

# MCI project Testing plan

---

Team number: 07

Project Title: Curriculum Mapping Tool

## 1. Introduction

### 1.1 Project Overview

Curriculum Mapping Tool is a convenient tool providing a view and editing function of the structure and relationships amongst degrees and courses in the Computer Science Curriculum to the course leaders. The functionalities of this web application span all systems and browsers. This testing plan identifies the items and the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

### 1.2 Test Objective

The objective of the test is to verify that the functionalities of the Curriculum Mapping Tool according to the requirements and features listed in the below chapter, which include typical activities, i.e., verifying UI/UX in web elements and website layout, data searching, data rendering, and data manipulation, etc. Our aim is to conduct robust automated tests to our product before delivering it to our client.

The final product of the test will include:

- Before testing phase: test plan and test cases.
- During the test phase: test tool, test data, and error logs and execution logs.
- After the testing cycle is over: test reports and defect report.

### 1.3 Test Principles

- There will be common, well defined, consistent procedures for all team members supporting testing activities.
- Testing will be divided into 3 distinct phases, each with clearly defined objectives and goals.
- Testing will be implemented in different environments to emulate a production environment. Table 1 shows the browsers and operating systems where we plan to test.

Browser	Version	Operating System
Google Chrome	Latest stable release	Microsoft Windows 11; MacOS 12
Mozilla Firefox	Latest stable release	Microsoft Windows 11; MacOS 12
Safari	Latest stable release	Microsoft Windows 11; MacOS 12
Microsoft Edge	Latest stable release	Microsoft Windows 11; MacOS 12

Table 1. Planned test environment

## 2. Test Strategy

### 2.1 Test approach

We will adopt a testing approach called Requirements-Based Testing (RBT). The purpose of RBT is to ensure that all the client requirements are satisfied<sup>[1]</sup>. With RBT, we will develop tests to validate whether our website can work as expected by the client. For how we will conduct RBT, please see section 2.3 below.

### 2.2 Test Scope

During development, we use multiple NPM packages. We do not plan to test those packages and assume that they are all functioning correctly. Instead, we will test the website features. Figure 1 shows the two groups of our website features: website UI/UX, and website functionality. UI/UX and functionalities are either approved by the client or derived from

client requirements. Since they are tightly connected to the proposition of the client, we will take all of them as our testing scope.

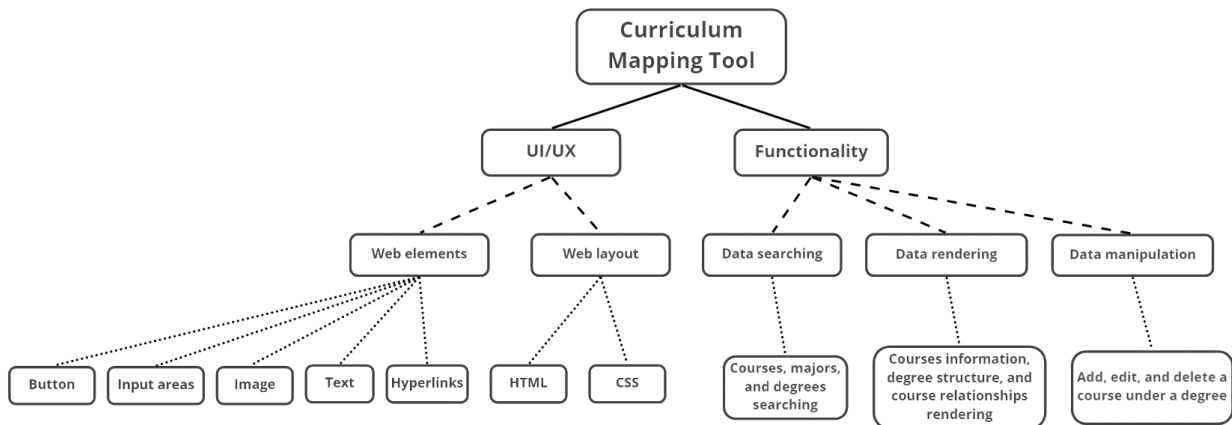


Figure 1. Curriculum Mapping Tool main features overview

### 2.3 RBT Process Overview

We adjusted the general RBT process to make it suitable for our project. This process will take place once a new feature is created.

