Introduction 1

1.1 Purpose

This document contains the Ticket application system requirements and its analysis. Its purpose is to serve as a reference for the system design with its intended audience who are the project coordinator, project supervisor, system designers and the CA326 module demonstration panel.

1.2 Scope

The goal of this development is to produce an app that will allow users to be able to purchase tickets and sell tickets securely.

The app will aim to prevent ticket fraud, duplication and ticket touts from occurring. This app is intended for users who want to buy tickets to an event knowing their tickets are totally legitimate and they're not fake or overpriced. This is the main benefit of using this app as it allows users to buy tickets and know 100% they have got a legitimate ticket for the correct price. By doing so the app would eradicate millions of euro falling into corrupt hands to ticket duplication and ticket touts.

1.3 Business context

There are two possible business contexts in relation to this product. These contexts are:

- Event coordinator: The product could be potentially used by events. This organization could potentially use this product as it is, or they could choose to integrate it into an existing product.

- Advertising: The product could also have the additional feature of internal advertising within the product. This could be general advertising, or it could be more targeted advertising relating more specifically to different peoples interests as stated in their profile and could generate funds from advertisers. These contexts are both viable independently of each other, but at the same time it is also conceivable that these two contexts could be used in conjunction with each other

1.4 Glossary

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible

SQL Structured Query Language, SQL is a standardized query language for requesting information from a database.

Firebase is a mobile and web application development platform developed by Firebase but now owned by Google.

Redbrick DCU‘s computing and networking society. Redbrick has servers that have SQL installed on already.

1.5 Document Overview

Section 2 contains an overall description of the product

2.1 details the product and system functions where the general functionality is discussed

2.2 contain user characteristics and objectives.

2.3 is operational scenarios.

2.4 Assumptions and constraints.

Section 3 contains details of the functional requirements where we describe the possible effects of a software system i.e. what are system must accomplish.

Section 4 has the system architecture that shows the distribution functions across potential system modules that we use.

Section 5 contains the high-level design of the system. It includes system models showing the relationship between system components and the systems and its environment.

Section 6 his section provides an initial version of the project plan, including the major tasks to be accomplished, their interdependencies, and their tentative start/stop dates.

Finally, section 7 is where the appendix is.

General Description 2

2.1 Product / System Functions

Here is the list of the main functions of the project. There may be additional functions depending on further requirements. Each function listed below will be discussed in detail in section 3.

- Sign up/Register
- Create Profile
- Generating QR code
- Purchase Tickets
- Resale of Tickets
- Validating Tickets
- Listing Tickets

2.2 User Characteristics and Objectives

The demographic of this app would consist of event coordinators and event attendees. Each event will have its own demographic, making the apps demographic the union of all the events demographics.

Being an event coordinator would require a level of technology ability which would be adequate for using this app. An event coordinator would be used to selling tickets through various mediums so using this app will cause no problems. The event coordinator will have to create a page which contains info about the event. They will also use the app to verify user's tickets. From a coordinator's perspective selling their tickets through this app will cut out fraud and possible bad publicity.

The event attendees would not require a technical background. Once they have an android or iOS device. They must create a profile and provide basic information. They then can purchase tickets for events. From a customer's perspective buying a ticket through this app will guarantee that the ticket is valid.

2.3 Operational Scenarios

User signs up

After downloading the app, the user must sign up and create a profile. First, they must enter their name and agree to the terms and conditions. Now they must create their profile. Fields such as first name, surname, email and country of residents are mandatory. Other fields such as a town or telephone number are optional.

After filling in these fields a validation email will be sent to the user's email. Once the user clicks on the link they have successfully signed up.

A logged in user buys a ticket

The user can now look through all the events (filtered to the country of residents). The user can filter the events by genre or keywords. After selecting an event the user is given more detail about the event. The user then selects “purchase ticket” and is asked how many tickets he/she wants. The user then must complete a captcha for security reasons. The user is then prompted to enter their card details to purchase the tickets. After this is completed they will receive an email containing a copy of their receipt.

