

## School of Computing: Assessment brief

<b>Module title</b>	Web Services and Web Data
<b>Module code</b>	COMP3011/XJCO3011
<b>Assignment title</b>	Flight Search and Booking System
<b>Assignment type and description</b>	<p>This assignment is based on a mini project. The project work will be divided into two parts corresponding to the two coursework elements of this module:</p> <p><b>Part 1 (Coursework 1): Setting out technical specifications and designing system architecture.</b>          You will start working on this coursework on 20/2/2023, and the submission deadline is 27/3/2023</p> <p><b>Part 2 (Coursework 2): Implementing system components</b>          You will start working on this coursework on 27/3/2023, and the submission deadline is 12/5/2023</p> <p>This document provides guidance for both Coursework 1 and Coursework 2.</p> <p>In Coursework 1, you will be working within a team of other students to determine the technical specifications and develop the design of a distributed web-based system for finding and booking flights.</p> <p>The system comprises, but is not necessarily restricted to, two main categories of web services' providers:</p> <ul style="list-style-type: none"> <li>• Airlines (think for example of British Airways or Turkish Airlines)</li> <li>• Payment providers (think for example of PayPal or Klarna).</li> </ul> <p>In addition to web services' providers, flight aggregator applications (think for example of Skyscanner), interact with the web services to enable users – among other things – to:</p> <ul style="list-style-type: none"> <li>• find the most suitable flights for a journey,</li> <li>• book seats on certain flights,</li> <li>• use payment services to pay for bookings,</li> <li>• receive electronic tickets, and</li> <li>• manage their bookings.</li> </ul> <p>In Coursework 2, each member of the team will be required to implement one of three possible components:</p> <ul style="list-style-type: none"> <li>• An airline web service, or</li> </ul>

	<ul style="list-style-type: none"> <li>• a payment provider web service, or</li> <li>• a flight aggregator (booking engine) application.</li> </ul> <p>In addition to the above, you may decide as a result of working on Coursework 1 that other supporting services may be needed and some members of the team should implement these in Coursework 2.</p>
<b>Rationale</b>	<p>Coursework 1 will give you the opportunity to develop various skills needed when working in a real engineering and business environment such as contributing ideas in committee meetings, talking to and negotiating with stakeholders, understanding and analyzing real business situations (system analysis), as well as setting out detailed requirements and specifications of technical systems.</p> <p>In Coursework 2, you will develop your skills in coding and deploying RESTful web services which have become increasingly important as more and more systems are using them.</p>
<b>Deliverables</b>	<p>At the end of Coursework 1 each team will submit a group report containing:</p> <ol style="list-style-type: none"> <li>1. Detailed analysis of the business situation (the flight search and booking process).</li> <li>2. Detailed description of the working principle of the proposed web-based system.</li> <li>3. The architecture of the system and the interaction between its sub-systems (components).</li> <li>4. Detailed design of all databases maintained by various services.</li> <li>5. Detailed technical specifications of the web services' APIs, as well as the specifications of the client applications.</li> <li>6. An appendix containing brief minutes of each team meeting, and the main outcomes of the meeting. The minutes of every meeting should include a list of members who attended and those who were absent.</li> </ol> <p>The report should not exceed 15 A4 pages when typeset in single column, using Times New Roman font size 11 and single line spacing. The page count includes all figures and diagrams but excludes the cover page, the table of contents' page, and the appendices.</p> <p>At the end of Coursework 2, each student should upload their own code base to Minerva, and if they have implemented a web service, the service should be deployed to Pythonanywhere.com.</p>
<b>Weighting</b>	<p>Coursework 1 is worth 30% of the total module mark.</p> <p>Coursework 2 is worth 30% of the total module mark as well.</p>
<b>Submission deadline</b>	<p>The submission deadline for Coursework 1 is Monday 27/3/2023.</p> <p>The submission deadline for Coursework 2 is Friday 12/5/2023.</p>
<b>Submission method</b>	<p>The deliverables for both coursework should be submitted electronically to Minerva.</p>

