Systems Design & Security: Team Project

Overview

Coursework will form 50% of the assessment for COM2008/COM3008. The team project will be completed in teams of four. 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

There will be two submission links under Assessment in Blackboard. One is for your team report, and the other is for your team software bundle.

- 1 team report (typically 10-12pp) to be submitted as PDF to Blackboard (only one report per team), **by 15:00 Friday week 10.**
- 1 zipped bundle of your software system to be submitted to Blackboard (only one bundle 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 virtually with a representative of the team you are testing, who will share with you their software bundle. You will run a series of prescribed tests (to be released later) of their system.

- 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 HomeBreaks Plc, which manages a collection of small properties for short-term lease by hosts to renting guests. The software system will capture details about the properties, their hosts and guests, and will manage bookings and reviews of the properties. The security aspect is that contact details for individuals and the precise locations of properties are not revealed to a casual enquirer, but only to authorised users, according to the business rules.

Stakeholder roles

The different kinds of stakeholder include host, guest and enquirer roles. Hosts supply all information about the properties offered for lease and accept bookings from guests. Guests make bookings for properties for short periods and write reviews about their stay. Enquirers are future guests, who search for properties to rent during a period, before they make a booking.

Business information

Each property is publicly advertised with a short name, e.g. "country cottage" or "first floor flat"; a slightly longer description, e.g. "stone built cottage with off-road parking, fully equipped kitchen and garden to front"; a general location, e.g. "Haydon Bridge, United Kingdom"; and indicates whether the host offers breakfast at this property (true or false). Each property reports: the maximum number of guests (based on the maximum number who can sleep there), the number of beds, bedrooms and bathrooms; this is derived from other information. It is also possible to view related reviews (see below). Each property is offered by a host, who may offer more than one property. A host is publicly advertised with a host-name, e.g. "Sandra", and states whether they have earned the title of "superhost" (true or false).

Properties may be rented by guests. Each guest is publicly advertised with a guest-name, e.g. "James". Both hosts and guests related to a person. The details of a person are confidential. A person has a title, e.g. "Ms" or "Dr"; a forename, a surname, an email address and a mobile number. Each person is related to a host, or a guest, or both. Each person and each property is related to a confidential address. An address consists of a house (number or name), a street name, a place name (village, town, or city) and a postcode. The house and postcode uniquely identify an address. Each address relates to a person, a property, or both.

A property offers a number of facilities, where each facility is themed after a zone in the property. The six possible kinds of facility have standard names: sleeping, bathing, kitchen, living, utility and outdoors. Each facility offers a different set of amenities (services or features), which are either present or absent in any given property. The customer (HomeBreaks Plc) views each facility as a separate table of check-boxes which can be ticked or not (follow this in your database design). The ticked amenities offered in each facility will be displayed in the UI when requested.

The sleeping facility may offer: bed linen, or towels as amenities. Furthermore, this facility is related to one or more bedrooms. Each bedroom has: bed1 and bed2, which are each described as one of: single bed, double bed, kingsize bed, bunk bed (or bed2 is left blank if only bed1 is supplied). From this, the bedroom derives how many beds it has, and how many people may sleep there (all beds apart from the single bed accommodate two sleepers). The sleeping facility reports how many bedrooms, how many beds, and how many sleepers there are.

The bathing facility may offer: a hair dryer, shampoo or toilet paper. Furthermore, this facility is related to one or more bathrooms, where each bathroom may offer any of: a toilet, a bath, or a shower and declares whether or not this bathroom is shared (with the host). The bathing facility reports how many bathrooms there are.

The kitchen facility may offer: a refrigerator, a microwave, an oven, a stove (i.e. hobs), a dishwasher, tableware (plates and cutlery), cookware (pots and pans) or basic provisions (i.e. oil, salt, pepper). The living facility may offer: wifi, television, satellite, streaming, a DVD player or board games. The utility facility may offer: (central) heating, a washing machine, a drying machine, a fire extinguisher, a smoke alarm, or a first aid kit. The outdoor facility may include: free on-site parking, on-road parking, or a paid car-park; and may also include: a patio (a garden terrace with seating), or a barbeque. These amenities are either present or absent.

Each property is related to one or more charge bands, describing the cost of renting at different times of the year. Each charge band has a start date and end date (for when these rates apply), and lists: a price per night, a service charge and a cleaning charge. The set of charge bands associated with a property cover all dates continuously for the coming year, up to end December 2022.

System operations

A host self-registers with the system when they first want to offer a property. A guest self-registers when they first want to book a property. At registration, they supply their public host- or guest-name, their confidential person contact details and their confidential address. Their email address serves as a user ID, and they pick a secure password. The login details belong to an individual person, but users always log in to their host or guest role. Enquirers are not authenticated.

The property offering process works as follows. The host logs in to their host role. They create a new property entry, with all the public information, and supply the confidential address of the property. Then, the system presents the host with each facility in turn (sleeping, bathing, kitchen, living, utility, outdoors) and asks them to check which amenities are offered. For the sleeping and bathing facilities, the host also has to create a number of bedrooms (typically one or two) and bathrooms (typically one) and checks what amenities are offered there.

