Systems Design & Security: Team Project

Overview

This project is part of the learning and teaching method for COM2008/COM3008, and constitutes another 20 guided learning hours (complementing the 40 lecture hours). It will also form 50% of the assessment for the module. So you should expect to devote quite a big chunk of time to this.

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 module grade. There will be a further testing stage, completed individually in week 11. This will be worth 10% of the module grade.

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 Blackboard (only one per team), by 15:00 Friday week 10.
- 1 zipped bundle of your software system to be submitted to Blackboard (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 an academic publisher, which requires a system to manage the submission, reviewing and publishing of academic articles for the journals operated by the publisher. Published articles are organised into quarterly editions, grouped into annual volumes of their respective journal.

Stakeholder roles

The different kinds of stakeholder interacting with the publication system include editor, reviewer, author and reader roles. Editors are responsible for one or more journals and operate the publication system for that journal. Reviewers are responsible for assessing the quality of articles that have been submitted (to be published, or rejected). Authors are academics who submit articles for review, hoping they will be published. Readers are those who wish to access published articles. The same person may perform one or more of these roles at different times.

Business information

The system will store the following information. The publisher operates several journals, including the uniquely-named "Journal of Computer Science", the "Journal of Software Engineering" and "The Journal of Artificial Intelligence". Each of these has its own unique ISSN (International Standard Serial Number – search for examples, and invent fictitious ISSNs for these journals) and board of editors, where the same editor may serve on the board of one or more journals. Where a journal has multiple editors, one is the chief editor of that journal.

Each journal publishes one "volume" per calendar year. Each volume consists of four to six "editions" (also known as "numbers"). Each edition contains published articles, where the number of these can vary from 3-8 depending on how many are accepted. Volumes are described as "vol. X", where X is an integer counting from 1 when the journal was first ever published. Editions are described as "vol. X, no. Y", where Y is the ordinal number of the edition within the volume. Volumes have a year of publication. Editions have a month of publication. Articles published within an edition have a page-number range "xxx-yyy" to identify where in the journal they appear; and page numbering starts from 1 for each new volume and continues consecutively throughout all the editions of that volume.

The readers of a journal can be anyone, typically academics seeking articles to read. No information is logged about readers. Other stakeholders record their title (Prof, Dr, Mr, Ms) their full name (all forenames as one field, a surname as another field), their current university affiliation, their contact email address (used as the login ID) and a password. Where stakeholders are registered by someone else, they may have a temporary or shared password, which they may change later. Editors can be on the board of many journals and each journal has one or more editors. Authors can write many articles and each article has one or more authors. Reviewers can review many submissions, and a submission is reviewed by three reviewers.

The publication system tracks articles, before and after they are published. An article records the title of the article, an "abstract" (a summary paragraph) and is linked to a PDF file with the full article text. An article is also linked to one or more authors (see below), whose details include: full name, university affiliation and email address. One of these is the main "corresponding author" (to whom readers may write). Published articles are assigned to an edition of a volume of a journal and also have a unique page-range (within that volume). A published article is identified by the unique journal name, volume, number, and page range.

An unpublished article is called a "submission." The fields of a submission overlap with those of an article: it has a title, an abstract and is linked to the authors, one of whom is the main author, who supplies an initial password for all the authors. A submission also has a generated unique ID, by which the submission is tracked in the publication system. A submission has a link to the PDF of the draft article. A submission waits in a queue until it has received three reviews (from independent reviewers), with three initial verdicts on the quality of the article. In response to the reviews, the authors will make corrections to the article and replace the draft PDF with a final version of the PDF, along with a written response to each review. The reviewers then replace their interim verdicts with final verdicts on the article. So a submission eventually links (at any one time) to one PDF, three verdicts, three reviews and responses. A PDF file or a verdict therefore may, or may not, be linked to a submission (and will not be linked, if it has been replaced).

Each review follows a standard format. It comes from one reviewer, whose name is anonymised. The system keeps track of which reviewers have been assigned to a submission and gives each reviewer an anonymous ID, styled as: "reviewer 1" … "reviewer 3" for that submission (these IDs relate only to the current submission). The review form refers to the reviewer only by this ID and the form itself is identified by reviewer ID and submission ID. The form contains a number of standard text fields including: a box containing the reviewer's summary of the article (to demonstrate how they understood the article), a box containing a list of typographical errors to fix, and then a series of boxes listing 1..n major criticisms of the article, which indicate questions that must be addressed by the authors. The review is linked to an initial judgement about how acceptable the reviewer considers the article to be (see below).