Event sign up

After downloading the app, the event coordinator must sign up. First, they must enter their name, email and agree to the terms and conditions. Now they must create a profile for the event. This will contain a description, date, time, locations and number of tickets. They must also provide bank account as this account will be used for the profit of sales. 3 official references must be provided to validate the event. A validation email will also be sent. The event will not be published in the app for 3-4 working days as the references need to be contacted.

User resells a ticket

The user has decided that they can no longer make the event. The user now selects “view tickets” and selects the ticket in question. They then select “resell ticket”. The user is then prompted to confirm the resale of their ticket. The ticket has now been put back on sale and the user will receive their refund (minus a handling fee) as soon as the ticket is bought by another user.

Event validates tickets

The event coordinators use the cameras on their devices to scan the user’s QR codes. This gives the access to the user’s tickets. If the user has a ticket for the event they can enter.

2.4 Assumptions & Constraints

All users will have access to Wi-Fi or data to download the app.

All users are able to understand English which will be the language of our app.

All users will have an email.

All users will have a payment method.

Time constraints: the project has completion due date for DCU and this is the first two weeks of April 2017.

User’s constraints: Will we be able to understand and meet the needs of the end user’s requirements for the new system within the time frame.

Events constraints: Will we be able to provide the event organizers with the tools, training and testing finished product within the time frame allocated to us by DCU.

Functional Requirements 3

3.1 Sign Up/Register

Description: This function is the first step the user takes on the way to getting tickets. There will be a sign-up option when the user opens the app for the first time. There will be two options(a) for a new user and (b) for the new event coordinator. Pressing either of these options will bring up an initial form that will prompt the person for a few mandatory pieces of information where case b will slightly differ from case a. After completing this mandatory form, the user or event coordinator will be redirected to another form.

Criticality: Obviously this function is essential as every user/event coordinator needs to have a profile. We need personal information about them as to verify people and give them a unique QR code.

Technical Issues*:* The registration form will fit in with the overall app layout. We will handle the form inputs themselves using functions, which will access a firebase database to store user information. Once completed, this will create an entry. The user's table contains a foreign key which links to the user profile table.

Dependencies: None.

3.2 Create Profile

Description: After completing the initial registration form, the person will be redirected to a second form. Within this form, the users can enter more information about themselves creating a user profile. The profiles themselves will contain information such as Name, Age, Location, Email etc. Certain information will be optional, but some will be mandatory to enter.

Criticality: The app relies on every person having a verified profile so this need to be done by everyone using the app.

Technical Issues: Once again, the form will be and handled with functions and Firebase real-time database. Once completed, this form creates an entry for the user in the user profiles table. The user profile table contains optional information about a user, which is associated with the user's mandatory information through the foreign keys, so all user information can be easily referenced.

Dependencies: This area is dependent on the user having registered and agreed to our terms.

3.3 QR Code Generator

Description: After giving the relevant information a unique QR code will be generated for the user to have on their profile.

Criticality: The app relies on each user having a unique QR code so this is a pivotal part of our app. It's a very important feature as it will be used to assign tickets to a user when they purchase tickets

Technical Issues: We will aim to use a QR code generator to give everyone a unique QR code for their profile

Dependencies: None

3.4 Login

Description: This is a simple user login script that prompts a user for their username and password. The user has entered this info during the registration process, so we simply reference our database of users looking for the user name that's trying to login.

Criticality: This function is vital to the project. Without this, we have the problem of casual people abusing all functions of the system.

Technical Issues: This will require some security to stop unauthorized access to the system. An email will be sent to verify every user.

Dependencies: Depends on registration of the user name.

3.5 Edit Profile

Description: After creating their original profile on registration, it is possible that the user will choose to change some of the information on their profiles. This "Edit Profile" function will be available to all users through a user menu, which is available after login. From here, the users public profile values can be edited and updated.

