THE UNIVERSITY OF HUDDERSFIELD

School of Computing and Engineering

ASSIGNMENT SPECIFICATION

Module details	
Module Code	CIT2202
Module Title	Web Development
Course Title/s	BACB, IT, Web Design, Web Prog, Web Prog with Cyber Security, Computing

Assessment weighting, type and contact details		
Title	Web Application Development	
Weighting	60%	
Mode of working for assessment task	Individual/Group Note : if the assessment task is to be completed on an individual basis there should be no collusion or collaboration whilst working on and subsequently submitting this assignment.	
Module Leader	Matthew Mantle	Contact details: SJ5/06 m.e.mantle@hud.ac.uk
Module Tutor/s	????	

Submission and feedback details	
Hand-out date	Term 1: Week 1
How to submit your work.	You need to do several things to submit the work: 1. Compress your website into a zipped folder (no .rar or 7z please). This folder should contain all the files necessary for the website (PHP files, HTML docs, CSS etc). You are **not** required to submit copies of the database tables 2. On Brightspace, under Assignments, submit your zipped assignment website. 3. A resources statement. You are encouraged to undertake independent research, find and use suitable libraries and frameworks, learn from other people's examples and follow tutorials to learn new skills. However, it is important that you clearly acknowledge the resources you have used to help you develop your application. You will need to download and complete the resources statement from Brightspace and submit this along with your work. Again, this should be submitted on Brightspace.

Submission and feedback details	
Submission date/s and times	14th January 2021 11:59PM
Expected amount of independent time you should allocate to complete this assessment	24hrs (assuming you have completed all the module practical work fully, 3x8hr days
Submission type and format	You need to do several things to submit the work: 1. Compress your website into a zipped folder (no .rar or 7z please). This folder should contain all the files necessary for the website (PHP files, HTML docs, CSS etc). You are **not** required to submit copies of the database tables 2. On Brightspace, under Assignments, submit your zipped assignment website. 3. A resources statement. You are encouraged to undertake independent research, find and use suitable libraries and frameworks, learn from other people's examples and follow tutorials to learn new skills. However, it is important that you clearly acknowledge the resources you have used to help you develop your application. You will need to download and complete the resources statement from Brightspace and submit this along with your work. Again, this should be submitted on Brightspace.
Date by which your grade and feedback will be returned	4th Feb 2021

Additional guidance information	
Your responsibility	It is your responsibility to read and understand the University regulations regarding conduct in assessment. Please pay special attention to the assessment regulations (section 4) on Academic Misconduct. In brief: ensure that you; 1. DO NOT use the work of another student - this includes students from previous years and other institutions, as well as current students on the module. 2. DO NOT make your work available or leave insecure, for other students to view or use. 3. Any examples provided by the module tutor should be appropriately referenced, as should examples from external sources. Further guidance can be found in the SCEN Academic Skills Resource and UoH Academic Integrity Resource module in Brightspace. If you experience difficulties with this assessment or with time management, please speak to the module tutor/s, your Personal Academic Tutor, or the School's Guidance Team. (sce.guidance@hud.ac.uk).
Requesting a Late Submission	You are reminded to 'back-up' your work as late submission requests will not be given for lost work, which includes work lost due to hardware and software failure/s. Late submission requests will only be approved if you can demonstrate genuine, unexpected circumstances along with independent supporting evidence (e.g. medical certificate) that may prevent you submitting an assessment on time. Submit your request for Late Submission via MyHud/MyStudies within 2 working days of the due date. Late submission requests, up to a maximum of 10 working days, but typically 1-5 working days, will be considered provided that there is appropriate evidence which clearly indicates reasons for the request. You will have 5 working days after submitting a request to provide the evidence. Failure to submit evidence will result in the request being rejected and your work being marked as a late submission (see below). If you are unable to submit work within the maximum late submission period of 10 days, contact the School's Guidance Team. (sce.guidance@hud.ac.uk), as you may need to submit a claim for Extenuating Circumstances (ECs).