Each response follows a similar standard format. It uses the same reviewer ID and submission ID, and contains a list of 1..n answers to each critical question put by that reviewer, explaining how the issue was fixed in the revised version of the article (or possibly how the reviewer misunderstood the issue, but changes have been made in the article to clarify this). The answers must correspond exactly to the questions put by the reviewers.

The verdicts are expressed following a style known as "Champion and Detractor". There are only four possibilities: Strong Accept (champion), Weak Accept, Weak Reject and Strong Reject (detractor). It is possible to automate article acceptance based on three final verdicts (see below). A verdict is like a review in the way it refers to the reviewer and submission by their secret IDs.

Business rules

Editors operate the publication system with highest privilege; authors and reviewers have restricted privileges described later below. A chief editor is initially appointed for each journal. This person can appoint other editors to the board for that journal. The role of chief editor can be transferred to another editor on the board. An editor may choose to retire from the board, so long as at least one editor remains for that journal (if only one, that person becomes chief editor).

The main rule that is enforced to ensure academic integrity is that there are no conflicts of interest between the editors, reviewers and the authors seeking to be published. An editor cannot preside over an edition, if their affiliation(s) overlap with the affiliation(s) of any submitting author to that edition. If this is the case, the editor must step down (temporarily retire) from the board, following the rule above. Similarly, a reviewer cannot review an article, if their affiliation(s) overlap with the affiliation(s) of any of the submitting authors of that article.

The second rule that is enforced to mitigate against accidental disclosure of intellectual property is that unpublished articles must be kept as confidential as possible, only visible to authorised reviewers and the editors of the journal.

The publishers operate an "Open Access" policy for their journals, which means that articles are free of charge to any potential reader (not paid for by subscribers). This means that the system must cover its own costs. This is done is by asking potential authors also to review other articles. The business rule is that three other articles must be reviewed, to cover the publication costs of each submission.

When an article is submitted by the main author, all the authors of that article are automatically registered as authors with the system. For the duration of the time that their submission is under consideration, any author may access the status of their submission (submitted, reviews received, initial verdict, responses received, final verdict, completed). Once the submission has been accepted or rejected by an editor (completed), these privileges are removed.

To cover the costs of each submission, three reviews must be conducted of other, unrelated articles. The authors automatically become reviewers, and are able to see many articles awaiting review in a confidential queue, for which there are no conflicts of interest with any of the authors. They may select three of these papers to review. As soon as this happens, only those three articles are visible and the rest of the confidential list is no longer accessible.

Reviewers retain their privileges until they have given their final verdict. Until this point, they can see their original reviews, including the questions they put to the (other) authors, the original and revised article and the answers they have received from those authors. They check that the authors have made suitable corrections. As soon as they submit their final verdict, all their reviewer privileges are removed and they can no longer access any of the above details.

Once three final verdicts have been received for a submission, any of the editors of the journal may decide to accept or reject, based on the final verdicts (the initial verdicts play no part; they were only indicative). The rules for this are mostly automatic, but must be confirmed by an editor. If a submission has only champions and no detractors, it is accepted. If it has only detractors and no champions, it is rejected. If it has both champions and detractors, it must be discussed by the editors and a manual decision is taken (either way). If it has no champions or detractors, take a majority decision based on weak accept/reject judgements.

To satisfy GDPR[1], stakeholder personal information may only be retained for as long as that person has a role in the system. Editors only remain if they are on the board of at least one journal. Other stakeholders may only be retained if they are fulfilling either an author or reviewer role, and vanish from the system when these roles terminate. (One reason for this is that academics may change affiliations and email addresses frequently, especially early in their career).

System operations

Editors perform the following tasks:

- one chief editor for each journal may self-register initially in the system and then controls who is appointed to the board for that journal;
- the chief editor may register other academics as editors for the board of their journal; the same individual can be an editor for more than one journal;
- the appointed editors may login with their temporary password (chosen by the chief editor when appointing them) and change this to something more private;
- the chief editor may pass their role as chief editor on to another editor;
- an editor may retire (possibly temporarily, to avoid a conflict of interest) from the board for a journal, so long as at least one chief editor remains on the board;
- an editor may see a list of articles under consideration, about which they are entitled to take a final decision, but which excludes any conflicts of interest;
- any editor on the board may make the final decision to publish or reject an article, according to the “Champion and Detractor” rules;

[1] the General Data Protection Regulation (see lecture 2)

- a published article is added to the next edition in the current volume of the journal;
- if the next edition is already full, the editor may delay publishing to a subsequent edition;
- the chief editor may publish the next edition of the journal when it is ready, thereby making all the contained articles available to readers.

