

COMP 1531 S2 Group Project

Online Survey System – Iteration 3

A. Requirement Specification drafted by your consultancy firm

Based on the feedback from the School of Computer Science's advisory board, your team realises that there are some changes required in your current design.

(1) Security:

The major changes to the design of the security infrastructure are summarised below:

- Four kinds of users can access the system: "admin", "staff", "student" and guest
- Prevent unauthorised access to dashboards of different "roles"
- Design a new workflow to implement the "register guest user" feature. The work-flow process is described below:

Workflow Process: Register Guest User

A guest user will **only** be able **fill** out a survey for a particular course-offering. As public users will no longer have access to the system, any user who is currently not a student, staff or admin **will need to register** for "guest" access if they wish to fill out survey for particular course-offerings. The workflow process to register a "guest-user" is provided below. The registered guest-user will now be able to log into the system with the supplied "guest" user-name and password and fill out the survey for the registered course-offerings. Each survey can be filled out **once** only.

Workflow Process: Guest User Registration

If a non-UNSW staff or student (i.e. a public respondent) wishes to fill out a survey, they must first be registered with the survey system with a "guest user account". This is the workflow process for registering a guest user.

- (1) Public respondent will click on "Register Guest User link" on survey home-page.
- (2) Survey system pops up a guest user registration form that requests the user to provide a *user id*, *password* and *course-offering* of interest
- (3) Public respondent provides the requested details and clicks on submit
- (4) The request is forwarded to the admin user
- (5) Admin user authorises the guest user registration request
- (6) The guest user details (user id and password provided are forwarded to the authentication module)
- (7) Authentication module registers the "guest user" into the database, recording the user id, password and role as "guest".
- (8) Authentication module sends back a "successful authentication" message back to the admin user
- (9) Admin user assigns guest user to the course offering requested

- (10) Guest user is directed to their dashboard where they see their course offering (if it has an open survey associated to it)
- (11) Once the survey is filled out, the survey will disappear from their dashboard

B. Deliverables and Marking Criteria

(1) Presentation (8 marks)

For this presentation, your tutors will play the role of the customer. Your task is to "sell" your design and application to them

- Each team will be provided 18 minutes.
- Please ensure that you have already uploaded the working software to the CSE server, prior to the start of the lab. All demonstrations must be based on software uploaded to CSE server.
- Please ensure that you have done a PIP install of all necessary Flask extensions prior to the start of the lab
- At the start of the presentation, provide a list of core user-stories that you will be demonstrating. (This only needs to include what you will be presenting. A full list of all user-stories and acceptance criteria will be submitted separately. More details on this below).
- A total of 8 marks will be awarded for this presentation. Marks will be awarded based on (i) quality of your presentation (ii) demonstration of the key features of your system (iii) how well you can sell your design to the customer.

(2) Artifacts to be submitted

(a) (20 marks) – Final Report

A final report is to be submitted that includes the following artifacts

- i. A front-sheet that lists (1) your group-name (2) your group members (3) Percentage of work contributed by each team member and signed by all group members.

(Note: If you are currently experiencing difficulties working together as a group, please contact either your tutor or myself to have the conflict resolved immediately. Complaints brought to our notice in the week of the demonstration will **not** be accepted. However, if a complaint is brought to our attention in the weeks before the project, we will endeavour to resolve the issue and if the "person" continues to fail to contribute (despite our efforts to resolve), this person will be awarded a mark of 0 for the group project iteration 3 component.

- ii. A final, updated list of user-stories along with clearly-defined acceptance criteria to comply with the revised project specification. The list of user-stories must include all the features of your system. This list **must also** be submitted as a separate deliverable through the submission system.
- iii. Sequence diagram (from iteration 2) to capture survey workflow process. This must show the (a) survey creation by admin (b) survey review by staff (c) survey filled out by student (d) survey closure by admin

- iv. A revised (if any changes needed) conceptual class diagram with **attributes only**.
- v. An entity-relationship (ER) design (The ER design must map to the relational schema of your database)
- vi. An architecture diagram that identifies the software architecture (key components and their interactions) of your system
- vii. A log that records the responsibilities allocated to each team member, progress of tasks using a velocity chart (scanned copy of hand-drawing will suffice, no sophisticated tool needed)
- viii. A reflection of your implementation and what would your team do different if you had to re-implement this application

Note: Design, software architecture and conformance of design to implementation will form part of the report assessment mark. Here, the tutors will also check your source-code. If your implementation does not correspond to the design submitted in your report, (some teams had this problem in iteration 2), marks will be deducted.

(b) (22 marks) – Functionality and Testing

Two types of testing will be done – (i) User Acceptance Testing (ii) Unit Testing

(i) User Acceptance Testing (12 marks)

- All features of the system must be implemented including metrics
- A complete list of user-stories and their associated acceptance-criteria will have to be submitted as a separate pdf file. Your tutor will run through each user-story and the acceptance criteria to ensure that the functionality described in the user-story has been achieved.

(ii) Unit Test Cases (10 marks)

A comprehensive set of test-cases must be submitted for the acceptance criteria defined above. Please adhere to the following guide-lines in implementing the test-cases.

- Test-cases must be written for **five** core user-stories.
- Of the chosen user-stories, at a minimal the following features must be included: (1) create mandatory, optional questions (2) create a survey (3) enrol a student
- The test-cases can be implemented with any testing framework such as **unittest** or **pytest**.
- Each user-story must have multiple test-cases defined. For example, if you are writing test-cases for "create a survey for a course-offering", you could be testing for (i) what happens if a survey is attempted to be created with 0 questions (ii) what happens if a survey is created with duplicate questions etc. Note, these test-cases must match the acceptance criteria that you have defined for the chosen user-story.
- The test-cases must be submitted in a file **"tests.py"**. They will be run by your tutor. Hence, make sure that the test-cases execute successfully.

To guide you with this exercise, we have provided you with two sets of test-cases.

Set 1: These test-cases have been written for two user-stories (a) enrol a student (b) add a staff based on our model solution implementation of the online survey application.

Set 2: These test-cases have been written for the auction case-study that you have been working in the labs over the past few weeks

(c) (4 marks) – Bonus marks for optional feature implementation

Teams that have implemented the optional feature “register guest user” **successfully** will be eligible for bonus marks. There is a total of four bonus marks allocated to the implementation of this feature and is awarded based on good design and correct implementation of the feature.

(3) Final Submission

- All working software including test-cases (Artifact-2b(ii)) will need to be submitted by Monday, Week 13, 11:59.
- You will need to demonstrate in your lab session using the code you have submitted to the CSE server
- All reports (final report, user-stories and acceptance criteria for user acceptance tests – Artifacts 2a & 2b(i)) will need to be submitted by Friday, Week 13, 11:59. Reports should be in PDF format
- You will be advised on further details on how the report and working software are to be submitted in due course.