Testing

Testing is a crucial aspect of software development that involves systematically evaluating the behaviour, performance, and reliability of a software system. It is applied throughout the development lifecycle to ensure that the software meets the specified requirements and functions as intended. Testing is useful because it helps identify defects, bugs, and inconsistencies in the code, allowing developers to fix them before deployment. It enhances the quality and stability of the software, improves user satisfaction, and reduces the risk of issues or failures in production. By validating the software against expected outcomes and edge cases, testing provides confidence in the system's correctness, functionality, and performance, ultimately leading to a more reliable and successful product.

Application Features

An educational app designed to educate users on phishing attacks would encompass several key features to effectively convey important information and promote a deeper understanding of phishing prevention. Firstly, the app could provide interactive tutorials and simulations that illustrate various phishing techniques and common red flags to watch out for. It may also include real-life case studies and examples to demonstrate the potential consequences of falling victim to phishing attacks. To engage users actively, the app could offer quizzes or interactive games to test their knowledge and reinforce key concepts. Additionally, incorporating a reporting mechanism within the app would empower users to report suspected phishing attempts and contribute to a larger network of threat detection. Overall, a comprehensive educational app on phishing attacks should provide informative content, interactive learning experiences, and ongoing awareness to equip users with the knowledge and skills necessary to protect themselves in today's digital landscape.

System Testing

- unit testing:

For the main feature of the application, the learning of a course, a way to unit test it would be to test the state of the course, whether it is not started, ongoing or completed. Also, another test could check if the retrieved content of the course is the same as the one stored in the database.

- integration testing:

The integration testing would overview the interaction between the modules handling the courses and the sections since a course contains multiple sections. Also the module used for testing the user’s knowledge after learning a course. A top-down approach would be used, staring from checking the provided interface for the user to interact on such as lists, buttons and moving downwards to the domain classes such as course, section.

- system testing:

For system testing, I considered testing the availability of the application given multiple users and how the users find the app in terms of usability and friendliness. In this way, the app would also be tested for reliability and stability.