Criticality: Not critical but a nice feature to have. It's added for increased functionality.

Technical Issues: Simple script that will call the users profile information from the Firebase database and present it to them to edit.

Dependencies: This is dependent on the register and login functions.

3.6 Add Event

Description: When an event coordinator has been verified they will be able to create an event where they can add tickets. Depending on the event there will be a template to fill out.

Criticality: This is a huge part of our app as people need tickets to buy so events need to add tickets to allow this. Without this, there would be no way for users to get tickets.

Technical Issues: Template to fill out as a form where event coordinator can edit their choice. There will be many templates they can follow i.e. if it is a sporting event there will be defined options like the venue, sports type already there so they can just fill in relevant information.

Dependencies: Depends on an event occurring

3.7 Edit event

Description: Allow admin/event coordinator to edit and delete information like Venues, Bands and Concert. System will allow them to change certain information about an event.

Criticality: Excellent feature to have if event date got changed or venue had to relocate as tickets can be edited and not have to start the entire process from scratch.

Technical issues: An event coordinator can change through an edit button option where the can change the form that is in the database.

Dependencies: Depends on there already being an event created.

3.8 List tickets

Description: A user can enter a location and all events that are on the system will appear where the user picked. Can filter ticket searches by whatever user searches or has preferences to go and see.

Criticality: A nice feature that suggests events to them if the location has been added. Searching for tickets is a feature that is needed because a user needs to be able to have the opportunities to see all events on the system.

Technical issues: Tickets will be filtered with a database like Firebase or MySQL, so tickets will be listed in their correct id.

Dependencies: Depends on there been events taking place

3.9 Select a ticket

Description: If a user wants to go to the event the system needs to let the user select the specific ticket to purchase. Once selected the user will be asked to confirm their selection and then be able to purchase it.

Criticality: For a user to go to an event they will need to be allowed to select a ticket, so this feature is a necessity.

Technical issues: We would use a system where a user can click on the ticket and a new page will load where the user be brought to buy the ticket or see more details about the ticket

Dependencies: Depends on the user selection.

3.10 Purchase tickets

Description: For purchasing tickets, a payment will be needed. As the system needs to deal with financial transactions we will need use a secure online payment system,

Criticality: Key feature to have in the system as people will need to have a way of getting tickets and events need money to gain a profit from their event. Events are a business at the end of the day.

Technical issue: Use system like stripe or PayPal where users can purchase the tickets are will be a secure way of transaction.

Dependencies: A user having a debit/credit card or money in their accounts to pay for the tickets.

3.11 Assign ticket

Description: The system needs to correctly assign the ticket onto the correct user profile when a ticket is purchased. The purchased ticket is assigned to the profile. The user can't see the full ticket only the receipt of it.

Criticality: This is one of the key features and is one way our app is trying to eradicate ticket touts, ticket duplication and fraud.

Technical issues: When purchased the ticket id is attached to a user profile. We will be using a database to figure out what users bought which tickets and from the database we will be able to assign the ticket onto the user_id().

Dependencies: Depends on which user bought which ticket.

3.12 Resale of Tickets

Description: If a user buys a ticket and can't go anymore there will be an option to resell the ticket to other users. The unique thing is users will not be able to sell it at anything over the original price.

Criticality: This is one of the main features in the app. It gets rid of overpricing ticket touts as users can't pay anything over the sale price of a ticket.

Technical issues: Have a queue or stack system where we put people's tickets back up to resell.

Dependencies: Reselling depends on a person having a ticket.

3.13 Security

Description: Our system will have security features from buying tickets to setting up a profile. Also, cross-site scripting and database injection attacks must be protected from happening.

Criticality: This is needed as the system is handling money and people's personal details so need to have security features embedded into the app so personal information is not seen by everyone.

