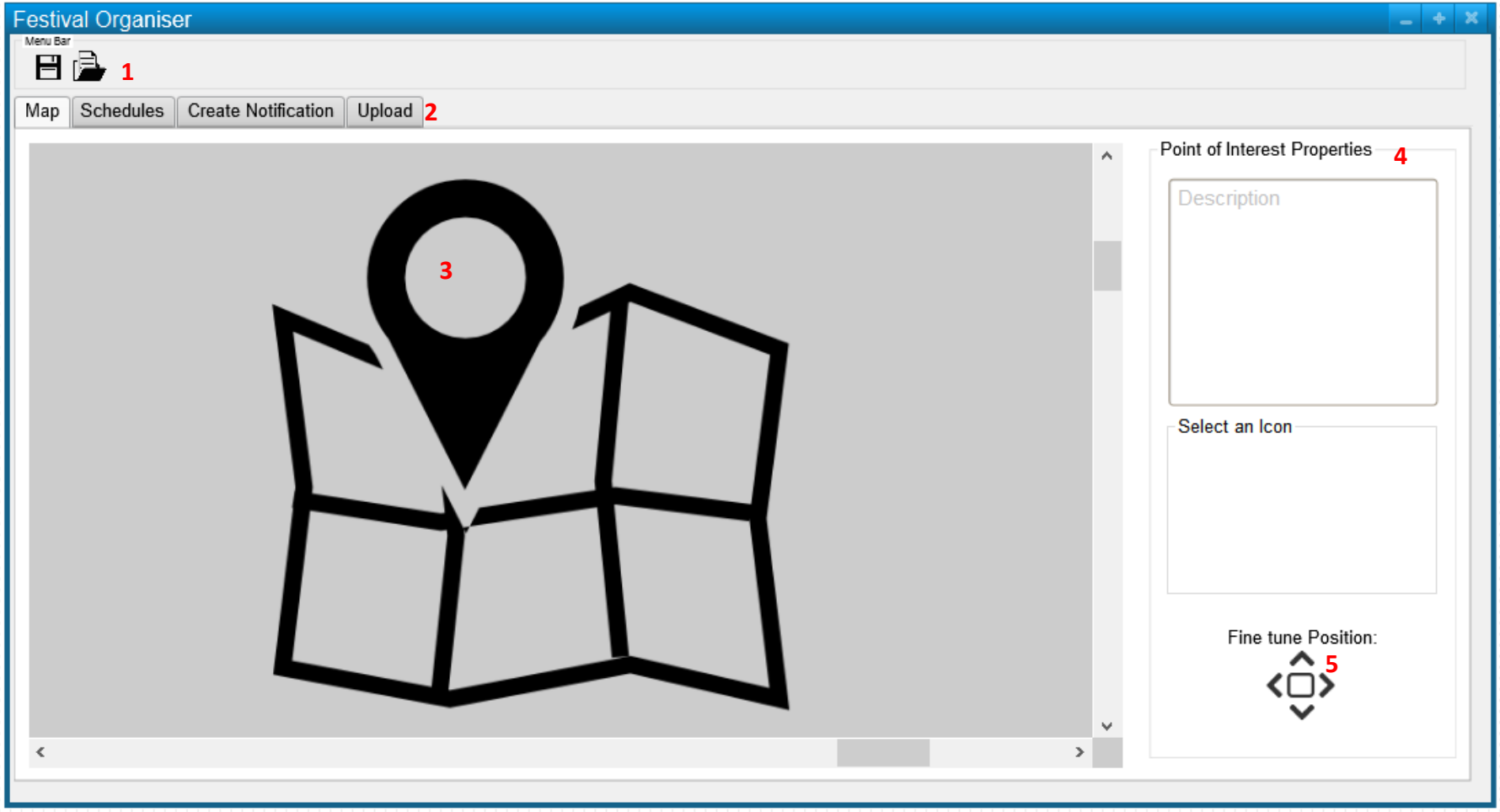


Editing a Schedule for a festival that is already online



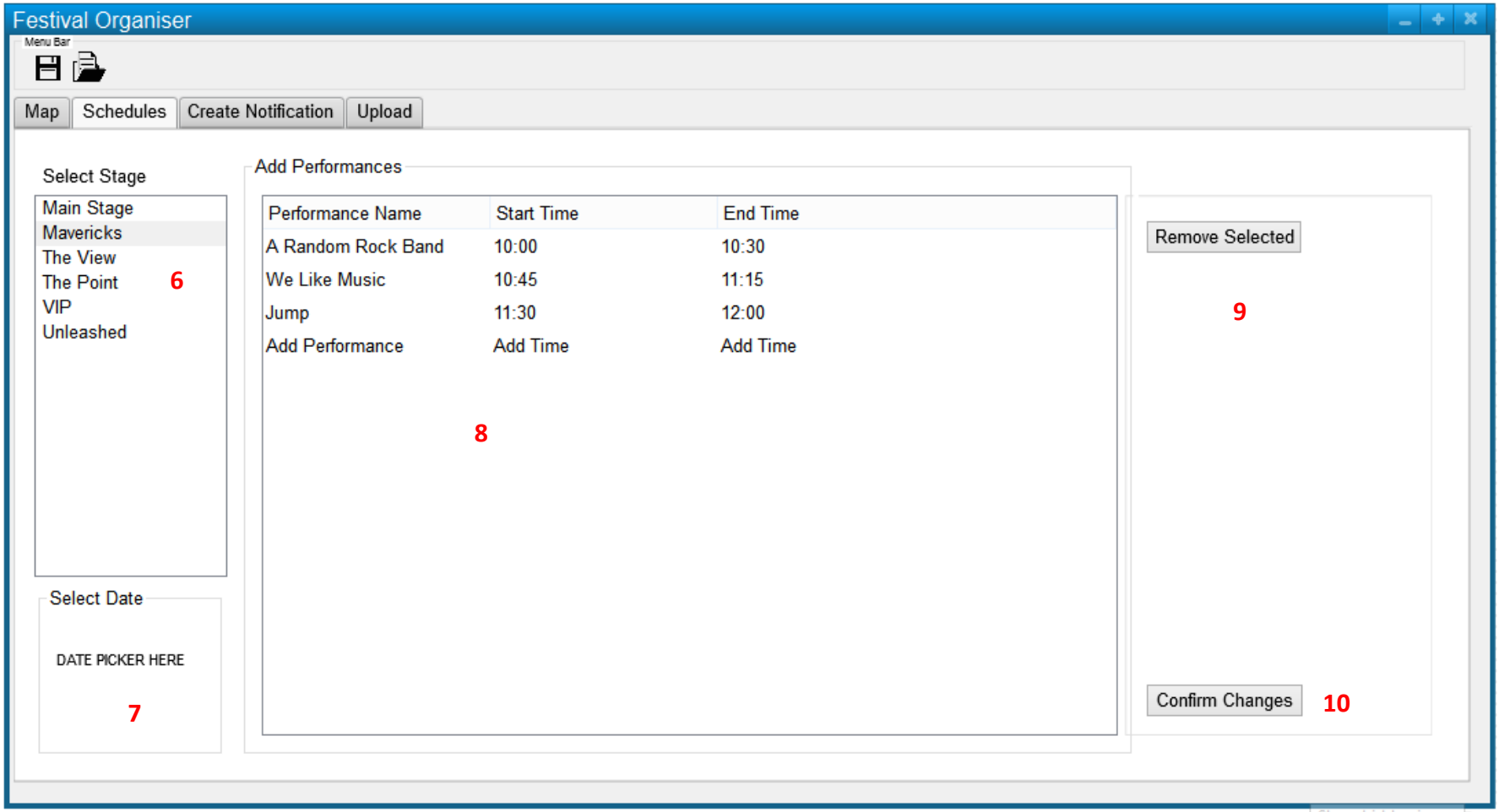
O. GUI Designs

Festival Organiser



This window represents the festival organiser application with a tabbed view layout, currently selected is the map creator/editor tab where the user can select a base layer map (image or pdf file) and then go on to add points of interests and stages . Below is the description of each element in this window.

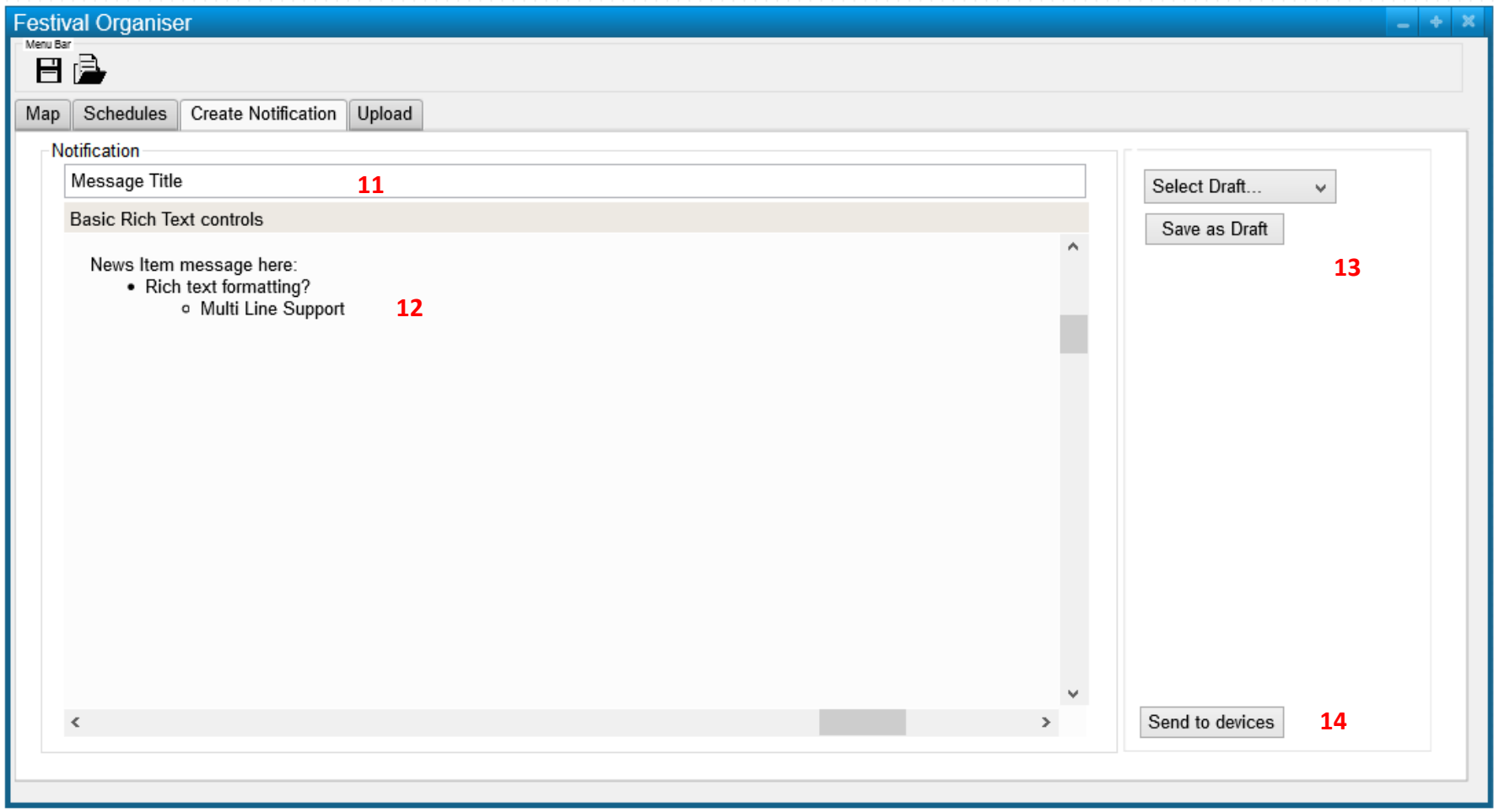
1. This is the main menu bar where the following buttons will be placed:
    - New project
    - Save project
    - Save project As…
    - Load project
    - Undo last action
    - Redo last action
2. This is the tab bar where the user has the options to edit different parts of the project.
3. This is the main mapping area. Here a base layer image/pdf will be ‘uploaded’ into the application where the user can then pan and zoom the image and add a new point of interest (or stage) to the right click location. The properties of the point can then be changed on the section to the right (number 4). Any point of interest will also be clickable to change the properties and dragable to move it to a new location.
4. The properties menu for a point of interest/stage allows the user to define a description of the point (viewable in the mobile application), choose an icon and select an icon.
5. To fine tune the position of the point (i.e. it’s not exactly in the right location) the user can ‘nudge’ the point up, down, left and right by clicking on the respective arrow button.



