Michael Butler
mjb@ecs.soton.ac.uk
Dana Dghaym
D.Dghaym@ecs.soton.ac.uk
Thai Son Hoang
T.S.Hoang@ecs.soton.ac.uk
Yvonne Howard
ymh@ecs.soton.ac.uk

COMP1216. Software Modelling and Design (2019-20)

Coursework 2: Formal Modelling

Issue date:	12 May 2020
Submission deadline:	4pm, 29 May 2020

This coursework, to be undertaken in groups (the same group as Coursework 1), will contribute 15% towards the total for the unit. The assignment concerns the formal modelling of an interactive quiz system from Coursework 1. It is intended to develop your skills in writing formal models using Event-B, to gain the experience of using the Rodin Platform and also re-enforce group working.

Please bear in mind the University Academic Integrity regulations: http://www.calendar.soton.ac.uk/sectionIV/academic-integrity-regs.html

1 System Outline: An Interactive Quiz System

The outline of the system is as described in Coursework 1, i.e., http://edshare.soton.ac.uk/20332/

For Coursework 2, you will model the system using Event-B according to the following requirements.

1.1 Users

The system manages a set of registered users. A user can register with the system with a password. Registered users can log in to the system using their password. Once logged in, the user user can log out of the system. (Here, we *do not* model the fact that a user can log in using multiple devices.)

REQ 1	The system manages a set of registered users

REQ 2	The system should allow a user to register with a password

REQ 3	The system should only allow a user to log in to the system using the correct password

REQ 4	A logged-in user can log out of the system

1.2 Quizzes

Only logged in users can create new quizzes. The user who created a quiz becomes the *creator* of the quiz. Each quiz contains a *sequence* of questions. The creator of a quiz, once logged in, can create new questions and their answers for that quiz, can update the existing questions of the quiz and their answers, can remove a question from the quiz, and can remove the quiz itself.

REQ 5	A logged in user can create a new quiz and becomes the creator of the quiz

REQ 6	Each quiz contains a “sequence” of questions

REQ 7	A logged in creator of a quiz can create new questions and answers of the quiz

REQ 8	A logged in creator of a quiz can update the existing questions and their answers of the quiz

REQ 9	A logged in creator of a quiz can remove a question from the quiz

REQ 10	A logged in creator of a quiz can remove the quiz

For each question, the number of possible answers (i.e., the options) is between 2 and 4. Amongst the options, only one of the answers is the correct one for the question.

REQ 11	Each question has between 2 and 4 possible options

REQ 12	Each question has exactly one correct answer, which must be one of the options of the question

1.3 Running Quizzes

The quiz can be shared to other registered users of the system by the quiz's creator, who must be logged in. Once shared the quiz is accessible to the other users.

REQ 13	The logged in creator of a quiz can share the quiz with other registered users, who then have access to the quiz.

The state machine of a running quiz can be seen in Figure 1. Any logged in user who has access to a quiz can start the quiz and becomes the quiz's host. Moreover, a quiz can have more than one *running instance* at the same time. Once started, participants can join a running instance of a quiz. Participants can be logged in users or anonymous users.

REQ 14	A logged in user who has access to a quiz can begin that quiz and becomes the quiz's host

REQ 15	A quiz can have multiple running instances at the same time

REQ 16	Participants (logged in users or anonymous users) can join a running quiz instance

At each step, the host shows the question and the participants answer the question via their device. When a question is finished (e.g., when the all participants have answered or when the host terminates the question early), a summary of the question is shown. After showing the summary of the question, the host can move to the next question of the quiz if any. At any point, the host can decide to terminate the quiz early. At the end of a quiz, a summary of the quiz is shown.

