**Development testing:**

In the development phase of our project, as part of our agile development process to ensure integrity of the project, we will be using technics known as **test driven development**. As with any software development projects, bugs and errors are inevitable during the development process. Thus, before the actual coding process in each component as we write our skeleton code, we will implement test functions for each crucial method with regards to their functionalities, which is known as **unit testing**. We will be using **JUnit** to write the test cases, since all members in our team is familiar with java and the JUnit framework is very well documented. It makes our implementation easier and offers us a versatile utility so that we can carry out more comprehensive testing on our methods.

One major component of our project is how to **render the tube map** on our web application since the idea of our entire project is to make online courses more accessible to people via tube-map like graphical interface. It is critical this map correctly render every time, its underlying logic and algorithm work with different course sets, and no defect would happen in any circumstances.

|  |  |
| --- | --- |
| **Functionality to be tested** | **Testing methods** |
| Basic ability to render the map given a small set of courses under one unit | We plan to use the courses in computer science as a start, and then create small artificial databases. This is the first step of our unit test, and fundament of this component. It should be tested with every major change to our project. |
| Ability to render an actual map with many courses from many units | An extension to the basic testing scheme above, with larger created databases of courses. The size of the database should be close to the requirement of our client. |
| Ability to handle difficult graphs and potential corrupted databases | We will create different databases with courses that are hard to render, e.g. graphs with isolated nod (courses without connections to other courses), graphs with courses that are more often connected to other courses (difficult to render so many connections in one map).  Also, we would create databases with corrupted datapoints, to test the methods robustness and reliability. |
| Ability to handle very large databases of courses | We will create a database with exception large number of courses, which would far exceed the amount required by our client. This is to further ensure the robustness of our code and open up rooms for potential improvement on our algorithm. It might also benefit our client in the future. |

**Release testing:**

As we progress further in our development, we will gradually come close to an actual product. We need to make sure the produced system delivers the expected performance and dependability. We also need to know if the delivered performance meets the actual client need and satisfy all stakeholders of our project as we have expected. Thus, we would implement strategy known as **User story driven development**. We will implement test cases based on each user story provided above, many other potential user stories identified in the development process. We will use **curl** scripts to programmatically call our web application to streamline our releasing testing.

The key user story of our project is “*As a student, I want to be able to see the prerequisites of my chosen course and for what other courses my course is a prerequisite so that I can make an informed decision on what courses I can do or need to do.*” As the key user story, it best represents the purpose of our project – to make courses and knowledge more accessible. It also matches the core implementation of our project. Therefore, it is critical this test offer required performance for the key user story.

|  |  |
| --- | --- |
| **Functionality to be tested** | **Testing method** |
| Ability to correctly display each node on the map via searching courses | Writing test cases to call web api with different searches |
| Ability to zoom in and out to see all relevant nodes | Writing test cases to call web api with different searches, and see if their connected nodes matches the database |
| Ability to scroll up and down for pre/post requisites | Writing test cases to call web api with different searches, do scroll up and down to check if these are implemented. Then also check if the course is a pre/post requisite of the search |
| Ability to handle exceptional flows | Writing test cases with valid inputs that don’t exist and invalid inputs. See if the system provides list of similar courses for valid inputs and error handling for invalid inputs. |
| Ability to satisfy the need of the student from the user story | We will ask other students, and potential other learners (not only students) to search courses that they are interested in our system. We will ask them if the system has made the requirement and learning process for their course clearer, if they have benefited from knowing and studying the connected courses in the map, if it is easier to use than a simple list of courses based on criteria. We will ask their feedback and opinion on improvement. |