This tab of the Festival organiser represents the area where the user can create schedules for each stage.

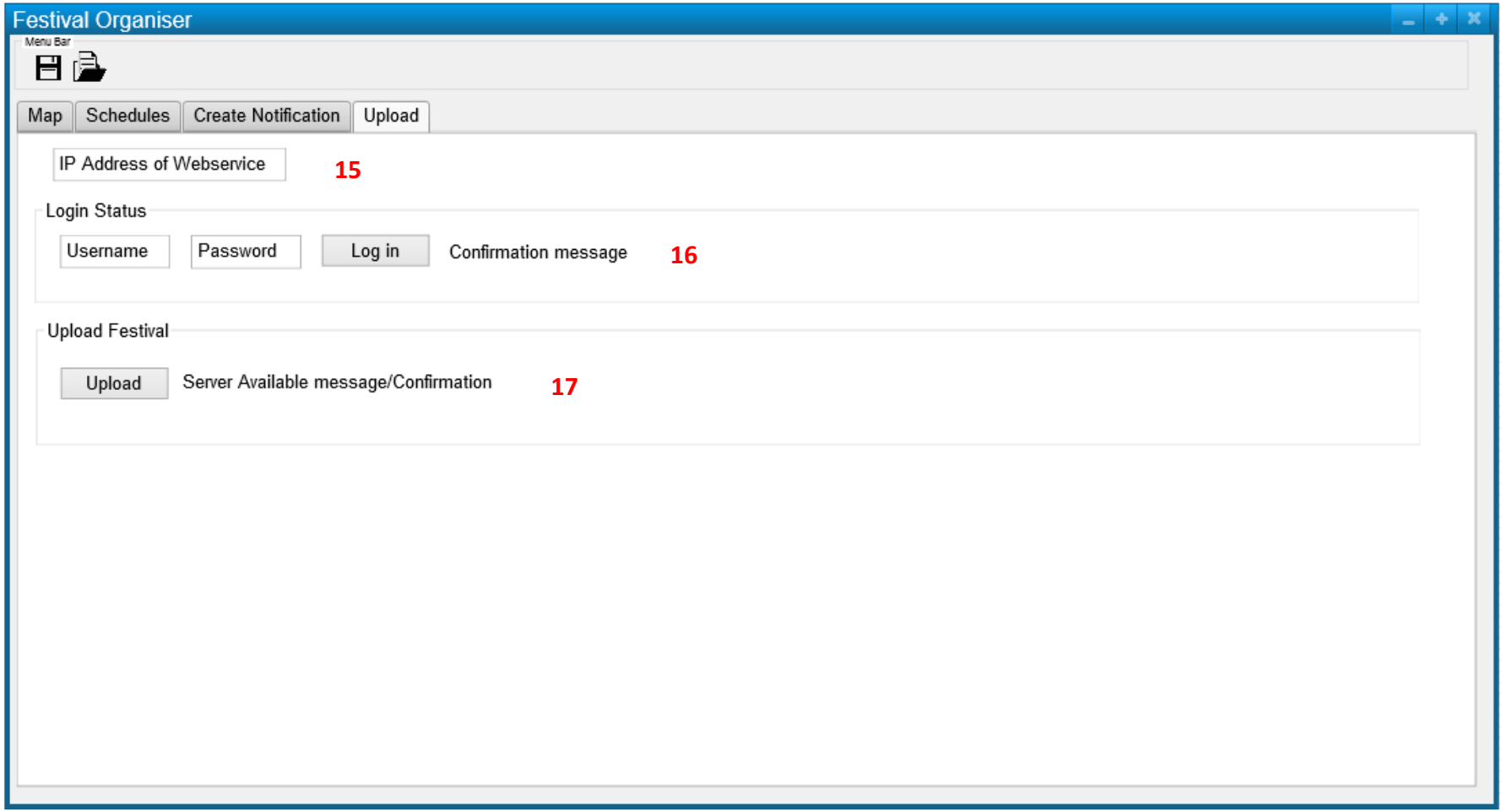
6. This is the area where the stage is selected, a date can then be selected (7) and then performances added/changed (8). Note: a stage is created on the previous tab in the mapping menu, this should be made aware via a label in this area.
7. This is the Date picker for the day of the schedule to be created/changed for the selected stage (6).

8. This is where acts can be added and edited. The acts are automatically arranged via the time they are on. Any conflicts of performances are highlighted. Adding a new performance is quite straight forward, the user enters the information for the act on the bottom row of the table. Any performance that needs inforamtion changing is done via changing the respective cell/s for the performance by selecting and changing the value.
9. This Remove selected button does what it says, removes the selected performance from the table. A dialog should pop up to confirm any deletion. The user may select more than one row at a time to delete multiple performances.
10. To confirm the changes the user will select the Confirm changes button. If the festival is already live (uploaded to the online access system) then a dialog will display reminding the user these changes should be pushed online, the user can then choose to do so from that dialog. If they do not a warning icon should appear on the main window to alert the user that changes have not been uploaded.



