School of Science
and Engineering
University of Dundee

AC32006/AC52001 Assignment 2 – Database Implementation

Deadline for Submission: Friday of Week 11 at 5pm (1st December 2023)

Hand in Method: MyDundee - You should upload a **single PDF document** containing all the components of your report. Failure to upload as a single file will reduce your overall mark. Your database (and website if used) should be left "live" on the AWS servers until the end of January 2024 for marking.

Date Feedback will be Received by: This will be received within the University's 3 week policy (allowing for Christmas vacation).

Penalty for Late Submission: One grade point per day late (meaning if a submission is one day late and marked as a C2 it will receive a C3 grade). A day is defined as each 24 hour period following the submission deadline including weekends and holidays. Assignments submitted more than 5 days after the agreed deadline will receive a zero mark (AB).

Percentage of Module: This assignment is worth 37% of AC32006 (BSc students) / 27% of AC52001 (MSc students).

Overview of Assignment

Brief

The assignment is designed to give you practice in the preparation of the database that you designed in Assessment 1; it is to be conducted in the same teams as Assessment 1.

BSc students are expected each to spend approximately 30 hours working on this assignment, which is worth 37% of your total grade for this module. MSc students are expected each to spend approximately 22 hours working on this assignment, which is worth 27% of your total grade for this module.

Background

This assignment follows on from the logical design performed in Assessment 1.

In this assignment you are to develop an **application** supported by a **relational database** implemented in **MySQL** which corresponds to your team's submitted design. Your application may be written in *any language* you chose for *whichever platform* you consider appropriate; however:

- your MySQL database must be hosted on AWS **servers**
- it is **recommended** that you build a **web application** using **PHP** and host everything on AWS servers **from the start of development** (AWS runs **PHP 8.2**) but you are free to develop your front end in any way that you see fit
- your MySQL should demonstrate standard **CRUDS** functionality appropriate to each of your users/user groups. Note the data displayed can be the result of a **query**, it does not have to be drawn from a *single* table in your database.

- in addition to implementing views, your MySQL should demonstrate *at least three* **advanced SQL** queries using the techniques as covered in the 3rd SQL video
- you are free to use any 3rd party tools, web templates and PHP code that you think is appropriate to your application; however, any 3rd party items used **must be referenced** in your report to avoid any potential for plagiarism

In particular, access *should* be managed by providing a**ccess privileges through views**. You should ensure that the different users have access to the data through views, *not* by directly reading and writing to the base data tables. The selection of user views, the queries that they support and other features should be **consistent** with the description of the company that you developed in Assessment 1.

Fitness for purpose, the **amount and quality of appropriate functionality** and the **attractiveness of the application** are your **key objectives**.

Deliverables

Your assessment will be based on:

- a **report** (max. 10 pages (to include diagrams but *not* including any cover pages or references)). The **first page** of your report must include connection details (e.g. **URL(s)**) and **login details for all users** of your application and any other details that may be required to access the system (including direct database access via MySQL Workbench); *systems which cannot be easily accessed will not be marked*. Please use **short usernames** and **short passwords** (and ideally the **same password** for all your users). The report should be prepared for the company's CEO and summarise your approach to the development of your system and include a **user guide** with separate sections for **each type of user** of your system, together with a short technical annex (for the company's IT department) which provides **examples** of the SQL statements used to configure and administer the database, including managed access for the different users, indication of indexes and stored procedures used, and the **advanced queries** implemented. Note that a table of contents is **not** required.
- your **MySQL database** and the **front end used to access it:**
    - **structure, features, quality** of content (*amount* of content is less important, as long as what is there allows reasonable demonstration of any search and filtering features) of the database
    - **appearance** and **functionality** of the front end
- your system must be left accessible and functional on the AWS servers until the end of January 2024 for marking purposes.

The special coversheet for this assignment (upon which your team number, names of team members and the numbers of pages in total should be clearly shown) should be included. Keep a copy of what you submit in case there are problems with your submission.

Submission & Assessment

Your report (with cover sheet) should be a **single PDF document** submitted via MyDundee; your **team number** should be included in the filename. You will also be required to complete a **peer review** (via the online peer review system) to ensure equitable distribution of marks.

Submission deadline is 5pm on Friday 1st December 2023 (Friday of Week 11).

Constraints:

Your report should be a **single PDF document** and your MySQL database and front end must be left accessible and functional on the AWS servers until the end of January 2024 for marking purposes.

Marking Scheme

Marks will be allocated as follows:

- Report (content and layout) – **20%**
- Database (structure / content / features) – **40%**
- Front end fitness for purpose - **15%**
- Front end functionality - **15%**
- Front end neatness / attractiveness - **10%**

A Comprehensive database features, structure and comprehensive content. Some additional database maintenance functions (for example transaction control). Excellent use of SQL and complete database structure with excellent use of views. Attractive user interface. No omissions from the brief.

B Good database features, structure and substantial content. Good use of SQL and complete database structure with good use of views. Good user interface. There may be minor omissions from the brief.

C Adequate database features, structure and adequate content. Adequate use of SQL and limited use of views. Adequate user interface. There may be some omissions from the brief.

D Minimal database features, structure and minimal content. Limited use of SQL and limited or no use of views. Minimal user interface. There may be some omissions from the brief.

Fail Poor database features, structure and inadequate content. Poor use of SQL and poor or no use of views. Poor user interface. There may be major omissions from the brief.