



*3<sup>rd</sup> Year Project:*

Innovative Cinema Listings

# Functional Specification

*Authors:*

Shaun Carey 16450454

Nigel Guven 14493422

*Supervisor:*

Charles Daly

# Functional Specification

## Table of Contents

1. Introduction.....	3
1.1. Overview	
1.2. Business Context	
1.3. Glossary of Terms	
2. General Description.....	6
2.1. Product/System Functions	
2.2. User Characteristics and Objectives	
2.3. Operational Scenarios	
2.4. Constraints	
3. Functional Requirements.....	8
3.1. User Functionality	
3.1.1. First-Time Usage	
3.1.2. Manage Preferences/Criteria Search	
3.1.3. Add/Remove Films to a Saved Section	
3.1.4. Display Further Information on Saved Films	
3.2. System Functionality	
3.2.1. Dynamic Scraping of Web Data	
3.2.2. Continuous Usage of Google Maps API	
4. System Architecture Diagram .....	12
5. High-Level Specification.....	13
5.1. Context Diagram	
5.2. Data Flow Diagram	
5.3. User Sequence Diagram	
6. Appendices.....	16
6.1. Research Tools and References	
6.2. Sites for Inspiration	

# Functional Specification

## 1. Introduction

### 1.1 Overview

The application is being developed for Android operating systems, mainly for the mobile phone market. It stores listings for cinemas across all companies in one single location. Users will be able to provide their location to the application which will then produce an array of films which are showing in their area based on distance parameters. The user can swipe right on films that interest them or left otherwise.

According to the selection of films, the user may examine what cinemas are hosting the film and at what time. Then the user can choose where they would like to see that film.

Ultimately, this application aims to invigorate the film industry by bringing public data from different companies into a single simple distribution centre. It will be much easier to use than having to peruse multiple websites for information on cinema listings. Also, it gives the user a multiple-choice perspective which may guide into making a more informed decision on their choice of film.

### 1.2 Business Context

The potential for commercial usage of the Innovative Cinema Listings is immense. Influenced by the mobile dating application, Tinder, users can swipe between films based on personal preference. Applications like these that are simple to use and contain useful information are quite popular especially on Android devices which occupy 70% of the mobile operating systems market.

The application can be used as an analytics engine whereby companies may track the popularity of films based on geographic location and the swiping mechanism of the application. This would also give a new payload of information to cinemas to track their guests' preferences which allows the calculation of how long a film should stay in a theatre.

# Functional Specification

## 1.3 Glossary of Terms

### ***Android:***

Android is a mobile OS which was developed by Google and based upon the Linux Kernel. It is specifically designed for handheld devices like smartphones, smart watches and tablets.

### ***Operating System:***

An operating system is software that can manage a computer's hardware and software resources. The OS uses multi-tasking to schedule tasks and allocate memory to various applications.

### ***Android Studio:***

Android Studio is the official IDE for the Android operating system. It comes included with an interface design package and the Android SDK.

### ***Python:***

Python is a high-level programming language which supports multiple programming paradigms. It is utilised in web development, software development and machine learning algorithms.

### ***Java:***

Java is a high-level programming language which is object-oriented and follows the 'write once, run anywhere' memorandum which allows it to run on any platform which supports Java without needing recompilation.

### ***BeautifulSoup:***

BeautifulSoup is a package in Python which allows for the parsing of XML and HTML documents, it is useful for web scraping and data mining from websites where it can extraction specific information.

## Functional Specification

### ***Firebase:***

Firebase is a mobile and web application development platform which provides a database for applications which is located on the Firebase cloud. Objects are stored as JSON files.

### ***AJAX:***

Ajax is a set of technologies which allow users to create asynchronous web applications whereby it can dynamically manipulate content on a webpage without interfering with the page.

### ***Google Maps:***

Google Maps is a web mapping application which offers streets, real-time traffic conditions and route planning algorithms

### ***API:***

A set of functions, tools and procedures that allow the creation of applications which access features of an operating system or another service.

### ***Gradle:***

Gradle is a build file Automator that assembles packages into a neat collection using its build logic. It allows developers to customise their own packages like in Android while also calling on previously built infrastructure.

# Functional Specification

## 2. General Description

### 2.1 Product/System Functions

The following is a list of the functions of our application split up into the relevant sections. Each function will be elaborated in more detail in the functional requirements section.

