Unit testing:

Unit testing is a software testing technique in which individual units or components of a software system are tested in isolation from the rest of the system. The purpose of unit testing is to verify that each unit of the software system is working as expected and meets the specified requirements.

A real-life example of unit testing is testing a calculator application. In this case, each function or method of the calculator application can be tested individually to ensure that it performs the expected calculation accurately. For example, the add() function can be tested by passing in two numbers and verifying that the result is correct. Similarly, the subtract(), multiply(), and divide() functions can be tested individually.

Integration testing:

Integration testing is a software testing technique that involves testing the interaction between different components or modules of a software system. The purpose of integration testing is to identify any defects that may arise due to the integration of different components and ensure that they work together as expected.

A real-life example of integration testing is testing a banking application that consists of different modules such as customer account management, transaction processing, and reporting. In this case, the interaction between these modules can be tested to ensure that they are working together correctly. For example, a test scenario can be created to verify that when a customer makes a deposit, the transaction is recorded correctly in the customer's account and reflected in the reporting module.

System testing:

System testing is a software testing technique that involves testing the entire system as a whole to ensure that it meets the specified requirements and performs as expected. The purpose of system testing is to identify any defects that may arise due to the integration of different components and ensure that the system meets the user's needs.

A real-life example of system testing is testing a web-based e-commerce platform. In this case, the entire system can be tested to ensure that it performs as expected, including the functionality of the shopping cart, checkout process, payment processing, and order fulfillment. The system can be tested in different environments, such as different browsers and operating systems, to ensure that it works correctly in various scenarios.

Acceptance testing:

Acceptance testing is a software testing technique that involves testing the system with real data and user scenarios to ensure that it meets the user's requirements and expectations. The purpose of acceptance testing is to verify that the system is ready for deployment and use by the end-user.

A real-life example of acceptance testing is testing a mobile banking application. In this case, the application can be tested with real user scenarios, such as checking account balances, making transfers, and paying bills. The acceptance testing can be done in collaboration with the end-users to ensure that the application meets their needs and expectations. Any issues identified during acceptance testing can be addressed before the application is released to the end-users.