Technical issues: For verifying user's payments we will using a captcha test first invented by Alex Turning and for users setting up a profile they will have verified their profile through the email they use an email will be sent that contains a link that the user clicks on that validates the account.

Dependencies: People having a working email.

3.14 GUI functionality

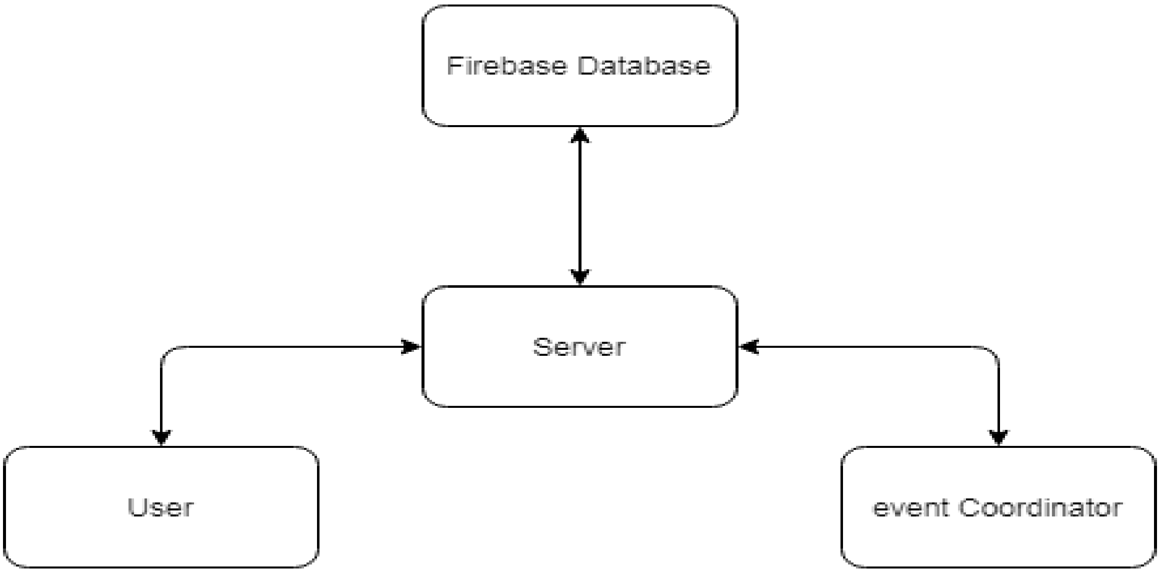
Description: The system should be simple to use for users and admins so from the first time opening the app no manual is required for either user buying a ticket or selling. The system will adhere to Android and iOS legislation too.

Criticality: An app could be well coded but if people can't use it what's the point! An app needs to be easy to use for users to use the app so is critical to get right. People need stay interested and allow them to get to the information or page they require as quickly as possible

Technical issue: Using some software like react native where we can make a user-friendly interface that works with android and iOS. Must make the accessibility of the app a priority too so as many users could use it.

Dependencies: None

System Architecture 4



User: The user uses the application to view information about events. This side of the app will contain an accessible and pleasing UI to attract sales.

Event Coordinator: The event coordinator uses the application to create & edit events. This side of the app will be more focused on allowing the coordinator to enter information in a clear and concise manner.

Server: The server is used to send information to and from the database to the application.