The booking process works as follows. Initially an enquirer searches the public information about properties and hosts, and selects properties in a given location between a start and end date. This returns a set of bookmarks, where a bookmark relates to one property, describes the start and end date and derives whether the property is available for the requested period. The enquirer can view the list of bookmarks, and examine each property, host and facility in more detail (only the public information).

When ready to book, the guest logs in to their guest role. They request to book a property between a start and end date. So long as the property is available during these dates, this creates a provisional booking, which will later be accepted (confirmed) or rejected (turned down) by the host. Each booking is by one guest for one property (though there may be many other bookings by the same guest, and many other bookings for the same property).

Both hosts and guests can see a list of bookings in their home screen. The host sees a list of bookings made for each of their properties. The guest sees a list of bookings they have requested (on many properties). The host may choose to accept or reject a provisional booking, but cannot accept overlapping bookings on the same property. A guest must login to see whether their provisional booking has been accepted or rejected. We assume bookings remain visible until after their end-date, when they are automatically deleted.

For each booking, it is possible to derive: the number of nights, the price per night, the service charge, the cleaning charge and the total charge (the sum of the costs for the nights and the service and cleaning charge). Once the booking is accepted by the host, the guest and host will be able to see each other's confidential contact information. We assume the guest contacts the host directly to pay for their stay. The guest will also be able to see the precise address of the property. All confidential information will be viewable by both parties until the stay has been completed (the end date), after which it is no longer viewable.

After their stay, the guest submits a review. Each review is by one guest and for one property, though many reviews may be submitted for the property, and a guest may submit many reviews. Each review contains a text description of how satisfied the guest was, and gives a 1..5 (low to high) score in six categories: cleanliness, communication (response time), check-in (how easy), accuracy (of public info.), location (quality), and value (for money). For each property, it is possible to derive the average scores for each category over all reviews submitted, and the overall average rating. A host becomes a superhost if they have maintained a 4.7 average rating for their properties.

User interface considerations

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

- when an enquirer searches, or a host or guest logs in, they should be presented with a home-screen 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; and not view data relating to other roles;
- confidential information should only be shared among the stated parties at the points mentioned; and an enquirer may never see any confidential information.

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:

- Register a person as a host, another person as a guest, and a third person as both host and guest.
- Add a property for one host, and two properties for the other host, with different selections of amenities at each property.
- Allow an enquirer to search for properties in a location, returning zero, one or more bookmarks, depending on the location, with availability indicated.
- Allow the enquirer to click through to see the facilities and amenities of a property, but not access any confidential information.
- Make a provisional booking as a guest, which goes into the host's queue. Make another provisional booking which is refused due to overlapping with an *accepted* booking (not the other *provisional* booking, which may overlap).
- Have the host accept a booking. Check that the guest can see this. Check that the paired host and guest can now see each other's confidential information (person and address details), but no other host or guest can see this information.
- Have two guests write two reviews of the same property. Check afterwards the property's average review scores and list the comments.

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 interpretations you made of the requirements;
- 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;
- 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.

Individual Weighting

Your report must finish with **two separate measures** of the contribution and the effort put in by individuals in the team:

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

The first of these is a **factual account** of what each person did; and the second is an **agreed sharing** of credit for effort invested, especially if this was disproportionate. These measures are distinct; a student may have invested significant effort, but fewer of their contributions may eventually have been included.

This last section **must be signed off by all team-members** to be valid, otherwise equal effort and contributions are assumed by default. If there is serious disagreement, this should be reported, and different parties should report their different views of contribution/effort. Assessors reserve the right to arbitrate and alter effort scores in this case.

Marking Scheme

The marking scheme will be as follows:

- 30% for accurate UML Diagrams
- 60% for correctly operating software
- 10% for team working strategy

Individual marks will be weighted *partly* according to the whole team score, and *partly* according to the share of contributions/effort, following a **non-linear scheme** that encourages collaborative working, rather than heroic individual efforts. No individual mark will be boosted by more than one degree-class. Low-contributors may have their mark reduced below the pass threshold. Non-contributors will receive a mark of zero.

Hints on Team Working

This team project is large enough that you must tackle it by a divide-and-conquer strategy. This means you must 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. All of these aspects are important (not just the coding). The marking scheme is designed to reward collaborative working. Individual weighting will not excessively reward heroic efforts. It is your collective responsibility to make the team work as a group.

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

From week 5, there will be several Online Helpdesk surgeries, operated at different times by senior student Demonstrators who have signed up for this. There will be additional labs, operated by University Teachers, to help build up your coding skills. Please see the times advertised in Blackboard, under the module overview.

We will use the Blackboard discussion forums 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 is not one already on this topic. Module instructors will respond to questions posted there on a regular basis, so that answers to common questions can be seen by all. Email will not be used for technical questions.

AJHS, October 2021