Activity 2

RMIT's enrollment system was used to draw up user stories with acceptance criteria. A user story starts with the pain point of a user. Outlined below are the pain points to be turned into user stories.

First, RMIT's course description landing pages can be notoriously outdated because the href attributes of their links are hard-coded. For example, in the last semester, a great many students were misled into believing that they would study Calculus instead of Statistics in Mathematics for Computing 2 even though they accessed its course description page through none other than the program plan page; the correct course description page was in fact available via one of the hidden sub-links found on the course description landing page that contained outdated information. The software solution to reliably update all the links to course description pages will help not only guide students to correct information but also give them confidence in RMIT's quality of education overall.

Second, when adding a course, students do not know whether they have met the course prerequisites unless they are at pains to find the correct course description page and read the list of prerequisite courses whose names are again hard-coded and thus may have been changed, or some of which are actually discontinued. The software solution to quickly assess whether the student is eligible to enrol in a course based on her academic records and show which prerequisite courses to complete first all on the same page as when adding a course will help improve the student's experience with enrolment.

The pain points are now converted to user stories with acceptance criteria below. It is assumed that the features will be built on the current pages of the RMIT website.

Acceptance Criteria (GWT)

As a: *Student* **I want to:** *find the most up-to-date course description* **So that I can:** *have an accurate understanding of what is taught in the course and prepare for it early if needed.*
Scenario: *Student is taken to a course description page with the latest information* **Given:** *Student navigates to the Program Plan page, Add Course page, or any other page where a link to the course description of her interest can be found* **When:** *Student clicks on the course description link* **Then:** *Student is taken to the most up-to-date course description page*

s3774430 Myeonghoon Sun

Acceptance Criteria (Rule-Oriented)

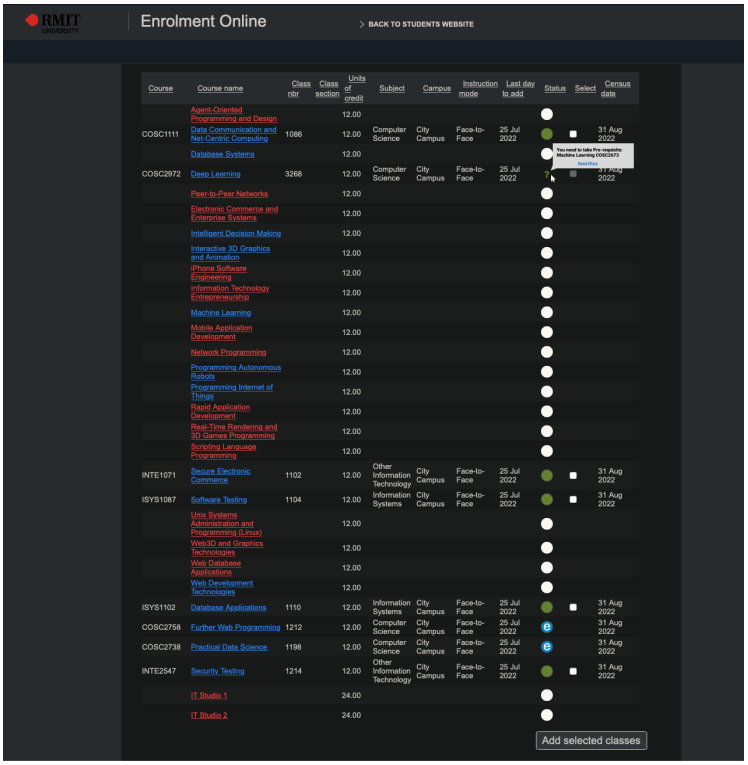

Fig 1: Mock-up page for the feature described in the second user story

As a: *Student*

I want to: *see which courses I am eligible to take from the Add Course page*

So that I can: *only enrol in courses whose enforced prerequisites I have met*

Rules:

- The checkbox for an ineligible course is greyed out and unclickable
- The green question mark is used for its status symbol to prompt the student to bring a mouse cursor over it
- Upon hover, the bubble text box shows why the select button is greyed out.
- The student can find more out how to meet the prerequisite(s) by clicking “Read More”

The Acceptance Criteria (AC) above are sound because they are written from the perspective of a user in vocabulary understandable both to stakeholders and developers, and are to the point as well as testable; first, all the links to course description pages can be programmed to display the most-up-to-date information in the latest term and, therefore, whether they indeed display expected information can be tested; second, whether the rules are complied with can also be tested.

s3774430 Myeonghoon Sun

Activity 3

In software testing, validation differs from verification in that the former's goal is to validate the customer's expectations and requirements for the product whereas the latter's goal is to verify that the product is being built correctly by the developer as per specifications.

In Activity 1, unit testing, a form of verification, was conducted to verify that the function being built works correctly as per the specifications; the unit tests written for the addStudentCourse function have successfully verified that it executes error-free under the required constraints; the first grouped test has confirmed that the maximum number of courses a student can enrol in is 4; the second grouped test has verified that a student cannot enrol in the same course; the third grouped test has confirmed that a course cannot admit more than 100 students.

On the other hand, in Activity 2, user stories with acceptance criteria were utilised not only to document what needs to be built but, most importantly, to validate the customer's requirements and discover such constraints as exemplified above in the explanation of verification, all before delivery. Drawing up such user stories with acceptance criteria as seen in Activity 2 can serve as an opportunity for the developer and the customer to reach a consensus of what is meant by "Done". With validation in place, now the customer knows their course description landing page should contain the most up-to-date information in accordance with the acceptance criteria. The developer now can refer to clear-cut documentation or constraints around which to re-design the course description pages.

Therefore, it is in the synthesis of these two forms of testing that the right product that works right can be developed by the developer and in turn delivered to the customer.