Database: The database is used to store all the information about the events, users, coordinators and tickets

High-Level Design 5

5.1 Use Case Diagram

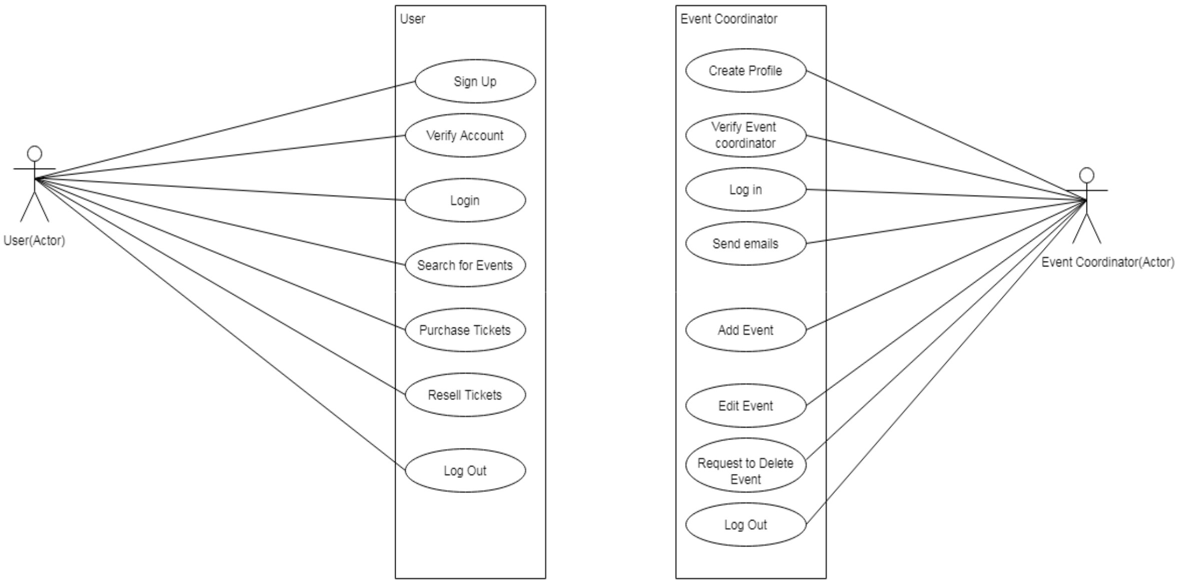


Fig 5.1.1

Fig 5.1.1 contains a use case diagram where it aided us in the gathering of our functional requirements. It describes the set of actions that we want our system to perform with its actors.

5.2 Sequence Diagram

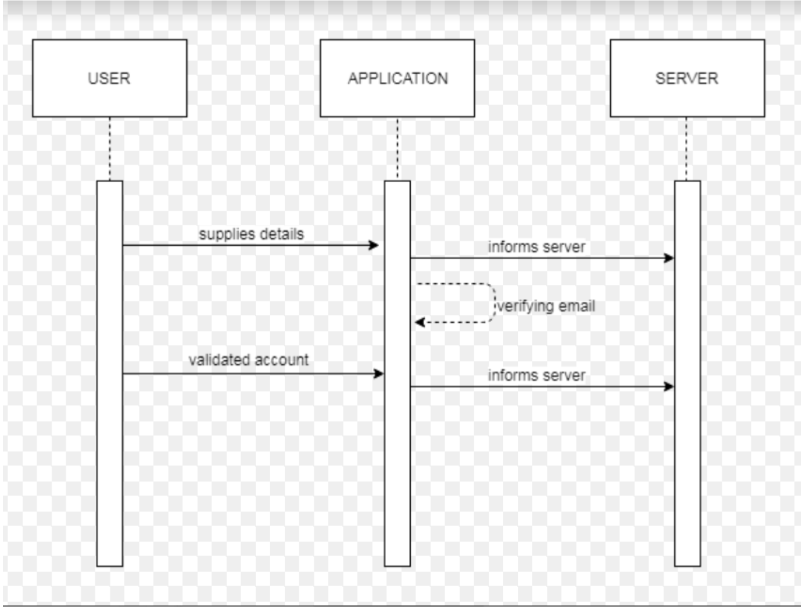


Fig 5.2.1

Figure 5.2.1 is a sequence diagram that shows the sequence of events when a user signs up to the app. You can see how the flow of data goes between the user to the application to the server. You can see how the email will be sent when a user signs up to verify them.