#### *User functionality*

- On first open: Accept Privacy Agreement, probe user for information on their preferred genres etc.
- Manage Preferences/Search Films by Criteria.
- Add/Remove films to 'liked' and 'ignored'.
- Display more information on 'liked' films.

#### *System functionality*

- Dynamic scraping of film/cinema data and incorporating it into one, clean interface using BeautifulSoup and Firebase.
- Continuous use of Google maps.

### 2.2 User Characteristics and Objectives

The application will be available to Android users (70% of the phone market) who also have internet access. The target audience is pretty much every Android user who is interested in viewing films, whether it's a late teen wanting something action-filled, or a parent wanting to find something nearby for their younger children to watch. The user would be expected to have very basic knowledge in the use of touchscreen applications (which should be a given if they own an Android machine) as it will be designed with a very basic user-friendly interface. In the rare case that a user does not have much expertise on navigating touch-screen apps, it is hoped that this basic interface will accommodate its users by not being too complex while still being fit for purpose.

Users who are familiar with similar applications such as Trivago or Tinder should have no problem with this app as a lot of our inspiration is being taken from these: with the user filling in criteria as well as the swiping right/left feature to favorite/ignore certain films, respectively. Once the user has sorted their films into the 2 categories (right and left, or 'liked' and 'ignored'), they will be able to check each for their most nearby showing, closest showing in terms of time, rating, and other relevant information. Also, there will be a distance in a metric specified by the users which can be opened in Google Maps API and create a route from the user's location to the specified cinema.

# Functional Specification

## 2.3 Operational Scenarios

As users won't have a 'user' profile with personal information per se, and due to the app not containing any form of authorization (username/password), each user will be given access to every user function with no restrictions from the get-go. We took this design approach as to not draw users away when they realize they must spend time signing up for something that doesn't utilize any personal information anyway.

### *First time opening:*

The app will welcome the user and prompt them for their favorite genres and other settings such as the maximum preferred distance, etc. and alert them to the fact that they can change these settings at any point. They will then be asked to accept a privacy agreement as is standard in most applications. This privacy agreement contains information which allows the application to receive a location and also enforce the user to connect to Wi-Fi in order for the application to function. From this point onwards, the app will operate as normal for the user.

### *Maintenance and Lack of Access:*

In a case where the app is down due to maintenance or in the case of no access to an internet point, the user will be alerted to this fact and will be told to check back later.

## 2.4 Constraints

The following is a list of constraints which dawned on us while coming up with the design of our project. Other than the standard and expected time constraints such as the project deadline, we've also thought of a few more:

- Possible drawback: The application may take a few seconds to load due to the scraping that will be happening in the background. Users with a low attention span may be drawn away by this. We may incorporate a cycle of data to refresh while a user is on the application.
- User feedback: Unless a few willing participants are willing to take part in a feedback form, we probably won't be able to generate much feedback from users who would be willing to use an app such as this on a regular basis.
- Web Scraping Legality: Although we are now confident of the legality of this project, there were a few hiccups at the beginning over the fact of whether taking this information from other sites is even legal. After some discussions, emails for permission, and reading of some terms of services, we are confident that we can continue forward.
- Memory/Storage: One of the main problems we face whether we have the storage space on our server accounts to include every function that we wish to include. Not being able to include some of our features could also be considered another user drawback.

## Functional Specification

### 3. Functional Requirements

#### 3.1 User Functionality

##### 3.1.1 First-Time Usage

###### *Description*

Once the user has downloaded the app and opened it, they will be greeted with a welcome message. Thereafter, they will be told to read and accept a privacy agreement to continue any further. Once this is agreed upon, the user will be asked if they want to edit any of their preferences (distance, favorite genres...) or to just continue to the app with no preferences for the time-being. After this stage, the user has full access to navigate throughout the application.

###### *Criticality*

This is one of the most essential requirements out of the user-related ones. Not only do we want the users to feel welcome and offer them a chance to filter out films they may not be interested in right off that bat, privacy agreements are now required by law if a business intends to collect personal information from users. Although we have no plans to do this yet, it may be an option in the future. It is recommended nowadays for every business to include a privacy policy, even if you don't collect any personal information. Even if all the page says is that you don't collect information, it's generally a good idea to include one.