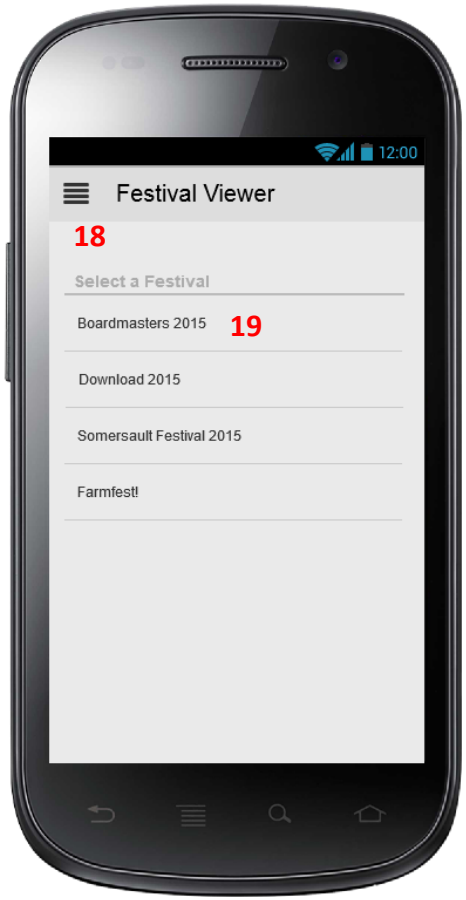
This is the create notification tab of the Festival organiser. This is where the user will create a news item to send to the devices that have registered to the uploaded festival (note: This tab will only be active if the user is logged in to the online access system as a festival organiser, a log in dialog should be present if the user is not. A festival created by the festival organiser must also be live and running on the online access system).

11. This is the text box that the user enters the title of the notification/news item.
12. This is the area for the main content of the news item. The description should relate to the title. Rich Text support is given to allow easy formatting of the text (colour, font size and font formatting).
13. This is where the user can choose to save the news item as a draft to come back to it at a later time. This will also save the main festival project.
14. This is where the news item will be sent to the registered devices. If the user is not already logged in to the online access system the application will prompt the user to then log in. As well as being logged in the festival should also be up and running on the online access system.



This final tab of the festival organiser application shows the area where the user can upload/re-upload a festival to the online access system.

15. This is the IP address of the location of the Online access system. By default this field will be filled out automatically. The user should only change this if prompted by the owner of the online access system.
16. This is where the user will login and authenticate themselves with the online access system. A confirmation/error message will be presented after attempting to log in. If an error occurs a suitable message will be displayed to advise the user what to do next.
17. This is where the user will upload/modify an existing festival. (If a festival already exists at time of upload the user will be asked if they would like to make changes/overwrite the festival, a suitable dialog should be presented to the user to allow management of any festivals already uploaded.) The festival project must be saved locally before uploading (this will be advised if the user has not already done so). A confirmation/ error message will be presented on the attempt to upload. If an error occurs a suitable message will be displayed to advise the user what to do next.



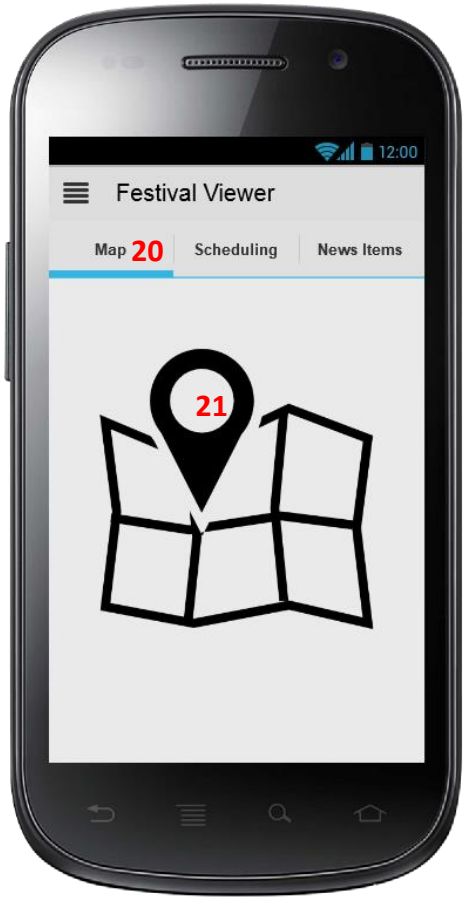
Festival Viewer

This part of the description of the GUI is for the Festival viewer mobile application. This is used by the festival goers.

The first screen shown when loading the application will be the main menu where the user can choose a festival to view (Screen grab to the left). Below is a description of each element in the application layout:

18. This icon is an easily recognisable control in most android applications. It denotes the side menu of the application. This remains present throughout the application. It will link to a side menu where the user can quickly get back to this screen or choose a festival.

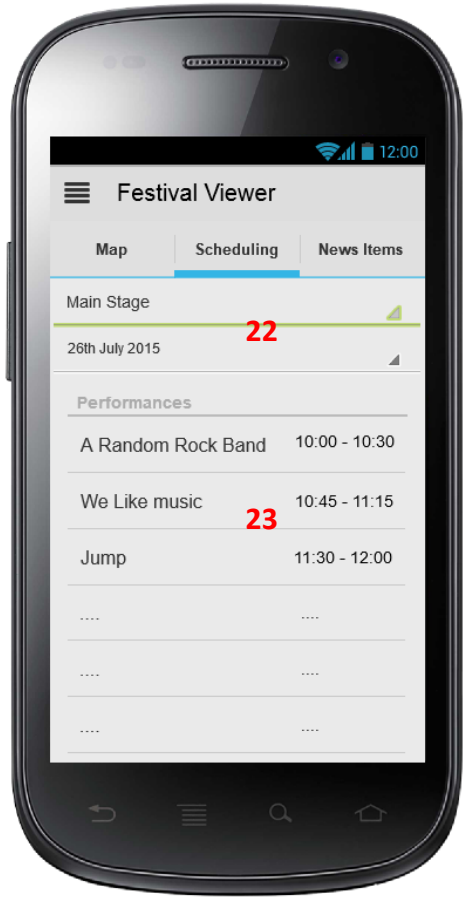
19.This is the main selection of the festivals present on the online access system. These are uploaded by the festival organisers. Upon selecting the festival the user will be asked if they would like to subscribe to the festival to receive notifications from the festival organiser (A simple tick next to the list item will denote if the user is subscribed to the festival). The main festival layout will then be displayed.