1. Define test acceptance criteria

Test acceptance criteria should at least be accepted by the client. It will be derived from the existing client requirements. So, if a product can meet the criteria, the corresponding client requirement is considered to be achieved. Appendix 1 shows some examples of how the test acceptance criteria are connected to client requirements.

2. Execute test

Test execution consists of 4 steps:

Step 1: Determine the type of testing according to the website feature to be tested (see section 3.1).

Step 2: Design and build a group of test cases with useful techniques and tools (see section 3.2).

Step 3: Document all the bugs found in testing and report them to the developer (see section 4).

Step 4: After the developer fixes bugs, repeat step 1 to step 3 until all the test cases are passed.

3. Document test result

After performing a test on a new feature (we consider “a test on a new feature” as to run a group of test cases), we will record the test result. The test result may contain the description of the new feature, which client requirement will be realized by this feature, the name of the tester and reviewer, the planned result, and the achieved result. Also, the test result will indicate whether the developer should fix any problems. For details, see section 3.4.

## 3. Execution Strategy

### 3.1 Test Methods

Overall, we take a bottom to top approach which generalized in the three types of tests listed below:

- Unit test. This type of testing will consist of most of our testing cases, we will perform unit testing to check if the functionalities initially built work as per their design in an atomic fashion. Unit testing can help us isolate each part of the program, therefore, identifying the defects from the smallest components. For example, we will use automatic testing tools, e.g., Junit to test the input areas, database connection, link testing etc.

- Integration test. This type of testing will also be performed in a phase which most of the unit testing cases passed, then some individual modules can be tested as a group. For example, we will group ‘input areas’ from the client side and ‘searching functionality’ from the back end to check whether it delivers its output as per expected.

- System test. This type of testing will be performed last for the context needed is the entire system. At this phase, we will evaluate the whole Application whether it is compliant with the specified requirements.

For example, we might mimic user behavior to check whether the workflow behaves as intended.

### 3.2 Test Schedule

We have performed several unit tests and planned to commence the integration test before next Friday (27/05/2022), so that by Week 12, we can perform the system test to exam the application thoroughly before handing it over to our client.

### 3.3 Test Report.

Every week, we rotate the role of tester, who will conduct the test and record the results. Each tester is responsible for following the testing principles, writing test cases, conducting tests and recording the results. Please see the Appendix 2 testing report for the metrics of our testing report. The testing report will be one of our project deliverables when we close the project.

## 4. Defect Tracking & Reporting

The following flowchart depicts defect tracking process and please see Appendix 3 for the bug severity report.

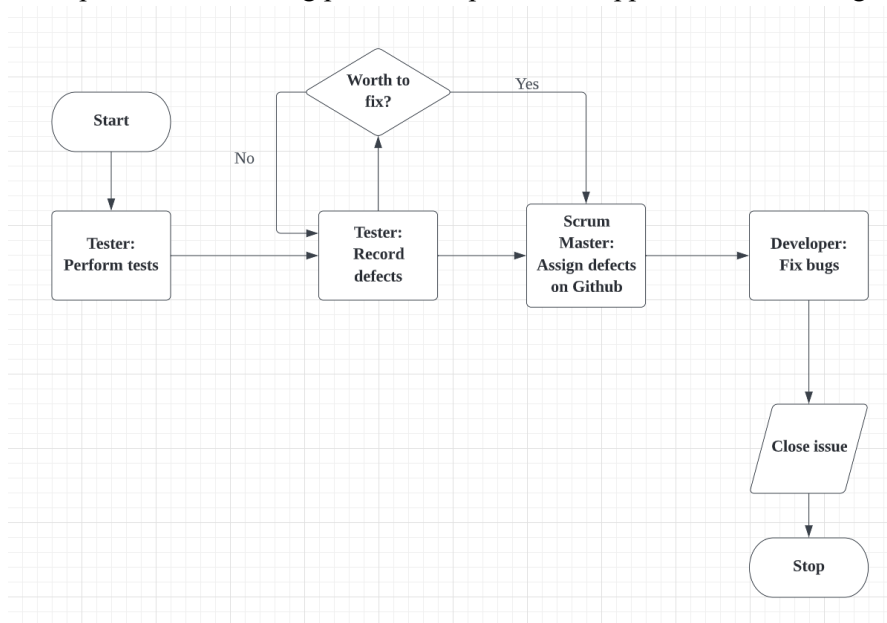


Figure 2. Defect tracking and reporting process in the team

## 5. Risks and Mitigation

As the project progresses to this stage, some of the risks have been identified and the applicable action to mitigate their impact has been considered.