Additional guidance information	
Extenuating Circumstances (ECs)	An EC claim is appropriate in exceptional circumstances, when an extension is not sufficient due to the nature of the request, or it concerns an examination or In-Class Test (ICT). You can access the EC claim form on the Registry website; where you can also find out more about the process. You will need to submit independent, verifiable evidence for your claim to be considered. Once your EC claim has been reviewed you will get an EC outcome email from Registry. If you are unsure what it means or what you need to do next, please speak to the Student Support Office – Room SJ1/01 An approved EC will extend the submission date to the next assessment period (e.g July resit period).
Late Submission (No ECs approved)	Late submission, up to 5 working days, of the assessment submission deadline, will result in your grade being capped to a maximum of a pass mark. Submission after this period, without an approved extension, will result in a 0% grade for this assessment component.
Tutor Referral available	NO
Resources	• Please note: you can access free Office365 software and you have 1 Tb of free storage space available on Microsoft's OneDrive – Guidance on downloading Office 365.

Web Application Development

Assignment Aims

This assignment aims to provide students with the opportunity to demonstrate skills in developing database driven web applications.

Learning Outcomes

Knowledge and Understanding Outcomes
2. Critically evaluate security risks and formulate a strategy to secure a database driven web application.
Ability Outcomes
3. Use a range of established techniques to integrate database content into dynamic, usable web applications
4. Write maintainable code for web applications

Assessment Brief

At the end of this document are several scenarios. You have been allocated a specific scenario that you must work with for this assignment.

- See the document *students-scenarios* on Brightspace for your assigned scenario.
- A database for each scenario can also be found on Brightspace. Each database has been populated with some example data.
- You ***MUST*** use your assigned scenario and database as the basis for your assignment work. **Do not make any changes to the database structure**.

Based on your allocated scenario and database, you need to create a PHP web application that demonstrates the following:

Task 1: Browse Functionality (20%)

Depending on your allocated scenario, you are required to implement functionality where a user can browse information from one of the following tables.

1) Huddland Parliament: ***members*** table
2) Kirklees Hotel Finder: ***hotels*** table
3) Queensgate Airlines: ***flights*** table

- Your web application should present a list of items from the database table. This must be dynamically generated from the database.
- Clicking on an item should take the user to a page showing full details for the item. Again, this page must be dynamically generated using database content.

For example, if you have been assigned the Kirklees Hotel Finder scenario you would create a PHP page that showed a list of all the hotels. Clicking on a hotel would present a page that showed the further details for the hotel e.g. the star rating, the price etc.

Successfully completing the above will give you a **C grade**. Attempt the following for additional marks.

- Make effective use of SQL joins to display detailed information for items e.g. if doing the hotel finder scenario information such as the amenities available at the hotel would also be displayed. Displaying information from all relevant tables will result in more marks.
- Instead of simply listing all items from the database table provide a search facility. For example, if doing the hotels scenario users would enter the name of a hotel as a search term, matching hotels would be displayed and the user would click on one of these to get the hotel details. More ambitious examples e.g. using a single page for the form and the results, providing validation and feedback, flexible searching etc. will result in more marks.

Task 2: Create Functionality (20%)

Again, based on your allocated scenario, you are required to implement functionality in your web application where a user can insert a new row into one of the following database tables.

Huddland Parliament: ***members*** table
Kirklees Hotel Finder: ***hotels*** table
Queensgate Airlines: ***flights*** table

- You will need to create an HTML form where users will enter the details of the new item.
- Once submitted this data should be inserted into the database table, and the user should receive confirmation of the new item's details.

For example, if you are doing the Queensgate Airlines scenario you would create a form where users could enter the origin id, destination id, departure date and time etc. for a new flight. After submitting the form, a new row would be added to the flights table and the user would be presented with a confirmation page showing the details of the new flight. The new data would then appear on the browse page.

Successfully completing the above will give you a **C grade**. Attempt the following for additional marks.

- A simple solution to this task would involve users entering ids for the foreign keys e.g. the users would type in the *origin_id* code. For higher marks dynamically generate drop-down menus, checkboxes, radio buttons etc. for related data.
- Validation of user input using PHP (not HTML attributes or JavaScript). User input should validated and sanitised. This could involve simple tests such as checking if fields have been completed or more complex checks e.g. checking you have the correct number of crew members. The scenario descriptions provide more detail on some of the validation requirements, but please ask tutors for further clarification.

Task 3: Authentication and Authorisation (20%)

For task 3 you are required to add authentication to your web application i.e. users should only be able to access the site if they enter correct details at a login screen. User details, including usernames and passwords have been provided for you in the ***users*** table. User's email addresses are used as their usernames. If you look in the users table you will see that the passwords have been hashed. The following lists the usernames and passwords of these users:

Username	Password
k.l.hutton@assign3.ac.uk	password
y.miandad@assign3.ac.uk	letmein
s.laxman@assign3.ac.uk	password2

- Your application should authenticate user details against the data in the users table i.e. the login system should be database driven.
- Users should only be able to access the site if they have first logged in.
- You should also provide functionality for logging users out. Once logged out, users should be returned to the login screen.