This is the main tabbed layout of the view of the selected festival. The tab active is the map part of the festival.

20. This is the tabbing area of the festival layout. The user can select or side scroll through each tab.

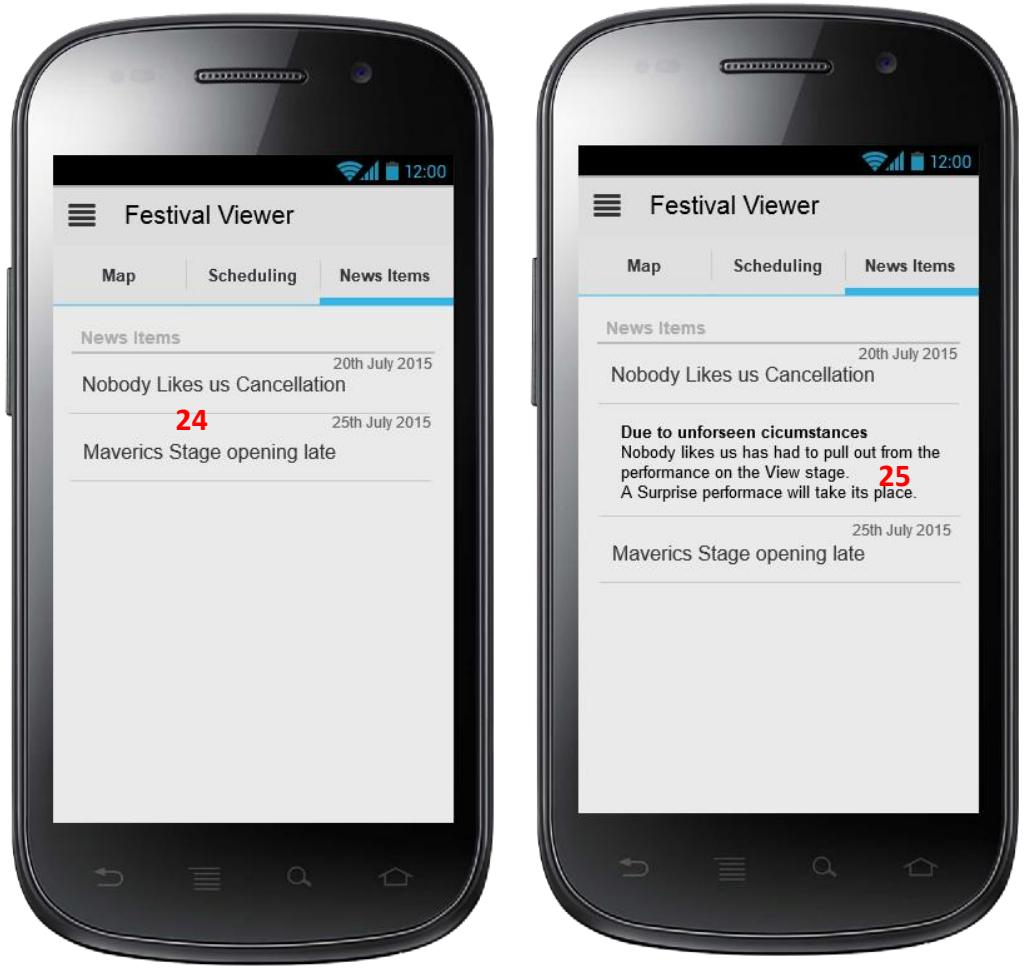
21. This is the main map of the festival. The user can pan and pinch zoom the uploaded map from the festival organiser. The user will also be able to select points of interest/stages to view the description.



This is the scheduling tab of the selected festival layout. Here the user can select a stage (or their own personal schedule) and a date to view the scheduled performances.

22. This is where the user can select the stage (or their own personal schedule) and a date to view the scheduled performances. This is done via two drop down lists.

23. After making a selection the user will be presented with this list of performances. A long press of the item in the list should give the user the option to store the selected performance to the personal schedule.