#	Risk	Impact	Trigger	Mitigation Plan
1	Test new defects exists in our application that will affect our project delivery	high	Unexposed defects in the program	If these new defects become showstoppers, it will greatly impact the overall schedule. Person in charge: Ruwen Yi
2	Changes to the original requirements or design	high	Original designs turned out to be not feasible	Requirements or design implements will be closely monitored Person in charge: Ying Liao
3	Delays in training on the automation test tools	high	Developers are more occupied when the semester is about to end	The test schedule will move out in an appropriate number of days. Person in charge: Yingyu Wei
4	Lack of personal resources when testing is to begin	Low	Personal laptops are broken	The team will allocate reasonable resources to this tester. Person in charge: Yuhao Wang

## Reference

Predrag, S, Marija, RS, 2010, 'Requirements-based testing process in practice', *International Journal of Industrial Engineering and Management*, vol. 1, pp. 155-161, viewed 18 May 2022,

<[https://www.researchgate.net/publication/318969267\\_Requirements-based\\_testing\\_process\\_in\\_practice](https://www.researchgate.net/publication/318969267_Requirements-based_testing_process_in_practice)>

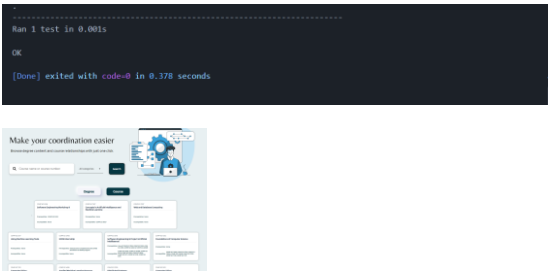
## Appendix 1

### Acceptance Criteria Example

User Stories	Acceptance criteria
As a course coordinator, I want to see all the COMP SCI courses in one page, so that I can have an overview of the COMP SCI courses.	Given that I have opened the website homepage, when I click the “course” button, then all the COMP SCI courses are shown up in card form on the page.
As a course coordinator, I want to know what core courses and elective courses a degree has, so that I can have an overview of the degree structure of a specific degree.	Given that I have opened the website homepage, when I click one of the degree cards on the homepage, then the website jumps to a degree structure page to show the core courses and elective courses of this degree.
As a course coordinator, I want to be able to add new cards to a specific degree, so that I can change the degree structure.	Given that I have opened the degree structure page, when I click the “add” symbol on the page, then the website pops up a window for adding a course. The pop-up window is a form where I should at least enter course name, course code when adding a new course. After I added a new course, it shows on the degree structure page.

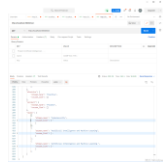
## Appendix 2

### Test Report Example

Scenario Id	B-01	Test scenario	Click event	Tester	Yuhao Wang
Test date	15/5	Linked GitHub issue	#81	URL	http://localhost:5000/
OK/NOK	OK	Test Method	Unit test (Selenium)	Comments	Null
Description	Check the jumping function of the course button on homepage				
Expected result	Unit test script should exit with code=0, and no error message logged. Course function can jump to the right webpage.		Actual output data /result	 Unit test passed without error, and jump correctly.	

### Appendix 3

#### Bug Report Example

Bug ID	Test platform	File attachment	Severity <sup>1</sup>	Status <sup>2</sup>	Closure Reason
B-06	Postman		Major	Closed	Error found and fixed by modifying database.
<b>Description</b>		Data catalogue mistake. Comp sci 3314 course should belong to Artificial Intelligence stream, but it created a new stream. This problem should be caused by wrong blank space in database.			

1 Severity, choice of Critical, Major, Minor and Trivial

2 Status, choice of New, Need More Info, Closed