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 several key changes/additions to both the **functional** and **non-functional** requirements of the online survey system. Upon further discussions with your client, your team has drafted the following proposal in agreement with the client.

In view of the proposed changes, you have outlined the high-level software architecture for the revised online survey system as shown below.
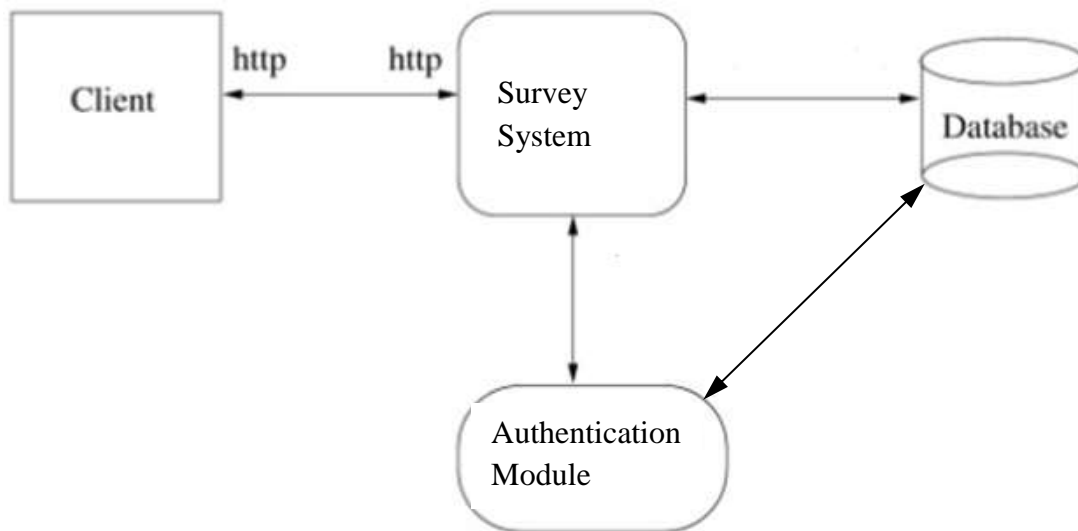
The online survey system will implement a **three-tier client server** architecture. A client will access the *survey system* from a browser using HTTP. The survey system will use an *authentication module* to implement the security infrastructure. Both the survey system and the authentication module will use a **database** as the persistence layer. Each of the components in the above architecture is described below.

(1) Persistence

As the client have emphasised that robustness, reliability and efficiency are paramount to the survey system which will eventually need to cater for different faculties, you have indicated to the client that the data-tier should be migrated **from a flat file-system to a database**. This option will be more reliable and scalable to handle a large number of users. The database will also be to provide for more sophisticated and efficient query mechanisms over flat-file storage.   The authentication module and the survey system will both use the database as their persistence layer.

(2) Security:
Public users will no longer have access to the survey forms in the system. To ensure this, the website will now have a secure authentication mechanism. A **log-in** mechanism will be provided to authenticate that all users of the online system. The authentication workflow is described in section (3). Three kinds of users can access the system: "admin", "staff" and "student". Each of these users is described below:

Student:

A student is able to log into the survey system using their **zID** and **password**. When students log in, they should be able to see the course offerings that they are currently enrolled in and associated with an **"open"** survey. Students should be able to click on those course offerings, which takes them to the survey form to be completed. Once the survey has been completed for a particular course-offering, the latter will be removed from the list of course-offerings displayed on the student's dashboard. A student will be able to see on their dashboard, links to results of surveys that have **"closed".**

Admin:

An admin user is regarded as the "super-user" and is able to log into the survey system with an admin **user-name** and **password**. The admin user retains the responsibilities from the previous iteration. The responsibilities of the admin user are summarised below:

- An admin user will be able to add a question/delete a question a question from the pool of generic or optional questions. Deleted questions still appear in surveys to which they have already been added.
- An admin user will be able to create a survey for a particular course-offering from the pool of generic and optional questions and define a start-time and end-time for the survey
- An admin will be able to "close" a survey after the survey period has lapsed.

Staff:

A staff can log into the system using their **staff-id** and **password**. When staffs log into the system, the assigned list of course-offerings with associated surveys in **"review"** stage is displayed on their dashboard. A staff can click on a particular course-offering. This takes them to the survey form to be reviewed. The staff **cannot** make any changes to the generic questions. However, they can add/delete one or more optional questions. Once they have reviewed the survey, the changes are saved.

(3) Authentication Module

The security infrastructure will be implemented by an authentication module. The responsibilities of the authentication module are specified below:

- The authentication module is provided with a csv file **"passwords.csv"** that will contain the **zIDs**, **passwords** and **role** ("staff or student") for all users (student and staff). The headers are:

  | zID | Password | role

  The authentication module will import these security details into the database. Additionally, the authentication module will store the authentication details for the admin user in the database.

- The authentication module will implement the following workflow process to authenticate a user to access the survey application.

  Workflow Process 1: User-Authentication (Login a user (student, staff or admin))

  (1) User launches the home-page of the online survey system
 (2) User is presented with a login form where they will input their username and password

(3) The survey application **passes** the security credentials to the **authentication module** which authenticates their input credentials with the stored credentials on the database.

(4) If the user is successfully authenticated, the authentication module **will pass the "role" information** of the logged-in user to the survey application. The survey system takes the user to the appropriate dashboard based on their role as outlined below:

   a. If the user is a "student", then the survey system displays a student dashboard showing the list of course-offerings that the student is currently "enrolled" in and associated with a "open" survey

   b. If the user is an "admin", then the survey system displays an admin dashboard

   c. If the user is a "staff", then the survey system displays a staff dashboard that shows the list of course-offerings currently taught by that staff and associated with surveys in "review" stage.

