Systems Design & Security: Team Project

Overview

This project will form 50% of the assessment for COM2008/COM3008. The project will be completed in teams of four, randomly created from self-submitted pairs of students. The project brief will be released Monday week 4. Teams will hand in their reports and software in Friday week 10 (one report and software bundle per team). This will be worth 40% of the grade. There will be a further testing stage, completed individually in week 11. This will be worth the last 10%.

Submission details

- 1 team report (typically 10-15pp) to be submitted as a paper copy, with a coversheet having barcodes of all team-members, to the DCS assignment mailbox (opposite DCS reception), by 15:00 Friday week 10.
- 1 electronic backup of the team report to be submitted as PDF to MOLE (only one per team), by 15:00 Friday week 10.
- 1 zipped bundle of your software system to be submitted to MOLE (only one per team), by 15:00 Friday week 10.

Individual testing

Students will be allocated individually to test another team's system in week 11. You will arrange to meet the team you are testing, and run through a series of prescribed tests (to be released).

- Submit your online testing web-form any time between Monday-Friday in week 11, but no later than 17:00 Friday week 11.

Late submissions

Standard lateness penalties will apply (deducting 5% of your actual score per working day).

Objective

The objective of this project is to create a software system to meet the needs of an organisation described below. The system will be implemented in Java and will include a database in MySQL and a user interface in Java Swing. You will develop a number of UML models during your analysis, which will form part of your report. The design of the database and the user interface should follow directly from these models. The system should be able to perform the requested functions. The system will support different user roles having different access rights, and will be robust to obvious cyber-attacks.

Background Information

Your customer is a small private university which requires an information system to manage the registration and progression of students through their degree programmes. The university has a small number of departments, each offering a limited number of degrees.

Stakeholder roles

The different kinds of stakeholder interacting with the system include administrator, registrar, teacher and student roles. Administrators handle all the business of setting up the system and user accounts. Registrars handle all the business of student registration for degrees and modules. Teachers handle the business of assigning grades for students taking modules and making decisions

about when students may progress to the next level. Students may only see their own record and may not directly update any information.

Business information

The system will store the following information. The university consists of a growing number of departments, each of which has a full name (e.g. “Computer Science”) identified by an abbreviated three-letter code (e.g. “COM”). A department offers a number of degrees, and some degrees are interdisciplinary, offered by two or more departments, but have one lead department.

Each degree has a full name (e.g. “MEng Software Engineering”) and is identified by an abbreviated code (e.g. “COMU03”) based on the lead department’s code, a letter “U” or “P” indicating undergraduate or postgraduate entry, and a unique two-digit serial number. Each degree may, or may not be offered with a year in industry (if so, “with a Year in Industry” will also be added to the name of the degree; and this will have a different code from the non-industry version).

Each degree has a number of levels of study, indicating the difficulty of the subject content studied at that level. Levels have the names “certificate” (level 1), “diploma” (level 2), “bachelors” (level 3) and “masters” (level 4). Students taking a year in industry are said to be on “placement” (level “P”). Levels of study are identified by the single-character code (“1..4” or “P”). In the sequence of levels, the year in industry is always taken in the penultimate (last but one) academic year (so respectively in year 3 of a BSc, or in year 4 of an MEng degree).

Each degree consists of a number of modules, taught at the different levels. Each module may be taught on more than one degree. A module must be approved for a given level of a given degree and may either be specified as core (obligatory) or not (optional) for this degree and level. The same module may be obligatory for some degrees and optional for others; likewise the same module may be approved for level 2 on one degree, but for level 3 on another (interdisciplinary) degree.

Each module has a full name (e.g. “Programming in Java”) and an abbreviated code (e.g. “COM1003”) based on the teaching department’s code and a 4-digit number. Each module carries a number of credits (by default, 20 in levels 1-3, but 15 in level 4; while undergraduate dissertations carry 40 credits, and masters’ dissertations 60 credits) and is taught either in autumn, in spring, over the summer or across the academic year.

Each student at the university has a title (“Mr” or “Ms”), a surname (family name) and forename (one or more given names, stored as a single field). A student is always registered for a particular degree. Upon registration, they are given a unique registration number and email address. The email is a concatenation of the initial letters of their forename(s), their full surname, and a unique distinguishing integer (e.g. “JSmith23”, if there have been 22 previous “J Smith” names), followed by the rest of the university’s address. Each student also has a personal tutor, whose name is stored as a single string (e.g. “Dr Louise Baxter”).

Each academic year, a student must register for a period of study. This is similar to, but independent of the notion of level of study, since it is possible for a student to repeat the same level in a following period of study. Each period of study has a label (sequential letters of the alphabet: “A”, “B”, etc.), a start-date and an end-date; and also refers to a level of study to be undertaken during that period. A period of study is identified uniquely from the label and the student’s registration number.

When a student is registered for a period of study, they must sign up for a suitable collection of modules that are approved for the level of study. Undergraduates sign up for 120 credits, and postgraduates (1-year masters) sign up for 180 credits (which includes a 60-credit summer

dissertation). Students must take all the obligatory modules and may take any selection of optional modules for their level, up to the correct credit total. They may not take fewer or more than the specified total credits in a period of study.