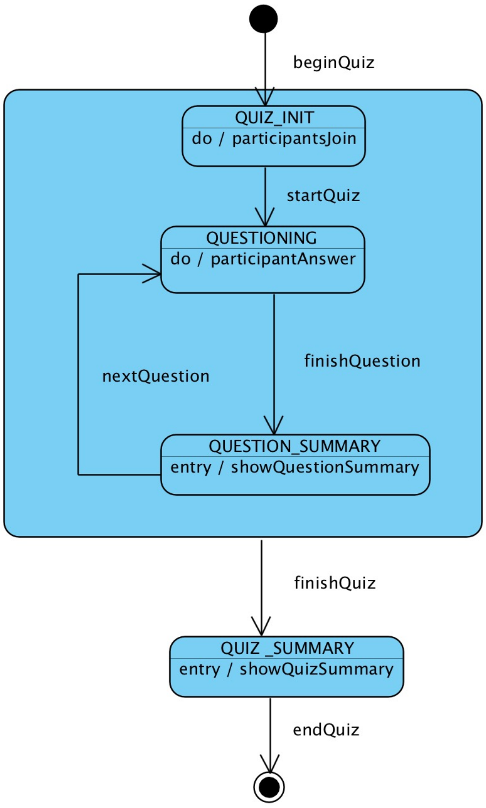

Figure 1: State Machine of a Running Quiz

REQ 17	At each step, the host shows the question and the participants answer the question

REQ 18	When a question is finished, a summary of the question is shown

REQ 19	After showing the summary of the question, the host can move to the next question of the quiz (if any)

REQ 20	The host can finish the quiz when there are no more questions or terminate the quiz early

REQ 21	At the end of a quiz, the quiz's summary is shown

2 Tasks (50 marks)

2.1 Task 1 (5 marks)

Identify the main *entities* from the system requirements and create a *class diagram* to represent the important relationships between these entities. You may reuse your class diagram from Coursework 1.

2.2 Task 2 (45 marks)

Use the system requirements and the class diagram to create an Event-B model of the interactive quiz system. Make sure to

- Identify appropriate Event-B sets and constants
- Identify appropriate Event-B variables and invariants. The invariants should clearly specify any constraints between state variables. You may lose marks by not having invariants to represent constraints.
- Identify appropriate Event-B events. Your model should have the events:
  - User registers to the system
  - User logs in to the system
  - User logs out of the system
  - User creates a quiz
  - User removes a quiz
  - User create a questions (including its options, and the correct answer)
  - User updates a question
  - User removes a question
  - User shares a quiz to another user
  - User hosts a quiz
  - Register user joins a running quiz
  - Anonymous user (guest) joins a running quiz
  - The quiz's host starts the quiz
  - Participant answers the question
  - Finish a question and show the question summary
  - The host continues to the next question
  - Finish a quiz and show the quiz summary
  - The host ends a running quiz

- Identify constraints on the variables and model these as invariants, e.g., only logged in user who has access to the quiz may host the quiz.

Hints You should consider using extension refinement to structure your Event-B model.
Remarks

- Up to 10 marks will be awarded for a model developed using the Rodin platform (see the submission instructions below).
- Up to 5 marks will be awarded for a model using extension refinement.
- This means a paper-based solution without refinement will have a maximum of 35/50 marks.

3 Submission Instructions

Each group should submit a written report (PDF format, one report per group) giving your answer to each of the tasks above. Clearly indicate your group number, member names and email IDs on the front page. Make sure your Event-B models are appropriately commented and are included in the *PDF report*. You should also include an *archive of your project from the Rodin tool containing your models*. Up to 10 marks will be awarded for the model developed using the Rodin platform. You should use the automated hand-in facilities found on the student Web pages at: https://handin.ecs.soton.ac.uk/.

The group allocations are on the course Noteswiki page.

If you feel there are any ambiguities in the requirements feel free to make your own interpretation, but make sure any interpretations you make are *clearly indicated in the report*. You should work *together as a group* to accomplish these tasks. It is the responsibility of each group to make initial contact and arrange their own group meetings. Please inform us of any problems contacting your group members. You should avoid discussing your solutions with other groups.

NB: Group size is *four*. It may be necessary to run one or two groups of three depending on the size of the class, or any students dropping out. In these cases the workload will be reduced accordingly.

- Group of 3: You are *not required* to model the events corresponding to
  - User logs out of the system
  - User removes a quiz
  - User updates a question
- Group of 2: You are *not required* to model the events corresponding to
  - User logs out of the system
  - User removes a quiz
  - User updates a question
  - Anonymous user (guest) joins a running quiz

  Furthermore, you can omit the details about question summary and quiz summary of a running quiz.