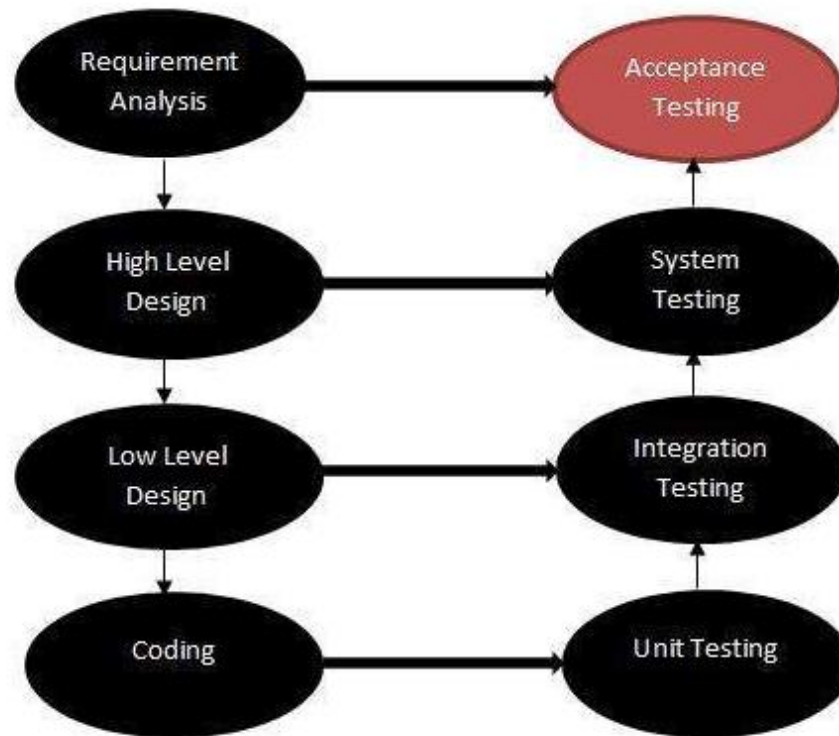


Software Testing Analysis



1. Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

The basic and important testing is unit testing as it ensures that whether each component is working properly or not. We tested each of the functionalities implemented individually starting from frontend text boxes, buttons to backend functions. Each part was a unit to us.

This testing was **White Box Testing** as we went through each and every implemented component. This was our first testing for our project, and we wanted to ensure that these work properly as the future implementations depend on these. So, we tested every component thoroughly while building it.

2. Integration Testing

Upon completion of unit testing, the units or modules are to be integrated which gives rise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

Four approaches to Integration testing:

1. Bottom Up Integration

Bottom-up testing is an approach to **integrated testing** where the lowest level components are **tested** first, then used to facilitate the **testing** of higher-level components. The process is repeated until the component at the top of the hierarchy is **tested**.

2. Top Down Integration

Top-down integration testing is an integration testing technique used in order to simulate the behavior of the lower-level modules that are not yet integrated

3. Big Bang Integration

Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

4. Sandwich Integration

Sandwich Testing is the combination of bottom-up approach and top-down approach, so it uses the advantage of both bottom up approach and top down approach. Initially it uses the stubs and drivers where stubs simulate the behavior of missing component. It is also known as the **Hybrid Integration Testing**.

We followed **Sandwich Integration Testing**. Our project consists of a web App and mobile App. Both the apps were implemented simultaneously to ensure we don't fall behind. For web App one of us started working for frontend and other one was working with the functions which should be implemented to communicate with backend. Later we integrated the frontend with the functions implemented. So here the **front end** was implemented in **Top Down** approach where a basic template was first created and each of the required components were handled individually and the **backend** functionalities in **bottom up** fashion where multiple functions were implemented and integrated them as a component. Similar fashion was followed for mobile app too.

This strategy of sandwich integration helped us to work in parallel and was time saving too.

3. System Testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

Before the final meeting with the client, we performed testing for each of the functionalities for different kind of scenarios possible. We ensure that all the functionalities worked perfectly fine. We also tested the users experience with the app by asking one of our friends to use the app who was not familiar with the app previously. This testing phase was an important one as it made us confident to go ahead for final meeting with the client.

Different types of **system testing** performed:

1. **Usability Testing** - mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives. This was checked by making a person to use the app who was previously unknown to it.
2. **Load Testing** - is necessary to know that a software solution will perform under real-life loads. We performed this by loading the data continuously using the app from multiple phones at a time.
3. **Regression Testing** - involves testing done to make sure none of the changes made over the course of the development process have caused new bugs. It also makes sure no old bugs appear from the addition of new software modules over time.

4. Acceptance Testing

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the SRS requirements and verify if it has met the required criteria for delivery to end users.

Acceptance testing was performed manually. This testing is one of the most important part of the project as this is the final testing which determines whether the product developed met the client's requirements and expectations. For that we manually gave a demo to our client of both the android app and web app. Multiple users were able to use both the apps at the same time in front of the client. The features were also tested for different scenarios in front of client itself.

This testing was **Black Box Testing** as this testing was done for the client and their target audience.