(5) If the authentication is unsuccessful, the module returns an "invalid" authentication response to the survey system. The survey system then requests the user to re-enter their username and password.

(4) The online survey system

The survey system will retain all of the features from the previous iteration with some enhancements which are outlined below.

Workflow Process 2: Creation of the survey

The creation of the survey will follow a workflow that transitions through different stages as depicted in the workflow diagram below.
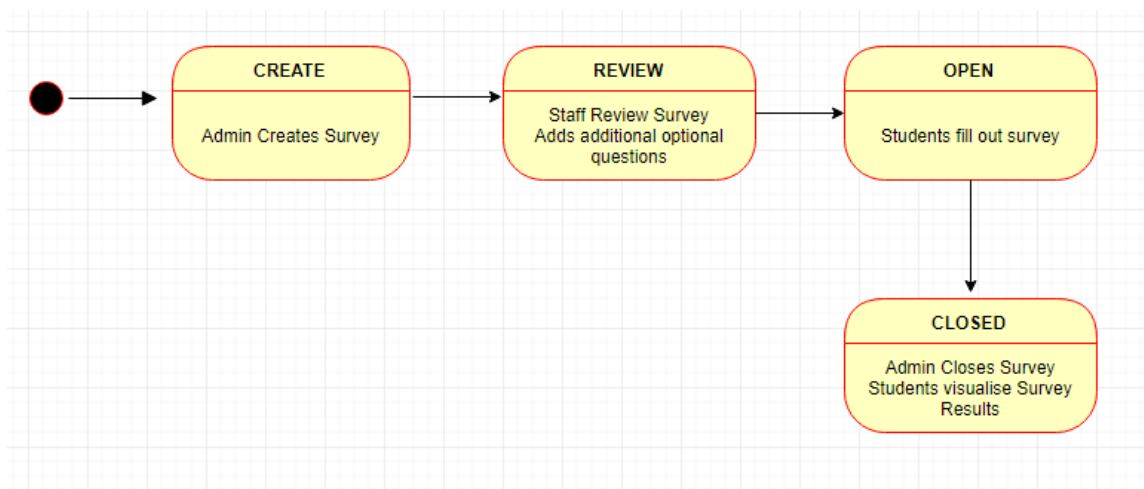
(1)   An admin user initiates the workflow process for the creation of a survey for a particular course-offering by clicking on a link e.g., "Add a survey".

(2)   Admin uses a part or all of the questions from the mandatory question pool and also specifies a start-date and an end-date for the survey. The survey is currently in *Create* phase.   The survey is not visible to both students and "staff-in-charge" at this stage.

(3)   Once the survey has been saved by the admin, the survey is automatically assigned to the "staff-in-charge" of the course offering and proceeds to the next phase, *Review*.

(4)   During the *Review* phase, the form created by the admin will appear on the designated staff member's dashboard. The designated staff will log into the survey system, review the survey and save the changes. The survey is not visible on the student's dashboard at this stage. The staff member will then be able to access the survey form and add questions to the survey form from the

optional pool of questions. The staff member will be able to see the questions added by the admin but will not be able to alter them in any way. Once done, the staff member can release the survey form to the students. The survey proceeds to the **Open** phase.

(5) The survey will now appear on the dashboard of any student, enrolled in that course offering. Enrolled students will be able to log in and take the survey. Upon completion of the survey, the student submits the responses. The responses will be saved to the database. The completed survey will disappear from the student's dashboard to prevent them from re-taking the survey.

(6) **Closed -** After a certain period of time, the admin can choose to take down the survey. For this iteration, the system will not implement any automated scheduling mechanism. The change of the survey workflow status to **Closed** will be manually initiated by the admin user. Once a survey is **Closed** it will no longer be visible to the students or the staff members. Once a survey is closed, the survey results for that course offering will appear on the staff and student's dashboard for viewing.

Survey Questions and Reponses

- The survey system should also be to handle **two** types of questions, Mandatory and Optional. The admin will be able to add questions of both types and they should appear in their own categories in the admin's pool of questions.

- The survey system should allow an admin to delete a question from the mandatory or optional pool. The survey forms that use the deleted question should still be able to display and take answers for that question. However, the question should no longer appear as a choice when creating new surveys.

- The survey system should now be able to handle questions with different response types (e.g., MCQ, text-based). At a minimal, for the current iteration the survey system should have support for text-based questions in addition to MCQ based questions, but the system should be flexible to accommodate more response types in the future. As before, the MCQ based questions can have the same response options.

- A course-offering (e.g., COMP 1531 17s2) will have only survey form associated with it.

- All survey responses will be stored to the database.

- Survey results stored in the system should not be affiliated with the user to preserve anonymity.

Provided files:

The survey system is provided with **two csv files** by the enrolment system. These files must be loaded into the database by the survey system when the application is started.

1. **"courses.csv"** which will contain all the courses that will use the survey system. The headers are:
   | Course name |

2. **"enrolments.csv"** which will contain the courses that each user is affiliated with.
   The headers are: **| zID | Course |**

Metrics

The website should now have a *Metrics* section for the admin. The admin can access the submissions for any given survey form from *Metrics* dashboard. The name of the course offering should appear under the *Metrics* dashboard as soon as the admin creates the survey form, i.e. when the survey form is in the **Review Stage**. The stats page for a survey form should show all the results from users so far. The board has decided to include graphs at the top of the page to help visualise results quickly. Once a survey has been closed, the results of the survey should be visible from the student's dashboard.