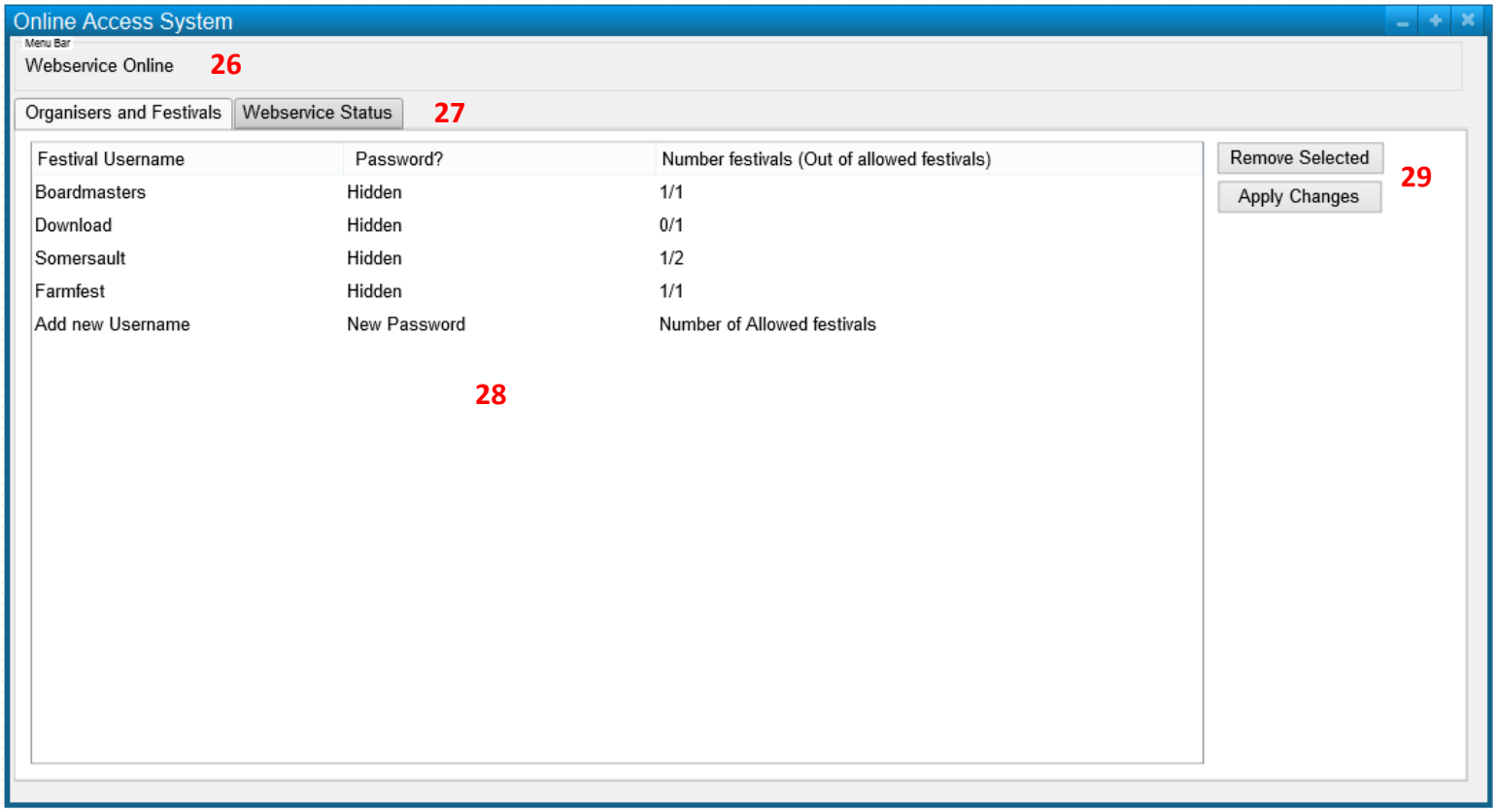


The final tab of the selected festival layout is for the news items (or notifications) that are sent out via the festival organiser.

24.The news items, with the title and date, are presented in a basic list shown here. (note: The news items should be presented in order of date with the most recent first.)

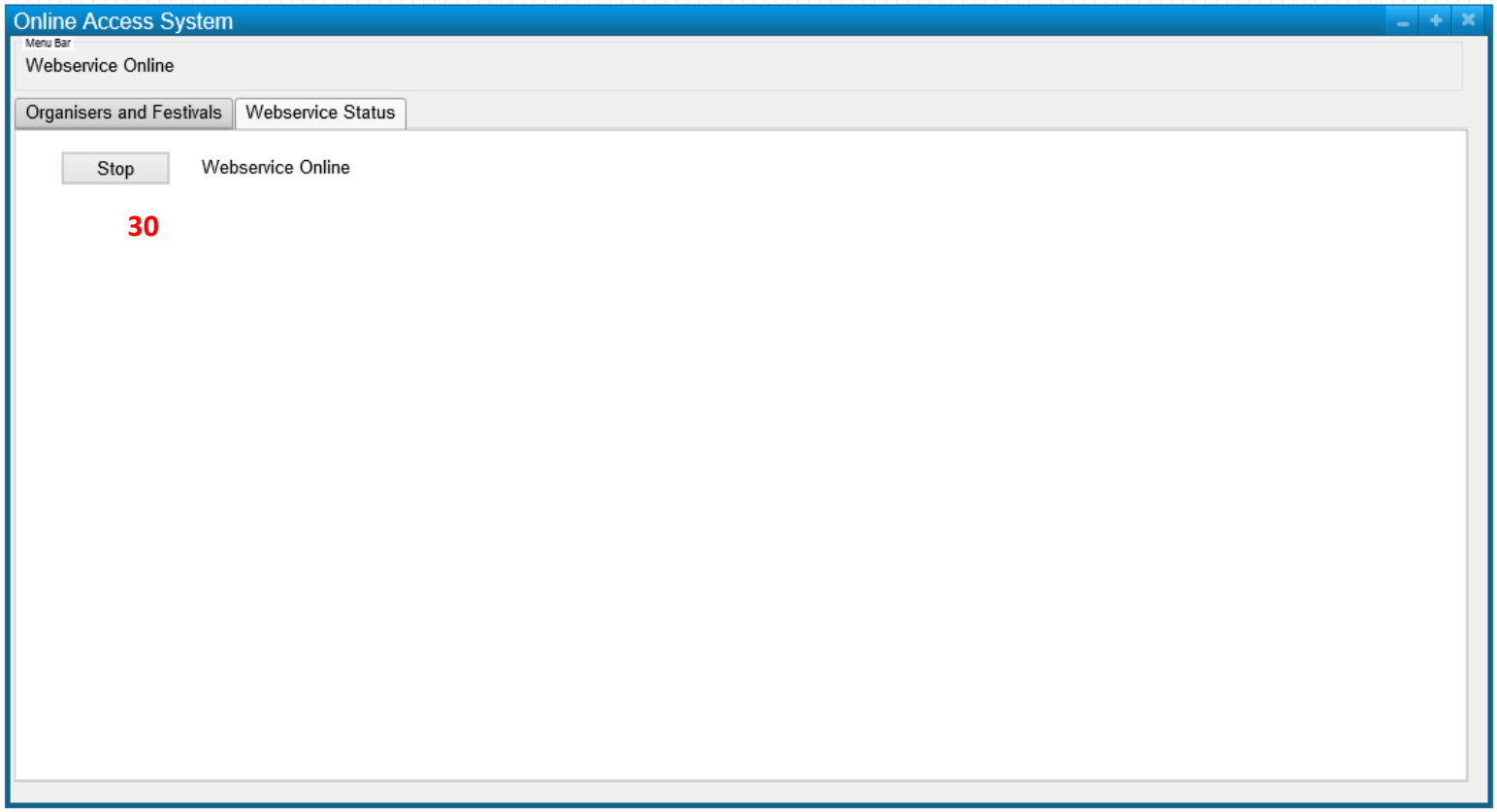
25. Upon clicking on a news item the list will expand to include the message given with the news item.

