UCL Department of Computer Science

Databases COMP3013/GC06   spring term, 2015

Databases coursework briefing

This document explains the arrangements for the coursework, a group project in which you will design and build a database system. This project runs through most of the term and counts for 30% of the marks available for the course.

Through this coursework you will develop your knowledge of database technologies and your skills using database development methods. Each week (until week 8) we will provide a one hour lecture-format tutorial to explain the technologies and methods you will need to use in your projects. These are followed immediately by lab sessions when groups will meet to work on their project and will be able to discuss their work with staff and demonstrators. The project is self-directed in the sense that you will decide precisely what you will produce beyond the core requirements given here. You will likely need to investigate additional techniques and technologies needed to achieve your design.

The deliverable is a video of a demonstration of your working system. The video should be uploaded to Youtube and with a spoken narration should show the visual display of your system's user interface during interaction with the system. A short supplementary report will also be required along with submission of your code.

The deadline for submission of the completed work is the end of the ninth taught week of term (Friday 20 March, 11pm).

To pass the coursework your system must achieve at least a minimum working function, that is, the primary functional requirements specified below. Marks above the pass mark will be awarded for design ambition, design quality and build quality. Additional features that the system might have are also given below.

The coursework must be performed in groups of three and these must have exclusively postgraduate or undergraduate members. There are significantly higher expectations for the systems that the undergraduate groups will build. Groups must be formed by the end of the first week and any students who are not already in a group by that point will be allocated to a group by the tutor.

Your design work should be focused on the database design. You should translate the requirements as you interpret them into what data should be stored and the transactions that those data should support, how the data will be modified and who will have access to it. 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 check that the database is in BCNF.

You should submit your entity relationship diagram as a separate submission at the end of the second week of term, via moodle. The diagram will be the best view you have at that point, even if it may change later.

You should also develop interaction sketches for the user interface design, showing the facilities that the system will give the user and the implied transactions the user will have with the system.

You should build your demonstrators using WAMPserver. WAMP provides web server services on localhost under Windows using Apache, MySQL and PHP. WAMP is installed on the lab machines we use in the scheduled lab sessions and in the lab tutorials we will be explaining how you can use it. You may not use any proprietary development frameworks. You can install WAMP on your own windows machines or use MAMP under OSX or a version of XAMPP.

Undergraduate groups should host their projects in the cloud on Microsoft Azure. Azure supports PHP with MySQL integrated with version control services such as Github. This allows groups to develop their systems on their local or lab machines and then push their code into the cloud to run on Azure. The Friday lab tutorial lectures given by Dr Roberts will cover Cloud and Azure. Postgraduate groups are also encouraged to host in the cloud to gain experience of hosting a real service and managing a cloud-based MySQL instance.

Submission requirements and assessment

You will be assessed on the basis of the video demonstration of your system. The demonstration should be 5 to 8 minutes long with voice-over narration. You can use a free trial of Camtasia to make your video http://www.techsmith.com/camtasia.html

Videos should be uploaded to Youtube. Make sure the link is clearly given in your report. The report will be used to inform the assessment of the video.

A pass mark will be awarded for all demonstrators that demonstrate a functional implementation of the core functions. Marks will be awarded for the functional sophistication of the system, the design quality and the build

quality. The presentation design of your user interface and its usability is not being assessed and you should avoid spending time on such aspects; our concern is with the development of the database and the functionality that it makes possible.

The members of each group will receive the same mark.

The supplementary report, submitted via moodle, should contain:

1. URL for the Youtube video
2. Your entity relationship diagram, giving any assumptions that it makes about the processes that use the data.
3. A listing of your database schema with an explanation of how it translates the ER diagram.
4. An analysis showing that the database schema is in third normal form.
5. A listing and explanation of database queries.
6. You must submit a copy of your code via moodle and provide us with access to your system if it is hosted on Azure.

System concept and requirements

The concept for the system is that of a 'virtual learning environment' for peer assessment. Peer assessment is an educational method where students give and receive assessments on each others' work. For this project you should focus on a scenario in which a college-level class studying consumer informatics is divided into twenty groups of three students to each produce a report surveying reviews of the new LG G Flex 2 smartphone. Such a report would summarise the most important reviews (techradar.com, wired.com etc), comparing different comments on the same features of the Flex 2.

Each group makes assessments of the reports created by several other groups and each group receives several assessments of their own report from other groups. The assessments each group provides are gradings (e.g., 1 to 5) against a small number of criteria (e.g., the report has identified the most important review sources, etc). Comments are provided to explain the gradings.

You should develop a learning environment for this peer assessment scenario. The learning environment will be a web facing database system accessed through a browser. Each group will have its own profile and space on the system. Through this space the group will provide access to its own report and obtain access to the reports of other groups; upload or submit its assessments of other groups' reports; receive assessments of its own report from other groups.

The following are minimum features your systems must provide:

- student-users will upload free text reports to their group account Done - Izzatul
- student-users will be able to visually browse the other groups' reports they have been allocated to review Yusuke
- student-users will submit grading assessments and comments on the reports assigned to them Done - Yusuke
- student-users will be able to view assessments on their own reports made by other groups Done - Izzatul
- Student-users will be able to know the ranking of their own aggregated mark within the aggregated marks for all groups. Done - Izzatul

additional features include the following:

- student-users will upload their reports as XML files. The system will extract the data elements from the file and insert them into the database Izzatul / Yusuke
- student-users in a group will be able to discuss submissions with other members of their group via a forum which is threaded, browseable and searchable Yusuke
- student-users will be able to see the aggregate assessment grades received by each group that provides a grade on their own submissions ???? Not anonymous?
- administrator-users will have a separate interface through which student registration will be managed and groups defined from the student registration lis Izzatul
- the administrator-user interface will support searching for details of a particular student and browsing of student details Fadhli (search student details & edit function: SELECT & UPDATE statements)
- the administrator-user interface will allow particular groups to be allocated to the peer assessment of particular other groups Izzatul
- the administrator-users will be able to see a list of the groups ranked according with the aggregation of peer assessments on their submissions Izzatul