

Coursework briefing

This document explains the arrangements for the coursework, a group project in which you will develop a business intelligence service for ebay users. Through this coursework you will develop your knowledge of database technologies and your skills with database development methods. The project runs through the term with a final demonstration to tutors in the last week of term and submission of a minimal joint report.

The project will be supported by one-hour lecture-format tutorials (provided by Dr Roberts) to explain technologies and methods you may use in your projects. These tutorial sessions support lab sessions when groups will meet to work on their project and discuss their progress with a teaching assistant. The project is self-directed in the sense that you will decide how you will achieve the requirements that we set for you in the design brief below. You will likely need to investigate additional techniques needed to implement your systems.

The coursework should be performed in groups of three and groups must be self-formed by the end of the first week using the [wiki](#) on Moodle.

Design brief

Each group will create a service of their own design for users of ebay. The service should provide users with advice about marketplace and user activity on ebay, and about the state of selected auctions. We suggest that you develop a service to provide an analysis of the selling prices of certain types of camera and whether current live auctions for particular cameras are achieving their ebay market value. The service might provide advice on when and what to bid in an auction, for example a camera that is currently underbid relative to the price it might be expected to achieve.

The service will need to use the ebay production environment APIs through which it will extract data about items and auctions which it will store in its own relational database. The service should be hosted in Microsoft Azure and interrogate the stored data to provide advice to users about the expected value of selected items and about bidding in auctions for selected items.

Each group will need to register with the ebay developer network <https://developer.ebay.com/api-docs/static/creating-edp-account.html> ebay provides several sets of APIs, all in RESTful form; <https://go.developer.ebay.com/api-documentation>. Calls to an API typically return a data payload in JSON format. You will likely need to use more than one API, such as the Feed API, the Finding API and the analytics API. Shredding of the ebay payloads to populate the database can take advantage of an ebay SDK <https://go.developer.ebay.com/ebay-sdks>.

Groups should build their service using the facilities available on Azure. Each group will have an Azure account with a fixed amount of credit, which will need to be used carefully. A core component will be the database, which must be a SQL database, and should be of a good level of sophistication. Groups will analyse the data needs of their system by creating an entity relationship diagram (ERD). They will translate the ERD into the design of a SQL database schema needed to store the data their chosen service will provide. The schema must be in third normal form. You may expect your schema to have eight or more tables.

A critical constraint is that you must design and implement the SQL database directly following the ERD, along with all the required SQL code to access the database. You cannot use automated tools to create the database using a specification language, object-relational or similar mapping frameworks, or anything that generates SQL code automatically. You can write SQL scripts to create and configure your database (this is standard SQL code), and help manage the database.

You can use an Azure SQL service, such as SQL Database or Azure Database for MySQL. Alternatively you can configure your own virtual machine(s) or use Docker to provide a SQL database of your choice. You have the choice of which other web, front-end and back-end technologies you use, but bear in mind that the primary goal is the design and implementation of a robust database so don't get side-tracked by trying to use the latest and greatest application frameworks!

Your group should have a GitHub repository, used properly, with regular commits. It should be possible to build and deploy your application by pulling the latest version from the repository.

A well-designed application will have a gateway machine connected to the internet, with all the other components and services running in a virtual private network. The security of the application should be addressed and you should identify and implement strategies to keep it secure. For example, as you are using SQL the application should not be susceptible to SQL injection attacks.

Your application design should address scalability, so the application could potentially scale-up to thousands of simultaneous users. This might mean having multiple database servers, using caches, micro-services, and other strategies. However, you should balance scalability against cost, in particular as you have limited Azure credit. Identify scaling strategies that minimise cost while still giving flexibility, and how to control costs. It is better to have an application that has clearly defined upper performance or scalability limits than one which simply consumes resources in an uncontrolled way. Beware that storage can be expensive, so think carefully about how you store data.

Your application should have a presentable user interface, but as noted earlier the focus of the work and our assessment of it is the SQL-based storage, acquisition and management of data having identified what data is needed to provide the service you have defined, and how well your service performs in the analysis and use of the data.

Project Deliverables

We will arrange a 15 minute demonstration of each group's system in the final week of term (during an extended practicals session). Each group should submit a brief (2 page max) document providing a description of the service you have created and giving the location for accessing your work on Azure. The report must include as an Appendix (i.e., additional to the 2 pages) an entity relationship diagram for the database you have created, a listing of the database schema and an explanation of why the database is in third normal form. You should also supply the URL to your GitHub repository, so that both the final version of the application and the history (commits) can be viewed. You can make the repository private provided that the markers are given access.

Assessment

Marks will be awarded for the extent and sophistication of what you use and create and will be judged primarily on what we see in the demonstration. Marks will be awarded in these categories:

- 20% for use of ebay tools and services, in both extent and depth including the use of the ebay APIs and SDK as well as the breadth of ebay data exploited.
- 30% for database design and implementation including the definition of the normalised database schema and SQL queries for creating the auctions and items reports provided by your Service.
- 30% for use of Azure-based data management facilities for managing the hosted Service, more marks will be awarded for the effectiveness of Azure facilities exploited.
- 20% for the effectiveness of the Service created. Marks will be awarded for the likely value to users of the Service and its likely operational effectiveness, including the relative scalability and how security has been addressed.

The assessment of the work will be provided using Moodle's rubric marks tool showing how and why marks are awarded in each category. In addition to these marks, a summary comment on the project will be provided.