Successfully completing the above will give you a ***B grade***. Attempt the following for additional marks.

- Implement authorisation. The users table contains a role column. A value of 2 indicates the user is an admin user. Only users with a role of admin should be able to access the create functionality.

Task 4: Writing Maintainable Code (30%)

Consideration will be given to how you have coded your web application. Successfully implementing techniques to address duplicate code (even if this simply means using includes) will earn you a ***D grade***. Higher grades will be awarded to applications that:

- Structure code effectively using MVC principles. This could be through the use of functions or using object-oriented programming.
- Use suitable design patterns to create a robust, maintainable web application.

Task 5: CSS and Design (10%)

Using CSS you should specify the design for the site.

- The pages must be viewable at 1024x768.
- All text should be legible. Avoid 'noisy' background images and make sure there is clear contrast.
- Pages should load quickly. Avoid images/media that are unnecessarily large in file size.
- There should be a cohesive consistent design.
- The site should be easy to use and navigate.

Meeting the above requirements will earn you a **C grade**. To improve the grade attempt to meet these additional requirements:-

- CSS manipulates a wide variety of properties and extensive use of CSS.
- The use of CSS is complex with use of advanced techniques and results in an original, visually impressive design.

Marking Scheme

Task 1: Browse Functionality (20%)

Grade	Descriptor

A++	A truly outstanding piece of work. Meets all the requirements with no room for improvement. The work will show a depth of understanding and typically feature something that is novel and challenging to implement while still in keeping with the assignment requirements.
A+	Fully meets the basic requirements and fully meets **both** of the additional requirements, with no significant room for improvement.
A	Fully meets the basic requirements and fully meets **both** of the additional requirements, with room for improvement.
B	Fully meets the basic requirements and fully meets **one** of the additional requirements. Or fully meets the basic requirements and partially meets **both** of the additional requirements.
C	Meets the basic requirements
D	Shows basic understand but only partially meets the basic requirements.
E/F	Fails to meet the basic requirements.

Task 2: Create Functionality (20%)

Grade	**Descriptor**
A++	A truly outstanding piece of work. Meets all the requirements with no room for improvement. The work will show a depth of understanding and typically feature something that is novel and challenging to implement while still in keeping with the assignment requirements.
A+	Fully meets the basic requirements and fully meets **both** of the additional requirements, with no significant room for improvement.
A	Fully meets the basic requirements and fully meets **both** of the additional requirements, with room for improvement.
B	Fully meets the basic requirements and fully meets **one** of the additional requirements. Or fully meets the basic requirements and partially meets **both** of the additional requirements.
C	Meets the basic requirements.
D	Shows basic understand but only partially meets the basic requirements.
E/F	Fails to meet the basic requirements.

Task 3: Authentication and Authorisation (20%)

Grade	**Descriptor**
A++	A truly outstanding piece of work. Meets all the requirements with no room for improvement. The work will show a depth of understanding and typically feature something that is novel and challenging to implement while still in keeping with the assignment requirements.
A+	Fully meets the basic requirements and **both** of the additional requirements, with no significant room for improvement.
A	Fully meets the basic requirements and **both** of the additional requirements, with room for improvement.
B	Fully meets the basic requirements
C/D	Partially meets the basic requirements e.g. login system isn't database driven.
E/F	Fails to meet the basic specified requirements.

Task 4: Writing Maintainable Code (30%)