<b>Feedback provision</b>	Online via Minerva and face to face in lab sessions.
<b>Learning outcomes assessed</b>	<p>By doing Coursework 1 you will:</p> <p>Learn how to analyse business and technical processes and fully document how they are performed.</p> <p>Learn how to set out the requirements and technical specifications of RESTful web services.</p> <p>Learn how to develop the architecture and the process flow of web-based systems.</p> <p>Learn how to explore alternative solution methods and make justifiable design decisions.</p> <p>Develop your personal skills of working in a team representing different business and technical concerns. This includes, but is not limited to, cooperating, communicating and negotiating with other team members as well as collectively developing technical specifications and system designs.</p> <p>Develop your skills in documenting the requirements, specifications and architecture of distributed software systems.</p> <p>By doing Coursework 2 you will:</p> <p>Develop your skills in programming, deploying, and testing RESTful web services using state of the art platforms and programming languages.</p> <p>Develop your skills in coordinating the development of web services with partners and stakeholders.</p>
<b>Module lead</b>	Dr Ammar Alsalka
<b>Other Staff contact</b>	

## 1. Assignment guidance

To promote their business and to increase the chances of customers finding suitable flights, a group of airlines decided to form an alliance (consortium). One of the aims of this alliance is to develop a distributed web-based system that allows flight aggregator applications (booking engines) to find the most suitable flights for a customer, book seats on a flight, and pay for this online. For this purpose, the alliance also partnered with a number of payment service providers to facilitate online payment for the flights.

In Coursework 1, you will work within a team of other students to develop the requirements, design, and technical specifications of a distributed web-based system comprising both airlines and payment service providers. The system should also include client applications that communicate with both airlines and payment providers to facilitate the process of finding and booking flights.

Each member of the team will represent a particular business, and later will, in Coursework 2, develop the working code of the corresponding web service or client application. Three types of businesses will be equally represented in each team:

- Airline companies.
- Payment providers.
- Flight aggregator developers.

In addition to the above three businesses, you may decide that additional supporting businesses may be needed (e.g. a directory service).

Each team will comprise about 12 students equally representing the above three businesses, i.e. four students will be wearing the hat of four different airlines, four students will be wearing the hat of four different payment services, and four will be representing four different developers of flight aggregators. If additional common services are deemed necessary, one or two students should independently work on these.

You will use the weekly lab sessions of this module to meet up with your team members to discuss and elaborate the system requirements and its architecture. There will be five lab sessions, hence five meetings, during the period of Coursework 1 (from Week 17 commencing 20/2/23 to Week 21 commencing 20/3/23). You should take brief minutes of each meeting (including list of attendees) and document the outcome of the meeting. Each team should appoint a team leader and a minutes' taker (this could be made to rotate).

The following simplifications will be assumed:

- Customers can only search for and book one way flights.
- Airlines do not require user authentication or logging in to allow users to find or book flights.
- The flight aggregation software is command line driven. It is up to implementers of the applications to decide on the names of the commands, their syntax, and the output presentation.

To allow meaningful comparisons between various implementations and simplify the marking process, the following constraints should be adhered to:

- All web services are to adopt RESTful design constraints.
- Implementation of the web services should use the DJANGO development platform (<https://www.djangoproject.com/>) and deployed on Pythonanywhere (<https://www.pythonanywhere.com/>). Further guidance and support on DJANGO and Pythonanywhere will be provided in lectures and lab sessions.

- All client software (flight aggregators) should be implemented in Python 3.x
- For composing requests, client software must use the Requests library (<https://pypi.org/project/requests/>)

## 2. Assessment tasks

### Coursework 1:

1. Examine websites of various airlines, and carefully analyse the processes of finding and booking suitable flights (please be careful not to accidentally pay for any flight).
2. Document the sequence of the above processes from start to finish and use appropriate representation methods such as flow charts and use-case diagrams. Also, as part of this activity identify all relevant actors (e.g. servers, human users, client software, etc) involved in the above processes, the role of each actor in a process, and the nature of the data exchanged between actors at each stage of a process.
3. Design and document appropriate and efficient database models for each web service in the system, i.e. for airlines, payment providers, and any additional supporting services (if needed).
4. Accurately and completely specify all API end points that each web service should provide to the rest of the system.
5. Identify the necessary functions and operations of flight aggregator applications that use the services of airlines and payment providers to allow users to find, book, and manage flights.
6. Document all the above in a report that clearly yet briefly presents all essential information.

### Coursework 2:

1. Implement the web service on your local machine and fully test its operation.
2. Once you are satisfied with your service, upload it to python anywhere.
3. Coordinate with other team members to make sure that your service is communicating properly with other services and client applications.
4. Collectively test the entire system with other team members.