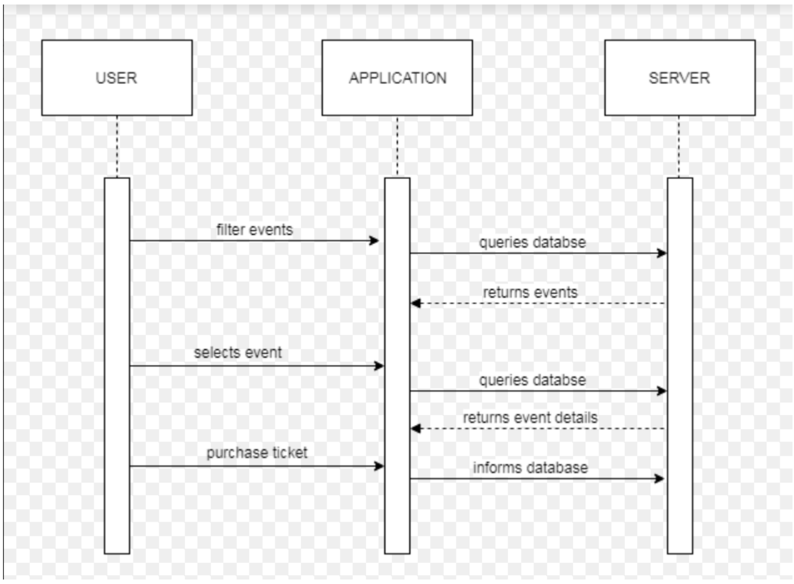


Fig 5.2.2

Figure 5.2.2 is another sequence diagram that depicts the sequence of events that will happen when a user purchases a ticket. You can see how the data flows between the user, application and the server. From the user inputs on the application data goes to the database and you can see how the database will query and the server will return information back to the application which will in turn return information back to user.

5.3 Context Diagram

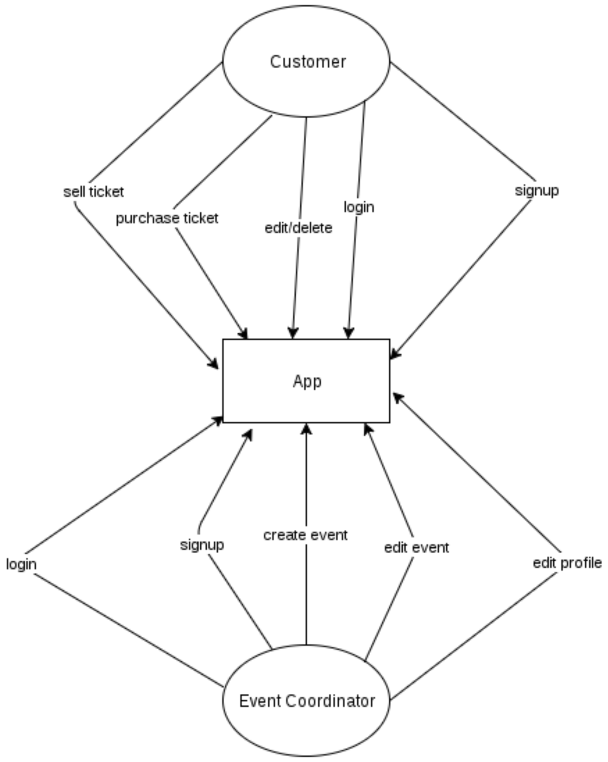
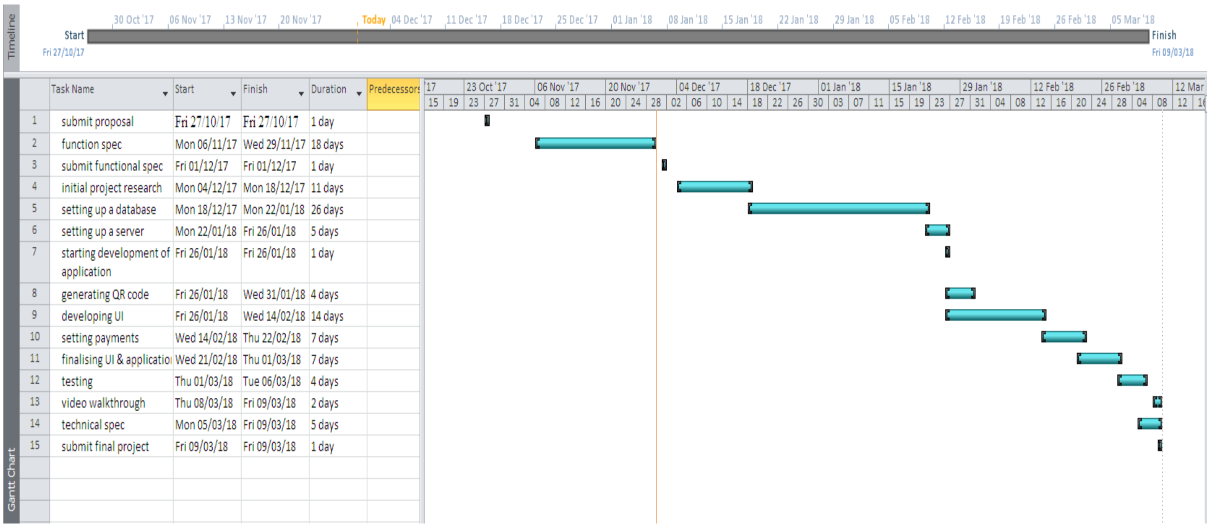