###### *Technical Issues*

No major ones that come to mind. Perhaps the use of cookies to determine whether it's a user's first time may be a problem to us as it's not something we're familiar with, but we don't foresee any problems in learning how to implement it.

###### *Dependencies*

This function is dependent on the user having downloaded the app and it being their first time using it.



## Functional Specification

### 3.1.2 Manage Preferences/Criteria Search

#### *Description*

The user may enter their profile settings to change any preference they may have entered incorrectly on start-up, or anything they may just want to change in general. The app will filter out films that the user may not be interested in. The user may choose to enter no criteria, at which point the app will display every film currently being shown in cinemas.

#### *Criticality*

Vital. Users may go from wanting to only see Action movies displayed on one week, to wanting to see every film currently showing on a different week.

#### *Technical Issues*

Different cinemas may class the same film as a different genre, meaning there's a small chance the user may have films displayed which may be of no interest to them. A very minor issue as the user can choose to ignore said film anyway.

#### *Dependencies*

This function is dependent on the user having set their preferences on the initial opening.

### 3.1.3 Add/Remove films to a 'Saved' section

#### *Description*

Once the user has confirmed their preferences, they will be met with the 'main' interface, which is quite Tinder-like in appearance. Rather than showing various dating profiles, it will show all films which match all the criteria which the user has entered, with a 'like' and 'ignore' button somewhere nearby. It is hoped that the app will also incorporate swiping right for like and swiping left for ignore.

#### *Criticality*

The main user function of the application.

#### *Technical Issues*

None which stand out from a user perspective. From the system side of things, deciding what film matches certain criteria could prove to be tricky and will have to be quite reliant on keywords.

#### *Dependencies*

This function is dependent on the user having set their film preferences.

## Functional Specification

### 3.1.4 Display more information on 'Saved' Films

#### *Description*

This is where the user will be able to view all information regarding the films they have added to their liked/interested category. From local cinema times, to ratings, to brief descriptions/reviews, everything will be all in one neat, elegant interface for the user so that they don't have to go browsing various cinema and rating websites.

#### *Criticality*

Vital – and the main idea behind the application.

#### *Technical Issues*

None apparent.

#### *Dependencies*

This function is dependent on the user having sorted through all films matching their criteria and have added each to a 'liked' and 'ignored' category.

# Functional Specification

## 3.2 System Functionality

### 3.2.1 Dynamic Scraping of Web Data

#### *Description*

This function is being operated in the back-end of the application. A Python script will web scrape data from various access points on the internet. These websites are the areas of key information which take data such as cinema listing times, movie genres, ratings and descriptions and place them onto a Firebase database. It is from the database that the Android application will then easily read data and display it on a clean user interface.

#### *Criticality*

This function collects all the data for the application and so it is vital.

#### *Technical Issues*

The plan to develop the user interface is on Android Studio which does not support Python directly. However, there are some workarounds by including Python in the Android Gradle files. This problem must be overcome or else solved by keeping the Python script and UI application separate.

#### *Dependencies*

This function is not dependent on any other functions but rather is the backbone behind all other functions in the schematics of the application.

### 3.2.2 Continuous use of Google Maps API

#### *Description*

Google Maps API will play an important role in the application. Users will view a list of nearby cinemas from a Maps window on the UI. Also, users may progress from the application to a map with a specified route to a cinema once they have chosen a film and click the specific link.

#### *Criticality*

Not too critical, but it would add to our goal of having everything in one app rather than the user having to navigate to Google maps separately to get the location of the desired cinema.

## Functional Specification

### *Technical Issues*

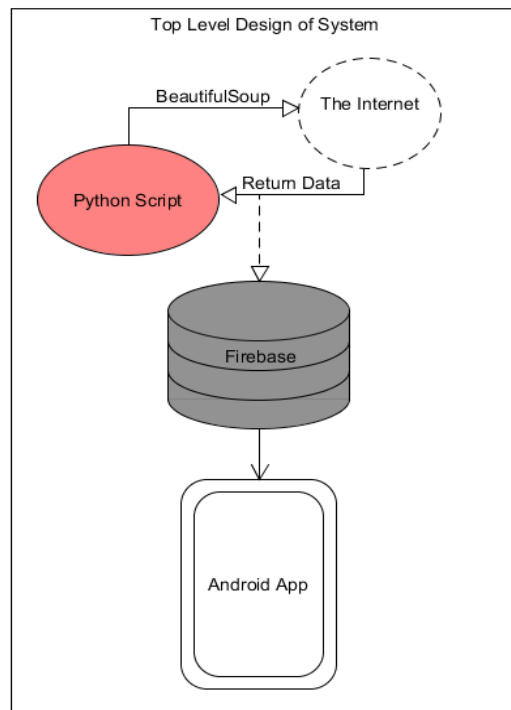
There are issues regarding location of the user. If a user disables their location on the Android settings, then the application may have profound irregularities.