Online Access System



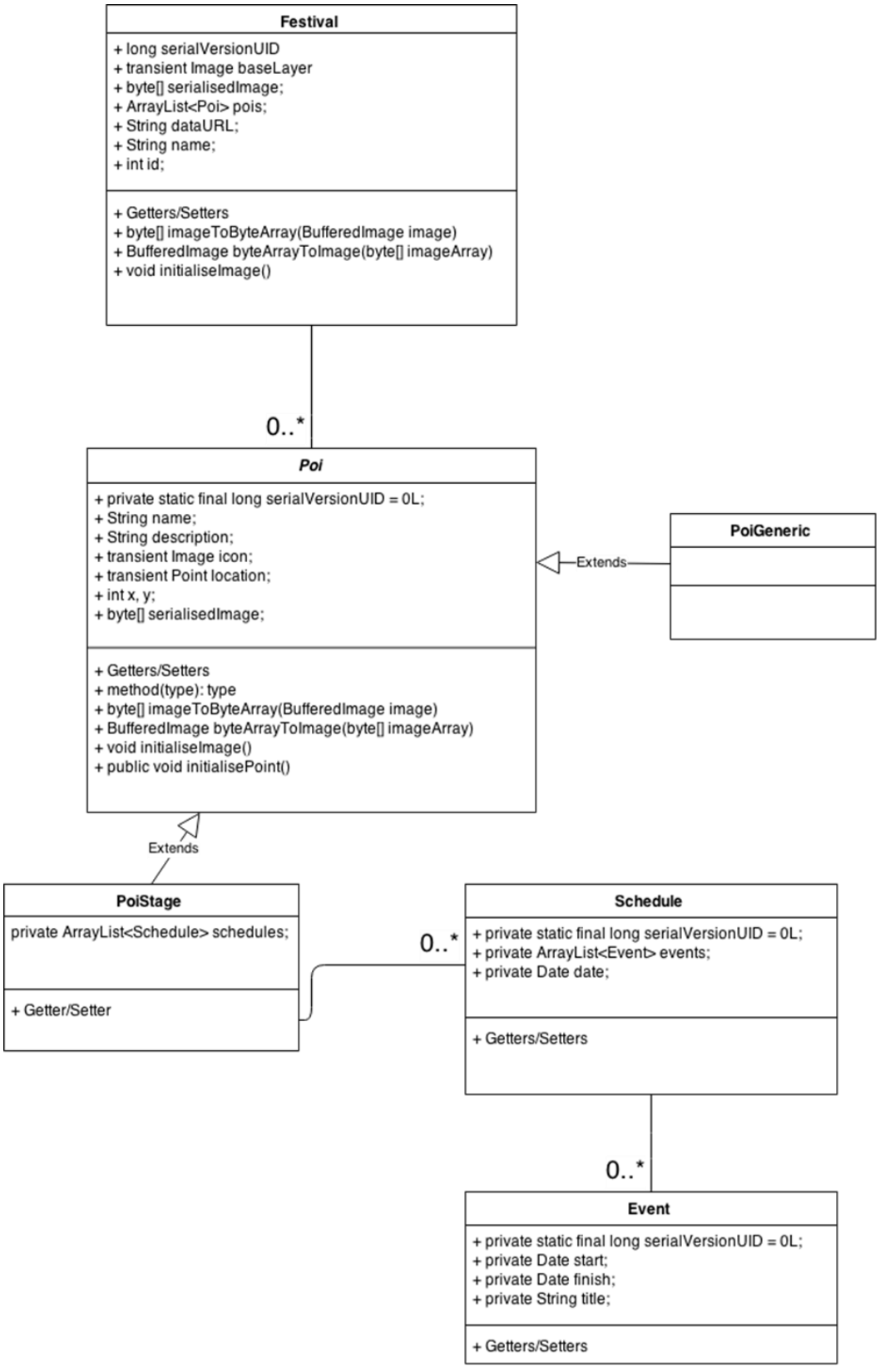
The final part of the system is the Online Access System that simply manages the authentication and regristration of the Festival organisers. As well as monitoring the status of the Webservice that provides the service for the mobile application as well as the uploading of festivals for the Festival organiser application. To give the system a sense of uniformity the window shown above is very similar to the layout of the festival organiser. The online access system is used by the owner of the webservice (i.e. the developer). Below is a description of the current tab shown in the screen grab above:

26. This is the menu bar that shows the current status of the webservice. It should show wether the service is online or offline.
27. This is the tabbed layout of the application.
28. This is the Festival organiser table showing all of the regisitered festival organisers, their passwords (which are hidden until clicked upon) and the number of festivals they are allowed. To add a new festival organiser to the system the administrator of the webservice can simply type in a new username which should automatically generate a strong password in the next cell across. The administrator can then set how many festivals the user is allowed on the system (by default it should just be one). To edit any of the information the administrator can change any value in any cell.
29. The 'Remove selected' button will remove the selected row from the table (A dialog to confirm the removal will be presented before this is done). Any changes to the table must be confirmed before leaving the tab or closing the application otherwise the information changed will be lost (the user will be reminded of this via a dialog.) The user can also use the 'Apply changes' button to save the changes.