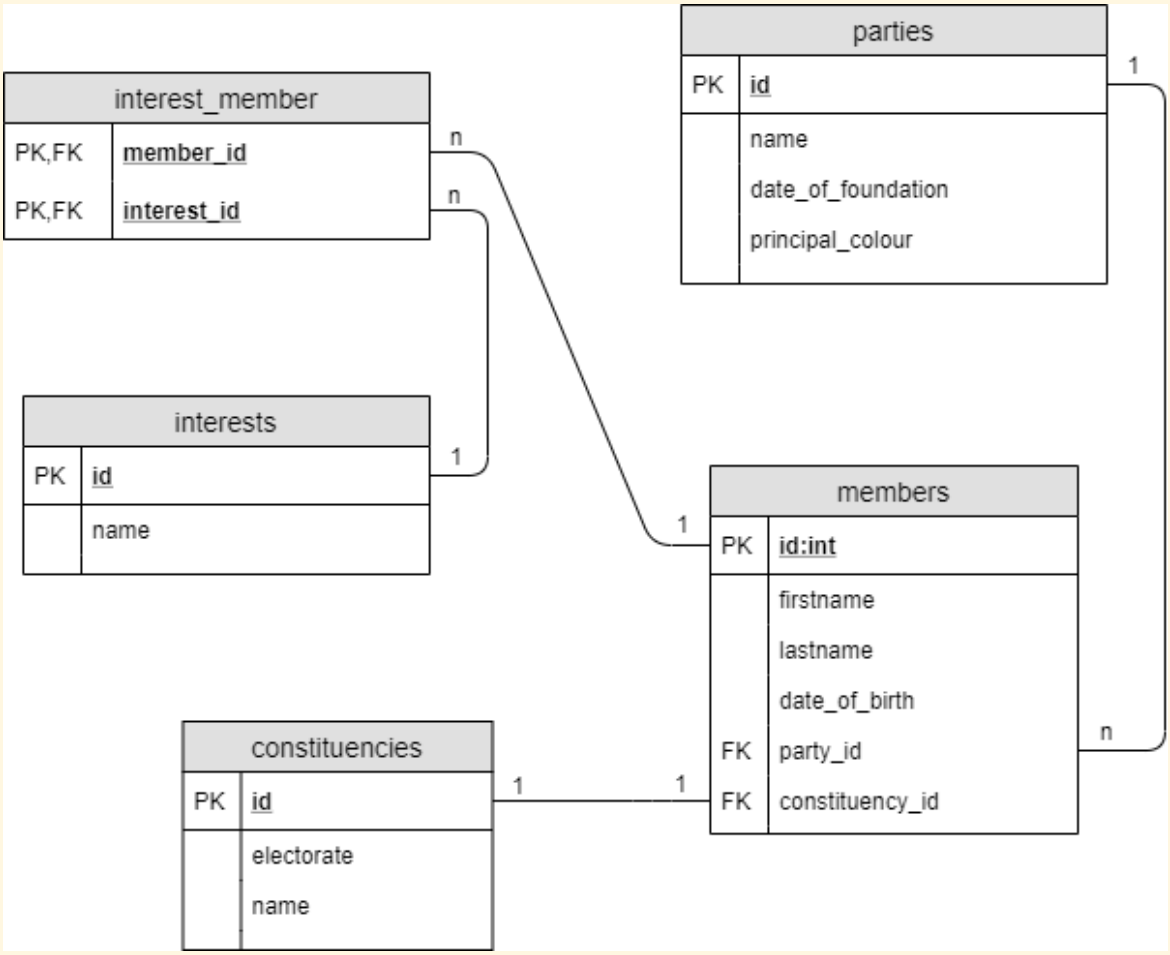
Grade	**Descriptor**
A++	A truly outstanding piece of work. Meets all the requirements with no room for improvement. The work will show a depth of understanding and typically feature something that is novel and challenging to implement while still in keeping with the assignment requirements.
A+	Fully meets the basic requirements and fully meets **both** of the additional requirements, with no significant room for improvement.
A	Fully meets the basic requirements and fully meets **both** of the additional requirements, with room for improvement.
B	Fully meets the basic requirements and fully meets **one** of the additional requirements. Or fully meets the basic requirements and partially meets **both** of the additional requirements.
C	Meets the basic requirements and partially meets **one** of the additional requirements.
D	Meets the basic requirements.
E/F	Fails to meet the basic requirements.

Task 5: CSS and Design (10%)

Grade	Descriptor
A+	A truly outstanding piece of work. Meets all the requirements with no room for improvement. The work will show a depth of understanding and typically feature something that is novel and challenging to implement while still in keeping with the assignment requirements.
A	Fully meets the basic requirements and **both** of the additional requirements
B	Fully meets the basic requirements and **one** of the additional requirements with minor/no room for improvement
C	Meets the basic requirements with minor/no room for improvement
D	Meets most of the basic requirements but with significant room for improvement
E/F	Fails to meet most of the specified requirements.