At the end of an academic year, grades are entered for each student and each module taken, relating to the completed period of study. It is possible to have one or two grades recorded: the first grade records the result of the first examination; and the second grade is optionally entered for a resit examination, if the student failed to pass at their first attempt (otherwise this is left blank).

The student is then given an overall grade for the completed period of study, which is the weighted arithmetic mean of their (best) scores for all the modules (viz. weighted by the credit-value of the modules). However, in the calculation of this weighted mean, the scores for any retaken examinations are capped at a maximum of the pass-mark (viz. the actual resit grade is converted into a bare pass-grade, if it is greater than the pass-mark). The pass-mark for modules at levels 1-3 is 40%. The pass-mark for modules at level 4 is 50%.

At the end of a period of study, it is determined whether the student may progress. If a student passes every module and obtains a weighted mean greater than the pass-mark (as above), they pass and may progress. If a student has marginally failed in one module (20 credits for levels 1-3, 15 credits for level 4) with a score no more than 10% below the pass-mark, but has still met the overall average requirement, they obtain a conceded pass and may progress. Otherwise, they fail for this period and may not progress (viz. their average is too low; or they failed one module too badly; or they failed too many modules). This calculated outcome, and mean grade, are added to the record of the completed period of study.

A student who progresses may be registered for the next higher level in the following period, or else they may graduate (if finishing a bachelor's or master's degree). A student who fails may choose to repeat a given level of study just once. In this case, the grades for modules already passed are transferred to the new period and they retake any failed modules (just once), whose grades are capped. However, certain special rules apply in graduating years (see below).

Upon graduation, a student is awarded a degree class according to a weighting of the scores achieved for each level. For 3- and 4-year degrees, the level 1 average does not count towards the degree-class, the level 2 average counts once towards the degree-class and the level 3 (and possibly level 4) averages both count twice towards the final degree-class (viz. have double the weighting of level 2). For 1-year MSc programmes, a classification is given on the basis of the weighted average grade of all modules taken, including the dissertation project. The different degree-class names are specified in the following table:

Weighted mean grade	< 39.5	39.5 – 44.4	44.5 – 49.4	49.5 – 59.4	59.5 – 69.4	>= 69.5
1-year MSc	fail	fail	fail	pass	merit	distinction
BSc, BEng degrees	fail	pass (non-honours)	third class	lower second	upper second	first class
MComp, MEng degrees	fail	fail	fail	lower second	upper second	first class

Grade boundaries and degree classification

Further special rules apply. If an undergraduate student fails their degree at level 3, they may only resit for a pass (non-honours) degree. If they fail a 4-year degree at level 4, then they must immediately graduate with the equivalent bachelor's degree with credits already obtained. If a

postgraduate MSc student fails their MSc dissertation, they may be considered for a PGDip (postgraduate diploma) if they have passed all the taught modules (or nearly passed, using the rules above for a conceded pass). If they have only passed 60 credits of taught modules, they may be considered for a PGCert (postgraduate certificate) instead.

System operations

Administrators perform the following tasks:

- add and remove user accounts and passwords, granting suitable privileges to users to perform only the designated tasks for their role;
- add and remove university departments from the system (as the university grows, more departments will be added);
- add and remove degree courses, linking them to the one or many departments that teach the degree, indicating the lead department for the degree;
- add and remove modules, linking them to the degrees and levels of study for which they are approved (stating whether core or not);
- display the results of performing the above tasks, so that they can see that the system has responded appropriately to adding or deleting something.

Registrars perform the following tasks:

- add and remove students, linking them to their chosen degree and registering them for their initial period of study (subsequent periods are determined automatically);
- add and drop optional modules on behalf of a student at a given period and level of study (compulsory modules are added automatically);
- check that student registrations are complete and correct, ensuring that students are taking suitable approved modules for the level of study;
- check that that the module credits taken by students in a period of study sum to the correct total (120 for undergraduates, 180 for postgraduates);
- display the results of performing the above tasks, so that they can see that the system has responded appropriately to adding or deleting something.

Teachers perform the following tasks:

- add or update initial grades, resit grades, or repeat year grades for each student taking a module at a given period and level of study;
- calculate the weighted mean grade for a period of study and determine whether a student may progress (according to the rules given above);
- thereby cause a student to be registered for their next period of study, either progressing to the next level, or repeating a level, or graduating, or failing to progress;
- calculate the overall degree result at the end of a student's career and determine what kind of degree class they have obtained (according to the rules above);
- view the current status of any student, showing what that student has achieved at each period and level of study, showing what the outcome was for each level and eventually for the whole degree.

Students perform the following task:

- view their own current status, showing what they have achieved at each period and level of study, showing what the outcome was for each level and eventually for the whole degree.

User interface considerations

The system is a single system used by all four kinds of stakeholder.