### *Dependencies*

The Maps functionality is dependent on enabling the location of the Android phone. Otherwise, the map will be an optional interface which the user can choose to visit at any point.

## Functional Specification

### 4. System Architecture Diagram



**Fig 4.1**

**Fig 4.1** illustrates the system architecture of the application. There are three main functional modules.

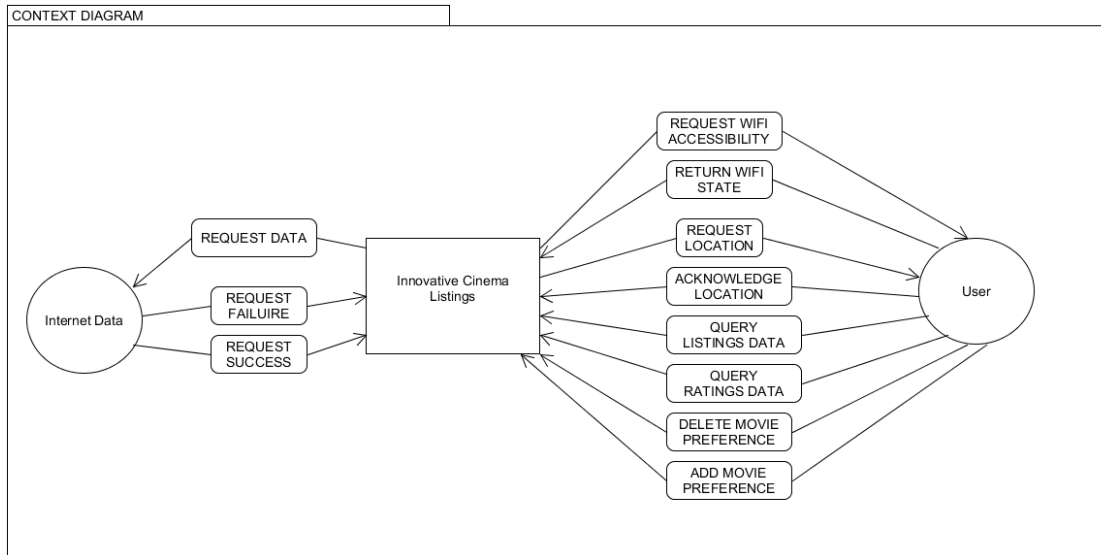
1. **Python Script** – Coordinates the dynamic web scraping of cinema listings information in cycles from the internet. Some libraries such as BeautifulSoup are utilised to extract HTML tag information.
2. **Firebase** – This database for Android applications will retrieve data from the python scripts and will store the information in categories of Cinemas, Listings, Locations, Ratings etc.
3. **Android Application** – Written in Java, the front-end of the application will display all the information from the database in a clean user-friendly interface. Users interact with the application at this level.