Fig 5.3.1

Figure 5.3.1 is a context diagram and shows all possible paths the user or event coordinator can take and how the entities interact with each other and what relationships it entails. This diagram is a high-level view of our system.

Preliminary Schedule 6

6.1 Task List & Gantt Graph



	Task Name	Start	Finish	Duration	Predecessors
1	submit proposal	Fri 27/10/17	Fri 27/10/17	1 day	
2	function spec	Mon 06/11/17	Wed 29/11/17	18 days	
3	submit functional spec	Fri 01/12/17	Fri 01/12/17	1 day	
4	initial project research	Mon 04/12/17	Mon 18/12/17	11 days	
5	setting up a database	Mon 18/12/17	Mon 22/01/18	26 days	
6	setting up a server	Mon 22/01/18	Fri 26/01/18	5 days	
7	starting development of application	Fri 26/01/18	Fri 26/01/18	1 day	
8	generating QR code	Fri 26/01/18	Wed 31/01/18	4 days	
9	developing UI	Fri 26/01/18	Wed 14/02/18	14 days	
10	setting payments	Wed 14/02/18	Thu 22/02/18	7 days	
11	finalising UI & applicatio	Wed 21/02/18	Thu 01/03/18	7 days	
12	testing	Thu 01/03/18	Tue 06/03/18	4 days	
13	video walkthrough	Thu 08/03/18	Fri 09/03/18	2 days	
14	technical spec	Mon 05/03/18	Fri 09/03/18	5 days	
15	submit final project	Fri 09/03/18	Fri 09/03/18	1 day	

Fig 6.1.1

Figure 6.1.1 shows the proposed schedule of each task to be completed in the lead up to the project submission. We will be using this as a guide to keep on track with things but there could be some small adjustments made to it during the time we do the project.

Appendices Resources 7

Research tools:

- https://www.w3schools.com/
- https://www.redbrick.dcu.ie/
- https://firebase.google.com/
- https://developer.android.com/index.html
- https://www.udemy.com/

Related sites:

- http://www.ticketmaster.ie/
- https://www.viagogo.ie/
- https://www.tickets.ie/
- https://www.ticketswap.ie
- https://www.seatwave.ie/