Authors perform the following tasks:

- one main author is in charge of submitting an article for consideration; and by submitting, they self-register as an author and register all the co-authors of the article;
- the co-authors may login with their temporary password (chosen by the main author when submitting) and change this to something more private;
- all authors may log in to track the status of their submitted article, until the editor dealing with this submission decides to accept or reject the article;
- all logged-in authors can see the reviews (of their own submission), the initial verdicts, their responses, the final verdicts and the initial and revised versions of their article;
- one main author is responsible for responding to the criticisms expressed in the reviews and for submitting the revised article, along with a response to each reviewer;
- each response must include a list of answers to each of that reviewer's questions, where the review contained a list of questions (or criticisms).

Reviewers perform the following tasks:

- the authors of a submission temporarily become reviewers (of other articles); they gain reviewer privileges automatically, upon submitting their own article;
- the reviewers select three different articles from the confidential queue of unpublished articles; they can only see articles for which there is no conflict of interest;
- upon selecting the three articles they will review, all other confidential articles are hidden;
- any reviewer may submit a review form containing the standard text fields, and a list of individual criticisms of the reviewed article that must be addressed;
- at the same time, the reviewer submits an initial verdict, according to the "Champion and Detractor" style;
- when the revised article and author-responses come in, each reviewer checks that the response to their criticisms was appropriate and then submits a final verdict;
- upon submitting a final verdict, they lose all reviewer privileges and can no longer see any confidential details relating to the reviewed articles.

Readers perform the following tasks:

- view the available journals, volumes and editions (also known as numbers);
- view a list of articles, indexed by journal, volume and number;
- view the abstract of any article, selected by journal, volume, number and page-range;
- access the PDF file for selected articles.

User interface considerations

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

- a user should be presented with a welcome-screen that allows them to log in to one or other of their relevant roles; some users will have more than one role-permission.
- while logged in as a particular role, a user should only be able to access data that is relevant to that role (but they may logout and login to a different role).

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 three named journals to the system; and appoint a chief editor for each of the three journals (a chief editor is self-appointing), who appoints a board of editors to each journal, including some who serve on the board of more than one journal.
- Retire the chief editor of one journal, such that a new chief editor is automatically chosen from the other editors, or the chief editor cannot retire if they are the sole editor.
- Submit five articles to one of the journals, such that two have single authors, two have two authors, each from the same university, and one has three authors, each from a different university; and such that
- One pair of articles have a common author; and a different pair of articles have authors that share a university affiliation; and a third article has an author whose affiliation is the same as one of the journal's editors; and there are no other conflicts of interest.
- Login as that conflicted editor to show that they cannot make any decision about the article with which they have a conflict of interest.
- Login as a reviewer with no conflicts of interest, to show that they can see all four of the other articles that are not their own submission.
- Login as the reviewers with a conflict of interest, to show that they can only see three other articles, with which they have no conflicts of interest.
- Progress these five submissions through the reviewing stage, such that
  - the reviewers select or confirm the three articles they are going to review;
  - the reviewers submit their reviews, with a varying number of criticisms/questions from each reviewer and a verdict from each reviewer;
- Login as the main author of one of these submissions, to show that they can access their original article, the reviews and the initial verdicts.
- Progress the five submissions to the final acceptance stage, such that
  - the main author creates a response to each review, with corresponding answers to the questions put by each reviewer;
  - the main author submits the responses with an updated version of the article.
- Login as the reviewers again, and submit final verdicts for the five papers, such that
  - one has two strong accepts and a weak reject; one has one strong accept and two weak accepts; one has a strong accept, a weak reject and a weak accept;
  - one has a strong reject and two weak accepts; one has two weak accepts and a weak reject.
- Login as one of the journal editors and confirm the publishing decisions made about the five papers, pushing the accepted papers to the next edition; and publish the edition.
- Browse the system as a reader, to show that you can access the recently published edition.

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

Final Team Report

The main purpose of the team report (10-15pp) 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 denote 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 contributions-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. One such example tool is Jvider: http://www.jvider.com/download. You may prefer to 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 Blackboard 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, 16 October 2019