## 3. General guidance and study support

For help on writing reports, please see the skills@library web pages:

[https://library.leeds.ac.uk/info/1401/academic\\_skills](https://library.leeds.ac.uk/info/1401/academic_skills)

## 4. Assessment criteria and marking process

Coursework 1 will be marked through the group report and an interview with the team members during lab sessions in teaching weeks 22, 23, and 24 (the exact date of the interview will be made available to each team in due course).

For Coursework 1 all team members will be awarded the same mark. However, if a person was absent for more than 2 of the 5 meetings without mitigating circumstances approved by the School's Student Education Service, they will be awarded a mark of zero in Coursework 1.

For Coursework 2, each student will be awarded different marks based on the completeness and quality of their implementation.

## 5. Presentation and referencing

In any report submitted for this coursework, the quality of written English will be assessed. As a minimum, you must ensure:

- Paragraphs are used
- There are links between and within paragraphs although these may be ineffective at times
- There are (at least) attempts at referencing
- Word choice and grammar do not seriously undermine the meaning and comprehensibility of the argument
- Word choice and grammar are generally appropriate to an academic text

These are pass/ fail criteria. So irrespective of marks awarded elsewhere, if you do not meet these criteria you will fail overall.

Please do NOT fill your report by copying text from online sources as we are only interested to know what you have done yourselves.

## **6. Submission requirements**

For Coursework 1, the group report should be saved as PDF and uploaded to Minerva by the deadline.

For Coursework 2:

If you have developed a web service:

1. Once you are satisfied that your API is working properly on the local host, open a free account and upload the server (API) code to pythonanywhere.com. Detailed instructions on how to upload a Django project to pythonanywhere.com are available here <https://help.pythonanywhere.com/pages/DeployExistingDjangoProject>
2. When you register for a free account at [www.pythonanywhere.com](http://www.pythonanywhere.com), please use your university username when prompted for an account name. Hence, if your university username is sc15xyz, your root domain address would be sc15xyz.pythonanywhere.com. This will make it much easier to know who owns which account during coursework assessment.
3. Make sure that you have properly uploaded your server code to pythonanywhere.com and tested it thoroughly.
4. Nominate me as your teacher on pythonanywhere.com. This will enable me to access and view your API code directly on the website. You can do this from the 'teacher' tab in your 'account' page. The username you have to use is 'ammarsalka'.
5. Using your admin site, add me as another admin user to your web service, use 'ammar' for username and invent a simple password for this account.
6. Prepare a plain text file called readme.txt containing:
  - i. The name of your pythonanywhere domain.
  - ii. The password I have to use to login to my admin account on your service.
  - iii. Any other information I need in order to use your service.
7. Make a copy of your Django project directory (the outer one) and rename it.
8. Write a brief report (up to 5 pages) that clearly explains how you implemented the service.
9. Put the Django project, the readme file, and the report in one directory. The name of this directory should be your university email address (without the @leeds.ac.uk)
10. Compress the directory using Zip, and upload to Minerva.

If you have implemented a client software:

1. Make sure that your client was written in Python 3.x and fully tested on one of our lab machines.
2. Prepare instructions on using the client (like a quick help page with a list of all commands and how to use them), and save into a readme.txt file.
3. Write a brief report (up to 5 pages) that clearly explains how you implemented the application.

4. Put the readme.txt file, your client program, and the report in one directory. The name of this directory should be your university email address (without the @leeds.ac.uk)
5. Compress the directory with Zip, and upload to Minerva.

## **7. Academic misconduct and plagiarism**

Academic integrity means engaging in good academic practice. This involves essential academic skills, such as keeping track of where you find ideas and information and referencing these accurately in your work.

By submitting this assignment you are confirming that the work is a true expression of your own work and ideas and that you have given credit to others where their work has contributed to yours.

For Coursework 1, you are required to work as a team and you are encouraged to exchange ideas, knowledge, and to help each other in every possible way. You are also expected to contribute equally to the writing of the group report.

For Coursework 2, each student is required to work on their own implementation and deployment of their assigned service or client application. You are not allowed to copy any code from colleagues within or outside of your team. However, since you will be implementing a sub-system within a bigger system, you are expected to be talking and coordinating with your team members to ensure that your code is communicating correctly with the rest of the system.

## **8. Assessment/ marking criteria**

The 30 marks of Coursework 1 will be awarded as follows:

Quality, clarity and completeness of the report:	20 marks
Group performance in the interview:	10 marks

For Coursework 2, the 30 marks are awarded as follows:

The software provides all required functionality:	20 marks
The software is efficient, reliable, and of good quality:	10 marks