Scenario 1: Huddland Parliament

The country of Huddland has a parliamentary system similar to the UK. A data model for the Huddland parliament is shown below:



- The parliament is made up of Members of Parliament (MPs). MP details are stored in the members table.
- Each MP represents a single constituency (district of the country), and each constituency is represented by a single MP.
- Each MP belongs to a single political party.
- MPs each have a number of political interests e.g. health and social care, transport, international development etc.

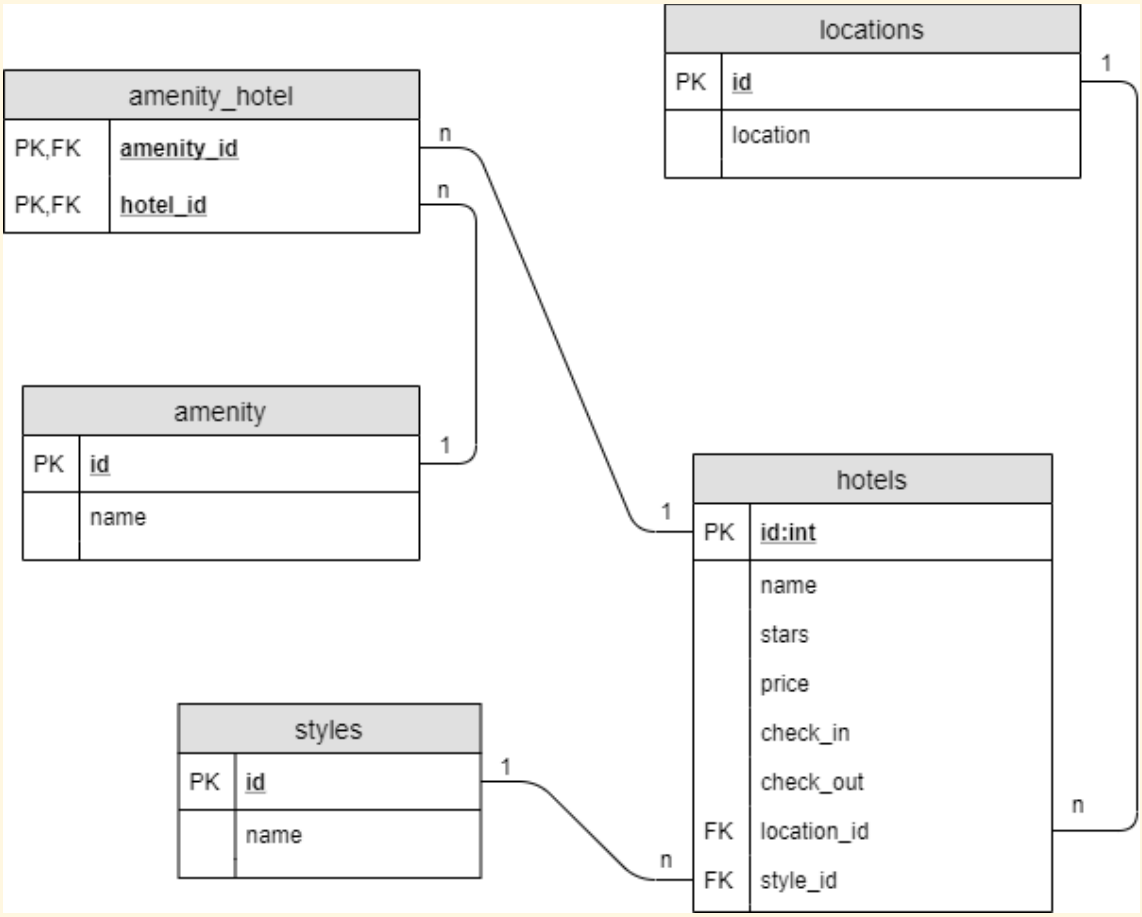
Here are some things to consider when implementing the create functionality

- MPs must be at least eighteen years in order to be elected to parliament
- There is a one-to-one relationship between constituency and member i.e. the same constituency can't be represented by more than a single MP. A single MP can only represent one constituency.
- MPs must have at least one political interest.

The Huddland Parliament database can be downloaded from BrightSpace.

Scenario 2: Kirklees Hotel Finder

After noticing recent interest in the Huddersfield area as a holiday destination, the local authority have decided to develop a web application that will allow users to search for hotels in the Kirklees area. A data model for the Kirklees Hotel Finder is shown below:



- Hotels have a name, a star rating (1-5), price etc.
- Hotels can have a number of amenities e.g. swimming pool, restaurant, gym etc.
- A hotel is categorised under specific style e.g. luxury, budget, business etc.
- A hotel is based in a location within Kirklees e.g. Batley, Mirfield, Colne Valley etc.

