System Level Testing

System Level Testing is different than unit testing in the sense that where unit testing tests individual parts of a program, system-level testing tests the program as a whole. Therefore, typically this system-level testing tends to happen later on in the project, after the integration steps have been taken. The biggest part and purpose for the system level testing is to make sure a program meets the specifications or requirements for the project.

There are several different kinds of tests that can be performed on a system level to test a program, such as:

*GUI Testing* – This kind of testing is to make sure the features and elements of the Graphical User Interface. There are many different ways this can be done, such as simply running the programs and going through various actions to see how the GUI responds and make sure each action does what it is supposed to. This can be a problem, however because especially with larger programs there can be hundreds or even hundred of thousands of GUI elements, actions and sequence combinations. Due to this, it is preferred that tests are automated and programs are written to simulate user behavior and run test cases

*Usability Testing* – This kind of testing is centered around the user experience. This is primarily performed by end users to get their input as to whether the product is usable / meets expectations and requirements.

*Load Testing* – Typically done with web servers or online services, this kind of testing is designed to test how much the program or service can handle. This is good for stress testing to see what will happen if many people try to use the service at once. That way, you can make sure that the service is up and running when people need it.

*Security Testing* – This kind of testing has primarily concerned with the security of a program or system. This means that the information in the program is secure and the program cannot be modified or exploited in any way by outside users. This can be done in several different ways, either through something like a vulnerability scan where someone comes and audits the system and looks for areas that may be lacking, or though a penetration test where someone would actively try to break into the system and point out any flaws if they can.

These are just a few examples of system-level tests that can be performed in order to make sure a system is functional and meets the necessary requirements based on what was defined originally in the project.