The final tab in this application is the Web service Status tab. Here the user can stop or start the service using the ‘Stop’ button at number 30. The status of the service is also displayed next to the button. Before starting or stopping the service the user will be prompted via a dialog to confirm this is the right choice.

P. Festival Class Diagram



Q. Feature List With difficulties

Features (Mobile Application) In order of priority (highest first)	Difficultly (out of 10)
• Retrieve a pannable and zoomable detailed map of a music festival containing the locations of the: o Stages o 3rd Party Companies; fast food stalls/marquees; merchandise stalls/marquees; o Bars o Other entertainment o Toilets/Showers o ATM machines o General convenience marquees	5
• Each item on the map will be clickable to bring up a dialog of general information about the location: o Stages shall contain a schedule of events. o Other places shall have opening times.	4
• A schedule of each stage.	5
• The ability to create a personal schedule with notifications for favourite performances.	5
• Notifications/Alerts to any important news items regarding the festival.	7

Features (Desktop Application) In order of priority (highest first)	Difficultly (out of 10)
• Login to authorise use of uploading/changing data of a festival	6
• The ability to allow the user to upload a map (Image file or PDF) and then add 'widgets' (pin drops) for all the locations required on top of the initial map.	5
• Adding of scheduling information for each stage.	5
• Editing of the scheduling information in case of change or cancelation of any performance.	6
• The ability to push out notifications/alerts of any important news items regarding the festival to the users of the mobile application.	7

Features (A storage mechanism/service) In order of priority (highest first)	Difficultly (out of 10)
• Authorise Festival organisers to upload and change festival information	6
• Store the created map from the desktop	5
• Store the scheduling information	5
• Allow the desktop/web app full access to the stored information.	5
• Allow the mobile application read access to the map and scheduling information	5
• Push out mechanism to send notifications to mobile users and keep a log of notifications	8

Optional 'Nice to have' feature List In order of Priority (highest first)	Difficulty (out of 10)
• Notifications of any last minute changes to the scheduling of the performances.	8
• Map and schedule information will be available offline on the mobile application	7
• Social media integration on the mobile application: o Links to festival social feeds, twitter and facebook o Ability to submit/share photos to twitter and instagram	6
• The ability to pin point the mobile user's own tent and car on the map.	3
• The ability to save the mobile user's favourite places on the map.	3
• Forecasted weather for the festival on the mobile application	2

R. Testing Results

Functional Requirements	Test Planned	Result to Pass	Achieved Result	Result
Web service				
The web service must authorise Festival organisers to upload and change festival information.	Test incorrect and correct login details.	Only the correct login details will be accepted.	Correct login details accepted and false ones rejected.	PASS
The web service must be able to handle the storing and retrieval of a festival file that contains data such as the image files required for a base layer map and Point of Interest images.	Test that a festival file can be uploaded to the S3 storage.	A festival file has been uploaded to the S3 storage.	Festival file successfully uploaded to S3 storage.	PASS
The web service must be able to store the given scheduling information given to it from the desktop application.	Test that the upload mechanism works from the Organiser application	Uploads from the organiser application	Upload successful.	PASS
The web service must provide the means to allow methods to retrieve and update all the festival information from the festival organiser's desktop application.	Test that the organiser application can download and edit a festival.	A festival is downloaded, edited and re-uploaded.	Download, edit and re-upload successful.	PASS
The web service must provide the feature to allow the festival goer's mobile app to retrieve any festival information.	Test that the mobile app can receive festival information.	Festival data is displayed on the mobile device.	Successful download and display.	PASS
The web service must be able to take a given message from the festival organiser application and push that message to all subscribed festival goers' mobile devices.	Test that the festival organiser app, upon sending a new message, is sent to the registered mobile devices.	A message is sent from the organiser application to the registered mobile devices.	Message appeared on the mobile device.	PASS

Organiser Application				
The festival organiser must be able to log in with their details to change or upload their festival.	Confirmed under the requirement "The web service must authorise Festival organisers to upload and change festival information."			PASS
The application must support the ability to dynamically create a map and add/remove points of interest.	Test that a map can be added and a point of interest can be added and then removed.	A map is created and a POI is added then removed	Map created, POI added then removed.	PASS
The application must be able to add scheduling data for each day of the festival to a stage.	Test that scheduling data can be added for different days to a stage.	Multiple days exists with a schedule.	Schedules created for a stage for different days.	PASS
The scheduling information must be able to be changed or removed during, or before, a festival has started.	Confirmed under the requirement "The web service must provide the means to allow methods to retrieve and update all the festival information from the festival organiser's desktop application."			PASS
The application must be able to send news items to the web service for processing. These news items are then handled by the web service and pushed out to the festival's subscribers.	Confirmed under the requirement "The web service must be able to take a given message from the festival organiser application and push that message to all subscribed festival goers' mobile devices.**"**			PASS
Mobile Application				
The mobile application must include the ability to download a chosen festival from the web service and display the map developed by the festival organiser. It should include the points of interest.	Test that a selected festival will display the map information and it's POIs	A festival's map is displayed	Map is displayed	PASS

The mobile application must be able to interact with each point of interest on the map. On selecting a point of interest a description should appear.	Test that the POI is interact-able.	On selection of a POI its description is displayed	Description is displayed on selection of POI.	PASS
Each stage should contain a scheduling of performances. It should be filter on selection of a date.	Test that the scheduling is displayed correctly and is filterable.	Scheduling is displayed of each stage and can be filtered by date.	Scheduling displayed. Filterable by date.	PASS
The mobile application should allow the user to create a personal schedule of their own favourite performances.	Test that the mobile application can create a personal schedule.	A personal schedule is created.	No feature exists.	FAIL
The mobile application must allow the web service to send it messages if the user has subscribed to receive a notification about a particular festival.	The web service must be able to take a given message from the festival organiser application and push that message to all subscribed festival goers' mobile devices.			PASS