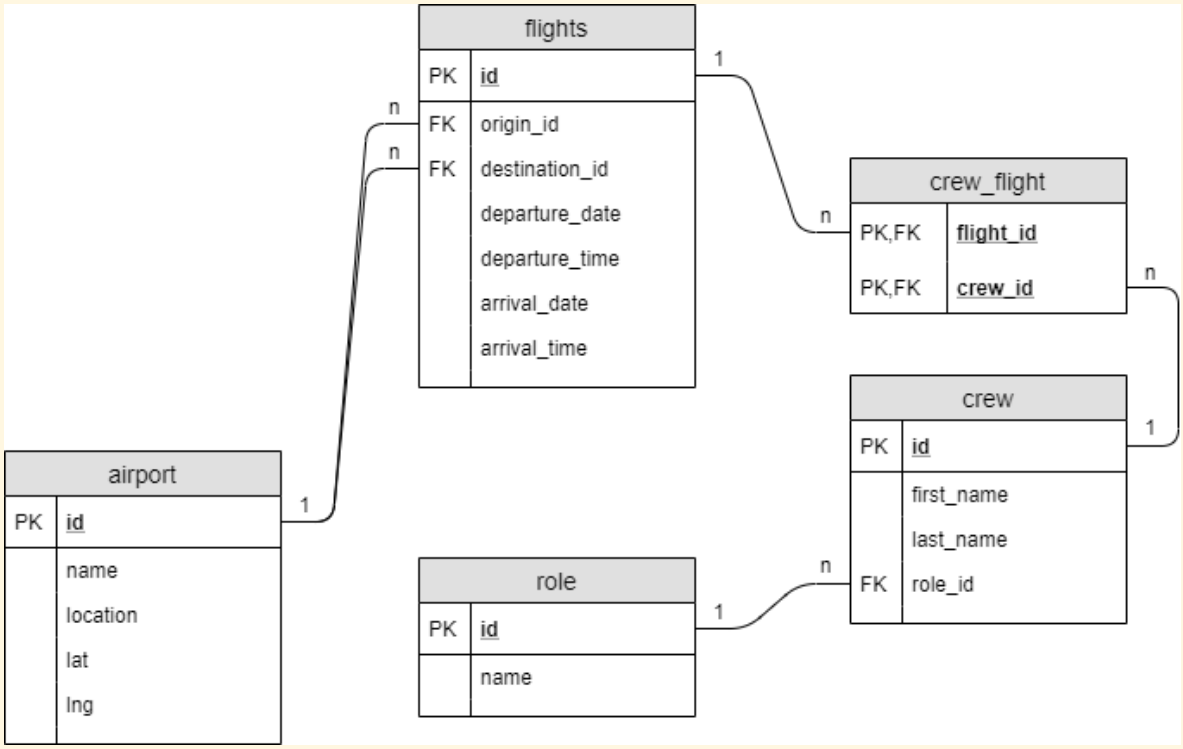
Here are some things to consider when implementing the create functionality:

- A hotel's star rating must be between one and five.
- The check-out time (the time a guest needs to leave on the day of departure) must be at least two hours before the check-in time (the earliest a guest can use their room on the day of arrival).

The Kirklees Hotel Finder database can be downloaded from Brightspace.

Scenario 3: Queensgate Airlines

Queensgate Airlines operate flights between European airports. They store information about their flights in a database. A data model for this database is shown below.



- A flight has an origin (the airport in leaves from) and a destination (the airport it flies to), it also has a departure date and time and arrival date and time.
- Airports have a name e.g. Liverpool John Lennon, and a location (the actual city or region where they are based) e.g. Liverpool. The location of the airport in the form of a latitude and longitude is also stored.
- Each flight is staffed by a number of crew members. Crew members have different roles e.g. pilot, flight attendant.

Here are some things to consider when implementing the create functionality:

- The departure date and time must be before the arrival date and time.
- Each flight requires four crew members. A pilot and three flight attendants.

The Queensgate Airlines database can be downloaded from Brightspace.

Note that the roles table in this scenario **shouldn't** be used when implementing authorisation. There is a separate *role* column in the users table that should be used for this purpose.