8.0 System Testing

This section will discuss the four (4) types of testing done, namely, Unit Testing, Integration Testing, System Testing, and Acceptance Testing. These types will also be discussed in detail as to how they were conducted for the developed project management system.

8.1 Unit Testing

This is where developers test individual modules and functions. They are expected to see if they are running as intended. The developers were tasked to individually develop specific modules and functions. During the development period, they had also conducted unit testing for each of the modules and functions they have developed. The developers had implemented white box testing, which is a testing method that has testers know the actual program or function of the code and how it works. This was chosen above black box because it is a quicker way of testing and an easier way for the programmers to spot the error, investigate, and remediate the problem. Another implementation of user testing that the developers used was having other programmers in the team to test the module or function to verify that the program is indeed working as intended. They also evaluate the module or function and give their comments and suggestions to better improve the functionality, user experience, and design.

8.2 Integration Testing

Integration testing is the combination of the fully tested units of the system. This is where you integrate two (2) or more functioning modules to see if they work while coexisting with each other. Through this, the developers can check if the newly developed modules do not disrupt the continuity of the system and more importantly, the process. This type of testing also allows the developers to spot any interface defects between the modules such as redundant elements or differences in the user design and user experience. This also helps the developers gauge the capability of the system to handle more than one (1) module. This was implemented to the developed project management system as the developers had integrated their individual modules. Through this they not only saw that the modules work as intended individually, but that modules that were connected had the intended functionality. Functions such as Task Delegation and Task Completion were cohesive in the business process of the TEI.

8.3 System Testing

This type of testing is where the developers combine all modules and functions together to form the entirety of the system. Here, they check if the system runs all modules and functions as intended without any bugs or crashes. This is where the efforts of the programmers on the individual parts of the system come together to see the totality of its effect on the whole of the company's processes. Through this, the developers can see if all modules and functions are full working and can coexist in the system environment. As regards to the project management system that was developed, the development team had done this by dedicating a sit-down session to run the entire system. The developers used test cases that they had created which involves the modules and functions as well as their expected results. These test cases were then executed to see if the system gave the result they were expecting. System testing also included the flow in between processes, such as creating a project, delegating tasks, and accomplishing the tasks. The flow of the process was then compared to the actual business process if the results were as intended or better improved. The development team also tested the ease of use or user experience, if the system was easy to use and easy to understand for their intended users. Lastly, they ran a real business scenario through the system to see how it works in a business setting and if it would properly function.

8.4 Acceptance Testing

Acceptance Testing is the last type of testing that has actual users of the system test the system to see if they are satisfied with how it operates and if it coincides with their business processes. This testing is perhaps the most crucial, as this is where the efforts of the development team are truly evaluated on whether the system developed was indeed inline with their requirements and specifications. This type of testing is oftenly called User Acceptance Testing (UAT).

This was applied to the project management system by having the users test the system themselves. They were then given a form to fill up that had functions relating to their role as a user, may it be an Executive, a Department Head, a Supervisor, or a Staff employee *(Refer to Appendix V)*. These functions were the rated from one (1) to five (5), with five (5) being the highest rating. The users also had a section wherein they were free to give their feedback regarding the system. The developers had nine (9) users to test the system, 3 Departement Heads, 3 Supervisors, and 3 Staff employees. The developers were not able to have an Executive test the system as they were not available at the time they were conducting the UAT.

The UAT form was divided into five (5) sections, namely Data Security, System Functionality, which was different for each other user types, Reports, User Interface Design, and General Project Considerations. For Data Security, across the nine (9), they had an average of 4.97 for the four (4) factors under it. For System Functionality, Department Head's had a total average of 4.63 for 29 factors *(Refer to Table 8-2)*. As for Supervisors, the total average for 29 factors is 4.70 *(Refer to Table 8-3)*. For the Staff, they had 19 factors with a total average of 4.77 *(Refer to Table 8-4)*. The Reports section had an average of 4.51, the User Interface Design had an average of 4.53, and lastly, General Project Considerations had a total average of 4.74 *(Refer to Table 8-1)*.

8.5 Test Cases

Test cases are made to serve as a guide for not only developers but testers as well as these outlines how the system should be tested to ensure that each module is run through. These cases also include the expected result for certain actions as well as error checking and validation for the system

The developers had created test cases to run through during system testing. These cases helped the developers test the system as thorough as possible, making sure these cases went through all major functions and modules the project management system in question has. The developers have created fourteen (14) sections, covering every module and function, for the test cases for the developed system *(Refer to Appendix U)*.