COMP3297 Software Engineering
Department of Computer Science
The University of Hong Kong

Elaboration 2 Task Sheet
The Tutoria Project

In Elaboration 1 you dealt with risks associated with uncertainty in your use of Django and in your understanding of requirements. You achieved this by implementing thin vertical prototypes using the Django framework. You now have an executable in place that provides some of the basic functionality required in Tutoria.

In Elaboration 2, you will deal with remaining risks associated with other Django features, and implement additional requirements of high priority to users to confirm that your design is workable. These areas deal with:

- registration and authentication;
- session locking and ending (including distribution of payments);
- tutor search.

As in the previous iteration of this Phase, you will:

- revise your specification of requirements where necessary and add detail to them;
- design, implement, test and baseline a further increment of the architecture.

Early in this iteration you will also demonstrate your Elaboration 1 executable to ensure it meets expectations.

Description: Iteration 2 (E2) of Elaboration

Building on your E1 executable, add required functionality in the areas listed above. If you do not have use cases covering those parts of the requirements you must add them at the start of this iteration. The functionality is not particularly complex and SSDs are not required. For tutor search, some teams left it in *Book Session* in E1 while other teams extracted it into its own use case. Regardless of its location, in E2 the documentation of this functionality and that of booking a session must be completed in detail, including all Extensions.

Functionality associated with use of *coupons* is not required in E2.

With reference to stakeholder needs, your E2 executable will add the following to your E1 executable:

Registration and Authentication:

- (for Users) register with Tutoria, specifying a username and password for subsequent authentication.
- (for Users) retain access to their account if they forget their password.
- (for Tutors) register as a tutor and specify whether they are a contracted or private tutor.
- (for Students) register as a student.
- (for Administrators) control access to Tutoria through user authentication based on username and password.
- (for Administrators) send mail to the user containing a token for password reset after receiving a lost password request.

You might find Django's authentication framework useful here. This is not a constraint, however, and you are free to use or develop alternative solutions if you wish.

In E2 you are not required to implement profile initialization or management for tutors or students. That functionality is straightforward and low risk; you will add it in Construction.

Session Locking and Ending:

You documented the corresponding use case(s) in Inception and revised them in Elaboration 1.

- (for Tutors) receive payments for tutorial sessions promptly following their completion.
- (for Tutors) prevent students from cancelling sessions less than 24 hours from the scheduled start time.
- (for Tutors) prevent students from booking sessions less than 24 hours from the scheduled start time *(added by client during E1).*
- (for Tutors) receive notification when a payment is made to their wallet.
- (for MyTutors) receive a commission on payments for sessions.
- (for Students) receive an invitation to submit a review.

Note: Your use cases described the required functionality without placing constraints on the design. There are many ways to design and implement this functionality – any approach is fine that can satisfy the requirements.

If you choose to implement a trigger via the equivalent of a *cron* job, ease of implementation and availability of third-party applications will vary a lot depending on the underlying OS of your development environment (we will not deploy to a production server in this project). If you choose this approach, it will be sufficient just to provide some manual means of triggering execution of the use case(s). This will be enough to support testing of the use case implementation(s).

Tutor Search:

- (for Students) search for tutors, specifying any or none of the following: university, university course, subject tag, hourly price range, whether tutor is contracted or private, and whether to show all tutors who satisfy the criteria or only tutors with at least one available timeslot in the next 7 days.
- (for Students) search for a tutor by family name or given name(s).
- (for Students) view short profiles of tutors that meet their specifications.
- (for Students) sort search results to view tutor profiles ordered by hourly rate.

Deliverables (E2)

1) Analysis models

Refine and extend your Domain Model if necessary.

2) Design models

Refine and extend the Design Class Diagram that documents your Django model classes.

Extend your documentation of use case realizations to add those developed in this iteration. As in E1, provide sequence diagrams showing how elements of your realizations interact within the framework to provide the required request/response cycles. If you used a written form of documentation for your E1 realizations, you should now replace those with sequence diagrams.

3) Implementation

Extend the previous increment (that is, your E1 implementation) by implementing the new parts of your design for this iteration. Your executable should provide the functionality in the Description above.

4) Updated Artefacts

If you need to update or extend earlier documents, make your revisions and submit new versions of them.

Delivery:

Please upload a zipped file to Moodle containing new and modified documents and diagrams as described above with two hard copies delivered to our TAs as soon as possible afterwards..

Deadline:

November 12 (Sunday), 11:55pm