# Functional Specification

## 5. High-Level Specification

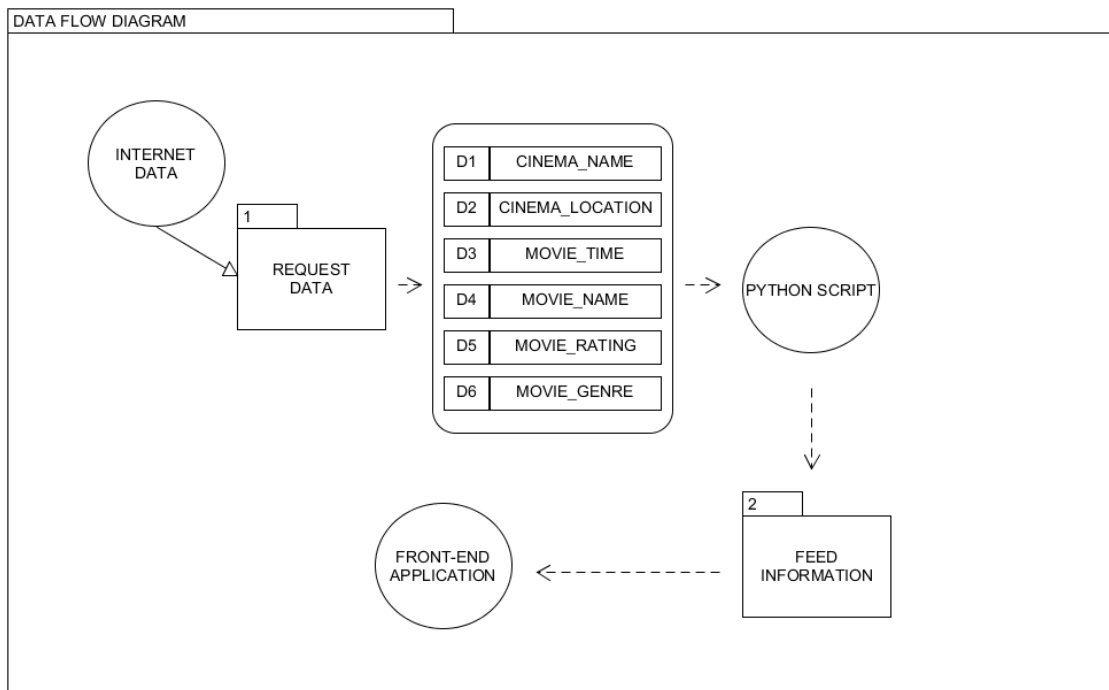
*The following diagrams are modelled using Structured Systems Analysis and Design Methodologies. They are designed using UMLet, an open source UML tool for drawing diagrams:*

