

UCL Department of Computer Science

Coursework brief

This document explains the arrangements for the coursework, a group project in which you will design and build **an event management system**. It doesn't matter what kind of events your system is used for; it could be short courses or perhaps musical events, short talks or sports events.

Through this coursework you will develop your knowledge of web technologies and your skills with database development methods. This project runs through the remainder of the term and counts for 65% of the marks available for the module.

The lab sessions can be used to work on your projects and discuss the work with teaching assistants. The project is self-directed in the sense that you will decide how you will achieve the requirements in the design brief below. You will likely need to investigate additional techniques needed to implement your systems.

The coursework must be performed in groups of three. Each group will build a functioning web-based system that achieves the capability requirements set out below in the Design Brief. You should build your systems using a WAMP, MAMP or LAMP (xAMP) server on your local machine(s). xAMP provides web server services on localhost under Windows, Mac or Linux using Apache, MySQL and PHP. You should already have installed this software and got it working.

Your web-based system should be hosted on server running a LAMP stack. You do not need to use any proprietary development frameworks. You will need to populate your database with illustrative data. You should translate the requirements as you interpret them into:

- A set of dynamic web pages written using HTML5 and CSS3 and JavaScript.
- A set of web services written in PHP
- A database created using MySQL

You will decide what web pages your system will have, what web services your system will provide, what data should be stored and the transactions that those data should support. You must create an entity relationship diagram to fully represent the data of interest and their relationships and attributes. You should translate that diagram systematically into a database schema defining the tables and relationships for your database. You should show that the database is in third normal form.

Submission requirements

The deliverables are a video of a demonstration of your working system, a design report and your source code. The video demonstration should be about 5 to 8 minutes long with voice-over narration. You can use free screen recorder tools such as Jing or CamStudio to make your video. The video must show explicitly how each capability listed in the design brief below is achieved in your system. It must go through each capability in the order in which it is listed in the Design Brief. The narration must refer to the capability and preferably the video should flash back to the visual list. With a spoken narration

the video must show each capability you have achieved either through interaction with the user interface of your system.

Videos should be uploaded to Youtube and made unlisted. You need to submit your design report (a single pdf file only) which should contain the link to your video.

The design report should contain:

1. A URL for your Youtube video
2. A description of the architecture of your web application explaining how it aligns with the Model, View, Controller pattern. You should explain:
 - The logic you have embedded in the client-side of your web application such as form validation and the dynamic generation of content.
 - The web services that support your application including detail about the format of information when it is transferred between the client and server.
3. A description of your database design and SQL queries including:
 - Your entity relationship diagram.
 - A listing of your database schema with an explanation of how it corresponds to the ER diagram.
 - A listing and explanation of your database queries (i.e., in relation to the relevant event management capability).

As well as the design report, you should submit an archive containing your source code files (HTML CSS, PHP etc.). **Please only submit source code**, no binary data files or database dumps. Please put your group number in the zip file name. Alternatively, you may submit a link to a Github repository. You will be asked to provide us with access to your web application at some point. You will be contacted and provided with information about a date and time for this. There is no need to have your application accessible at times other than the date and time specified.

Assessment

Your project work will be marked against the rubric visible at the upload link. Marks for your project work will be awarded first for the capabilities (ie: functional requirements) your system achieves, and second, for evidence of the design process you followed. You should show that you have thought about the structure of your web application, considering where the logic for the various elements of the application should reside. You should also demonstrate that you have thought about how the functionality should be broken down into various components of the application. The members of a group will receive the same mark.

Design brief

Your event management systems should have as many of the capabilities listed below as you are able to develop. Marks will be awarded for each capability achieved and will take account of the quality and completeness of the design and implementation.

The presentation design of your user interface and its usability is not being assessed and you should avoid spending time on these. You may use an HTML, CSS, and JS framework like Bootstrap if you wish although no extra marks will be given for this if you do.

Our concern is with the design and implementation of:

- The application logic including interactive functions like input validation,
- The queries on the database to achieve the system capabilities.

1.	All users can register with the system and create an account. A user must log in to the system before using it. A user can be a host and may set up their own events. A user can also be a participant and book tickets to other people's events.	10
2.	A user may create an event, setting suitable conditions and features of the event including the description, location, date, categorisation, number of tickets available and end date.	10
3.	A user can search the system for categories of events and can browse events within certain categories or timeframes.	10
4.	A user can book a ticket for an event. The system will manage the ticket sales until the set end time. No further ticket sales will be permitted if the event is full or if the end time for ticket sales of that event has been reached.	15
5	Participants can view a list of events they are attending and receive emailed updates on events that are imminent.	10
6.	Users hosting events can view the progress of ticket sales and generate a list of participants for an event.	10
7.	Participants can give feedback and ratings on events after they have happened. Ratings and feedback are visible to other users.	15

The deadline for submission is the end of Monday 9th January; that is the first taught week of next term.

You should submit a draft set of web pages (wireframes) and an entity relationship diagram as a separate submission at the end of the eighth week of term. These materials will be the best view you have at that point and may change later. The material won't be assessed but we will give you formative feedback on it. To create your wireframes you can use a web app like Lucidchart. To create your entity relationship diagram you may use one of the web apps: LucidChart, Vertabelo or the freely distributed MySQL Workbench.