- accounts are assigned to each individual user with privileges according to their role;
- when a user logs in, they should be presented with a welcome-screen suitable for their role, that allows them to select only the tasks they should perform;
- a user should only be able to access or view data that is relevant to the tasks they perform in their role;
- a user should not be able to intervene in tasks carried out by another role (unless both roles perform the same tasks);

Security considerations

The system will support user authentication (through a login and password). The passwords known to the system must be stored securely, such that a hacker could not download and use them. The system will support authorisation (of users to perform specific tasks). The system will be resistant to privilege escalation (obtaining higher authorisation). The system will be resistant to SQL-injection (triggering malicious database updates).

Software System

Your software system should be able to perform the following test-tasks, as evidence that its functionality works as desired:

- Add the four departments Business School (BUS), Computer Science (COM), Psychology (PSY) and Modern Languages (LAN).
- Add the degrees MSc in Business Administration (lead BUS), MEng Software Engineering with a Year in Industry (lead COM), BSc Information Systems (lead COM, partners BUS, LAN), MPsy Cognitive Science (lead PSY, partner COM).
- Add realistic core and optional modules for each of the above degrees (take inspiration from the Sheffield University module-guide), ensuring that modules are supplied by all relevant partner-departments; the MSc degree will be all-core, but other degrees will have 20cr free choice at level 1, all-core at level 2, and 40cr free choice at levels 3 and 4.
- Register a student for each of the above degrees, and select both suitable and unsuitable options for their free-choice modules, showing how your system prevents administrators from picking the wrong modules and checks that a student's credit-totals are correct.
- Progress these students through the levels, such that
    - the MSc student gets a conceded pass on taught modules, eventually passing MSc with merit;
    - the MEng student passes through all levels, including the year in industry, getting a 1st class result;
    - the BSc student takes resits at level 1 and passes, but fails at level 2 resits, and also after repeating level 2, so is prevented from progressing;
    - the MPsy student passes through three levels, but fails catastrophically at level 4, so is graduated with a BSc class 2/i instead.

In the testing-stage, we will check whether your system can perform these operations.

Final Team Report

The main purpose of the team report is to show your design process, leading to the implemented system, which will be handed in separately and tested. The data capture and data normalisation stage are especially important and should be done accurately, reflecting the background information exactly, and not contain extraneous material. The report should contain the following:

- a short introduction, clarifying any interpretation you made of the requirements;
- a UML use case diagram showing the main actors and use cases to be implemented in the system, linking actors to their use cases, with a suitable system boundary;
- a UML class diagram of the initial information model, developed by analysing the given background information, showing classes, attributes, associations and association classes. All associations should have end roles and multiplicities;
- a UML class diagram of the normalised database model (using the UML database profile), which should have normalised all the relationships in the initial information model and identified primary and foreign keys. All remaining associations should be directed, according to table-linkage;
- a UML state machine diagram (using nested state machines where appropriate) showing the design of the user interface, modelling different user screens as the states. The diagram should show which actions are allowed in which states (missing transitions are ignored actions).
- some screenshots (max 2 sides) showing off what you think are the best aspects of what your system can do (screenshots before/after critical events are best). Don't cram in so much that it is unreadable;
- a short discussion of the security features your system implemented.

Your report must finish with two measures of the effort put in by individuals in the team: the first is a factual account of what each person did; and the second is an agreed sharing out credit for effort invested, especially if this was disproportionate:

- a table describing what actual tasks were carried out by each individual;
- divide up 100 points among the team members, according to effort invested;

This last section **must be signed off by all team-members** to be valid.

Hints on Team Working

This team project is large enough that you have to tackle it by a divide-and-conquer strategy. This means you have to learn how to work effectively as a team. Choose a team leader and delegate tasks to different team members. Check up regularly that assigned tasks have been completed. When work has been completed, have someone else in the team review it to point out any possible mistakes or things that need to be improved. You can book break-out rooms in the Diamond for your team meetings. Meet often, and do a little each week.

You may find you have a different spread of skills among the team. Use this: some may be good at UML design, others at database normalisation, others at Java Swing coding, etc.

Tools and Technology

The UML diagrams can either be developed in any of the suggested Open Source UML tools (see end of Project Management lecture), or even in a drawing package such as Visio (or even PowerPoint). Please use a UML 2.x compliant notation. **Do not use non-UML database diagrams**.

The software should be implemented in Java, using Swing to build the user interface. You may use any GUI designer tool that generates your Swing look-and-feel, so long as you know how to link the generated code to the events that your system must process. You may also build your Swing GUI by hand, from the ground up, if you understand that better.

The MySQL database accounts are created by the DCS IT support team and will be distributed by your lecturer (when he gets them). These group accounts are on the Computer Science internal network, not available to other students or outside the department. You will need to use the Connector/J driver for MySQL (instructions to follow in lectures).

Further Help

We will use the discussion forum in MOLE for this course to answer further technical questions. Please follow the etiquette of using the named threads to ask questions, or share answers, on similar topics; and only post a new thread if there isn't one already on this topic.

AJHS will lurk in the discussion forum and will respond to questions on a fairly regular basis, so that answers to common questions can be seen by all.

AJHS, 12 October 2018