### 5.1



**Fig 5.1** shows the external entities which interact with the system.

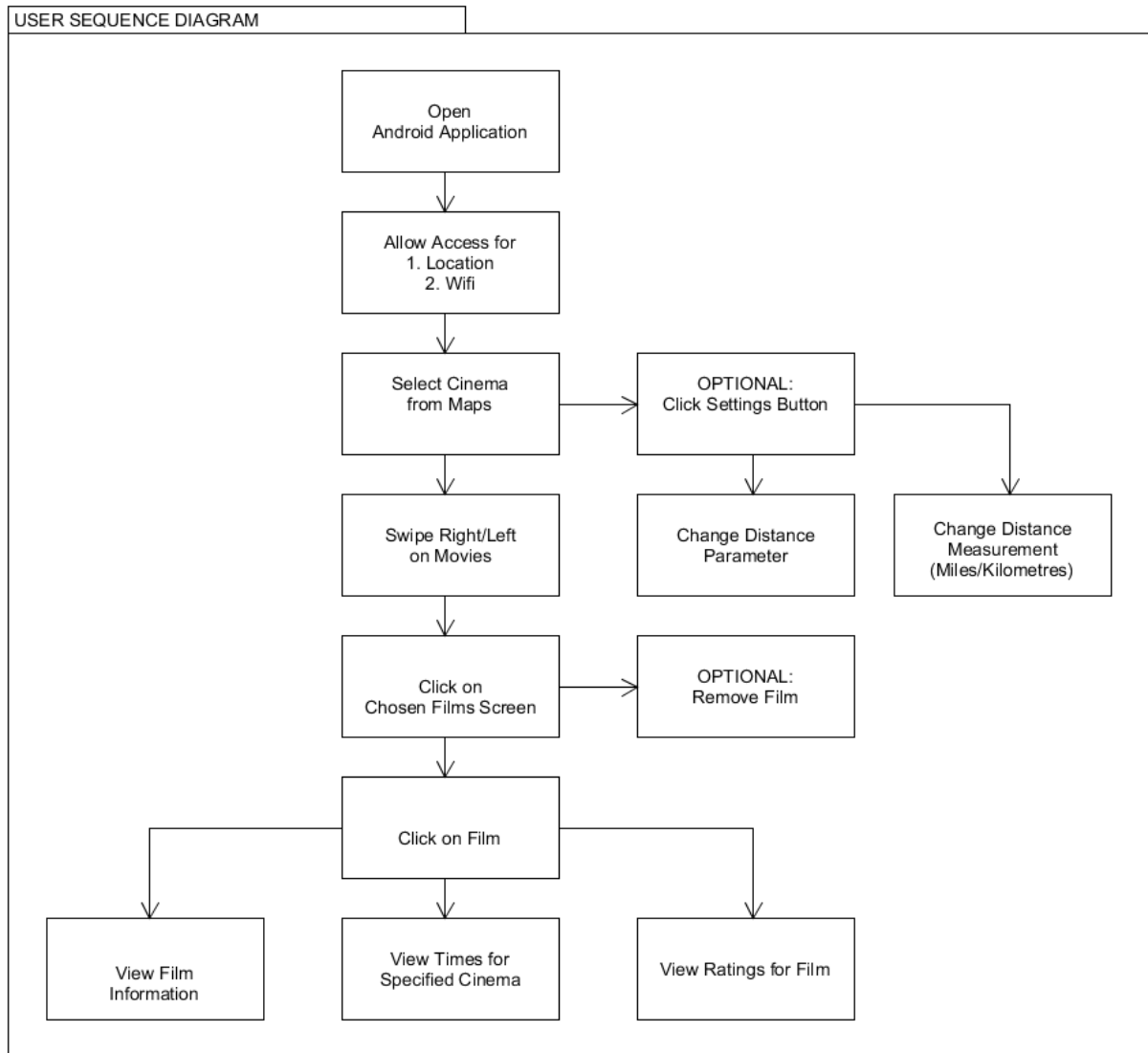
### 5.2



**Fig 5.2** shows the flow of data throughout the entire system.

## Functional Specification

### 5.3



**Fig 5.3** illustrates the sequence a user will take through the front-end interface of the application. Includes optionality and multiple-choice sequences. Options are shown from the first availability of their selection on the user interface.

## Functional Specification

## 6. Project Work Plan

[illegible]



### 7. Appendices

#### 7.1 Research Tools and References:

Python	<a href="https://www.python.org/">https://www.python.org/</a>
BeautifulSoup	<a href="https://pypi.org/project/beautifulsoup4/">https://pypi.org/project/beautifulsoup4/</a>
Firebase	<a href="https://firebase.google.com/">https://firebase.google.com/</a>
Android	<a href="https://www.android.com/">https://www.android.com/</a>
Java	<a href="https://www.java.com/en/">https://www.java.com/en/</a>
UMLet	<a href="https://www.umlet.com/">https://www.umlet.com/</a>
Google Maps	<a href="https://maps.google.com/">https://maps.google.com/</a>
Excel	<a href="https://office.live.com/start/Excel.aspx">https://office.live.com/start/Excel.aspx</a>

#### 7.2 Sites for Inspiration:

Entertainment.ie	<a href="https://entertainment.ie/">https://entertainment.ie/</a>
Tinder	<a href="https://tinder.com/?lang=en">https://tinder.com/?lang=en</a>
Trivago	<a href="https://www.trivago.ie/">https://www.trivago.ie/</a>
IMDb	<a href="https://www.imdb.com/">https://www.imdb.com/</a>
Odeon Cinemas	<a href="https://www.odeoncinemas.ie/">https://www.odeoncinemas.ie/</a>

# DCU School of Computing Assignment Submission

**Student Name(s):** Nigel Guven Shaun Carey  
**Student Number(s):** 14493422  
**Programme:** BSc in Computer Applications  
**Project Title:** Innovative Cinema Listings Functional Specification  
**Module code:** CA326  
**Lecturer:** David Sinclair  
**Project Due Date:** 5th December

## Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying is a grave and serious offence in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion, or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references.

I have not copied or paraphrased an extract of any length from any source without identifying the source and using quotation marks as appropriate. Any images, audio recordings, video or other materials have likewise been originated and produced by me or are fully acknowledged and identified.

This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. I have read and understood the referencing guidelines found at <http://www.library.dcu.ie/citing&refguide08.pdf> and/or recommended in the assignment guidelines.

I understand that I may be required to discuss with the module lecturer/s the contents of this submission.

I/me/my incorporates we/us/our in the case of group work, which is signed by all of us.

**Signed:**    
Nigel Guven Shaun Carey

Print this page